BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ISO 9001:2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

PHÂN CỤM DỮ LIỆU

BÀI TOÁN VÀ CÁC GIẢI THUẬT THEO TIẾP CẬN PHÂN CẤP

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

PHÂN CỤM DỮ LIỆU

BÀI TOÁN VÀ CÁC GIẢI THUẬT THEO TIẾP CẬN PHÂN CẤP

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Giáo viên hướng dẫn: PGS.TS Nguyễn Thanh Tùng
Sinh viên: Phạm Ngọc Sâm
Mã sinh viên: 1351010049

HẢI PHÒNG - 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o------

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
-------o0o-------

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Phạm Ngọc Sâm Mã sinh viên: 1351010049

Lớp*:* CT1301 Ngành: Công nghệ thông tin

Tên đề tài:

Phân cụm dữ liệu: Bài toán và các giải thuật theo tiếp cận phân cấp

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp.

   a. Nội dung:

   - Thế nào là khai phá dữ liệu, khám phá tri thức từ cơ sở dữ liệu.
   - Kỹ thuật phân cụm dữ liệu trong khai phá dữ liệu, phân loại các thuật toán phân cụm và các lĩnh vực ứng dụng tiêu biểu.
   - Một số thuật toán phân cụm theo tiếp cận phân cấp: Thuật toán CURE, thuật toán BIRCH.
   - Xây dựng chương trình demo một trong số các thuật toán phân cụm phân cấp trình bày.

   b. Các yêu cầu cần giải quyết:

   - Về lý thuyết: Nắm được các nội dung 1-3 trong mục a.
   - Về thực hành: Xây dựng được chương trình demo một trong số các thuật toán phân cụm phân cấp trình bày.

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập tốt nghiệp.

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: **Nguyễn Thanh Tùng**

Học hàm, học vị: *Phó giáo sư, Tiến sĩ.*

Cơ quan công tác: *Nguyên cán bộ nghiên cứu Viện Khoa học và Công nghệ Việt Nam.*

Nội dung hướng dẫn:

.....

.....

.....

.....

.....

.....

.....

.....

Đề tài tốt nghiệp được giao ngày 25 tháng 03 năm 2013

Yêu cầu hoàn thành xong trước ngày 25 tháng 06 năm 2013

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên	*Người hướng dẫn Đ.T.T.N*
Phạm Ngọc Sâm	PGS.TS Nguyễn Thanh Tùng

Hải phòng, ngày……tháng….năm 2013

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

.....

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu…):

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn (ghi bằng cả số và chữ):

.....

.....

.....

.....

.....

.....

.....

Hải phòng, ngày …tháng …năm 2013

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

PHIẾU NHẬN XÉT TÓM TẮT CỦA NGƯỜI CHẤM PHẢN BIỆN

1. Đánh giá chất lượng đề tài tốt nghiệp về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh và bản vẽ, giá trị lý luận và thực tiễn của đề tài.

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

1. Cho điểm của cán bộ phản biện (ghi cả số và chữ)

……
……
……
……
……
……
……
……
……
……

Hải Phòng, ngày…tháng … năm 2013

Cán bộ phản biện

LỜI CẢM ƠN

Với lòng biết ơn sâu sắc, tôi xin chân thành cảm ơn thầy giáo PGS.TS **Nguyễn Thanh Tùng** đã định hướng và giúp đỡ tôi tận tình trong suốt quá trình làm khóa luận.

Tôi xin chân thành cảm ơn các thầy, cô giáo khoa Công nghệ thông tin đã truyền dạy những kiến thức thiết thực trong suốt quá trình học, đồng thời tôi xin cảm ơn nhà trường đã tạo điều kiện tốt nhất cho tôi hoàn thành khóa luận này.

Trong phạm vi hạn chế của một khóa luận tốt nghiệp, những kết quả thu được còn là rất ít và quá trình làm viêc khó tránh khỏi những thiếu sót, tôi rất mong nhận được sự góp ý của các thầy cô giáo và các bạn.

Hải phòng, ngày 25 tháng 06 nắm 2013

Sinh viên

Phạm Ngọc Sâm

DANH MỤC HÌNH VÀ CÁC CHỮ VIẾT TẮT

CSDL: Cơ sở dữ liệu.

KDD: Khai phá tri thức trong cơ sở dữ liệu - Knowledge Discovery in Databases.

PCDL: Phân cụm dữ liệu

CF: Cluster Features

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

CURE (Clustering Using Representatives)

MỤC LỤC

LỜI MỞ ĐẦU

Trong những năm gần đây, cùng với sự phát triển vượt bậc của công nghệ điện tử và truyền thông, khả năng thu thập và lưu trữ thông tin của các hệ thống thông tin không ngừng được nâng cao. Theo đó, lượng thông tin được lưu trữ trên các thiết bị nhớ không ngừng tăng lên. Khai phá dữ liệu là một lĩnh vực khoa học mới xuất hiện, nhằm tự động hóa việc khai thác những thông tin, những tri thức tiềm ẩn, hữu ích từ những CSDL lớn cho các đơn vị, tổ chức, doanh nghiệp,… từ đó làm thúc đẩy khả năng sản xuất, kinh doanh, cạnh tranh cho các đơn vị, tổ chức này. Những ứng dụng thành công trong khám phá tri thức, cho thấy khai phá dữ liệu là một lĩnh vực phát triển bền vững mang lại nhiều lợi ích và có nhiều triển vọng, đồng thời có ưu thế hơn hẳn so với các công cụ phân tích dữ liệu truyền thống. Hiện nay, khai phá dữ liệu đã và đang được ứng dụng ngày càng rộng rãi trong các lĩnh vực như: *thương mại, tài chính, điều trị y học, viễn thông, tin-sinh.*

Một trong những hướng nghiên cứu chính của khai phá dữ liệu là phân cụm dữ liệu (Data Clustering). Phân cụm dữ liệu là quá trình tìm kiếm và phát hiện ra các cụm dữ liệu tự nhiên tiềm ẩn trong cơ sở dữ liệu lớn, từ đó cung cấp thông tin, tri thức hữu ích cho việc ra quyết định. Có rất nhiều kĩ thuật trong phân cụm dữ liệu như: phân cụm dữ liệu phân hoạch, phân cụm dữ liệu phân cấp, phân cụm dựa trên mật độ,.. Tuy nhiên các kĩ thuật này đều hướng tới hai mục tiêu chung đó là *chất lượng* các cụm khám phá được và *tốc độ* thực hiện của thuật toán. Trong đó, kĩ thuật phân cụm dữ liệu phân cấp là một kĩ thuật có thể đáp ứng được những mục tiêu này và có khả năng làm việc với các CSDL lớn.

Nghiên cứu và ứng dụng một cách hiệu quả các phương pháp khai phá dữ liệu là vấn đề hấp dẫn, đã và đang thu hút sự quan tâm chẳng những của các nhà nghiên cứu, ứng dụng mà của cả các tổ chức, doanh nghiệp. Do đó, em đã chọn đề tài nghiên cứu "*Phân cum dữ liệu: Bài toán và các giả thuật theo tiếp cận phân cấp"* cho đồ án tốt nghiệp của mình.

Nội dung của đồ án gồm 3 chương:

Chương 1: Khái quát về khai phá dữ liệu: Trong chương này em trình bày tổng quan về khai phá dữ liệu, quy trình khai phá, các kỹ thuật khai phá và các ứng dụng của khai phá dữ liệu, cuối cùng là các thách thức đặt ra.

Chương 2: Trình bày về các phương pháp phân cụm dữ liệu, trong đó đồ án đi sâu vào tìm hiểu về phương pháp phân cụm phân cấp với 2 thuật toán điển hình là: BIRCH và CURE.

Chương 3: Chương trình demo: Để khẳng định cho khả năng và hiệu quả của thuật toán phân cụm phân cấp, xây dựng một chương trình demo đơn giản sử dụng thuật toán CURE.

Cuối cùng là phần kết luận trình bày tóm tắt các kết quả thu được và các đề xuất cho hướng phát triển của đề tài.

CHƯƠNG I: KHÁI QUÁT VỀ KHAI PHÁ DỮ LIỆU

1.1 Khai phá dữ liệu (Data Mining) là gì?

Với sự phát triển nhanh chóng và vượt bậc của công nghệ điện tử và truyền thông, khả năng lưu trữ thông tin không ngừng tăng. Theo đó lượng thông tin được lưu trữ trên các thiết bị nhớ cũng tăng cao. Với sự ra đời và phát triển rộng khắp của cơ sở dữ liệu (CSDL) đã tạo ra sự "*bùng nổ*" thông tin trên toàn cầu, một khái niệm về *"khủng hoảng"* phân tích dữ liệu tác nghiệp để cung cấp thông tin có chất lượng cho những quyết định trong các tổ chức tài chính, thương mại, khoa học… đã ra đời từ thời gian này. Như John Naisbett đã cảnh báo *"Chúng ta đang chìm ngập trong dữ liệu mà vẫn đói tri thức"*.

Dữ liệu không phải là cái quan trọng mà là thông tin từ dữ liệu, chính vì vậy một lĩnh vực khoa học mới xuất hiện giúp tự động hóa khai thác những thông tin, tri thức hữu ích, tiềm ẩn trong các CSDL chính là *Khai phá dữ liệu (Data Mining)*.

Khai phá dữ liệu là một lĩnh vực khoa học tiềm năng, mang lại nhiều lợi ích, đồng thời có ưu thế hơn hẳn so với các công cụ phân tích dữ liệu truyền thống. Hiện nay, khai phá dữ liệu được ứng dụng rộng rãi trong các lĩnh vực: *Phân tích dữ liệu hỗ trợ ra quyết định, điều trị y học, tin-sinh học, thương mại, tài chính, bảo hiểm, text mining, web mining…*

Do sự phát triển nhanh về phạm vi áp dụng và các phương pháp tìm kiếm tri thức, nên có nhiều quan điểm khác nhau về khái niệm khai phá dữ liệu. Ở một mức trừu tượng nhất định, chúng ta có định nghĩa về khai phá dữ liệu như sau:

"*Khai phá dữ liệu là quá trình tìm kiếm, phát hiện các tri thức mới, hữu ích tiềm ẩn trong cơ sở dữ liệu lớn"*.

1.2 Quy trình khai phá dữ liệu.

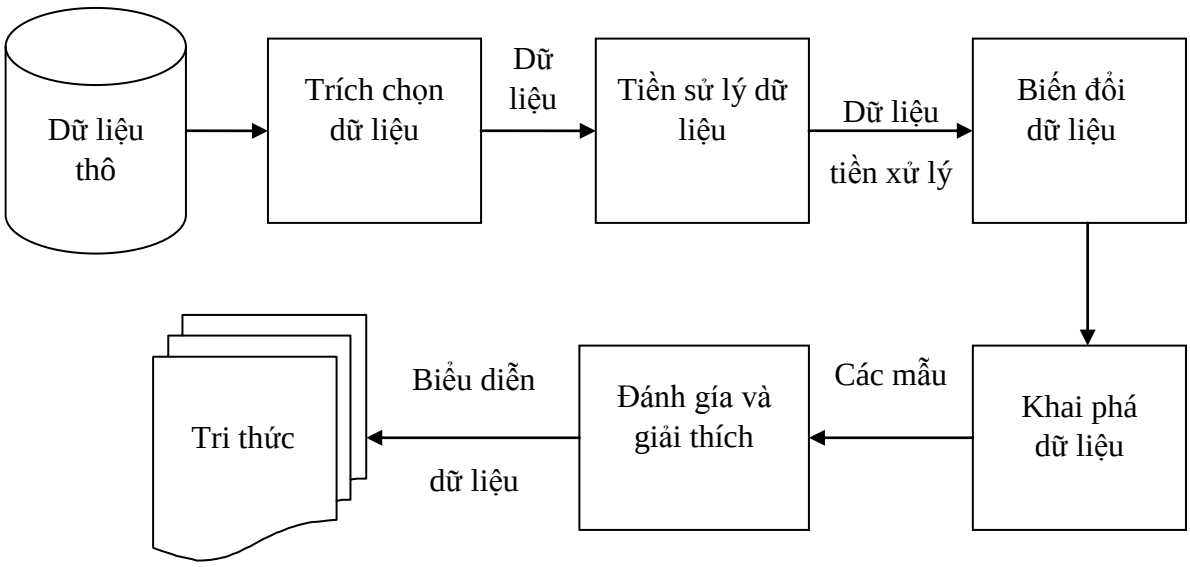
Khám phá tri thức trong CSDL (Knowledge Discovery in Databases – KDD) là mục tiêu chính của khai phá dữ liệu, do vậy khái niệm về khai phá dữ liệu và KDD được xem là tương đương nhau. Tuy nhiên, nếu phân chia một cách chi tiết thì khai phá dữ liệu là một bước chính trong quá trình KDD.

Khám phá tri thức trong CSDL là lĩnh vực liên quan đến nhiều ngành như: *Tổ chức dữ liệu, xác suất, thống kê, lý thuyết thông tin, học máy, CSDL, thuật toán, trí tuệ nhân tạo, tính toán song song và hiệu năng cao,…* Các kỹ thuật chính áp dụng trong khám phá tri thức phần lớn được thừa kế từ các ngành này.

Quá trình khám phá tri thức có thể phân ra các công đoạn sau:

- *Trích lọc dữ liệu:* Là bước tuyển chọn những tập dữ liệu cần được khai phá từ các tập dữ liệu lớn (databases, data warehouses, data repositories) ban đầu theo một số tiêu chí nhất định.
- *Tiền xử lý dữ liệu:* Là bước làm sạch dữ liệu (xử lý dữ liệu thiếu, dữ liệu nhiễu, dữ liệu không nhất quán,…), tổng hợp dữ liệu (nén, nhóm dữ liệu, xây dựng các histograms, lấy mẫu, tính toán các tham số đặc trưng,…), rời rạc hóa dữ liệu, lựa chọn thuộc tính… Sau bước tiền xử lý này dữ liệu sẽ nhất quán, đầy đủ và được rút gọn lại.
- *Biến đổi dữ liệu:* Là bước chuẩn hóa và làm mịn dữ liệu để đưa dữ liệu về dạng thuận lợi nhất nhằm phục vụ việc áp dụng các kỹ thuật khai phá.
- *Khai phá dữ liệu:* Là bước áp dung những kỹ thuật phân tích (phần nhiều là các kỹ thuật học máy) nhằm khai thác dữ liệu, trích lọc những mẫu tin (information patterns), những mối quan hệ đặc biệt trong dữ liệu. Đây được xem là bước quan trọng và tiêu tốn thời gian nhất của toàn bộ quá trình KDD.
- *Đánh giá và biểu diễn tri thức:* Những mẫu thông tin và mối quan hệ trong dữ liệu đã được phát hiện ở bước khai phá dữ liệu được chuyển sang và biểu diễn ở dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật,… Đồng thời bước này cũng đánh giá những tri thức khai phá được theo những tiêu chí nhất định.

Hình 1.1 dưới đây mô tả các công đoạn của KDD.



Hình 1.1. Các bước thực hiện quá trình khai phá dữ liệu

1.3 Các kỹ thuật khai phá dữ liệu.

Theo quan điểm máy học (Machine Learning) thì các kỹ thuật khai phá dữ liệu bao gồm:

- *Học có giám sát (Supervised Learning):* Là quá trình phân lớp các đối tượng trong cơ sở dữ liệu dựa trên một tập các ví dụ huấn luyện về các thông tin về nhãn lớp đã biết.
- *Học không có giám sát (Unsupervised Learning):* Là quá trình phân chia một tập các đối tượng thành các lớp hay cụm (Cluster) tương tự nhau mà không biết trước các thông tin về nhãn lớp.
- *Học nửa giám sát (Semi-Supervised Learning):* Là quá trình phân chia một tập các đối tượng thành các lớp dựa trên một *tập nhỏ các ví dụ huấn luyện* với thông tin về *nhãn lớp* đã biết

Nếu căn cứ vào các lớp bài toán cần giải quyết, thì khai phá dữ liệu bao gồm các kỹ thuật sau

- *Phân lớp và dự đoán (Classification and prediction):* Là việc xếp các đối tượng vào những lớp đã biết trước. Ví dụ, phân lớp các bệnh nhân, phân lớp các loài thực vật,…Hướng tiếp cận này thường sử dụng một số kỹ thuật của học máy như cây quyết định (decision tree), mạng nơ-ron nhân tạo (neural network),… Phân lớp và dự đoán còn được gọi là học có giám sát.
- *Phân cụm (Clustering/Segmentation):* Là việc xếp các đối tượng theo từng cụm tự nhiên.
- *Luật kết hợp (Association rules):* Là *việc* phát hiện các luật biểu diễn tri thức dưới dạng khá đơn giản. Ví dụ: "70% nữ giới vào siêu thị mua phấn thì có tới 80% trong số họ cũng mua thêm son".
- *Phân tích hồi quy (Regression analysis):* Là *việc* học một hàm ánh xạ từ một tập dữ liệu thành một biến dự đoán có giá trị thực. Nhiệm vụ của phân tích hồi quy tự như của phân lớp, điểm khác nhau là ở chỗ thuộc tính dự báo là liên tục chứ không rời rạc.
- *Phân tích các mẫu theo thời gian (Sequential/Temporal patters): Tương* tự như khai phá luật kết hợp nhưng có quan tâm đến tính thứ tự theo thời gian.

- *Mô tả khái niệm và tổng hợp (Concept description and summarization*): Thiên về mô tả, tổng hợp và tóm tắt các khái niệm. Ví dụ*: tóm tắt văn bản.*

Hiện nay, các kỹ thuật khai phá dữ liệu có thể làm việc với rất nhiều kiểu dữ liệu dữ liệu khác nhau. Một số dạng dữ liệu điển hình là: CSDL giao tác, CSDL quan hệ hướng đối tượng, dữ liệu không gian và thời gian, CSDL đa phương tiện, dữ liệu văn bản và web,…

1.4 Các ứng dụng của khai phá dữ liệu.

Như đã nói ở trên, khai phá dữ liệu là một lĩnh vực liên quan tới nhiều ngành khoa học như: *hệ CSDL, thống kê, trực quan hóa*… Hơn nữa, tùy vào cách tiếp cận được sử dụng, khai phá dữ liệu có thể áp dụng một số kỹ thuật như mạng nơ-ron, phương pháp hệ chuyên gia, lý thuyết tập thô, tập mờ,…So với các phương pháp này, khai phá dữ liệu có một số *ưu thế* rõ rệt.

- Phương pháp học máy chủ yếu được áp dụng đối với các CSDL đầy đủ, ít biến động và tập dữ liệu không quá lớn. Trong khi đó, các kỹ thuật khai phá dữ liệu có thể được sử dụng đối với các CSDL chứa nhiễu, dữ liệu không đầy đủ hoặc biến đổi liên tục.
- Phương pháp hệ chuyên gia được xây dựng dựa trên những tri thức cung cấp bởi các chuyên gia. Những dữ liệu này thường ở mức cao hơn nhiều so với những dữ liệu trong CSDL khai phá, và chúng thường chỉ bao hàm được các trường hợp quan trọng. Hơn nữa, giá trị và tính hữu ích của các mẫu phát hiện được bởi hệ chuyên gia cũng chỉ được xác nhận bởi các chuyên gia.
- Phương pháp thống kê là một trong những nền tảng lý thuyết của khai phá dữ liệu, nhưng khi so sánh hai phương pháp với nhau có thể thấy các phương pháp thống kê có một số điểm yếu mà chỉ khai phá dữ liệu mới khắc phục được.

