ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐÀO MỸ HẠNH

CỤM DỮ LIỆU VÀ ỨNG DỤNG TRONG PHÂN TÍCH LƯƠNG CỦA CÁN BỘ TRƯỜNG CAO ĐẲNG NGHỀ HÀ NAM

Chuyên ngành: KHOA HỌC MÁY TÍNH

Mã số chuyên ngành: 60 48 0101

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Thái Nguyên - 2015

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn tập thể các thầy cô trong khoa đào tạo sau đại học trường Đại học Công nghệ Thông tin và Truyền thông Thái Nguyên đã trang bị cho tôi những kiến thức cơ bản trong những năm học tập tại trường để tôi có thể hoàn thành tốt bản luận văn tốt nghiệp này.

Tôi xin cảm ơn các đồng nghiệp và người thân đã động viên, giúp đỡ tôi trong quá trình nghiên cứu và thực hiện luận văn.

Đặc biệt, tôi xin cảm ơn **GS.TS Vũ Đức Thi**, người đã trực tiếp, tận tâm hướng dẫn, giúp đỡ, cung cấp tài liệu và tạo mọi điều kiện thuận lợi cho tôi nghiên cứu thành công luận văn tốt nghiệp của mình.

Thái Nguyên, ngày … tháng … năm 2015

Tác giả luận văn

Đào Mỹ Hạnh

LỜI CAM ĐOAN

Tôi xin cam đoan toàn bộ nội dung bản luận văn này là do tôi tự sưu tầm, tra cứu và sắp xếp cho phù hợp với nội dung yêu cầu của đề tài.

Nội dung luận văn này chưa từng được công bố hay xuất bản dưới bất kỳ hình thức nào và cũng không được sao chép từ bất kỳ một công trình nghiên cứu nào.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác. Tôi cũng xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện luận văn này đã được cảm ơn và các thông tin trích dẫn trong luận văn đã được chỉ rõ nguồn gốc.

Nếu sai tôi xin hoàn toàn chịu trách nhiệm.

Thái Nguyên, ngày … tháng … năm 2015

Người cam đoan

Đào Mỹ Hạnh

DANH MỤC TỪ VIẾT TẮT

CSDL: Cơ sở dữ liệu

KPDL: Khai phá dữ liệu

PCDL: Phân cụm dữ liệu

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH VẼ

MỤC LỤC

MỞ ĐẦU

Khám phá tri thức - Khai phá dữ liệu (Knowledge discovery - Data mining) là một lĩnh vực quan trọng của ngành Công nghệ thông tin, đã và đang thu hút sự quan tâm đông đảo các nhà khoa học trên thế giới và trong nước tham\gia nghiên cứu. Khai phá dữ liệu ra đời vào những năm cuối thập kỷ 80 của thế kỷ XX, nó là lĩnh vực được nghiên cứu nhằm tự động khai thác thông tin, tri thức mới hữu ích, tiềm ẩn từ các CSDL lớn, kho dữ liệu,... Những vấn đề được quan tâm trong khai phá dữ liệu là phân lớp nhận dạng mẫu, luật kết hợp, phân cụm dữ liệu, ... Trong đó, phân cụm dữ liệu (Data Clustering) là một trong những kỹ thuật khai thác dữ liệu có hiệu quả. Phân cụm dữ liệu là quá trình tìm kiếm và phát hiện ra các cụm hoặc các mẫu dữ liệu tự nhiên trong cơ sở dữ liệu lớn. Phân cụm dữ liệu đã được ứng dụng trong nhiều lĩnh vực khác nhau như giáo dục, y tế, kinh tế, bảo hiểm, phân đoạn ảnh, ...

Việc áp dụng phân cụm dữ liệu để phân tích trong ngành kế toán hiện nay là rất cần thiết, bởi lượng dữ liệu lưu trữ lương khá lớn, việc phân tích đánh giá lương để đưa ra các chiến lược cân đối nguồn chi phí của đơn vị, dự báo quỹ lương và có kế hoạch cân đối tài chính cho phù hợp cũng gặp nhiều khó khăn. Ngoài ra việc phân tích lương còn phục vụ công tác quản lý nhân sự, giúp nắm được tình hình sử dụng con người của đơn vị từ đó đưa ra các chính sách tuyển dụng phù hợp, có các giải pháp tạo động lực cho người lao động bằng các chính sách tài chính.

Việc phân cụm dữ liệu để phân tích lương cho kết quả thu được sẽ phân loại theo giá trị lương của mỗi cán bộ, phân loại ra các mức thu nhập cao thấp khác nhau từ đó đưa ra các chính sách cân đối thu chi để có những chính sách ưu đãi phù hợp mà vẫn đảm bảo tài chính của đơn vị.

Với các lý do như vậy tôi chọn đề tài: **"Một số phương pháp phân cụm dữ liệu và ứng dụng trong phân tích lương của cán bộ trường Cao đẳng Nghề Hà Nam"** làm đề tài luận văn tốt nghiệp. Bố cục luận văn gồm có 3 chương:

Chương I: Tổng quan về khai phá dữ liệu và phân cụm dữ liệu.

Chương II: Một số thuật toán phân cụm dữ liệu điển hình

Chương III: Ứng dụng phương pháp phân nhóm dữ liệu vào phân tích lương của cán bộ trường Cao đẳng Nghề Hà Nam.

CHƯƠNG I: TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU VÀ PHÂN CỤM DỮ LIỆU

1.1 Khai phá dữ liệu

1.1.1 Giới thiệu về khai phá dữ liệu

Khai phá dữ liệu (Data Mining) là một khái niệm ra đời vào những năm cuối thập kỉ 80 của thế kỉ XX. Khai phá dữ liệu là một lĩnh vực được nghiên cứu nhằm tự động khai thác thông tin, tri thức mới hữu ích, tiềm ẩn từ các CSDL lớn, kho dữ liệu,... Ngoài thuật ngữ khai phá dữ liệu người ta còn một số thuật ngữ khác có ý nghĩ tương tự như: trích chọn dữ liệu (Knowledge extraction), nạo vét dữ liệu (Data dredging), phân tích dữ liệu mẫu (Pattern Analisys), phát hiện tri thức từ CSDL (Knowlegde Discovery in Databases. Các bước cơ bản trong quá trình phát hiện tri thức từ CSDL là [6]:

(1) Làm sạch dữ liệu: Loại bỏ dữ liệu nhiễu và không đồng nhất

(2) Tích hợp dữ liệu: Các nguồn dữ liệu khác nhau được tích hợp với nhau

(3) Trích chọn dữ liệu: Chọn các dữ liệu liên quan đến phân tích

(4) Chuyển đổi dữ liệu: Chuyển dữ liệu sang phù hợp để khai phá

(5) Khai phá dữ liệu: Bước thiết yếu để tìm ra mẫu dữ liệu

(6) Đánh giá các mẫu: Kiểm định dựa vào mục tiêu ban đầu của chúng

(7) Biểu diễn tri thức: Hiển thị, biểu diễn kết quả sao có thể hiểu được

Trong 7 giai đoạn của quá trình khám phá tri thức thì giai đoạn 5 (Khai phá dữ liệu) là giai đoạn quan trọng nhất.

Trong những năm gần đây, rất nhiều các phương pháp và thuật toán mới về KPDL liên tục được công bố. Điều này chứng tỏ những ưu thế, lợi ích và khả năng ứng dụng thực tế to lớn của KPDL.

1.1.2 Quá trình khai phá dữ liệu

Về bản chất khai phá dữ liệu là giai đoạn t tìm ra được những thông tin mới, tiềm ẩn trong CSDL và chủ yếu phục vụ cho quá trình mô tả và dự đoán.

Mô tả dữ liệu là tổng kết hoặc diễn tả những tính chất hoặc đặc tính chung của những thuộc tính dữ liệu trong kho dữ liệu mà con người có thể hiểu được.

Dự đoán là dựa trên những dữ liệu hiện thời để dự đoán những quy luật được phát hiện từ các mối liên hệ giữa các thuộc tính của dữ liệu trên cơ sở đó chiết xuất ra các mẫu, dự đoán được những giá trị chưa biết hoặc những giá trị tương lai của các biến quan tâm.

Quá trình khai phá dữ liệu gồm các bước chính như sau:

- Xác định nhiệm vụ: Xác định các vấn đề chính cần giải quyết.

- Xác định dữ liệu liên quan: Dùng để xây dựng giải pháp.

- Thu thập và tiền xử lý dữ liệu: Thu thập các dữ liệu liên quan và tiền xử lý chúng sao cho thuật toán khai phá dữ liệu có thể hiểu được.

- Giải thuật khai phá dữ liệu: Lựa chọn thuật toán khai phá dữ liệu và thực hiện việc khai phá dữ liệu để tìm được các mẫu có ý nghĩa.

1.1.3 Các kỹ thuật khai phá dữ liệu

- Khai phá dữ liệu thường sử dụng các phương pháp sau:

+ Luật kết hợp (AssoCi ation rules): Là phát hiện và đưa ra mối liên hệ giữa các giá trị dữ liệu trong CSDL.

+ Phân cụm dữ liệu (Data Clustering): Sắp xếp các đối tượng theo từng cụm dữ liệu tự nhiên, tức là số lượng và tên cụm chưa được biết trước. Các đối tượng được gom cụm sao cho độ tương đồng (similar) giữa các đối tượng trong cùng một cụm là lớn nhất và mức độ tương đồng giữa các đối tượng nằm trong các cụm khác nhau là nhỏ nhất. Phân cụm còn được gọi là học không giám sát (Unsupervised Learning).

- Khai phá dữ liệu dự đoán thường sử dụng các phương pháp sau:

+ Phân lớp (Classfication): Là quá trình xếp một đối tượng vào một trong những lớp đã biết trước (Ví dụ: phân lớp các học sinh theo kết quả thi). Phân lớp còn được gọi là học có giám sát (Supervised learning).

+ Hồi quy (Regression): Phương pháp hồi quy tương tự như phân lớp dữ liệu nhưng khác ở chỗ nó dùng để dự đoán các giá trị liên tục còn phân lớp dữ liệu dùng để dự đoán các giá trị rời rạc

- Ngoài các phương pháp trên còn rất nhiều các phương pháp khác như:

+ Cây quyết định (DeCi sion Trees)

+ Mạng nơ-ron (Neural Network)

+ Trực quan hóa (Visualization)

+ Biểu diễn mô hình (Model Evaluation)

+ Phương pháp tìm kiếm (Search Method)

+ Phân tích theo trình tự thời gian (Time series Analysis)

1.1.4 Ứng dụng của Khai phá dữ liệu

Khai phá dữ liệu được ứng dụng trong nhiều lĩnh vực khác nhau nhằm khai thác nguồn dữ liệu được lưu trữ trong các hệ thống thông tin. Một số ứng dụng điển hình trong khai phá dữ liệu có thể liệt kê như sau:

- Thương mại: Như phân tích dữ liệu bán hàng và thi trường, phân tích đầu tư, phát hiện gian lận, chứng thực hóa khách hàng, dự báo xu hướng phát triển,...

- Thông tin khoa học: Quan sát thiên văn, dự báo thời tiết, dữ liệu gene, tìm kiếm so sánh các hệ gene và thông tin di truyền (sinh học),...

- Mạng viễn thông: Phân tích các cuộc gọi điện thoại và hệ thống giám sát lỗi, sự cố, chất lượng dịch vụ,...

- Phân tích dữ liệu và hỗ trợ ra quyết định, điều trị y học, khai phá Web, tài chính và thị trường chứng khoán, bảo hiểm, giáo dục, du lịch,...

1.1.5 Các xu thế và vấn đề cần giải quyết trong khai phá dữ liệu

Một số hướng nghiên cứu chính của Khai phá dữ liệu hiện nay [6]:

Xu hướng khai phá dữ liệu đang nỗ lực hơn nữa đối với việc thăm dò các lĩnh vực ứng dụng mới, cải tiến phương pháp mở rộng, tương tác, tích hợp khai thác dữ liệu với dịch vụ web, cơ sở dữ liệu, kho dữ liệu, các hệ thống điện toán đám mây và khai thác mạng xã hội,... Các xu hướng khác bao gồm việc khai thác dữ liệu thời gian và không gian, dữ liệu sinh học, hệ thống dữ liệu kĩ thuật, các dữ liệu đa phương tiện và khai phá dữ liệu văn bản, khai pha web, các dữ

liệu phân tán, dữ liệu thời gian thực, dòng dữ liệu, khai thác dữ liệu hình ảnh, âm thanh và vấn đề an ninh trong khai thác dữ liệu. Việc khám phá được nhiều tri thức khác nhau từ các kiểu dữ liệu khác nhau, tính chính xác và hiệu quả, khả năng mở rộng và tích hợp, xử lý nhiễu và tính hữu ích của dữ liệu được khai phá.

Khai phá dữ liệu liên quan đến nhiều ngành, nhiều lĩnh vực trong thực tế, vì vậy các thách thức và khó khăn ngày càng nhiều, càng lớn hơn. Sau đây là một số các thách thức và khó khăn cần được quan tâm:

- Các cơ sở dữ liệu lớn với hàng trăm trường, hàng triệu bản ghi và kích thước lên tới nhiều Gi-ga byte (GB) hoặc nhiều Tê-ra byte (TB).

- Số lượng các trường lớn (các thuộc tính, các biến) làm cho số chiều của bài toán trở nên cao. Đặc biệt lưu ý đến dữ liệu không gian, số chiều cao có thể rất thưa và bị lệch nhiều.

- Việc dữ liệu thay đổi nhanh có thể làm cho các mẫu phát hiện trước đó không hợp lệ. Thêm vào đó các biến đã đo trong một cơ sở dữ liệu ứng dụng cho trước có thể bị sửa đổi, xóa bỏ hay tăng thêm các phép đo mới.

- Dữ liệu bị thiếu và bị nhiễu.

- Mối quan hệ phức tạp giữa các trường (dữ liệu hỗn hợp).

- Tính dễ hiểu của các mẫu.

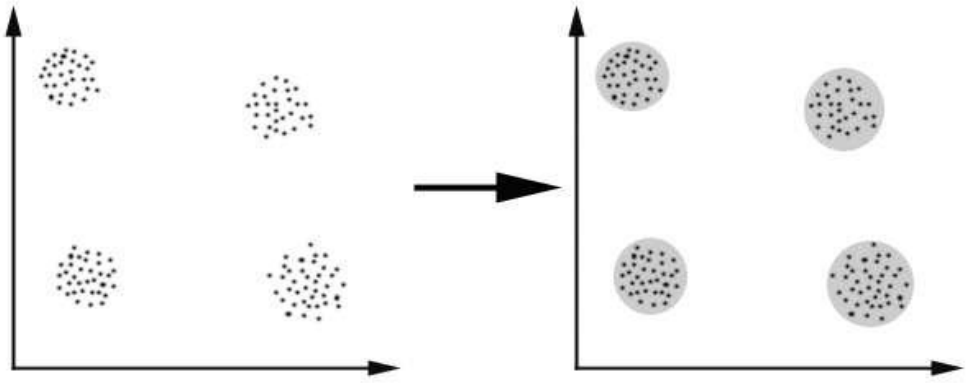
- Tích hợp với các hệ thống khác.

1.2 Kỹ thuật phân cụm trong Khai phá dữ liệu

1.2.1 Tổng quan về kỹ thuật phân cụm

Phân cụm dữ liệu là quá trình nhóm một tập các đối tượng tương tự nhau trong tập dữ liệu vào các cụm sao cho các đối tượng thuộc cùng một cụm là tương đồng, còn các đối tượng thuộc các đối tượng khác nhau sẽ không tương đồng.

Mục đích chính của khai phá dữ liệu là nhằm khám phá cấu trúc của mẫu dữ liệu để thành lập các nhóm dữ liệu từ tập dữ liệu lớn, theo đó nó cho phép người ta đi sâu vào phân tích và nghiên cứu cho từng cụm dữ liệu này nhằm khám phá và tìm kiếm các thông tin tiềm ẩn, hữu ích phục vụ cho việc ra quyết định. Phân cụm dữ liệu được sử dụng rộng rãi trong nhiều lĩnh vực trên thực tế như: nhận dạng ảnh, nghiên cứu thị trường, phân cụm gen trong sinh học ...



Hình 1.1: Phân cụm dữ liệu

Phân cụm dữ liệu là một kỹ thuật trong Khai phá dữ liệu nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tự nhiên tiềm ẩn và quan trọng trong tập dữ liệu lớn để từ đó cung cấp thông tin, tri thức cho việc ra quyết định.

Phân cụm dữ liệu còn có thể được sử dụng như một bước tiền xử lý cho các thuật khai phá dữ liệu khác như là phân loại và mô tả đặc điểm, có tác dụng trong việc phát hiện ra các cụm.

Trong học máy, phân cụm dữ liệu được xem là vấn đề học không có giám sát, vì nó phải giải quyết vấn đề tìm một cấu trúc trong tập hợp dữ liệu chưa biết trước các thông tin về lớp hay các thông tin về tập huấn luyện.

Một vấn đề thường gặp trong phân cụm dữ liệu là hầu hết các dữ liệu cần cho phân cụm đều có chứa dữ liệu "nhiễu" do quá trình thu thập thiếu chính xác hoặc thiếu đầy đủ, vì vậy cần phải xây dựng các 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 chính xác hoặc các đối tượng dữ liệu khuyết thiếu 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.

Tóm lại, phân cụm dữ liệu là một vấn đề khó vì người ta phải giải quyết các vấn đề con cơ bản sau:

- Biểu diễn dữ liệu.

- 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.

Theo các nghiên cứu thì đến nay chưa có một phương pháp phân cụm tổng quát nào có thể giải quyết trọn vẹn cho tất cả các dạng cấu trúc cụm dữ liệu.

Hơn nữa, các phương pháp phân cụm cần có cách thức biểu diễn cấu trúc các cụm dữ liệu khác nhau, với mỗi cách thức biểu diễn khác nhau sẽ có một thuật toán phân cụm phù hợp. Phân cụm dữ liệu đang là một vấn đề mở và khó vì người ta cần phải đi giải quyết nhiều vấn đề cơ bản như đã đề cập ở trên một cách trọn vẹn và phù hợp với nhiều dạng dữ liệu khác nhau. Đặc biệt đối với dữ liệu hỗn hợp, đang ngày càng tăng trưởng không ngừng trong các hệ quản trị dữ liệu, đây cũng là một trong những thách thức lớn trong lĩnh vực khai phá dữ liệu.

1.2.2 Một số khái niệm cần thiết khi tiếp cận phân cụm dữ liệu

1.2.2.1 Các kiểu dữ liệu và thuộc tính trong phép phân cụm

Các cấu trúc dữ liệu thường sử dụng trong các thuật toán phân cụm là:

+ Ma trận dữ liệu: gồm n hàng, p cột. Trong đó n là số đối tượng, p là số thuộc tính của mỗi đối tượng.

+ Ma trận phi tương tự: gồm n hàng, m cột. Phần tử d(i,j) chứa khoảng cách hay độ khác biệt giữa đối tượng i và j. Phần lớn các thuật toán phân cụm sử dụng cấu trúc ma trận phi tương tự.

Trong khai phá dữ liệu nói chung và phân cụm dữ liệu nói riêng ta thường xử lý các kiểu dữ liệu:

- Dữ liệu xác thực (Categorical Data)
- Dữ liệu văn bản (Text Data)
- Dữ liệu chuỗi thời gian (Time-Series Data)
- Dữ liệu mạng (Network Data)
- Dữ liệu liên kết (Linked Data)

- Dữ liệu đa phương tiện (Multimedia Data)

- Dữ liệu không gian (Space Data)

Dựa trên kích thước miền có các loại thuộc tính như sau:

- Thuộc tính liên tục (Continuous Attributes): màu sắc, nhiệt độ, ...

- Thuộc tính rời rạc (Discrete Attributes): điểm số, số quyển sách, ...

Dựa trên phép đo có các loại thuộc tính như sau:

- Thuộc tính định danh (Nominal Attributes)

- Thuộc tính có thứ tự (Ordinal Attributes)

- Thuộc tính khoảng (Interval Attributes)

- Thuộc tính tỉ lệ (Ratio Attributes)

- Thuộc tính nhị phân (Binary Attributes)

- Thuộc tính số (Numeric Attributes)

Sự hiểu biết về quy mô, sự liên quan của các loại dữ liệu, các thuộc tính rất hữu ích trong việc giải thích các kết quả của thuật toán phân cụm dữ liệu.

1.2.2.2 Đo độ tương đồng

Để đánh giá chất lượng phân cụm người ta tìm cách thích hợp để xác định "khoảng cách" giữa các đối tượng (phép đo độ tương tự dữ liệu). Đây là các hàm để đo sự giống nhau giữa các cặp đối tượng dữ liệu, giá trị của hàm tính độ đo tương tự càng lớn thì sự giống nhau giữa các đối tượng càng lớn và ngược lại.

Một số phép đo độ tương tự áp dụng đối với các kiểu dữ liệu khác nhau:

+ Thuộc tính khoảng:

Khoảng cách Minskowski: d(x,y) = $(\sum_{i=1}^{n}|x_i - y_i|^{q})^{1/q}$, với q là số nguyên dương.

Khoảng cách Euclide: d(x,y) = $\sqrt{(\sum_{i=1}^{n}(x_i - y_i)^2}$, (trường hợp đặc biệt của khoảng cách Minskowski trong trường hợp q = 2).

Khoảng cách Manhattan: d(x,y) = $\sum_{i=1}^{n}|x_i - y_i|^{q}$, (trường hợp đặc biệt của khoảng cách Minskowski trong trường hợp q = 1).

Khoảng cách cực đại: d(x,y) = $Max_{i=1}^{n}|x_i - y_i|$, đây là trường hợp của khoảng các Minskowski trong trường hợp q$\rightarrow \infty$.

+ Thuộc tính nhị phân:

		Đối tượng y		Tổng
		1	0	
Đối tượng x	1	a	b	a + b
	0	c	d	c + d
	Tổng	a +c	b + d	p = a + b + c + d

Bảng 1.1: Thuộc tính dữ liệu nhị phân

Các phép đo độ tương tự đối với dữ liệu thuộc tính nhị phân được định nghĩa như sau:

- Hệ số ghép đơn giản: d(x,y) = $\frac{a+d}{p}$

- Hệ số Jacard: d(x,y) = $\frac{a}{a+b+c}$

+ Thuộc tính định danh: Độ đo phi tương tự giữa hai đối tượng x và y được định nghĩa như sau: d(x,y) = $\frac{p-m}{p}$, trong đó m là số cặp trùng nhau và p là tổng số các thuộc tính.

