

Mục lục

Chương 1. TỔNG QUAN VỀ PHÂN CỤM TRONG KHAI PHÁ DỮ LIỆU VÀ CÁC KHÁI NIỆM CƠ BẢN.....	3
1.1. Giới thiệu chung	3
1.2. Khai phá dữ liệu là gì?	3
1.3. Quá trình khai phá tri thức trong cơ sở dữ liệu	3
1.4. Các kỹ thuật áp dụng trong khai phá dữ liệu.....	3
1.4.1. Các kỹ thuật tiếp cận trong khai phá dữ liệu	3
1.4.2. Các dạng dữ liệu có thể khai phá	3
1.5. Ứng dụng của khai phá dữ liệu.....	3
1.6. Phân cụm dữ liệu và ứng dụng	3
1.6.1. Mục đích của phân cụm dữ liệu	3
1.6.2. Các bước cơ bản để phân cụm	3
1.6.3. Các loại đặc trưng	3
1.6.4. Các ứng dụng của phân cụm	3
1.6.5. Phân loại các thuật toán phân cụm	3
1.7. Các khái niệm và định nghĩa.....	3
1.7.1. Các định nghĩa phân cụm	3
1.7.2. Các độ đo gần gũi.....	3
1.7.2.1 CÁC ĐỊNH NGHĨA	3
1.7.2.2. CÁC ĐỘ ĐO GẦN GŨI GIỮA 2 ĐIỂM	3
1.7.2.3. HÀM GẦN GŨI GIỮA MỘT ĐIỂM VÀ MỘT TẬP	3
1.7.2.4. CÁC HÀM GẦN GŨI GIỮA HAI TẬP	3
Chương 2. CÁC THUẬT TOÁN PHÂN CỤM TUẦN TỰ.....	3
2.1. Số các cách phân cụm có thể	3
2.2. Thuật toán phân cụm tuần tự - BSAS	3
2.3. Ước lượng số cụm	3
2.4. Sửa đổi thuật toán BSAS - Thuật toán MBSAS	3
2.5. Thuật toán phân cụm tuần tự hai ngưỡng - TTSAS	3
2.6. Giai đoạn tinh chế	3
Bài tập chương 2	3
Chương 3. CÁC THUẬT TOÁN PHÂN CỤM PHÂN CẤP	3
3.1. Giới thiệu	3
3.2. Các thuật toán tích tụ - GAS	3
3.2.1. Một số định nghĩa.....	3
3.2.2. Một số thuật toán tích tụ dựa trên lý thuyết ma trận	3
3.2.3. Monotonicity và Crossover.....	3
3.2.4. Một số thuật toán tích tụ dựa trên lý thuyết đồ thị	3
3.2.5. Ảnh hưởng của ma trận gần gũi tới sơ đồ phân cụm	3
3.3. Các thuật toán phân rã - GDS	3
3.3.1. Cải tiến sơ đồ GDS.....	3
3.4. Lựa chọn phân cụm tốt nhất	3
Bài tập chương 3	3

BẢNG TỪ VIỆT TẮT

Từ hoặc cụm từ	Từ tiếng Anh	Từ tiếng Việt
BSAS	Basic Sequential Algorithmic Scheme	Sơ đồ thuật toán phân cụm tuần tự cơ sở
CSDL	Data Base	Cơ sở dữ liệu
GAS	Generalized Agglomerative Scheme	Sơ đồ tích tụ tổng quát
GDS	Generalized Divisive Scheme	Sơ đồ phân rã tổng quát
GTAS	Graph Theory – based Algorithmic Scheme	Sơ đồ thuật toán dựa trên lý thuyết đồ thị
KDD	Knowledge Discovery in Databases	Khai phá tri thức trong cơ sở dữ liệu
MBSAS	Modified Basic Sequential Algorithmic Scheme	Sơ đồ thuật toán phân cụm tuần tự cơ sở sửa đổi
MST	Minimum Spanning Tree	Cây khung nhỏ nhất
MUAS	Matrix Updating Algorithmic Scheme	Sơ đồ thuật toán biến đổi ma trận
SM	Similarity Measure	Độ đo tương tự
TTSAS	Two – Threshold Sequential Algorithmic Scheme	Sơ đồ thuật toán tuần tự 2 ngưỡng
UPGMA	Unweighted Pair Group Method Average	Phương pháp trung bình theo cặp không trọng số
UPGMC	Unweight Pair Group Method Centroid	Phương pháp trọng tâm theo cặp không chọn số
WPGMA	Weighted Pair Group Method Average	Phương pháp trung bình theo cặp trọng số
WPGMC	Weighted Pair Group Method Centroid	Phương pháp trọng tâm theo cặp trọng số

Chương 1.

TỔNG QUAN VỀ PHÂN CỤM TRONG KHAI PHÁ DỮ LIỆU VÀ CÁC KHÁI NIỆM CƠ BẢN

1.1. Giới thiệu chung

Những năm 60 của thế kỷ trước, người ta đã bắt đầu sử dụng các công cụ tin học để tổ chức và khai thác các CSDL. Cùng với sự phát triển vượt bậc của các công nghệ điện tử và truyền thông, khả năng thu thập và lưu trữ và xử lý dữ liệu cho các hệ thống tin học không ngừng được nâng cao, theo đó, lượng thông tin được lưu trữ trên các thiết bị nhớ không ngừng tăng lên. Thống kê sơ bộ cho thấy, lượng thông tin trên các hệ thống tin học cứ sau 20 tháng lại tăng gấp đôi [3]. Cuối thập kỷ 80 của thế kỷ 20, sự phát triển rộng khắp của các CSDL ở mọi quy mô đã tạo ra sự bùng nổ thông tin trên toàn cầu. Vào thời gian này, người ta bắt đầu đề cập đến khái niệm khủng hoảng phân tích dữ liệu tác nghiệp để cung cấp thông tin với yêu cầu chất lượng ngày càng cao cho người làm quyết định trong các tổ chức tài chính, thương mại, khoa học,...

Đúng như John Naisbett đã cảnh báo “*Chúng ta đang chìm ngập trong dữ liệu mà vẫn đòi tri thức*”. Lượng dữ liệu khổng lồ này thực sự là một nguồn “*tài nguyên*” có nhiều giá trị bởi thông tin là yếu tố then chốt trong mọi hoạt động quản lý, kinh doanh, phát triển sản xuất và dịch vụ, ... nó giúp những người điều hành và quản lý có hiểu biết về môi trường và tiến trình hoạt động của tổ chức mình trước khi ra quyết định để tác động đến quá trình hoạt động nhằm đạt được các mục tiêu một cách hiệu quả và bền vững.

Khai phá dữ liệu (Data Mining) là một lĩnh vực mới xuất hiện, nhằm tự động khai thác những thông tin, những tri thức có tính tiềm ẩn, hữu ích từ những CSDL lớn cho các đơn vị, tổ chức, doanh nghiệp,... từ đó làm thúc đẩy khả năng sản xuất, kinh doanh, cạnh tranh cho các đơn vị, tổ chức này. Các kết quả khoa học cùng những ứng dụng thành công trong khám phá tri thức, cho thấy, khai phá dữ liệu là một lĩnh vực phát triển bền vững, mang lại nhiều lợi ích và có nhiều triển vọng, đồng thời có ưu thế hơn hẳn so với các công cụ phân tích dữ liệu truyền thống. Hiện nay, khai phá dữ liệu đã ứng dụng ngày càng rộng rãi trong các lĩnh vực như: Thương mại, tài chính, điều trị y học, viễn thông, tin – sinh,....

1.2. Khai phá dữ liệu là gì?

Khai phá dữ liệu là một hướng nghiên cứu mới ra đời hơn một thập niên trở lại đây, các kỹ thuật chính được áp dụng trong lĩnh vực này phần lớn được thừa kế từ lĩnh vực CSDL, học máy, trí tuệ nhân tạo, lý thuyết thông tin, xác suất thống kê, và tính toán hiệu năng cao. Do sự phát triển nhanh của khai phá dữ liệu về phạm vi áp dụng và các phương pháp tìm kiếm tri thức, nên đã có nhiều quan điểm khác nhau về khai phá dữ liệu. Tuy nhiên, ở một mức trừu tượng nhất định, chúng ta định nghĩa khai phá dữ liệu như sau [10]:

Định nghĩa : *Khai phá dữ liệu là một quá trình tìm kiếm, phát hiện các tri thức mới, tiềm ẩn, hữu dụng trong CSDL lớn.*

Khai phá tri thức trong CSDL (Knowledge Discovery in Databases - KDD) là mục tiêu chính của khai phá dữ liệu, do vậy hai khái niệm khai phá dữ liệu và KDD được các nhà khoa học trên hai lĩnh vực xem là tương đương với nhau. Thế nhưng, nếu phân chia một cách chi tiết thì khai phá dữ liệu là một bước chính trong quá trình KDD.

1.3. Quá trình khai phá tri thức trong cơ sở dữ liệu

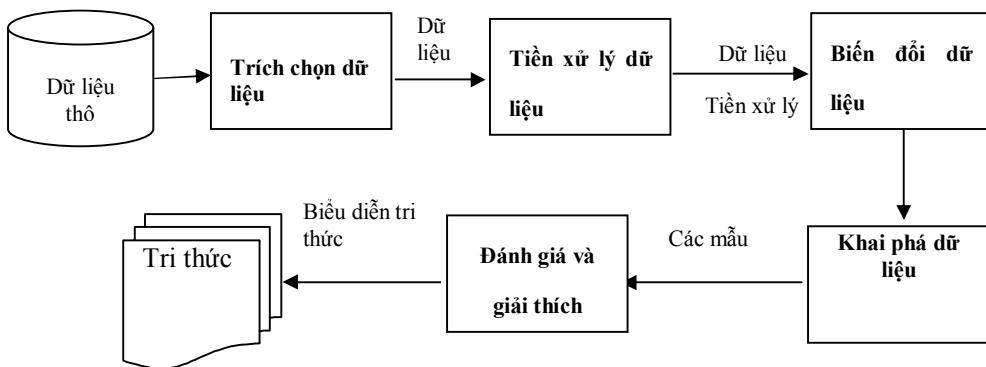
Khai phá tri thức trong CSDL, KDD, là lĩnh vực liên quan đến các ngành như : thống kê, học máy, CSDL, thuật toán, trực quan hóa dữ liệu, tính toán song song và hiệu năng cao,...

Quá trình KDD có thể phân thành các giai đoạn sau [3][10]:

- *Trích chọn dữ liệu* : là bước trích chọn những tập dữ liệu cần được khai phá từ các tập dữ liệu lớn (databases, data warehouses, data repositories) ban đầu theo một số tiêu chí nhất định.
- *Tiến xử lý dữ liệu* : là bước làm sạch dữ liệu (xử lý với dữ liệu không đầy đủ, dữ liệu nhiễu, dữ liệu không nhất quán, .v.v.), rút gọn dữ liệu (sử dụng hàm nhóm và tính tổng, các phương pháp nén dữ liệu, sử dụng histograms, lấy mẫu, .v.v.), rồi rác hóa dữ liệu (rồi rác hóa dựa vào histograms, dựa vào entropy, dựa vào phân khoảng, .v.v.). Sau bước này, dữ liệu sẽ nhất quán, đầy đủ, được rút gọn, và được rác hóa.
- *Biến đổi dữ liệu* : Đây là bước chuẩn hóa và làm mịn dữ liệu để đưa dữ liệu về dạng thuận lợi nhất nhằm phục vụ cho các kỹ thuật khai phá ở bước sau.

- *Khai phá dữ liệu*: Đây là bước áp dụng những kỹ thuật phân tích (phần nhiều là các kỹ thuật của học máy) nhằm để khai thác dữ liệu, trích chọn được những mẫu thông tin, những mối liên hệ đặc biệt trong dữ liệu. Đây được xem là bước quan trọng và tốn nhiều thời gian nhất của toàn quá trình KDD.
- *Đánh giá và biểu diễn tri thức*: những mẫu thông tin và mối liên hệ trong dữ liệu đã được khai phá ở bước trên được chuyển dạng và biểu diễn ở một dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật, .v.v. Đồng thời bước này cũng đánh giá những tri thức khai phá được theo những tiêu chí nhất định.

Các giai đoạn trong KDD được thể hiện trực quan như hình 1.1 dưới đây:



Hình 1-1. Các bước thực hiện trong quá trình khai phá tri thức

1.4. Các kỹ thuật áp dụng trong khai phá dữ liệu

1.4.1. Các kỹ thuật tiếp cận trong khai phá dữ liệu

Khai phá tri thức trong CSDL là một lĩnh vực liên ngành, bao gồm: Tổ chức dữ liệu, học máy, trí tuệ nhân tạo và các khoa học khác.

Nếu theo quan điểm của học máy (Machine Learning), thì các kỹ thuật trong khai phá dữ liệu, bao gồm :

- ❖ *Học có giám sát (Supervised learning)* : Là quá trình gán nhãn lớp cho các phần tử trong CSDL dựa trên một tập các ví dụ huấn luyện và các thông tin về nhãn lớp đã biết.
- ❖ *Học không có giám sát (Unsupervised learning)* : Là quá trình phân chia một tập dữ liệu thành các lớp hay là cụm (clustering) dữ liệu tương tự nhau mà chưa biết trước các thông tin về lớp hay tập các ví dụ huấn luyện.

- ❖ *Học nửa giám sát (Semi - Supervised learning)* : Là quá trình phân chia một tập dữ liệu thành các lớp dựa trên một tập nhỏ các ví dụ huấn luyện và một số các thông tin về một số nhãn lớp đã biết trước.

Nếu căn cứ vào lớp các bài toán cần giải quyết, thì khai phá dữ liệu bao gồm các kỹ thuật áp dụng sau [10]:

- ❖ *Phân lớp và dự đoán (classification and prediction)*: xếp một đối tượng vào một trong những lớp đã biết trước. Ví dụ: phân lớp các dữ liệu bệnh nhân trong hồ sơ bệnh án. Hướng tiếp cận này thường sử dụng một số kỹ thuật của học máy như cây quyết định (decision tree), mạng nơ ron nhân tạo (neural network), .v.v. Phân lớp và dự đoán còn được gọi là học có giám sát.
- ❖ *Luật kết hợp (association rules)*: là dạng luật biểu diễn tri thức ở dạng khá đơn giản. Ví dụ: “60 % nữ giới vào siêu thị mua phấn thì có tới 80% trong số họ sẽ mua thêm son”. Luật kết hợp được ứng dụng nhiều trong lĩnh vực kinh doanh, y học, tin-sinh, tài chính và thị trường chứng khoán, .v.v.
- ❖ *Phân tích chuỗi theo thời gian (sequential/ temporal patterns)*: tương tự như khai phá luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán vì nó có tính dự báo cao.
- ❖ *Phân cụm (clustering/ segmentation)*: xếp các đối tượng theo từng cụm dữ liệu tự nhiên. *Phân cụm* còn được gọi là học không giám sát (unsupervised learning).
- ❖ *Mô tả khái niệm (concept description and summarization)*: thiên về mô tả, tổng hợp và tóm tắt khái niệm. Ví dụ: tóm tắt văn bản.

1.4.2. Các dạng dữ liệu có thể khai phá

Do khai phá dữ liệu được ứng dụng rộng rãi nên nó có thể làm việc với rất nhiều kiểu dữ liệu khác nhau. Sau đây là một số dạng dữ liệu điển hình [10] : *CSDL quan hệ*, *CSDL đa chiều (multidimensional structures, data warehouses)*, *CSDL dạng giao dịch*, *CSDL quan hệ - hướng đối tượng*, *dữ liệu không gian và thời gian*, *dữ liệu chuỗi thời gian*, *CSDL đa phương tiện*, *dữ liệu Text và Web*, ...

1.5. Ứng dụng của khai phá dữ liệu

Khai phá dữ liệu là một lĩnh vực được quan tâm và ứng dụng rộng rãi. Một số ứng dụng điển hình trong khai phá dữ liệu có thể liệt kê như sau : *Phân tích dữ liệu và hỗ trợ ra quyết định, điều trị y học, Text mining & Web mining, tin-sinh (bio-informatics), tài chính và thị trường chứng khoán, bảo hiểm (insurance), .v.v.*

1.6. Phân cụm dữ liệu và ứng dụng

1.6.1. Mục đích của phân cụm dữ liệu

Phân loại là một trong những hành vi nguyên thuỷ nhất của con người nhằm nắm giữ lượng thông tin không lồ họ nhận được hằng ngày vì sự xử lý mọi thông tin như một thực thể đơn lẻ là không thể. Phân cụm dữ liệu nhằm mục đích chính là khai phá cấu trúc của mẫu dữ liệu để thành lập các nhóm dữ liệu từ tập dữ liệu lớn, theo đó, cho phép người ta đi sâu vào phân tích và nghiên cứu cho từng cụm dữ liệu này nhằm khai phá và tìm kiếm các thông tin tiềm ẩn, hữu ích phục vụ cho ra quyết định.

Một vài ví dụ về ý nghĩa thực tiễn của phân cụm dữ liệu như sau :

- *Khám phá ra các vị trí địa lý thuận lợi cho việc xây dựng các kho hàng phục vụ mua bán hàng của một công ty thương mại*
- *Xác định các cụm ảnh như ảnh của các loài động vật như loài thú, chim,... trong tập CSDL ảnh về động vật nhằm phục vụ cho việc tìm kiếm ảnh*
- *Xác định các nhóm người bệnh nhằm cung cấp thông tin cho việc phân phối các thuốc điều trị trong y tế*
- *Xác định nhóm các khách hàng trong CSDL ngân hàng có vốn các đầu tư vào bất động sản cao...*

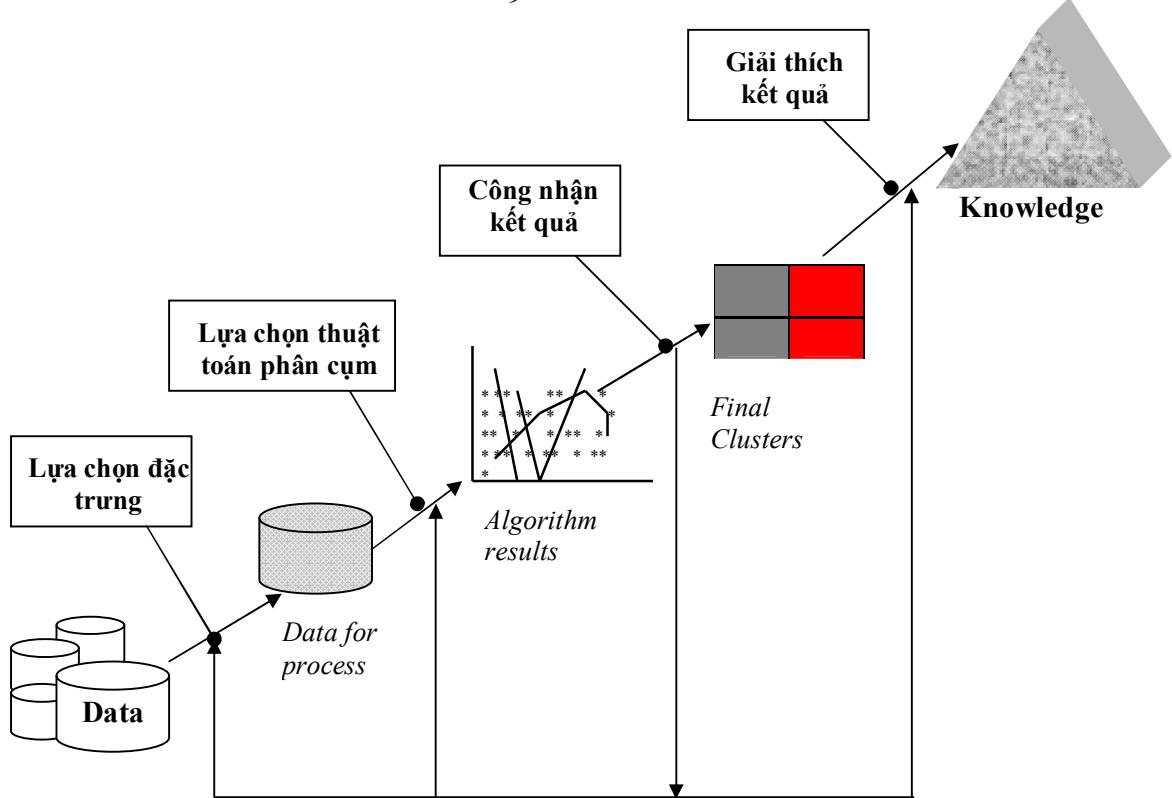
Như vậy, phân cụm dữ liệu là một phương pháp xử lý thông tin quan trọng và phổ biến, nó nhằm khám phá mối liên hệ giữa các mẫu dữ liệu bằng cách tổ chức chúng thành các cụm tương tự.

Tiếp theo, giả sử rằng tất cả các dạng dữ liệu được biểu diễn bởi khái niệm đặc trưng, các đặc trưng hình thành nên vector đặc trưng ℓ - chiều. Thuật ngữ phân cụm được hiểu là phân cụm dữ liệu.

1.6.2. Các bước cơ bản để phân cụm

- **Chọn lựa đặc trưng :** Các đặc trưng phải được chọn lựa một cách hợp lý để có thể “*mã hóa*” nhiều nhất thông tin liên quan đến công việc quan tâm. Mục tiêu chính là phải giảm thiểu sự dư thừa thông tin giữa các đặc trưng. Các đặc trưng cần được tiền xử lý trước khi dùng trong các bước sau.
- **Chọn độ đo gần gũi:** Đây là một độ đo chỉ ra mức độ tương tự hay không tương tự giữa hai vector đặc trưng. Phải đảm bảo rằng tất cả các vector đặc trưng góp phần như nhau trong việc tính toán độ đo gần gũi và không có đặc trưng nào át hẳn đặc trưng nào. Điều này được đảm nhận bởi quá trình tiền xử lý.
- **Tiêu chuẩn phân cụm:** Điều này phụ thuộc vào sự giải thích của chuyên gia cho thuật ngữ “*dễ nhận thấy*” dựa vào loại của các cụm được chuyên gia cho rằng đang ẩn dấu dưới tập dữ liệu. Chẳng hạn, một cụm loại *chặt* (compact) của các vector đặc trưng trong không gian ℓ -chiều có thể dễ nhận thấy theo một tiêu chuẩn, trong khi một cụm loại “*dài và mỏng*” lại có thể được dễ nhận thấy bởi một tiêu chuẩn khác. Tiêu chuẩn phân loại có thể được diễn đạt bởi hàm chi phí hay một vài loại quy tắc khác.
- **Thuật toán phân loại:** Cần lựa chọn một sơ đồ thuật toán riêng biệt nhằm làm sáng tỏ cấu trúc cụm của tập dữ liệu.
- **Công nhận kết quả:** Khi đã có kết quả phân loại thì ta phải kiểm tra tính đúng đắn của nó. Điều này thường được thực hiện bởi việc dùng các kiểm định phù hợp.
- **Giải thích kết quả:** Trong nhiều trường hợp, chuyên gia trong lĩnh vực ứng dụng phải kết hợp kết quả phân loại với bằng chứng thực nghiệm và phân tích để đưa ra các kết luận đúng đắn. Trong một số trường hợp, nên có cả bước khuynh hướng phân cụm; trong bước này có các kiểm định khác nhau để chỉ ra một dữ liệu có hay không một cấu trúc phân cụm. Ví dụ như tập dữ liệu của ta có thể hoàn toàn ngẫu nhiên vì vậy mọi cố gắng phân cụm đều vô nghĩa.

Các lựa chọn khác nhau của các đặc trưng, độ đo gần gũi, tiêu chuẩn phân cụm có thể dẫn tới các kết quả phân cụm khác nhau. Do đó, việc lựa chọn một cách hợp lý nhất hoàn toàn dựa vào kiến thức và kinh nghiệm của chuyên gia. Tính chủ quan (của chuyên gia) là một thực tế mà ta phải chấp nhận.



Hình 1-2. Các bước trong quá trình phân cụm

1.6.3. Các loại đặc trưng

Có bốn loại đặc trưng, đó là:

- **Các đặc trưng danh nghĩa (nominal):** Gồm các đặc trưng mà các giá trị của nó mã hoá các trạng thái. Chẳng hạn cho một đặc trưng là giới tính của một người thì các giá trị có thể của nó là 1 ứng với nam và 0 ứng với nữ. Rõ ràng là bất kỳ sự so sánh về lượng nào giữa các giá trị loại này đều là vô nghĩa.
- **Các đặc trưng thứ tự (ordinal):** Là các đặc trưng mà các giá trị của nó có thể sắp một cách có ý nghĩa. Ví dụ về một đặc trưng thể hiện sự hoàn thành khoa học của một sinh viên. Giả sử các giá trị có thể là 4, 3, 2, 1 tương ứng với các ý nghĩa: "xuất sắc", "rất tốt", "tốt", "không tốt". Các giá trị này được sắp xếp theo một thứ tự có ý nghĩa nhưng sự so sánh giữa hai giá trị liên tiếp là không quan trọng lắm về lượng.
- **Các đặc trưng đo theo khoảng (interval-scaled):** Với một đặc trưng cụ thể nếu sự khác biệt giữa hai giá trị là có ý nghĩa về mặt số lượng thì ta có đặc trưng đo theo khoảng (còn gọi là *thang khoảng*). Ví dụ về đặc trưng nhiệt độ, nếu từ

10-15 độ thì được coi là rét đậm, còn nếu dưới 10 độ thì được coi là rét hại, vì vậy mỗi khoảng nhiệt độ mang một ý nghĩa riêng.