Với những ưu điểm trên, khai phá dữ liệu hiện đang được áp dụng một cách rộng rãi trong nhiều lĩnh vực kinh doanh và đời sống như: *marketing, tài chính, ngân hàng và bảo hiểm, khoa học, y tế, an ninh internet…* Rất nhiều tổ chức và công ty lớn trên thế giới đã áp dụng thành công kỹ thuật khai phá dữ liệu vào các hoạt động sản xuất, kinh doanh của mình và thu được những lợi ích to lớn. Các công ty phần mềm lớn trên thế giới cũng rất quan tâm và chú trọng tới các công cụ khai phá dữ liệu vào bộ Oracle9i, IBM đã đi tiên phong trong việc phát triển các

ứng dụng khai phá dữ liệu với các ứng dụng khai phá dữ liệu với các ứng dụng như Intelligence Miner…

Các ứng dụng này được chia thành 3 nhóm ứng dụng khác nhau và quản lý khách hàng, cuối cùng là các ứng dụng vào phát hiện và xử lý lỗi hệ thống mạng.

Nhóm 1: Phát hiện gian lận (Fraud detection)

Gian lận là một trong những vấn đề nghiêm trọng đối với các công ty viễn thông, nó có thể làm thất thoát hàng tỷ đồng mỗi năm. Có thể chia ra làm 2 hình thức gian lận khác nhau thường xảy ra đối với các công ty viễn thông

Trường hợp thứ nhất xảy ra khi một khách hành đăng ký thuê bao với ý định không bao giờ thanh toán khoản chi phí sử dụng dịch vụ.

Trường hợp thứ hai liên quan đến một thuê bao hợp lệ nhưng lại có một số hoạt động bất hợp pháp gây ra bởi một người khác

Những ứng dụng này sẽ thực hiện theo thời gian thực bằng cách sử dụng dữ liệu chi tiết cuộc gọi, một khi xuất hiện một cuộc gọi nghi ngờ gian lận, lập tức hệ thống sẽ có hành động ứng xử phù hợp, ví dụ như một cảnh báo xuất hiện hoặc từ chối gọi nếu biết đó là cuộc gian lận.

Hầu hết các phương thức nhận diện gian lận đều dựa vào việc so sánh hành vi sử dụng điện thoại của khách hàng trước kia với hành vi hiện tại để xác định xem đó là cuộc gọi hợp lệ không.

Nhóm 2: Các ứng dụng quản lý và chăm sóc khách hành

Các công ty viễn thông quản lý một khối lượng lớn dữ liệu về thông tin khách hàng và chi tiết cuộc gọi (call detail records). Những thông tin này có thể cho ta nhận diện được những đặc tính của khách hàng và thông qua đó có thể đưa ra các chính sách chăm sóc khách hàng thích hợp dự đoán hoặc có những chiến lược tiếp thị hiệu quả.

Một trong các ứng dụng phổ biến của khai phá dữ liệu là phát hiện luật kết hợp giữa các dịch vụ viễn thông khách hàng sử dụng. Hiện nay trên một đường điện thoại khách hàng có thể sử dụng rất nhiều dịch vụ khác nhau, ví dụ như: gọi điện thoại, truy cập internet, tra cứu thông tin từ hộp thư tự động, nhắn tin, gọi 108, ... Dựa trên cơ sở dữ liệu khách hàng, chúng ta có thể khám phá các liên kết trong việc sử dụng các dịch vụ, có thể đưa ra các luật như (khách hàng gọi điện thoại quốc tế) => (truy cập internet), v.v…Trên có sở phân tích được các luật như vậy, các công ty

viễn thông có thể điều chỉnh việc bố trí nơi đăng ký các dịch vụ phù hợp, ví dụ điểm đăng ký điện thoại quốc tế nên bố trí gần với điểm đăng ký internet chẳng hạn.

Một ứng dụng khác phục vụ chiến lược marketing đó là sử dụng kỹ thuật khai phá luật kết hợp của khai phá dữ liệu để tìm ra tập các thành phố, tỉnh nào trong nước thường gọi điện thoại với nhau. Ví dụ, ta có thể tìm ra tập phổ biến (Cần Thơ, HCM, Hà Nội) chẳng hạn. Điều này thật sự hữu dụng trong việc hoạch định chiến lược tiếp thị hoặc xây dựng các vùng cước phù hợp.

Một vấn đề khá phổ biến ở các công ty viễn thông hiện nay là sự thay đổi nhà cung cấp dịch vụ (customer churn), đặc biệt đối với các công ty điện thoại di động. Đây là vấn đề khá nghiêm trọng ảnh hưởng đến tốc độ phát triển thuê bao, cũng như doanh thu của các nhà cung cấp dịch vụ. Thời gian gần đây các nhà cung cấp dịch vụ di động luôn có chính sách khuyến mãi lớn để lôi kéo khách hàng. Điều đó dẫn đến một lượng không nhỏ khách hàng thường xuyên thay đổi nhà cung cấp để hưởng những chính sách khuyến mãi đó. Các kỹ thuật khai phá dữ liệu hiện nay có thể dựa trên dữ liệu tiền sử để tìm ra các quy luật, từ đó có thể tiên đoán trước được khách hàng nào có ý định rời khỏi mạng trước khi họ thực hiện. Sử dụng các kỹ thuật khai phá dữ liệu như xây dựng cây quyết định (decision tree), mạng nơ-ron nhân tạo (neural network) trên dữ liệu cước (billing data), dữ liệu chi tiết cuộc gọi (call detail data), dữ liệu khách hàng (customer data) tìm ra các quy luật, nhờ đó ta có thể tiên đoán trước ý định rời khỏi mạng của khách hàng, từ đó công ty viễn thông sẽ có các ứng xử phù hợp nhằm lôi kéo khách hàng.

Cuối cùng, một ứng dụng cũng rất phổ biến đó là phân lớp khách hàng (classifying). Dựa vào kỹ thuật học trên cây quyết định (decision tree learning) xây dựng được từ dữ liệu khách hàng và chi tiết cuộc gọi có thể tìm ra các quy luật để phân loại khách hàng. Ví dụ, ta có thể phân biệt được khách hàng nào thuộc đối tượng kinh doanh hay nhà riêng dựa vào các luật sau:

Luật 1: Nếu không quá 43% cuộc gọi có thời gian từ 0 đến 10 giây và không đến 13% cuộc gọi vào cuối tuần thì đó là khách hàng kinh doanh.

Luật 2: Nếu trong 2 tháng có các cuộc gọi đến hầu hết từ 3 mã vùng giống nhau và dưới 56,6% cuộc gọi từ 0-10 giây thì đó là khách hàng riêng.

Trên cơ sở tìm được các luật tương tự như vậy, ta dễ dàng phân loại khách hàng, từ đó có chính sách phân khúc thị trường hợp lý.

Nhóm 3: Các ứng dụng phát hiện và cô lập lỗi trên hệ thống mạng viễn thông (Network fault isolation).

Mạng viễn thông là một cấu trúc cực kỳ phức tạp với nhiều hệ thống phần cứng và phần mềm khác nhau. Phần lớn các thiết bị trên mạng có khả năng tự chuẩn đoán và cho ra thông điệp trạng thái, cảnh báo lỗi (status and alarm message). Với mục tiêu là quản lý và duy trì độ tin cậy của hệ thống mạng, các thông tin cảnh báo phải được phân tích tự động và nhận diện lỗi trước khi nó xuất hiện làm giảm hiệu năng của mạng. Bởi vì số lượng lớn cảnh báo độc lập và có vẻ như không quan hệ gì với nhau nên vấn đề nhận diện lỗi không ít khó khăn. Kỹ thuật khai phá dữ liệu có vai trò sinh ra các luật giúp hệ thống có thể phát hiện lỗi sớm hơn khi nó xảy ra. Kỹ thuật khám phá mẫu tuần tự (sequential/temporal patterns) của data mining thường được ứng dụng trong lĩnh vực này thông qua việc khai thác cơ sở dữ liệu trạng thái mạng (network status data).

1.5 Một số thách thức đặt ra cho việc khai phá dữ liệu.

- Số đối tượng trong cơ sở dữ liệu thường rất lớn.
- Số chiều (thuộc tính) của cơ sở dữ liệu lớn.
- Dữ liệu và tri thức luôn thay đổi có thể làm cho các mẫu đã phát hiện không còn phù hợp.
- Dữ liệu bị thiếu hoặc nhiễu.
- Quan hệ phức tạp giữa các thuộc tính.
- Giao tiếp với người sử dụng và kết hợp với các tri thức đã có.
- Tích hợp với các hệ thống khác

1.6 Kết luận chương.

Khai phá dữ liệu là một lĩnh vực khoa học mới xuất hiện, nhằm tự động hóa khai thác những thông tin, tri thức hữu ích, tiềm ẩn trong các CSDL, giúp chúng ta giải quyết tình trạng ngày một gia tăng trong những năm qua: "*Ngập trong dữ liệu mà vẫn đói tri thức".* Các kết quả nghiên cứu cùng với những ứng dụng thành công trong khai phá dữ liệu, khám phá tri thức cho thấy khai phá dữ liệu là một lĩnh vực khoa học tiềm năng, mang lại nhiều lợi ích, đồng thời có ưu thế hơn hẳn so với các công cụ phân tích dữ liệu truyền thống.

Nội dung của chương 1 đã trình bày khái quát về khai phá dữ liệu, tóm tắt quá trình khai phá, các phương pháp, các ứng dụng và những thách thức.

CHƯƠNG 2: PHÂN CỤM DỮ LIỆU VÀ CÁC GIẢI THUẬT THEO TIẾP CẬN PHÂN CẤP

2.1 Phân cụm dữ liệu (Data Clustering) là gì?

Phân cụm dữ liệu - PCDL (*Data Clustering*) là một trong những hướng nghiên cứu trọng tâm của lĩnh vực khai phá dữ liệu (Data Mining) khám phá tri thức (KDD). Mục đích của phân cụm là nhóm các đối tượng vào các cụm sao cho các đối tượng trong cùng một cụm có tính tương đồng cao và độ bất tương đồng giữa các cụm lớn, từ đó cung cấp thông tin, tri thức hữu ích cho việc ra quyết định.

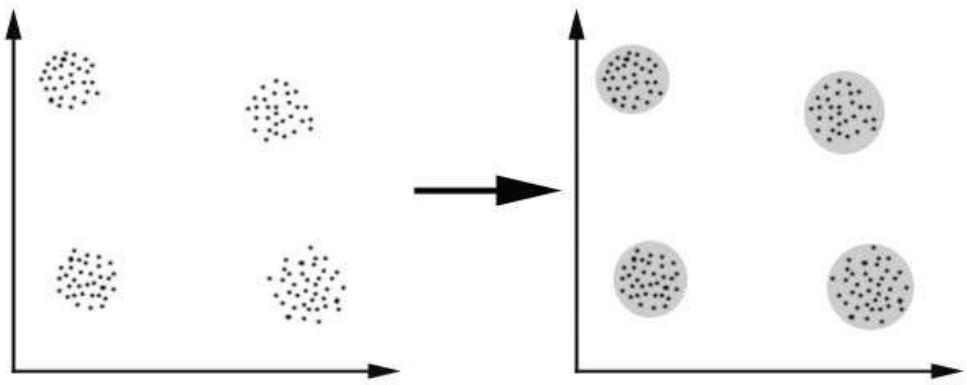
Data mining là một quá trình tìm kiếm, chắt lọc các tri thức mới, tiềm ẩn, hữu dụng trong tập dữ liệu lớn.

Phân cụm dữ liệu hay phân cụm, gom cụm, cũng có thể gọi là phân tích cụm, phân tích phân đoạn, phân tích phân loại; "*là một kỹ thuật trong Data mining nhằm tìm kiếm, phát hiện, nhóm một tập các đối tượng thực thể hay trừu tượng thành lớp các đối tượng tương tự trong tập dữ liệu lớn để từ đó cung cấp thông tin, tri thức cho việc ra quyết định.".*

Một cụm là một tập hợp các đối tượng dữ liệu mà các phần tử của nó *"tương tự"* (Similar) nhau cùng trong một cụm và *"phi tương tự"* (Dissimilar) với các đối tượng trong các cụm khác. Một cụm các đối tượng dữ liệu có thể xem như là một nhóm trong nhiều ứng dụng.

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 khi thực hiện phương pháp phân cụm.

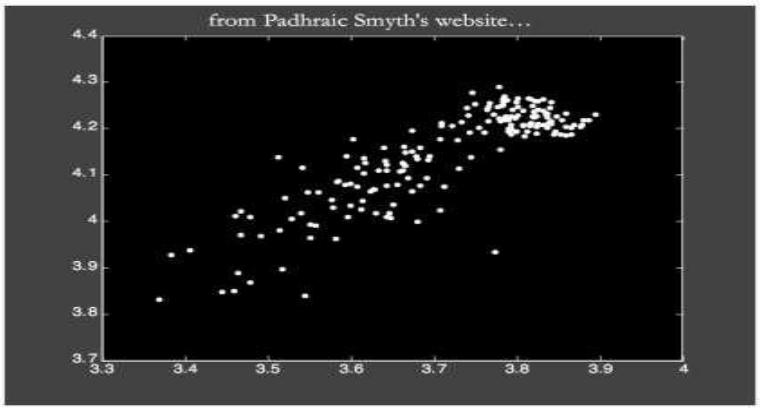
Chúng ta có thể minh hoạ vấn đề phân cụm như Hình 2.1 sau đây:



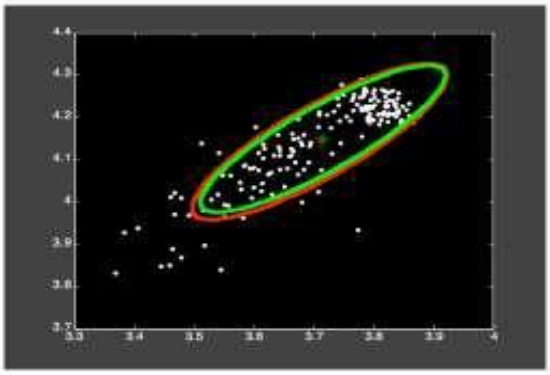
Hình 2.1: Mô phỏng vấn đề phân cụm dữ liệu

Trong hình trên, sau khi phân cụm chúng ta thu được bốn cụm trong đó các phần tử "*gần nhau*" hay là "*tương tự*" thì được xếp vào một cụm, trong khi đó các phần tử "*xa nhau*" hay là "*phi tương tự*" thì chúng thuộc về các cụm khác nhau.

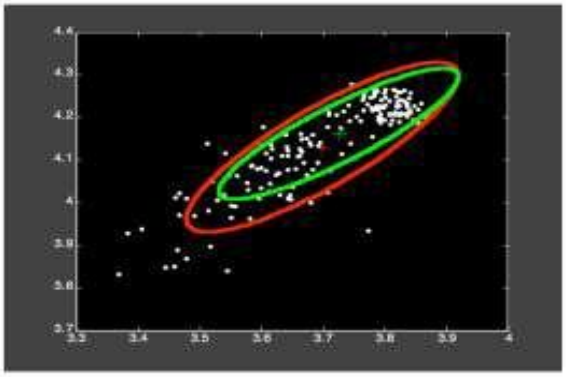
Để minh hoạ cụ thể hơn cho vấn đề này ta có thể quan sát các hình ảnh sau:



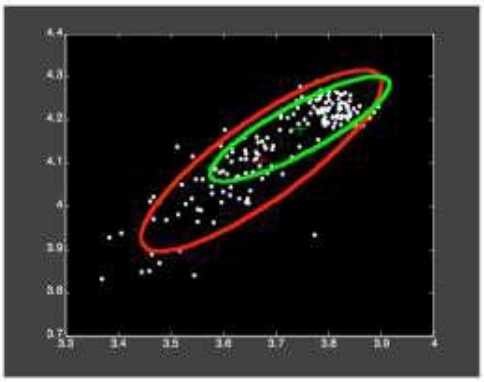
Hình 2.2: Dữ liệu nguyên thủy



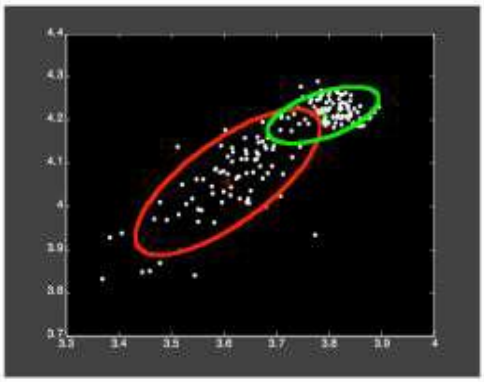
Hình 2.3



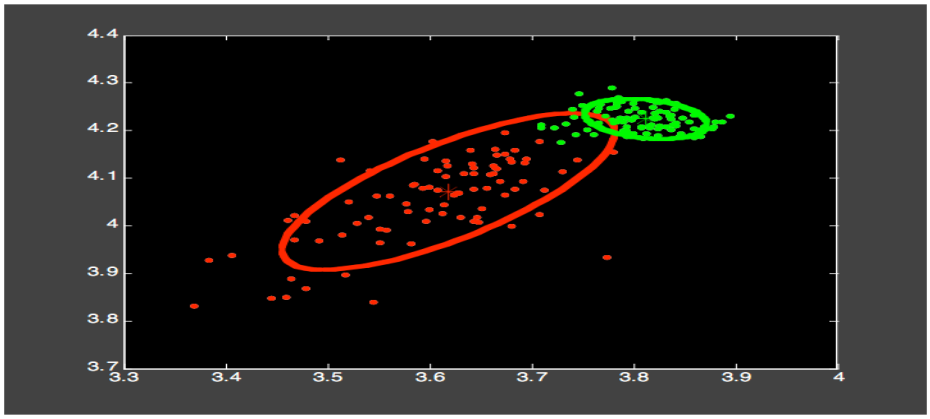
Hình 2.4



Hình 2.5



Hình 2.6



Hình 2.7: Kết quả của quá trình phân cụm

Các hình từ 2.2 đến 2.7 là thể hiện quá trình phân cụm từ khi"*bắt đầu*" cho đến khi "*kết thúc*".

Trong *PCDL khái niệm (Concept Clustering)* thì hai hoặc nhiều đối tượng cùng được xếp vào một cụm nếu chúng có chung một định nghĩa về khái niệm hoặc chúng xấp xỉ với các khái niệm mô tả cho trước, như vậy, ở đây PCDL không sử dụng khái niệm "tương tự" như đã trình bày ở trên.

Trong học máy, phân cụm dữ liệu được xem là vấn đề học không có giám sát, vì nó phải đi giải quyết vấn đề tìm một cấu trúc trong tập hợp các dữ liệu chưa biết biết trước các thông tin về lớp hay các thông tin về tập ví dụ huấn luyện. Trong nhiều trường hợp, khi phân lớp (Classification) được xem là việc học có giám sát thì phân cụm dữ liệu là một bước trong phân lớp dữ liệu, trong đó PCDL sẽ khởi tạo các lớp cho phân lớp bằng cách xác định các nhãn cho các nhóm dữ liệu.