+ Thuộc tính có thứ tự: Giả sử i là thuộc tính thứ tự có M_i giá trị (M_i kích thước miền giá trị): Các trạng thái M_i được sắp thứ tự như sau: [1...M_i], 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 thứ tự có các miền giá trị khác nhau, vì vậy ta chuyển đổi chúng về cùng miền giá trị [0,1] bằng cách thực hiện phép biến đổi sau cho mỗi thuộc tính: $z_i^{(j)} = \frac{r_i^{(j)} - 1}{M_i - 1}$, với i = 1, ..., M_i.

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.

+ Thuộc tính tỷ 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. Phép biến đổi logarit này thích hợp trong trường hợp các giá trị của thuộc tính là số mũ.

Trong thực tế, khi tihns 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 đánh trọng số 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 hóa 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 tự dữ liệu được xác định như sau: $d(x,y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}$.

Người ta có thể chuyển đổi giữa các mô hình cho các kiểu dữ liệu trên. Tùy từng trường hợp dữ liệu cụ thể mà người ta sử dụng các mô hình 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 và góp phần xây dựng thuật toán phân cụm dữ liệu có hiệu quả cao trong việc đảm bảo chất lượng cũng như chi phí tính toán của thuật toán.

1.2.3 Các yêu cầu đối với kĩ thuật phân cụm dữ liệu

Hầu hết các nghiên cứu và phát triển các thuật toán phân cụm dữ liệu nói chung đều nhằm thỏa mãn các yêu cầu cơ bản sau:

- Có khả năng mở rộng, gia tăng: Một đặc trưng rất đáng quan tâm trong các lĩnh vực như web đó là khả năng cập nhật phân nhóm có tính tăng. Những tài liệu mới cần phải được đưa vào các phân nhóm tương ứng mà không phải phân nhóm lại toàn bộ tập tài liệu.

- Khả năng thích nghi với các kiểu và thuộc tính dữ liệu khác nhau: Có nhiều thuật toán phân nhóm, có những thuật toán phù hợp với dữ liệu số, có những thuật toán khi áp dụng cho loại dữ liệu nhị phân hay dữ liệu ảnh,…
- Nhận biết được các nhóm với hình thù bất kỳ: Một nhóm có thể có hình dạng bất kỳ vì vậy mà việc phát triển thuật toán có khả năng xác định các nhóm với hình thù bất kỳ là quan trọng và cần thiết.
- Tối thiểu miền tri thức cho xác định các tham số đầu vào: Miền tri thức đầu vào cần thiết cho một thuật toán phân nhóm càng ít, chi phí cho việc phân nhóm càng giảm và nó càng khả thi hơn.
- Thích nghi với dữ liệu đa chiều: Dữ liệu thông thường thường có số chiều ít, từ hai đến ba chiều mà một số thuật toán phân nhóm đưa ra kết quả rất tốt. Bên cạnh đó, dữ liệu đa chiều (nhiều hơn ba chiều) cũng rất đa dạng và cần thiết được phân nhóm cho nhiều ứng dụng thực tế. Với loại dữ liệu này, việc phân loại dựa vào kiến thức con người tỏ ra có hiệu quả, tuy nhiên với khối lượng dữ liệu lớn như vậy, việc sử dụng kiến thức chuyên gia là tốn kém nên chúng tôi cần tìm các thuật toán phân nhóm để giải quyết được vấn đề này.
- Phân nhóm trên một số ràng buộc: Trong một số ứng dụng, chúng tôi cần phân nhóm trên cơ sở dữ liệu chứa các liên kết bắt buộc giữa hai hay nhiều đối tượng. Việc phân nhóm cần đảm bảo các đối tượng này thỏa mãn các ràng buộc đó.
- Khả năng khử nhiễu: Một vấn đề có thể xảy ra với nhiều thuật toán phân nhóm đó là sự xuất hiện của nhiễu và các dữ liệu thừa. Một thuật toán phân nhóm tốt phải có khả năng giải quyết những kiểu nhiễu này và đưa ra các phân nhóm có chất lượng cao và không bị ảnh hưởng bởi nhiễu. Trong phân nhóm có thứ bậc, ví dụ các tính toán khoảng cách láng giềng gần nhất và láng giềng xa nhất, rất nhạy cảm với các dữ liệu thừa do đó không nên được sử dụng nếu có thể. Phương thức trung bình kết nối là thích hợp nhất với dữ liệu bị nhiễu.
- Hiệu suất: Trong lĩnh vực web, mỗi một câu lệnh tìm kiếm có thể trả về hàng trăm và thỉnh thoảng là hàng nghìn trang web. Việc phân nhóm các kết quả này trong một thời gian chấp nhận được là rất cần thiết. Cần phải chú ý rằng một

vài hệ thống chỉ phân nhóm trên các đoạn tin được trả lại trên hầu hết các máy tìm kiếm chứ không phải toàn bộ trang web. Đây là một chiến thuật hợp lý trong việc phân nhóm kết quả tìm kiếm nhanh nhưng nó không chấp nhận được với phân nhóm tài liệu vì các đoạn tin không cung cấp đầy đủ thông tin về nội dung thực sự của những tài liệu này. Một thuật toán phân nhóm online nên có khả năng hoàn thành trong thời gian tuyến tính nếu có thể.

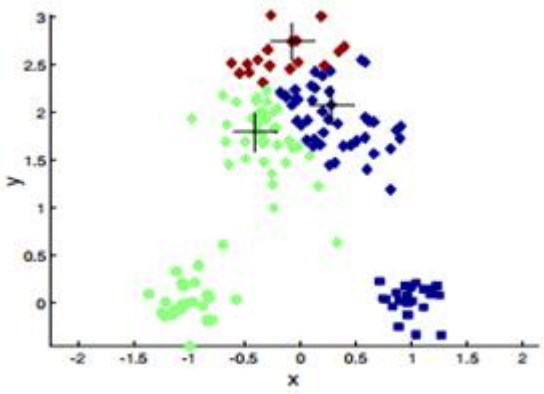
- Việc biểu diễn kết quả: Một thuật toán phân nhóm là tốt nếu nó có khả năng biểu diễn một sự mô tả của các phân nhóm mà nó đưa ra ngắn gọn và chính xác với người sử dụng. Các tổng kết của phân nhóm nên có đủ tiêu biểu về nội dung tương ứng để người sử dụng có thể đưa ra quyết định nhanh xem phân nhóm nào mà họ cảm thấy quan tâm.

1.2.4 Các hướng tiếp cận trong phân cụm dữ liệu

Các kĩ thuật phân cụm có rất nhiều cách tiếp cận và các ứng dụng trong thực tế, nó đều hướng tới hai mục tiêu chung đó là chất lượng của các cụm khám phá được và tốc độ thực hiện của thuật toán. Hiện nay các kĩ thuật phân cụm có thể phân loại theo các phương pháp tiếp cận chính như sau [6]:

1.2.4.1 Phương pháp phân hoạch:

Phân cụm phân hoạch (Partitioning Methods) chia một tập hợp dữ liệu có n phần tử thành k nhóm cho đến khi xác định số các cụm được thiết lập. Số các cụm được thiết lập là các đặc trưng được lựa chọn trước. Phương pháp này là tốt cho việc tìm các cụm hình cầu trong không gian Euclidean.



Hình 1.2: Ví dụ minh họa phân cụm phân hoạch

Phương pháp này cũng phụ thuộc vào khoảng cách cơ bản giữa các điểm để lựa chọn các điểm dữ liệu nào có quan hệ là gần nhau với mỗi điểm khác và các điểm dữ liệu nào không có quan hệ hoặc có quan hệ là xa nhau so với mỗi điểm khác. Tuy nhiên, phương pháp này không thể xử lý các cụm có hình dạng kì quặc hoặc các cụm có mật độ các điểm dày đặc. Các thuật toán phân hoạch dữ liệu có độ phức tạp rất lớn khi xác định độ tối ưu toàn cục cho vấn đề phân cụm dữ liệu, do nó phải tìm kiếm tất cả các cách phân hoạch có thể được. Chính vì vậy, trên thực tế thường đi tìm giải pháp tối ưu cục bộ cho vấn đề này bằng cách sử dụng một hàm tiêu chuẩn để đánh giá chất lượng của cụm cũng như để hướng dẫn cho quá trình tìm kiếm phân hoạch dữ liệu. Như vậy, ý tưởng chính của thuật toán phân cụm phân hoạch tối ưu cục bộ là sử dụng chiến lược tham lam (Greedy) để tìm kiếm nghiệm.

1.2.4.2 Phương pháp phân cụm phân cấp

Phân cụm phân cấp (Hierarchical Methods) xây dựng một phân cấp trên cơ sở các đối tượng dữ liệu đang xem xét. Nghĩa là sắp xếp một tập dữ liệu đã cho thành một cấu trúc có dạng hình cây, cây phân cấp này được xây dựng theo kĩ thuật đệ quy.

Có hai cách tiếp cận phổ biến của kĩ thuật này đó là:

- Phân cấp tích tụ (Bottom-Up): Từ dưới lên, mỗi đối tượng là một nhóm.
- Phân cụm chia nhỏ (Top-Down): Từ trên xuống, tất cả các đối tượng là một nhóm.

Ưu điểm của phương pháp này là có thể làm việc tốt với các tập dữ liệu lớn.

Hạn chế: khó xác định phương pháp tích tụ hay chia nhỏ; nhạy cảm với các dữ liệu nhiễu và cá biệt; thường gặp khó khăn với các cụm có hình dạng lồi.

Thực tế áp dụng, có nhiều trường hợp kết hợp cả hai phương pháp phân cụm phân hoạch và phân cụm phân cấp, nghĩa là kết quả thu được của phương pháp phân cấp có thể cải tiến thông qua bước phân cụm phân hoạch.

1.2.4.3 Phương pháp phân cụm dựa trên mật độ

Phân cụm dựa trên mật độ (Density-Based Methods) nhóm các đối tượng dữ liệu dựa trên hàm mật độ xác định, mật độ là số các đối tượng lân cận của một đối tượng dữ liệu theo một nghĩa nào đó. Trong cách tiếp cận này, khi một 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 này phải lớn hơn một ngưỡng đã được định trước. Phương pháp phân cụm dựa trên 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 kì.

Phân cụm dựa trên mật độ có thể khắc phục được các phần tử ngoại lai hoặc giá trị nhiễu rất tốt, tuy nhiên việc xác định các tham số mật độ của thuật toán là rất khó khăn, trong khi các tham số này lại có tác động rất lớn đến kết quả phân cụm.

1.2.4.4 Phương pháp phân cụm dựa trên lưới

Phân cụm dựa trên lưới (Grid-Based Methods) thích hợp với dữ liệu nhiều chiều, dựa trên cấu trúc dữ liệu lưới để phân cụm, phương pháp này chủ yếu tập trung áp dụng cho lớp dữ liệu không gian. Mục tiêu của phương pháp này là lượng hóa dữ liệu thành các ô tạo thành cấu trúc dữ liệu lưới. Sau đó các thao tác phân cụm chỉ cần làm việc với các đối tượng trong từng ô trên lưới chứ không phải các đối tượng dữ liệu. Cách tiếp cận dựa trên lưới này không di chuyển các đối tượng trong các ô mà xây dựng nhiều mức phân cấp của nhóm các đối tượng trong một ô.

Phương pháp này gần giống với phương pháp phân cụm phân cấp nhưng chúng không trộn các ô, đồng thời giải quyết khắc phục yêu cầu dữ liệu nhiều chiều mà phương pháp phân cụm dựa trên mật độ không giải quyết được. Ưu điểm của phương pháp phân cụm 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 ô trong mỗi chiều của không gian lưới.

Một số phương pháp khác:

Phân cụm mờ: Sử dụng kỹ thuật mờ để phân cụm dữ liệu. Các thuật toán thuộc loại này chỉ ra lược đồ phân cụm thích hợp với tất cả các hoạt động đời sống hàng ngày, chúng chỉ xử lý dữ liệu thực không chắc chắn.

Phân cụm sử dụng mạng Kohonen: Loại phân cụm này dựa trên khái niệm của các mạng nơ-ron. Mạng Kohonen 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 tương ứng ra.

CHƯƠNG II:

MỘT SỐ THUẬT TOÁN PHÂN CỤM DỮ LIỆU ĐIỂN HÌNH

2.1 Thuật toán K-Means

Thuật toán phân nhóm K-Means do MacQueen đề xuất trong lĩnh vực thống kê năm 1967, mục đích của thuật toán k-means là sinh ra k cụm dữ liệu $\{C_1, C_2, \ldots, C_k\}$ từ một tập dữ liệu ban đầu gồm n đối tượng trong không gian d chiều $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ (i = 1, …, n), sao cho hàm tiêu chuẩn:

$\sum_{i=1}^{k} \sum_{x \in C_I} D^2(x - m_i)$ đạt giá trị tối thiểu.

Trong đó: m_i là trọng tâm của cụm C_i, D là khoảng cách giữa hai đối tượng.

Trọng tâm của một cụm là một véc-tơ, trong đó giá trị của mỗi phần tử của nó là trung bình cộng các thành phần tương ứng của các đối tượng véc-tơ dữ liệu trong cụm đang xét. Tham số đầu vào của thuật toán là số cụm k, tập CSDL gồm n phần tử và tham số đầu ra của thuật toán là các trọng tâm của các cụm dữ liệu. Độ đo khoảng cách giữa các đối tượng dữ liệu thường được sử dụng dụng là khoảng cách Euclide. Hàm tiêu chuẩn và độ đo khoảng cách có thể được xác định cụ thể hơn tuỳ vào ứng dụng hoặc các quan điểm của người dùng.

Thuật toán K-Means bao gồm các bước cơ bản sau:

INPUT: - Tập các đối tượng dữ liệu $X = \{x_1, x_2, \ldots, x_n\}$

- Số lượng nhóm: k

OUTPUT: Các nhóm C_i (i = 1, .., k) sao cho hàm tiêu chuẩn E đạt giá trị tối thiểu.

Bước 1: Khởi tạo

Chọn k đối tượng m_j (j = 1, ..., k) là trọng tâm ban đầu của k nhóm từ tập dữ liệu (việc lựa chọn này có thể là ngẫu nhiên hoặc theo kinh nghiệm).

Bước 2: Tính toán khoảng cách và gán nhóm

Đối với mỗi đối tượng Xi ($1 \le i \le n$) , tính toán khoảng cách từ nó tới mỗi trọng tâm mj với j = 1, …, k, sau đó tìm trọng tâm gần nhất đối với mỗi đối tượng.

Bước 3: Cập nhật lại trọng tâm

Đối với mỗi j = 1,.., k, cập nhật trọng tâm cụm mj bằng cách xác định trung bình cộng của các véc-tơ đối tượng dữ liệu.

Bước 4: Điều kiện dừng

Lặp các bước 2 và 3 cho đến khi các trọng tâm của cụm không thay đổi.

Thuật toán K-Means phân tích quá trình phân cụm đơn giản nên có thể áp dụng đối với tập dữ liệu lớn. Tuy nhiên nhược điểm của thuật toán này là chỉ áp dụng với dữ liệu có thuộc tính số và khám phá ra các cụm có dạng hình cầu, K-Means còn rất nhạy cảm với nhiễu và các phần tử ngoại lai trong dữ liệu. Chất lượng của thuật toán K-means phụ thuộc nhiều vào các tham số đầu vào như: số cụm k và k trọng tâm khởi tạo ban đầu. Trong trường hợp các trọng tâm khởi tạo ban đầu mà quá lệch so với các trọng tâm cụm tự nhiên thì kết quả phân cụm của K-Means là rất thấp, nghĩa là các cụm dữ liệu được khám phá rất lệch so với các cụm trong thực tế. Việc chọn k đối tượng làm trọng tâm có thể là ngẫu nhiên hoặc theo kinh nghiệm.

Phương pháp xác định k đối tượng làm trọng tâm: Ở đây, tôi đề cập đến thuật toán phân nhóm nữa giám sát dựa trên tập dữ liệu giống đã gán nhãn để sinh ra các ràng buộc nhằm dẫn dắt quá trình phân cụm cũng như việc xác định các đối tượng làm trọng tâm.

Cho tập dữ liệu ban đầu $X = \{x_1, x_2, ..., x_n\}$, gọi $S \subseteq X$ là tập giống (tập dữ liệu đã gán nhãn) trong đó với mỗi đối tượng $x_i \in S$ chúng tôi cung cấp cho nó một phân hoạch (nhóm) C_j. Giả sử rằng bất kể nhóm nào trong C cũng có ít nhất một đối tượng xi thuộc tập giống. Tiến hành phân hoạch tập giống S thành k cụm giống tách rời $\{S_1, S_2, ..., S_k\}$ do đó mọi đối tượng $x_i \in S_h$ đều nằm trong cụm C_j tương ứng. Nhiệm vụ cần giải quyết là từ k cụm giống $S = (S_1, S_2, ..., S_k)$ chúng ta phải phân hoạch tập dữ liệu X thành k phân hoạch tách rời $C = (C_1, C_2, ..., C_k)$.

Để mô tả rõ hơn cho vấn đề này, chúng tôi cài đặt thuật toán Seeded-Kmeans sử dụng cụm giống S để khởi tạo cho thuật toán K-Means. Do vậy thay vì phải khởi tạo k đối tượng trọng tâm ngẫu nhiên từ tập tài liệu thì khởi tạo k cụm

hạt giống. Với tập giống là những phần tử đã được gán nhãn do người dùng thực hiện phân cụm xác định từ tiêu chuẩn cụ thể đối với từng mục đích phân nhóm.

Thuật toán Seeded-Kmeans bao gồm các bước cơ bản sau:

INPUT: - Tập các đối tượng dữ liệu X = X = $\{x_1, x_2, ..., x_n\}$

- Số lượng nhóm: k

- Tập giống S = $\{S_1, S_2, \ldots, S_k\}$

OUTPUT: Các nhóm C_i (i = 1, .., k) sao cho hàm tiêu chuẩn E đạt giá trị tối thiểu.

Bước 1: Khởi tạo

Chọn k đối tượng m_j (j = 1, ..., k) là trọng tâm ban đầu của k cụm từ với j = 1, …, k.

Bước 2: Tính toán khoảng cách và gán nhóm

Đối với mỗi đối tượng X_i ($1 \leq i \leq n$), tính toán khoảng cách từ nó tới mỗi trọng tâm m_j với j = 1, …, k, sau đó tìm trọng tâm gần nhất đối với mỗi đối tượng.

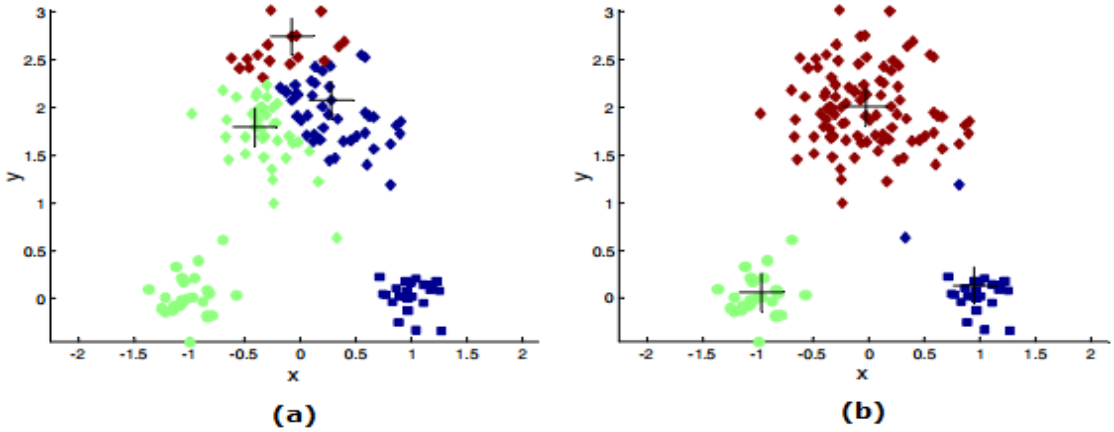
Bước 3: Cập nhật lại trọng tâm

Đối với mỗi j = 1,.., k, cập nhật trọng tâm cụm mj bằng cách xác định trung bình cộng của các véc-tơ đối tượng dữ liệu.

Bước 4: Điều kiện dừng

Lặp các bước 2 và 3 cho đến khi các trọng tâm của cụm không thay đổi. Hiệu quả của các thuật toán: để đánh giá tính hiệu quả của hai thuật toán trên, tiến hành thử nghiệm với bộ dữ liệu là tập các điểm trong không gian 2 chiều.

Khi tiến hành phân nhóm, cả hai thuật toán K-Means và Seeded–Kmean đều gán cùng số nhóm k = 3, tuy nhiên đối với thuật toán K–Means việc khởi tạo 3 đối tượng làm trọng tâm bằng cách chọn ngẫu nhiên và không trùng lặp nhưng với thuật toán Seed–Means việc khởi tạo 3 đối tượng làm trọng tâm được lấy từ tập giống đã được định nghĩa. Với CSDL mẫu được mô tả trong hình sau thì kết quả phân nhóm của hai thuật toán này như hình sau: (các điểm được phát hiện cùng một nhóm được minh họa bằng màu giống nhau).



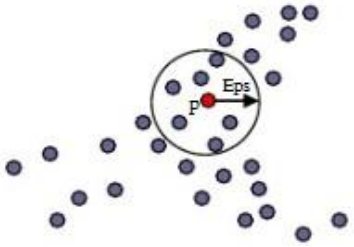
Hình 2.1: Kết quả phân nhóm thuật toán K–Means (a), Seed–Kmeans (b)

2.2 Thuật toán DBSCAN

Thuật toán phân nhóm dựa trên mật độ thông dụng nhất là thuật toán DBSCAN (Density - Based Spatial Phân nhóm of Applications with noise) do Ester, P. Kriegel và J. Sander đề xuất năm 1996. Thuật toán đi tìm các đối tượng mà có số đối tượng láng giềng lớn hơn một ngưỡng tối thiểu. Một nhóm được xác định bằng tập tất cả các đối tượng liên thông mật độ với các láng giềng của nó. Thuật toán DBSCAN dựa trên các khái niệm mật độ có thể áp dụng cho các tập dữ liệu không gian lớn đa chiều. Sau đây là một số định nghĩa và bổ đề được sử dụng trong thuật toán DBSCAN [5].