- **Các đặc trưng đo theo tỷ lệ (ratio-scaled):** Cũng với ví dụ nhiệt độ ở trên ta không thể coi tỷ lệ giữa nhiệt độ Hà Nội 10 độ với nhiệt độ Matxcova 1 độ mang ý nghĩa rằng Hà Nội nóng gấp mười lần Matxcova. Trong khi đó, một người nặng 100 kg được coi là nặng gấp hai lần một người nặng 50 kg. Đặc trưng cân nặng là một đặc trưng đo theo tỷ lệ (*thang tỷ lệ*).

1.6.4. Các ứng dụng của phân cụm

Phân cụm là một công cụ quan trọng trong một số ứng dụng. Sau đây là một số ứng dụng của nó:

- **Giảm dữ liệu:** Giả sử ta có một lượng lớn dữ liệu (N). Phân cụm sẽ nhóm các dữ liệu này thành m cụm dữ liệu dễ nhận thấy và $m \ll N$. Sau đó xử lý mỗi cụm như một đối tượng đơn.
- **Rút ra các giả thuyết:** Các giả thuyết này có liên quan đến tính tự nhiên của dữ liệu và phải được kiểm tra bởi việc dùng một số tập dữ liệu khác.
- **Kiểm định giả thuyết:** Ta sẽ phân cụm để xét xem có tồn tại một tập dữ liệu nào đó trong tập dữ liệu thỏa mãn các giả thuyết đã cho hay không. Chẳng hạn xem xét giả thuyết sau đây: “*Các công ty lớn đầu tư ra nước ngoài*”. Để kiểm tra, ta áp dụng kỹ thuật phân cụm với một tập đại diện lớn các công ty. Giả sử rằng mỗi công ty được đặc trưng bởi tầm vóc, các hoạt động ở nước ngoài và khả năng hoàn thành các dự án. Nếu sau khi phân cụm, một cụm các công ty được hình thành gồm các công ty lớn và có vốn đầu tư ra nước ngoài (không quan tâm đến khả năng hoàn thành các dự án) thì giả thuyết đó được cung cấp bởi kỹ thuật phân cụm đã thực hiện.
- **Dự đoán dựa trên các cụm:** Đầu tiên ta sẽ phân cụm một tập dữ liệu thành các cụm mang đặc điểm của các dạng mà nó chứa. Sau đó, khi có một dạng mới chưa biết ta sẽ xác định xem nó sẽ có khả năng thuộc về cụm nào nhất và dự đoán được một số đặc điểm của dạng này nhờ các đặc trưng chung của cả cụm.

Cụ thể hơn, phân cụm dữ liệu đã được áp dụng cho một số ứng dụng điển hình trong các lĩnh vực sau [13] :

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

1.6.5. Phân loại các thuật toán phân cụm

Các thuật toán phân cụm có thể được xem như các sơ đồ cung cấp cho ta các cụm “để nhận thấy” bởi việc chỉ xem xét một phần của tập chứa tất cả các cách phân cụm của X . Kết quả phân cụm phụ thuộc vào thuật toán và tiêu chuẩn phân cụm. Như vậy, một thuật toán phân cụm là một chức năng học cố gắng tìm ra các đặc trưng riêng biệt của các cụm ẩn dấu dưới tập dữ liệu. Có nhiều cách để phân loại các thuật toán phân cụm, sau đây là một cách phân loại:

- **Các thuật toán phân cụm tuần tự (Sequential Algorithms):**

Các thuật toán này sinh ra một cách phân cụm duy nhất, chúng là các phương pháp trực tiếp và nhanh. Trong hầu hết các thuật toán thuộc loại này, tất cả các vector đặc trưng tham gia vào thuật toán một hoặc vài lần (không hơn 6 lần). Kết quả cuối cùng thường phụ thuộc vào thứ tự các vector tham gia vào thuật toán. Những sơ đồ loại này có khuynh hướng sinh ra các cụm có hình dạng chật siêu cầu hoặc siêu elip xoay tuỳ theo độ đo được dùng.

- **Các thuật toán phân cụm phân cấp (Hierarchical Algorithms)**

- **Các thuật toán tích tụ (Agglomerative Algorithms):**

Chúng sinh ra một dãy cách phân cụm mà số cụm, m , giảm dần ở mỗi bước. Cách phân cụm ở mỗi bước là kết quả của cách phân cụm ở bước trước đó bằng việc trộn hai cụm vào một. Các đại diện chính của loại này là thuật toán liên kết đơn (phù hợp với các cụm dài và mỏng) và thuật toán liên kết đầy đủ (phù hợp với các cụm chật). Các thuật toán tích tụ thường dựa trên lý thuyết đồ thị và lý thuyết ma trận.

- **Các thuật toán phân rã (Divise Algorithms):**

Sinh ra một dãy cách phân cụm mà số cụm, m , tăng dần ở mỗi bước. Cách phân cụm ở mỗi bước là kết quả cách phân cụm ở bước trước đó bằng việc chia đôi một cụm đơn.

- **Các thuật toán phân cụm dựa trên việc tối ưu hóa hàm chi phí:**

Hàm chi phí J đo độ “đã nhận thấy” của các cách phân cụm. Thường thì số các cụm, m , là cố định. Thuật toán sẽ dùng các khái niệm về phép tính vi phân và sinh ra các cách phân cụm liên tiếp trong khi cố gắng tối ưu hóa J . Thuật toán sẽ dừng khi một tối ưu địa phương được xác định. Các thuật toán loại này cũng được gọi là các sơ đồ tối ưu hóa hàm lặp. Chúng được phân tiếp như sau:

- **Các thuật toán phân cụm chật hay rõ:**

Vector thuộc hoàn toàn vào một cụm cụ thể. Việc đưa một vector về các cụm cụ thể được thực hiện một cách tối ưu theo tiêu chuẩn phân cụm tối ưu.

- **Các thuật toán phân cụm theo các hàm xác suất:**

Dựa vào lý thuyết phân lớp Bayes và mỗi vector x được phân về cụm thứ i nếu $p(C_i | x)$ là lớn nhất (xác suất để x được phân đúng vào cụm C_i).

- **Các thuật toán phân cụm mờ:**

Các vector thuộc về một cụm nào đó với một độ chắc chắn.

- *Các thuật toán phân cụm theo khả năng :*

Trong trường hợp này ta đo khả năng một vector đặc trưng thuộc về một cụm nào đó.

- *Các thuật toán phát hiện biên phân tách :*

Các thuật toán này cố gắng đặt các biên phân tách một cách tối ưu giữa các cụm.

• *Các thuật toán khác*

- *Các thuật toán phân cụm nhánh và cành :*

Các thuật toán này cung cấp cho ta các cách phân cụm tối ưu toàn cục mà không phải xét tới tất cả các cách phân cụm có thể, với m cố định và một tiêu chuẩn phân cụm định trước. Nhưng đòi hỏi rất nhiều tính toán.

- *Các thuật toán phân cụm di truyền :*

Sử dụng dân số ban đầu của các cách phân cụm có thể và sinh ra các số dân mới một cách lặp đi lặp lại. Số dân mới này nhìn chung chứa các cách phân cụm tốt hơn so với thế hệ trước, theo một tiêu chuẩn đã định trước.

- *Phương pháp thư giãn ngẫu nhiên :*

Đảm bảo rằng với các điều kiện chắc chắn, độ hội tụ theo xác suất tới cách phân cụm tối ưu toàn cục nhưng tốn nhiều thời gian tính toán.

- *Thuật toán phân cụm tìm khe :*

Xem mỗi vector đặc trưng như là một biến ngẫu nhiên x . Chúng dựa trên một giả định được công nhận rộng rãi rằng vùng phân bố của x nơi có nhiều vector tương ứng với vùng mật độ cao của hàm mật độ xác suất (probability density function), vì vậy việc ước lượng các hàm mật độ xác suất sẽ làm rõ các khu vực nơi các cụm hình thành.

- *Thuật toán học cạnh tranh:*

Không dùng các hàm chi phí, chúng tạo ra vài cách phân cụm và các cách này hội tụ tới cách dễ nhận thấy nhất. Các đại diện tiêu biểu của loại này là sơ đồ học cạnh tranh cơ bản và thuật toán học lỗ rò.

- *Các thuật toán dựa trên kỹ thuật biến đổi hình thái học :*

Cố gắng đạt được sự phân chia tốt hơn giữa các cụm.

1.7. Các khái niệm và định nghĩa

1.7.1. Các định nghĩa phân cụm

a. Định nghĩa 1:

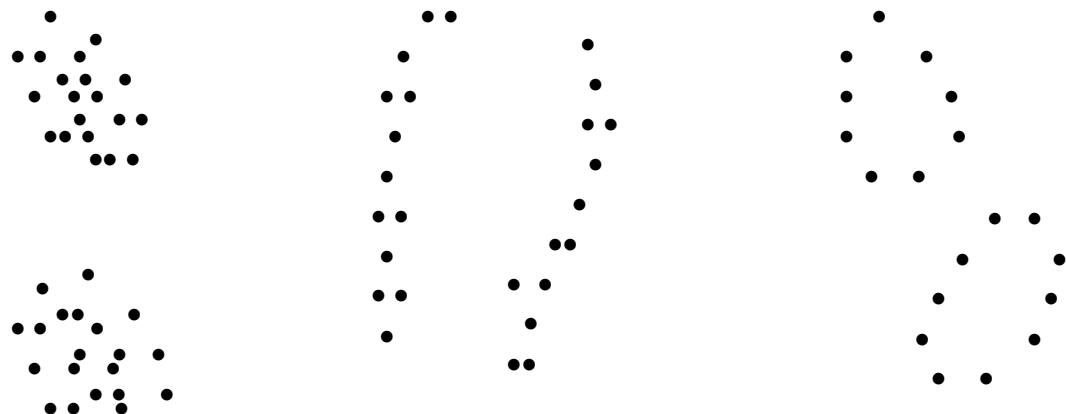
Cho X là một tập dữ liệu:

$$X = \{x_1, x_2, \dots, x_N\} \quad (1.1)$$

Ta định nghĩa m -phân cụm của X như một sự phân chia X thành m tập (cụm): C_1, C_2, \dots, C_m sao cho thoả 3 điều kiện:

- $C_i \neq \emptyset, \quad i \in \{1, 2, \dots, m\}$
- $\bigcup_{i=1}^m C_i = X$
- $C_i \cap C_j = \emptyset, \quad i \neq j; \quad i, j \in \{1, \dots, m\}$

Thêm vào đó, các vector trong một cụm là *tương tự nhau* hơn so với các vector thuộc một cụm khác. Lượng hoá thuật ngữ *tương tự* và *không tương tự* phụ thuộc rất nhiều vào loại của cụm. Chẳng hạn, loại cụm chật thì có một số độ đo phù hợp, trong khi loại cụm có hình dáng dài và mỏng lại phù hợp hơn với các độ đo loại khác (xem hình 1.3). Với định nghĩa trên, mỗi vector chỉ thuộc về một cụm riêng nên loại phân cụm này thỉnh thoảng còn được gọi là *chặt* hay *rõ* (*hard or crisp*).



(a) Các tập chật.

(b) Các tập dài và mỏng

(c) Các tập dạng cầu và elipxit

Hình 1-3. Hình dạng các loại cụm

Dựa vào khái niệm tập mờ ta có thể định nghĩa như sau:

b. Định nghĩa 2: Một sự phân cụm mờ tập X thành m cụm được mô tả bởi m hàm thuộc u_j sao cho:

$$u_j: X \rightarrow [0, 1], j \in \{1, \dots, m\} \quad (1.2)$$

$$\begin{aligned} \text{và } \sum_{j=1}^m u_j(x_i) &= 1, i \in \{1, 2, \dots, N\} \\ 0 < \sum_{i=1}^N u_j(x_i) &< N, j \in \{1, 2, \dots, m\} \end{aligned} \quad (1.3)$$

Mỗi cụm trong trường hợp này có thể không được định nghĩa chính xác. Nghĩa là mỗi vector x thuộc về nhiều hơn một cụm, với mỗi cụm nó lại thuộc về với độ thuộc u_j :

- u_j gần 1: mức độ thuộc của x vào cụm thứ j cao;
- u_j gần 0: mức độ thuộc của x vào cụm thứ j thấp.

Nếu một hàm thuộc có giá trị gần 1 với hai vector thì hai vector này được coi là tương tự nhau.

Điều kiện (1.3) đảm bảo rằng không tồn tại một cụm mà không chứa bất kỳ vector nào. Định nghĩa 1 là một trường hợp riêng của định nghĩa 2 khi hàm thuộc chỉ nhận hai giá trị 0 và 1, lúc này nó được gọi là hàm đặc trưng.

1.7.2. Các độ đo gần gũi

1.7.2.1 CÁC ĐỊNH NGHĨA

Chúng ta sẽ bắt đầu với việc định nghĩa liên quan đến độ đo giữa các vector sau đó mở rộng chúng cho trường hợp độ đo giữa các tập vector.

a. Một độ đo không tương tự (Dissimilarity Measure - DM) d trên X là một hàm:

$$d: X \times X \rightarrow R$$

trong đó R là tập số thực, sao cho:

- $\exists d_0 \in R : -\infty < d_0 \leq d(x, y) < +\infty, \forall x, y \in X$ (1.4)

- $d(x, x) = d_0, \forall x \in X$ (1.5)

- $d(x, y) = d(y, x), \forall x, y \in X$ (1.6)

Ngoài ra nếu:

$$d(x, y) = d_0 \text{ nếu và chỉ nếu } x = y \quad (1.7)$$

$$\text{và } d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in X \quad (1.8)$$

thì d được gọi là một DM metric. (1.7) chỉ ra rằng độ đo không tương tự nhỏ nhất khi hai vector là đồng nhất. Dễ dàng nhận thấy khoảng cách Euclid là một độ đo không tương tự metric (DM metric).

b. Một độ đo tương tự (Similarity Measure - SM) s trên X là một hàm:

$$s: X \times X \rightarrow R$$

sao cho:

- $\exists s_0 \in R : -\infty < s(x, y) \leq s_0 < +\infty, \forall x, y \in X$ (1.9)

- $s(x, x) = s_0, \forall x \in X$ (1.10)

- $s(x, y) = s(y, x), \forall x, y \in X$ (1.11)

Ngoài ra nếu:

$$s(x, y) = s_0 \text{ nếu và chỉ nếu } x = y \quad (1.12)$$

$$\text{và } s(x, y).s(y, z) \leq [s(x, y) + s(y, z)].s(x, z), \forall x, y, z \in X \quad (1.13)$$

thì s được gọi là một SM metric.

c. Tiếp theo ta sẽ mở rộng định nghĩa trên để có thể đo độ gần gũi giữa các tập con của X.

Cho U là một lớp các tập con của X , nghĩa là các $D_i \subset X, i = 1, \dots, k$ và $U = \{D_1, D_2, \dots, D_k\}$. Một độ đo gần gũi ϕ trên U là một hàm:

$$\phi : U \times U \rightarrow R$$

Các công thức (1.4) – (1.8) cho độ đo không tương tự và (1.9) - (1.13) cho độ đo tương tự được lặp lại với việc thay thế x, y, X lần lượt bởi D_i, D_j, U .

Thông thường, các độ đo gần gũi giữa hai tập D_i, D_j được định nghĩa thông qua độ đo gần gũi giữa các phần tử của chúng.

• **Ví dụ:**

Cho $X = \{x_1, \dots, x_6\}$ và $U = \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_3, x_4, x_5\}, \{x_1, x_2, x_3, x_4, x_5\}\}$ và hàm không tương tự sau đây:

$$d_{\min}^{ss}(D_i, D_j) = \min_{x \in D_i, y \in D_j} d_2(x, y)$$

Với d_2 là khoảng cách Euclid giữa hai vector.

Giá trị nhỏ nhất có thể của d_{\min}^{ss} là 0.

Vì khoảng cách Euclid giữa một vector với bản thân nó bằng 0 nên

$$d_{\min}^{ss}(D_i, D_i) = 0$$

$$\text{và } d_{\min}^{ss}(D_i, D_j) = d_{\min}^{ss}(D_j, D_i)$$

Vì vậy hàm này là một độ đo không tương tự nhưng nó không phải là một độ đo không tương tự metric vì (1.7) không thoả mãn. Thật vậy, hãy xét các vector D_i, D_j có phần tử chung, chẳng hạn: $\{x_1, x_2\}$ và $\{x_1, x_4\}$ thì

$$d_{\min}^{ss}(\{x_1, x_2\}, \{x_1, x_4\}) = 0$$

trong khi chúng là hai tập khác nhau.

Một cách trực giác thì các định nghĩa trên cho thấy các DM là “ngược” với các SM. Chẳng hạn, nếu d là một DM (metric) với $d(x, y) > 0, \forall x, y \in X$ thì $s = a/d$ với $a > 0$ là một SM (metric); $s = d_{\max} + k - d$ cũng là một SM (metric), với d_{\max} là khoảng cách lớn nhất trong mọi cặp phần tử của X . Các nhận xét tương tự cũng đúng cho độ đo tương tự và không tương tự giữa các tập vector.

Trong phần tiếp theo, ta sẽ kí hiệu b_{\max} và b_{\min} lần lượt là các giá trị *max* và *min* của tập dữ liệu X . (khoảng cách lớn nhất và nhỏ nhất trong mọi cặp phần tử của X).

1.7.2.2. CÁC ĐỘ ĐO GẦN GŨI GIỮA 2 ĐIỂM

a. Các vector thực

- Các độ đo không tương tự:

+ *Các DM metric có trọng số L_p* :

$$d_p(x, y) = \left(\sum_{i=1}^{\ell} w_i |x_i - y_i|^p \right)^{1/p} \quad (1.14)$$

$w_i \geq 0, \forall i \in \{1, \dots, \ell\}$

w_i là hệ số trọng số thứ i , chúng được sử dụng chủ yếu với các vector giá trị thực.

- Nếu $w_i = 1, \forall i \in \{1, \dots, \ell\}$ ta có các DM metric không trọng số.

- Nếu $p = 2$ ta có khoảng cách Euclid.

- Các DM metric có trọng số L_2 được tổng quát hoá như sau:

$$d(x, y) = \sqrt{(x - y)^T B (x - y)} \quad (1.15)$$

Với B là ma trận đối xứng xác định dương.

- Chuẩn Manhattan L_1 (có trọng số):

$$d_1(x, y) = \sum_{i=1}^{\ell} w_i |x_i - y_i| \quad (1.16)$$

- Chuẩn L_∞ (có trọng số):

$$d_\infty(x, y) = \max_{1 \leq i \leq \ell} \{w_i |x_i - y_i|\} \quad (1.17)$$

Chuẩn L_1 và L_∞ có thể được xem như ước lượng trên và ước lượng dưới của chuẩn L_2 , thật vậy:

$$d_\infty(x, y) \leq d_2(x, y) \leq d_1(x, y)$$

Khi $\ell = 1$ thì tất cả các chuẩn L_p trùng nhau.

Dựa vào các DM trên ta có thể định nghĩa các SM tương ứng là

$$s_p(x, y) = b_{\max} - d_p(x, y)$$

+ Các DM khác là:

$$d_G(x, y) = -\log_{10} \left(1 - \frac{1}{\ell} \sum_{j=1}^{\ell} \frac{|x_j - x_p|}{b_j - a_j} \right) \quad (1.18)$$

ở đây, b_j và a_j là các giá trị lớn nhất và nhỏ nhất của đặc trưng thứ j . Dễ dàng thấy đây là một DM metric và nó không chỉ dựa trên x và y mà còn dựa vào toàn bộ tập X .

+ Một độ đo không tương tự nữa là:

$$d_Q(x, y) = \sqrt{\frac{1}{\ell} \sum_{j=1}^{\ell} \left(\frac{x_j - y_j}{x_j + y_j} \right)^2} \quad (1.19)$$

• Các độ đo tương tự

+ Tích nội:

$$s_{inner}(x, y) = x^T y = \sum_{i=1}^{\ell} x_i y_i$$

Trong phần lớn trường hợp, tích nội được dùng khi các vector được chuẩn hoá sao cho chúng có cùng độ dài a . Vì vậy, cận trên và cận dưới của tích nội là $+a^2$ và $-a^2$, và nó phụ thuộc vào góc giữa x và y . Một độ đo không tương tự tương ứng với tích nội là:

$$d_{inner}(x, y) = b_{\max} - s_{inner}(x, y)$$

+ Độ đo Tanimoto:

Được dùng cho cả các vector có giá trị thực cũng như rời rạc:

$$s_T(x, y) = \frac{x^T y}{\|x\|^2 + \|y\|^2 - x^T y} \quad (1.20)$$

Sau khi biến đổi:

$$s_T(x, y) = \frac{1}{1 + \frac{(x-y)^T(x-y)}{x^T y}}$$

Như vậy độ đo Tanimoto giữa x và y tỷ lệ nghịch với khoảng cách Euclid bình phương giữa x và y chia cho tích nội giữa chúng. Nếu các vector x và y được chuẩn hoá để chúng có cùng độ dài a thì biểu thức sau cùng dẫn tới:

$$s_T(x, y) = \frac{1}{-1 + 2 \frac{a^2}{x^T y}}$$

Trong trường hợp này, độ đo Tanimoto tỷ lệ nghịch với $a^2 / x^T y$. Vì thế nếu coi tích nội giữa hai vector biểu thị mức độ liên quan giữa chúng thì nếu hai vector càng liên quan đến nhau, độ đo Tanimoto giữa chúng càng lớn.

+ Một độ đo khác cũng hay dùng được định nghĩa là:

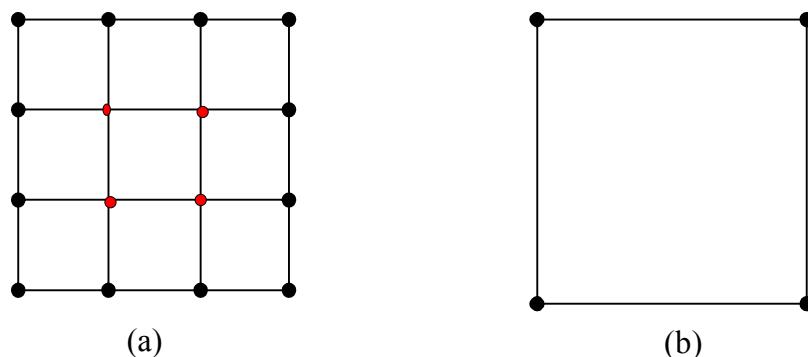
$$s_c(x, y) = 1 - \frac{d_2(x, y)}{\|x\| + \|y\|} \quad (1.21)$$

Dễ thấy, độ đo này đạt $\max = 1$ khi $x = y$ và đạt $\min = 0$ khi $x = -y$.

b. Các vector rời rạc

Bây giờ sẽ xét trường hợp các vector x mà các đặc trưng của nó lấy giá trị trong các tập rời rạc hữu hạn $F = \{0, 1, \dots, k-1\}$ với k là một số nguyên dương.

Rõ ràng là có tới k^ℓ vector $x \in F^\ell$. Chúng nằm trên các đỉnh của một lưới ℓ -chiều như hình 1.6a. Khi $k = 2$, lưới này rút lại thành một siêu lập phương đơn vị H_2 (hình 1.6b).



(a) Lưới 2 chiều với $k = 4$.

(b) Siêu lập phương H_2 (hình vuông).

Hình 1-4. Phân bố các vector rời rạc trên lưới ℓ - chiều

Xét $x, y \in F^\ell$ và đặt: $A(x, y) = [a_{ij}]$, $i, j = 0, 1, \dots, k-1$ (1.22)

Là một ma trận $k \times k$. Các phần tử a_{ij} là số vị trí mà vector đầu tiên có ký hiệu i và phần tử tương ứng của vector thứ hai có ký hiệu j ; $i, j \in F$. Ma trận này gọi là bảng

ngẫu nhiên. Hầu hết các độ đo gần gũi giữa hai vector có giá trị rời rạc có thể biểu diễn qua sự kết hợp các phần tử của ma trận $A(x, y)$.

- **Các độ đo không tương tự**

- + **Khoảng cách Hamming:**

Được định nghĩa là số vị trí hai vector khác nhau. Sử dụng ma trận A , ta có thể định nghĩa khoảng cách Hamming là:

$$d_H(x, y) = \sum_{i=0}^{k-1} \sum_{j=0, j \neq i}^{k-1} a_{ij} \quad (1.23)$$

Nghĩa là ta chỉ việc tính tổng các vị trí không phải đường chéo của A . Khi $k = 2$, vector $x \in F^\ell$ là một vector nhị phân và khoảng cách Hamming trở thành:

$$d_H(x, y) = \sum_{i=1}^{\ell} (x_i + y_i - 2x_i y_i) = \sum_{i=1}^{\ell} (x_i - y_i)^2 \quad (1.24)$$

Khi $x \in F_1^\ell$ mà $F_1 = \{-1, 1\}$. x được gọi là *vector lưỡng cực* và khoảng cách Hamming là:

$$d_H(x, y) = 0,5 \left(\ell - \sum_{i=1}^{\ell} x_i y_i \right) \quad (1.25)$$

Độ đo tương tự tương ứng là:

$$s_H(x, y) = d_{\max} - d_H(x, y)$$

- + **Khoảng cách ℓ_1 :**

Được định nghĩa trong trường hợp các vector có giá trị liên tục:

$$d_1(x, y) = \sum_{i=1}^{\ell} |x_i - y_i| \quad (1.26)$$

Khoảng cách này và khoảng cách Hamming trùng nhau khi các vector có giá trị nhị phân.