Một vấn đề thường gặp trong PCDL đó là hầu hết các dữ liệu cần cho phân cụm đều có chứa dữ liệu "nhiễu" (noise) do quá trình thu thập thiếu chính xác hoặc thiếu đầy đủ, vì vậy cần phải xây dựng chiến lược cho bước tiền xử lý dữ liệu nhằm khắc phục hoặc loại bỏ "nhiễu" trước khi bước vào giai đoạn phân tích phân cụm dữ liệu. "Nhiễu" ở đây có thể là các đối tượng dữ liệu không không chính xác, hoặc là các đối tượng dữ liệu khuyết thông tin về một số thuộc tính. Một trong các kỹ thuật xử lý nhiễu phổ biến là việc thay thế giá trị của các thuộc tính của đối tượng "nhiễu" bằng giá trị thuộc tính tương ứng của đối tượng dữ liệu gần nhất.

Phân cụm dữ liệu là một vấn đề khó, vì rằng người ta phải đi giải quyết các vấn đề con cơ bản như sau:

- Xây dựng hàm tính độ tương tự.

- Xây dựng các tiêu chuẩn phân cụm.
- Xây dựng mô hình cho cấu trúc cụm dữ liệu
- Xây dựng thuật toán phân cụm và xác lập các điều kiện khởi tạo
- Xây dựng các thủ tục biểu diễn và đánh giá kết quả phân cụm.

Phân cụm dữ liệu là bài toán thuộc vào lĩnh vực học máy *không giám sát* và đang được ứng dụng rộng rãi để khai thác thông tin hữu ích từ dữ liệu.

Phân cụm dữ liệu là quá trình phân chia một tập dữ liệu ban đầu thành các cụm sao cho các đối tượng trong cùng một cụm "*tương tự*" nhau. Vì vậy phải xác định được một phép đo "*khoảng cách*" hay phép đo tương tự giữa các cặp đối tượng để phân chia chúng vào các cụm khác nhau. Dựa vào hàm tính độ tương tự này cho phép xác định được hai đối tượng có tương tự hay không. Theo quy ước, giá trị của hàm tính độ đo tương tự càng lớn thì sự tương đồng giữa các đối tượng càng lớn và ngược lại. Hàm tính độ phi tương tự tỉ lệ nghịch với hàm tính độ t ương tự.

2.2 Thế nào là phân cụm tốt?

- Một phương pháp tốt sẽ tạo ra các cụm có chất lượng cao theo nghĩa có sự tương tự cao trong một lớp (intra-class), tương tự thấp giữa các lớp (inter-class).
- Chất lượng của kết quả gom cụm phụ thuộc vào: độ đo tương tự sử dụng và việc cài đặt độ đo tương tự.
- Chất lượng của phương pháp gom cụm cũng được đo bởi khả năng phát hiện các mẫu bị che (hidden patterns).

2.3 Bài toán phân cụm dữ liệu

Bài toán phân cụm dữ liệu thường được hiểu là một bài toán học không giám sát và được phát biểu như sau:

Cho tập X = $\{x_1,..., x_n\}$ gồm n đối tượng dữ liệu trong không gian p-chiều, $x_i \in R^p$. Ta cần chia X thành k cụm đôi một không giao nhau: $X = \bigcup_{i=1}^{k} C_i$, $C_i \cap C_j = \varnothing,\ i \neq j$, sao cho các đối tượng trong cùng một cụm thì tương tự nhau và các đối tượng trong các cụm khác nhau thì khác nhau hơn theo một cách nhìn nào đó.

Số lượng k cụm có thể được cho trước hoặc xác định nhờ phương pháp phân cụm. Để thực hiện phân cụm ta cần xác định được mức độ tương tự giữa các đối tượng, tiêu chuẩn để phân cụm, trên cơ sở đó xây dựng mô hình và các thuật toán phân cụm theo nhiều cách tiếp cận. Mỗi cách tiếp cận cho ta kết quả

phân cụm với ý nghĩa sử dụng khác nhau.

2.4 Các ứng dụng của phân cụm

Phân cụm DL là một trong những công cụ chính được ứng dụng trong nhiều lĩnh vực. Một số ứng dụng điển hình trong các lĩnh vực sau:

- *Thương mại:* Trong thương mại, PCDL 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 CSDL khách hàng.
- *Sinh học:* Trong sinh học, PCDL được sử dụng để xác định các loại sinh vật, phân loại các Gen với chức năng tương đồng và thu được các cấu trúc trong các mẫu.
- *Phân tích dữ liệu không gian:* Do sự đồ sộ của dữ liệu không gian như dữ liệu thu được từ các hình ảnh chụp từ vệ tinh, các thiết bị y học hoặc hệ thống thông tin địa lý (GIS),… người dùng rất khó để kiểm tra các dữ liệu không gian một cách chi tiết. PCDL có thể trợ giúp người dùng tự động phân tích và xử lý các dữ liêu không gian như nhận dạng và chiết xuất các đặc tính hoặc các mẫu dữ liệu quan tâm có thể tồn tại trong CSDL không gian.
- *Lập quy hoạch đô thị:* Nhận dạng các nhóm nhà theo kiểu và vị trí địa lý,… nhằm cung cấp thông tin cho quy hoạch đô thị.
- *Nghiên cứu trái đất*: Phân cụm để theo dõi các tâm động đất nhằm cung cấp thông tin cho nhận dạng các vùng nguy hiểm.
- *Địa lý:* Phân lớp các động vật và thực vật và đưa ra đặc trưng của chúng.
- *Web Mining*: PCDL có thể khám phá các nhóm tài liệu quan trọng, có nhiều ý nghĩa trong môi trường Web. Các lớp tài liệu này trợ giúp cho việc khám phá tri thức từ dữ liệu, …

2.5 Các yêu cầu đối với thuật toán phân cụm dữ liệu

- Scalability: Có khả năng mở rộng
- Khả năng làm việc với các loại thuộc tính khác nhau
- Khám phá ra các cụm có hình dạng bất kì
- Yêu cầu tối thiểu về tri thức lĩnh vực nhằm xác định các tham biến đầu vào
- Khả năng làm việc với dữ liệu có chứa nhiễu (outliers).
- Không nhạy cảm với thứ tự các bản ghi nhập vào.
- Khả năng làm việc với dữ liệu nhiều chiều.
- Có thể diễn dịch và khả dụng.

2.6 Các kiểu dữ liệu và phép đo độ tương tự

2.6.1 Cấu trúc dữ liệu

Các thuật toán phân cụm hầu hết sử dụng hai cấu trúc dữ liệu điển hình sau:

- **Ma trận dữ liệu**

Biểu diễn *n* đối tượng và *p* biến (là các phép đo hoặc các thuộc tính) của đối tượng, có dạng ma trận $n \times p$

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{21} & \cdots & x_{2f} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- **Ma trận bất tương tự**

Lưu trữ một tập các trạng thái ở gần nhau sẵn có giữa n cặp đối tượng bởi một ma trận cỡ $n \times n$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & ... & ... & ... \\ d(3.1) & d(3,2) & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ d(n,1) & d(n,2) & ... & ... & 0 \end{bmatrix}$$

Với d(i,j) là sự khác nhau hoặc sự bất tương tự giữa các đối tượng i và j. Nói chung, d(i,j) là số không âm và gần bằng 0 khi đối tượng i và j là tương tự nhau ở mức cao hoặc *"gần"* với đối tượng khác. Rõ ràng, ma trận bất tương tự là ma trân đối xứng, nghĩa là d(i,j) = d(j,i) và ta cũng có d(i,i) = 0 với mọi $1 \le i \le n$.

Ma trận dữ liệu thường được gọi là ma trận dạng 2, trái lại ma trận bất tương tự được gọi là ma trận dạng 1, trong đó các dòng và các cột biểu diễn các thực thể khác nhau trước, các thực thể giống nhau biểu diễn sau. Nhiều thuật toán gom cụm hoạt động trên ma trận bất tương tự. Nếu dữ liệu được trình bày dưới dạng một ma trận dữ liệu, khi đó nó có thể được chuyển vào ma trận bất tương tự trước khi áp dụng các thuật toán phân cụm.

Không có định nghĩa duy nhất về sự tương tự và bất tương tự giữa các đối tượng dữ liệu. Tùy thuộc vào dữ liệu khảo sát, định nghĩa *"Sự tương tự/bất tương*

tự" sẽ được đánh giá cụ thể. Thông thường độ bất tương tự được biểu diễn qua độ đo khoảng cách d(i,j). Độ đo khoảng cách phải thỏa mãn các điều kiện sau:

(i) $d(i,j) \geq 0$

(ii) d (i,j) = 0 nếu i = j

(iii) d (i,j)=d(j,i)

(iv) $d(x,z) \leq d(x,y) + d(y,z)$.

2.6.2 Các kiểu dữ liệu

Sau đây là các kiểu của dữ liệu, và ứng với mỗi kiểu dữ liệu thì có một hàm tính độ đo tương tự để xác định khoảng cách giữa 2 phân tử của cùng một kiểu dữ liệu. Tất cả các độ đo đều được xác định trong không gian metric. Bất kỳ một metric nào cũng là một độ đo nhưng ngược lại thì không đúng. Độ đo ở đây có thể là tương tự hoặc phi tương tự.

Một tập dữ liệu X là không gian metric nếu:

- Với mỗi cặp x,y thuộc X đều xác định được một số thực d(x,y) theo một quy tắc nào đó và được gọi là khoảng cách của x,y.
- Quy tắc đó phải thoả mãn các tính chất sau:

a) d(x,y) > 0 nếu x ≠ y.

b) d(x,y) = 0 nếu x = y.

c) d(x,y) = d(y,x).

d) d(x,y) <= d(x,z) + d(z,y).

Cho một CSDL **D** chứa n đối tượng trong không gian k chiều trong đó *x, y, z* là các đối tượng thuộc **D***: x*=(*x1, x2,. ., x*$_k$*); y=(y1, y2,. ., y*$_k$*); z=(z1, z2,. ., z*$_k$*),* trong đó x_i, y_i, z_i với $i = \overline{1,k}$ là các đặc trưng hoặc thuộc tính tương ứng của các đối tượng *x, y, z.* Vì vậy, hai khái niệm "*các kiểu dữ liệu*" và *"các kiểu thuộc tính dữ liệu"* được xem là tương đương với nhau, như vậy, chúng ta sẽ có các kiểu dữ liệu sau*:*

Phân loại các kiểu dữ liệu (kiểu thuộc tính) dựa trên kích thước miền:

- *Thuộc tính liên tục (Continuous Attribute):* nếu miền giá trị của nó là một miền bao gồm vô hạn, không đếm được các giá trị. Thí dụ như các thuộc tính nhiệt độ hoặc cường độ âm thanh.

- *Thuộc tính rời rạc (Discrette Attribute)*: Nếu miền giá trị của nó là tập hữu hạn, đếm được. Thí dụ như các thuộc tính về số serial của một cuốn sách, số thành viên trong một gia đình, …

Lớp các *Thuộc tính nhị phân* là trường hợp đặc biệt của thuộc tính rời rạc mà miền giá trị của nó chỉ có 2 phần tử được diễn tả như: *Yes / No* hoặc *Nam/Nữ, False/true,...*

Phân loại các kiểu dữ liệu (kiểu thuộc tính) dựa trên hệ đo

Giả sử rằng chúng ta có hai đối tượng x, y và các thuộc tính 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 Scale):* 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 - nghĩa là nếu x và y là hai đối tượng thuộc tính thì chỉ có thể xác định là $x \neq y$ hoặc x = y.
- *Thuộc tính có thứ tự (Ordinal Scale)*: là thuộc tính định danh có thêm tính thứ tự, nhưng chúng không được định lượng. Nếu x và y là hai giá trị của một thuộc tính 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 thuộc tính xếp loại sinh viên thành các mức: Giỏi, khá, trung bình, kém
- *Thuộc tính khoảng (Interval Scale):* Nhằm để đo các giá trị theo xấp xỉ tuyến tính. Với thuộc tính khoảng, chúng ta có thể xác định một đối tượng là đứng trước hoặc đứng sau một đối tượng khác với một khoảng là bao nhiêu. Nếu $x_i > x_j$ thì ta nói hai đối tượng i và j cách nhau một khoảng $x_i - x_j$ ứng với thuộc tính x. Một thí dụ về thuộc tính khoảng là số *Serial* của một đầu sách trong thư viện.
- *Thuộc tính tỉ lệ (Ratio Scale):* là thuộc tính khoảng nhưng được xác định một điểm mốc tương đối, 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* (Categorical), còn *thuộc tính khoảng* và *thuộc tính tỉ lệ* được gọi là *thuộc tính số* (Numeric).

Người ta còn đặc biệt quan tâm đến dữ liệu không gian (Spatial Data). Đây là loại dữ liệu có các thuộc tính số khái quát trong không gian nhiều chiều, dữ liệu không gian mô tả các thông tin liên quan đến không gian chứa đựng các đối tượng,

thí dụ như thông tin về hình học, … Dữ liệu không gian có thể là dữ liệu liên tục hoặc rời rạc:

- *Dữ liệu không gian rời rạc:* có thể là một điểm trong không gian nhiều chiều và cho phép ta xác định được khoảng cách giữa các đối tượng dữ liệu trong không gian.
- *Dữ liệu không gian liên tục:* bao chứa một vùng trong không gian.

Thông thường, các thuộc tính số được đo bằng các đơn vị xác định như là kilogams hay là centimeter. Các đơn vị đo có ảnh hưởng đến các kết quả phân cụm. Thí dụ như thay đổi độ đo cho thuộc tính cân nặng từ kilogams sang Pound có thể mang lại các kết quả khác nhau trong phân cụm. Để khắc phục điều này người ta phải chuẩn hoá dữ liệu, tức là sử dụng các thuộc tính dữ liệu không phụ thuộc vào đơn vị đo. Thực hiện chuẩn hoá như thế nào phụ thuộc vào ứng dụng và người dùng, thông thường chuẩn hoá dữ liệu được thực hiện bằng cách thay thế mỗi một thuộc tính bằng thuộc tính số hoặc thêm các trọng số cho các thuộc tính.

Sau đây là các phép đo độ tương tự áp dụng đối với các kiểu dữ liệu khác nhau:

1) Thuộc tính khoảng (Interval Scale):

Các đơn vị đo có thể ảnh hưởng đến phân tích cụm. Vì vậy để tránh sự phụ thuộc vào đơn vị đo, cần chuẩn hóa dữ liệu.

Các bước chuẩn hóa dữ liệu

(i) Tính giá trị trung bình và sai số tuyệt đối trung bình

$$m_f = \frac{1}{h}\ x_{1f} + x_{2f} + ... x_{nf}$$

$$s_f = \frac{1}{h}\ \left|x_{1f} - m_f\right| + \left|x_{2f} - m_f\right| + ... + \left|x_{nf} - m_f\right|$$

(ii) Tính độ đo chuẩn:

$$z_{ij} = \frac{x_{ij} - m_f}{s_f}$$

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 như sau:

- *Khoảng cách Minskowski:*

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|^{q}\ ^{1/q}$$

Trong đó *q* là số tự nhiên dương.

- *Khoảng cách Euclide:*

$$d\ \ x,y\ \ = \sqrt{\sum_{i=1}^{p}\ \ x_i - y_i\ ^{2}}$$

Đây là trường hợp đặc biệt của khoảng cách Minskowski trong trường hợp q=2.

- *Khoảng cách Manhattan:*

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

Đây là trường hợp đặc biệt của khoảng cách Minskowski trong trường hợp q=1.

- Khoảng cách Chebychev

$$d(x,y) = \max_{i=1}^{p}\ \ |x_i - y_i|$$

Đây là trường hợp của khoảng cách Minskowski trong trường hợp q → ∞

***2) Thuộc tính nhị phân*:**

Giả sử tất cả τ thuộc tính về đối tượng đều là nhị phân biểu thị bằng 0 và 1. Xét bảng tham số sau về hai đối tượng x và y:

	y = 1	y = 0	
x = 1	a	b	a + b
x = 0	c	d	c + d
	a + c	b + d	a + b + c + d

Hình 2.8: Bảng tham số

Trong đó

- a là tổng số các thuộc tính có giá trị là 1 trong cả hai đối tượng x, y.
- b là tổng số các giá trị thuộc tính có giá trị là 1 trong x và 0 trong y.
- c là tổng số các giá trị thuộc tính có giá trị là 0 trong x và 1 trong y.
- d là tổng số các giá trị thuộc tính có giá trị là 0 trong cả x và y.

Ta có tổng số các thuộc tính về đối tượng p = a + b + c + d.

Các phép đo độ tương tự giữa hai đối tượng trong trường hợp dữ liệu thuộc tính nhị phân được định nghĩa như sau:

- Hệ số đối sánh đơn giản:

$$d\ \ x,y\ \ = \frac{a+b}{p}$$

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:

$$d\ \ x,y\ \ = \frac{a}{a+b+c}$$

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*.

Ví dụ: Bảng hồ sơ bệnh nhân:

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

Có 8 thuộc tính *Name, Gender, Fever, Cough, Test-1, Test-2, Test-3, Test-4* trong đó:

- *Gender* là thuộc tính nhị phân đối xứng.
- Các thuộc tính còn lại là nhị phân bất đối xứng

Ta gán các trị Y và P bằng 1 và trị N được gán bằng 0. Tính khoảng cách giữa các bệnh nhân dựa vào các bất đối xứng dùng hệ số Jacard ta có

$$d\ \ Jack, Mary\ \ = \frac{0+1}{2+0+1} = 0.33$$

$$d\ \ Jack, Jim\ \ = \frac{1+1}{1+1+1} = 0.67$$

$$d\ \ Jim, Mary\ = \frac{1+2}{1+1+2} = 0.75$$

Như vậy, theo tính toán trên, Jim và Marry có khả năng mắc bệnh giống nhau nhiều nhất vì *d(Jim, Marry)=0.75* là lớn nhất.

3) Thuộc tính định danh (nominal Scale):

Có hai phương pháp để tính toán sự tương tự giữa hai đối tượng:

- *Phương pháp 1:* Đối sánh đơn giản

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

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

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

- *Phương pháp 2:* Dùng một số lượng lớn các biến nhị phân
  - Tạo biến nhị phân mới cho từng trạng thái định danh.
  - Các biến thứ tự có thể là liên tục hay rời rạc
  - Thứ tự của các trị là quan trọng, Ví dụ: *hạng.*
  - Có thể xử lý như tỉ lệ khoảng như sau:
    - Thay thế x_{if} bởi hạng của chúng.
    - Ánh xạ phạm vi của từng biến vào đoạn [0,1] bằng cách thay thế đối tượng i trong biến thứ f bởi $r_{if} \in 1, \ldots, M_f$
    - Tính sự khác nhau dùng các phương pháp cho biến tỉ lệ theo khoảng.

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

4) Thuộc tính có thứ tự (Ordinal Scale):

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 $r_i \in \{1 \ldots M_i\}$.

- 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

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

  - Sử dụng công thức tính độ phi tương tự của thuộc tính khoảng đối với các giá trị $z_i^{(j)}$ trên đây thu được độ phi tương tự của thuộc tính có thứ tự.

5) Thuộc tính tỉ lệ (Ratio Scale)

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.

Các phương pháp tính độ tương tự:

- Xử lý chúng như các biến thang đo khoảng
- Áp dụng các biến đổi logarithmic
- Xử lý chúng như dữ liệu thứ tự liên tục.
- Xử lý chúng theo hạng như thang đo khoảng.