Định nghĩa 1: Các lân cận của một điểm p với ngưỡng Eps, ký hiệu

NEps(p) được xác định như sau: NEps(p) = {q ∈D | khoảng cách Dist(p, q) ≤ Eps}, D là tập dữ liệu cho trước.



Hình 2.2: Lân cận của p với ngưỡng Eps

Một điểm p muốn nằm trong một nhóm C nào đó thì NEps(p) phải có tối thiểu MinPts điểm. Theo định nghĩa trên, chỉ những điểm thực sự nằm trong nhóm mới thoả mãn điều kiện là điểm thuộc vào nhóm. Những điểm nằm ở biên

của nhóm thì không thỏa mãn điều kiện đó, bởi vì thông thường thì lân cận với ngưỡng Eps của điểm biên thì bé hơn lân cận với ngưỡng cũng Eps của điểm nhân. Để tránh được điều này, ta có thể đưa ra một tiêu chuẩn khác để định nghĩa một điểm thuộc vào một nhóm như sau: Nếu một điểm p muốn thuộc một nhóm C phải tồn tại một điểm q mà p $\in$ NEps(q) và số điểm trong NEps(q) phải lớn hơn số điểm tối thiểu.

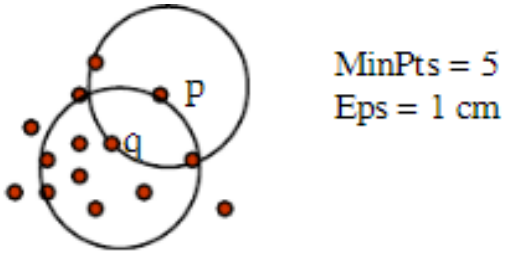
Định nghĩa 2: Mật độ đến được trực tiếp (Directly Density- reachable)

Một điểm p được gọi là mật độ đến được trực tiếp từ điểm q với ngưỡng Eps và MinPts trong tập đối tượng D nếu:

1) $p \in N_{Eps}(q)$ Với $p \in N_{Eps}(q)$ là tập con của D.

2) $||p \in N_{Eps}(q)|| \geq MinPts$, điều kiện đối tượng nhân.

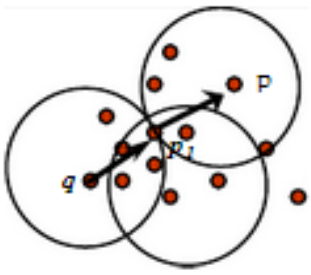
Điểm q gọi là điểm nhân. Ta thấy rằng nó là một hàm phản xạ và đối xứng đối với hai điểm nhân và bất đối xứng nếu một trong hai điểm đó không phải là điểm nhân.



Hình 2.3: Mật độ đến được trực tiếp

Định nghĩa 3: Mật độ đến được (Density- Reachable)

Một điểm p được gọi là mật độ đến được từ một điểm q với hai tham số Eps và MinPts nếu tồn tại một dãy $p = p_1, p_2,\ldots,p_n = 1$ sao cho p_i+1 là mật độ đến được trực tiếp từ pi với i = 1,…, n-1.

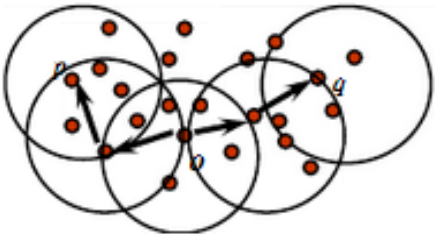


Hình 2.4: Mật độ đến được

Hai điểm biên của một nhóm C có thể không đến được nhau bởi vì cả hai có thể đều không thoả mãn điều kiện nhân. Mặc dù vậy, phải tồn tại một điểm nhân trong C mà cả hai điểm đều có thể đến được từ điểm đó.

Định nghĩa 4: Mật độ liên thông (Density - Connected)

Đối tượng p là mật độ liên thông với điểm q theo hai tham số Eps với MinPts nếu như có một đối tượng o mà cả hai đối tượng p, q điều là mật độ đến được o theo tham số Eps và MinPts nếu như có một đối tượng o mà cả hai đối tượng p, q đều là mật độ đến được p theo tham số Eps và MinPts.



Hình 2.5: Mật độ liên thông

Định nghĩa 5: Nhóm và nhiễu

Cho D là một tập các đối tượng dữ liệu. Một tập con C khác rỗng của D được gọi là một nhóm theo Eps và MinPts nếu thỏa mãn điều kiện:

(1) Cực đại: $\forall p, q \in D$, nếu $p \in C$ và q là mật độ đến được p theo Eps và MinPts thì $q \in C$.

(2) Với $\forall p,q \in C$, p là mật độ liên thông với q theo Eps và MinPts.

Mọi đối tượng không thuộc nhóm nào cả thì gọi là nhiễu.

Với hai tham số Eps và MinPts cho trước, ta có thể khám phá các nhóm theo hai bước:

Bước 1: Chọn một điểm bất kỳ từ tập dữ liệu ban đầu thỏa mãn điều kiện nhân.

Bước 2: Lấy tất cả các điểm đến được mật độ với điểm nhân đã chọn ở trên để tạo thành cụm.

Bổ đề 1: Giả sử p là một đối tượng trong D, trong đó ||NEps(p)|| || ≥ MinPts, tập $O = \{o|o \in D$ và o là mật độ đến được từ p theo Eps và MinPts$\}$ là một nhóm theo Eps và MinPts.

Như vậy nhóm C không hoàn toàn duy nhất, tuy nhiên mỗi một điểm trong C đến được mật độ từ bất cứ một điểm nhân nào của C, vì vậy C chứa đúng một số điểm liên thông với điểm nhân tùy ý.

Bổ đề 2: Giả sử C là một nhóm theo Eps và MinPts, p là một điểm bất kỳ trong C với $||N_{Eps}(p)|| \geq$ MinPts.. Khi đó C trùng với tập $O = \{o | o \in D$ và o là mật độ đến được từ p theo Eps và MinPts$\}$.

Thuật toán:Để xác định các nhóm, DBSCAN bắt đầu với một điểm p bất kỳ và tìm tất cả các điểm lân cận mật độ với p. Nếu p là điểm nhân thì thủ tục này sẽ tạo ra một cụm (bổ đề 2). Nếu p là điểm biên, thì không có điểm nào kề mật độ với p và DBSCAN thăm điểm kế tiếp trong tập dữ liệu. Việc tìm các đối tượng lân cận mật độ được thực hiện bằng các truy vấn vùng liên tục. Một truy vấn cùng trả về tất cả các đối tượng giao với một vùng truy vấn xác định. Những truy vấn như thế được hỗ trợ một các hữu hiệu bởi các phương thức truy xuất dữ liệu không gian như cây R*(R*-Tree) [2] với dữ liệu từ một không gian véc-tơ hoặc cây M (M-tree) [3] với dữ liệu từ một không gian số liệu.

Nếu sử dụng các giá trị toàn cục cho Eps và MinPts, DBSCAN có thể trộn 2 nhóm (định nghĩa 5) thành một nhóm nếu hai nhóm này "gần" với nhau. Khoảng cách giữa hai tập điểm S1 và S2 được định nghĩa là:

dist(S1,S2) = min{dist(p, q) | p $\in$S1, q $\in$ S2}.

Hai tập điểm có các điểm tối thiểu của một nhóm mỏng sẽ tách rời nhau chỉ nếu khoảng cách giữa hai tập lớn hơn Eps. Do đó, lời gọi đệ quy của DBSCAN có thể là cần thiết đối với các nhóm được phát hiện với giá trị cao hơn MinPts. Tuy nhiên, đây không phải là một nhược điểm vì ứng dụng đệ quy của DBSCAN tạo ra một thuật toán cơ bản, khá hiệu quả.

Thuật toán DBSCAN bao gồm các bước sau:

INPUT: - Tập các đối tượng dữ liệu X = {x_1, x_2, …, x_n}

- Ngưỡng đến được giữa 2 đối tượng: Eps

- Số lượng các đối tượng tối thiểu trong nhóm: MinPts

OUTPUT: Các nhóm C_i (i = 1, .., k)

Bước 1: Chọn một đối tượng p tùy ý.

Bước 2: Lấy tất cả các đối tượng có mật độ đến được từ p với Eps và MintPts

Bước 3: Nếu p là điểm nhân thì tạo ra một nhóm theo Eps và MintPts.

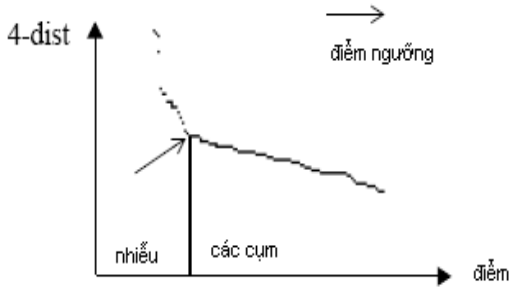
Bước 4: Nếu p là điểm biên, không có điểm nào có mật độ đến được từ p và DBSCAN sẽ đi thăm điểm tiếp theo của tập dữ liệu.

Bước 5: Quá trình tiếp tục cho đến khi tất cả các đối tượng được xử lý.

Thuật toán DBSCAN có thể tìm ra các cụm với hình thù bất kỳ, trong khi đó tại cùng một thời điểm ít bị ảnh hưởng bởi thứ tự của các đối tượng dữ liệu nhập vào. Khi có một đối tượng được chèn vào chỉ tác động đến một láng giềng xác định. Mặt khác, DBSCAN yêu cầu người dùng xác định bán kính Eps của các láng giềng và các láng giềng tối thiểu MinPts, thông thường các tham số này được xác định bằng phép chọn ngẫu nhiên hoặc theo kinh nghiệm.

Phương pháp xác định tham số Eps và MinPts: Trong phần này, chúng tôi xây dựng một heuristic đơn giản nhưng hiệu quả để xác định các tham số Eps và MinPts của nhóm "mỏng nhất" trong CSDL. Heuristic này được dựa trên các quan sát sau: Gọi d là khoảng cách từ điểm p tới k láng giềng gần nhất, thì d láng giềng của p chứa đúng k+1 điểm đối với hầu hết các điểm p, d láng giềng của p chứa nhiều hơn k+1 điểm chỉ khi một số điểm có cùng khoảng cách d tới điểm p. Hơn nữa, thay đổi hệ số k đối với một điểm trong nhóm không tạo ra một sự thay đổi lớn nào đối với d. Điều này chỉ xảy ra khi k láng giềng gần nhất của p với k=1, 2, 3… được xác định xấp xỉ trên một đường thẳng. Với k cho trước, định nghĩa một hàm k-dist từ CSDL D tới các số thực, ánh xạ mỗi điểm thông qua hàm khoảng cách tới k láng giềng gần nhất của nó. Khi các điểm trong CSDL được sắp xếp theo thứ tự giảm dần giá trị k-dist của nó, đồ thị của hàm này đưa ra một số dấu hiệu có liên quan đến phân bố mật độ trong CSDL. Chúng tôi gọi đồ thị này là đồ thị k-dist đã sắp xếp. Nếu ta chọn một điểm p bất kỳ, gán tham số Eps với k-dist(p) và gán tham số MinPts với k, tất cả các điểm bằng hoặc nhỏ hơn giá trị k-dist sẽ là những điểm hạt nhân. Nếu có thể xác định điểm ngưỡng với giá trị k-dist lớn nhất trong nhóm "mỏng nhất" của D, thì sẽ có các giá trị tham số mật độ.

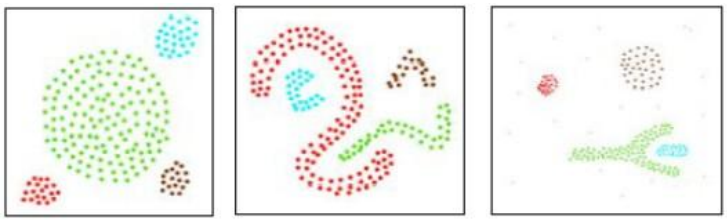
Điểm ngưỡng (threshold point) là điểm đầu tiên trong vùng đầu tiên của đồ thị k-dist đã được sắp xếp (xem hình sau). Tất cả các điểm có giá trị cao hơn k-dist (ở bên trái điểm ngưỡng) có thể coi là nhiễu, tất cả các điểm còn lại (ở bên phải điểm ngưỡng) được gán cho một số nhóm nào đó.



Hình 2.6: Đồ thị đã sắp xếp 4-dist đối với CSDL mẫu 3

Nói chung, việc xác định ra vùng đầu tiên một cách tự động khá khó khăn, nhưng lại rất dễ dàng đối với người sử dụng tự xác định thấy vùng này trên đồ thị. Vì vậy, chúng tôi đề xuất một phương pháp tương tác để xác định điểm ngưỡng.

Hiệu quả của thuật toán: Để đánh giá hiệu quả của DBSCAN, tiến hành thử nghiệm thuật toán với bộ dữ liệu chứa các điểm trong không gian 2 chiều cho kết quả thực nghiệm cho thấy thời gian chạy của DBSCAN cao hơn tổng số lượng các điểm. Với CSDL mẫu trong hình sau thì DBSCAN cho kết quả như hình vẽ sau: (các điểm được phát hiện cùng một nhóm được minh họa bằng màu giống nhau).



Hình 2.7: Các nhóm phát hiện được bởi và DBSCAN

Nếu các cụm có mật độ khác nhau nhiều thì DBSCAN sẽ không giữ được tính hiệu quả. Trên những dữ liệu như thế ta phải áp dụng mật độ của cụm có mật độ thấp nhất cho tất cả các cụm khác. Với các cụm có mật độ rất cao thì DBSCAN tốn nhiều thời gian để xác định lân cận của các điểm một cách không cần thiết.

2.3 Thuật toán BIRCH

BIRCH (Balanced Interative ReduCi ng and Clustering using Hierarchies) là thuật toán phân cụm phân cấp sử dụng chiến lược Top-down. Ý tưởng của BIRCH là không lưu toàn bộ đối tượng dữ liệu của các cụm trong bộ nhớ mà chỉ lưu các tham số thống kê. Đối với mỗi cụm dữ liệu, BIRCH chỉ lưu bộ ba (N, LS, SS), trong đó N là số đối tượng trong cụm, LS là tổng các giá trị thuộc tính của các đối tượng trong cụm, và SS là tổng bình phương của các giá trị thuộc tính của các đối tượng trong cụm. Bộ ba này được gọi là đặc trưng cụm (Cluster Feature – CF). Khi đó các cụm trong tập dữ liệu ban đầu sẽ được cho dưới dạng một cây CF.

Cây CF là cây cân bằng, nhằm để lưu trữ các đặc trưng của cụm. Cây CF chứa các nút trong và nút lá. Nút trong lưu giữ tổng các đặc trưng cụm của các nút con của nó. Một cây CF được đặc trưng bởi hai tham số:

Yếu tố nhánh (B): Nhằm xác định số tối đa các nút con của mỗi nút trong của cây;

Ngưỡng (T): Khoảng cách tối đa giữa bất kỳ một cặp đối tượng trong nút lá của cây, khoảng cách này còn gọi là đường kính của các cụm con được lưu tại các nút lá.

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

Thuật toán BIRCH thực hiện qua giai đoạn sau:

Bước 1: Duyệt tất cả các đối tượng trong CSDL và xây dựng một cây CF khởi tạo. 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, tất cả các nút trỏ 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 xây dựng một cây CF nhỏ hơn bằng cách điều khiển bởi tham số T (vì tăng T sẽ làm hòa nhập một số các 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 kỹ thuật phân cụm thí 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 đánh giá từ bước 3: Đây là một bước tùy chọn để duyệt lại tập dữ liệu và gắn 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 .

Đánh giá thuật toán BIRCH.

Ưu điểm: Nhờ sử dụng cây CF, BIRCH có tốc độ phân cụm nhanh, độ phức tạp O(n) (vì BIRCH chỉ duyệt toàn bộ dữ liệu một lần). BIRCH được áp dụng với tập dữ liệu lớn, đặc biệt phù hợp với dữ liệu gia tăng theo thời gian.

Nhược điểm: Chất lượng cụm được khám phá bởi BIRCH là không tốt. Tham số T ảnh hưởng lớn đến kích thước và tính tự nhiên của cụm.

2.4 Thuật toán INCREMENTAL DBSCAN

Hiện nay đã có nhiều thuật toán phân nhóm động, trong phần này chúng tôi sử dụng thuật toán DBSCAN như là một cơ sở cho thuật toán phân nhóm động.

DBSCAN, như đã giới thiệu trong [5], được ứng dụng cho CSDL tĩnh. Trong môi trường CSDL thường xuyên cập nhật, ta cần có thuật toán phân nhóm động bởi những mẫu dữ liệu này có thể thay đổi theo thời gian. Sau khi thêm và xóa từ CSDL, việc phân nhóm được khám phá bởi DBSCAN phải được cập nhật.

Trong phần trước, chúng tôi khảo sát phần nào của một nhóm đang tồn tại bị ảnh hưởng bởi sự cập nhật của CSDL. Dựa trên khái niệm hình thức của các nhóm, ta có thể chứng minh rằng thuật toán phân nhóm dữ liệu động thu được cùng kết quả như thuật toán DBSCAN tĩnh. Đây là một tiến bộ quan trọng của thuật toán động này.

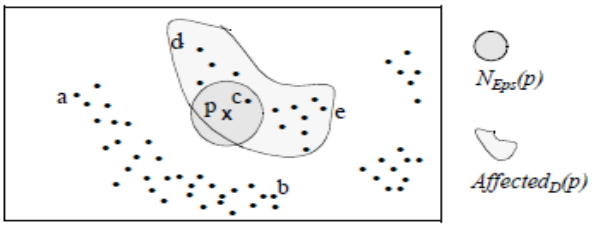
2.4.1 Các đối tượng bị ảnh hưởng

Trong phần này chúng tôi sẽ chứng tỏ rằng các thay đổi với quá trình phân nhóm của CSDL D là bị giới hạn trong lân cận của một đối tượng p được thêm vào hay xóa đi. Các đối tượng đã chứa trong NEps(p) có thể thay đổi đặc tính đối tượng nòng cốt của chúng, nghĩa là nòng cốt có thể trở thành đối tượng không nòng cốt và ngược lại. Các đối tượng đã chứa trong

N2Eps(p)\NEps(p) giữ đặc tính nòng cốt của chúng, nhưng các đối tượng không nòng cốt có thể thay đổi trạng thái liên thông của chúng, nghĩa là các đối tượng biên có thể trở thành các đối tượng lạc loài hoặc ngược lại, vì lân cận Eps của chúng có thể chứa các đối tượng với một đặc tính nòng cốt đã thay đổi. Đối với tất cả các đối tượng bên ngoài N2Eps(p), nó quyết định rằng bản thân các đối tượng này lẫn các đối tượng trong lân cận Eps của chúng cũng không thay đổi đặc tính đối tượng nòng cốt. Vì thế, trạng thái liên thông của các đối tượng này là không thay đổi.

Sau khi thêm vào vài đối tượng p, thì các đối tượng không nòng cốt (các đối tượng biên hoặc đối tượng lạc loài) trong NEps(p) có thể trở thành các đối tượng nòng cốt ngụ ý rằng các phân bố liên thông mới có thể được thiết lập có nghĩa là dây chuyền $p_1,p_2,\ldots,p_n,p= = r$, $p_n = s$ với p_i+1 có phân bố có thể đi đến trực tiếp từ pi với hai đối tượng r và s có thể trở thành không có phân bố có thể đi đến được với nhau trước khi thêm vào. Sau đó, một trong những p_i với $i < n$ phải được chứa trong NEps(p).

Khi xóa vài đối tượng p, các đối tượng tượng nòng cốt trong NEps(p) có thể trở thành đối tượng không nòng cốt ngụ ý rằng phân bố liên thông có thể bị hủy bỏ, nghĩa là có thể không còn dây chuyền $p_1, p_2, \ldots, p_n, p_1 = r, p_n = s$ với p_i+1 có phân bố có thể đi đến trực tiếp từ pi với hai đối tượng r và s đã có phân bố có thể đi đến được với nhau trước khi xóa. Một lần nữa, một trong những p_i với $i < n$ phải được chứa trong NEps(p).



Hình 2.8: Các đối tượng bị ảnh hưởng trong một CSDL mẫu

Hình trên minh họa sự bàn luận của chúng tôi sử dụng một CSDL mẫu với các đối tượng hai chiều và một đối tượng p được thêm vào hoặc bị xóa đi. Các đối tượng a và b là phân bố liên thông Eps như đã mô tả và MinPts = 4 không sử dụng một trong các phần tử của NEps(p). Vì thế, a và b cũng cùng thuộc nhóm độc lập với p. Nói cách khác, các đối tượng d và e trong D\NEps(p) chỉ là phân bố liên thông qua c trong NEps(p) nếu đối tượng p tồn tại, để cho mối quan hệ thành viên nhóm của d và e bị ảnh hưởng bởi p.

Nói chung, việc thêm hay xóa một đối tượng p, thì tập các đối tượng bị ảnh hưởng, nghĩa là tập các đối tượng có khả năng thay đổi mối quan hệ thành viên nhóm sau khi cập nhật, là một tập các đối tượng trong NEps(p) cộng thêm tất cả các đối tượng có phân bố có thể đi đến được từ một trong những đối tượng này trong D∪{p}. Mối quan hệ thành viên nhóm của tất cả đối tượng khác không ở trong tập các đối tượng bị ảnh hưởng sẽ không thay đổi. Điều này là trực giác của việc xác định và theo dõi bổ đề sau. Cụ thể, bổ đề khẳng định rằng một nhóm C trong CSDL không phụ thuộc vào việc thêm hoặc xóa một đối tượng nòng cốt của nhóm nằm bên ngoài tập bị ảnh hưởng của D đối với p (ký hiệu là AffectecdD(p). Ghi chú rằng nhóm là được xác định duy nhất bởi bất kỳ đối tượng nòng cốt của nó. Vì thế, theo định nghĩa

AffectedD(p) thì tất cả các đối tượng nòng cốt của nhóm nằm ngoài (hoặc trong) tập AffectedD(p).

Định nghĩa 6: (Các đối tượng bị ảnh hưởng) Gọi D là một CSDL gồm đối tượng và p là một vài đối tượng (hoặc ở trong hoặc ở ngoài D). Chúng tôi định nghĩa tập các đối tượng trong D bị ảnh hưởng bởi việc thêm hoặc xóa p như:

$$Affected_D(p) = N_{Eps}(p) \cup \{q \mid \exists\ o \in N_{Eps}(p) \wedge\ q >_{D \cup \{p\}} o\}.$$

Bổ đề 1: Cho D là một tập các đối tượng và p là các đối tượng. Thế thì:

$\forall o \in D: o \notin Affected_D(p) \Rightarrow \{q \mid q >_{D \setminus \{p\}} o\} = \{q \mid q >_{D \cup \{p\}} o\}$.