- **Các độ đo tương tự:**

Một độ đo tương tự được sử dụng rộng rãi cho các vector rời rạc là độ đo Tanimoto. Độ đo này yêu cầu phải tính tất cả các cặp toạ độ tương ứng của x và y trừ những cặp mà cả hai toạ độ đều bằng không. Điều này rất dễ hiểu nếu ta coi giá trị toạ độ thứ i của x như là độ đo sở hữu của x đối với đặc trưng thứ i , vì vậy cặp $(0, 0)$ là ít quan trọng hơn tất cả các cặp còn lại.

Bây giờ ta định nghĩa:

$$n_x = \sum_{i=1}^{k-1} \sum_{j=0}^{k-1} a_{ij} \quad \text{và} \quad n_y = \sum_{i=0}^{k-1} \sum_{j=1}^{k-1} a_{ij} \quad \text{thì}$$

n_x là toạ độ khác không của x .

n_y là toạ độ khác không của y .

Khi đó độ đo Tanimoto được định nghĩa là:

$$s_T(x, y) = \frac{\sum_{i=1}^{k-1} a_{ii}}{n_x + n_y - \sum_{i=1}^{k-1} \sum_{j=1}^{k-1} a_{ij}} \quad (1.27)$$

+ Các hàm tương tự khác giữa x và y được định nghĩa thông qua ma trận A . Một số hàm thì quan tâm đến số vị trí mà hai vector giống nhau nhưng khác không. Trong khi các hàm khác tính tất cả các vị trí của hai vector giống nhau.

Hàm tương tự trong trường hợp đầu là:

$$\text{và } \frac{\sum_{i=1}^{k-1} a_{ii}}{\ell - a_{00}} \quad (1.28)$$

Còn các trường hợp sau là:

$$\frac{\sum_{i=0}^{k-1} a_{ii}}{\ell} \quad (1.30)$$

c. Các vector giá trị hỗn hợp

Trong thực tế, ta cũng hay gặp các trường hợp khi không phải tất cả các đặc trưng của vector đặc trưng đều có cùng giá trị thực hoặc rời rạc. Có ba cách khắc phục:

- *Cách 1:*

Dùng các độ đo gần gũi cho vector thực vì các vector rời rạc có thể được so sánh một cách chính xác theo nghĩa các độ đo gần gũi cho vector thực, trong khi điều ngược lại nói chung không cho kết quả hợp lý. Độ đo được đề xuất tốt cho trường hợp này là khoảng cách ℓ_1

- *Cách 2:*

Cách này chuyển các đặc trưng giá trị thực thành rời rạc. Nếu một đặc trưng x_i lấy giá trị trong khoảng $[a, b]$ ta chia đoạn này thành k đoạn con. Nếu giá trị x_i nằm trong đoạn con thứ r thì $x_i := r - 1$. Kết quả là ta có một vector rời rạc và có thể dùng bất kỳ độ đo rời rạc nào đã nói trước đây.

- *Cách 3:*

Cho x, y là hai vector ℓ -chiều có giá trị hỗn hợp. Khi đó hàm tương tự giữa hai vector được định nghĩa là:

$$s(x, y) = \frac{\sum_{q=1}^{\ell} s_q(x, y)}{\sum_{q=1}^{\ell} w_q} \quad (1.31)$$

với $s_q(x, y)$ là độ tương tự giữa các đặc trưng thứ q của x, y và w_q là trọng số tương ứng với đặc trưng thứ q .

+ w_q :

- Nếu ít nhất một trong hai đặc trưng thứ q của x, y là không xác định thì $w_q = 0$.
- Nếu đặc trưng thứ q của x, y là giá trị nhị phân và cả hai đều = 0 thì $w_q = 0$.
- Các trường hợp còn lại: $w_q = 1$
- Nếu tất cả các $w_q = 0$ thì $s(x, y)$ là không xác định.

+ $s_q(x, y)$:

- Trường hợp x, y là nhị phân:

$$s_q(x, y) = 1 \text{ nếu } x_q = y_q = 1; \text{ ngược lại } s_q(x, y) = 0. \quad (1.32)$$

- Trường hợp x, y có giá trị danh nghĩa hoặc thứ tự:

$$s_q(x, y) = 1 \text{ nếu } x_q \text{ và } y_q \text{ có cùng giá trị; ngược lại } s_q(x, y) = 0.$$

- Trường hợp x, y có giá trị được đo theo khoảng hoặc theo tỷ lệ:

$$s_q(x, y) = 1 - \frac{|x_q - y_q|}{r_q} \quad (1.33)$$

r_q là độ dài của khoảng chứa giá trị của các đặc trưng thứ q .

d. Các độ đo mờ

Trong phần này chúng ta sẽ xét các vector thực x, y mà những đặc trưng của nó có giá trị nằm trong đoạn $[0, 1]$.

- x_i càng gần 1 thì càng chắc chắn để khảng định x_i là đặc trưng của x .
- x_i càng gần 0 thì càng chắc chắn để khảng định x_i không là đặc trưng của x
- x_i càng gần 0,5 thì càng thiếu chắc chắn để khảng định x_i có là đặc trưng của x hay không.
- $x_i = 0,5$ thì không thể khảng định x_i là đặc trưng của x hay không. Đây là sự tổng quát của logic nhị phân. Nhưng ở logic nhị phân có sự tuyệt đối chắc chắn về sự xuất hiện của một sự kiện còn trong logic mờ thì không, độ chắc chắn thể hiện trong giá trị của x_i .

Sự tương đương giữa 2 biến nhị phân

$$(a \leftrightarrow b) = ((\text{NOT } a) \text{ AND } (\text{NOT } b)) \text{ OR } (a \text{ AND } b) \quad (1.34)$$

Một điểm thú vị là toán tử AND (OR) giữa hai biến nhị phân có thể được xem như toán tử *min* (*max*) trên chúng. Còn toán tử NOT được xem như $1 - a$.

Thay vào (1.34) ta có độ đo tương tự giữa hai biến có giá trị thực trong đoạn $[0, 1]$ là:

$$s(x_i, y_i) = \max \{ \min \{1 - x_i, 1 - y_i\}, \min \{x_i, y_i\} \} \quad (1.35)$$

Vì vậy, ta dễ dàng định nghĩa độ tương tự (mờ) giữa hai vector x, y trong không gian ℓ -chiều là:

$$s_F^q(x, y) = \left(\sum_{i=1}^{\ell} s(x_i, y_i)^q \right)^{\frac{1}{q}} \quad (1.36)$$

• Nhận xét:

- + Giá trị \max và \min của s_F là $\ell^{1/q}$ và $\frac{1}{2}\ell^{1/q}$.
- + Khi $q \rightarrow +\infty$, $s_F(x, y) = \max_{1 \leq i \leq \ell} s(x_i, y_i)$
- + Khi $q = 1$, $s_F(x, y) = \sum_{i=1}^{\ell} s(x_i, y_i)$

e. Dữ liệu bị thiếu

Nếu xảy ra trường hợp một vài đặc trưng của vector đặc trưng không xác định, ta có thể dùng các kỹ thuật sau:

- Loại bỏ tất cả các vector bị thiếu đặc trưng, cách này thường được sử dụng khi số vector loại này là nhỏ so với tổng số các vector đặc trưng.
- Với đặc trưng thứ i , tìm giá trị trung bình dựa trên giá trị tương ứng của tất cả các vector đặc trưng của X . Sau đó thay thế giá trị này cho các giá trị không xác định.
- Với mọi cặp đặc trưng x_i, y_i của vector đặc trưng x, y ta định nghĩa b_i như sau:

$$b_i = \begin{cases} 0 : \text{Nếu cả } x_i, y_i \text{ đều có sẵn (đặc trưng } x_i, y_i \text{ không bị mất);} \\ 1 : \text{Ngược lại} \end{cases} \quad (1.37)$$

và độ đo gần gũi giữa x và y là:

$$\phi(x, y) = \frac{\ell}{\ell - \sum_{i=1}^{\ell} b_i} \sum_{\forall i: b_i=0} \phi(x_i, y_i) \quad (1.38)$$

trong đó $\phi(\cdot)$ là độ đo gần gũi giữa hai giá trị vô hướng.

- Tìm các độ đo gần gũi trung bình, $\phi_{avg}(i)$ giữa tất cả các vector đặc trưng trong X theo tất cả các thành phần i . Với các vector không có đặc trưng thứ i thì bỏ qua vector này khi tính $\phi_{avg}(i)$. Đặt:

$$\psi(x_i, y_i) = \begin{cases} \phi_{avg}(i) & : \text{nếu một trong } x_i, y_i \text{ không có sẵn} \\ \phi(x_i, y_i) & : \text{ngược lại} \end{cases} \quad (1.39)$$

Khi đó $\wp(x, y) = \sum_{i=1}^{\ell} \psi(x_i, y_i)$ (1.40)

1.7.2.3. HÀM GẦN GŨI GIỮA MỘT ĐIỂM VÀ MỘT TẬP

Trong nhiều sơ đồ phân cụm, một vector x được gán vào một cụm C bởi việc tính độ gần gũi giữa x và C , $\wp(x, C)$.

Có hai cách định nghĩa $\wp(x, C)$

- Theo cách thứ nhất, tất cả các điểm của C góp phần vào $\wp(x, C)$, đó là:

- Hàm gần gũi max:

$$\wp_{\max}^{\text{ps}}(x, C) = \max_{y \in C} \wp(x, y) \quad (1.41)$$

- Hàm gần gũi min:

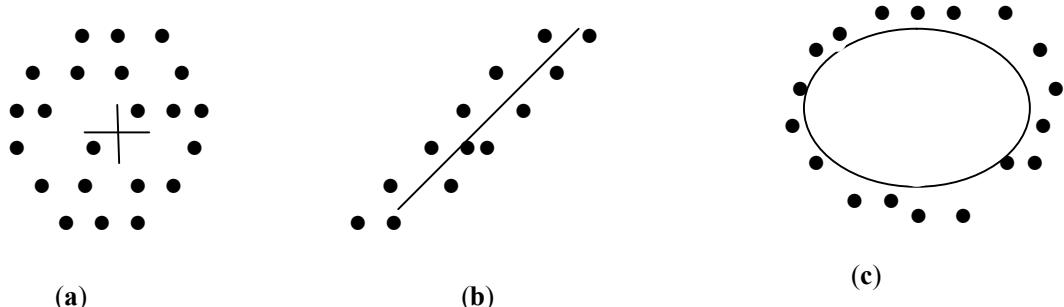
$$\wp_{\min}^{\text{ps}}(x, C) = \min_{y \in C} \wp(x, y) \quad (1.42)$$

- Hàm gần gũi trung bình:

$$\wp_{avg}^{\text{ps}}(x, C) = \frac{1}{n_C} \sum_{y \in C} \wp(x, y) \quad (1.43)$$

- Theo cách thứ 2, C có một đại diện và độ đo gần gũi giữa x và C là độ đo gần gũi giữa x và đại diện của C . Có ba loại đại diện được dùng phổ biến là:

- Đại diện điểm: thích hợp cho các cụm chật (xem hình 1.7a)
- Đại diện siêu phẳng: thích hợp cho các cụm có dạng đường thẳng hay tuyến tính (xem hình 1.7b).
- Đại diện siêu cầu: thích hợp cho các cụm có dạng cầu (xem hình 1.7c).



Hình 1-5. Các loại cụm và đại diện của nó

a. Các đại diện là điểm

- Vector trung bình (điểm trung bình)

$$m_p = \frac{1}{n_C} \sum_{y \in C} y \quad (1.44)$$

Có thể không thích hợp với không gian rời rạc vì m_p có thể nằm ngoài không gian F^t . (n_C là số vector trong tập C)

- Tâm trung bình $m_C \in C$ định nghĩa là:

$$\sum_{y \in C} d(m_C, y) \leq \sum_{y \in C} d(z, y), \quad \forall z \in C \quad (1.45)$$

Với d là một độ đo không tương tự.

- Tâm median $m_{med} \in C$ định nghĩa là:

$$med(d(m_{med}, y) | y \in C) \leq med(d(z, y) | y \in C), \quad \forall z \in C \quad (1.46)$$

Thường được dùng khi độ đo gần gũi giữa hai điểm không là một metric.

Trong đó, T là tập q giá trị vô hướng và $med(T)$ là số nhỏ nhất trong T sao cho $med(T) \geq$ số thứ tự $[(q+1)/2]$ của T . Một cách để xác định $med(T)$ là xếp các phần tử của T tăng dần và chọn lấy phần tử thứ $[(q + 1)/2]$.

b. Các đại diện siêu phẳng:

Phương trình tổng quát của một siêu phẳng H là:

$$\sum_{j=1}^{\ell} a_j x_j + a_0 = a^T x + a_0 = 0 \quad (1.47)$$

Trong đó, a và x là các vector ℓ -chiều.

Khoảng cách từ một điểm x tới siêu phẳng H là:

$$d(x, H) = \min_{z \in H} d(x, z) \quad (1.48)$$

Khi dùng khoảng cách Euclid ta có:

$$d(x, H) = \frac{|a^T x + a_0|}{\sqrt{\sum_{j=1}^{\ell} a_j^2}} \quad (1.49)$$

c. Các đại diện siêu cầu

Phương trình tổng quát của một siêu cầu Q là:

$$(x - c)^T (x - c) = r^2 \quad (1.50)$$

Trong đó, c và r lần lượt là tâm và bán kính của siêu cầu.

Khoảng cách từ một điểm tới một siêu cầu là:

$$d(x, Q) = \min_{z \in Q} d(x, z) \quad (1.51)$$

1.7.2.4. CÁC HÀM GẦN GŨI GIỮA HAI TẬP

- Hàm gần gũi *max*

$$\wp_{\max}^{ss}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \wp(x, y) \quad (1.52)$$

Nếu \wp là một độ đo không tương tự thì hàm gần gũi *max* không là một độ đo vì nó không thoả (1.5). Ngược lại, nếu \wp là một độ đo tương tự thì hàm gần gũi *max* là một độ đo nhưng không là metric.

- Hàm gần gũi *min*

$$\wp_{\min}^{ss}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \wp(x, y) \quad (1.53)$$

Nếu \wp là một độ đo tương tự thì hàm gần gũi *min* không là một độ đo. Ngược lại, nếu \wp là một độ đo không tương tự thì hàm gần gũi *min* là một độ đo nhưng không metric.

- Hàm gần gũi *trung bình*

$$\wp_{avg}^{ss}(D_i, D_j) = \frac{1}{n_{D_i} n_{D_j}} \sum_{x \in D_i} \sum_{y \in D_j} \wp(x, y) \quad (1.54)$$

hàm gần gũi trung bình không là một độ đo cả khi \wp là một độ đo.

- Hàm gần gũi giữa các giá trị đại diện

$$\wp_{mean}^{ss}(D_i, D_j) = \wp(m_{D_i}, m_{D_j}) \quad (1.55)$$

Hàm này là một độ đo khi \wp là một độ đo.

- Hàm khác:

$$\wp_e^{ss}(D_i, D_j) = \sqrt{\frac{n_{D_i} n_{D_j}}{n_{D_i} + n_{D_j}}} \wp(m_{D_i}, m_{D_j}) \quad (1.56)$$

Chú ý rằng, các độ đo gần gũi giữa hai tập được xây dựng trên các độ đo gần gũi giữa các điểm. Một cách trực giác, có thể hiểu rằng các lựa chọn hàm gần gũi khác nhau sẽ dẫn tới các kết quả phân cụm khác nhau. Hơn nữa, nếu ta sử dụng các độ đo gần gũi khác nhau giữa các điểm thì ngay cả khi ta sử dụng cùng một độ đo gần gũi giữa các tập, nói chung, cũng đưa đến các kết quả phân cụm khác nhau. Cách duy nhất để đạt được sự phân cụm một cách hợp lý tập dữ liệu là bằng cách thử - sai, và tất nhiên là bằng ý kiến của các chuyên gia trong lĩnh vực ứng dụng.

Chương 2.

CÁC THUẬT TOÁN PHÂN CỤM TUẦN TỤ

Trong chương trước, ta đã giới thiệu một số độ đo gần gũi. Mỗi một loại độ đo (tương tự hoặc không tương tự) phù hợp với một loại cụm mà ta cần phát hiện. Chương này sẽ tập trung vào thuật toán phân cụm tuần tự.

2.1. Số các cách phân cụm có thể

Với một thời gian và tài nguyên cho trước, cách tốt nhất để phân cụm là tìm ra tất cả các cách có thể và chọn lựa cách "để thấy nhất" theo tiêu chuẩn phân cụm đã chọn trước. Nhưng cách này là không thể thậm chí với cả các giá trị trung bình của số vector đặc trưng.

Ký hiệu $S(N, m)$ là số cách phân cụm N vector về m cụm, ta có các tính chất sau:

- $S(N, 1) = 1$
- $S(N, N) = 1$
- $S(N, m) = 0$ với $m > N$.

Ký hiệu L_{N-1}^k là danh sách tất cả các cách phân $N - 1$ vector về k cụm, với $k = m, m - 1$. Vector thứ N hoặc được thêm vào một trong các cụm của bất kỳ thành viên nào thuộc L_{N-1}^k , hoặc hình thành nên một cụm mới. Vì vậy ta có thể viết:

$$S(N, m) = m \cdot S(N - 1, m) + S(N - 1, m - 1) \quad (2.1)$$

Nghiệm của phương trình trên là số Stirling được cho bởi công thức:

$$S(N, m) = \frac{1}{m!} \sum_{i=0}^m (-1)^{m-i} \binom{m}{i} i^N \quad (2.2)$$

Đặc biệt, khi $m = 2$ ta có:

$$S(N, 2) = 2^{N-1} - 1 \quad (2.3)$$

Ví dụ 2.1: Giả sử $X = \{x_1, x_2, x_3\}$. Chúng ta sẽ tìm tất cả các cách để phân X thành hai cụm. Dễ suy ra rằng $L_2^1 = \{\{x_1, x_2\}\}$ và $L_2^2 = \{\{x_1\}, \{x_2\}\}$. Thay vào công thức (2.1)

ta tìm được $S(3, 2) = 2 \times 1 + 1 = 3$. Thật vậy, các cụm tạo ra là:

$$L_3^2 = \{\{x_1, x_3\}, \{x_2\}\}; \{\{x_1\}, \{x_2, x_3\}\}; \{\{x_1, x_2\}, \{x_3\}\}$$

Một vài con số từ (2.2) là:

- $S(15, 3) = 2\ 375\ 101$
- $S(20, 4) = 45\ 232\ 115\ 901$
- $S(25, 8) = 690\ 223\ 721\ 118\ 368\ 580$
- $S(100, 5) = 10^{68}$

Các con số này chứng minh cho lời khăng định trên rằng việc tìm tất cả các cách phân cụm là không thể.

2.2. Thuật toán phân cụm tuần tự - BSAS

Giả sử, tất cả các vector chỉ được tham gia một lần duy nhất vào thuật toán và số lượng cụm là không được biết trước.

Ký hiệu $d(x, C)$ là khoảng cách từ điểm x đến cụm C .

Người dùng cần vào các tham số là:

- Ngưỡng không tương tự Θ .
- Số cụm lớn nhất cho phép q .

Ý tưởng cơ bản của thuật toán là: mỗi vector sẽ được hoặc đưa vào một cụm đã có hoặc tạo ra một cụm mới dựa vào khoảng cách của nó tới các cụm đã có sẵn.

Gọi m là số cụm mà thuật toán tạo ra, ta có sơ đồ thuật toán sau:

Sơ đồ thuật toán phân cụm tuần tự cơ sở
(Basic Sequential Algorithmic Scheme - BSAS)

```

1.    $m = 1; C_m = \{x_1\}$                                 // Khởi tạo
2. For i = 2 to N do
    Begin
        a. Tìm  $C_k : d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$ 
        b. If ( $d(x_i, C_k) > \Theta$ ) AND ( $m < q$ ) then
            i.    $m = m + 1; C_m = \{x_i\}$           // Tạo ra một cụm mới
        c. Else
            ii.   $C_k = C_k \cup \{x_i\}$            // Thêm  $x_i$  vào cụm gần nhất
            iii. Cập nhật các đại diện (nếu cần thiết)
    - End {if}
End {For}

```

Các lựa chọn độ đo $d(x, C)$ khác nhau dẫn tới kết quả của thuật toán cũng khác nhau. Khi cụm C được đại diện bằng một vector, $d(x, C)$ trở thành:

$$d(x, C) = d(x, m_C) \quad (2.4)$$

ở đây m_C là đại diện của cụm C . Nếu sử dụng vector trung bình làm đại diện thì việc cập nhật vector này có thể lặp lại theo công thức sau:

$$m_{C_k}^{new} = \frac{(n_{C_k}^{new} - 1).m_{C_k}^{old} + x}{n_{C_k}^{new}} \quad (2.5)$$

$n_{C_k}^{new}$ là số vector của cụm C_k sau khi thêm vector x vào C_k

$m_{C_k}^{new}$ ($m_{C_k}^{old}$) là một vector đại diện cho C_k sau khi (trước khi) thêm vector x vào C_k

Các thuật toán ở đó mỗi cụm được đại diện bởi một vector thành viên của cụm được gọi là các thuật toán dựa trên tiêu chuẩn phân cụm tổng quát và các thuật toán mà tất cả các vector được sử dụng bằng vector đại diện của nó (vector trung bình chẵng hạn) được gọi là các thuật toán dựa trên tiêu chuẩn phân cụm cục bộ.

Không khó để thấy rằng thứ tự các vector tham gia vào thuật toán BSAS có tầm quan trọng trong các kết quả phân cụm. Thứ tự các vector khác nhau có thể dẫn tới các kết quả phân cụm khác nhau.

Một nhân tố quan trọng khác ảnh hưởng tới kết quả phân cụm là sự chọn lựa ngưỡng Θ . Giá trị ngưỡng này ảnh hưởng trực tiếp tới số cụm được sinh ra. Nếu Θ quá nhỏ, thuật toán sẽ tạo ra các cụm không mong muốn; mặt khác, nếu Θ quá lớn, thuật toán sẽ tạo ra ít cụm. Trong cả hai trường hợp, số cụm sinh ra đều không phù hợp với tập dữ liệu. Khi số cụm lớn nhất được phép q là không giới hạn, chúng ta để tuỳ thuật toán “quyết định” về số các cụm.



Hình 2-1. Sự phụ thuộc của số cụm được tạo ra và số cụm lớn nhất được phép q.

Xét ví dụ hình 2.1, ở đây ba cụm là chặt và khá độc lập được tạo thành bởi các điểm của X . Nếu $q = 2$, thuật toán BSAS sẽ không thể tìm ra ba cụm. Trong trường hợp đó, hai cụm bên phải sẽ gộp thành một cụm. Mặt khác, nếu q không giới hạn, thuật toán BSAS có thể đưa ra ba cụm (với một lựa chọn xấp xỉ Θ). Tuy nhiên, ràng buộc q trở nên cần thiết khi phải phân chia thực hiện mà ở đó các tài nguyên tính toán bị giới hạn.

• Nhận xét

- Sơ đồ BSAS có thể sử dụng độ đo tương tự thay cho độ đo không tương tự với sửa đổi nhỏ; nghĩa là, toán tử *min* được thay bằng toán tử *max*.
- Thuật toán BSAS, với phân cụm theo điểm đại diện, có khuynh hướng hình thành nên các cụm chặt. Do đó, nó không thích hợp nếu cần phải đưa ra nhiều loại cụm khác nhau.

2.3. Ước lượng số cụm

Phần này sẽ mô tả một phương pháp đơn giản để xác định số cụm. Phương pháp này thích hợp với thuật toán BSAS cũng như các thuật toán khác và số cụm sinh ra không phụ thuộc vào tham số đầu vào. Kí hiệu Θ là thuật toán BSAS với một ngưỡng không tương tự Θ .

- **For $\Theta = a$ to b do step c**

- Thực hiện s lần thuật toán BSAS (Θ), mỗi lần nhập dữ liệu vào theo thứ tự khác nhau.
- Ước tính số cụm, m_Θ , là các kết quả thường xuyên xuất hiện nhất trong s lần chạy thuật toán BSAS (Θ).