Trong thực tế, khi tính độ đo tương tự dữ liệu, người ta chỉ xem xét một phần các thuộc tính đặc trưng đối với các kiểu dữ liệu hoặc là đánh trọng số cho cho tất cả các thuộc tính dữ liệu. Trong một số trường hợp, người ta loại bỏ đơn vị đo của các thuộc tính dữ liệu bằng cách chuẩn hoá chúng, hoặc gán trọng số cho mỗi thuộc tính giá trị trung bình, độ lệch chuẩn. Các trọng số này có thể sử dụng trong các độ đo khoảng cách trên, thí dụ với mỗi thuộc tính dữ liệu đã được gán trọng số tương ứng w_i ($1 \le i \le k$), độ tương đồng dữ liệu được xác định như sau:

$$d(x, y) = \sqrt{\sum_{i=1}^{p} w_i (x_i - y_i)^2}$$

Người ta có thể chuyển đổi giữa các mô hình cho các kiểu dữ liệu trên, thí dụ dữ liệu kiểu hạng mục có thể chuyển đổi thành dữ liệu nhị phân và ngược lại. Thế nhưng, giải pháp này rất tốt kém về chi phí tính toán, do vậy, cần phải cân nhắc khi áp dụng cách thức này.

Tóm lại, tuỳ từng trường hợp dữ liệu cụ thể mà người ta sử dụng các cách tính độ tương tự khác nhau. Việc xác định độ tương tự dữ liệu thích hợp, chính xác,

đảm bảo khách quan là rất quan trọng, góp phần xây dựng thuật toán PCDL hiệu quả cao trong việc đảm bảo chất lượng cũng như chi phí tính toán của thuật toán.

2.7 Các hướng tiếp cận bài toán phân cụm dữ liệu

Có rất nhiều các phương pháp gom cụm khác nhau. Việc lựa chọn phương pháp nào tuỳ thuộc vào kiểu dữ liệu, mục tiêu và ứng dụng cụ thể. Nhìn chung, có thể chia thành các phương pháp sau:

2.7.1 Phương pháp phân hoạch.

Cho một cơ sở dữ liệu D chứa n đối tượng, tạo phân hoạch thành tập có k cụm sao cho:

- ❖ Mỗi cụm chứa ít nhất một đối tượng
- ❖ Mỗi đối tượng thuộc về một cụm duy nhất
- ❖ k cụm tìm được thỏa mãn tiêu chuẩn tối ưu đã định.

Các phương pháp phân hoạch

Phương pháp *heuristic* điển hình được biết đến là *k-means và k-medoids.*

2.7.2 Phương pháp phân cấp

Phân cụm phân cấp sắp xếp một tập dữ liệu đã cho thành một cấu trúc có dạng hình cây, cây phân cấp này được xây dựng theo kỹ thuật đệ quy bằng phương pháp trên xuống (Top down) hoặc phương pháp dưới lên (Bottum up).

- **Phương pháp "dưới lên" (Bottom up)**: Phương pháp này bắt đầu bằng cách khởi tạo mỗi đối tượng riêng biệt là một cụm, sau đó tiến hành nhóm các đối tượng theo một độ đo tương tự (như khoảng cách giữa hai trung tâm của hai nhóm), quá trình này được thực hiện cho đến khi tất cả các nhóm được kết nhập thành một nhóm (mức cao nhất của cây phân cấp) hoặc cho đến khi các điều kiện kết thúc thỏa mãn. Như vậy, cách tiếp cận này sử dụng chiến lược tham lam trong quá trình phân cụm.
- **Phương pháp "trên xuống" (Top Down)**: Bắt đầu với trạng thái là tất cả các đối tượng được xếp trong cùng một cụm. Sau mỗi vòng lặp thành công, một cụm được tách thành các cụm nhỏ hơn theo giá trị của một phép đo độ tương tự nào đó cho đến khi mỗi đối tượng là một cụm, hoặc cho đến khi điều kiện dừng thỏa mãn. Cách tiếp cận này sử dụng chiến lược chia để trị trong quá trình phân cụm.

Hai thuật toán phân cụm phân cấp điển hình là thuật toán CURE, và thuật toán BIRCH.

Trong nhiều ứng dụng thực tế, người ta kết hợp cả hai phương pháp phân cụm phân hoạch và phương phân cụm phân cấp, nghĩa là kết quả thu được của phương pháp phân cấp có thể cải tiến thông quan bước phân cụm phân hoạch. Phân cụm phân hoạch và phân cụm phân cấp là hai phương pháp PCDL cổ điển, hiện nay đã có nhiều thuật toán cải tiến dựa trên hai phương pháp này đã được áp dụng phổ biến trong Data Mining.

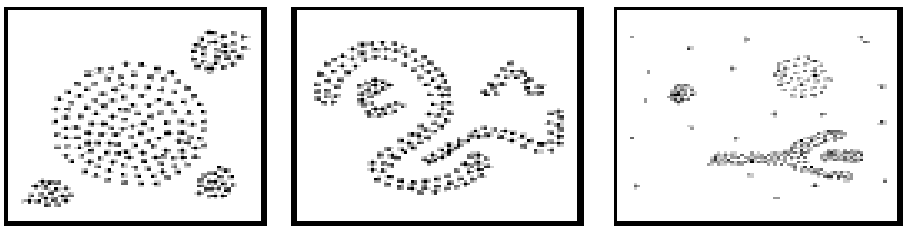
2.7.3 Phương pháp dựa vào mật độ (Density based Methods)

Gom cụm dựa trên sự liên thông địa phương và hàm mật độ. Theo phương pháp này các điểm có mật độ cao hơn sẽ ở cùng một cụm.

Đặc trưng của phương pháp:

- Phát hiện ra các cụm có hình dạng bất kì.
- Phát hiện nhiễu.

Phương pháp này nhóm các đối tượng theo hàm mật độ xác định. Mật độ được định nghĩa như là số các đối tượng lân cận của một đối tượng dữ liệu theo một ngưỡng nào đó. Trong cách tiếp cận này, khi một cụm dữ liệu đã xác định thì nó tiếp tục được phát triển thêm các đối tượng dữ liệu mới miễn là số các đối tượng lân cận của các đối tượng này phải lớn hơn một ngưỡng đã được xác định trước. Phương pháp phân cụm dựa vào mật độ của các đối tượng để xác định các cụm dữ liệu có thể phát hiện ra các cụm dữ liệu với hình thù bất kỳ. Tuy vậy, việc xác định các tham số mật độ của thuật toán rất khó khăn, trong khi các tham số này lại có tác động rất lớn đến kết quả phân cụm dữ liệu. Hình dưới đây là một minh hoạ về các cụm dữ liệu với các hình thù khác nhau dựa trên mật độ được khám phá từ 3 CSDL khác nhau.



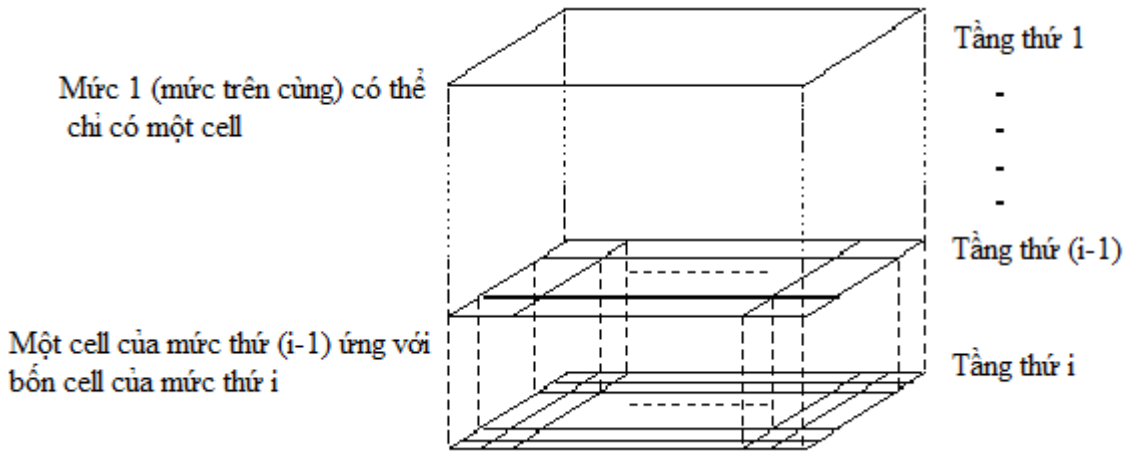
CSDL 1 CSDL 2 CSDL 3

Hình 2.9: Một số hình dạng cụm dữ liệu khám phá được bởi kỹ thuật PCDL dựa trên mật độ

Một số thuật toán PCDL dựa trên mật độ điển hình như DBSCAN, OPTICS, DENCLUE, …

2.7.4 Phân cụm dữ liệu dựa trên lưới

Kỹ thuật phân cụm dựa trên mật độ không thích hợp với dữ liệu nhiều chiều, để giải quyết cho đòi hỏi này, người ta đã dử dụng phương pháp phân cụm dựa trên lưới. Đây là phương pháp dựa trên cấu trúc dữ liệu lưới để PCDL, phương pháp này chủ yếu tập trung áp dụng cho lớp dữ liệu không gian. Thí dụ như dữ liệu được biểu diễn dưới dạng cấu trúc hình học của đối tượng trong không gian cùng với các quan hệ, các thuộc tính, các hoạt động của chúng. Mục tiêu của phương pháp này là lượng hoá tập dữ liệu thành các ô (Cell), các cell này tạo thành cấu trúc dữ liệu lưới, sau đó các thao tác PCDL làm việc với các đối tượng trong từng Cell này. Cách tiếp cận dựa trên lưới này không di chuyển các đối tượng trong các cell mà xây dựng nhiều mức phân cấp của nhóm các đối tượng trong một cell. Trong ngữ cảnh này, phương pháp này gần giống với phương pháp phân cụm phân cấp nhưng chỉ có điều chúng không trộn các Cell. Do vậy các cụm không dựa trên độ đo khoảng cách (hay độ đo tương tự đối với các dữ liệu không gian) mà nó được quyết định bởi một tham số xác định trước. Ưu điểm của phương pháp PCDL dựa trên lưới là thời gian xử lý nhanh và độc lập với số đối tượng dữ liệu trong tập dữ liệu ban đầu, (thay vào đó là chúng phụ thuộc vào số cell trong mỗi chiều của không gian lưới). Một thí dụ về cấu trúc dữ liệu lưới chứa các cell trong không gian như **Hình 2.10** sau:



Hình 2.10 : Mô hình cấu trúc dữ liệu lưới

Một số thuật toán PCDL dựa trên cấu trúc lưới điển hình như: STING, WAVECluster, CLIQUE,…

2.7.5 Phương pháp dựa trên mô hình (Gom cụm khái niệm, mạng neural)

Phương pháp này cố gắng tìm ra các phép xấp xỉ tốt cho các tham số mô hình sao cho khớp với dữ liệu một cách tốt nhất. Chúng có thể sử dụng chiến lược phân cụm phân hoạch hoặc chiến lược phân cụm phân cấp, dựa trên cấu trúc hoặc mô hình mà chúng giả định về tập dữ liệu và cách mà chúng tinh chỉnh các mô hình này để nhận dạng ra các phân hoạch.

Phương pháp PCDL dựa trên mô hình cố gắng khớp giữa dữ liệu với mô hình toán học, nó dựa trên giả định rằng dữ liệu được tạo ra từ một hỗn hợp của các phân phối xác suất cơ bản. Các thuật toán phân cụm dựa trên mô hình có hai tiếp cận chính: Mô hình thống kê và Mạng Nơ ron. Phương pháp này gần giống với phương pháp dựa trên mật độ, bởi vì chúng phát triển các cụm riêng biệt nhằm cải tiến các mô hình đã được xác định trước đó, nhưng đôi khi nó không bắt đầu với một số cụm cố định và không sử dụng cùng một khái niệm mật độ cho các cụm.

2.7.6 Phân cụm dữ liệu có ràng buộc

Sự phát triển của phân cụm dữ liệu không gian trên CSDL lớn đã cung cấp nhiều công cụ tiện lợi cho việc phân tích thông tin địa lý, tuy nhiên hầu hết các thuật toán này cung cấp rất ít cách thức cho người dùng để xác định các ràng buộc trong thế giới thực cần phải được thoả mãn trong quá trình PCDL. Để phân cụm dữ liệu không gian hiệu quả hơn, các nghiên cứu bổ sung cần được thực hiện để cung cấp cho người dùng khả năng kết hợp các ràng buộc trong thuật toán phân cụm.

Thực tế, các phương pháp trên đã và đang được phát triển và áp dụng nhiều trong PCDL. Đến nay, đã có một số nhánh nghiên cứu được phát triển trên cơ sở của các phương pháp tiếp cận trong PCDL đã trình bày ở trên như sau:

- *Phân cụm thống kê :* Dựa trên các khái niệm phân tích thống kê, nhánh nghiên cứu này sử dụng các độ đo tương tự để phân hoạch các đối tượng, nhưng chúng chỉ áp dụng cho các dữ liệu có thuộc tính số.
- *Phân cụm khái niệm :* Các kỹ thuật phân cụm được phát triển áp dụng cho dữ liệu hạng mục, chúng phân cụm các đối tượng theo các khái niệm mà chúng xử lý.
- *Phân cụm mờ :* Sử dụng kỹ thuật mờ để PCDL, trong đó một đối tượng dữ liệu có thể thuộc vào nhiều cụm dữ liệu khác nhau. Các thuật toán thuộc loại này chỉ ra lược đồ phân cụm thích hợp với tất cả hoạt động đời sống hàng ngày,

chúng chỉ xử lý các dữ liệu thực không chắc chắn. Thuật toán phân cụm mờ quan trọng nhất là thuật toán FCM (Fuzzy c-means) .

❖ *Phân cụm mạng Kohonen :* loại phân cụm này dựa trên khái niệm của các mạng nơ ron. Mạng Kohnen có tầng nơ ron vào và các tầng nơ ron ra. Mỗi nơ ron của tầng vào tương ứng với mỗi thuộc tính của bản ghi, mỗi một nơ ron vào kết nối với tất cả các nơ ron của tầng ra. Mỗi liên kết được gắn liền với một trọng số nhằm xác định vị trí của nơ ron ra tương ứng.

Tóm lại, các kỹ thuật PCDL trình bày ở trên đã được sử dụng rộng rãi trong thực tế, thế nhưng hầu hết chúng chỉ nhằm áp dụng cho tập dữ liệu với cùng một kiểu thuộc tính. Vì vậy, việc PCDL trên tập dữ liệu có kiểu hỗn hợp là một vấn đề đặt ra trong Data Mining trong giai đoạn hiện nay. Phần nội dung tiếp theo của luận văn sẽ trình bày tóm lược về các yêu cầu cơ bản làm tiêu chí cho việc lựa chọn, đánh giá kết quả cho các phương pháp phân cụm PCDL.

2.8 Các vấn đề có thể gặp phải

- Kỹ thuật phân nhóm hiện tại không giải quyết được tất cả các yêu cầu một cách đầy đủ (đồng thời).
- Việc phân cụm một dữ liệu với kích thước và số lượng lớn là vấn đề khó khăn bởi vì độ phức tạp thời gian tăng cao.
- Tính hiệu quả của phương pháp phụ thuộc vào định nghĩa của "khoảng cách" (khi phân cụm dựa trên khoảng cách).
- Nếu một khoảng cách không tồn tại, chúng ta phải "định nghĩa" nó, mà không phải lúc nào việc này cũng dễ dàng, đặc biệt là trong không gian đa chiều.
- Kết quả của thuật toán phân cụm có thể được hiểu theo nhiều cách khác nhau. (Trong nhiều trường hợp có thể tùy theo ý riêng của từng người).

2.9 Phương pháp phân cấp (Hierarchical Methods)

Có nhiều kỹ thuật phân cụm dữ liệu khác nhau. Việc lựa chọn phương pháp tuỳ thuộc vào yêu cầu cụ thể. Trong đồ án này trình bày về kỹ thuật phân cụm dữ liệu phân cấp.

Trong phân cụm phân cấp, tập dữ liệu được tổ chức thành một cây mà mỗi đỉnh của nó là một cụm.

Cây các cụm

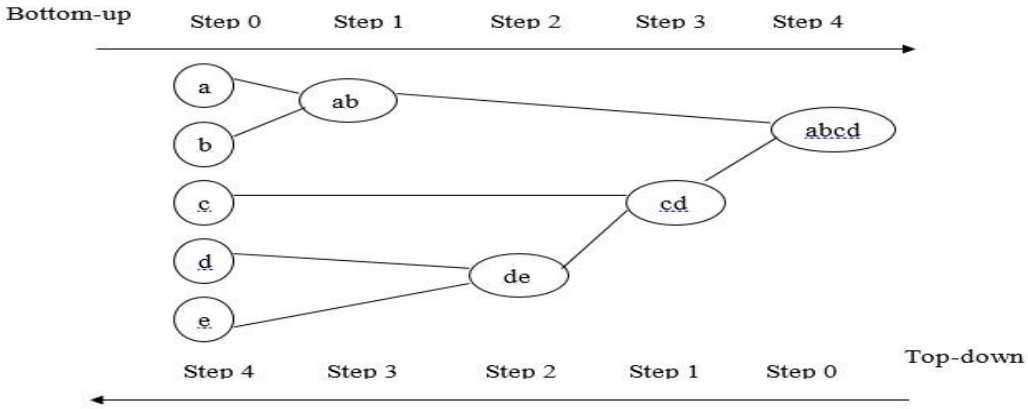
Phân cấp cụm tạo ra cây các cụm hay còn gọi là *Dendrogram.*

- Các lá của cây biểu diễn các đối tượng riêng lẻ.

- Các nút trong của cây biểu diễn các cụm.

Hai loại phương pháp tạo kiến trúc cụm

1. *Gộp – Agglomerative hay Bottom-up (từ dưới lên)*
   - Đưa từng đối tượng vào cụm riêng của nó
   - Tại mỗi bước tiếp theo, trộn hai cụm tương tự nhất cho đến khi chỉ còn một cụm hay thỏa điều kiện kết thúc.
2. *Phân chia – Divisive hay Top-down (từ trên xuống)*
   - Bắt đầu bằng một cụm lớn chứa tất cả các đối tượng
   - Phân chia cụm phân biệt nhất thành các cụm nhỏ hơn và xử lý cho đến khi có *k* cụm hay thỏa điều kiện kết thúc.



Hình 2.11: Phân cụm phân cấp Top-down và Bottom-up

Các khoảng cách giữa các cụm

Thường có 3 cách được dùng để định nghĩa khoảng cách giữa các cụm:

- Phương pháp liên kết đơn (láng giềng gần nhất):

$$d(i,j) = \min_{x \in C_i, y \in C_j} d(x,y)$$

- Phương pháp liên kết hoàn toàn (láng giềng xa nhất)

$$d(i,j) = \max_{x \in C_i, y \in C_j} d(x,y)$$

- Phương pháp liên kết trung bình (khoảng cách trung bình cặp khác nhóm)

$$d(i,j) = \text{avg}_{x \in C_i, y \in C_j} d(x,y) \cdot$$

Ưu điểm của các phương pháp phân cấp

- Khái niệm đơn giản.
- Lý thuyết tốt.

Khi cụm được trộn/tách, quyết định là nhất quán, do đó số các phương án khác nhau cần được xem xét sẽ giảm.

Điểm yếu của phương pháp phân cấp

- Trộn/Tách các cụm là nhất quán, do đó các quyết định sai là không thể khắc phục về sau.
- Các phương pháp phân chia cần thời gian tính toán lớn.
- Các phương pháp phân cấp không khả thi (triển khai) đối với các tập dữ liệu lớn.