Bổ đề 2 : Cho D là một tập các đối tượng. Ngoài ra $D^* = D \cup \{p\}$ sau khi thêm vào một đối tượng *p* hoặc $D^* = D \setminus \{p\}$ sau khi xóa đối tượng p và gọi c là một đối tượng nòng cốt trong D^*. $C = \{o \mid o >_{D^*} c\}$ là một nhóm trong D^* và:

$C \subseteq Affected_D(p) \Leftrightarrow \exists q, q': q \in N_{Eps}(q',\ q' \in NEps(p), c >_{D^*} q, q$

là đối tượng nòng cốt trong D^* và q' là đối tượng nòng cốt trong $D \cup \{p\}$.

Theo bổ đề 2, chiến lược chung để cập nhật công việc phân nhóm sẽ bắt đầu thuật toán DBSCAN chỉ với các đối tượng nòng cốt mà ở trong lân cận Eps của một đối tượng nòng cốt (trước đó) trong NEps(p). Tuy nhiên, không cần thiết khám phá lại phân bố liên thông đã được biết từ việc phân nhóm trước và không bị thay đổi bởi tác vụ cập nhật. Với mục đích đó, chúng tôi chỉ cần xem xét các đối tượng nòng cốt trong lân cận Eps của các đối tượng đó mà thay đổi đặc tính đối tượng nòng cốt của chúng như một kết quả của việc cập nhật.

Trong trường hợp thêm những đối tượng này có thể là liên thông sau khi thêm. Trong trường hợp xóa phân bố liên thông giữa chúng có thể bị mất. Trong trường hợp tổng quát, thông tin này có thể được xác định bằng cách sử dụng rất ít các truy vấn vùng. Thông tin còn lại cần điều chỉnh việc phân nhóm có thể được suy từ mối quan hệ thành viên nhóm trước khi cập nhật. Định nghĩa 7 giới thiệu các khái niệm hình thức cần thiết để diễn giải cách tiếp cận này. Ghi nhớ: các đối tượng với một đặc tính đối tượng nòng cốt thay đổi đều được định vị trong NEps(p).

Định nghĩa 7: (các đối tượng hạt giống-seed cho việc cập nhật) cho D là một tập các đối tượng và p là một đối tượng được thêm hoặc xóa.

Ta có các khái niệm sau:

$UpdSeed_{Ins} = \{q \mid q$ là một đối tượng nòng cốt trong $D \cup \{p\}, \exists\ q':$

q' là đối tượng nòng cốt trong $D \cup \{p\}$ nhưng không ở trong *D* và $q \in N_{Eps}(q')\}$.

Ta gọi các đối tượng q∈UpdSeed "các đối tượng hạt giống để cập nhật". Ghi chú rằng những tập này có thể được tính một cách hữu hiệu hơn nếu ta bổ sung lưu trữ cho mỗi đối tượng một số đối tượng trong lân cận của nó khi khởi tạo phân nhóm CSDL. Sau đó, chúng tôi chỉ cần thực hiện truy vấn vùng đơn cho đối tượng p được thêm hoặc xóa để dò tìm tất cả đối tượng q' cùng với đặc tính đối tượng nòng cốt thay đổi (nghĩa là các đối tượng trong NEps(p) với số = MinPts trong trường hợp xóa).

Chỉ đối với các đối tượng q' (nếu có bất kỳ) ta phải truy vấn NEps(q') để xác định tất cả các đối tượng q trong tập UpdSeed. Vì tại lúc này lân cận Eps của p vẫn còn trong bộ nhớ chính chúng tôi kiểm tra tập này đầu tiên với lân cận q' và thực hiện một truy vấn vùng bổ sung chỉ nếu có thêm các đối tượng với đặc tính đối tượng cố lõi thay đổi sau khi cập nhật (khác với các đối tượng được thêm hoặc bị xóa p) là không được thường xuyên lắm.

Vì thế, trong hầu hết các trường hợp chúng tôi chỉ phải thực hiện truy vấn lân cận Eps cho p và thay đổi bộ đếm số đối tượng trong lân cận của các đối tượng được truy vấn.

2.4.2 Trường hợp thêm

Khi thêm một đối tượng mới p, phân bố liên thông mới có thể được thiết lập, nhưng không có cái nào bị hủy bỏ. Trong trường hợp này, đủ để hạn chế ứng dụng thủ tục phân nhóm đối với UpSeedIns.

Nếu ta phải thay đổi quan hệ thành viên nhóm cho một đối tượng từ C thành D chúng tôi thực hiện thay đổi quan hệ thành viên nhóm như thế cho tất cả thành viên nhóm như thế cho tất cả các đối tượng khác trong C. Việc thay đổi quan hệ thành viên nhóm của những đối tượng này không liên quan đến ứng dụng thuật toán phân nhóm nhưng có thể được xử lý bởi lưu trữ đơn giản thông tin về các nhóm được trộn.

Khi thêm một đối tượng p vào CSDL D, ta phân biệt các trường hợp sau:

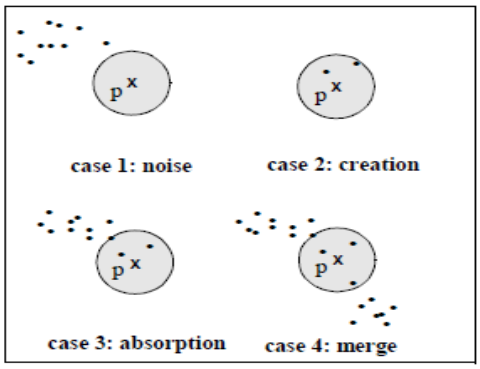
(1) Lạc loài: UpdSeedIns là rỗng, nghĩa là không có các đối tượng nòng cốt mới sau khi thêm p. Vì thế, p là một đối tượng lạc loài và không có thay đổi nào khác.

(2) Tạo lập: UpdSeedlns chỉ chứa các đối tượng nòng cốt không thuộc vào một nhóm nào trước khi thêm p, nghĩa là chúng là các đối tượng lạc loài hoặc bằng p và nhóm mới chứa các đối tượng lạc loài này hay p được tạo.

(3) Sự thu hút: UpdSeedIns chứa các đối tượng nòng cốt là các thành viên của một nhóm C trước khi thêm. Đối tượng p và có thể vài đối tượng lạc loài bị thu hút vào nhóm C.

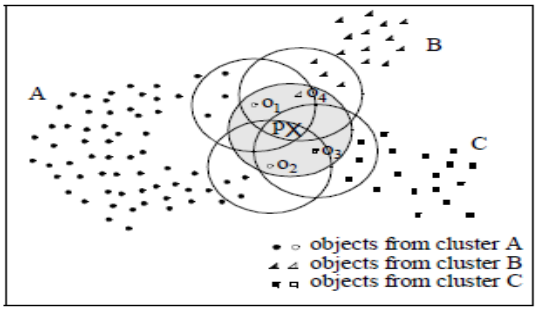
(4) Trộn: UpdSeedIns chứa các đối tượng nòng cốt là các thành viên của nhiều nhóm trước khi thêm. Tất cả các nhóm này và đối tượng p được trộn thành một nhóm.

Hình 2.11 minh họa cách hình thành đơn giản nhất của các trường hợp khác nhau khi thêm một đối tượng p vào cùng một CSDL các điểm 2 chiều, sử dụng tham số Eps như đã miêu tả MinPts=3.



Hình 2.9: Các trường hợp khác nhau của thuật toán

Trong hình 2.12 trình bày thêm một ví dụ phức tạp của việc trộn các nhóm khi thêm một đối tượng p. Trong ví dụ này giá trị cho Eps như được miêu tả và MinPts=6. Vì thế, điểm thêm vào p không phải là đối tượng nòng cốt, nhưng O_1, O_2, O_3 và O_4 là các đối tượng nòng cốt sau khi cập nhật. Công việc phân nhóm trước có thể được cập nhật bằng cách chỉ phân tích lân cận Eps của những đối tượng này: nhóm A được trộn với B và C vì O_1 và O_4 cũng như O_2 và O_3 có phân bố có thể đến được trực tiếp lẫn nhau, ngầm định cho cho việc trộn B và C.



Hình 2.10: Thể hiện trộn các nhóm A, B, C bằng thuật toán thêm

Sự thay đổi quan hệ thành viên nhóm cho các đối tượng trong trường hợp trộn các nhóm có thể được thực hiện rất hữu hiệu bằng lưu trữ đơn giản thông tin về các nhóm được trộn. Lưu ý rằng loại trộn "chuyển tiếp" này chỉ có thể xảy ra nếu MinPts là lớn hơn 5, vì ngược lại p sẽ là một đối tượng nòng cốt và sau đó tất cả các đối tượng trong NEps(p) sẽ có phân bố có thể đến được từ p.

2.4.3 Trường hợp xóa

Ngược lại với việc thêm vào, khi ta xóa một đối tượng p, các phân bố liên thông có thể bị hủy bỏ, nhưng không có liên thông mới nào được thiết lập. Trường hợp khó khăn cho việc xóa xảy ra khi nhóm C của p không còn phân bố liên thông, thông qua các đối tượng nòng cốt (trước) trong NEps(p) sau khi xóa p.

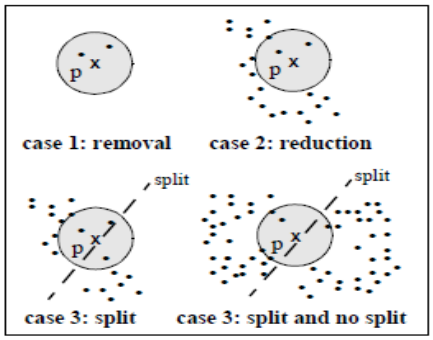
Trong trường hợp này, ta không biết tổng quát có bao nhiêu đối tượng phải kiểm tra trước khi có thể xác định C phải được chia ra hay không. Tuy nhiên, trong hầu hết các trường hợp, tập các đối tượng này là rất nhỏ vì việc phân chia một nhóm là không thường xuyên và tổng quát tình huống không chia sẽ được dò tìm trong một lân cận của đối tượng bị xóa p. Khi xóa một đối tượng p từ một CSDL D ta có thể phân biệt các trường hợp sau:

(1) Loại bỏ: UpdSeedDel là rỗng, nghĩa là không có các đối tượng nòng cốt trong lân cận các đối tượng mà có thể mất đặc tính đối tượng nòng cốt của chúng khi xóa p. Sau đó p bị xóa khỏi D và cuối cùng các đối tượng khác trong NEps(p)thay đổi từ nhóm C cũ sang thành phần lạc loài. Nếu điều này xảy ra thì nhóm C hoàn toàn bị hủy bỏ vì sau đó C không có các đối tượng nòng cốt ngoài NEps(p).

(2) Giảm bớt: Tất cả các đối tượng trong UpdSeedDel là có phân bố có thể đến được trực tiếp với nhau. Thế thì p được xóa khỏi D và vài đối tượng trong NEps(p) có thể trở thành lạc loài.

(3) Khả năng phân chia: Các đối tượng trong UpdSeedDel không có phân bố có thể đến trực tiếp với nhau. Các đối tượng này chỉ thuộc vào nhóm C trước khi xóa p. Bây giờ ta phải kiểm tra xem các đối tượng này là có phân bố liên thông với nhau hay không trong nhóm cũ C. Phụ thuộc vào sự tồn tại của các phân bố liên thông như thế, ta có thể phân biệt tình huống phân chia và không phân chia.

Hình 2.13 minh họa các trường hợp khác nhau khi xóa p từ một CSDL mẫu với các đối tượng 2 chiều sử dụng các tham số Eps như đã miêu tả và MinPts=3.



Hình 2.11: Các trường hợp khác nhau của thuật toán xóa

Ghi chú rằng tình huống như đã miêu tả trong trường hợp 3 có thể xảy ra đồng thời. Nếu trường hợp (3) xảy ra, thì thủ tục phân nhóm cũng phải xem xét các đối tượng bên ngoài UpdSeedDel, nhưng nó ngừng lại trong các trường hợp tình trạng không phân chia ngay sau khi các đối tượng từ tập UpdSeedDel có phân bố liên thông với nhau.

Trường hợp (3) được cài đặt bởi một thủ tục tương tự với thuật toán DBSCAN bắt đầu song song từ các phần tử của tập UpdSeedDel. Sự khác nhau chính đó là các ứng viên để mở rộng sau này được quản lý trong một hàng đợi thay vì stack. Vì thế, việc tìm theo bề rộng trước các phân bố liên thông bị mất được thực hiện hiệu quả hơn tìm kiếm theo chiều sâu trước vì các lý do sau:

Trong tình huống không phân chia, ta ngừng ngay sau khi tất cả các thành viên của UpdSeedDel được tìm thấy là phân bố liên thông với số lượng các đối tượng nhỏ nhất (yêu cầu số lượng nhỏ nhất cho các truy vấn vùng) được dò đầu tiên.

Tình huống phân chia trong trương hợp tổng quát là trường hợp có chi phí lớn hơn vì các thành phần của nhóm bị chia thực sự phải được khám phá. Thuật toán ngừng khi gần như thành phần cuối cùng được duyệt đến. Thông thường, một nhóm chỉ được chia thành hai thành hai phần và một trong hai phần đó là tương đối nhỏ. Sử dụng phương pháp tìm kiếm theo bề rộng trước chúng tôi chỉ phải duyệt phân nhỏ hơn và một tỷ lệ nhỏ phần lớn hơn.

Thuật toán: Giả sử rằng, chúng tôi đã có một tập các đặc trưng của các đối tượng và nhóm dữ liệu sau khi được phân tích và tiến hành phân nhóm bởi thuật toán DBSCAN. Mục tiêu của thuật toán là tiến hành xác định nhóm và xác nhập cho các đối tượng trong CSDL liệu mới vào CSDL hiện hành theo các nguyên lý đã được trình bày trong thuật toán phân nhóm dữ liệu động.

Mã giả đề xuất cho thuật toán phân nhóm dữ liệu động gồm các bước:

INPUT:

- Old RTree: Tập các đối tượng dữ liệu cũ;
- New RTree: Tập các đối tượng dữ liệu mới;
- Point: đối tượng đầu tiên được duyệt.

OUTPUT: New RTree: Tập các đối tượng dữ liệu mới đã được cập nhật.

```
IncrementalDBscan ( Old RTree, New RTree, Point)
Get the Neighborhood Points of the Point in Both the RTree
For Every Neighborhood Point of the Given Point
        If the Neighborhood Point is a core Point
                If the Neighborhood Point was not a Core Point earlier
                        If the Neighborhood Point belongs to a Cluster
                        Add the Neighborhood point to a List 'Change' to
                process Later
                Else
                        Mark to Change the Cluster Later
```

Get the Neighborhood Points of the Neighborhood Point and add them into a list Np

For every Point in the List Np

If the Point is Noise

Mark to Change Later

If the Point Belongs to a Cluster

Add the point to the List 'Change' to process Later

Else

Add the Neighborhood Point to the List 'Change' to process

If No New Core Points

Assign the Cluster ID as Noise

Else

If 'Change' List has No elements Then

Assign all Marked Points to a New Cluster

Else If Change List has Only one Element

Add the Point to the 'Change' Cluster

Else

Update All the CusterID of the Points in the 'Change' List

End IncrementalDBSCAN.

2.5 Thuật toán phân nhóm cây hậu tố

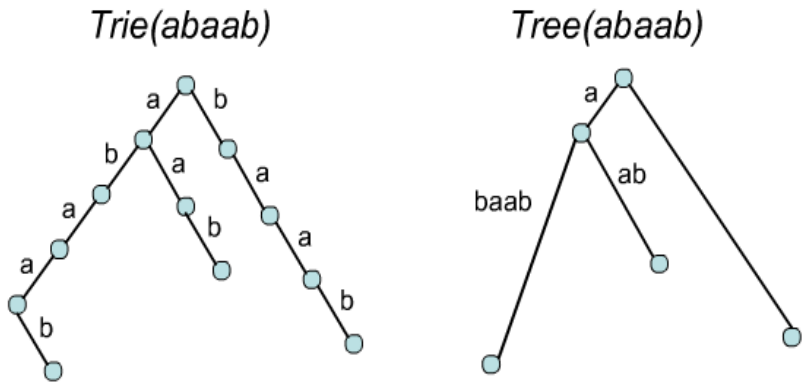
2.5.1 Cây hậu tố

a. Giới thiệu về cây hậu tố

Để tìm các xâu con chung cho hai hay nhiều chuỗi chúng tôi cần sử dụng một cấu trúc đặc biệt là cây hậu tố. Một cây hậu tố cho xâu s chứa tất cả các suffix của s và hỗ trợ tìm kiếm với tốc độ cao, cụ thể tìm xem một xâu có độ dài l có phải là xâu con của s không và nó chạy trong thời gian O(l).

Trước hết, chúng tôi phải làm rõ hai khái niệm suffix trie và cây hậu tố.

Để có cái nhìn một cách trực quan về suffix trie và cây hậu tố, chúng tôi xem xét ví dụ sau: xét xâu S=abaab.



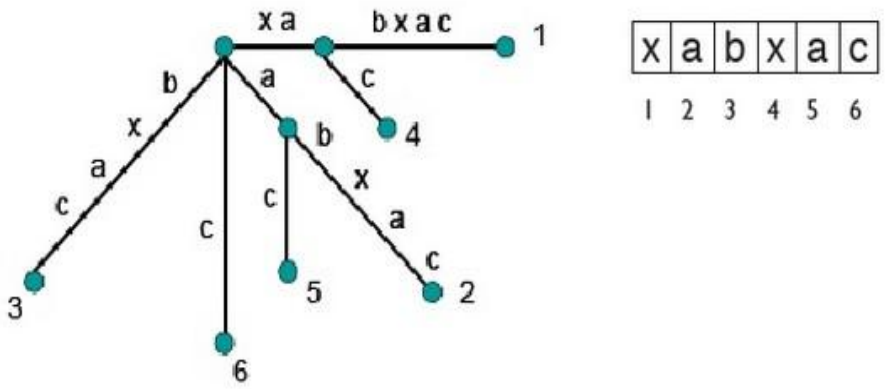
Hình 2.12: Suffix trie và cây hậu tố của xâu S = abaab

Cấu trúc dữ liệu cây hậu tố, Tree(T) là một cây biểu diễn tất cả các suffix của xâu S. CTDL này có kích thước tuyến tính: |Tree(T)|= O(|T|) và có thể được xây dựng trong thời gian tuyến tính O|T| trong khi đó suffix trie Trie(T) lại có kích thước khá lớn |Trie(T)| = O($|T^2|$) bởi lẽ Trie(T) cũng biểu diễn tất cả các suffix của T, tuy nhiên mỗi cạnh lại được gán nhãn là một kí tự. Điều này làm cho kích thước của nó tăng lên khá nhiều.

Định nghĩa 1[9]: Cây hậu tố T của chuỗi m kí tự S là cây có hướng, có gốc có các tính chất sau:

- Các đường đi từ gốc đến lá tương ứng 1-1 với các hậu tố của S.
- Mỗi nút trong, trừ nút gốc, có ít nhất là hai con.
- Mỗi cạnh được gán nhãn là một xâu con khác rỗng của S.
- Không có hai cạnh nào của cùng một nút có nhãn bắt đầu bằng cùng một kí tự.
- Với một lá bất kỳ sự kết hợp các nhãn của các cạnh trên đường đi từ gốc tới I được gọi là một suffix của S bắt đầu tại vị trí $Is_i…,s_m$.

Dưới đây là một cây hậu tố của sâu S=xabxac:



Hình 2.13: Cây hậu tố cho chuỗi S = xabxac

Định nghĩa của cây hậu tố không đảm bảo một cây như vậy luôn tồn tại với mọi xâu S. Nếu một hậu tố của S lại là tiền tố của một hậu tố khác thì đường đi từ gốc đến nó sẽ không kết thúc bởi một nút lá. Ví dụ, nếu bỏ chữ c trong xâu S, hậu tố xa là tiền tố của hậu tố xabxa nên không có đường đi nào từ gốc đến lá tương ứng với xa.

Để đảm bảo luôn dựng được cây hậu tố người ta thường thêm một kí tự đặc biệt vào cuối xâu S, gọi là kí tự kết thúc, để không có bất cứ hậu tố nào là tiền tố của hậu tố khác. Kí tự này phải không xuất hiện trong xâu ban đầu, người ta thường chọn một kí tự không có trong bảng chữ cái. Trong các tài liệu thường dùng ký tự $ để kí hiệu.

b. Xây dựng cây hậu tố

Năm 1973 Weiner đã đưa ra thuật toán cài đặt cây hậu tố với độ phức tạp tuyến tính về thời gian, cho tới năm 1976 McCreight đã phát triển một thuật toán hiệu quả hơn về mặt không gian và năm 1995 Ukkonen đã đưa ra một thuật toán đơn giản, dễ hiểu và tối ưu hơn hai thuật toán trước đó, trong phần này chúng tôi chủ yếu tìm hiểu thuật toán xây dựng cây hậu tố của Ukkonen. Để giảm thời gian xây dựng cây hậu tố, Ukkonen đã đưa ra khái niệm Suffix link.

Định nghĩa 2 [9]: Cho trước:

- Xâu x_α, x là một kí tự, α là một xâu.
- Đỉnh trong v có nhãn là x_α.