- **Next Θ .**

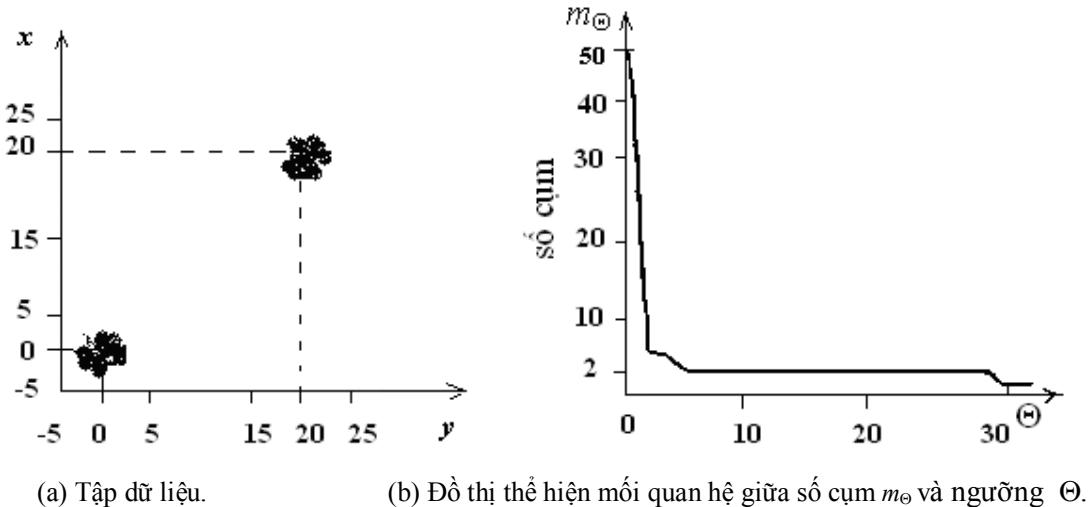
Giá trị a và b tương ứng là mức độ không tương tự nhỏ nhất và lớn nhất giữa tất cả các cặp vector trong X , nghĩa là $a = \min\{d(x_i, x_j)\}$ và $b = \max\{d(x_i, x_j)\} \forall i, j = 1, \dots, N$. Lựa chọn c trực tiếp bị ảnh hưởng bởi lựa chọn $d(x, C)$. Nếu s càng lớn, lấy mẫu thống kê càng rộng thì các kết quả có độ chính xác càng cao.

Tiếp theo, ta vẽ đồ thị biểu diễn mối quan hệ giữa số cụm m_Θ và ngưỡng Θ . Đồ thị này có một số miền phẳng (flat regions). Chúng ta ước tính số cụm tương ứng với số miền phẳng rộng nhất. Ta mong đợi rằng các miền phẳng ứng với các vector tạo thành các cụm chặt khá độc lập, đó là số cụm mong muốn.

Có thể giải thích vấn đề này bằng trực giác như sau: Giả sử rằng tập dữ liệu tạo thành hai cụm chặt và khá độc lập C_1 và C_2 . Gọi khoảng cách nhỏ nhất giữa hai vector trong C_1 (C_2) là r_1 (r_2) và giả sử rằng $r_1 < r_2$. Lấy r ($> r_2$) là số nhỏ nhất trong tất cả các khoảng cách $d(x_i, x_j)$ với $x_i \in C_1$ và $x_j \in C_2$. Rõ ràng, với $\Theta \in [r_2, r - r_2]$, số cụm tạo bởi thuật toán BSAS là hai. Hơn nữa, nếu $r \gg r_2$ thì khoảng đó rộng, và do đó nó tương ứng với một miền phẳng rộng trong đồ thị. Ví dụ 2.2 mô phỏng cho nhận xét này.

Ví dụ 2.2.

Xét hai phân bố Gaussian của các vector trong không gian hai chiều với giá trị trung bình $[0,0]^T$ và $[20, 20]^T$. Ma trận covariance là $\Sigma = 0.5I$ với cả hai phân bố, ở đây I là ma trận đồng nhất 2×2 . Sinh ra 50 điểm từ mỗi phân bố (hình 2.2a). Đồ thị kết quả chỉ ra trong hình 2.2b, với $a = \min\{d_2(x_i, x_j)\}$, $b = \max\{d_2(x_i, x_j)\} \forall x_i, x_j \in X$ và $c \approx 0.3$. Có thể thấy rằng miền phẳng lớn nhất có tung độ $m_\Theta = 2$, đó là số cụm ẩn dấu.



Hình 2.2. Đồ thị ước lượng số cụm

Nhu đã đề cập ở trên, giả sử rằng các vector đặc trưng tạo thành các cụm; nếu không, phương pháp này không thể áp dụng được. Hơn nữa, nếu các vector tạo thành các cụm chặt, không tách biệt, phương pháp này có thể đưa ra các kết quả không đáng tin cậy.

Trong một số trường hợp, nên xét tất cả số cụm, m_Θ , ứng với tất cả các miền phẳng có kích thước đáng kể trong đồ thị. Ví dụ, nếu có ba cụm, hai cụm đầu nằm gần nhau và cụm thứ ba nằm xa hơn, miền phẳng nhất có thể xuất hiện với $m_\Theta = 2$ và miền phẳng thứ hai với $m_\Theta = 3$. Nếu bỏ qua miền phẳng thứ hai, sẽ mất nghiệm là ba cụm.

2.4. Sửa đổi thuật toán BSAS - Thuật toán MBSAS

Nhu đã đề cập, tư tưởng cơ bản của thuật toán BSAS là mỗi vector input x được gán vào một cụm đã tạo từ trước hoặc thành lập nên cụm mới. Do đó, một quyết định cho vector x sẽ đạt được trước khi hình thành cụm cuối cùng, nó được xác định sau khi tất cả vector đã được xét. Sau đây ta sẽ tinh chế thuật toán BSAS,

còn được gọi là thuật toán BSAS sửa đổi (modified BSAS-MBSAS), để khắc phục hạn chế đó. Chi phí phải trả cho việc tinh chỉnh này là tất cả các vector của X được đưa vào thuật toán hai lần.

Thuật toán gồm hai pha. Pha thứ nhất quyết định số cụm sẽ được tạo thành, qua việc gán một số vector của X vào các cụm. Pha thứ 2, các vector còn lại chưa được gán vào cụm nào ở pha thứ nhất tiếp tục đưa vào thuật toán và gán nó vào các cụm thích hợp. Thuật toán MBSAS viết như sau:

Sơ đồ thuật toán phân cụm tuần tự sửa đổi

(Modified Basic Sequential Algorithmic Scheme - MBSAS)

Pha 1: Xác định số cụm

- $m = 1; C_m = \{x_1\}$; // Khởi tạo
- **For** $i = 2$ **to** N
 - Tìm C_k : $d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 - **If** ($d(x_i, C_k) > \Theta$) and ($m < q$) **then**
 - $m = m + 1; C_m = \{x_i\}$ // Tạo ra một cụm mới
 - **End if**
 - **END for**

Pha 2: Phân loại mẫu

- **For** $i = 1$ **to** N
 - **If** x_i chưa được gán vào một cụm **then**
 - Tìm C_k : $d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 - $C_k = C_k \cup \{x_i\}$ // Thêm x_i vào cụm gần nhất
 - Cập nhật các vector đại diện của cụm (nếu cần thiết)
 - **End if**
- **End for**

Số cụm đã được xác định trong pha thứ nhất là cố định. Do đó, trong suốt pha thứ 2 là lựa chọn các vector còn lại để đưa vào các cụm (đã tạo ra trong pha thứ nhất).

Khi vector trung bình được sử dụng làm đại diện cho cụm, nó phải được điều chỉnh theo công thức (2.5) sau khi gán mỗi vector vào cụm.

Cũng như trường hợp BSAS, thuật toán MBSAS bị ảnh hưởng bởi thứ tự các vector được đưa vào thuật toán.

Cuối cùng, phải nói rằng sau sửa đổi nhỏ, MBSAS có thể dùng được độ đo tương tự.

2.5. Thuật toán phân cụm tuần tự hai ngưỡng - TTSAS

Như đã biết, các kết quả của thuật toán BSAS và MBSAS phụ thuộc chặt chẽ vào thứ tự các vector được đưa vào thuật toán, cũng như phụ thuộc vào giá trị ngưỡng Θ . Lựa chọn không đúng Θ có thể dẫn tới các kết quả phân cụm vô nghĩa. Một cách để khắc phục những khó khăn này là định nghĩa hai ngưỡng Θ_1 và Θ_2 ($\Theta_2 > \Theta_1$). Gọi $d(x, C)$ là mức độ tương tự của vector x tới cụm gần nhất C

- Nếu $d(x, C) < \Theta_1$ thì x được đưa vào C .
- Nếu $d(x, C) > \Theta_2$ thì hình thành cụm mới và đưa x vào cụm này.
- Nếu $\Theta_1 \leq d(x, C) \leq \Theta_2$ thì việc gán x vào một cụm nào là chưa xác định (tạm thời bỏ qua x), và phải chờ đến giai đoạn sau (sau khi xét hết lượt các vector, sẽ quay lại xét các vector đã bỏ qua).

Các giá trị nằm giữa hai ngưỡng Θ_1 và Θ_2 gọi là vùng xám (gray area)

Đặt $Clas(x)$ là một biến đánh dấu; $Clas(x) = \text{True}$ nếu x đã được phân lớp, $Clas(x) = \text{False}$ nếu x chưa được phân lớp. Đặt m là số cụm được hình thành tính đến thời điểm hiện tại.

Cur_change: Tổng số vector đã được phân lớp tính đến lần duyệt X hiện tại.

Prev_change: Tổng số vector đã được phân lớp trong các lần duyệt trước của X .

Exists_change = $|Cur_change - Prev_change|$: Kiểm tra có tồn tại hay không ít nhất một vector đã được phân lớp ở lần duyệt X hiện tại (tức là: bước lặp hiện tại trong quá trình lặp). Nếu *Exists_change* = 0 thì không vector nào được đưa vào cụm trong lần duyệt X này, vector không được phân lớp đầu tiên bị "ép buộc" hình thành một cụm mới.

Giả sử rằng không có sự giới hạn số cụm (tức là $q = N$). Sơ đồ thuật toán là:

Sơ đồ thuật toán tuần tự hai ngưỡng (TTSAS)

(The Two – Threshold Sequential Algorithmic Scheme – TTSAS)

1. $m = 0$
2. $\text{Clas}(x_i) = \text{False}, \forall x_i \in X$
3. $\text{Prev_change} = 0; \text{Cur_change} = 0; \text{Exists_change} = 0$
4. **While** < tồn tại ít nhất một vector đặc trưng x_i mà $\text{Clas}(x_i) = \text{False}$ > **do**
 - a. **For** $i = 1$ to N **do**
 - i. **if** < $\text{Clas}(x_i) = \text{False}$ AND x_i là vector đầu tiên mà ta xét trong vòng lặp While mới AND $\text{Exists_change} = 0$ > **then**
 1. $m = m + 1$ //Tạo ra một cụm mới, $\Theta_1 \leq d(x_i, C) \leq \Theta_2$
 2. $C_m = \{x_i\}; \text{Clas}(x_i) = \text{True}$
 3. $\text{Cur_change} = \text{Cur_change} + 1$
 - ii. **Else if** $\text{Clas}(x_i) = \text{False}$ **then**
 1. Find $d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 2. **If** ($d(x_i, C_k) < \Theta_1$) **then**
 - $C_k = C_k \cup \{x_i\}; \text{Clas}(x_i) = \text{True}$
 - $\text{Cur_change} = \text{Cur_change} + 1$
 3. **else if** $d(x_i, C_k) > \Theta_2$ **then**
 - $m = m + 1$ //Tạo ra một cụm mới
 - $C_m = \{x_i\}; \text{Clas}(x_i) = \text{True}$
 - $\text{Cur_change} = \text{Cur_change} + 1$
 - iii. **Else if** $\text{Clas}(x_i) = \text{True}$ **then**
 1. $\text{Cur_change} = \text{Cur_change} + 1$**End {for}**
 - b. $\text{Exists_change} = |\text{Cur_change} - \text{Prev_change}|$
 - c. $\text{Prev_change} = \text{Cur_change}$
 - d. $\text{Cur_change} = 0$**End {while}**

Điều kiện **if** đầu tiên trong vòng lặp **For** đảm bảo rằng thuật toán sẽ kết thúc sau nhiều nhất là N lần duyệt X , (N lần thực hiện của vòng lặp For) nhưng theo lý thuyết đây là thuật toán $O(N^2)$. Nếu qua một lần duyệt X mà không đưa thêm được vector nào vào các cụm đã có trước thì điều kiện này "ép buộc" vector đầu tiên trong tập các vector còn lại (chưa được gán vào cụm nào) phải hình thành nên một cụm mới.

Tuy nhiên, trong thực hành, số lần duyệt thường nhỏ hơn N . Có thể chỉ ra rằng chi phí ít nhất của sơ đồ này cũng bằng với chi phí của hai sơ đồ trước, bởi vì trường hợp tổng quát nó cần tối thiểu hai lần duyệt trên X . Hơn nữa từ việc xác định một vector phải hoãn lại cho đến khi có đủ thông tin, ta thấy rằng thuật toán này ít bị ảnh hưởng bởi thứ tự của dữ liệu.

Cũng giống như các thuật toán trước, sự lựa chọn khác nhau của độ đo không tương tự giữa một vector và một cụm dẫn tới các kết quả khác nhau. Thuật toán này cũng có khuynh hướng hình thành nên các cụm chặt, khi sử dụng phân cụm theo điểm đại diện.

☞ **Chú ý:**

Tất cả các thuật toán đó không xuất hiện trạng thái khoá chết. Nghĩa là, thuật toán không bị rơi vào trạng thái mà ở đó có các vector không thể đưa vào hoặc là một cụm đã có hoặc là hình thành cụm mới, bất kể số lần chuyển dữ liệu vào thuật toán. Thuật toán BSAS và MBSAS tương ứng sẽ kết thúc sau một, hai lần duyệt X . Trong TTSAS tránh được tình trạng thái khoá chết khi ta "ép buộc" vector đầu tiên chưa được gán vào cụm nào hình thành nên một cụm mới (nếu ở lần duyệt hiện tại không đưa thêm được vector nào vào các cụm đã có từ trước).

Ví dụ 2.3.

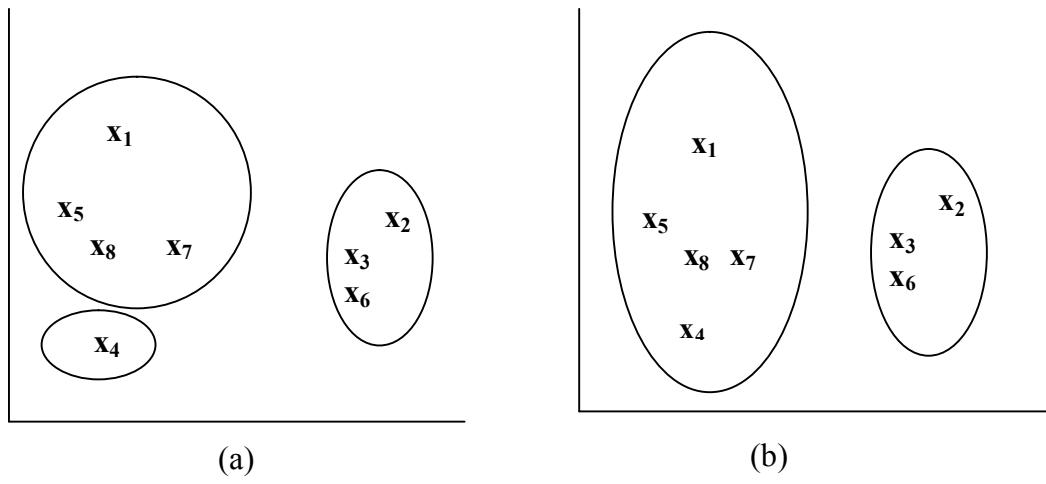
Xét các vector

$$x_1 = [2, 5]^T, x_2 = [6, 4]^T, x_3 = [5, 3]^T, x_4 = [2, 2]^T, \\ x_5 = [1, 4]^T, x_6 = [5, 2]^T, x_7 = [3, 3]^T, x_8 = [2, 3]^T.$$

Khoảng cách từ một vector x tới cụm C trong không gian Euclid là khoảng cách giữa x và vector trung bình của C .

Nếu đưa các vector theo thứ tự $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ vào thuật toán MBSAS và đặt $\Theta = 2.5$, ta được ba cụm, $C_1 = \{x_1, x_5, x_7, x_8\}$, $C_2 = \{x_2, x_3, x_6\}$ và $C_3 = \{x_4\}$ (hình 2.3a)

Mặt khác, nếu cũng đưa các vector theo thứ tự trên vào thuật toán TTSAS, với $\Theta_1 = 2.2$ và $\Theta_2 = 4$, ta được $C_1 = \{x_1, x_5, x_7, x_8, x_4\}$ và $C_2 = \{x_2, x_3, x_6\}$ (hình 2.3b). Trong trường hợp này, tất cả các vector được đưa vào các cụm trong lần duyệt X đầu tiên, ngoại trừ x_4 , nó được đưa vào cụm C_1 trong lần duyệt X thứ hai. Ở mỗi lần duyệt qua X , ta đưa được ít nhất một vector vào một cụm. Do đó, không vector nào bị ép buộc tạo thành một cụm mới bất kỳ.



Hình 2-3. Minh họa phân cụm bằng thuật toán MBSAS (a) và bằng thuật toán TTSAS (b)

Rõ ràng rằng thuật toán TTSAS cho nhiều kết quả có ý nghĩa hơn MBSAS. Tuy nhiên, chú ý rằng MBSAS cũng cho kết quả phân cụm giống trên nếu các vector đưa vào thuật toán theo thứ tự sau: $\{x_1, x_2, x_5, x_3, x_8, x_6, x_7, x_4\}$.

2.6. Giai đoạn tinh ché

Trong các thuật toán đã đưa ra, có thể xảy ra trường hợp có hai cụm được định vị rất gần nhau, và có thể hoà nhập hai cụm này thành một cụm. Các trường hợp như thế này không thể thực hiện bằng các thuật toán đó. Một cách để giải quyết vấn đề này là chạy thủ tục trộn đơn giản sau đây sau khi kết thúc các sơ đồ phân cụm.

Gọi M là tham số người sử dụng định nghĩa để xác định độ gần gũi $d(C_i, C_j)$ của hai cụm C_i và C_j .

Thủ tục trộn

1. **(A)** Tìm C_i, C_j ($i < j$) sao cho $d(C_i, C_j) = \min_{k, r=1 \dots m, k \neq r} d(C_k, C_r)$
2. **if** $d(C_i, C_j) \leq M$ **then**
 - a. Trộn C_i, C_j vào C_i và khử C_j
 - b. Cập nhật các đại diện của cụm C_i (nếu phân cụm theo điểm đại diện)
 - c. Đổi tên các cụm C_{j+1}, \dots, C_m thành C_j, \dots, C_{m-1}
 - d. $m = m - 1$
 - e. Go to **(A)**
3. **Else**
 - a. Stop
4. **End {if}**

Hạn chế khác của các thuật toán tuần tự là nó phụ thuộc vào thứ tự của các vector đưa vào xử lý. Ví dụ, trong thuật toán BSAS, đầu tiên x_2 được đưa vào cụm C_1 , và sau khi kết thúc thuật toán bốn cụm được hình thành. Khi đó, có thể x_2 sẽ gần với một cụm nào đó khác cụm C_1 . Tuy nhiên, không có cách nào để chuyển x_2 tới cụm gần nó nhất. Cách đơn giản để tránh điều này là sử dụng thủ tục sắp xếp lại như sau:

Thủ tục sắp xếp lại

1. For $i = 1$ to N do

- Tìm C_j sao cho $d(x_i, C_j) = \min_{k=1..m} d(x_i, C_k)$ // Tìm xem x_i gần với cụm nào nhất
- Đặt $b(i) = j$

End {For}

2. For $j = 1$ to m do

- Đặt $C_j = \{x_i \in X : b(i) = j\}$ // $b(i)$ lưu chỉ số của cụm C_j gần với x_i nhất
- Cập nhật lại các đại diện (nếu sử dụng).

End {For}

Trong thủ tục này, $b(i)$ lưu chỉ số của cụm C_j gần với x_i nhất. Thủ tục này có thể sử dụng sau khi kết thúc các thuật toán hoặc nếu thủ tục trộn cũng được sử dụng thì nó được thực hiện sau khi thuật toán trộn kết thúc.

Một biến thể của thuật toán BSAS ứng dụng cho phân cụm theo điểm đại diện là kết hợp hai thủ tục tinh chế đã được đề xuất trong [12]. Theo thuật toán này, thay vì bắt đầu với một cụm, ta bắt đầu với $m > 1$ cụm, mỗi cụm chứa một trong m vector đầu tiên của X . Áp dụng thủ tục trộn và sau đó đưa mỗi vector còn lại vào thuật toán. Sau khi đưa vector hiện tại vào một cụm và cập nhật lại các đại diện của nó, ta lại thực hiện thủ tục trộn. Nếu khoảng cách giữa vector x_i và cụm gần nó nhất lớn hơn một ngưỡng định trước thì hình thành một cụm mới chỉ chứa x_i . Sau khi tất cả các vector đã được đưa vào thuật toán, chạy thủ tục sắp xếp lại một lần nữa. Thủ tục trộn được áp dụng $N - m + 1$ lần.

Bài tập chương 2

Bài 2.1. Chứng minh (2.3) sử dụng quy nạp

Bài 2.2. Chứng minh công thức (2.5)

Bài 2.3. Bài tập này nhằm mục đích xem xét các ảnh hưởng của thứ tự các vector khi thực hiện thuật toán BSAS và MBSAS. Xét các vector 2 chiều sau: $x_1=[1, 1]^T$, $x_2=[1, 2]^T$, $x_3=[2, 2]^T$, $x_4=[2, 3]^T$, $x_5=[3, 3]^T$, $x_6=[3, 4]^T$, $x_7=[4, 4]^T$, $x_8=[4, 5]^T$, $x_9=[5, 5]^T$, $x_{10}=[5, 6]^T$, $x_{11}=[-4, 5]^T$, $x_{12}=[-3, 5]^T$, $x_{13}=[-4, 4]^T$, $x_{14}=[-3, 4]^T$.

Xét trường hợp mỗi cụm đại diện bằng vector trung bình của nó

- (a) Thực hiện thuật toán BSAS và MBSAS khi các vector được đưa ra theo thứ tự đã cho. Sử dụng khoảng cách Euclid giữa 2 vector và lấy $\theta=\sqrt{2}$
- (b) Thay đổi thứ tự các vector $x_1, x_{10}, x_2, x_3, x_4, x_{11}, x_{12}, x_5, x_6, x_7, x_{13}, x_8, x_{14}, x_9$ và thực hiện lại các thuật toán.
- (c) Thực hiện thuật toán theo thứ tự sau: $x_1, x_{10}, x_5, x_2, x_3, x_{11}, x_{12}, x_4, x_6, x_7, x_{13}, x_{14}, x_8, x_9$
- (d) Vẽ các vector đã cho và thảo luận kết quả của các lần thực hiện đó
- (e) Trình diễn một phép phân cụm ảo (trực quan-visual). Bao nhiêu cụm được hình thành từ các vector đã cho.

Bài 2.4. Xét cài đặt của ví dụ 2.2. Chạy các thuật toán BSAS và MBSAS, với $\theta=5$, sử dụng vector trung bình đại diện cho cụm. Thảo luận các kết quả.

Bài 2.5. Đặt s là một độ đo tương tự giữa một vector và một cụm. Biểu thị các thuật toán BSAS, MBSAS, TTSAS theo s.

Chương 3.

CÁC THUẬT TOÁN PHÂN CỤM PHÂN CẤP

3.1. Giới thiệu

Các thuật toán phân cụm phân cấp là một dạng khác với các thuật toán đã mô tả trong chương trước. Đặc biệt, thay vì đưa ra một cụm đơn, thuật toán đưa ra các cụm theo quan hệ phân cấp. Các thuật toán loại này thường gặp trong khoa học xã hội và phân loại sinh vật, khảo cổ học, khoa học máy tính và trong kỹ thuật.

Trước khi mô tả tư tưởng của các thuật toán chúng ta nhắc lại:

$$X = \{x_i \mid i = 1, \dots, N\}$$

là một tập các vector ℓ -chiều để phân cụm.

$$\mathfrak{R} = \{C_j \mid j = 1..m\}$$

là một phép phân cụm ở đây $C_j \subseteq X$.

Một phép phân cụm \mathfrak{R}_1 chứa k cụm được gọi là ẩn trong phép phân cụm \mathfrak{R}_2 chứa r cụm ($r < k$) nếu mỗi cụm trong \mathfrak{R}_1 là một tập con của một tập trong \mathfrak{R}_2 và ít nhất một cụm của \mathfrak{R}_1 là một tập con đúng của \mathfrak{R}_2 . Trong trường hợp này chúng ta viết $\mathfrak{R}_1 \subset \mathfrak{R}_2$.

Ví dụ 3.1:

Phép phân cụm $\mathfrak{R}_1 = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$ là ẩn trong $\mathfrak{R}_2 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$. Mặt khác, \mathfrak{R}_1 không ẩn trong $\mathfrak{R}_3 = \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}$ và cũng không ẩn trong $\mathfrak{R}_4 = \{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}$. Rõ ràng một phép phân cụm không ẩn trong chính nó.

Các thuật toán phân cụm phân cấp đưa ra sự phân cấp của các phép phân cụm ẩn. Cụ thể hơn, các thuật toán đó gồm N bước, bằng với số vector dữ liệu. Ở mỗi bước t , một phép phân cụm mới được sinh ra dựa trên phép phân cụm được sinh ra từ bước $t-1$. Có hai loại thuật toán phân cụm chính: Thuật toán tích tụ và thuật toán phân rã.