Một vấn đề cần quan tâm trong phân cụm dữ liệu là cần xử lý các dữ liệu ngoại lai (outlier data). Dữ liệu ngoại là những dữ liệu bất thường. Chúng sinh ra do đo đạc hay trong quá trình nhập dữ liệu, là kết quả của việc biến đổi dữ liệu. Các giá trị dữ liệu ngoại lai có thể sẽ làm sai lệch các kết quả phân cụm thu được. Do đó, trước khi tiến hành phân cụm dữ liệu cần loại bỏ đi các dữ liệu ngoại lai.

2.6.1 Thuật toán BIRCH

Thuật toán phân cụm cho tập dữ liệu lớn BIRCH (*Balanced Iterative Reducing and Clustering Using Hierarchies* - Giảm cân bằng lặp và sử dụng các cấp) là thuật toán phân cụm phân cấp sử dụng chiến lược phân cụm từ trên xuống (Top-down). Ý tưởng của thuật toán là không cần lưu toàn bộ các đối tượng dữ liệu của các cụm trong bộ nhớ mà chỉ lưu các đại lượng thống kê. Đối với mỗi dữ liệu của các cụm, BIRCH chỉ lưu một bộ 3 (n, LS, SS). Bộ ba này được gọi là các đặc trưng của cụm (Cluster Features – CF).

Một số thuộc tính cụm

Cho n điểm dữ liệu p chiều trong cụm: c_i, với i=1,2…n. Trọng tâm c_0, bán kính R và đường kính D của một cụm được định nghĩa là:

Trọng tâm:

$$\mathbf{C_o} = \frac{\sum_{i=1}^{n} \mathbf{C_i}}{\mathbf{n}}$$

Bán kính:

$$\mathbf{R} = \left(\frac{\sum_{i=1}^{n} (\mathbf{C_i} - \mathbf{C_o})^2}{\mathbf{n}} \right)^{\frac{1}{2}}$$

Đường kính:

$$D = \left(\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (c_i - c_j)^2}{n(n-1)} \right)^{\frac{1}{2}}$$

Giữa 2 cụm định nghĩa 5 khoảng cách để xác định độ gần.

- Giả sử trọng tâm của 2 cụm là: c_{01} và c_{02}, 2 khoảng cách D_0 và D_1 được xác định là:

Khoảng cách Euclide:

$$D_0 = \left((c_{01} - c_{02})^2 \right)^{\frac{1}{2}}$$

Khoảng cách Manhattan:

$$D_1 = |c_{01} - c_{02}| = \sum_{i=1}^{p} \left| c_{01}^{(i)} - c_{02}^{(i)} \right|$$

- Cho n_1 điểm dữ liệu p chiều trong cụm c_i với i=1,2…n_1, và n_2 điểm dữ liệu trong một cụm khác c_j với j=n_1 +1, n_1 +2,…. n_1 +n_2.

Gọi D_2 là khoảng cách trung bình giữa các cụm, D_3 là khoảng cách trung bình nội cụm và D_4 là khoảng cách chênh lệch gia tăng của 2 cụm, chúng được tính như sau:

$$D_2 = \left(\frac{\sum_{i=1}^{n1} \sum_{j=n1+1}^{n1+n2} (c_i - c_j)^2}{n_1 n_2} \right)^{\frac{1}{2}}$$

$$D_3 = \left(\frac{\sum_{i=1}^{n1+n2} \sum_{j=1}^{n1+n2} (c_i - c_j)^2}{(n_1 + n_2)(n_1 + n_2 - 1)} \right)^{\frac{1}{2}}$$

$$D_4 = \sum_{k=1}^{n1+n2} \left(c_k - \frac{\sum_{l=1}^{n1+n2} c_l}{n_1 + n_2} \right)^2 - \sum_{i=1}^{n1} \left(c_i - \frac{\sum_{l=1}^{n1} c_l}{n_1} \right)^2 - \sum_{j=n1+1}^{n1+n2} \left(c_j - \frac{\sum_{l=n1+1}^{n1+n2} c_l}{n_2} \right)^2$$

Thực tế D_3 là D của cụm gộp. Với D_0, D_1, D_2, D_3, D_4 chúng ta có thể sử dụng chúng để xác định độ gần của 2 cụm.

Đặc trưng cụm (Clustering Feature - CF):

Là một bộ 3 lưu giữ thông tin của cụm. Cho n điểm dữ liệu p chiều trong cụm C_i với i=1,2,…n, đặc trưng cụm (CF) được định nghĩa là 1 bộ 3:

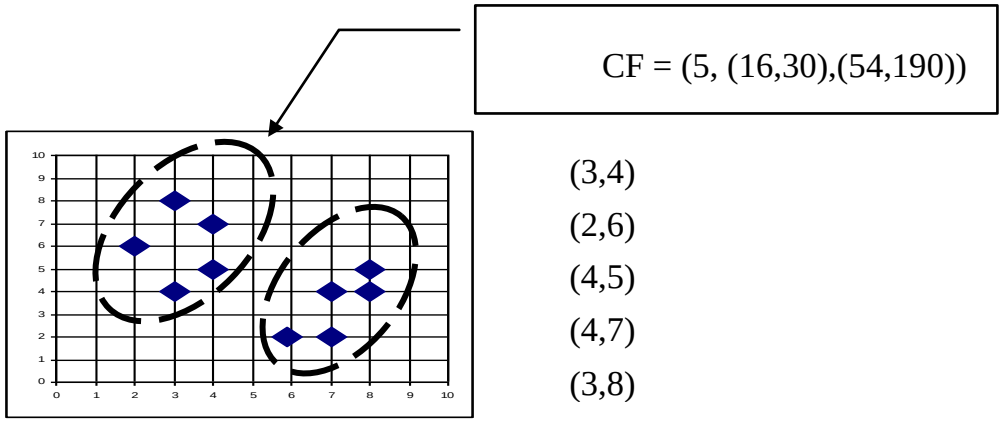
$CF = (n, LS, SS)$.

Trong đó:

n: *là số đối tượng (điểm) trong cụm.*

LS*: là tổng các giá trị thuộc tính của các đối tượng trong cụm,* $\sum_{i=1}^{n} c_i$

SS*: là tổng bình phương của các giá trị thuộc tính của các đối tượng trong cụm* $\sum_{i=1}^{n} c_i^2$



Hình 2.12: Xác định CF

Lý thuyết cộng CF:

Giả sử có $CF_1 = (n_1, LS_2, SS_1)$, và $CF_2 = (n_2, LS_2, SS_2)$ là các đặc trưng cụm của 2 cụm rời nhau. Khi đó CF gộp của 2 cụm này được tính là:

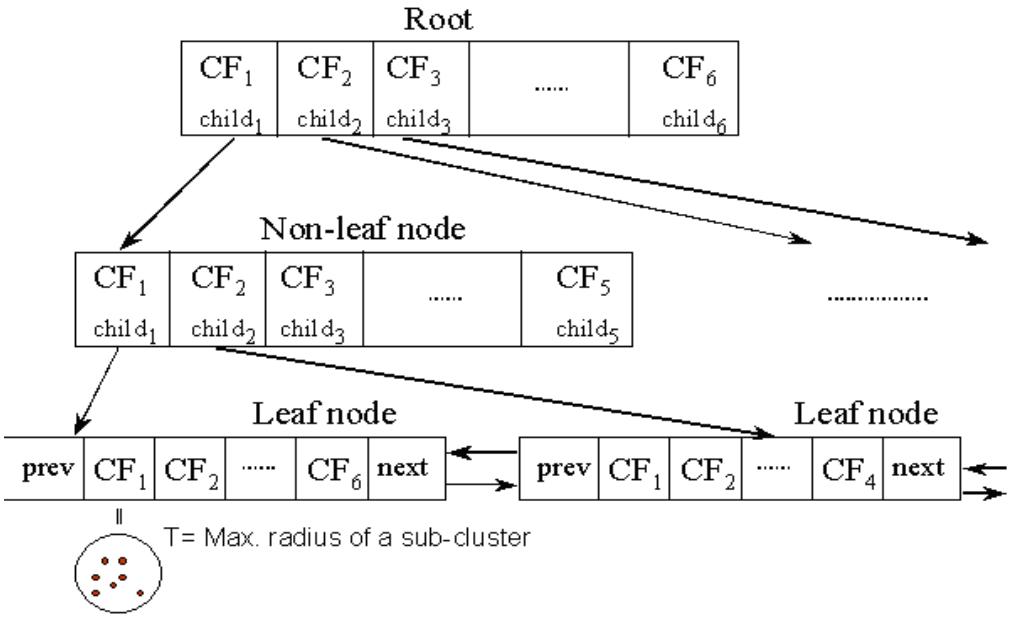
$$CF_1 + CF_2 = (N = n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2)$$

Từ định nghĩa CF và lý thuyết cộng, chúng ta biết rằng ứng với C_0, R, D, D_0, D_1,D_2, D_3, và D_4 đều có thể tính một cách dễ dàng.

Cây CF:

Các đặc trưng của cụm (Cluster Features – CF) được lưu giữ trong một cây gọi là cây CF (CF tree). Người ta đã chứng minh rằng, các đại lượng thống kê

chuẩn như độ đo khoảng cách, có thể xác định cây CF. Hình dưới đây biểu thị một ví dụ về cây CF. Chúng ta thấy rằng tất cả các nút trong lưu tổng các đặc trưng cụm CF của nút con, trong khi đó các nút lá lưu trữ các đặc trưng của cụm dữ liệu.



Hình 2.13: Ví dụ về cây CF

Cây CF là cây cân bằng, nhằm để lưu trữ các đặc trưng của cụm (CF). Cây CF chứa các nút trong và nút lá, nút trong là nút chứa các nút con và nút là thì không có con. Nút trong lưu trữ tổng các đặc trưng cụm (CF) của các nút con của nó.

Một cây CF được đặc trưng bởi 2 tham số:

- *Yếu tố phân nhánh (Branching Factor-**B**):* Nhằm xác định tối đa các nút con của một nút trong. Mỗi nút trong chữa nhiều nhất B mục dạng [CF_i, $child_i$], với i=1,2…B, $child_i$ là một con trỏ tới nút con thứ i, và CF_i là CF của cụm con thể hiện bởi child. Nút lá chứa nhiều nhất L mục dạng [CF_i], với i=1,2…L.
- *Ngưỡng (Threshloid-**T**):* Khoảng cách tối đa giữ bất kì một cặp đối tượng trong nút lá của cây, khoảng cách này còn được gọi là bán kính hoặc đường kính của các cụm con được lưu tại các nút lá.

Hai tham số này có ảnh hưởng đến kích thước của cây CF.

Cây CF được xây dựng tự động như là việc chèn thêm đối tượng dữ liệu mới. Nó được sử dụng cho việc định hướng chèn mới chính xác vào các cụm con với

mục đích phân cụm. Thực tế, tất cả các cụm được tạo thành như mỗi điểm dữ liệu được đưa vào cây CF.

Cây CF là biểu diễn rất nhỏ của bộ dữ liệu vì mỗi mục trong một nút lá không phải là một điểm dữ liệu duy nhất mà là một cụm con

Chèn một mục (entry) vào cây CF

Cho mục "Ent", tiến hành chèn vào cây CF như sau:

1. *Xác định lá phù hợp (identifying the appropriate leaf):* Bắt đầu từ gốc tiếp đó là lựa chọn khoảng cách metric D_0, D_1, D_2, D_3,D_4 như đã được định nghĩa trước, đệ quy sâu xuống cây CF bằng cách chọn nút con gần nhất.
2. *Điều chỉnh lá (Modifying the leaf):* Khi nó đạt được một nút lá, nó tìm mục lá gần nhất, và kiểm tra các nút có thể hấp thụ nó mà không vi phạm các điều kiện ngưỡng.
   - Nếu được, các CF cho nút được cập nhật.
   - Nếu không, một mục mới cho nó được thêm vào lá.
     - Nếu có không gian trên lá cho mục nhập với này, chúng ta đã hoàn thành.
     - Nếu không chúng ta phải chia nút lá. Nút tách được thực hiện bằng cách chọn cặp xa nhất các mục dựa trên các tiêu chí gần nhất.
3. *Thay đổi đường dẫn đến lá (Modifying the path to the leaf):* Sau khi đưa "Ent" vào 1 lá, chúng ta phải cập nhật các thông tin CF cho mỗi mục nút trong trên đường dẫn đến nút lá.
   - Trong trường hợp không có sự phân chia, điều này đơn giản là liên quan đến việc bổ sung thêm các CF nhằm phản ánh việc bổ sung thêm "Ent".
   - Chia nút lá đòi hỏi chúng ta chèn một mục nút trong mới vào nút cha, để mô tả nút lá mới được tạo ra.
     1. Nếu nút cha có không gian cho mục này, ở tất cả các cấp độ cao hơn, chúng ta chỉ cần cập nhật các CF để phản ánh việc bổ sung "Ent".
     2. Nếu không, chúng ta có thể phải chia nút cha, và như vậy lên đến nút gốc.
4. *Hoàn thiện hợp nhất (Merging Refinement):* Dữ liệu đầu vào có sự hiện diện của dữ liệu sai lệch, phân chia có thể ảnh hưởng đến chất lượng phân nhóm, và cũng

làm giảm việc khai thác, sử dụng không gian. Thêm các kết hợp đơn giản thường xuyên sẽ giúp cải thiện những vấn đề này: giả sử quá trình chia cắt dừng lại ở một số nút trong N_j, N_j có thể chứa các mục nhập thêm do thực hiện phân chia.

i. *Quét N_j để tìm hai mục gần nhất.*

ii. *Nếu chúng không phải là cặp phù hợp với sự phân chia, hợp nhất chúng.*

iii. *Nếu có nhiều mục hơn một trang nhớ có thể giữ, chia lại*

iv. *Quá trình tái phân chia, nguyên nhân sáp nhập đầy đủ các mục, mặt khác tiếp nhận các mục còn lại*

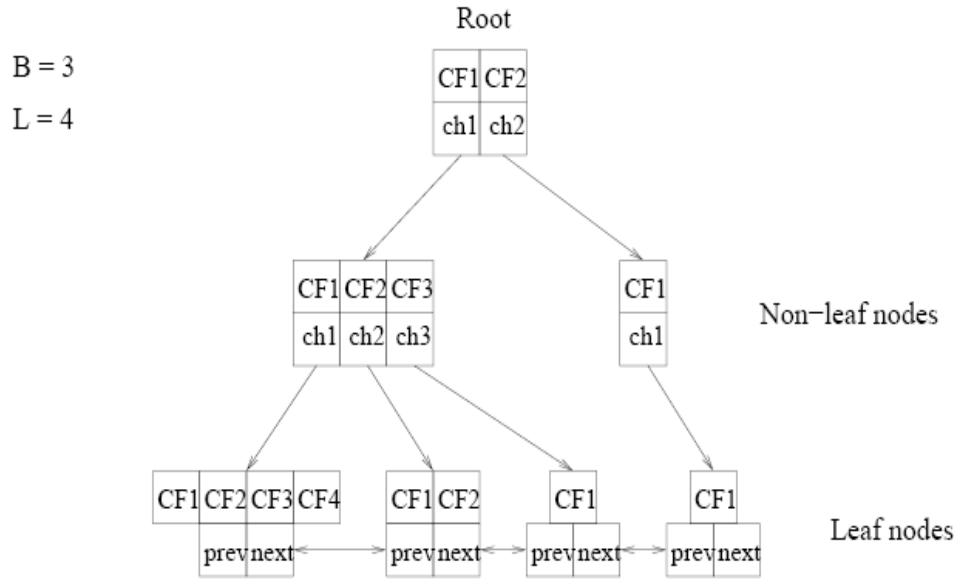
Tóm lại, nó cải thiện sự phân bố của các mục trong 2 con gần nhất, thậm chí còn tạo ra một không gian nút trống để sử dụng sau.

Ghi chú: Mốt số vấn đề sẽ được giải quyết trong các bước sau của toàn bộ thuật toán.

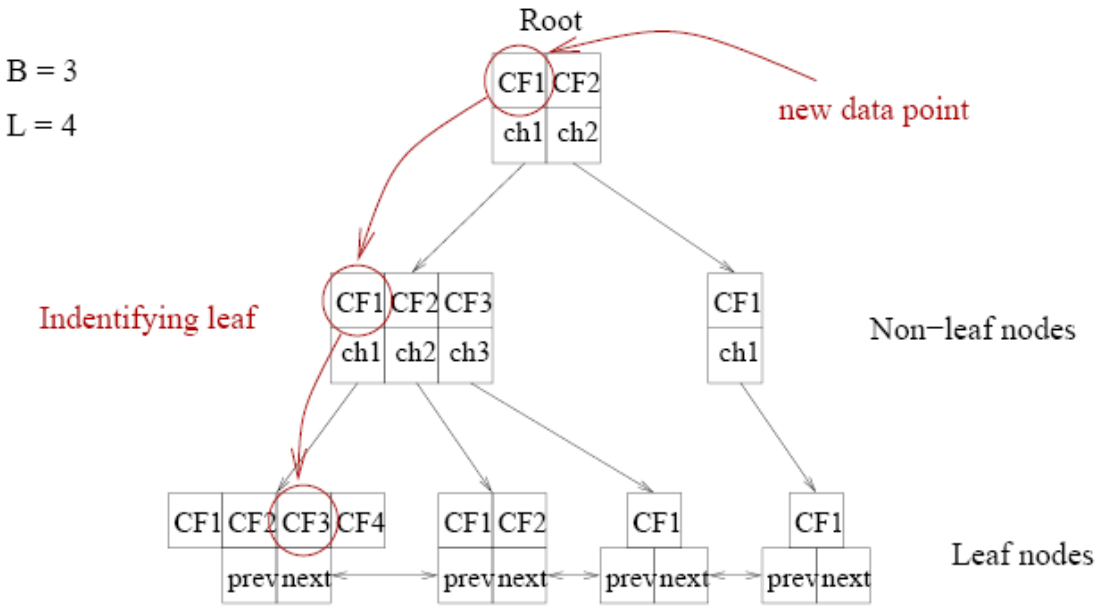
Vấn đề 1: Phụ thuộc vào thứ tự của dữ liệu đầu vào và mức độ sai lệch, hai cụm con mà không phải ở trong một cụm được lưu giữ trong cùng một nút. Nó sẽ được khắc phục với một thuật toán chung sắp xếp các mục trên các nút lá (Bước 3)

Vấn đề 2: Nếu điểm cùng một dữ liệu được chèn vào 2 lần, nhưng lại tại 2 thời điểm khác nhau, hai bản sao có thể được nhập vào mục (entries) lá riêng biệt. Nó sẽ được xử lý tại Bước 4 của thuật toán.