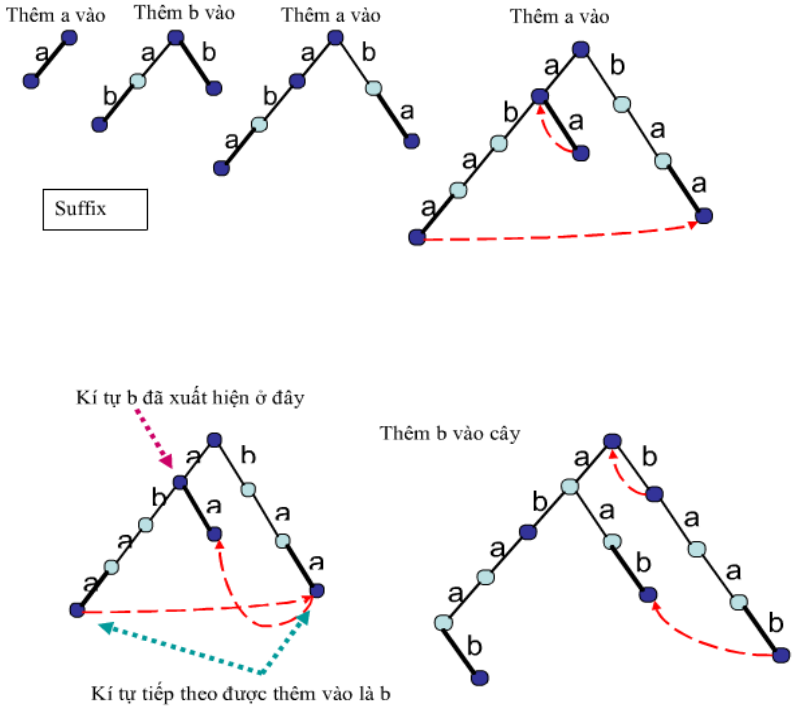
Nếu tồn tại một đỉnh $s(v)$ với nhãn α, thì con trỏ từ $v \rightarrow s(v)$ được gọi là suffix link ; Nếu α là xâu rỗng thì $s(v)$ chính là gốc ; Không có Sufix

link nào xuất phát từ gốc. Trong bất kỳ cây hậu tố T_i nào, nếu tồn tại đỉnh v có nhãn $x\alpha$ thì sẽ tồn tại đỉnh w thuộc T_i có nhãn α.

Xét xâu $T = t_1t_2t_n\$$, $P_i = t_1t_2\ldots t_i$ là prefix của T. Ý tưởng của thuật

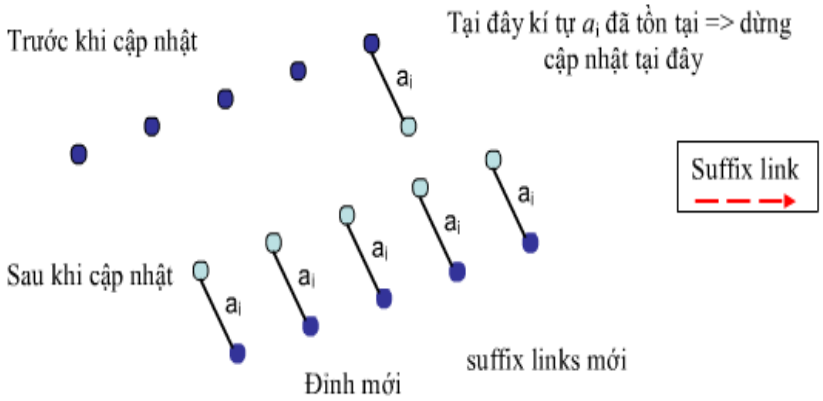
toán Ukkonen : cập nhật Trie(P_i) để thu được Trie(P_{i+1}).

Ta xét ví dụ xây dựng Suffix Trie từ xâu S=abaab.



Hình 2.14: Các bước tạo cây hậu tố của xâu S=abaab

Để thu được Trie(Pi+1) từ Trie(Pi): Ta sử dụng quy tắc thêm ai vào cây đã chứa ai được mô tả trực quan như hình dưới:



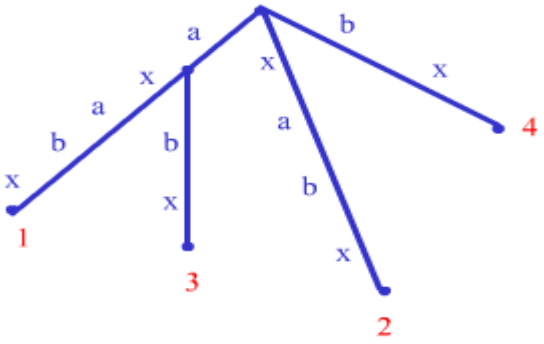
Hình 2.15: Quy tắc thêm kí tự ai vào cây đã chứa ai

Ý tưởng của thuật toán Ukkonen trong việc xây dựng cây hậu tố :

- Construct cây hậu tố T_1
- For i = 1 to m - 1 do

{Phase i + 1 buid T_i+1 from Ti by adding character number i+1}

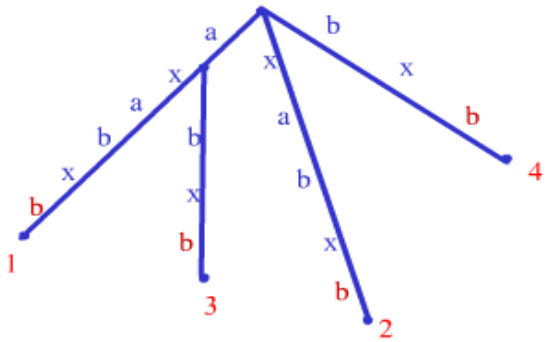
Có 3 quy tắc khi thêm một kí tự vào cây cho trước: Áp dụng với cây hậu tố tạo bởi xâu S=axabx:



Hình 2.16: Cây hậu tố T của xâu S = axabx

Tại bước i+1, extention thứ j đặt suffix S[j…i] là β

(1) Nếu β kết thúc tại lá trong cây Ti →thêm S(i+1) vào cuối nhãn của cạnh ứng với lá. Áp dụng vào cây T như hình sau: thêm vào kí tự b ta được xâu S = axabxb.

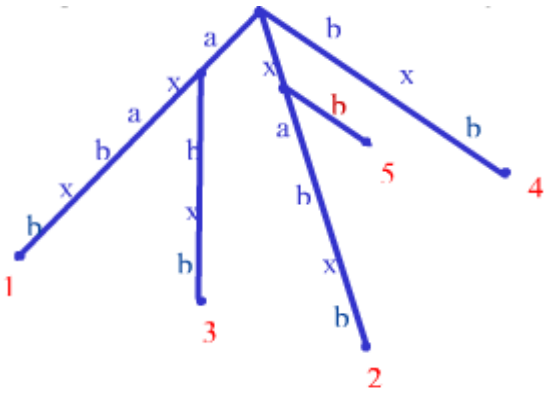


Hình 2.17: Cây hâu tố T của xâu S=axabxb theo quy tắc 1

(2) Nếu không có đường nào kết thúc của β bắt đầu với S(i+1) nhưng có ít nhất một đường được gán nhãn tiếp tục từ kết thúc của β → tạo cạnh ứng với lá

mới bắt đầu kết thúc của β gán cho cạnh đó nhãn S(i+1). Tạo ra đỉnh mới nếu β kết thúc bên trong một cạnh, gán cho lá ứng với điểm cuối của cạnh này là j.

Trở lại ví dụ trong hình 26: Điểm cuối của β là x; không có đường nào chứa "xb" nhưng có một đường được gán nhãn chứa "xa" → extention j = 5 được tạo.



Hình 2.18: Cây hậu tố T của xâu S = axabxb theo quy tắc 2

(3) Nếu tồn tại đường đi từ kết thúc của β mà bắt đầu với S(i+1) → không cần làm gì nữa.

2.5.2 Cây hậu tố - Cây hậu tố tổng quát

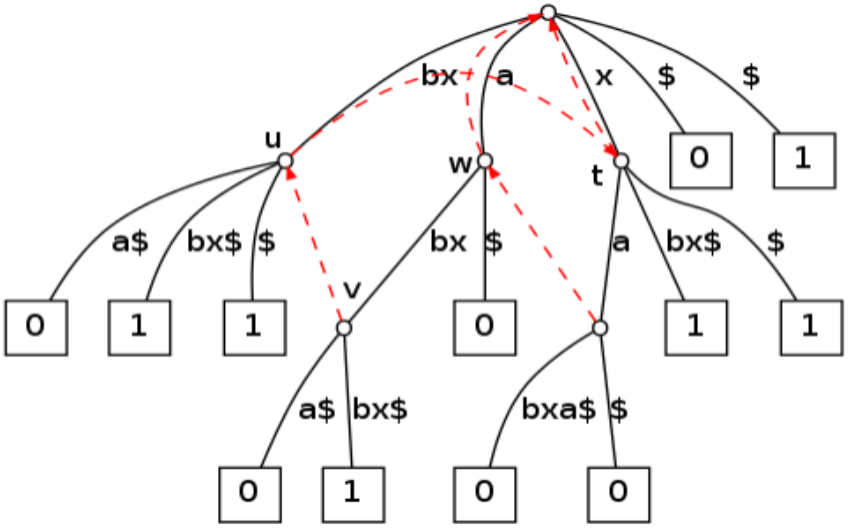
a. Khái niệm

Trong các phần trên ta đã từng bước xây dựng cây hậu tố cho một chuỗi. Để giải quyết bài toán tìm các nhóm từ chung của hai hay nhiều chuỗi hay văn bản ta cần mở rộng khái niệm cây hậu tố để chứa nhiều chuỗi hơn trong một cấu trúc dữ liệu chung.

Định nghĩa: Cho tập các chuỗi $\{S_1, S_2,..., S_K\}$, cây hậu tố tổng quát cho tập các chuỗi này là cây sao cho:

- Các đường đi từ gốc đến lá tương ứng 1-1 với các hậu tố của S_i.
- Mỗi nút trong, trừ nút gốc, có ít nhất là hai con.
- Mỗi cạnh được gán nhãn là một xâu con khác rỗng của S.
- Không có hai cạnh nào của cùng một nút có nhãn bắt đầu bằng cùng một kí tự.

Để phân biệt hậu tố của các chuỗi khác nhau, mỗi chuỗi được bổ sung một kí tự kết thúc khác nhau và không có trong bảng chữ cái. Mỗi nút lá của cây tương ứng với một hậu tố của một chuỗi nhất định và được gán nhãn bằng chỉ số của chuỗi đó. Hình 29 cho ta một ví dụ về cây hậu tố của {xabxa, abxbx}.



Hình 2.19: Cây hậu tố với các liên kết hậu tố cho 2 chuỗi xabxa và abxbx

b. Dựng cây hậu tố tổng quát trong thời gian tuyến tính

Áp dụng giải thuật Ukkonen ta dễ dàng dựng được cây hậu tố tổng quát trong thời gian O(N) với N là tổng độ dài các xâu.

Đầu tiên ta dựng cây hậu tố thông thường cho xâu S_1. Với các xâu S_2, S_3,..., S_K trước tiên ta tìm tiền tố dài nhất $S_k[1…i]$ đã tồn tại trong cây. Ta thực hiện các giai đoạn i+1, i+2,..., k của thuật toán Ukkonen để mở rộng cây hậu tố tổng quát phủ toàn bộ xâu.

Đi sâu vào chi tiết, việc tìm tiền tố dài nhất đã có trong cây đồng nghĩa với việc tìm đường đi dài nhất trong cây có nhãn $S_k[1…i]$ bằng cách quét từng kí tự trên đường đi từ gốc. Có hai trường hợp xảy ra:

1. Đường đi kết thúc ở nút v (có thể là nút gốc): thêm nút con mới nối với v bằng cạnh có nhãn là $S_k[i+1]$.

2. Đường đi kết thúc giữa một cạnh: chia đôi cạnh tại điểm đường đi kết thúc và tạo ra nút mới v. Tạo nút con của v nối với nó bằng cạnh $S_k[i+1]$.

Sau khi thực hiện xong bước trên bước mở rộng đầu tiên của giai đoạn i+1 đã hoàn thành, ta có thể đi theo nút cha của v, theo liên kết hậu tố v.v... để

thực hiện các bước mở rộng tiếp theo. Lưu ý rằng trong trường hợp thứ 2 ta cũng cần đảm bảo liên kết hậu tố của v sẽ được thiết lập trong bước mở rộng tiếp theo.

2.5.3 Thuật toán STC

Thuật toán phân nhóm cây hậu tố Suffix Tree Clustering (STC) [10] là một thuật toán phân nhóm thời gian tuyến tính dựa trên việc nhận dạng các nhóm từ chung của các văn bản. Một nhóm từ trong ngữ cảnh này là một chuỗi thứ tự của một hoặc nhiều từ. Chúng tôi định nghĩa một nhóm cơ bản (base nhóm) là một tập các văn bản có chia sẻ một nhóm từ chung.

STC có 3 bước thực hiện logic: (1) "Làm sạch" văn bản, (2) định nghĩa các nhóm cơ bản sử dụng một cây hậu tố và (3) kết hợp các nhóm cơ bản vào các nhóm.

Bước 1: Tiền xử lý (Pro-Precessing). Trong bước này, các chuỗi của đoạn văn bản biểu diễn mỗi tài liệu được chuyển đổi sử dụng các thuật toán chặt (Chẳng hạn như loại bỏ đi các tiền tố, hậu tố, chuyển từ số nhiều thành số ít). Phân ra thành từng câu (xác định các dấu chấm câu, các thẻ HTML). Bỏ qua các từ tố không phải là từ (chẳng hạn như kiểu số, các thẻ HTML và các dấu câu). Các chuỗi tài liệu nguyên gốc được giữ lại, cùng với các con trỏ tại vị trí bắt đầu của mỗi từ trong chuỗi chuyển đổi đến vị trí của nó trong chuỗi gốc. Việc có các con trỏ nhằm giúp hiển thị được đoạn văn bản gốc từ các nhóm từ khóa đã chuyển đổi.

Bước 2: Xác định các nhóm cơ sở. Việc xác định các nhóm cơ sở có thể được xem xét như việc tạo một chỉ số của các nhóm từ cho tập tài liệu. Điều này được thực hiện hiệu quả thông qua việc sử dụng cấu trúc dữ liệu gọi là cây hậu tố. Cấu trúc dữ liệu này có thể được xây dựng trong thời gian tuyến tính với kích cỡ của tập tài liệuvà có thể được xây dựng tăng thêm cho các tài liệu đang được đọc vào. Một cây hậu tố của một chuỗi S là một cây thu gọn chứa đựng tất cả các hậu tố của S. Thuật toán coi các tài liệu như các chuỗi của các từ, không

phải của các ký tự vì vậy các hậu tố chứa đựng một hoặc nhiều từ. Mô tả cụ thể về cây hậu tố như sau:

- Một cây hậu tố là cây có gốc và được định hướng.

- Mỗi node trong có tối thiểu 2 con.

- Mỗi cạnh được gắn nhãn là một chuỗi con của S và chuỗi đó khác rỗng. Nhãn của một node được xác định thông qua chuỗi nối tiếp của các nhãn được gắn cho các cạnh từ gốc tới node đó.

- Không có hai cạnh từ một node được gắn nhãn bắt đầu với từ giống nhau

- Với mỗi hậu tố s của S, tồn tại một suffix-node có nhãn là s.

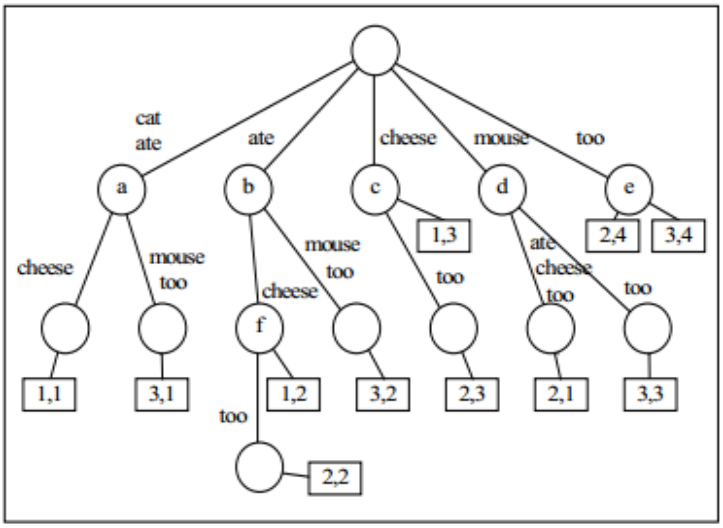
Cây hậu tố của một tập các chuỗi là một cây thu gọn chứa đựng tất cả các hậu tố của tất cả các chuỗi trong tập tài liệu. Mỗi suffix-node được đánh dấu để chỉ ra chuỗi mà nó thuộc về. Nhãn của suffix-node chính là một hậu tố của chuỗi đó. Để phân nhóm ta sẽ xây dựng cây hậu tố của tất cả các câu của tất cả các tài liệu trong tập tài liệu. Chẳng hạn có thể xây dựng cây hậu tố cho tập các chuỗi là

{“cat ate cheese”, “mouse ate cheese too”, “cat ate mouse too”}.

- Các node của cây hậu tố được vẽ bằng hình tròn.

- Mỗi suffix-node có một hoặc nhiều hộp gắn vào nó để chỉ ra chuỗi mà nó thuộc về.

- Mỗi hộp có 2 số (số thứ nhất chỉ ra chuỗi mà hậu tố thuộc về, số thứ hai chỉ ra hậu tố nào của chuỗi gắn nhãn cho suffix-node)



Hình 2.20: Cây hậu tố của các chuỗi "cat ate cheese", "mouse ate cheese too" and "cat ate mouse too"

Một số node đặc biệt a → f. Mỗi một node này biểu diễn cho một nhóm tài liệu và một nhóm từ chung được thiết đặt cho tất cả tài liệu. Nhãn của node biểu diễn nhóm từ chung. Tập các tài liệu gắn nhãn suffix-node là kế thừa của các node tạo bởi nhóm tài liệu. Do đó, mỗi node biểu diễn một nhóm cơ sở (base nhóm).

Ngoài ra, tất cả các nhóm cơ sở có thể (chứa 2 hoặc nhiều tài liệu) xuất hiện như các node trong cây hậu tố. Bảng sau liệt kê các node a → f trong hình 30 và các nhóm cơ sở tương ứng.

Node	Phrase	Docum
A	cat ate	1, 3
B	Eat	1, 2, 3
C	Cheese	1, 2
D	Mouse	2, 3
E	Too	2, 3
F	ate	1, 2

Bảng 2.1: Các nhóm cơ sở tương ứng

Mỗi nhóm cơ sở được gán một điểm số là một hàm của số lượng các tài liệu nhóm đó chứa đựngvà các từ hình thành nên nhóm từ của nó. Điểm số s(B) của nhóm cơ sở B với nhóm từ P là: s(B) = |B| . f(|P|) (*)

Trong đó: |B| là số lượng của các tài liệu trong nhóm cơ sở B, |P| là số lượng các từ có trong nhóm từ P mà có điểm số khác 0. Việc xét đến điểm số của nhóm từ P theo nghĩa như sau:

Thuật toán cài đặt một danh sách stoplist bao gồm các từ đặc trưng trên Internet dùng để xác định các từ khác. Các từ xuất hiện trong danh sách stoplist đó hay các từ xuất hiện quá ít trong một nhóm từ (3 hoặc ít hơn) hay quá nhiều (hơn 40% của tập tài liệu) sẽ được gán điểm số 0 cho nhóm từ.

Hàm f trong công thức (*) thực hiện trên các nhóm từ đơn, nó là tuyến tính cho các nhóm từ có độ dài từ 2 đến 6 và là hằng số với các nhóm có độ dài lớn hơn.

Bước 3: Kết nối các nhóm cơ sở

Các tài liệu có thể chia sẻ nhiều hơn một nhóm từ. Kết quả là, tập hợp tài liệu của các nhóm cơ sở khác nhau có thể trùng lặp và thậm chí là có thể là giống nhau. Để tránh việc có nhiều các nhóm gần giống nhau tại bước thứ 3 này của thuật toán việc trộn các nhóm cơ sở với một sự trùng lặp cao trong tập tài liệu của chúng (chú ý là các nhóm từ chung không xem xét trong bước này). Thuật toán đưa ra một độ đo tính tương tự giữa các nhóm dựa trên việc trùng lặp của tập tài liệu của chúng. Giả sử có hai nhóm cơ sở Bm và Bn với kích cỡ là |Bm| và |Bn| tương ứng. Với |Bm ∩ Bn| thể hiện số tài liệu chung của cả hai nhóm, độ tương tự giữa Bm và Bn là 1 nếu:

+) | Bm ∩ Bn| / |Bm| > 0.5 và

+) | Bm ∩ Bn| / |Bn| > 0.5

Ngược lại, độ tương tự là 0.

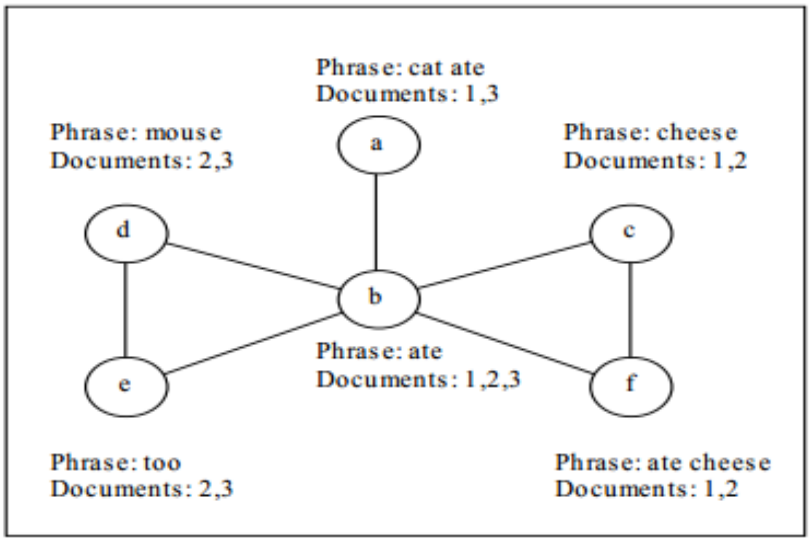
Hãy xem minh họa tiếp theo của ví dụ trong Hình 3.21:

- Mỗi node là các nhóm cơ sở.

- Hai node được nối với nhau khi độ tương tự là 1.

- Một nhóm được xác định là các thành phần được ghép nối trong đồ thị nhóm cơ sở.

- Mỗi một nhóm sẽ bao gồm tập của tất cả các tài liệu của các nhóm cơ sở trong nó.



Hình 2.21: Đồ thị các nhóm cơ sở

Xét trong ví dụ trên. Các thông số được thể hiện như hình trên. Mỗi nút là một cụm và mỗi cạnh nối với nhau thể hiện rằng độ tương tự giữa 2 cụm là lớn hơn 1 tức là các cụm có tồn tại một cạnh nối có thể hợp lại với nhau thành một cụm. như vậy sơ đồ trên thể hiện duy nhất một cụm.

Xét trường hợp, nếu giả sử rằng từ 'ate' có trong danh sách stoplist, thì nhóm cơ sở b sẻ bị loại ra bởi vì nó có chỉ số của nhóm từ là 0. Và do đó sẽ có 3 thành phần kết nối trong đồ thị, thể hiện 3 nhóm.