Phép phân cụm khởi tạo \mathfrak{R}_0 của thuật toán tích tụ bao gồm N cụm, mỗi cụm chứa một phần tử của X . Bước đầu tiên, phép phân cụm \mathfrak{R}_1 được đưa ra. Nó chứa $N-1$ tập, sao cho $\mathfrak{R}_0 \subset \mathfrak{R}_1$. Thủ tục này tiếp tục cho đến phép phân cụm cuối cùng

\mathfrak{R}_{N-1} , nó chứa một tập duy nhất, đó là tập dữ liệu X . Chú ý rằng với các kết quả của phép phân cụm phân cấp ta có:

$$\mathfrak{R}_0 \subset \mathfrak{R}_1 \subset \mathfrak{R}_2 \subset \dots \subset \mathfrak{R}_{N-1}$$

Các thuật toán phân rã có chiều ngược lại. Trong trường hợp này (thuật toán phân rã) phép phân cụm khởi tạo \mathfrak{R}_0 chứa một tập duy nhất X . Ở bước đầu tiên, phép phân cụm \mathfrak{R}_1 được đưa ra. Nó chứa hai tập, sao cho $\mathfrak{R}_1 \subset \mathfrak{R}_0$. Thủ tục này tiếp tục cho đến khi đạt được phép phân cụm cuối cùng \mathfrak{R}_{N-1} , kết quả ta có N tập, mỗi tập chứa một phần tử duy nhất. Trong trường hợp này ta có:

$$\mathfrak{R}_{N-1} \subset \mathfrak{R}_{N-2} \subset \mathfrak{R}_{N-3} \subset \dots \subset \mathfrak{R}_0$$

Phần tiếp theo dành cho các thuật toán tích tụ. Các thuật toán phân rã được thảo luận ngắn gọn trong phần 3.4

3.2. Các thuật toán tích tụ - GAS

Gọi $g(C_i, C_j)$ là hàm đo độ gần gũi giữa mọi cặp C_i, C_j của các cặp cụm có thể trong X , t là mức (cấp) hiện tại của quan hệ phân cấp. Sơ đồ tích tụ tổng quát như sau:

Sơ đồ tích tụ tổng quát

(Generalized Agglomerative Scheme - GAS)

1. Khởi tạo:

- 1.1. Chọn $\mathfrak{R}_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ là phép phân cụm khởi tạo.
- 1.2. $t = 0$.

2. Repeat:

- 2.1. $t = t + 1$

- 2.2. Trong số tất cả các cặp cụm (C_r, C_s) của \mathfrak{R}_{t-1} , tìm cặp (C_i, C_j) sao cho:

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s) : & \text{nếu } g \text{ là hàm không tương tự} \\ \max_{r,s} g(C_r, C_s) : & \text{nếu } g \text{ là hàm tương tự} \end{cases} \quad (3.1)$$

- 2.3. Đặt $C_q = C_i \cup C_j$ ta có phép phân cụm mới là $\mathfrak{R}_t = (\mathfrak{R}_{t-1} - \{C_i, C_j\}) \cup C_q$

Until <tất cả các vector nằm trên một cụm.>

Rõ ràng sơ đồ này tạo ra một sự phân cấp của N phép phân cụm sao cho mỗi phép phân cụm là ẩn trong tất cả các phân phép phân cụm thành công (các phép phân cụm ở mức trên của nó), nghĩa là $\mathfrak{R}_t \subset \mathfrak{R}_s$ với $t < s; s = 1, \dots, N - 1$. Nếu hai vector cùng vào một cụm ở mức t thì chúng sẽ ở lại cụm đó trong tất cả các phép

phân cụm tiếp theo. Đây là một cách khác để xem xét thuộc tính ẩn. Một sự bất lợi của thuộc tính ẩn đó là không có cách nào để khôi phục từ một phép phân cụm “xấu” mà nó xuất hiện ở mức trước của sự phân cấp.

Ở mức t có $N - t$ cụm. Do đó, để xác định một cặp cụm sẽ được trộn với nhau ở mức $t + 1$, phải xét tất cả $\binom{N-t}{2} = \frac{(N-t)(N-t-1)}{2}$ cặp cụm. Tổng số cặp trong suốt tiến trình phân cụm là:

$$\sum_{t=0}^{N-1} \binom{N-t}{2} = \sum_{k=1}^N \binom{k}{2} = \frac{(N-1)N(N+1)}{6}$$

nghĩa là, tổng số thao tác cần thiết bởi một sơ đồ tích tụ là N^3 . Độ phức tạp về thời gian tính toán của sơ đồ GAS là $O(N^3)$ với N là số phần tử cần phân cụm. Tuy nhiên, nếu tổ chức dữ liệu và cài đặt có hiệu quả thì thời gian tính toán sẽ giảm xuống. Hơn nữa, độ phức tạp của thuật toán phụ thuộc vào hàm g .

3.2.1. Một số định nghĩa

Có hai loại thuật toán tích tụ chính: Thuật toán dựa trên khái niệm của lý thuyết ma trận và thuật toán dựa trên khái niệm của lý thuyết đồ thị.

Trước khi thảo luận các thuật toán, chúng ta đưa ra một số định nghĩa:

- *Ma trận mẫu* $D(X)$ là ma trận cấp $N \times \ell$, $\ell \square$ là số chiều của vector, N là số vector cần phân cụm; dòng thứ i là toạ độ vector thứ i của X .
- *Ma trận tương tự (không tương tự)* $P(X)$ là ma trận cấp $N \times N$ mà phần tử nằm ở vị trí (i, j) bằng độ tương tự $s(x_i, x_j)$ (không tương tự $d(x_i, x_j)$) giữa hai vector x_i và x_j . Nó cũng có quan hệ với ma trận gần gũi trong cả hai trường hợp. P là ma trận đối xứng. Hơn nữa nếu P là ma trận tương tự thì các phần tử nằm trên đường chéo bằng giá trị lớn nhất của s , nếu P là ma trận không tương tự thì các phần tử nằm trên đường chéo của nó bằng giá trị nhỏ nhất của d .

Chú ý rằng, với một ma trận mẫu tồn tại nhiều hơn một ma trận gần gũi, phụ thuộc vào sự lựa chọn độ đo gần gũi $\varphi(x_i, x_j)$. Tuy nhiên, cố định $\varphi(x_i, x_j)$, dễ thấy rằng mỗi ma trận mẫu, tồn tại một ma trận gần gũi xác định. Mặt khác, một ma trận gần gũi có thể ứng với nhiều hơn một ma trận mẫu.

Ví dụ 3.2. Cho $X = \{x_i : i = 1, \dots, 5\}$,

với $x_1 = [1, 1]^T$, $x_2 = [2, 1]^T$, $x_3 = [5, 4]^T$, $x_4 = [6, 5]^T$ và $x_5 = [6.5, 6]^T$.

Ma trận mẫu của X là:

$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix}$$

Khi sử dụng độ đo Euclid, ma trận không tương tự của X là:

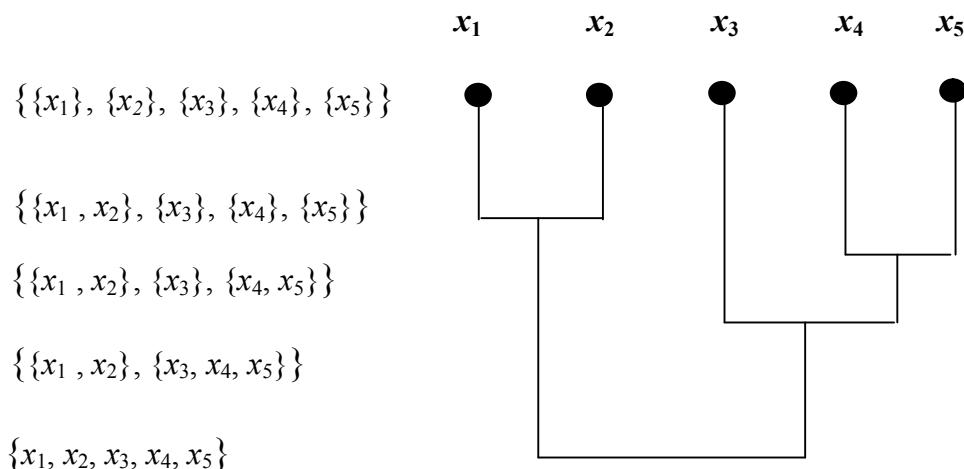
$$P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix}$$

Khi sử dụng độ đo Tanimoto (1.20), ma trận tương tự của X trở thành:

$$P'(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

Chú ý rằng trong $P(X)$ tất cả các phần tử nằm trên đường chéo bằng 0, vì $d_2(x, x) = 0$; còn trong ma trận $P'(X)$ tất cả các phần tử nằm trên đường chéo bằng 1, vì $s_T(x, x) = 1$.

Người ta dùng *sơ đồ ngữ nghĩa*, hay đơn giản là một *sơ đồ* là một cây phân cấp để biểu diễn dãy các phép phân cụm sinh ra bởi thuật toán tích tụ. Để làm rõ điều này, chúng ta lại xét tập dữ liệu đã cho trong ví dụ 3.2 Định nghĩa $g(C_i, C_j) = d_{\min}^{ss}(C_i, C_j)$. Sử dụng độ đo khoảng cách Euclid giữa 2 vector, khi đó dùng sơ đồ tích tụ tổng quát dãy các phép phân cụm tập dữ liệu X sinh ra như hình 3.1.

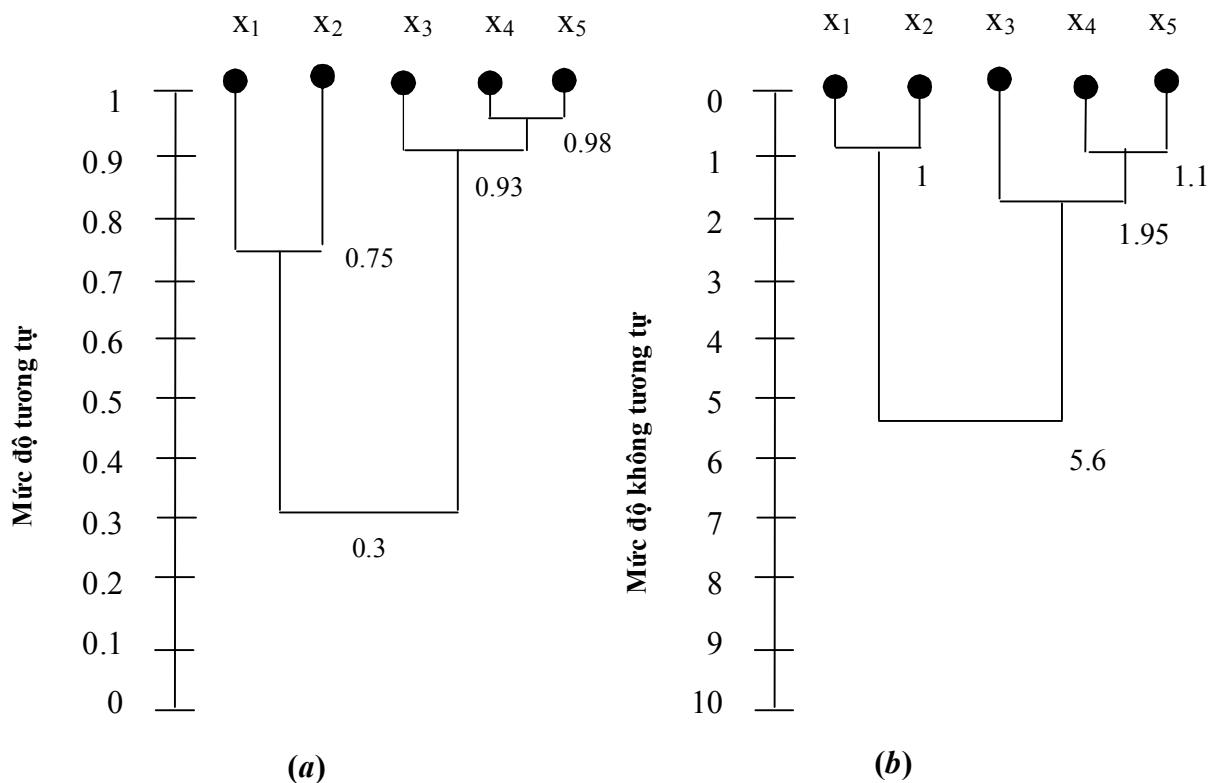


Hình 3-1. Sơ đồ phân cụm phân cấp với tập dữ liệu X trong ví dụ 3.2

Bước đầu tiên, x_1, x_2 tạo thành một cụm. Bước hai, x_4 và x_5 kết hợp với nhau hình thành nên cụm khác. Bước thứ ba, x_3 được đưa vào cụm $\{x_4, x_5\}$ và cuối cùng, ở bước bốn $\{x_1, x_2\}$ và $\{x_3, x_4, x_5\}$ được trộn thành cụm duy nhất X . Bên phải của hình 3.1 chỉ ra sơ đồ tương ứng. Mỗi bước của GAS ứng với một mức sơ đồ. Bằng cách cắt sơ đồ phân cụm ở một mức ngưỡng định trước ta được các cụm sinh ra ở mức đó.

Sơ đồ gần gũi là một sơ đồ xét mức độ gần gũi ở đó hai cụm được trộn với nhau ở lần đầu tiên. Khi sử dụng độ đo không tương tự (tương tự), sơ đồ gần gũi được gọi là một sơ đồ không tương tự (tương tự). Công cụ này có thể được sử dụng một cách tự nhiên hoặc ép buộc hình thành các cụm ở mức bất kỳ.

Hình 3.2 chỉ ra các sơ đồ tương tự và không tương tự của X ở ví dụ 3.2 khi dùng ma trận $P'(X)$ và $P(X)$.



Hình 3-2. Minh họa sơ đồ tương tự và không tương tự.

- Sơ đồ gần gũi (tương tự) với tập dữ liệu X sử dụng ma trận $P'(X)$ ở ví dụ 3.2
- Sơ đồ gần gũi (không tương tự) với tập dữ liệu X sử dụng ma trận $P(X)$ ở ví dụ 3.2

Trước khi thảo luận chi tiết về các thuật toán phân cấp, ta chú ý rằng loại thuật toán này xác định toàn bộ các phép phân cụm phân cấp, thay vì một phép phân cụm duy nhất. Việc xác định toàn bộ sơ đồ có thể rất hữu ích trong một số ứng dụng, chẳng hạn như phân loại sinh vật. Tuy nhiên, trong các ứng dụng khác, chúng ta chỉ quan tâm đến các phép phân cụm riêng phù hợp với dữ liệu. Nếu một ứng dụng sẵn sàng sử dụng các thuật toán phân cụm phân cấp, ta phải quyết định phép phân cụm phân cấp nào sẽ được sinh ra là phù hợp nhất với dữ liệu. Từ đó xác định mức ngưỡng thích hợp để cắt sơ đồ kết quả phân cấp. Giải thích tương tự cũng áp dụng với với các thuật toán phân rã sẽ xét sau này. Các phương pháp xác định mức ngưỡng để cắt sơ đồ được thảo luận trong phần cuối cùng của chương.

Trong các phần tiếp theo, nếu không nói rõ, chúng ta chỉ xét các ma trận không tương tự. Các thảo luận tương tự cũng áp dụng với các ma trận tương tự.

3.2.2. Một số thuật toán tích tụ dựa trên lý thuyết ma trận

Các thuật toán này có thể xem như trường hợp đặc biệt của GAS.

Đầu vào của sơ đồ là ma trận không tương tự cấp $N \times N$, $P_0 = P(X)$ của tập dữ liệu X . Ở mỗi mức t , khi hai cụm được trộn thành một cụm, kích thước của ma trận không tương tự P_t trở thành $(N - t) \times (N - t)$, P_t nhận được từ P_{t-1} bằng cách:

a. Xoá hai dòng và hai cột ứng với các cụm đã trộn

b. Thêm một dòng mới và một cột mới mà nó chứa các khoảng cách giữa cụm mới hình thành và các cụm cũ. Khoảng cách giữa cụm mới hình thành C_q (kết quả của quá trình trộn C_i và C_j) và một cụm cũ C_s là hàm :

$$d(C_q, C_s) = f(d(C_i, C_s), d(C_j, C_s), d(C_i, C_j)) \quad (3.2)$$

Điều này giải thích vì sao các thuật toán này có tên là thuật toán biến đổi ma trận. Tiếp theo, chúng ta đưa ra sơ đồ thuật toán biến đổi ma trận. Đặt t là mức phân cấp hiện tại.

Sơ đồ thuật toán biến đổi ma trận

(Matrix Updating Algorithmic Scheme -MUAS)

1. Khởi tạo

$$1.1. \quad \mathfrak{R}_0 = \{x_i\} : i = 1, \dots, N\}$$

$$1.2. \quad P_0 = P(X)$$

$$1.3. \quad t = 0$$

2. Repeat:

$$2.1. \quad t = t + 1$$

2.2. Tìm C_i, C_j sao cho $d(C_i, C_j) = \min_{r,s=1,\dots,N, r \neq s} d(C_r, C_s)$.

2.3. Trộn C_i, C_j thành cụm C_q và hình thành $\mathfrak{R}_t = (\mathfrak{R}_{t-1} \setminus \{C_i, C_j\}) \cup \{C_q\}$.

2.4. Xác định ma trận gần gũi P_t từ P_{t-1} .

Until <phép phân cụm \mathfrak{R}_{N-1} được hình thành, nghĩa là, tất cả các vector nằm trong cùng một cụm.>

Trong [11] đưa ra hàm khoảng cách giữa hai cụm tuân theo phương trình sau:

$$d(C_q, C_s) = a_i d(C_i, C_s) + a_j d(C_j, C_s) + b d(C_i, C_j) + c |d(C_i, C_s) - d(C_j, C_s)| \quad (3.3)$$

Khi chọn các tham số a_i, a_j, b và c sẽ có các độ đo không tương tự $d(C_i, C_j)$ khác nhau. Tiếp theo, chúng ta đưa ra các thuật toán từ MUAS và công thức (3.3) với các giá trị tham số khác nhau a_i, a_j, b, c .

Các thuật toán đơn giản hơn trong sơ đồ này là:

- **Thuật toán liên kết đơn**

Sử dụng công thức (3.3), nếu đặt $a_i = 1/2, a_j = 1/2, b = 0, c = -1/2$ thì

$$d(C_q, C_s) = \min \{d(C_i, C_s), d(C_j, C_s)\} \quad (3.4)$$

- **Thuật toán liên kết đầy đủ.**

Sử dụng công thức (3.3), nếu đặt $a_i = 1/2, a_j = 1/2, b = 0$ và $c = 1/2$

Ta có thể viết

$$d(C_q, C_s) = \max \{d(C_i, C_s), d(C_j, C_s)\} \quad (3.5)$$

Chú ý rằng khoảng cách giữa các cụm đã trộn C_i và C_j không có trong các công thức trên. Trong trường hợp sử dụng độ đo tương tự thì:

(a) với thuật toán liên kết đơn $d(C_q, C_s) = \max \{d(C_i, C_s), d(C_j, C_s)\}$

(b) với thuật toán liên kết đầy đủ $d(C_q, C_s) = \min \{d(C_i, C_s), d(C_j, C_s)\}$

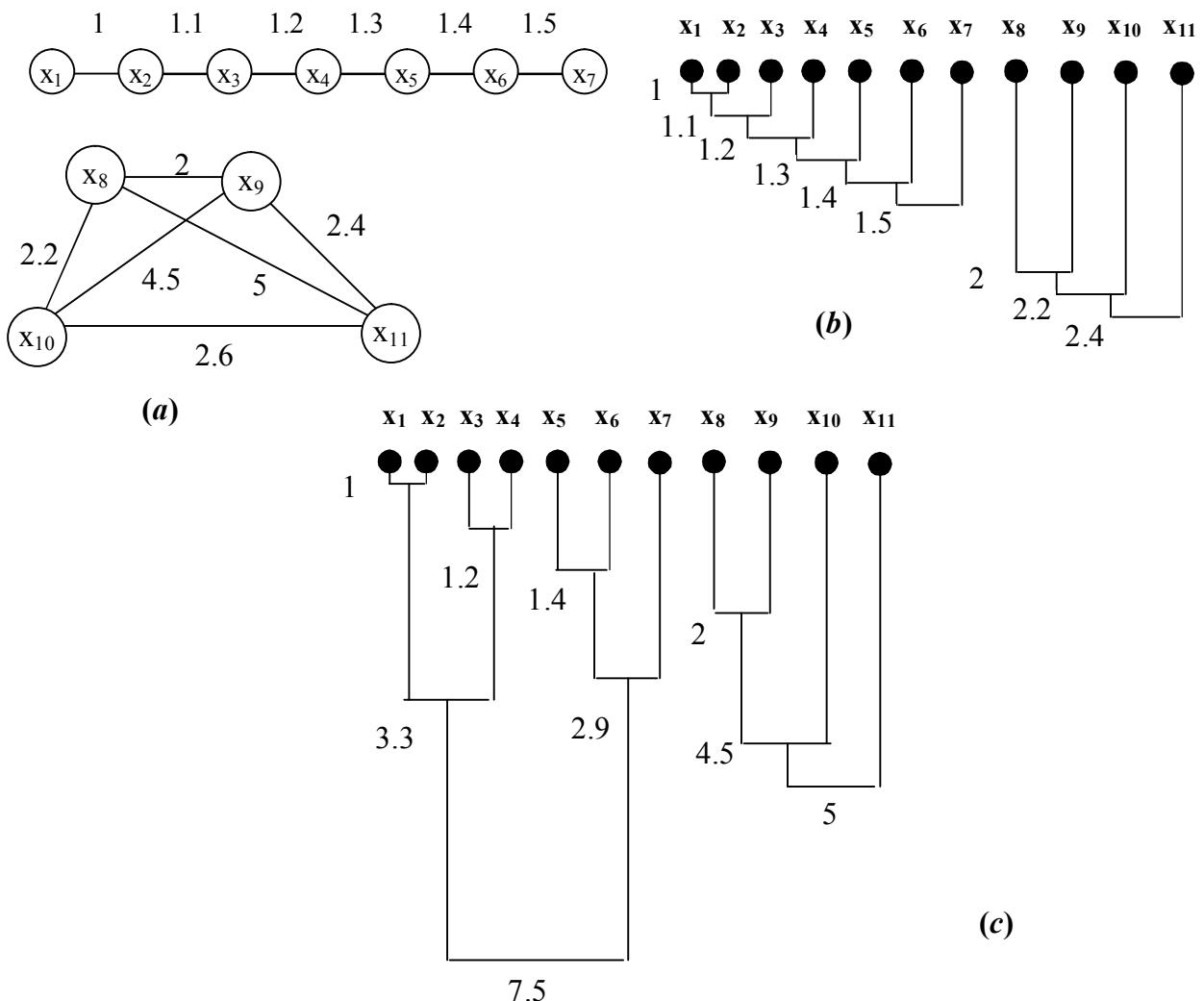
Để hiểu rõ hơn các thuật toán trên, ta xét ví dụ sau:

Xét tập dữ liệu chỉ ra trong hình 3.3a. Bảy điểm đầu tiên tạo thành một cụm dài trong khi đó bốn điểm còn lại tạo thành cụm chặt. Các số nằm trên các cạnh là khoảng cách Euclid giữa các điểm. Các khoảng cách đó cũng là khoảng cách giữa hai cụm điểm khởi tạo.

Hình vẽ 3.3b là sơ đồ sinh ra bởi ứng dụng của thuật toán liên kết đơn với tập dữ liệu này. Như thấy ở trên, đầu tiên thuật toán tìm lại được các cụm dài, và sau đó tìm lại cụm thứ hai ở mức tương tự cao hơn.

Hình 3.3c là sơ đồ sinh ra bởi thuật toán liên kết đầy đủ (mức của phép phân cụm cuối cùng không được chỉ ra). Chú ý rằng thuật toán này tìm lại các cụm chặt

Ví dụ 3.3



Hình 3.3. Tập dữ liệu X (a) và Sơ đồ không tương tự sinh ra bởi thuật toán liên kết đơn (b), thuật toán liên kết đầy đủ (c).

◆ **Nhận xét:**

Các thuật toán liên kết đơn và liên kết đầy đủ là hai thái cực của một họ thuật toán được mô tả bằng công thức (3.3). Thật vậy, các cụm sinh ra bằng thuật toán liên kết đơn được tạo thành ở mức không tương tự thấp trong sơ đồ không tương tự. Mặt khác, các cụm sinh ra bằng thuật toán liên kết đầy đủ hình thành ở mức độ không tương tự cao trong sơ đồ không tương tự. Có điều này vì trong thuật toán liên kết đơn (liên kết đầy đủ) các khoảng cách $d(C_i, C_s)$ và $d(C_j, C_s)$ nhỏ nhất (lớn nhất) thì khoảng cách $d(C_q, C_s)$ cũng nhỏ nhất (lớn nhất). Điều này cho thấy rằng thuật toán liên kết đơn có khuynh hướng hình thành các cụm dài và mảnh; thuật toán liên kết đầy đủ có khuynh hướng hình thành các cụm chặt

Phần còn lại của thuật toán sẽ thảo luận về sự thỏa hiệp giữa hai thái cực.