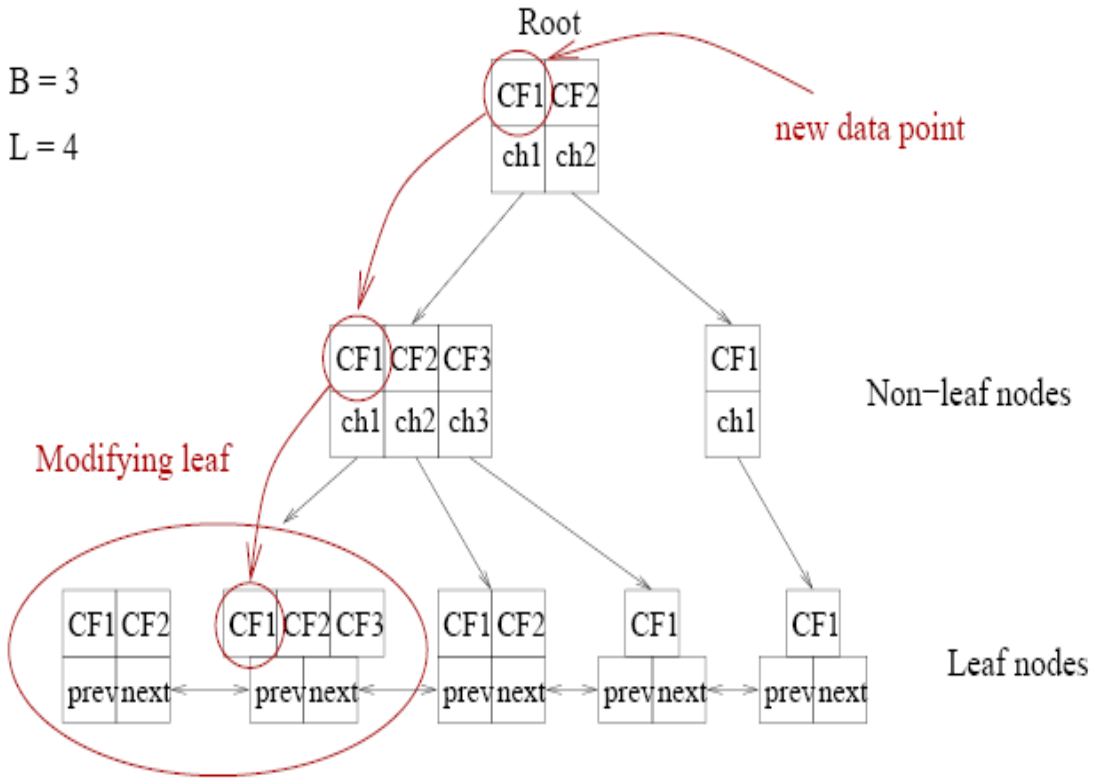
Mô tả quá trình chèn một mục vào cây CF



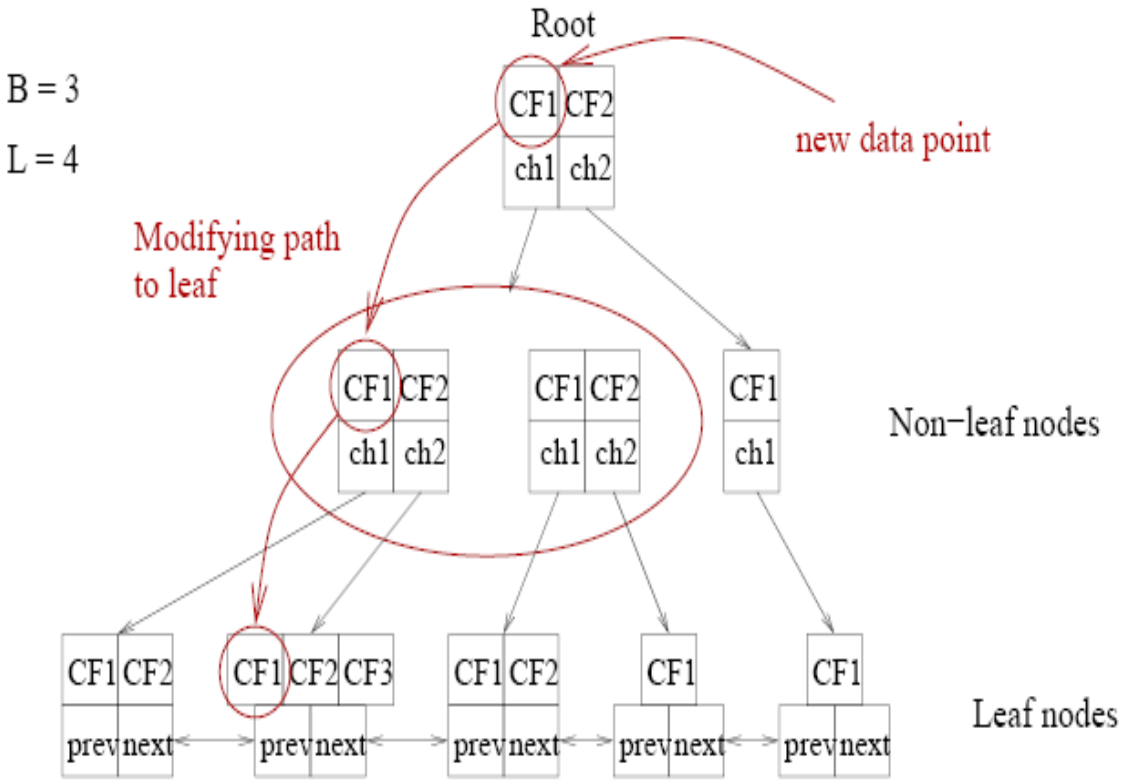
Hình 2.14: Cây CF ban đầu



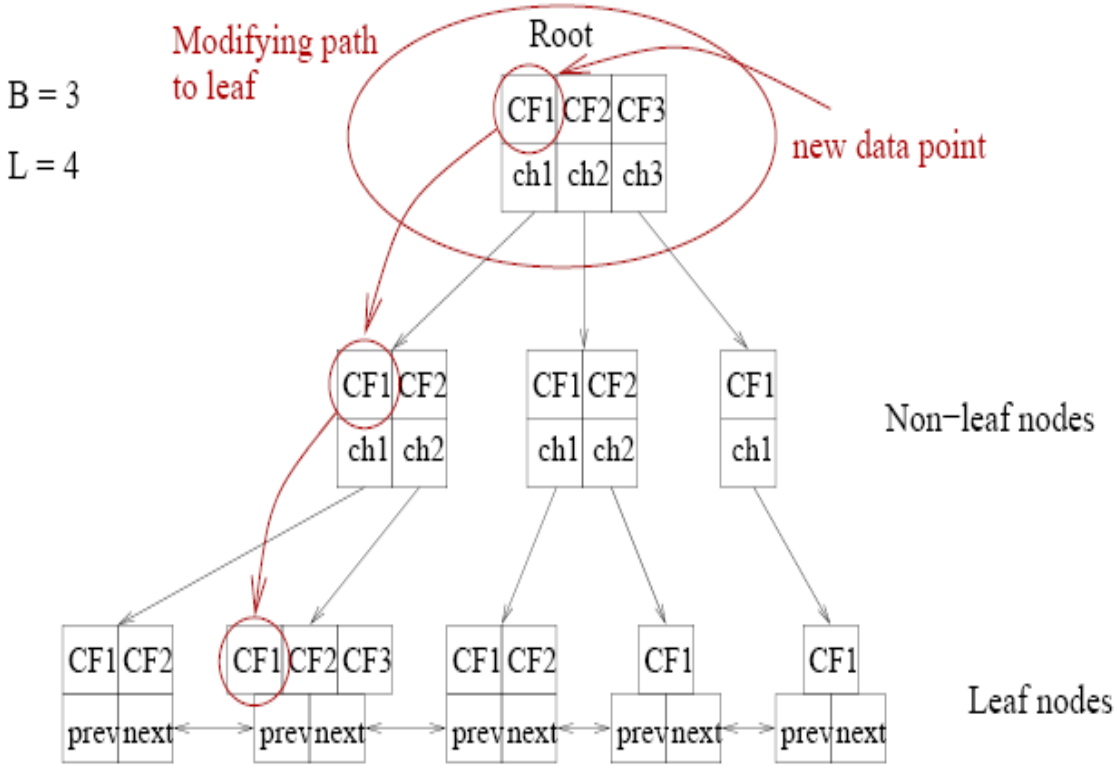
Hình 2.15: Xác định lá phù hợp



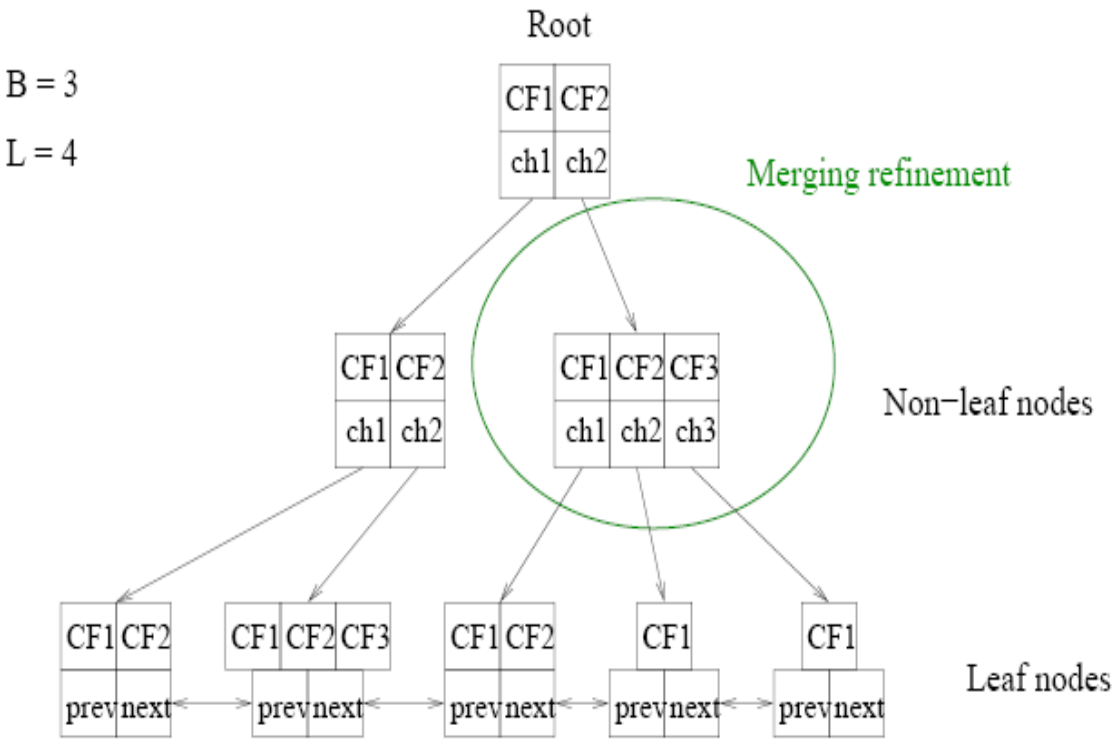
Hình 2.16: Điều chỉnh lá



Hình 2.17: Thay đổi đường đi tới lá



Hình 2.18: Thay đổi đường đi tới lá (tiếp)



Hình 2.19: Hoàn thiện hợp nhất

Các giai đoạn sau thực hiện của BIRCH:

Giai đoạn 1: BIRCH duyệt qua tất cả các đối tượng trong cơ sở dữ liệu và xây dựng một cây CF khởi tạo. Trong giai đoạn này, các đối tượng lần lượt được chèn vào nút lá gần nhất của cây CF (nút lá của cây đóng vai trò là cụm con). Sau khi chèn xong thì tất cả các nút trong cây CF được cập nhật thông tin. Nếu đường kính của cụm con sau khi chèn là lớn hơn ngưỡng T, thì nút lá được tách.

Quá trình này lặp lại cho đến khi tất cả các đối tượng đều được chèn vào trong cây. Ở đây ta thấy rằng, mỗi đối tượng trong cây chỉ được đọc một lần, để lưu toàn bộ cây CF trong bộ nhớ thì cần phải điều chỉnh kích thước của cây CF thông qua điều chỉnh ngưỡng T.

Giai đoạn 2: BIRCH lựa chọn một thuật toán phân cụm dữ liệu để thực hiện phân cụm dữ liệu cho các nút lá của cây.

Các bước cơ bản của Thuật toán BIRCH:

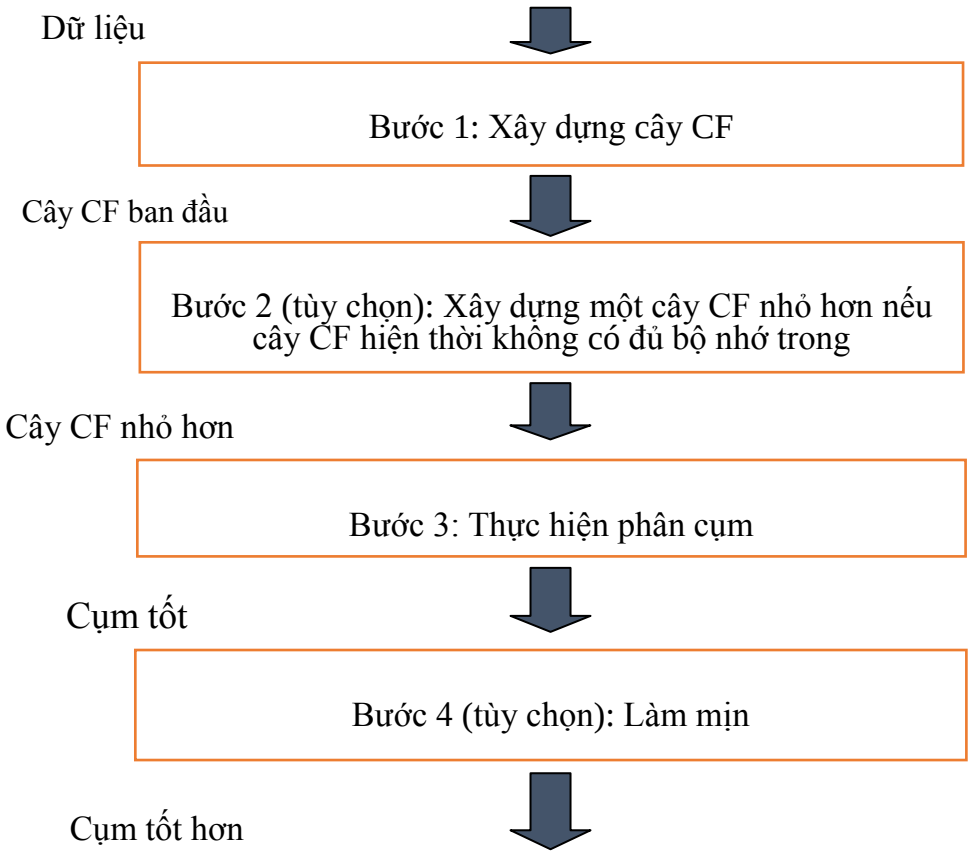
Bước 1: Các đối tượng dữ liệu lần lượt được chèn vào cây CF, sau khi chèn hết các đối tượng ta thu được cây CF khởi tạo. Mỗi một đối tượng được chèn vào nút lá gần nhất tạo thành cụm con. Nếu đường kính của cụm con này lớn hơn T thì nút lá được tách. Khi một đối tượng thích hợp được chèn vào nút lá thì tất cả các nút trở tới gốc của cây được cập nhật với các thông tin cần thiết.

Bước 2: Nếu cây CF hiện thời không có đủ bộ nhớ trong thì tiến hành dựng cây CF nhỏ hơn: kích thước của cây CF được điều khiển bởi tham số T và vì vậy việc chọn một giá trị lớn hơn cho nó sẽ hòa nhập một số cụm con thành một cụm, điều này làm cho cây CF nhỏ hơn. Bước này không cần yêu cầu bắt đầu đọc dữ liệu lại từ đầu nhưng vẫn đảm bảo hiệu chỉnh cây dữ liệu nhỏ hơn.

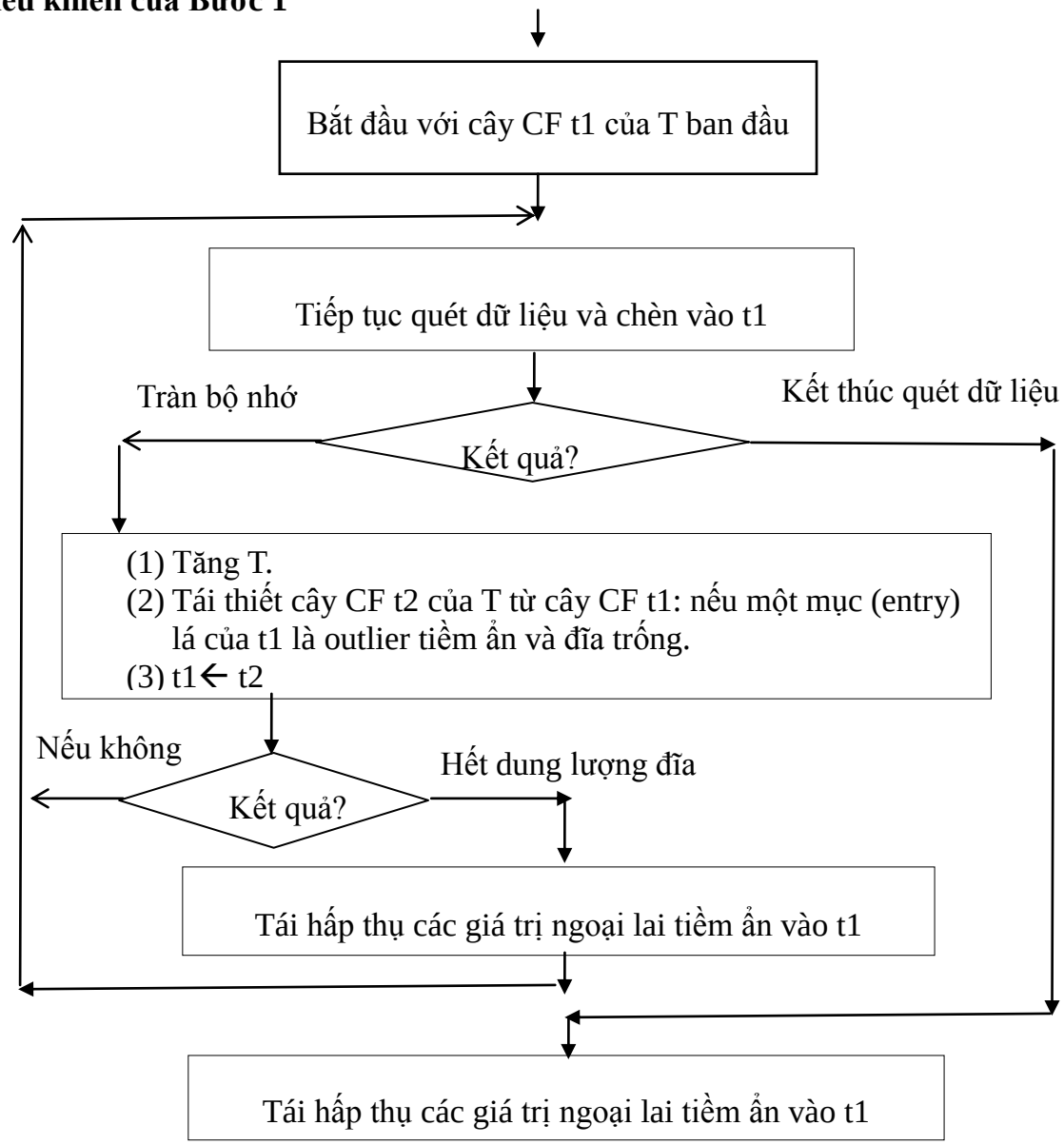
Bước 3: Thực hiện phân cụm: các nút lá của cây CF lưu giữ các đại lượng thống kê của các cụm con. Trong bước này, BIRCH sử dụng các đại lượng thống kê này để áp dụng một số kĩ thuật phân cụm ví dụ như k-means và tạo ra một khởi tạo cho phân cụm.