Hiệu quả thuật toán: Thời gian của việc tiền xử lý các tài liệu tại bước 1 của thuật toán STC hiển nhiên là tuyến tính với kích thước tập tài liệu. Thời gian của việc thêm các tài liệu vào cây hậu tố cũng tuyến tính với kích thước tập tài liệu theo thuật toán Ukkonen cũng như số lượng các node có thể bị ảnh hưởng bởi việc chèn này. Do vậy thời gian tổng cộng của STC tuyến tính với kích thước tập tài liệu. Hay thời gian thực hiện của thuật toán STC là O(n) trong đó n là kích thước của tập tài liệu.

Suffix tree clustering (STC) hay phân nhóm cây hậu tố là phương pháp phân nhóm kết quả trả về từ máy tìm kiếm được Carrot2 sử dụng với ưu thế về sự chính xác và nhanh chóng khi phân nhóm trên dữ liệu nhỏ (snippet). Michal Wroblewski [8] đã phân tích, so sánh kết quả bộ phân nhóm trả về của Carrot2 khi sử dụng các thuật toán khác nhau, kết quả thuật toán cây hậu tố đưa lại bộ phân nhóm có độ chính xác cao nhất.

Nhận xét rằng, thuật toán phân nhóm STC cho kết quả tương đối chính xác và thời gia xử lý khá nhanh (bởi độ phức tạp thuật toán O(n)). Điều dễ nhận thấy trong kết quả trên là thuật toán này là phân nhóm chồng lặp, điều này phù hợp với các tài liệu trong lĩnh vực Web bởi mỗi tài liệu bất kỳ sẽ có xu hướng thuộc một hoặc nhiều chủ đề khác nhau.

2.6 Thuật toán dựa vào phân loại véc-tơ hỗ trợ

2.6.1 Phương pháp SVM

Xét bài toán: Kiểm tra xem một tài liệu bất kỳ d thuộc hay không thuộc một phân nhóm c cho trước? Nếu d ∈ c thì d được gán nhãn là 1, ngược lại thì d được gán nhãn là -1.

Giả sử, lựa chọn được tập đặc trưng là T = {t_1,t_2, …, t_n}, thì mỗi tài liệu di sẽ được biểu diễn bằng một véc-tơ dữ liệu xi = (w_{i1}, w_{i2}, …, w_{in}), $w_{ij} \in R$ là trọng số từ tj trong dữ liệu di. Như vậy, tọa độ của mỗi véc-tơ dữ liệu xi tương ứng với tọa độ của một điểm trong không gian Rn.

Dữ liệu huấn luyện cúa SVM là tập các văn bản đã được gán nhãn trước:

D = {(x_1,y_1), (x_2, y_2), …, (x_i, y_i), trong đó, x_i là véc-tơ dữ liệu biểu diễn dữ liệu di ($x_i \in R_n$), $y_i \in \{+1,-1\}$, cặp (x_i, y_i) được hiểu là véc-tơ xi được gán nhãn là yi. Ý tưởng của SVM là tìm một mặt hình học (siêu phẳng) f(x) "tốt nhất" trong không gian n-chiều để phân chia dữ liệu sao cho các điểm x+ được gán nhãn 1 thuộc về phía dương của siêu phẳng (f(x+) > 0), các điểm x- được gán nhãn -1 thuộc về phía âm của siêu phẳng (f(x-) < 0). Với bài toán phân loại SVM, một siêu phẳng phân chia dữ liệu được gọi là "tốt nhất", nếu khoảng cách từ điểm dữ liệu gần nhất đến siêu phẳng là lớn nhất. Khi đó, việc xác định một tài liệu x □ D có thuộc phân loại c hay không, tương ứng với việc xét dấu của f(x), nếu f(x) > 0 thì x ∈ c, nếu f(x) ≤ 0 thì x ∉ c.

Cho tập dữ liệu: D ={(x_1,y_1), (x_2,y_2), …(x_i,y_i)}, $x_i \in R_n$, $y_i \in \{-1,1\}$

Trường hợp 1: Tập dữ liệu D có thể phân chia tuyến tính được mà không có nhiễu thì chúng ta có thể tìm được một siêu phẳng tuyến tính có dạng (1) để phân chia tập dữ liệu này. Siêu phẳng tốt nhất tương đương với việc giả bài toán tối ưu sau:

$$\begin{cases} Min\phi(w) = \frac{1}{2}\|w\|^2 \\ y_j(w^T x_i + b) \geq 1, i = 1 \end{cases} \quad (1)$$

Trường hợp 2: Tập dữ liệu huấn luyện D có thể phân chia được tuyến tính nhưng có nhiễu nghĩa là điểm có nhãn dương nhưng lại thuộc về phía âm của siêu phẳng, điểm có nhãn âm thuộc về phía dương của siêu phẳng. Bài toán (1) trở thành:

$$\begin{cases} \text{Min}\phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_j(w^T(x_i) + b) \geq 1 - \xi, i = 1,...,l \\ \xi_i \geq 0, i = 1,...,l \end{cases} \quad (2)$$

Với ξi gọi là biến số nới lỏng (slack variable)) $\xi_i \geq 0$.

C là tham số xác định trước, định nghĩa giá trị ràng buộc, C càng lớn thì mức độ vi phạm đối với những lỗi thực nghiệm càng cao.

Trường hợp 3: Tuy nhiên không phải tập dữ liệu nào cũng có thể phân chia tuyến tính được. Trong trường hợp này, chúng ta sẽ ánh xạ các véc-tơ dữ liệu x từ không gian n-chiều vào một không gian m-chiều (m > n) , sao cho trong không gian m-chiều này tập dữ liệu có thể phân chia tuyến tính từ không gian R^n vào không gian R^m. $\phi: R^n \rightarrow R^m$

Khi đó, véc-tơ xi trong không gian Rn sẽ tương ứng với véc-tơ ϕ (xi) trong không gian Rm.

Thay ϕ (xi) vào (2) ta có (3):

$$\begin{cases} Min\phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_j(w^T.\phi(x_i) + b) \geq 1 - \xi, i = 1,...,l \\ \xi_i \geq 0, i = 1,...,l \end{cases}$$

$$\begin{cases} Min\phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_j(w^T.\phi(x_i) + b) \geq 1 - \xi, i = 1,...,l \\ \xi_i \geq 0, i = 1,...,l \end{cases} \quad (3)$$

Việc tính toán trực tiếp ϕ (xi) là phức tạp và khó khăn. Nếu biết hàm nhân (Kernel function) $K(x_i,y_i)$, để tính tích vô hướng ϕ (x_i). ϕ (x_j) trong không gian m-chiều, thì chúng ta không cần làm việc trực tiếp với ánh xạ ϕ (x_i).

$$K(x_i, x_j) = \phi(x_i).\ \phi(x_j) \quad (4)$$

Một số hàm nhân hay dùng trong phân loại văn bản là :

Hàm tuyến tính (linear): $K(x_i, x_j) = x_i^T x_j$ (5)

Hàm đa thức (polynomial function): $K(x_i, x_j) = (x_i x_j + 1)^d$ (6)

Hàm RBF (radial basis function): $K(x_i, y_j) = \exp(-\gamma(x_i - x_j)^2)$ $\gamma \in R^+$ (7)

2.6.2 Phương pháp FSVM

Trong SVM thông thường thì các điểm dữ liệu đều có giá trị như nhau, mỗi một điểm sẽ thuộc hoàn toàn vào một trong hai lớp. Tuy nhiên trong nhiều trường hợp có một vài điểm sẽ không thuộc chính xác một lớp nào đó, những điểm này được gọi là những điểm nhiễu, hơn nữa mỗi điểm dữ liệu có thể sẽ không có ý nghĩa như nhau đối với siêu phẳng. Để giải quyết vấn đề này Lin CF. và Wang SD (2002) đã giới thiệu phương pháp FSVM bằng cách sử dụng một hàm thành viên để xác định giá trị đóng góp của mỗi điểm dữ liệu đầu vào của SVM vào việc hình thành siêu phẳng.

Bài toán được mô tả như sau:

$$\begin{cases} Min\phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} S_i\xi_i \\ y_j(w^T.\phi(x_i) + b) \geq 1 - \xi, i = 1,\ldots,l \\ \xi_i \geq 0, i = 1,\ldots,l \end{cases} \quad (8)$$

S_i là một thành viên thỏa $\sigma \leq s_i \leq 1$, σ là một hằng số đủ nhỏ > 0 thể hiện mức độ ảnh hưởng của điểm x_i đối với một lớp. Giá trị s_i có thể làm giảm giá trị của biến ξ_i, vì vậy điểm x_i tương ứng với ξ_i có thể giảm được mức độ ảnh hưởng hơn.

$$\begin{cases} \max L_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \sum_{i=1}^{l} \alpha_i y_j = 0 \\ 0 \le \alpha_i \le s_i C, i = 1,...,l \end{cases} \quad (9)$$

Chọn hàm thành viên: Việc chọn hàm thành viên S_i thích hợp rất quan trọng trong FSVM. Theo Chun hàm thành viên si dùng để giảm mức độ ảnh hưởng của những điểm dữ liệu nhiễu được mô tả trong [10] là một hàm xác định khoảng cách giữa điểm dữ liệu xi với trung tâm của nhóm ứng với x_i.

Gọi C^+= là tập chứa điểm x_i với y_i=1

$C^+ = \{x_i | x_i \in S$ và $y_i = 1\}$

Tương tự gọi $C^- = \{x_i | x_i \in S$ và $y_i = 1\}$ X+ và X- là trung tâm của lớp C^+, C^-. Bán kính của lớp C^+

$$r_+ = \max||X_+ - x_i|| \text{ với } x_i \in C^+ \quad (10)$$

Và bán kính của lớp C^- là:

$$r_- = \max||X_- - x_i|| \text{ với } x_i \in C^- \quad (11)$$

Hàm thành viên S_i được định nghĩa như sau:

$$s_i = \begin{cases} 1 - \dfrac{\|x_+ - x_i\|}{r_- + x_i} \\ 1 - \dfrac{\|x_- - x_i\|}{r_- - x_i} \end{cases} \quad (12)$$

δ là hằng số để tránh trường hợp $S_i = 0$.

Tuy nhiên FSVM với hàm thành viên (12) vẫn chưa đạt kết quả tốt do việc tính toán khoảng cách giữa các điểm dữ liệu với trung tâm của nhóm được tiến hành ở không gian đầu vào, không gian n chiều. Trong khi đó trường hợp tập dữ liệu không thể phân chia tuyến tính, để hình thành siêu phẳng ta phải đưa dữ liệu về một không gian khác với số chiều m cao hơn gọi là không gian đặc trưng (feature space). Vì vậy để có thể đạt kết quả tốt hơn, Xiufeng Jiang, Zhang Yi và Jian Cheng Lv (2006) đã xây dựng một hàm

thành viên khác dựa trên ý tưởng của hàm thành viên (12) nhưng được tính toán trong không đặc trưng m chiều.

Giả sử ϕ là một ánh xạ phi tuyến tính từ không gian R^n vào không gian R^m.

ϕ: $R^n \to R^m$

Khi đó, véc-tơ x_i trong không gian R^n sẽ tương ứng véc-tơ ϕ (x_i) trong không gian R^m.

Định nghĩa ϕ+ là trung tâm của lớp C^+ trong không gian đặc trưng:

$$\phi_+ = \frac{1}{n_+}\sum_{x_i \in C} + \phi(x_i) \tag{13}$$

n^+ là số phần tử của lớp C^+

$$\phi_- = \frac{1}{n_-}\sum_{x_i \in C} - \phi(x_i) \tag{14}$$

và ϕ - là trung tâm của lớp C^- trong không gian đặc trưng:

n_- là số phân tử lớp C^-

Định nghĩa bán kính của C^+:

$$r_+ = \max\|\phi_+ - \phi(x_i)\| \quad \text{với} \quad x_i \in C^+ \tag{15}$$

Và bán kính của C^-:

$$r_- = \max\|\phi_- - \phi(x_i)\| \quad \text{với} \quad x_i \in C^-$$

Khi đó:

$$\begin{aligned} r_+^2 &= \max\|\phi_+ - \phi(x_i)\|^2 \\ &= \max\left\{\phi^2(x') - 2\phi(x')\phi_+ + \phi_+^2\right\} \\ &= \max\left\{\phi^2(x') - \frac{2}{n_+}\sum_{x_i \in C^+}\phi(x_i)\phi(x') + \frac{1}{n_+^2}\sum_{x_i \in C^+}\sum_{x_j \in C^+}\phi(x_i)\phi(x_j)\right\} \end{aligned} \tag{16}$$

$$r_+^2 = \max\left\{K(x', x') - \frac{2}{n_+}\sum x_i \in C^+ K(x_i, x') + \frac{1}{n_+^2}\sum x_i \in C^+ \sum x_j \in K(x_i, x_j)\right\} \tag{17}$$

Với x' $\in C^+$ và n_+ là mẫu số huấn luyện trong lớp C^+. Tương tự:

$$r_-^2 = \max\left\{K(x',x') - \frac{2}{n_-}\sum x_i \in C^+ K(x_i,x') + \frac{1}{n_-^2}\sum x_i \in C^- \sum x_j \in K(x_i,x_j)\right\} \quad (18)$$

Với x'$\in C^-$ và n_- là mẫu số huấn luyện trong lớp C^-.

Bình phương khoảng cách giữa $x_i \in C^+$ và trung tâm của lớp trong không gian đặc trưng có thể được tính như sau:

$$d_{i+}^2 = K(x_i,x_i) - \frac{2}{n_+}\sum x_i \in C^+ K(x_i,x_j) + \frac{1}{n_+^2}\sum x_j \in C^+ \sum x_k \in C^+ K(x_i,x_k) \quad (19)$$

Tương tự như vậy bình phương khoảng cách giữa $x_i \in C^-$ và trung tâm của lớp trong không gian đặc trưng có thể tính như sau:

$$d_{i-}^2 = K(x_i,x_i) - \frac{2}{n_-}\sum x_i \in C^- K(x_i,x_j) + \frac{1}{n_-^2}\sum x_j \in C^- \sum x_k \in C^- K(x_i,x_k) \quad (20)$$

Với mỗi i (i=1,..., l), hàm thành viên S_i được mô tả như sau:

$$S_i = \begin{cases} 1 - \sqrt{\left\|d_{i+}^2\right\| / (r_+^2 + \delta)} \\ 1 - \sqrt{\left\|d_{i-}^2\right\| / (r_-^2 + \delta)} \end{cases} \quad (21)$$

Ta thấy S_i là một hàm của trung tâm và bán kính của mỗi lớp trong không gian đặc trưng.

Kết luận chương 2: Chương 2 tìm hiểu về một số thuật toán điển hình trong phân cụm dữ liệu. Trong đó có một số thuật toán phân cụm dữ liệu tĩnh và phân cụm dữ liệu động. Tùy từng ứng dụng và căn cứ vào ưu nhược điểm của từng thuật toán ta có thể tìm được các thuật toán phù hợp để áp dụng cho bài toán thực tế.

CHƯƠNG III:

ỨNG DỤNG PHƯƠNG PHÁP PHÂN NHÓM DỮ LIỆU VÀO PHÂN TÍCH LƯƠNG CỦA CÁN BỘ TRƯỜNG CAO ĐẲNG NGHỀ HÀ NAM

3.1 Đặt vấn đề

Để đánh giá phân tích dữ liệu lương, trong nội dung của luận văn có sử dụng một phần dữ liệu lương của trường Cao đẳng Nghề Hà Nam.

Thuật toán áp dụng: Áp dụng thuật toán DBSCAN ở chương II.

Thuật toán DBSCAN bao gồm các bước sau:

INPUT: - Tập các đối tượng dữ liệu $X = \{x_1, x_2, \ldots, x_n\}$

- Ngưỡng đến được giữa 2 đối tượng: Eps

- Số lượng các đối tượng tối thiểu trong nhóm: MinPts

OUTPUT: Các nhóm C_i (i = 1, .., k)

Bước 1: Chọn một đối tượng p tùy ý.

Bước 2: Lấy tất cả các đối tượng có mật độ đến được từ p với Eps và MintPts

Bước 3: Nếu p là điểm nhân thì tạo ra một nhóm theo Eps và MintPts.

Bước 4: Nếu p là điểm biên, không có điểm nào có mật độ đến được từ p và DBSCAN sẽ đi thăm điểm tiếp theo của tập dữ liệu.

Bước 5: Quá trình tiếp tục cho đến khi tất cả các đối tượng được xử lý.

Chương trình được xây dựng với mục đích thử nghiệm thuật toán DBSCAN đề ra ở chương 2 nên chương trình cần phải thực hiện các yêu cầu sau:

- Quản lý thông tin người dùng : thông tin về người dùng sử dụng hệ thống bao gồm tài khoản và mật khẩu truy cập vào hệ thống, mật khẩu cấp 2 dùng để lấy lại mật khẩu cấp 1 khi quên mật khẩu, chú thích.
- Quản lý thông tin Khoa/Viện : thông tin về Khoa/Viện mà hệ thống quản lý bao gồm : tên Khoa/Viện, chú thích.

- Quản lý thông tin giảng viên : thông tin về giảng viên thuộc các Khoa/Viện bao gồm họ và tên, giới tính, ngày sinh, quê quán, ảnh, chú thích, tên Khoa/Viện.
- Quản lý thông tin lương : thông tin về lương của giảng viên bao gồm tên giảng viên, lương cứng, phụ cấp, truy thu, tháng lĩnh lương.
- Phân cụm toàn bộ dữ liệu lương : sử dụng thuật toán Incremental DBSCAN để phân cụm toàn bộ dữ liệu lương của hệ thống.
- Phân cụm dữ liệu trong khoảng thời gian: sử dụng thuật toán Incremental DBSCAN để phân cụm các dữ liệu trong khoảng thời gian người dùng chọn.
- Phân cụm dữ liệu lương cán bộ thuộc Khoa/Viện : sử dụng thuật toán Incremental DBSCAN để phân cụm các dữ liệu lương của cán bộ thuộc Khoa/Viện do người dùng chọn.

Thông tin phân cụm gồm có: Tổng số cụm, số mẫu, số nhiễu, số phần tử trong cụm, phần trăm, biểu đồ các cụm, thông tin các phần tử trong từng cụm.

Input: Dữ liệu lương của cán bộ trường Cao đẳng Nghề Hà Nam.

Output: Các cụm dữ liệu đã được phân cụm với các thông tin: Tổng số cụm, số mẫu, số nhiễu, số phần tử trong cụm, phần trăm, biểu đồ các cụm, thông tin các phần tử trong từng cụm.

3.2 Giải quyết vấn đề:

3.2.1 Công cụ lựa chọn xây dựng chương trình phần mềm :

Dưới đây là công cụ được sử dụng để phát triển phần mềm trong hệ thống phân tích lương cán bộ :

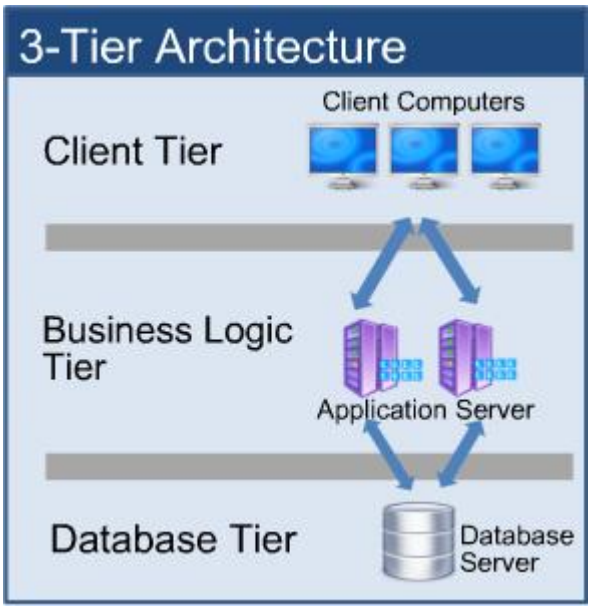
Ngôn ngữ lập trình : C#.

Thư viện hỗ trợ : Devexpress 14.1.6.

IDE : Visual studio 2012.

Database : SQL Server 2008 R2.

Mô hình code : Mô hình 3-Tier

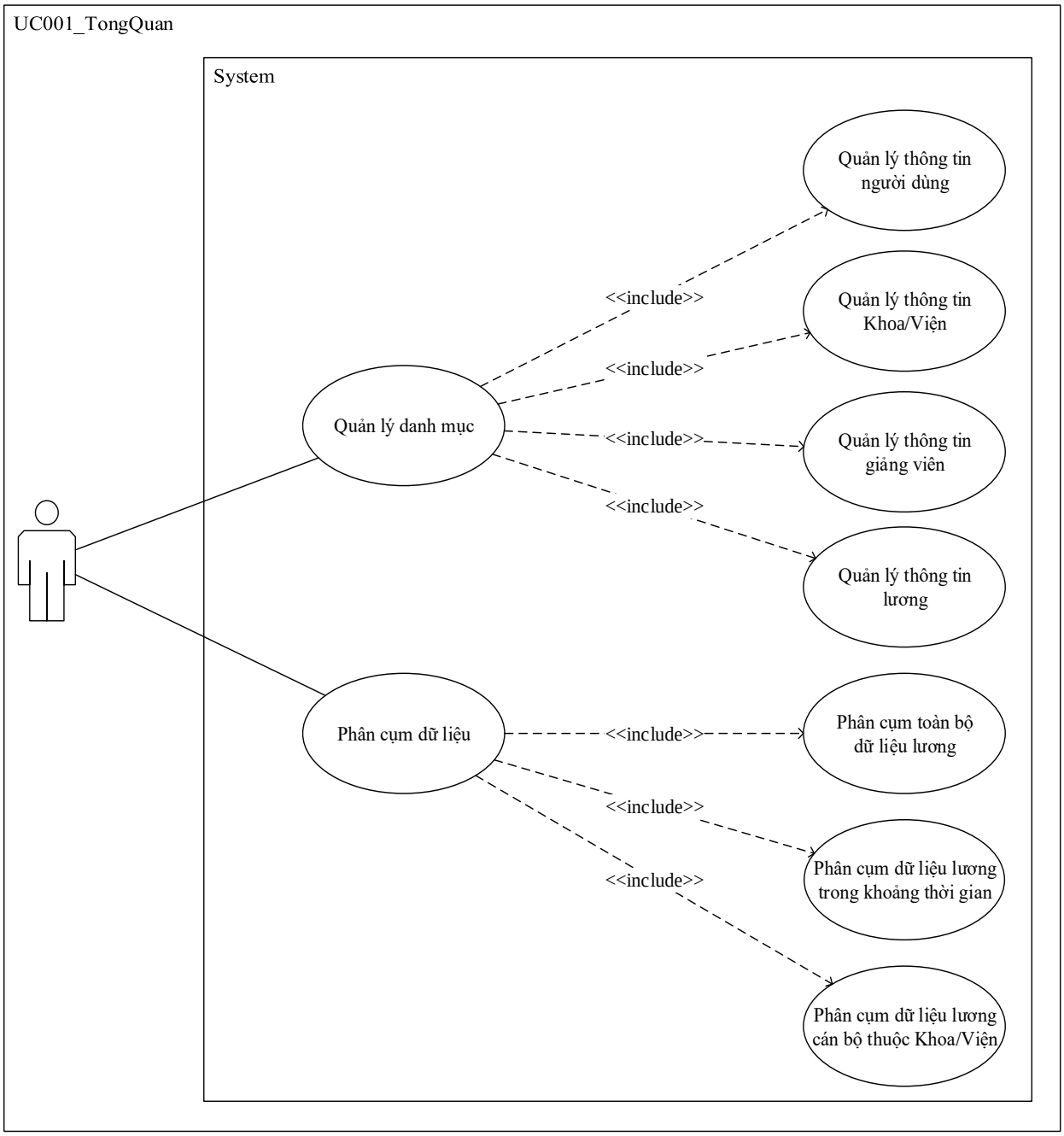


Hình 3.1: Mô hình 3-Tier.