▪ **Phương pháp trung bình theo cặp trọng số (The weighted pair group method average-WPGMA)**

Chọn $a_i = a_j = 1/2$, $b = 0$ và $c = 0$ thì

$$d(C_q, C_s) = \frac{1}{2}(d(C_i, C_s) + d(C_j, C_s)) \quad (3.6)$$

Trong trường hợp này, khi hai cụm C_i và C_j được trộn thành C_q thì khoảng cách giữa cụm mới hình thành C_q và cụm cũ C_s bằng giá trị trung bình của khoảng cách: giữa C_i và C_s và khoảng cách giữa C_j và C_s

▪ **Phương pháp trung bình theo cặp không trọng số (The Unweighted pair group method average - UPGMA)**

Chọn:

$$a_i = \frac{n_i}{n_i + n_j}, \quad a_j = \frac{n_j}{n_i + n_j}, \quad b = 0, \quad c = 0$$

ở đây n_i , n_j tương ứng là số vector thuộc các tập C_i và C_j . Trong trường hợp này, khoảng cách giữa C_q và C_s được định nghĩa như sau:

$$d(C_q, C_s) = \frac{n_i}{n_i + n_j} d(C_i, C_s) + \frac{n_j}{n_i + n_j} d(C_j, C_s) \quad (3.7)$$

▪ **Phương pháp trọng tâm theo cặp không trọng số (The Unweight pair group method centroid-UPGMC).**

Chọn:

$$a_i = \frac{n_i}{n_i + n_j}, \quad a_j = \frac{n_j}{n_i + n_j}, \quad b = -\frac{n_i n_j}{(n_i + n_j)^2}, \quad c = 0$$

ta có

$$d(C_q, C_s) = \frac{n_i}{n_i + n_j} d(C_i, C_s) + \frac{n_j}{n_i + n_j} d(C_j, C_s) - \frac{n_i n_j}{(n_i + n_j)^2} d(C_i, C_j) \quad (3.8)$$

Lấy đại diện của các cụm là các trọng tâm, nghĩa là:

$$m_q = \frac{1}{n_q} \sum_{x \in C_q} x \quad (3.9)$$

Và độ đo không tương tự là bình phương khoảng cách Euclid giữa các đại diện của cụm:

$$d(C_q, C_s) = \|m_q - m_s\|^2 \quad (3.10)$$

- **Phương pháp trọng tâm theo cặp trọng số (The weighted pair group method centroid - WPGMC)**

Chọn $a_i = a_j = \frac{1}{2}$; $b = -\frac{1}{4}$ và $c = 0$

$$\text{ta có } d(C_q, C_s) = \frac{1}{2} d(C_i, C_s) + \frac{1}{2} d(C_j, C_s) - \frac{1}{4} d(C_i, C_j) \quad (3.11)$$

Chú ý rằng công thức (3.11) là trường hợp riêng của công thức (3.8) nếu trộn các cụm có cùng số vector ($n_i = n_j$).

Một điểm đáng chú ý của thuật toán WPGMC là $d(C_q, C_s) \leq \min\{d(C_i, C_s), d(C_j, C_s)\}$.

- **Thuật toán biến đổi cực tiểu (thuật toán Ward)**

Trong thuật toán này, khoảng cách giữa hai cụm C_i và C_j được định nghĩa lại là:

$$d'(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} d(C_i, C_j) \quad (3.12)$$

Trong đó, $d(C_i, C_j) = \|m_i - m_j\|^2$ là bình phương khoảng cách Euclid của các vector trung bình trong cụm C_i, C_j . Do vậy bước 2.2 của thuật toán MUAS cần tìm cặp cụm C_i, C_j sao cho $d'(C_i, C_j)$ nhỏ nhất.

Hơn nữa, có thể chỉ ra rằng, khoảng cách này thuộc về họ của công thức (3.3). Bây giờ ta sẽ tính khoảng cách giữa cụm C_s với cụm mới hình thành C_q .

Nhân cả hai vế của (3.8) với $\frac{n_s(n_i + n_j)}{n_i + n_j + n_s}$, chú ý $n_i + n_j = n_q$ và biến đổi ta có:

$$d'(C_q, C_s) = \frac{n_i + n_s}{n_i + n_j + n_s} d'(C_i, C_s) + \frac{n_j + n_s}{n_i + n_j + n_s} d'(C_j, C_s) - \frac{n_s}{n_i + n_j + n_s} d'(C_i, C_j) \quad (3.13)$$

Định nghĩa

$$e_r^2 = \sum_{x \in C_r} \|x - m_r\|^2$$

là độ lệch của các vector trong cụm thứ r với vector đại diện của nó, và

$$E_t = \sum_{r=t}^{N-t} e_r^2 \quad (3.14)$$

là tổng độ lệch của các cụm ở mức thứ t (ở đây $N - t$ cụm được xét). Nay giờ chúng ta sẽ chỉ ra rằng thuật toán Ward tạo thành \mathfrak{R}_{t+1} bằng cách trộn hai cụm sao cho tổng độ lệch E_t tăng ít nhất. Giả sử rằng các cụm C_i và C_j được chọn để trộn thành cụm C_q . Đặt E_{t+1}^{ij} là tổng độ lệch sau khi trộn ở mức $t + 1$. Khi đó tất cả các cụm còn lại không bị ảnh hưởng. Hiệu $\Delta E_{t+1}^{ij} = E_{t+1}^{ij} - E_t$ và bằng

$$\Delta E_{t+1}^{ij} = e_q^2 - e_i^2 - e_j^2 \quad (3.15)$$

Xét thấy rằng:

$$\sum_{x \in C_r} \|x - m_r\|^2 = \sum_{x \in C_r} \|x\|^2 - n_r \|m_r\|^2 \quad (3.16)$$

Từ (3.16) và (3.15) ta có

$$\begin{aligned} \Delta E_{t+1}^{ij} &= \left(\sum_{x \in C_q} \|x\|^2 - n_q \|m_q\|^2 \right) - \left(\sum_{x \in C_i} \|x\|^2 - n_i \|m_i\|^2 \right) - \left(\sum_{x \in C_j} \|x\|^2 - n_j \|m_j\|^2 \right) \\ &= \sum_{x \in C_q} \|x\|^2 - \left(\sum_{x \in C_i} \|x\|^2 + \sum_{x \in C_j} \|x\|^2 \right) - n_q \|m_q\|^2 + n_i \|m_i\|^2 + n_j \|m_j\|^2 \\ &= n_i \|m_i\|^2 + n_j \|m_j\|^2 - n_q \|m_q\|^2 \end{aligned}$$

Công thức (3.15) được viết như sau:

$$\Delta E_{t+1}^{ij} = n_i \|m_i\|^2 + n_j \|m_j\|^2 - n_q \|m_q\|^2 \quad (3.17)$$

Trong thực tế thường sử dụng

$$n_i m_i + n_j m_j = n_q m_q \quad (3.18)$$

Nhân cả hai vế của (3.17) với n_q ($n_q = n_i + n_j$) và biến đổi, công thức (3.17) trở thành:

$$\Delta E_{t+1}^{ij} = \frac{n_i n_j}{n_i + n_j} \|m_i - m_j\|^2 = d'(C_i, C_j) \quad (3.19)$$

là khoảng cách tối thiểu bởi thuật toán Ward.

Ví dụ 3.4. Xét ma trận không tương tự sau :

$$P_0 = \begin{bmatrix} 0 & 1 & 2 & 26 & 37 \\ 1 & 0 & 3 & 25 & 36 \\ 2 & 3 & 0 & 16 & 25 \\ 26 & 25 & 16 & 0 & 1.5 \\ 37 & 36 & 25 & 1.5 & 0 \end{bmatrix}$$

Ở đây sử dụng bình phương khoẳng cách Euclid. Từ ma trận trên ta thấy, ba vector đầu tiên x_1, x_2 và x_3 rất gần nhau và xa với các vector khác. Tương tự, x_4 và x_5 nằm rất gần nhau và cách xa với ba vector x_1, x_2 và x_3 . Với bài toán này có tất cả 7 thuật toán được thảo luận trong cùng sơ đồ, chỉ khác là mỗi phép phân cụm được hình thành ở một mức độ không tương tự khác nhau.

Trước hết, ta xét thuật toán liên kết đơn. Vì ma trận P_0 đối xứng nên ta chỉ xét các phần tử phía trên đường chéo chính. Phần tử nhỏ nhất bằng 1 và xuất hiện ở vị trí (1, 2) của P_0 . Do đó, x_1 và x_2 cùng đưa vào một cụm và sinh ra $\mathfrak{R}_1 = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$. Tiếp theo, ta phải tính mức không tương tự giữa cụm mới hình thành và các cụm còn lại theo công thức (3.4). Kết quả ta có ma trận gần gũi P_1 là:

$$P_1 = \begin{bmatrix} 0 & 2 & 25 & 36 \\ 2 & 0 & 16 & 25 \\ 25 & 16 & 0 & 1.5 \\ 36 & 25 & 1.5 & 0 \end{bmatrix}$$

Dòng và cột đầu tiên ứng với cụm $\{x_1, x_2\}$. Trong P_1 , phần tử nhỏ nhất nằm trên đường chéo chính bằng 1.5. Điều này có nghĩa là ở giai đoạn tiếp theo, các cụm $\{x_4\}$ và $\{x_5\}$ sẽ trộn thành cụm $\{x_4, x_5\}$, sinh ra $\mathfrak{R}_2 = \{\{x_1, x_2\}, \{x_3\}, \{x_4, x_5\}\}$

Sử dụng công thức (3.4), ta có

$$P_2 = \begin{bmatrix} 0 & 2 & 25 \\ 2 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix}$$

Ở đây dòng (cột) đầu tiên tương ứng với $\{x_1, x_2\}$ và các dòng (cột) thứ hai và thứ ba ứng với $\{x_3\}$ và $\{x_4, x_5\}$. Như đã xét, giai đoạn tiếp theo $\{x_1, x_2\}$ và $\{x_3\}$ sẽ được trộn với nhau và $\mathfrak{R}_3 = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$. Ma trận gần gũi với P_3 trở thành

$$P_3 = \begin{bmatrix} 0 & [16] \\ 16 & 0 \end{bmatrix}$$

Ở đây dòng (cột) thứ nhất và thứ hai ứng với các cụm $\{x_1, x_2, x_3\}$ và $\{x_4, x_5\}$. Cuối cùng $R_4 = \{\{x_1, x_2, x_3, x_4, x_5\}\}$ sẽ được hình thành với mức tương tự = 16 và $P_4 = [0]$

Thực hiện tương tự, chúng ta có thể áp dụng P_0 với 6 thuật toán còn lại. Chú ý rằng, trong thuật toán Ward, ma trận không tương tự khởi tạo là $\frac{1}{2} P_0$ (theo định nghĩa trong (3.12)). Tuy nhiên, phải cẩn thận khi áp dụng phương pháp WPGMA, WPGMC. Trong các trường hợp này, khi thực hiện một phép trộn, các tham số a_i, a_j, b và c cần được điều chỉnh cho phù hợp với từng thuật toán. Các mức độ gần gũi mà mỗi phép phân cụm tạo ra ứng với mỗi thuật toán được trình bày trong bảng 3.1.

Bảng 3-1. Các kết quả của 7 thuật toán đã thảo luận khi áp dụng ma trận gần gũi của ví dụ 3.4

	SL	CL	WPGMA	UPGMA	WPGMC	UPGMC	Ward's Algorithm
R_0	0	0	0	0	0	0	0
R_1	1	1	1	1	1	1	0.5
R_2	1.5	1.5	1.5	1.5	1.5	1.5	0.75
R_3	2	3	2.5	2.5	2.25	2.25	1.5
R_4	16	37	25.75	27.5	24.69	26.46	29.74

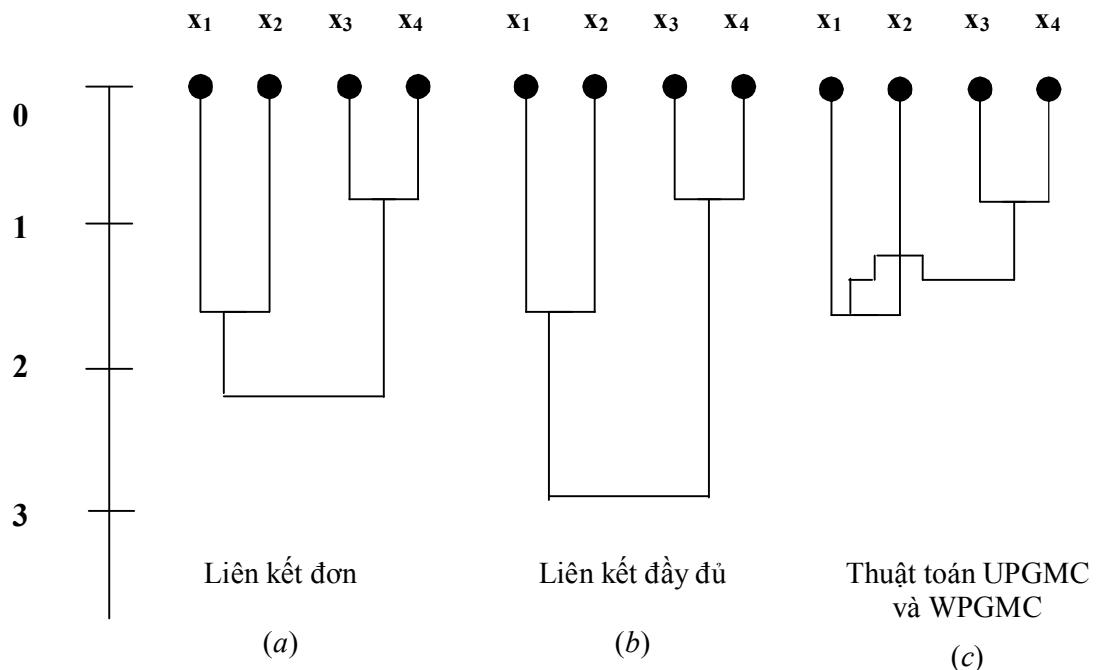
Chú ý rằng ví dụ vừa xét là một ví dụ hay với hai cụm chặt hoàn toàn xác định nằm cách xa nhau và tất cả các thuật toán làm việc tốt. Tuy nhiên, các đặc tính đặc biệt của mỗi thuật toán sẽ xuất hiện khi gặp phải những tình huống đòi hỏi khắt khe. Vì vậy ví dụ 3.3 đã chỉ ra các trường hợp khác nhau của thuật toán liên kết đơn và thuật toán liên kết đầy đủ. Đặc tính của các thuật toán khác, chẳng hạn như WPGMC và UPGMC sẽ được thảo luận trong phần tiếp theo.

3.2.3. Monotonicity và Crossover

Xét ma trận không tương tự sau:

$$P = \begin{bmatrix} 0 & 1.8 & 2.4 & 2.3 \\ 1.8 & 0 & 2.5 & 2.7 \\ 2.4 & 2.5 & 0 & 1.2 \\ 2.3 & 2.7 & 1.2 & 0 \end{bmatrix}$$

Lần lượt áp dụng các thuật toán liên kết đơn và thuật toán liên kết đầy đủ với ma trận P đã cho, ta có sơ đồ không tương tự mô tả trong hình 3.4a và 3.4b. Áp dụng các thuật toán UPGMC và WPGMC với P , ta được cùng sơ đồ hình 3.4c. Trong sơ đồ này ta thấy cụm $\{x_3, x_4\}$ hình thành ở mức không tương tự bằng 1.2, cụm $\{x_1, x_2\}$ hình thành ở mức không tương tự bằng 1.8, cụm $\{x_1, x_2, x_3, x_4\}$ hình thành ở mức không tương tự bằng 1.72. Ta thấy điều thú vị là cụm $\{x_1, x_2, x_3, x_4\}$ hình thành ở mức không tương tự thấp hơn mức không tương tự khi hình thành cụm $\{x_1, x_2\}$. Hiện tượng này gọi là crossover. *Crossover xuất hiện khi một cụm được hình thành ở một mức độ không tương tự thấp hơn một cụm nào đó đã hình thành trong các giai đoạn trước.*



Hình 3-4 . Sơ đồ không tương tự sinh ra bởi thuật toán Liên kết đơn, Liên kết đầy đủ, UPGMC và WPGMC với hiện tượng crossover .

Ngược lại với *crossover* là *monotonicity*. Một cách hình thức, điều kiện *monotonicity* có thể phát biểu như sau :

“Nếu các cụm C_i và C_j được chọn để trộn thành C_q , ở mức t của quan hệ phân cấp, thì phải thoả mãn điều kiện sau: $d(C_q, C_k) \geq d(C_i, C_j)$ với mọi C_k , $k \neq i, j, q$ ”, tức là mỗi cụm được hình thành ở mức độ không tương tự cao hơn một cụm nào đó đã hình thành trong các giai đoạn trước. *Monotonicity* chỉ liên quan đến các thuật toán phân cụm mà không liên quan đến ma trận gần gũi (khởi tạo). Điều này sẽ được chứng tỏ khi ta xét định đè sau:

Dinh de 1: Khi lựa chọn các tham số a_i, a_j, b và c trong công thức xác định khoảng cách từ cụm mới hình thành C_q tới các cụm khác

$$d(C_q, C_s) = a_i d(C_i, C_s) + a_j d(C_j, C_s) + b d(C_i, C_j) + c |d(C_i, C_s) - d(C_j, C_s)|$$

Nếu a_i và a_j không âm, $a_i + a_j + b \geq 1$ và

$$\text{hoặc là (a)} \quad c \geq 0$$

$$\text{hoặc là (b)} \quad \max\{-a_i, -a_j\} \leq c \leq 0$$

thì phương pháp phân cụm tương ứng thoả điều kiện monotonicity

Chứng minh:

(a) Theo giả thiết: $b \geq 1 - a_i - a_j$ thay vào công thức (3.3) và biến đổi ta có:

$$\begin{aligned} d(C_q, C_s) &\geq a_i d(C_i, C_s) + a_j d(C_j, C_s) + (1 - a_i - a_j) d(C_i, C_j) + c |d(C_i, C_s) - d(C_j, C_s)| \Rightarrow \\ d(C_q, C_s) &\geq d(C_i, C_j) + a_i [d(C_i, C_s) - d(C_i, C_j)] + a_j [d(C_j, C_s) - d(C_i, C_j)] + c |d(C_i, C_s) - d(C_j, C_s)| \end{aligned}$$

Theo bước 2.2 của MUAS trong phần 3.2.2:

$$d(C_i, C_j) = \min_{r,u} d(C_r, C_u) \Rightarrow d(C_i, C_j) \leq d(C_i, C_s) \text{ và } d(C_i, C_j) \leq d(C_j, C_s)$$

Nên số hạng thứ hai và số hạng thứ ba của bất đẳng thức cuối cùng không âm, $c \geq 0$ nên số hạng thứ tư cũng không âm. Do đó, ta có :

$$d(C_q, C_s) \geq d(C_i, C_j)$$

Vì vậy điều kiện monotonicity được thoả mãn.

(b) Từ giả thiết: $b \geq 1 - a_i - a_j$ nên theo phần (a) ta có:

$$d(C_q, C_s) \geq d(C_i, C_j) + a_i [d(C_i, C_s) - d(C_i, C_j)] + a_j [d(C_j, C_s) - d(C_i, C_j)] + c |d(C_i, C_s) - d(C_j, C_s)|$$

Để bỏ dấu giá trị tuyệt đối, xét trường hợp $d(C_i, C_s) \geq d(C_j, C_s)$ (trường hợp ngược lại xét tương tự).

$$\Rightarrow d(C_q, C_s) \geq d(C_i, C_j) + a_i [d(C_i, C_s) - d(C_i, C_j)] + a_j [d(C_j, C_s) - d(C_i, C_j)] + c [d(C_i, C_s) - d(C_j, C_s)]$$

Bằng cách cộng và trừ vé phải của bất đẳng thức với số hạng $c \cdot d(C_i, C_j)$ và sau đó biến đổi ta có:

$$d(C_q, C_s) \geq (a_j - c) [d(C_j, C_s) - d(C_i, C_j)] + d(C_i, C_j) + (a_i + c) [d(C_i, C_s) - d(C_i, C_j)]$$

$$\text{Từ giả thiết } \max\{-a_i, -a_j\} \leq c \leq 0 \Rightarrow a_j \geq 0; -c \geq 0 \Rightarrow a_j - c \geq 0;$$

$$-a_i \leq c \leq 0 \Rightarrow c + a_i \geq 0$$

và theo bước (2.2) của MUAS ta có:

$$d(C_i, C_j) = \min_{r,u} d(C_r, C_u) \Rightarrow d(C_i, C_j) \leq d(C_i, C_s) \text{ và } d(C_i, C_j) \leq d(C_j, C_s)$$

$$\Rightarrow d(C_q, C_s) \geq d(C_i, C_j).$$

Chú ý rằng định đề 1 là điều kiện đủ chứ không là điều kiện cần, nghĩa là các thuật toán đó không thoả các điều kiện của định đề này nhưng vẫn có thể thoả điều kiện *monotonicity*. Các thuật toán liên kết đơn, liên kết đầy đủ, UPGMA, WPGMA và Ward thoả các điều kiện của định đề 1. Vì vậy, các thuật toán đó thoả điều kiện *monotonicity*. Hai thuật toán UPGMC và WPGMC không thoả mãn điều kiện *monotonicity*. Hơn nữa, chúng ta có thể xây dựng các ví dụ để chứng tỏ rằng hai thuật toán đó vi phạm thuộc tính *monotonicity*, như hình 3.4c. Tuy nhiên không thể nói rằng một thuật toán không thoả điều kiện *monotonicity* thì nó luôn dẫn tới các sơ đồ *crossover*.

3.2.4. Một số thuật toán tích tụ dựa trên lý thuyết đồ thị

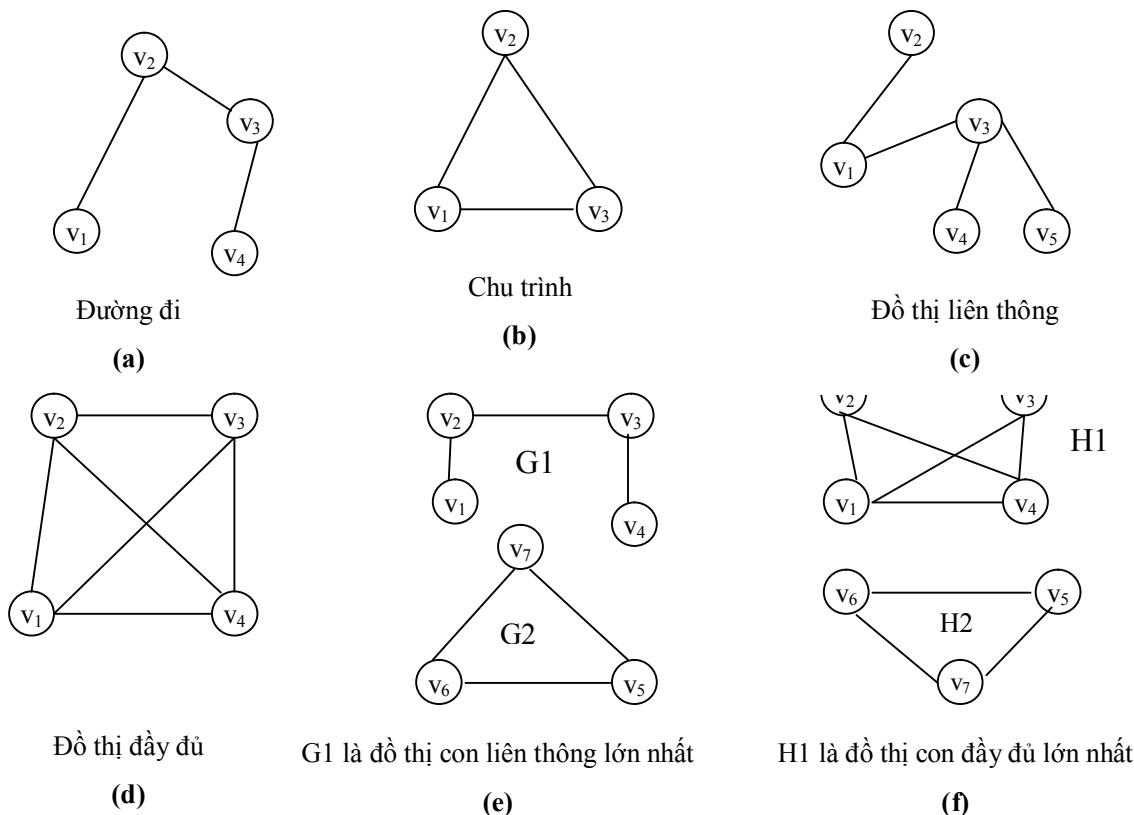
Trước khi mô tả các thuật toán thuộc họ này, chúng ta tóm tắt một số định nghĩa và khái niệm cơ bản của lý thuyết đồ thị sẽ dùng trong các thuật toán.