Bước 4: Phân phối lại các đối tượng dữ liệu bằng cách dùng các đối tượng trọng tâm cho các cụm đã được khám phá từ bước 3: đây là một bước tùy chọn để duyệt lại tập dữ liệu và gán nhãn lại cho các đối tượng dữ liệu tới các trọng tâm gần nhất. Bước này nhằm để gán nhãn cho các dữ liệu khởi tạo và loại bỏ các đối tượng ngoại lai.

Khái quát thuật toán phân cụm BIRCH



Dòng điều khiển của Bước 1



Các vấn đề cần quan tâm ở Bước 1:

– Xây dựng lại cây CF.

– Giá trị ngưỡng T.

– Outlier-handling Option.

– Delay Split Option.

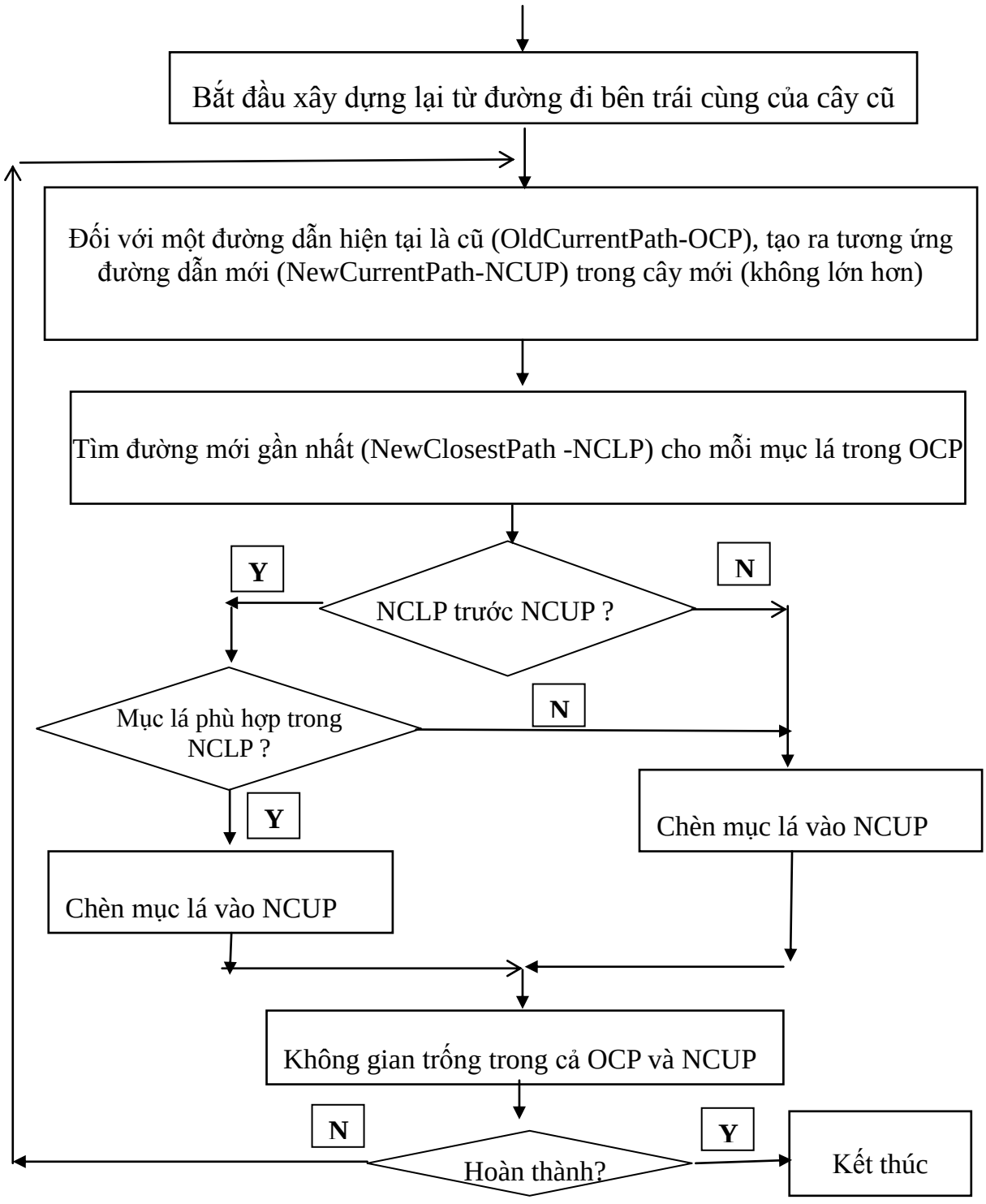
1) Xây dựng lại cây CF

Trong bước này ta sử dụng tất cả các mục lá của cây CF cũ để xây dựng lại một cây CF mới với ngưỡng lớn hơn. Trong quá trình xây dựng lại ta cần điều chỉnh đường đi tới nút lá. Đường đi tới nút lá tương ứng với một đường đi duy nhất tới nút

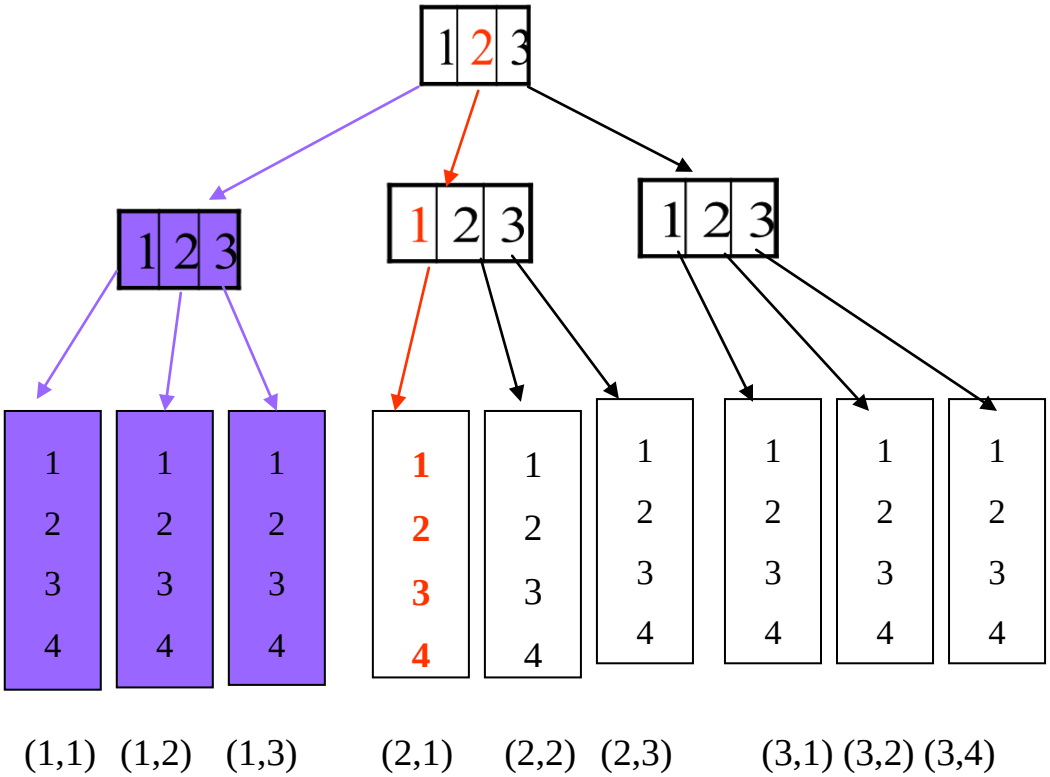
lá. Thuật toán xây dựng lại là các thuật toán quét và giải phóng đường đi cây cũ, và tạo ra đường đi cho cây mới.

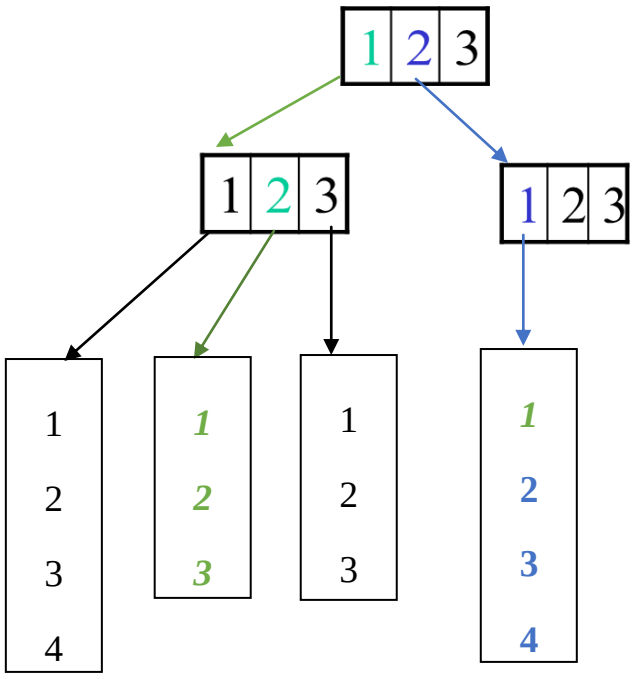
Kích thước của cây mới phải nhỏ hơn cây trước. Việc chuyển từ cây cũ sang cây mới cần ít nhất thêm h trang bộ nhớ, trong đó h là chiều cao cây cũ.

Thuật toán tái xây dựng ở Bước 1:



Xây dựng lại cây CF





Giải phóng các nút trong cây	OldCurrentPath (OCP)
NewClosestPath (NCLP)	NewCurrentPath(NCUP)

2) Giá trị ngưỡng T

Để tăng ngưỡng ta sử dụng phương pháp *heuristic.* Lựa chọn giá trị ngưỡng mới sao cho số lượng các điểm dữ liệu được quét dưới giá trị ngưỡng mới:

Phương pháp 1: Tìm nút lá đông nhất và hai mục gần nhất trên lá có thể được sáp nhập dưới ngưỡng mới.

Phương pháp 2: Giả sử lượng chiếm đóng các cụm lá tăng tuyến tính với điểm dữ liệu. một loạt các cặp giá trị: số lượng các điểm dữ liệu và khối lượng => khối lượng mới (một điểm dữ liệu mới, sử dụng tối thiểu hồi quy tuyến tính) => ngưỡng mới. Sử dụng một số phương pháp *heuristic* để điều chỉnh hai ngưỡng trên và chọn một.

3) Outlier-handling option

Outlier là là một giá trị ngoại lai hay nhiễu, trong cây CF nó đóng vai trò là một mục lá mật độ thấp, nó được đánh giá là quan trọng đối với mô hình phân nhóm tổng thể. Sử dụng một số không gian đĩa để xử lý giá trị ngoại lai.

Khi xây dựng lại các cây CF, một mục lá cũ chỉ được ghi vào đĩa nếu nó được coi là một outlier tiềm năng. Điều này có thể làm giảm kích thước của cây CF.

Một outlier không đủ tiêu chuẩn khi:

- Tăng trong giá trị ngưỡng;
- Sự thay đổi trong việc phân phối do nhiều dữ liệu được đọc.

Quét các outlier tiềm năng để hấp thu mà không gây ra phát triển quá kích thước cây:

- Hết không gian đĩa.
- Tất cả các điểm dữ liệu đã được quét.

4) Delay-Split option

Khi hết bộ nhớ. Có thể có nhiều hơn các điểm dữ liệu phù hợp trong cây CF hiện tại. Chúng ta có thể tiếp tục đọc dữ liệu điểm và ghi những điểm dữ liệu cần chia một nút vào đĩa cho đến khi hết không gian đĩa. Ưu điểm của phương pháp này là nhiều hơn các điểm dữ liệu phù hợp trong cây trước khi chúng ta phải xây dựng lại.

Đánh giá thuật toán BIRCH

Với cấu trúc cây CF được sử dụng, BIRCH có tốc độ thực hiện phân cụm dữ liệu nhanh và có thể áp dụng đối với tập dữ liệu lớn, đặc biệt, BIRCH hiệu quả khi áp dụng với tập dữ liệu tăng trưởng theo thời gian. BIRCH chỉ duyệt toàn bộ dữ liệu một lần với một lần quét thêm tùy chon, nghĩa là độ phức tạp của nó là O(n), với n là đối tượng dữ liệu. Thuật toán này kết hợp các cụm gần nhau và xây dựng lại cây CF, tuy nhiên mỗi nút trong cây CF có thể chỉ lưu trữ một số hữu hạn bởi kích thước của nó.

Hạn chế: Thuật toán có thể không xử lý được tốt nếu các cụm không có hình dạng cầu, bởi vì nó sử dụng khái niệm bán kính hoặc đường kính để kiểm soát ranh giới các cụm và chất lượng của các cụm được khám phá không được tốt. Nếu BIRCH sử dụng khoảng cách Eucle, nó thực hiện tốt chỉ với dữ liệu số. Mặc khác, tham số vào T có ảnh hưởng rất lớn tới kích thước và tính tự nhiên của cụm. Việc ép các đối tượng dữ liệu làm cho các đối tượng của một cụm có thể là đối tượng kết thúc cụm khác, trong khi các đối tượng gần nhau có thể hút bởi các cụm khác nếu chúng được biểu diễn cho thuật toán theo một thứ tự khác. BIRCH không thích hợp với dữ liệu đa chiều.

2.6.2 Thuật toán CURE

CURE (Clustering Using Representatives – Phân cụm dữ liệu sử dụng điểm đại diện) là thuật toán sử dụng chiến lược dưới lên (Botton-Up) của kĩ thuật phân cụm phân cấp.

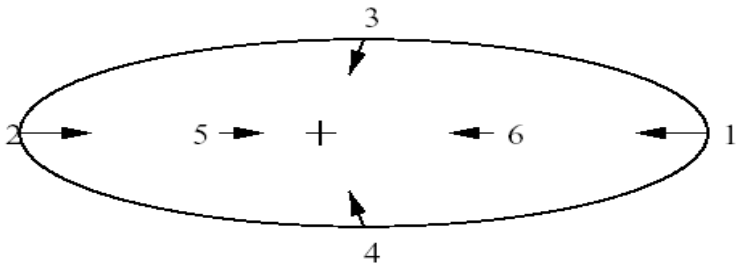
Trong hầu hết các thuật toán thực hiện phân cụm với các cụm hình cầu và kích thước tương tự, như vậy là không hiệu quả khi xuất hiện các phần tử ngoại lai. Thuật toán này định nghĩa một số cố định các điểm đại diện nằm rải rác trong toàn bộ không gian dữ liệu và được chọn để mô tả các cụm được hình thành. Các điểm này được tạo ra bởi trước hết lựa chọn các đối tượng nằm rải rác trong cụm và sau đó "co lại" hoặc di chuyển chúng về trung tâm cụm bằng nhân tố co cụm. Quá trình này được lặp lại và như vậy trong quá trình này, có thể đo tỷ lệ gia tăng của cụm. Tại mỗi bước của thuật toán, hai cụm có cặp các điểm đại diện gần nhau (mỗi điểm trong cặp thuộc về mỗi cụm khác nhau) được hòa nhập.

Như vậy, có nhiều hơn một điểm đại diện mỗi cụm cho phép CURE khám phá được các cụm có hình dạng không phải hình cầu. Việc co lại các cụm có tác dụng làm giảm tác động của các phần tử ngoại lai. Như vậy, thuật toán này có khả năng xử lý tốt trong trường hợp có các phần tử ngoại lai và làm cho hiệu quả với

hình dạng không phải hình cầu và kích thước độ rộng biến đổi. Hơn nữa, nó tỷ lệ tốt với CSDL lớn mà không làm giảm chất lượng phân cụm..

Đặt **c** là điểm đại diện cho mỗi cụm, chọn **c** rải rác sao cho có thể nắm bắt được hình dạng vật lý và hình học của cụm.

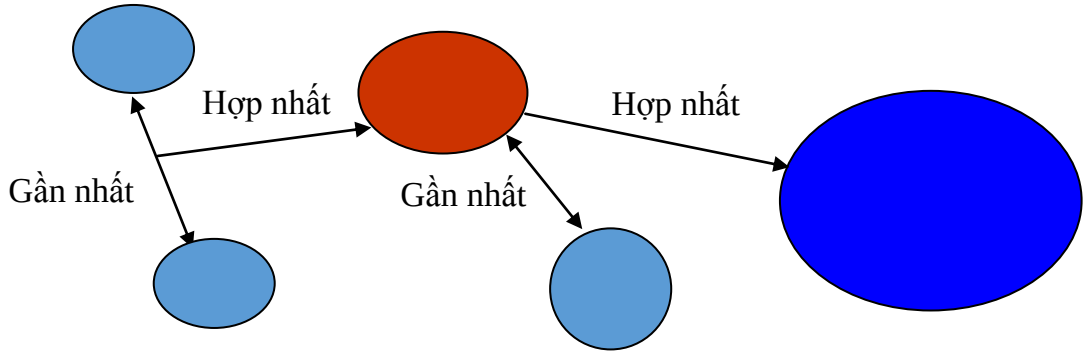
Các điểm này sau đó được "*co lại*" hoặc di chuyển chúng về trung tâm cụm bằng nhân tố co cụm α trong đó 0 <α <1.



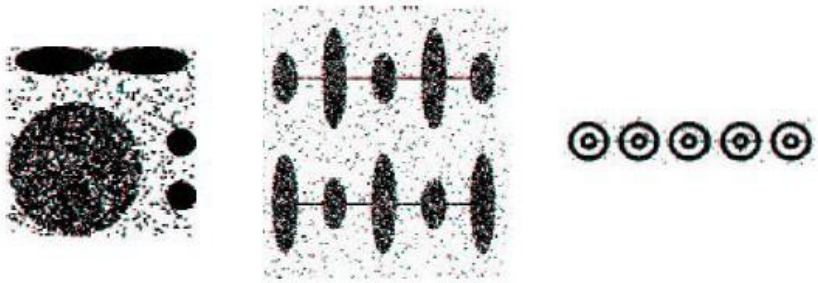
Những điểm này được sử dụng như là đại diện và ở mỗi bước của thuật toán, hai cụm với cặp gần gũi nhất của các đại diện được sáp nhập (d_{min}).

Sau mỗi lần sáp nhập, một điểm **c** được lựa chọn từ các đại diện ban đầu của cụm trước đó để đại diện cho cụm mới.

Cụm sáp nhập dừng cho đến khi mục tiêu k cụm được tìm thấy.



Với cách thức sử dụng nhiều hơn một phần tử đại diện cho các cụm, CURE có thể khám phá được các cụm có các hình thù và kích thước khác nhau trong cơ sở dữ liệu lớn. Việc co các đối tượng đại diện lại có tác dụng làm giảm tác động của các phần tử ngoại lai, vì vậy, CURE có khả năng xử lý đối với các phần tử ngoại lai. Hình dưới đây là ví dụ về các dạng và kích thước cụm dữ liệu được khám phá bởi CURE.



Hình 2.20: Cụm dữ liệu khai phá bởi thuật toán CURE

Để xử lý được các CSDL lớn, CURE sử dụng ngẫu nhiên và phân hoạch, một mẫu là được xác định ngẫu nhiên trước khi được phân hoạch, và sau đó được tiến hành phân cụm trên mỗi phân hoạch, như vậy mỗi phân hoạch là từng phần đã được phân cụm, các cụm thu hoạch, như vậy mỗi phân hoach là từng phần đã được phân cụm, các cụm thu được lại được phân cụm lần thứ hai để thu được các cụm con mong muốn, nhưng mẫu ngẫu nhiên không nhất thiết đưa ra một mô tả tốt cho toàn bộ tập dữ liệu.