3.2.2. Biểu đồ phân cấp chức năng

Từ những chức năng ta xác định trong chương I.Ta xây dựng biểu đồ phân cấp chức năng hệ thống.

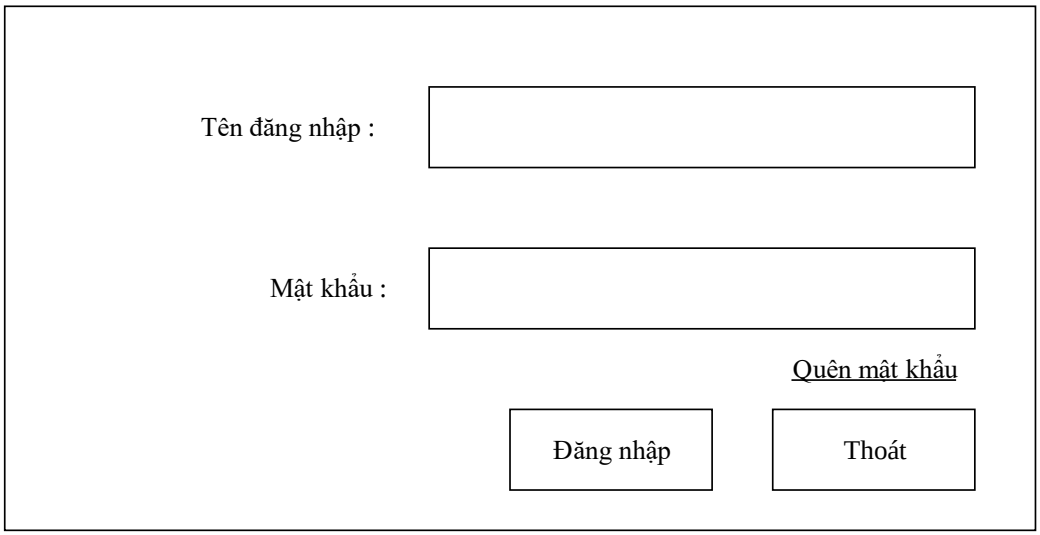
3.2.3 Mô hình tổng quan hệ thống



Hình 3.2: Mô hình use case tổng quan hệ thống.

3.2.4 Thiết kế giao diện chương trình:

3.2.4.1. Giao diện form đăng nhập:

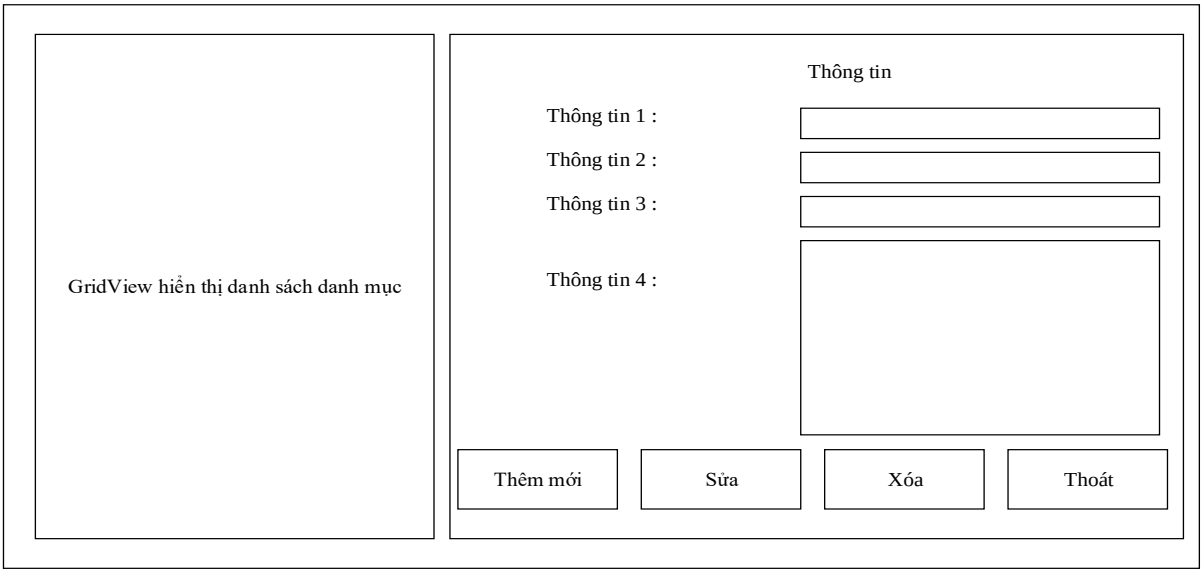


Hình 3.3: Giao diện form đăng nhập

- Người dùng sẽ nhập tài khoản truy cập và mật khẩu vào 2 textbox tên đăng nhập và mật khẩu.
- Sau khi nhập tài khoản truy cập và mật khẩu người dùng bấm nút đăng nhập để vào hệ thống.Hoặc bấm nút thoát để thoát khỏi chương trình.
- Bấm nút quên mật khẩu để lấy lại mật khẩu.

3.2.4.2. Giao diện form quản lý danh mục:

- Bố cục chung của các form quản lý danh mục : quản lý thông tin người dùng, quản lý khoa/viện, quản lý giảng viên, quản lý lương đều có bố cục như hình dưới.Bao gồm bên tay trái là bảng dữ liệu hiển thị tất cả dữ liệu danh mục.Bên tay trái bao gồm các thông tin nhập liệu và hiển thị chi tiết danh mục và các nút chức năng thêm mới, sửa, xóa danh mục.



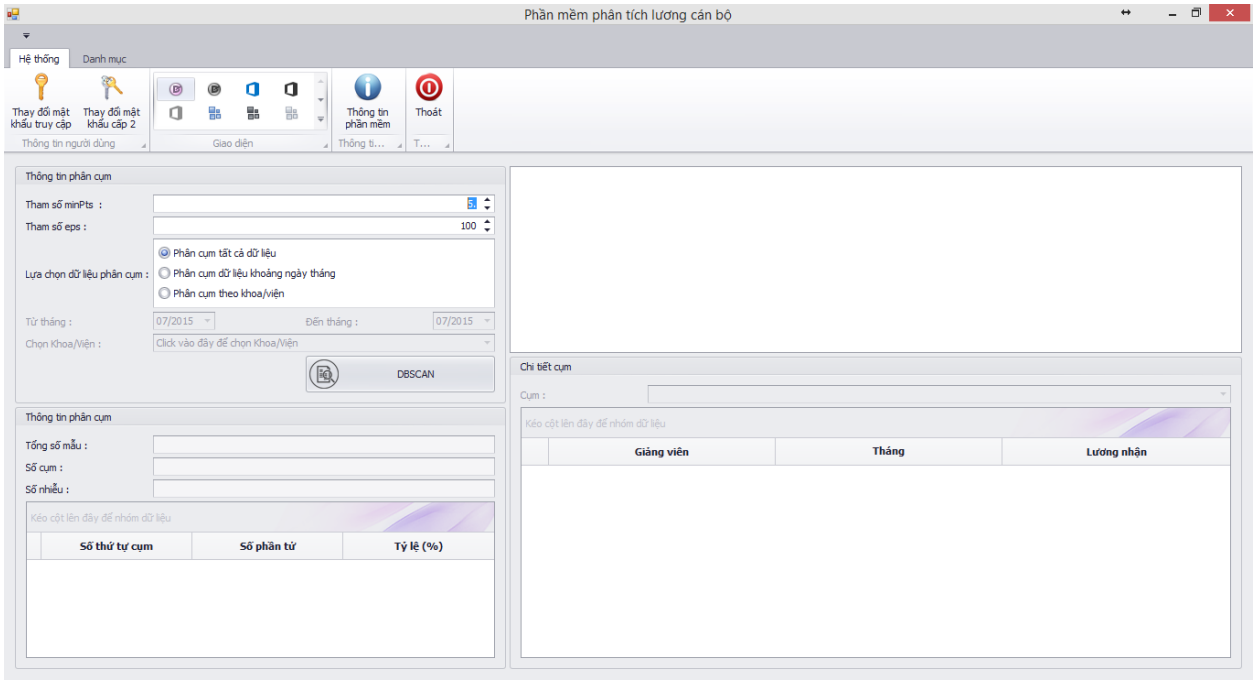
Hình 3.4: Giao diện form quản lý danh mục

3.2.4.3. Giao diện chương trình chính:

- Phía trên bao gồm các menu chức năng chính của chương trình : đổi mật khẩu truy cập, quản lý người dùng, quản lý giảng viên, …
- Phía dưới chia 4 ô : từ trái qua phải ô thứ nhất là ô chọn tham số cho phân cụm, ô thứ hai là biểu đồ, ô thứ 3 là thông số các cụm, ô thứ 4 là chi tiết các cụm.

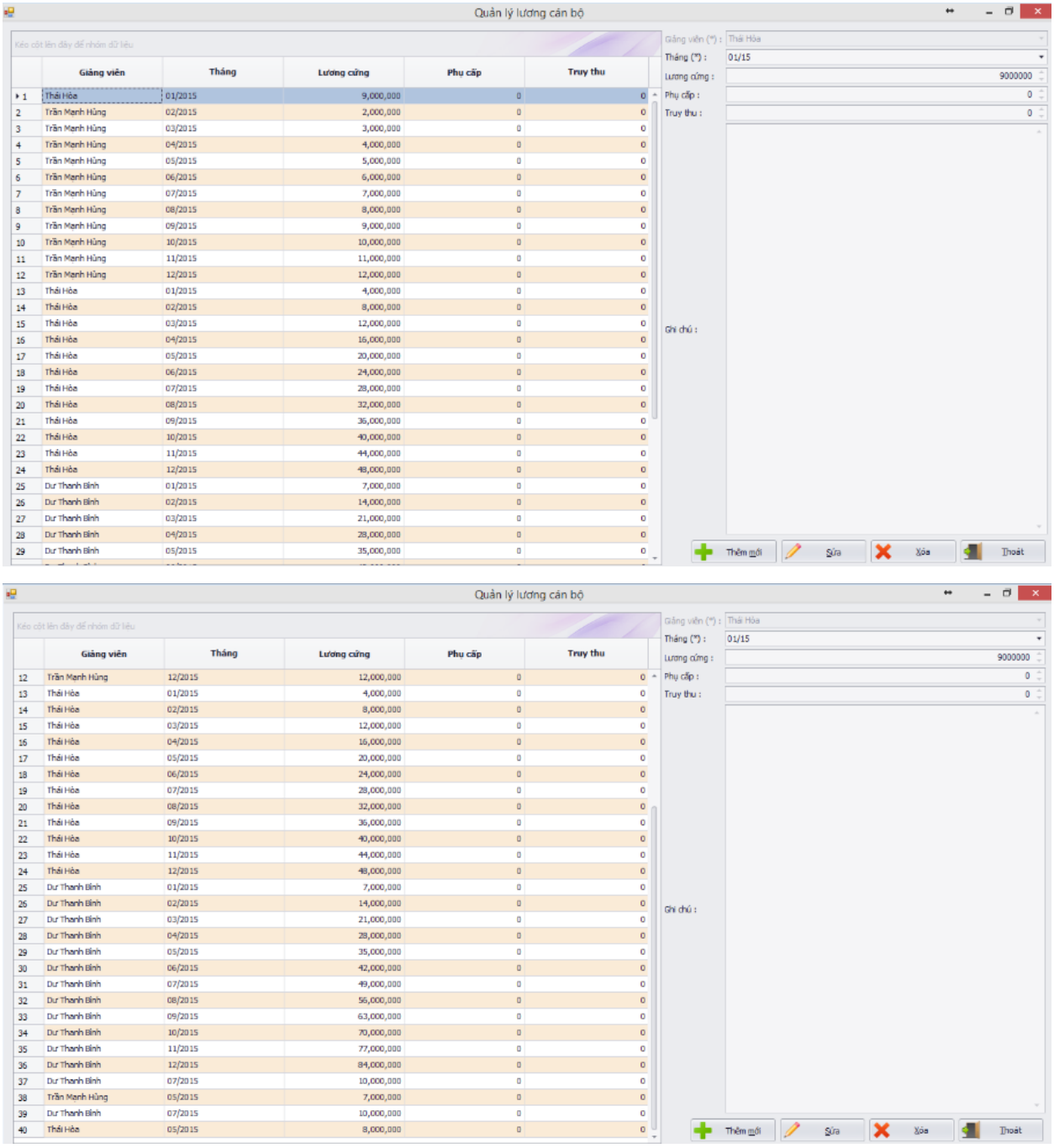
3.2.5 Chạy chương trình :

Đăng nhập thành công màn hình chính hiện lên :



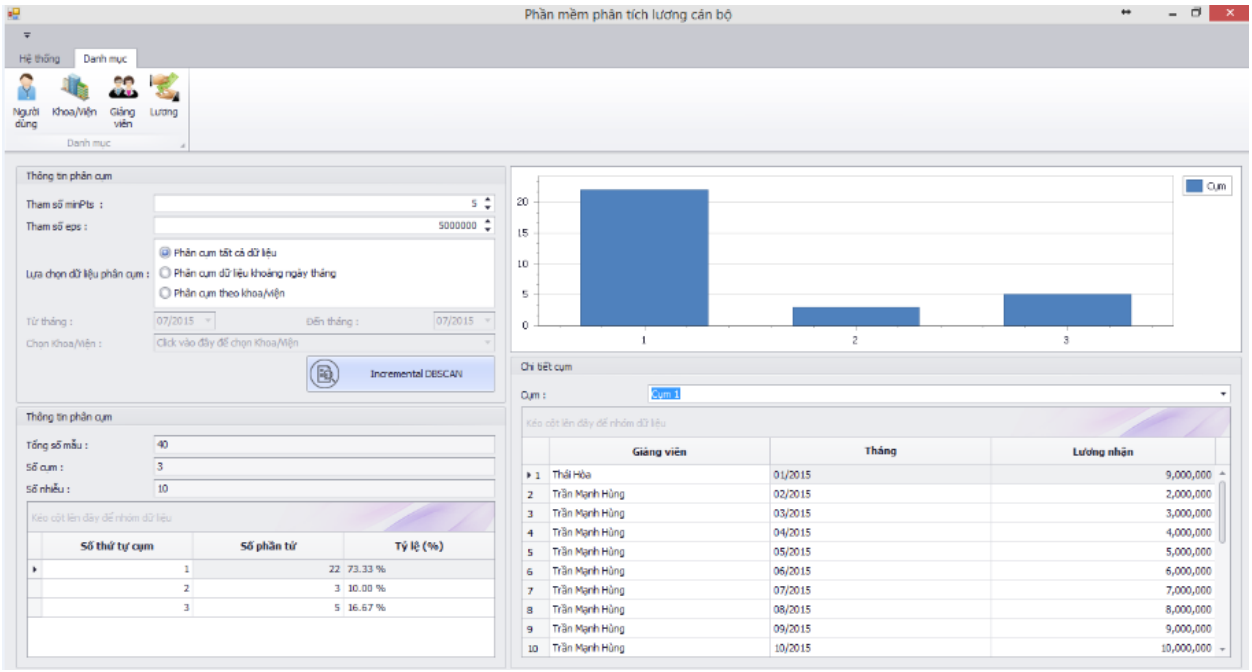
Hình 3.5: Màn hình chính

Tiến hành phân cụm với dữ liệu đầu vào như sau:



Hình 3.6: Dữ liệu đầu vào

Tiến hành phân cụm toàn bộ dữ liệu:



Hình 3.7: Kết quả phân cụm dữ liệu bởi Incremencal DBSCAN

Kết quả sau khi phân cụm như sau:

Tổng số mẫu: 40

Số cụm: 3

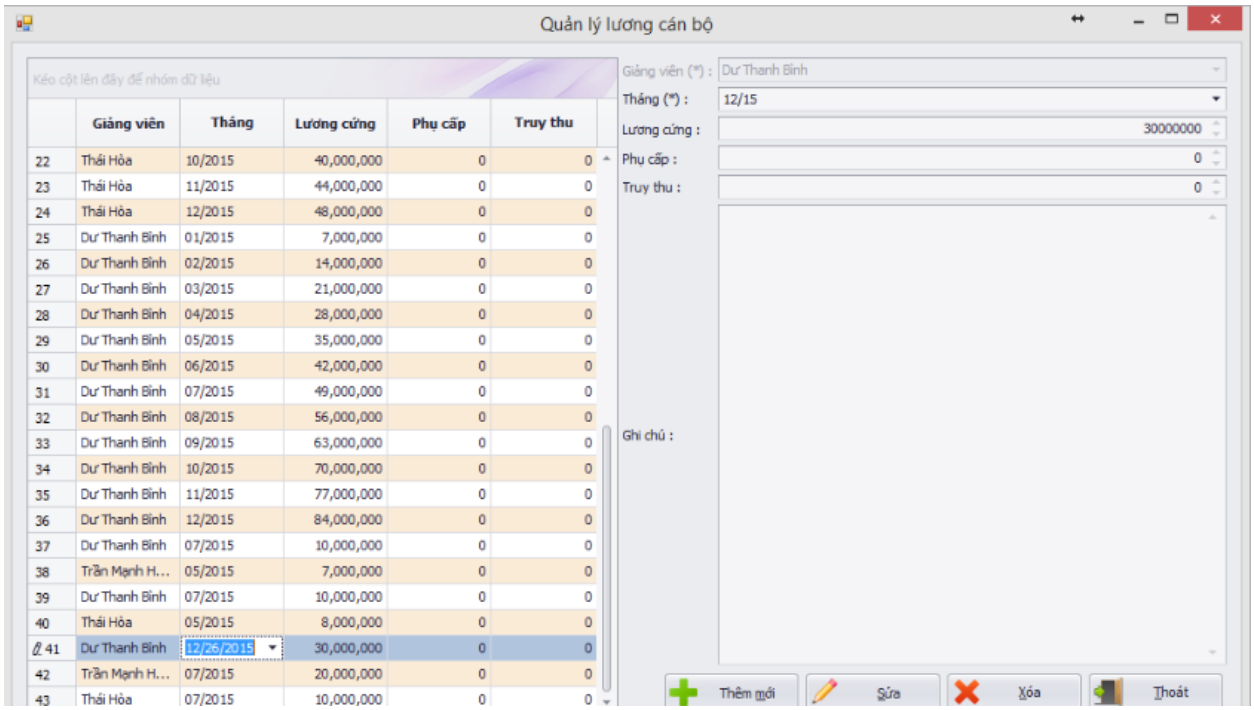
Số nhiễu: 10

Cụm 1: 22 phần tử chiếm 73.33%

Cụm 2: 3 phần tử chiếm 10%

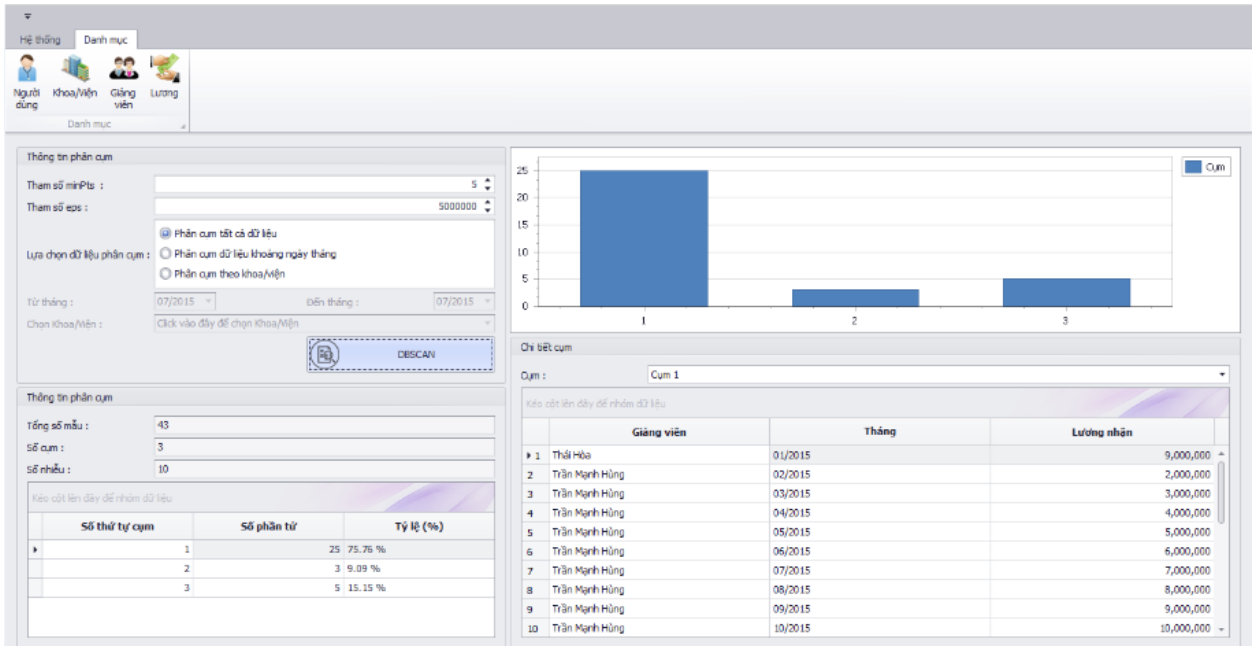
Cụm 3: 5 phần tử chiếm 16.67%

Trường hợp dữ liệu thêm mới:



Hình 3.8: Dữ liệu được thêm mới

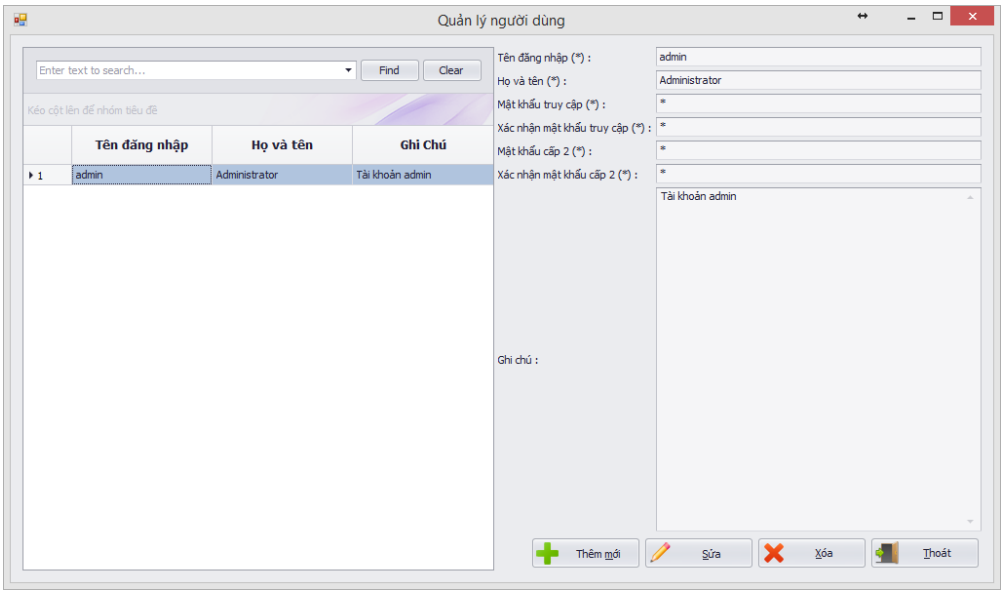
Sau khi thêm mới dữ liệu, kết quả phân cụm mới:



Hình 3.9: Kết quả phân cụm sau khi thêm dữ liệu mới

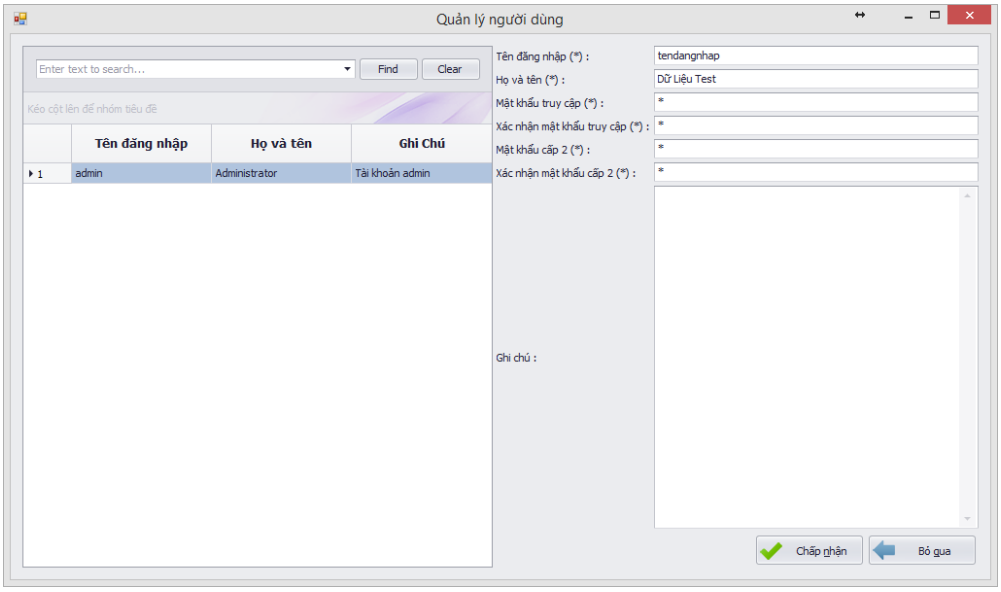
Tổng số mẫu: 43
Số cụm: 3
Số nhiễu: 10
Cụm 1: 25 phần tử chiếm 75.76%
Cụm 2: 3 phần tử chiếm 9.09%
Cụm 3: 5 phần tử chiếm 15.15%

3.2.6 Giao diện quản lý người dùng :



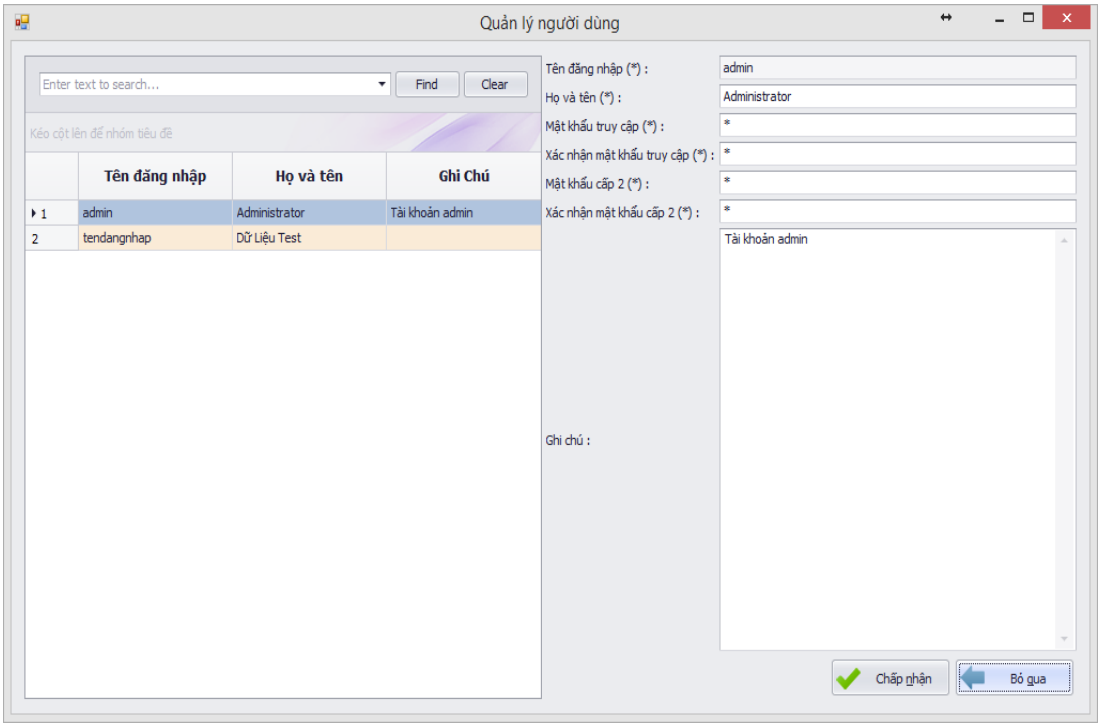
Hình 3.10: Màn hình quản lý người dùng

Để thêm mới người dùng ta click vào button thêm mới. Nhập dữ liệu và bấm nút chấp nhận để thêm mới dữ liệu và ấn bỏ qua để không thêm mới.



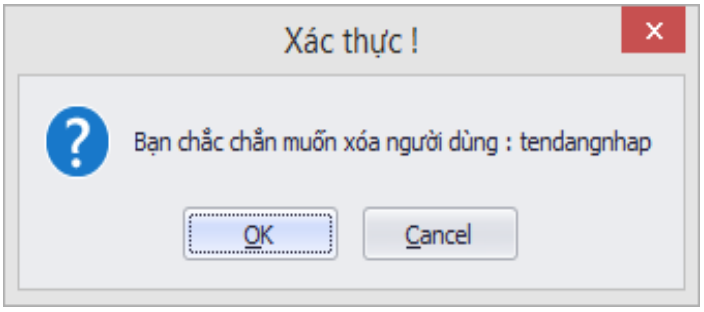
Hình 3.11: Màn hình thêm mới người dùng

Để sửa đổi dữ liệu ta click vào dữ liệu muốn sửa tại bảng dữ liệu. Rồi bấm nút sửa và nhập các thông tin sửa đổi vào các ô nhập dữ liệu. Bấm nút chấp nhận để thêm mới dữ liệu và ấn bỏ qua để bỏ qua bước sửa đổi.



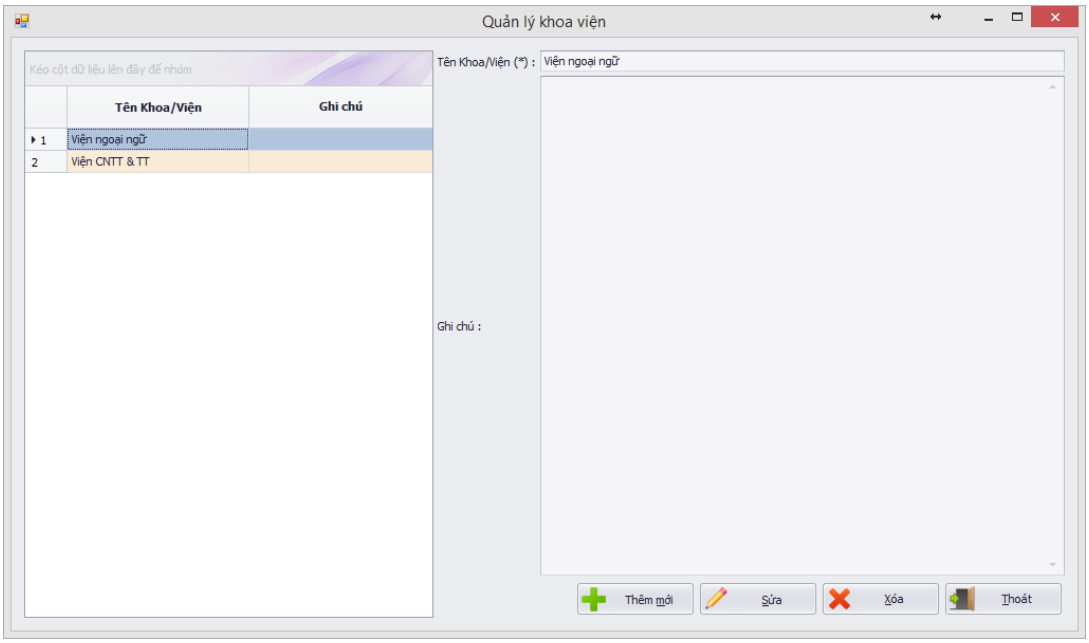
Hình 3.12: Màn hình sửa thông tin người dùng

Để xóa thông tin người dùng ta click vào dữ liệu muốn xóa tại bảng dữ liệu.Rồi bấm nút xóa để xóa thông tin người dùng.Hộp thoại xác nhận hiện lên bấm OK để xóa và bấm Cancel để bỏ qua.



Hình 3.13: Cửa sổ xác thực xóa thông tin người dùng

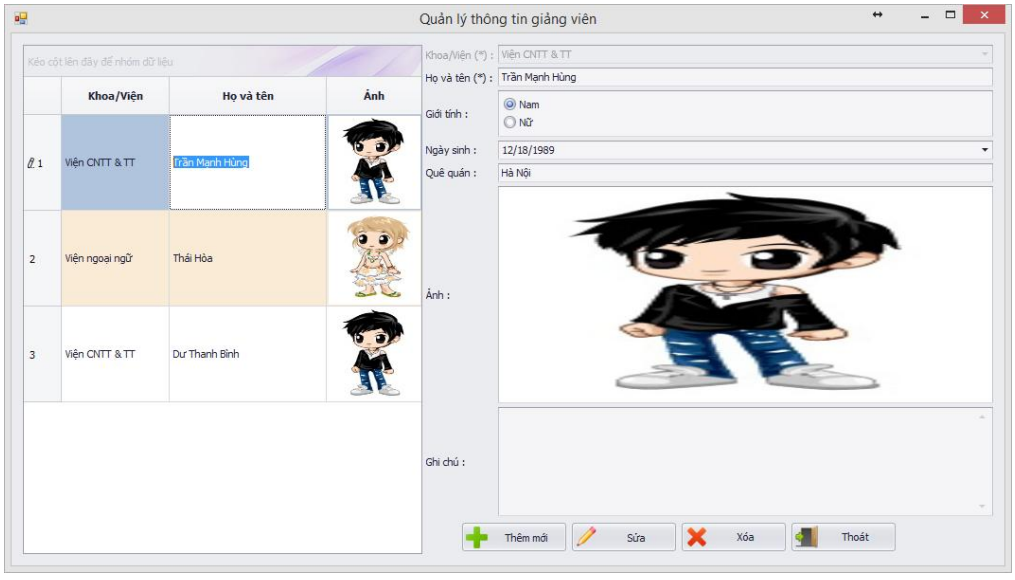
3.2.7 Giao diện quản lý Khoa/Viện:



Hình 3.14: Màn hình quản lý thông tin khoa/viện

Việc thêm mới, sửa, xóa thông tin khoa/viện cũng tương tự quản lý thông tin người dùng.

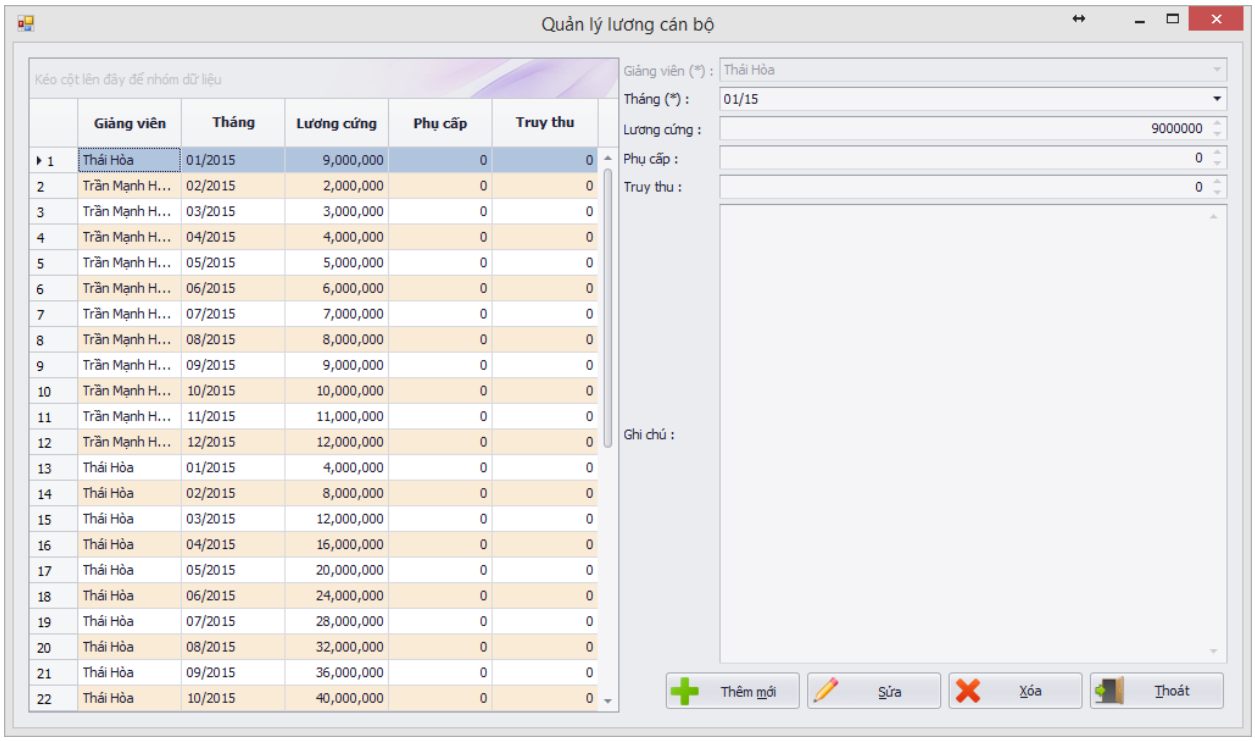
3.2.8 Giao diện quản lý giảng viên :



Hình 3.15: Màn hình quản lý thông tin giảng viên

Việc thêm mới, sửa, xóa thông tin giảng viên cũng tương tự quản lý thông tin người dùng.

3.2.9 Giao diện quản lý lương :



Hình 3.16 : Màn hình quản lý thông tin giảng viên

Việc thêm mới, sửa, xóa thông tin lương cũng tương tự quản lý thông tin người dùng.

Kết luận chương 3: Chương này đã xây dựng được chương trình với mục đích thử nghiệm thuật toán Incremental DBSCAN đã đề ra ở chương 2 và đã thực hiện các yêu cầu sau:

Quản lý thông tin người dùng, quản lý thông tin Khoa/ Viện, giảng viên, dữ liệu lương.

Phân cụm với toàn bộ dữ liệu, phân cụm theo khoảng thời gian, phân cụm theo cán bộ của Khoa/ Viện.

Thông tin phân cụm gồm có: Tổng số cụm, số mẫu, số nhiễu, số phần tử trong cụm, phần trăm, biểu đồ các cụm, thông tin các phần tử trong từng cụm.

KẾT LUẬN

Luận văn nghiên cứu, tìm hiểu, tổng hợp những nét đặc trưng nhất trong lĩnh vực Khai phá dữ liệu nói chung và phương pháp Phân cụm dữ liệu nói riêng. Luận văn đã trình bày được một số kỹ thuật và thuật toán phân cụm dữ liệu điển hình, dựa trên các phương pháp đã có, cài đặt thử nghiệm thuật toán Incremental DBSCAN trong bài toán phân tích lương của cán bộ giáo viên trường Cao đẳng Nghề Hà Nam theo từng yêu cầu cụ thể. Thuật toán thử nghiệm có ưu điểm vượt trội hơn so với các thuật toán phân cụm dữ liệu tĩnh đó là khi dữ liệu thay đổi ta không phải phân cụm dữ liệu lại từ đầu mà kết quả tự cập nhật theo dữ liệu được thêm mới. Điều này rút giảm thiểu thời gian, chi phí, giúp chúng ta đánh giá kết quả một cách đa chiều hơn.

Với những gì mà luận văn đã đạt được, hướng phát triển của luận văn như sau:

Về lý thuyết: Tiếp tục nghiên cứu các phương pháp, cách tiếp cận mới trong lĩnh vực Khai phá dữ liệu nói chung và phân cụm dữ liệu nói riêng như: phân cụm mờ, phân cụm thống kê,… tìm kiếm và so sánh, chọn lựa thuật toán tối ưu nhất để giải quyết bài toán đưa ra, nghiên cứu và tìm hiểu thêm về Khai phá dữ liệu dự đoán và mô tả.

Về thực tiễn: Phát triển bài toán với dữ liệu lớn hơn, quan tâm đến nhiều lựa chọn hơn. Phát triển các ứng dụng của Khai phá dữ liệu và phân cụm dữ liệu trên nhiều lĩnh vực trong đời sống.

Mặc dù đã cố gắng tập trung tham khảo nhiều tài liệu, tạp chí khoa học trong và ngoài nước, nhưng luận văn không thể tránh khỏi nhiều thiếu sót, rất mong được sự chỉ bảo đóng góp của các quý thầy cô giáo.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Lê Văn Phùng, Quách Xuân Trường (2012), *Khai phá dữ liệu (Data Mining)*, NXB Thông tin và Truyền thông.

[2] Phạm Đình Hồng, *Nghiên cứu phương pháp phân nhóm dữ liệu áp dụng vào hệ thống truy vấn thông tin*, Luận văn thạc sỹ khoa học máy tính – ĐH Đà Nẵng, 2013

Tiếng Anh

[3] Anil K.Jain (2010), *"Data Clustering: 50 Year Beyond K-Means"*, Pattenrn Recognition Letters, Volume 31 Issue 8.

[4] Beckmann N., Kriegel H.-P., Schneider R., Seeger B (1990), "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *Proc. ACM SIGMOD Int. Conf.on Management of Data*, Atlantic City, NJ, pp. 322-331.

[5] Ciaccia P., Patella M., Zezula (1997), "M-tree: An Efficient Access Method for imilarity Search in Metric Spaces", *Proc. 23rd Int. Conf. on Very Large Data Bases, Athens*, pp. 426-435.

[6] Ester M., Kriegel H.-P., Sander J., Xu X (1996), "A Density-Based Algorithm for iscovering Clusters in Large Spatial Databases with Noise", *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, OR, pp. 226-231.

[7] Gan, Guojun, Chaoqun Ma, and Jianhong Wu (2007), *Data Clustering: Theory, Algorithms, and Applications*, ASA-SIAM Serie on Statistics and Applied Probability, SIAM, Philadephia, American Statiscal Association, Alexandria, Virginia.

[8] Jiawei Han, Micheline Kamber and Jian Pei (2012), *Data Mining: Concepts and Techniques (3rd Edition)*, Morgan Kaufmann Publishers, USA.

[9] Michal Wroblewski (2003), *A hierarchical www pages clustering algorithm based on the vec-tơ space model, MASTER THESIS Submitted in partial fulfillment of the requirements for the degree of Master of Science*, Poznań University of Technology, Poland, July.

[10] Nathan Edwards (2005), *Lecture 12*: *suffix tree, Alorithms in Biosequence Analysis-Fall*, USA

[11] Oren Zamir and Oren Etzioni (1998), *Web document Clustering: A Feasibility Demonstration*, University of Washington, USA, ACM, 1998.

[12] R. Krishnapuram, A. Joshi, L. Yi (1999), *A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering*, Proc.IEEE Intl. Conf. Fuzzy Systems, Korea.

[13] Wai-chiu Wong và Ada Fu (2000), *Incremental Document Clustering for Web Page Classification, IEEE 2000 Int*, Conf. on Infor, Society in the 21st.

[14] Xiufeng Jiang, Zhang Yi and Jian Cheng Lv (2006), *Fuzzy SVM with a new fuzzy membership function, Neural Computing and Application*, Volume 15(3), pp. 268-276.

[15] Y. Yang và J. Pedersen (1997), *A Comparative Study on Feature Selection in Text Categorization*, In Proc. of the 14th International Conference on Machine Learning.