3.2.4.1. Một số định nghĩa và khái niệm cơ bản của lý thuyết đồ thị

- Cho tập V khác rỗng, E là tập hợp các cặp phần tử của V được sắp có thứ tự hoặc không có thứ tự. Cặp có (V, E) được gọi là một **đồ thị**. Ký hiệu đồ thị là $G = (V, E)$ hoặc đôi khi nếu không gây sự nhầm lẫn kí hiệu tắt là G . Các phần tử thuộc tập V gọi là **đỉnh** của đồ thị G . Với hai đỉnh $v_1, v_2 \in V$, nếu $e = (v_1, v_2) \in E$ là cặp sắp thứ tự thì e được gọi là một **cung** của đồ thị, hoặc nếu e là cặp không sắp thứ tự thì e được gọi là một **cạnh** của đồ thị. **Khuyên** là cạnh (hoặc cung) có hai đầu trùng nhau.
- **Đồ thị vô hướng** $G = (V, E)$ là đồ thị mà tập E chỉ gồm các cạnh. Nếu E chỉ gồm các cung thì G là **đồ thị có hướng**. Nếu E gồm cả cạnh và cung thì G là **đồ thị hỗn hợp**. Nếu trên mỗi cạnh hoặc cung thuộc E được gán một số thực thì G là **đồ thị có trọng số**, ngược lại ta có **đồ thị không trọng số**.
- **Đa đồ thị**: Đồ thị $G = (V, E)$ vô hướng (hoặc có hướng) là đa đồ thị khi và chỉ khi nó là đồ thị không khuyên và có ít nhất một cặp đỉnh được nối với nhau bằng ít nhất hai cạnh (hoặc hai cung nối theo thứ tự của cặp đỉnh).
- **Đơn đồ thị**: Đồ thị $G = (V, E)$ vô hướng (hoặc có hướng) là đơn đồ thị khi và chỉ khi nó là đồ thị không khuyên và mỗi cặp đỉnh được nối với nhau không quá một cạnh (hoặc cung).
- **Đồ thị con**: Đồ thị con của đồ thị $G = (V, E)$ là đồ thị có dạng $G_1 = (V_1, E_1)$ trong đó tập đỉnh V_1 là tập con của tập V , tập cạnh (cung) E_1 là tập con của E gồm các cạnh (cung) có hai đầu là hai đỉnh thuộc V_1 :

$$E_1 = \{e = (v_i, v_j), v_i \in V_1, v_j \in V_1 \mid e \in E\} \text{ với } V_1 \subset V.$$

- **Đồ thị liên thông:** Đồ thị $G = (V, E)$ là liên thông nếu luôn tìm được đường đi giữa hai đỉnh bất kỳ của nó. Ví dụ, trong hình 3.5c đồ thị con với các đỉnh v_1, v_2, v_4 và v_5 là liên thông.
- **Thành phần liên thông (vùng liên thông):** Cho đồ thị $G = (V, E)$ và đồ thị con của G là đồ thị $G_1 = (V_1, E_1)$. Đồ thị G_1 được gọi là một thành phần liên thông (hoặc vùng liên thông) nếu G_1 liên thông và không tồn tại một đường đi nào từ một đỉnh thuộc G_1 tới một đỉnh không thuộc G_1 .
- **Đồ thị đầy đủ:** Đồ thị $G = (V, E)$ là đầy đủ nếu mỗi đỉnh $v_i \in V$ là liên thông với mọi đỉnh trong $V - \{v_i\}$ (hình 3.5d).
- **Đồ thị con liên thông lớn nhất** của G là một đồ thị con liên thông G_1 có nhiều đỉnh nhất (hình 3.5e). **Đồ thị con đầy đủ lớn nhất** là một đồ thị con đầy đủ G_1 của G có nhiều đỉnh nhất (hình 3.5f).
- **Đường đi** có độ dài k (k nguyên dương) giữa đỉnh u và v trong đồ thị vô hướng G là dãy các đỉnh $u = v_0, v_1, \dots, v_k = v$ mà các cạnh $(v_i, v_{i+1}) \in E; i = 1, 2, \dots, k-1$. (hình 3.5a). Đỉnh u gọi là đỉnh đầu, đỉnh v gọi là đỉnh cuối của đường đi. Đường đi có đỉnh đầu và đỉnh cuối trùng nhau gọi là **chu trình** (hình 3.5b).
- **Bậc của đỉnh** trong đồ thị vô hướng: Ký hiệu tập các cạnh kề với đỉnh v là E_v , số phần tử của tập này gọi là bậc của đỉnh v . Ký hiệu là $m(v) = |E_v|$.



Hình 3-5. Minh họa đường đi và các loại đồ thị.

Trong các thuật toán phân cụm sau đây, chúng ta chỉ xét đơn đồ thị vô hướng, mỗi đỉnh ứng với một vector đặc trưng (hoặc mẫu đại diện bằng các vector đặc trưng).

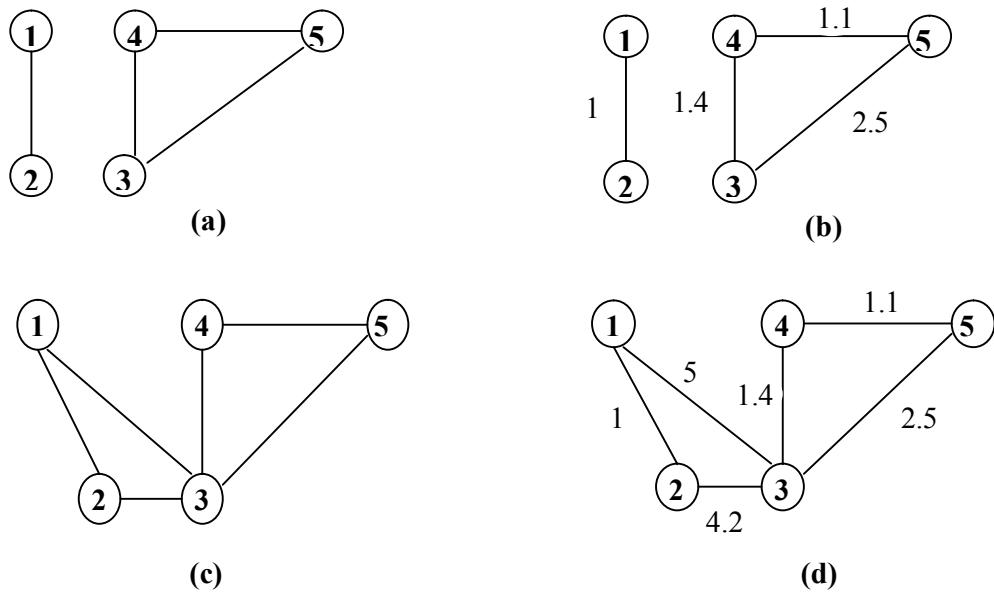
Khái niệm liên quan chặt chẽ với các thuật toán dựa trên lý thuyết đồ thị là đồ thị ngưỡng. **Đồ thị ngưỡng** là một đơn đồ thị vô hướng, không trọng số, N đỉnh, mỗi đỉnh ứng với một vector của tập dữ liệu X . Lấy a là một mức không tương tự. Đồ thị ngưỡng $G(a)$ có N đỉnh, cạnh $(v_i, v_j) \in G(a)$ nếu mức độ không tương tự giữa hai vector x_i và x_j nhỏ hơn hoặc bằng a , với $i, j = 1, \dots, N$. Tức là:

$$(v_i, v_j) \in G(a), \text{ nếu } d(x_i, x_j) \leq a; \quad i, j = 1, \dots, N. \quad (3.20)$$

Nếu sử dụng độ đo tương tự, định nghĩa này được sửa lại là:

$$(v_i, v_j) \in G(a), \text{ nếu } s(x_i, x_j) \geq a; \quad i, j = 1, \dots, N.$$

Đồ thị gần gũi $G_p(a)$ là một đồ thị ngưỡng $G(a)$ và tất cả các cạnh (v_i, v_j) được gán trọng số là độ đo gần gũi giữa vector x_i và x_j . Nếu sử dụng độ đo không tương tự (tương tự) để đo sự gần gũi giữa hai vector thì đồ thị gần gũi được gọi là đồ thị không tương tự (tương tự). Hình 3.6 chỉ ra các đồ thị ngưỡng và đồ thị gần gũi $G(3)$, $G_p(3)$, $G(5)$ và $G_p(5)$ xây dựng từ ma trận không tương tự $P(X)$ trong ví dụ 3.2.



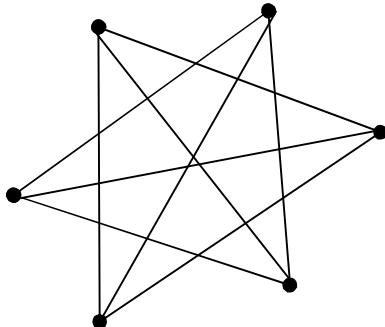
Hình 3-6. Các đồ thị ngưỡng và đồ thị gần gũi xây dựng từ ma trận không tương tự $P(X)$ của ví dụ 3.2

- (a). Đồ thị ngưỡng $G(3)$,
- (b). Đồ thị gần gũi (không tương tự) $G_p(3)$,
- (c). Đồ thị ngưỡng $G(5)$,
- (d). Đồ thị gần gũi (không tương tự) $G_p(5)$.

3.2.4.2. Các thuật toán

Phần này thảo luận về các thuật toán tích tụ dựa trên khái niệm lý thuyết đồ thị. Xét đồ thị G , N đỉnh, mỗi đỉnh ứng với một vector của X . Các cụm hình thành bằng cách nối các đỉnh với nhau để được các đồ thị con liên thông. Thông thường, ta thêm vào đồ thị thuộc tính $h(k)$, các đồ thị con phải thỏa mãn thuộc tính này thì mới hình thành các cụm hợp lệ. Vì vậy, hàm g trong GAS phải được thay thế bằng $g_{h(k)}$, với $h(k)$ là thuộc tính của đồ thị. Một số thuộc tính tiêu biểu là:

- **Khả năng liên kết đỉnh:** Khả năng liên kết đỉnh của một đồ thị con liên thông là số nguyên k lớn nhất sao cho từ đỉnh v_i đến đỉnh v_j bất kỳ có ít nhất k đường đi không có các đỉnh chung.
- **Khả năng liên kết cạnh:** Khả năng liên kết cạnh của một đồ thị con liên thông là số nguyên k lớn nhất sao từ đỉnh v_i đến đỉnh v_j bất kỳ có ít nhất k đường đi không có các cạnh chung.
- **Bậc của đỉnh:** Bậc của đỉnh trong đồ thị con liên thông là số nguyên k lớn nhất sao cho mỗi đỉnh có ít nhất k cạnh liền kề. (xem hình 3.7)



Khả năng liên kết đỉnh: 2

Khả năng liên kết cạnh: 2

Bậc của đỉnh: 3

Hình 3-7. Đồ thị với khả năng liên kết cạnh và đỉnh bằng 2 và bậc của đỉnh là 3

Sơ đồ tích tụ tổng quát sử dụng lý thuyết đồ thị còn gọi là sơ đồ thuật toán dựa trên lý thuyết đồ thị (Graph Theory – based Algorithmic Scheme - GTAS). Sơ đồ này có cùng số bước lặp với sơ đồ tích tụ tổng quát (GAS) trừ bước 2.2 được thay bằng:

$$g_{h(k)}(C_i, C_j) = \begin{cases} \min_{r,s} g_{h(k)}(C_r, C_s): & \text{với các hàm không tương tự} \\ \max_{r,s} g_{h(k)}(C_r, C_s): & \text{với các hàm tương tự} \end{cases} \quad (3.21)$$

Sơ đồ thuật toán dựa trên lý thuyết đồ thị

(Graph Theory – based Algorithmic Scheme – GTAS)

1. Khởi tạo:

1.1. Chọn $\mathfrak{R}_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ là phép phân cụm khởi tạo.

1.2. $t = 0$.

2. Repeat:

2.1. $t = t + 1$

2.2. Trong số tất cả các cặp cụm (C_r, C_s) của \mathfrak{R}_{t-1} , tìm cặp (C_i, C_j) sao cho:

$$g_{h(k)}(C_i, C_j) = \begin{cases} \min_{r,s} g_{h(k)}(C_r, C_s) : \text{với các hàm không tương tự} \\ \max_{r,s} g_{h(k)}(C_r, C_s) : \text{với các hàm tương tự} \end{cases}$$

2.3. Đặt $C_q = C_i \cup C_j$ ta có phép phân cụm mới là $\mathfrak{R}_t = (\mathfrak{R}_{t-1} \setminus \{C_i, C_j\}) \cup C_q$

Until <tất cả các vector nằm trên một cụm.>

Hàm gần gũi $g_{h(k)}(C_r, C_s)$ giữa hai cụm được định nghĩa dựa trên độ đo gần gũi giữa các vector (tức là các đỉnh của đồ thị) và các ràng buộc của thuộc tính $h(k)$ trên các đồ thị con (được tạo ra).

Một cách chi tiết hơn, $g_{h(k)}$ được định nghĩa là:

$$g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a : \text{Đồ thị con } G(a) \text{ định nghĩa bởi } C_r \cup C_s$$

thoả một trong ba điều kiện sau:

- a) *liên thông*
 - b) *liên thông và (b1) có thuộc tính $h(k)$*
 - c) *đầy đủ }*
- (3.22)

Tức là, các cụm (các đồ thị con liên thông) được trộn với nhau dựa trên độ đo gần gũi giữa các đỉnh của chúng và trong quá trình trộn luôn tạo ra một đồ thị con liên thông hoặc liên thông và có thuộc tính $h(k)$ hoặc là đầy đủ.

Sau đây chúng ta xét một số trường hợp riêng của thuật toán GTAS và ví dụ:

a. Thuật toán liên kết đơn

Ở đây sự liên thông là điều kiện duy nhất; không cần thuộc tính $h(k)$ cũng như tính đầy đủ của đồ thị. Do đó (b1) và (c) trong (3.22) được bỏ qua và công thức (3.22) đơn giản hóa thành

$$g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a : \text{Đồ thị con } G(a) \text{ định nghĩa bởi } C_r \cup C_s \text{ là liên thông}\} \quad (3.23)$$

Chúng ta mô phỏng thuật toán qua ví dụ sau:

Ví dụ 3.5. Xét ma trận không tương tự sau:

$$P = \begin{bmatrix} 0 & 1.2 & 3 & 3.7 & 4.2 \\ 1.2 & 0 & 2.5 & 3.2 & 3.9 \\ 3 & 2.5 & 0 & 1.8 & 2.0 \\ 3.7 & 3.2 & 1.8 & 0 & 1.5 \\ 4.2 & 3.9 & 2.0 & 1.5 & 0 \end{bmatrix}$$

Phép phân cụm đầu tiên \mathfrak{R}_0 là phép phân cụm trong đó mỗi vector của X hình thành nên một cụm (xem hình 3.8). Để xác định phép phân cụm tiếp theo, \mathfrak{R}_1 , bằng thuật toán liên kết đơn cần tính $g_{h(k)}(C_1, C_2)$ với mọi cặp cụm hiện có. Với $\{x_1\}$ và $\{x_2\}$, giá trị của $g_{h(k)}=1.2$. Với $\{x_1\}$ và $\{x_3\}$, $g_{h(k)}(\{x_1\}, \{x_3\}) = 3$. Các giá trị $g_{h(k)}$ còn lại được tính tương tự. Sử dụng công thức (3.21), ta thấy rằng $g_{h(k)}(\{x_1\}, \{x_2\}) = 1.2$ là giá trị nhỏ nhất và do đó $\{x_1\}$ và $\{x_2\}$ được trộn với nhau để sinh ra

$$\mathfrak{R}_1 = \{\{x_1, x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$

và $\{x_1\} \cup \{x_2\}$ là đồ thị liên thông đầu tiên trong $G(1.2)$.

Thực hiện tương tự, ta thấy rằng giá trị $g_{h(k)}(\{x_4\}, \{x_5\}) = 1.5$ là nhỏ nhất trong tất cả các cặp cụm. Do vậy:

$$\mathfrak{R}_2 = \{\{x_1, x_2\}, \{x_3\}, \{x_4, x_5\}\}.$$

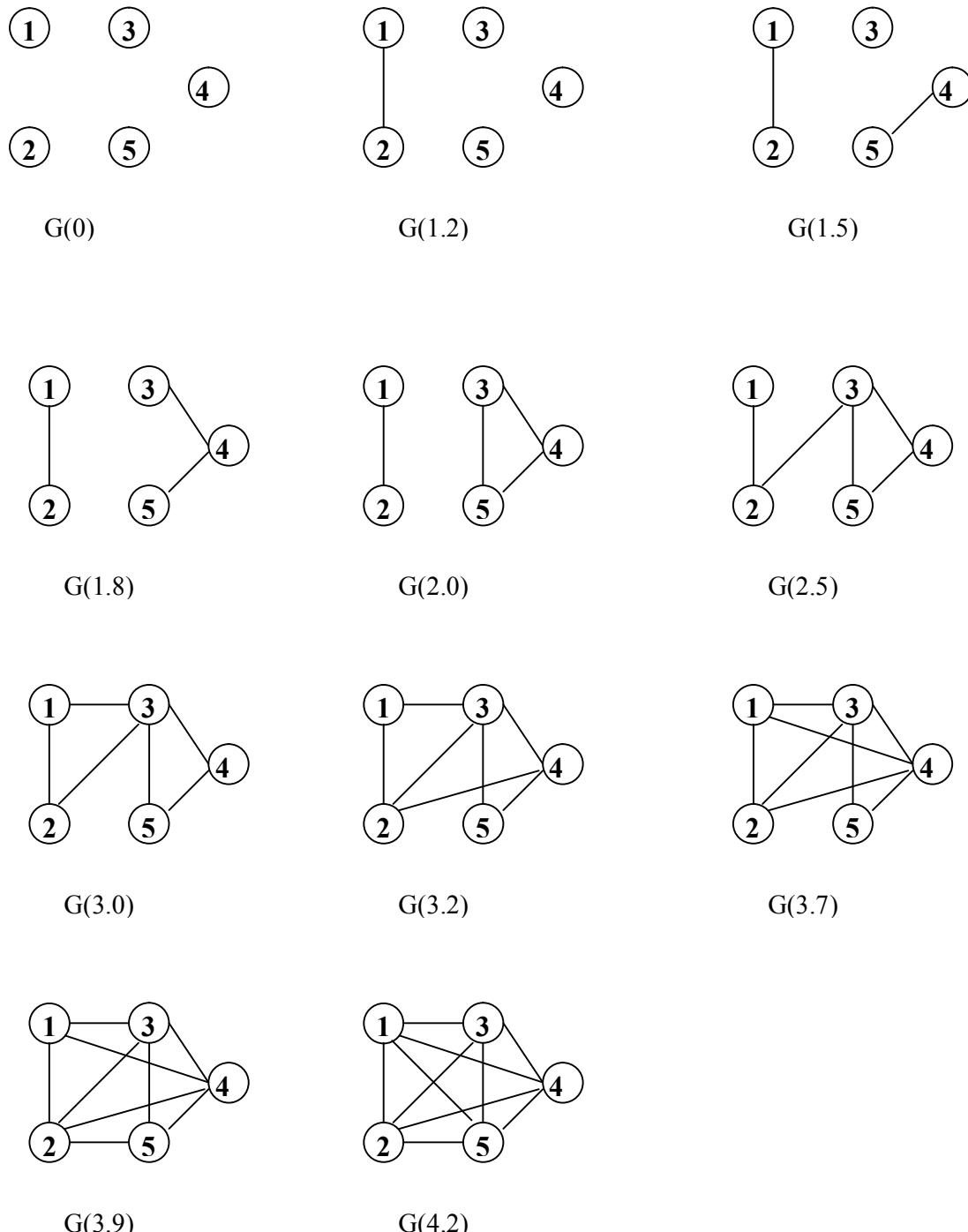
Để có phép phân cụm \mathfrak{R}_3 , trước hết chúng ta xét các cụm $\{x_3\}$ và $\{x_4, x_5\}$. Trong trường hợp này $g_{h(k)}(\{x_3\}, \{x_4, x_5\}) = \min \{d(x_3, x_4), d(x_3, x_5)\} = \min \{1.8, 2.0\} = 1.8$, do đó $\{x_3\} \cup \{x_4, x_5\}$ trở thành liên thông trong $G(1.8)$ lần đầu tiên. Tương tự, ta tìm được $g_{h(k)}(\{x_1, x_2\}, \{x_3\}) = 2.5$ và $g_{h(k)}(\{x_1, x_2\}, \{x_4, x_5\}) = 3.2$. Do đó:

$$\mathfrak{R}_3 = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}\}$$

Cuối cùng $g_{h(k)}(\{x_1, x_2\}, \{x_3, x_4, x_5\}) = 2.5$ và

$$\mathfrak{R}_4 = \{x_1, x_2, x_3, x_4, x_5\}$$

được hình thành ở mức này. Từ trên ta thấy rằng với $G(2.0)$, không phép phân cụm nào được tạo ra



Hình 3-8 . Các đồ thị ngưỡng của ma trận không tương tự P trong ví dụ 3.5

◆ **Nhận xét:**

Trong thuật toán liên kết đơn không cần thuộc tính $h(k)$, và công thức (3.23) về cơ bản giống với $g_{h(k)}(C_r, C_s) = \min_{x \in C_r, y \in C_s} d(x, y)$. Do đó thuật toán này tương đương với thuật toán liên kết đơn dựa trên lý thuyết ma trận và cả hai thuật toán cho cùng kết quả.

b. Thuật toán liên kết đầy đủ

Điều kiện duy nhất ở đây là tính đầy đủ nghĩa là bao qua thuộc tính $h(k)$ của đồ thị. Các đồ thị con chỉ hình thành nên các cụm hợp lệ nếu đồ thị đó là đầy đủ. Chúng ta sẽ mô phỏng thuật toán theo ma trận không tương tự ở ví dụ 3.5

Các phép phân cụm \mathfrak{R}_0 , \mathfrak{R}_1 và \mathfrak{R}_2 giống như được sinh ra bằng thuật toán liên kết đơn, và hình thành nên các cụm tương ứng với các đồ thị ngưỡng $G(0)$, $G(1.2)$ và $G(1.5)$. Tiếp tục với phép phân cụm \mathfrak{R}_3 ; $g(\{x_3\}, \{x_4, x_5\}) = 2$ bởi vì trong $G(2.0)$, $\{x_3\} \cup \{x_4, x_5\}$ trở thành đầy đủ trong lần đầu tiên. Tương tự $g_{h(k)}(\{x_1, x_2\}, \{x_3\}) = 3$ và $g_{h(k)}(\{x_1, x_2\}, \{x_4, x_5\}) = 4.2$. Do đó phép phân cụm \mathfrak{R}_3 giống như một phép phân cụm có được từ thuật toán liên kết đơn. Sự khác nhau duy nhất là nó hình thành ở đồ thị $G(2.0)$ thay vì $G(1.8)$. Cuối cùng, phép phân cụm \mathfrak{R}_4 được định nghĩa ở đồ thị $G(4.2)$.

• Nhận xét:

- Công thức (3.22) của thuật toán liên kết đầy đủ tương đương với:

$$g_{h(k)}(C_r, C_s) = \max_{x \in C_r, y \in C_s} d(x, y) \quad (3.25)$$

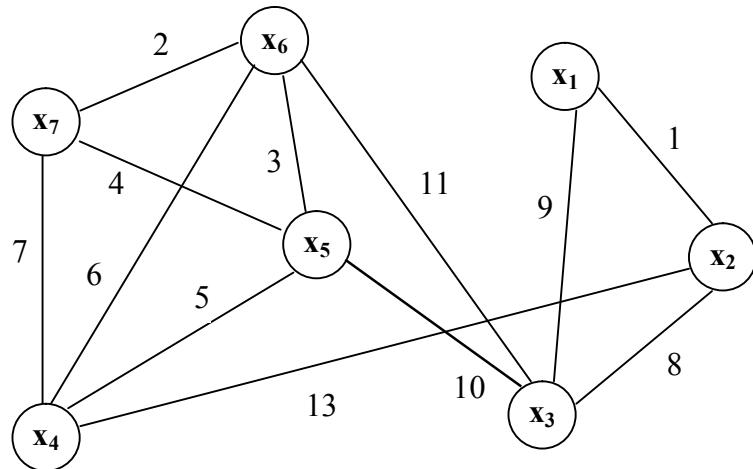
và do đó thuật toán này giống với thuật toán liên kết đầy đủ dựa trên lý thuyết ma trận.

- Các thuật toán liên kết đơn và liên kết đầy đủ có thể xem như hai thái cực của sơ đồ GTAS, bởi vì tiêu chuẩn hình thành nên cụm hợp lệ là yếu nhất với thuật toán liên kết đơn (chỉ cần đồ thị con tạo thành các cụm là liên thông) và mạnh nhất với thuật toán liên kết đầy đủ. Các thuật toán mà các đồ thị con tạo thành các cụm có thuộc tính $h(k)$ nằm giữa hai thái cực này. Chỉ cần thay đổi thuộc tính $h(k)$ trong công thức (3.22) (kéo theo là thay đổi $g_{h(k)}$) sẽ tạo ra một lớp các thuật toán khác nhau.