Thuật toán CURE được thực hiện qua các bước cơ bản sau

Bước 1: Chọn một mẫu ngẫu nhiên từ tập dữ liệu ban đầu.

Bước 2: Phân hoạch mẫu này thành nhiều nhóm dữ liệu có kích thước bằng nhau: ý tưởng ở đây là phân hoạch mẫu thành p nhóm dữ liệu bằng nhau, kích thước của mỗi phân hoạch là n'/p(n' là kích thước mẫu).

Bước 3: Phân cụm các điểm của mỗi nhóm : Thực hiện PCDL cho các nhóm cho đến khi mỗi nhóm được phân thành n'/pq(với q>1).

Bước 4: Loại bỏ các phần tử ngoại lai: Trước hết, khi các cụm được hình thành cho đến khi số các cụm giảm xuống một phần so với số các cụm ban đầu. Sau đó, trong trường hợp các phần tử ngoại lai được lấy mẫu cùng với quá trình pha khởi tạo mẫu dữ liệu, thuật toán sẽ tự động loại bỏ các nhóm nhỏ.

Bước 5: Phân cụm các cụm không gian : các đối tượng đại diện cho các cụm di chuyển về hướng trung tâm cụm, nghĩa là chúng được thay thế bởi các đối tượng gần trung tâm hơn.

Bước 6: Đánh dấu dữ liệu với các nhãn tương ứng.

Giả mã của thuật toán CURE:

Procedure cluster(*S*,*k*)
Begin
T:=build_kd_tree(*S*)

```
Q:=build_heap(S)
while size(Q) > k do {
        u:=extract_min(Q)
        v:=u.closest
        delete(Q,v)
        w:=merge(u,v)
        delete_rep(T,u); delete_rep(T,v); insert_rep(T,w)
        w.closest:=x /* x is an arbitrary cluster in Q */
        for each x ∈ Q do{
                if dist (w,x) < dist(w,w.closest)
                        w.closest:=x
                if x.closest is either u or v {
                  if dist(x,x.closest) < dist(x,w)
                        x.closest:=closest_cluster(T,x,dist(x,w))
                   else
                        x.closest:=w
                  relocate(Q,x)
                }
                else if dist(x,x.closest)> dist(x,w) {
                  x.closest:=w
                  relocate(Q,x)
                }
            }
        insert(Q,w)
}
End


Procedure merge(u,w)

Begin
        w:=u ∪ v
        w.mean:= (|u|.u.mean + |v|v.mean) / (|u| + |v|)
        tmpSet:=∅
        for i:=1 to c do {
                maxDist:=0
                foreach point p in cluster w do {
                  if i=1
                        minDist:=dist(p,w.mean)
                  else
                        minDist:=min{dist(p,q):q∈ tmpSet}
                if (minDist ≥ maxDist) {
                   maxDist:=mainDist
                   maxPoint:=p
```

```
        }
    tmpSet:=tmpSet ∪ { maxPoint}
    }
    Foreach point p in tmpSet do
    w.rep:=w.rep ∪ {p+α*(w.mean-p)}
return w
```

End

Độ phức tạp tính toán của thuật toán CURE là $O(n^2 log(n))$. Độ phức tạp không gian là O (n) do việc sử dụng kd-tree và heap. CURE là thuật toán tin cậy trong việc khám phá ra các cụm với hình thù bất kỳ và có thể áp dụng tốt đối với dữ liệu có phần tử ngoại lai và trên các tập dữ liệu hai chiều. Tuy nhiên, nó lại rất nhạy cảm với các tham số như số các đối tượng đại diện, tỉ lệ co của các phần tử đại

Đánh giá thuật toán CURE

Ưu điểm:

Bằng cách sử dụng trên một đại diện cho một cụm, CURE có khả năng khám phá được các cụm có hình thù và kích thước bất kỳ trong tập dữ liệu lớn. Việc co các đối tượng đại diện có tác dụng làm giảm tác động của các đối tượng ngoại lai. Do đó CURE có thể xử lý tốt các đối tượng ngoại lai. Tốc độ thực hiện của CURE nhanh O(N).

Nhược điểm:

CURE là dễ bị ảnh hưởng bởi các tham số cho bởi người dùng như cỡ mẫu, số cụm mong muốn.

2.10 Kết luận chương

Khai khá dữ liệu là vấn đề rất cần thiết trong cuộc sống của chúng ta. Từ khai phá dữ liệu chúng ta có thể phát hiện ra tri thức. Có rất nhiều cách để con người khai phá dữ liệu và một trong những cách đó là phân cụm dữ liệu.

Chương này đã trình bày khái quát về kỹ thuật phân cụm dữ liệu: phát biểu bài toán, các kiểu dữ liệu thường gặp, các độ đo đo khoảng cách, độ tương tự, các phương pháp tiếp cận bài toán phân cụm dữ liệu. Phương pháp tiếp cận phân cấp với 2 thuật toán CURE và BIRCH là nội dung chính của chương và đã được trình bày chi tiết.

CHƯƠNG 3: CHƯƠNG TRÌNH DEMO

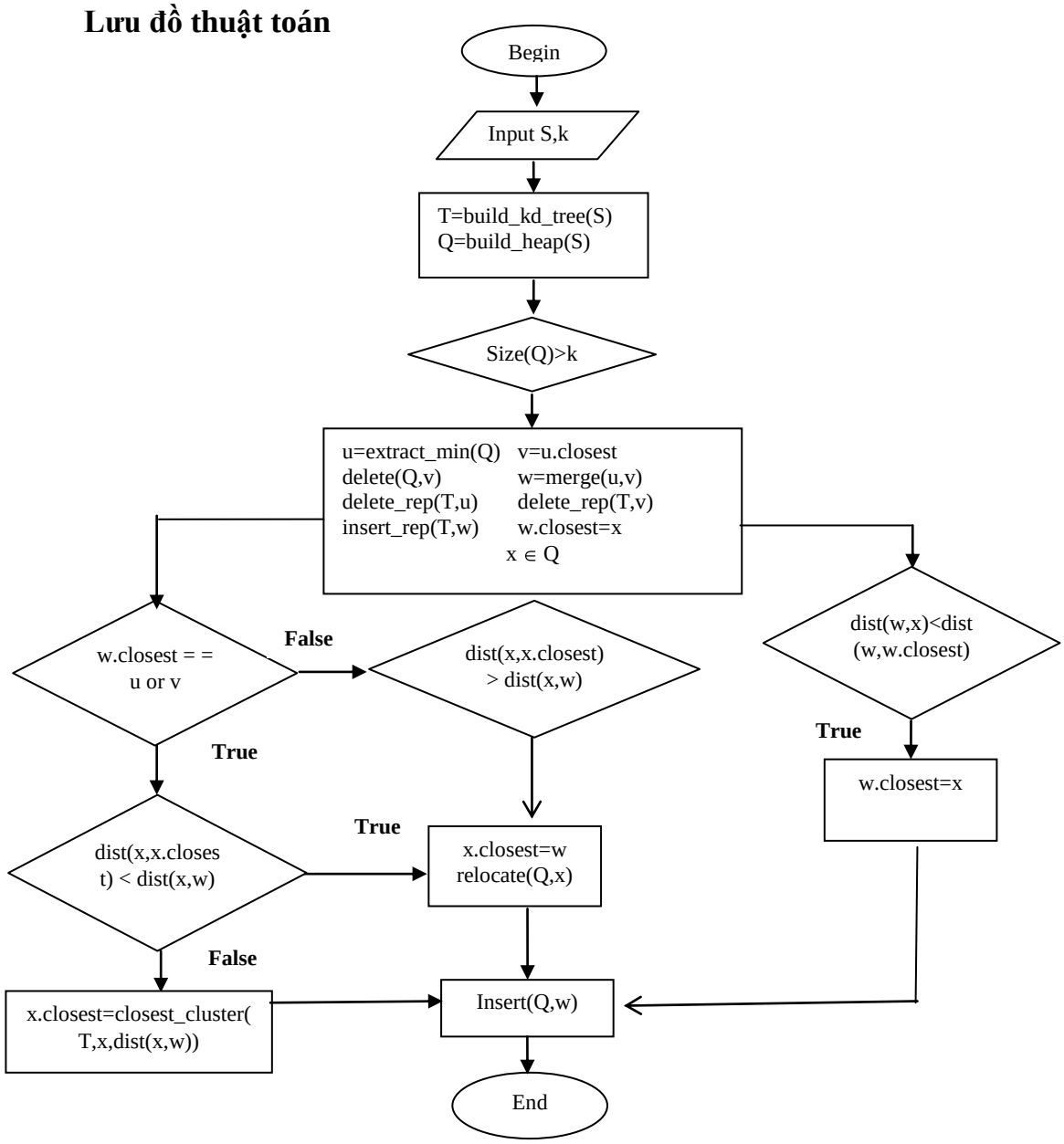
Để có cái nhìn trực quan về thuật toán phân cụm phân cấp em xin trình bày một chương trình demo đơn giản của thuật toán CURE.

3.1. Bài toán và lưu đồ thuật toán

Input: Mảng S chứa các điểm trong không gian 2 chiều, mỗi điểm đặc trưng bởi 2 thuộc tính (x,y), k số cụm cần chia.

Output: Các nhóm (cụm) điểm, trong đó các điểm gần nhau sẽ được gom vào một nhóm với mục tiêu là k cụm theo thuật toán CURE.

Lưu đồ thuật toán



3.2. Chương trình demo

Chương trình được lập trình bằng ngôn ngữ java trên nền JDK 1.7.0, phần mềm hỗ trợ code là JCreator version 5.0.

Sau đây các lớp và chức năng chính của từng lớp trong chương trình

1) Point.java
   - Lưu trữ các điểm vào cây KD cho việc tìm kiếm.
   - Tính toán khoảng cách Euclide từ một điểm t.
2) CompareCluster.java
   - Trợ giúp MinHeap (thực hiện bằng hàng đợi ưu tiên) so sánh 2 cụm và lưu trữ phù hợp trong đống (heap)
   - 2 cụm được so sánh dựa vào khoảng cách từ cụm gần nhất của chúng. Cặp cụm có khoảng cách thấp nhất được lưu trữ tại thư mục gốc của MinHeap.
3) Cluster.java
   - Một tập hợp các điểm và đề ra điểm đại diện.
   - Nó cũng lưu khoảng cách từ cụm láng giềng gần nhất của nó.
4) ClusterSet.java
   - Tạo ra một tập hợp các cụm cho một số lượng nhất định các điểm dữ liệu hoặc làm giảm số lượng các cụm với một số cố định của các cụm như quy định sử dụng thuật toán phân cụm phân cấp CURE.
   - ClusterSet sử dụng 2 cấu trúc dữ liệu. Cây KD được khởi tạo và sử dụng để lưu trữ các điểm trên toàn cụm.
   - MinHeap (Sử dụng java.util.PriorityQueue) được sử dụng để lưu trữ các cụm và lặp lại một số phân nhóm. Các MinHeap được sắp xếp lại trong từng bước để đưa cặp gần gũi nhất của cụm vào root của heap và cũng thay đổi các cách đo khoảng cách gần nhất cho tất cả các cụm.

Lớp này chỉ làm việc với các dữ liệu mẫu phân đoạn hoặc đã được thiết lập các khuôn cụm. Tính toán của tập hợp các cụm có thể được thực hiện từ xa trên một máy tính.

5) CURE.java

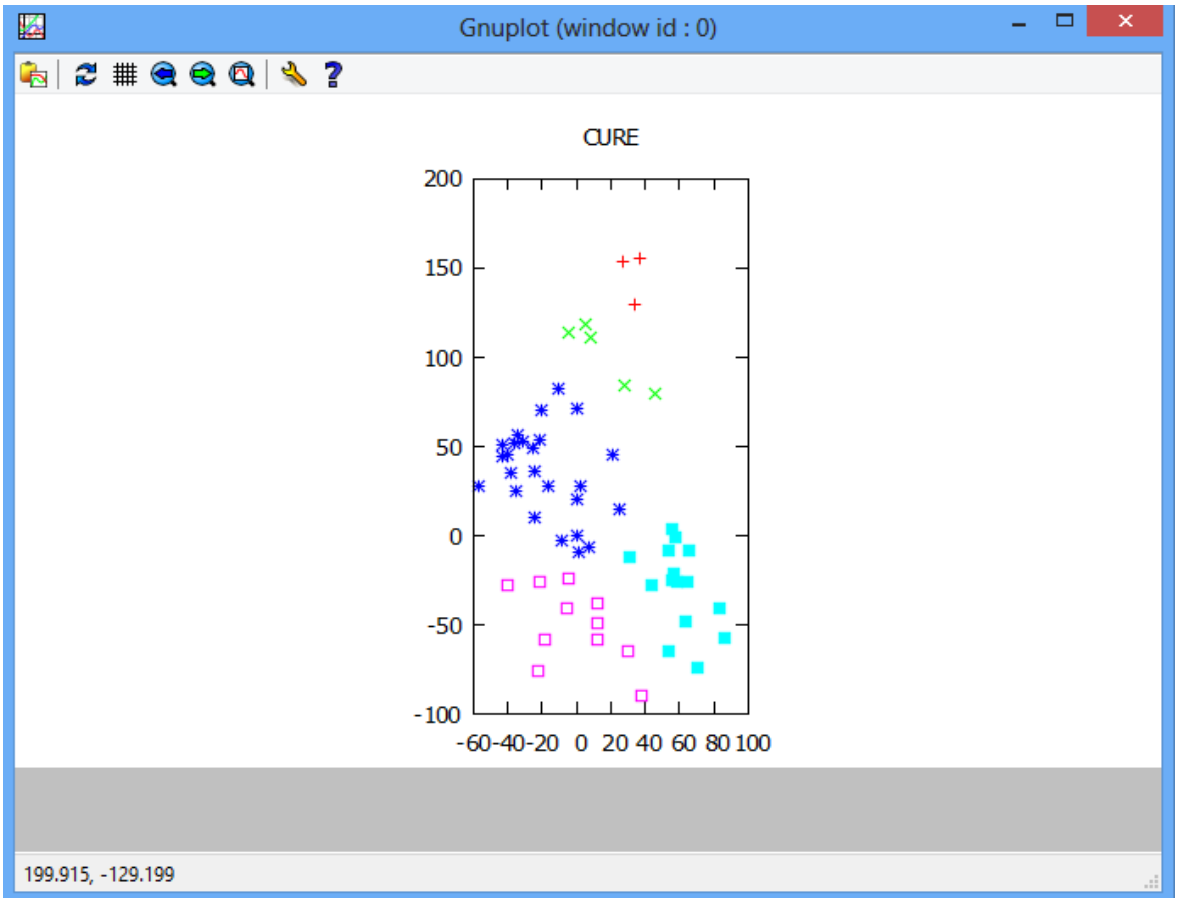
Áp dụng các bước của thuật toán CURE để phận cụm dữ liệu.

KD – Tree Implementation

Trong chương trình có sử dụng cây KD để lưu trữ các điểm trên toàn cụm. Mã nguồn mở KD Tree được sử dụng và phân phối bởi GNU General Public License

3.3. Chạy chương trình

Sau khi chạy chương trình, yêu cầu nhập một file chứa tọa độ các điểm (x,y), chương trình trả về một tập các file trong đó có một file tên là "plotcure.txt" ta mở file này bằng phần mềm gnuplot để xem các cụm đã được phân chia.



Hình 2.21: Kết quả của quá trình phân cụm

KẾT LUẬN

Kết quả đạt được của đồ án

Phân cụm dữ liệu là nhiệm vụ quan trọng trong khai phá dữ liệu, thu hút sự quan tâm của nhiều nhà nghiên cứu. Các kỹ thuật phân cụm đã và đang được ứng dụng thành công trong nhiều lĩnh vực khoa học, đời sống xã hội. Hiện nay, do sự phát triển không ngừng của công nghệ thông tin và truyền thông, các hệ thống CSDL ngày càng đa dạng, và tăng trưởng nhanh cả về chất lẫn về lượng. Hơn nữa, nhu cầu về khai thác các tri thức từ các CSDL này ngày càng lớn. Vì vậy, việc nghiên cứu các mô hình dữ liệu mới, áp dụng các phương pháp khai phá dữ liệu, trong đó có kỹ thuật phân cum dữ liệu là việc làm rất cần thiết có nhiều ý nghĩa.

Trong đồ án này, trước tiên em đã trình bày những hiểu biết của mình về khai phá dữ liệu sau đó là phần nội dung chính của đồ án: Bài toán phân cụm dữ liệu và một số giải thuật theo tiếp cận phân cấp. Ở phần nội dung chính em đã trình bày được thế nào là bài toán phân cụm dữ liệu, các cách tiếp cận, các ứng dụng, các kiểu dữ liệu có thể phân cụm, các độ đo độ tương tự. Đặc biệt, em tập trung đi sâu nghiên cứu về kỹ thuật phân cụm dữ liệu phân cấp và hai thuật toán điển hình của kỹ thuật này là BIRCH và CURE với cách thức tổ chức dữ liệu, thuật toán, đánh giá ưu nhược điểm của mỗi thuật toán.

Tuy nhiên, do năng lực và trình độ có hạn, thời gian thực hiện đồ án lại ngắn, nên trong quá trình thực hiện và trình bày đồ án không tránh khỏi những thiếu sót. Kính mong các thầy cô và các bạn quan tâm giúp đỡ chỉ bảo.

Hướng nghiên cứu phát triển của đề tài trong thời gian tới

Tìm hiểu và thử nghiệm thuật toán với một số ứng dụng thực tế.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Thị Ngọc, *Phân cụm dữ liệu dựa trên mật độ*, Đồ án tốt nghiệp đại học Ngành công nghệ Thông tin – ĐHDL Hải Phòng, 2008.

[2] Trần Thị Quỳnh, *Thuật toán phân cụm dữ liệu nửa giám sát và giải thuật di truyền*, Đồ án tốt nghiệp đại học Ngành công nghệ Thông tin – ĐHDL Hải Phòng, 2008.

[3] Nguyễn Lâm, *Thuật toán phân cụm dữ liệu nửa giám sát*, Đồ án tốt nghiệp đại học Ngành công nghệ Thông tin – ĐHDL Hải Phòng, 2007.

[4] Nguyễn Trung Sơn, *Phương pháp phân cụm và ứng dụng*, Luận văn thạc sĩ khoa học máy tính, Khoa công nghệ thông tin trường Đại học Thái Nguyên.

[5] Nguyễn Thị Hướng, *Phân cụm dữ liệu trong data mining*, Luận văn tốt nghiệp ngành công nghệ thông tin Đại học sư phạm Hà Nội.

[6] Tian Zhang, Raghu Ramakrishnan, Miron Livny. BIRCH: A New Data Clustering Algorithm and Its Applications. Data Mining and Knowledge Discovery, 1, 141–182 (1997), Kluwer Academic Publishers, 1997.

[7] Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, CURE: an efficient clustering algorithm for large databases, Information Systems Vol. 26, No.1, pp.35-58,Elsevier Science, 2001.

[8] J.Han, M. Kamber and A.K.H. Tung, Spatial Clustering Methods in Data Mining, Sciences and Engineering Research Council of Canada.