Ví dụ 3.6. Ví dụ này mô phỏng sự hoạt động của thuật toán GTAS thỏa mãn thuộc tính $h(k)$. Xét ma trận không tương tự sau:

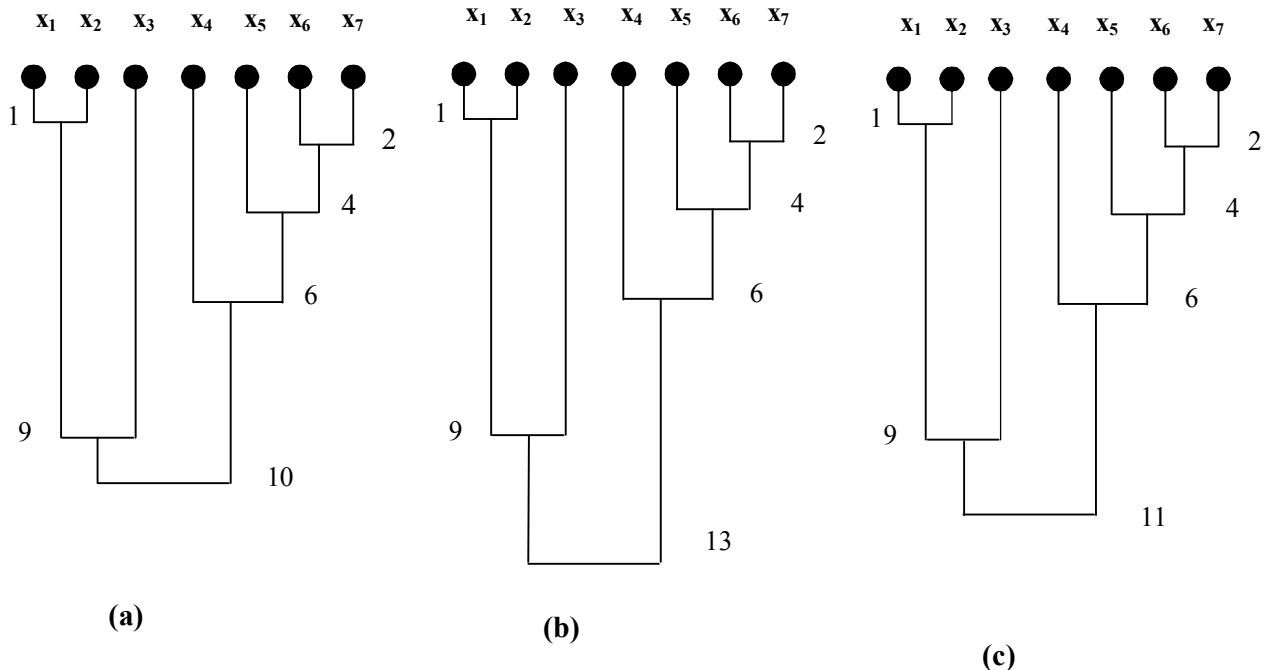
$$P(X) = \begin{bmatrix} 0 & 1 & 9 & 18 & 19 & 20 & 21 \\ 1 & 0 & 8 & 13 & 14 & 15 & 16 \\ 9 & 8 & 0 & 17 & 10 & 11 & 22 \\ 18 & 13 & 17 & 0 & 5 & 6 & 7 \\ 19 & 14 & 10 & 5 & 0 & 3 & 4 \\ 20 & 15 & 11 & 6 & 3 & 0 & 2 \\ 21 & 16 & 22 & 7 & 4 & 2 & 0 \end{bmatrix}$$

Hình 3.9 là đồ thị gần gũi $G(13)$ ứng với ma trận không tương tự này.



Hình 3-9. Đồ thị gần gũi $G(13)$ sinh ra từ ma trận không tương tự P trong ví dụ 3.6

Đặt $h(k)$ là thuộc tính bậc của đỉnh với $k = 2$, nghĩa là mỗi đỉnh có ít nhất hai cạnh liền kề. Sơ đồ ngưỡng chỉ ra trong hình 3.10a. Ở mức không tương tự 1, x_1 và x_2 hình thành một cụm bởi vì $\{x_1\} \cup \{x_2\}$ là đầy đủ trong $G(1)$, mặc dù thực tế thuộc tính $h(2)$ không thoả (nhớ rằng các điều kiện (b_1) và (c) trong công thức (3.22) là độc lập). Tương tự, $\{x_6\}$ và $\{x_7\}$ hình thành nên một cụm ở mức hai. Phép phân cụm tiếp theo được hình thành ở mức bốn, khi đó $\{x_5\} \cup \{x_6, x_7\}$ trở thành đầy đủ trong $G(4)$. Ở mức 6, x_4, x_5, x_6 và x_7 lần đầu tiên nằm trong cùng cụm. Mặc dù đồ thị con này là không đầy đủ nhưng nó thoả mãn $h(2)$. Cuối cùng, ở mức 9, x_1, x_2 và x_3 được đưa vào cùng một cụm. Chú ý rằng mặc dù tất cả các đỉnh của đồ thị có bậc lớn hơn hoặc bằng 2 nhưng phép phân cụm cuối cùng vẫn được hình thành ở mức 10 bởi vì ở mức 9 đồ thị con là không liên thông.



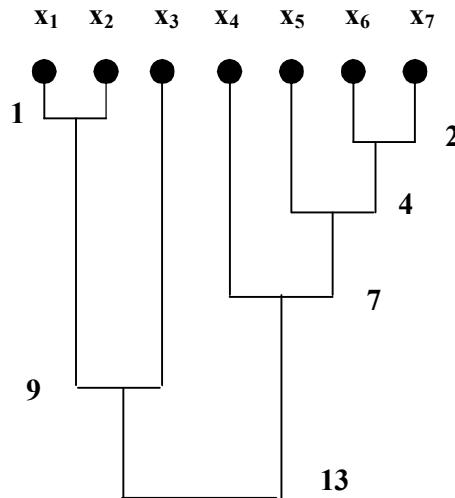
Hình 3-10. Các sơ đồ phân cụm dùng thuật toán GTAS thỏa thuộc tính $h(k)$ của ví dụ 3.6

- Sơ đồ không tương tự sinh ra khi $h(k)$ là thuộc tính bậc của đỉnh với $k = 2$
- Sơ đồ không tương tự sinh ra khi $h(k)$ là thuộc tính khả năng liên kết đỉnh, với $k = 2$
- Sơ đồ không tương tự sinh ra khi $h(k)$ là thuộc tính khả năng liên kết cạnh, với $k = 2$.

Giả sử rằng $h(k)$ là thuộc tính khả năng liên kết đỉnh với $k = 2$ nghĩa là tất cả cặp các đỉnh trong một đồ thị con liên thông được kết nối bởi ít nhất hai đường đi không có đỉnh chung. Sơ đồ không tương tự trong trường hợp này chỉ ra trong hình 3.10b. Cuối cùng, sơ đồ không tương tự sinh ra khi thuộc tính khả năng liên kết cạnh với $k = 2$ chỉ ra trong hình 3.10c.

Không khó để thấy rằng tất cả các thuộc tính với $k = 1$ là kết quả của thuật toán liên kết đơn. Mặt khác, khi tăng k , các đồ thị con tiến gần đến tính đầy đủ.

Giả sử rằng $h(k)$ là thuộc tính bậc của đỉnh với $k = 3$. Sơ đồ tương ứng chỉ ra trong hình 3.11. So sánh các sơ đồ của hình 3.10a và 3.11 chúng ta thấy rằng các cụm được tạo ra giống nhau (có cùng số phần tử) nhưng trong sơ đồ hình 3.11 các cụm được hình thành ở mức độ không tương tự cao hơn.



Hình 3-11. Sơ đồ ngưỡng của ví dụ 3.6 với thuộc tính bậc của đỉnh $k=3$

c. Các thuật toán phân cụm dựa trên cây khung nhỏ nhất.

Cây khung của đồ thị vô hướng có trọng số là một đồ thị liên thông, không có chu trình và trọng số của cây khung là tổng trọng số trên các cạnh của cây khung đó. Cây khung nhỏ nhất (Minimum Spanning Tree - MST) là cây khung có trọng số nhỏ nhất trong tất cả các cây khung của đồ thị. Để tạo ra một MST, có thể sử dụng thuật toán Prim hoặc Kruskal.

Chú ý rằng có thể có nhiều hơn một cây khung nhỏ nhất với mỗi đồ thị đã cho. Tuy nhiên, khi các cạnh của đồ thị G có trọng số khác nhau, thì MST là duy nhất.

Tìm MST cũng được xem như trường hợp đặc biệt của GTAS nếu thay $g_{h(k)}(C_r, C_s)$ bằng hàm gần gũi sau:

$$g(C_r, C_s) = \min_{i,j} \{w_{ij} : x_i \in C_r, x_j \in C_s\} \quad (3.26)$$

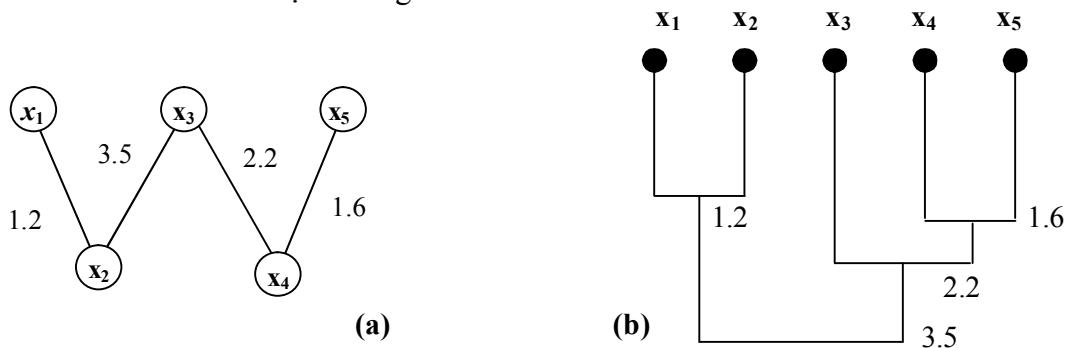
trong đó $w_{ij} = d(x_i, x_j)$

Khi xác định được MST, có thể xây dựng được một cây phân cấp của phép phân cụm như sau: Các cụm của phép phân cụm ở mức t bằng số thành phần liên thông có được tính đến thời điểm đưa cạnh có trọng số nhỏ nhất là t vào đồ thị. Điều này cho thấy rằng cây phân cấp này giống với cây phân cấp định nghĩa bằng thuật toán liên kết đơn (trong trường hợp khoảng cách giữa hai vector bất kỳ của X là khác nhau). Vì vậy, sơ đồ này có thể dùng để thay thế cho thuật toán liên kết đơn. Ví dụ sau mô phỏng hoạt động của sơ đồ này.

Ví dụ 3.7. Xét ma trận gân gùi sau:

$$P = \begin{bmatrix} 0 & 1.2 & 4.0 & 4.6 & 5.1 \\ 1.2 & 0 & 3.5 & 4.2 & 4.7 \\ 4.0 & 3.5 & 0 & 2.2 & 2.8 \\ 4.6 & 4.2 & 2.2 & 0 & 1.6 \\ 5.1 & 4.7 & 2.8 & 1.6 & 0 \end{bmatrix}$$

MST xây dựng từ ma trận gân gùi này đưa ra trong hình 3.12a và sơ đồ phân cụm tương ứng đưa ra trong hình 3.12b. Để thấy rằng cây khung nhỏ nhất là sơ đồ duy nhất của thuật toán liên kết đơn. Do đó, MST có thể sử dụng thay thế cho thuật toán liên kết đơn để tiết kiệm thời gian tính toán.



Hình 3-12. Cây khung nhỏ nhất của ma trận không tương tự (a) và Sơ đồ không tương tự tương ứng khi áp dụng thuật toán dựa trên MST (b) cho trong ví dụ 3.7.

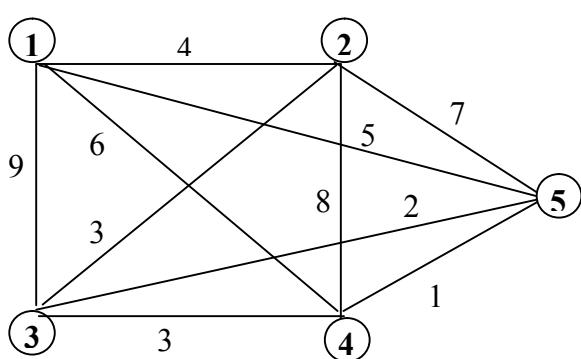
3.2.5. Ảnh hưởng của ma trận gân gùi tới sơ đồ phân cụm

Trong trường hợp mà các vector bao gồm các đặc trưng đo theo khoảng hoặc theo tỷ lệ, khả năng một vector trong tập dữ liệu X có khoảng cách bằng nhau với hai vector khác của X là rất nhỏ đối với hầu hết các bài toán thực tế. Trong trường hợp này, ma trận gân gùi P sẽ có ít nhất hai phần tử có cùng giá trị ở tam giác phía trên đường chéo chính (xem ví dụ 3.8). Điều quan tâm của ta là tìm hiểu xem các thuật toán phân cấp sử dụng các ma trận gân gùi như thế nào. Trước tiên chúng ta xét họ thuật toán dựa trên lý thuyết đồ thị qua ví dụ sau:

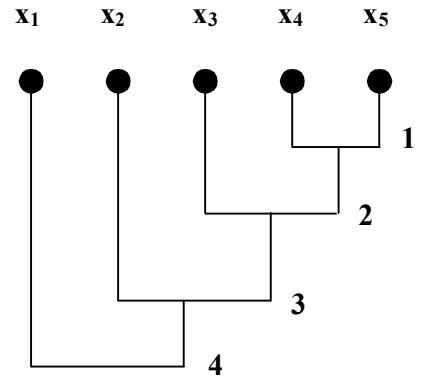
Ví dụ 3.8. Xét ma trận không tương tự sau:

$$P = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 \\ 4 & 0 & [3] & 8 & 7 \\ 9 & 3 & 0 & [3] & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$

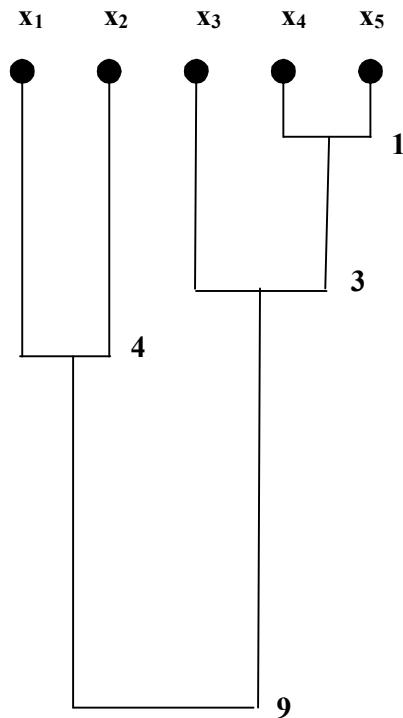
Chú ý rằng $P(2, 3) = P(3, 4) = 3$. Sau đây là đồ thị không tương tự $G(9)$ và các sơ đồ phân cụm khi thực hiện một số thuật toán trong ví dụ này. Chú ý là các sơ đồ của hình 3.13c và 3.13d là khác nhau.



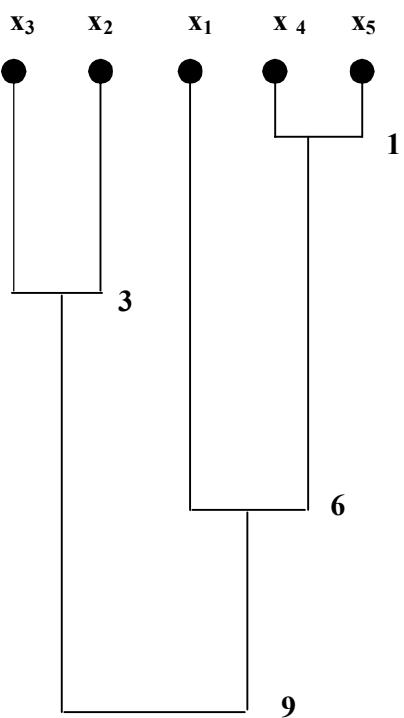
(a)



(b)



(c)



(d)

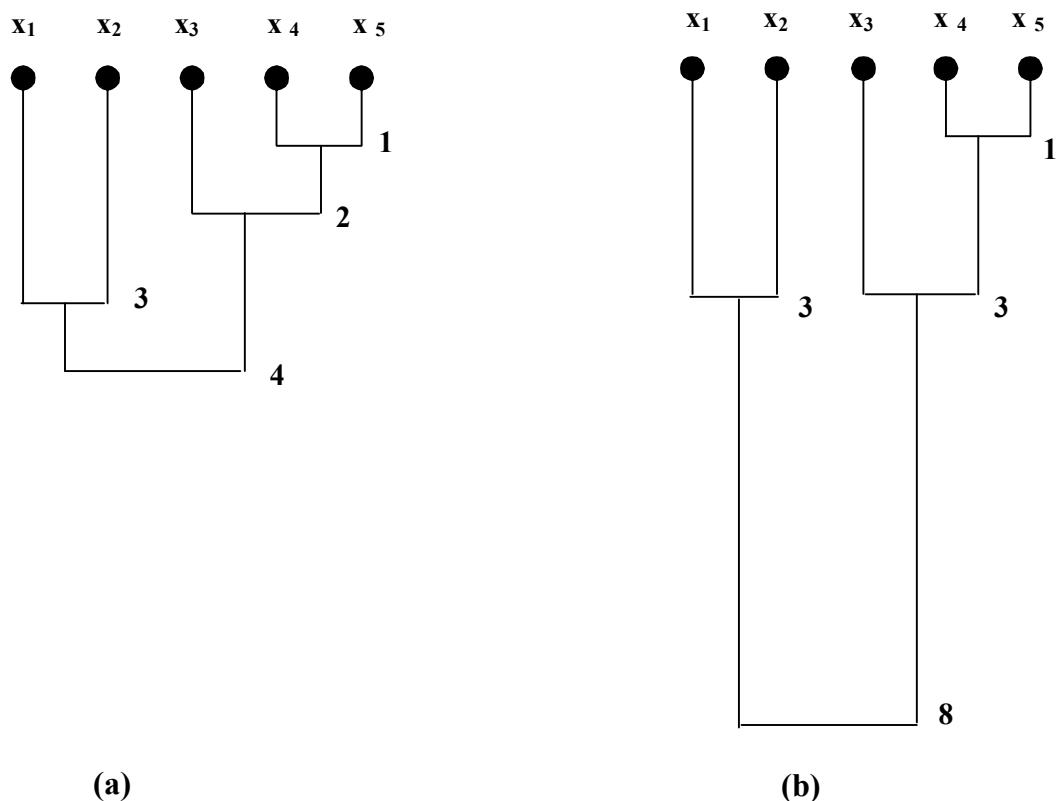
Hình 3-13. Các sơ đồ minh họa cho trường hợp ma trận không tương tự có hai phần tử bằng nhau trong ví dụ 3.8

- (a) Đồ thị không tương tự $G(9)$
- (b) Sơ đồ không tương sinh ra được bởi thuật toán liên kết đơn.
- (c) Sơ đồ không tương sinh ra bởi thuật toán liên kết đầy đủ khi cạnh $(3, 4)$ được xét trước cạnh $(2, 3)$.
- (d) Sơ đồ không tương sinh ra bởi thuật toán liên kết đầy đủ khi cạnh $(2, 3)$ được xét trước cạnh $(3, 4)$.

Tráo đổi $P(1, 2)$ và $P(2, 3)$ của P (vì ma trận gần gũi P là đối xứng nên $P(2, 1)$ và $P(3, 2)$ cũng bị tráo đổi) ta được ma trận không tương tự mới P_1 .

$$P_1 = \begin{bmatrix} 0 & [3] & 9 & 6 & 5 \\ 3 & 0 & 4 & 8 & 7 \\ 9 & 4 & 0 & [3] & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$

Sau đó ta lại vẽ các sơ đồ phân cụm ứng với các thuật toán liên kết đơn và liên kết đầy đủ. Trong trường hợp này, thuật toán liên kết đầy đủ sinh ra sơ đồ không phụ thuộc vào thứ tự cạnh (1, 2) hay (3, 4) được xét trước.



Hình 3-14. Sơ đồ không tương tự đạt được bởi thuật toán liên kết đơn (a) và thuật toán liên kết đầy đủ (b) với ma trận P_1 .

Ví dụ này cho thấy trong ma trận gần gũi có hai phần tử cùng giá trị, thuật toán liên kết đơn chỉ đưa ra một sơ đồ phân cụm, không cần quan tâm tới thứ tự các phần tử được xét. Mặt khác, thuật toán liên kết đầy đủ có thể dẫn tới các sơ đồ khác nhau nếu ta xét các phần tử đó theo những cách khác nhau. Các thuật toán khác dựa trên

lý thuyết đồ thị rơi vào giữa thuật toán liên kết đơn và thuật toán liên kết đầy đủ, có cách thức tương tự với thuật toán liên kết đầy đủ.

Các thuật toán dựa trên lý thuyết ma trận cũng được thực hiện tương tự. Tuy nhiên, trong quá trình biến đổi ma trận, ta có thể nhận được ma trận mới có nhiều phần tử có cùng giá trị ở phía trên đường chéo chính. Thuật toán liên kết đơn hầu như không phụ thuộc vào thứ tự xét các phần tử có cùng giá trị đó và luôn dẫn tới một sơ đồ gần gũi. Đường như yêu cầu đưa thêm vào thuộc tính khả năng liên kết (với các thuật toán dựa trên lý thuyết đồ thị) hoặc công thức (3.4) (với các thuật toán dựa trên lý thuyết ma trận) gây ra sự nhập nhằng và các kết quả phân cụm dễ bị ảnh hưởng bởi thứ tự xử lý các phần tử. Theo nhận xét này, thuật toán liên kết đơn có vẻ tốt hơn các thuật toán khác. Điều này không có nghĩa là tất cả các thuật toán khác không quan trọng. Nếu không sử dụng thuật toán liên kết đơn để xử lý bài toán nào đó, ta phải xem xét cẩn thận các phần tử trong ma trận gần gũi.

3.3. Các thuật toán phân rã - GDS

Các thuật toán phân rã thực hiện ngược với thuật toán tích tụ. Phép phân cụm khởi tạo chỉ có một cụm duy nhất là tập dữ liệu X . Ở bước đầu tiên, chúng ta tìm phân đoạn tốt nhất có thể của X để chia X thành hai cụm. Phương pháp trực tiếp là xét tất cả các cách phân đoạn tập X thành hai tập (2^{N-1} - 1 phân đoạn) sau đó chọn một phân đoạn tốt nhất theo một tiêu chuẩn định trước. Lặp lại cách làm này ở một trong hai tập mới sinh ra. Phép phân cụm cuối cùng gồm N cụm, mỗi cụm chứa một vector của X .

Sau đây sẽ mô tả sơ đồ phân rã tổng quát hình thức hơn. Ở đây, phép phân cụm thứ t chứa $t + 1$ cụm. Tiếp theo, đặt C_{tj} là cụm thứ j của phép phân cụm \mathfrak{R}_t (phép phân cụm thứ t); $t = 0, \dots, N-1; j = 1, \dots, t+1$. Đặt $g(C_i, C_j)$ là hàm không tương tự định nghĩa trên tất cả các cặp cụm có thể. Phép phân cụm khởi tạo \mathfrak{R}_0 chỉ chứa một tập duy nhất X , nghĩa là $C_{01} = X$. Để xác định phép phân cụm tiếp theo, chúng ta xét tất cả các cặp cụm mà nó hình thành nên một phân đoạn của X . Trong số các cặp cụm đó, chọn một cặp chẵng hạn (C_{11}, C_{12}) làm hàm g lớn nhất. Hai cụm đó hình thành nên phép phân cụm tiếp theo \mathfrak{R}_1 , tức là $\mathfrak{R}_1 = \{C_{11}, C_{12}\}$. Bước tiếp theo, chúng ta xét tất cả các cặp cụm có thể sinh ra bởi C_{11} và chọn một cặp cụm làm g lớn nhất. Lặp lại thủ tục này với C_{12} . Nay giả sử rằng từ hai cặp cụm kết quả, một cặp cụm kết quả bắt nguồn từ C_{11} có giá trị g lớn hơn. Cặp cụm này biểu thị là (C_{11}^1, C_{11}^2) . Sau đó, phép phân cụm mới \mathfrak{R}_2 chứa C_{11}^1, C_{11}^2 và C_{12} . Gán nhãn lại cho các

cụm này theo thứ tự là C_{21}, C_{22}, C_{23} ta có $\mathfrak{R}_2 = \{C_{21}, C_{22}, C_{23}\}$. Tiếp tục theo cách này sẽ hình thành nên tất cả các phép phân cụm tiếp theo. Sơ đồ phân rã tổng quát viết như sau:

Sơ đồ phân rã tổng quát (Generalized Divisive Scheme - GDS)

1. Khởi tạo

- 1.1. Chọn $\mathfrak{R}_0 = \{X\}$ là phép phân cụm khởi tạo
- 1.2. $t = 0$

2. Repeat

- 2.1. $t = t + 1$
- 2.2. **For** $i=1$ **to** t
 - 2.2.1. Trong tất cả các cặp cụm có thể (C_r, C_s) mà nó hình thành từ một phân đoạn tốt nhất của $C_{t-1,i}$, tìm cặp $(C_{t-1,i}^1, C_{t-1,i}^2)$ để giá trị $g(C_{t-1,i}^1, C_{t-1,i}^2)$ lớn nhất.
 - 2.3. Từ t cặp định nghĩa trong các bước trước, chọn một cặp để g lớn nhất. Giả sử cặp đó là $(C_{t-1,j}^1, C_{t-1,j}^2)$
 - 2.4. Phép phân cụm mới là: $\mathfrak{R}_t = (\mathfrak{R}_{t-1} \setminus \{C_{t-1,j}\}) \cup \{C_{t-1,j}^1, C_{t-1,j}^2\}$
 - 2.5. Gán lại nhãn cho các cụm của \mathfrak{R}_t .

Until <mỗi vector nằm trong một cụm>.

Các lựa chọn khác nhau của g dẫn tới các thuật toán khác nhau.

Ví dụ 3.9. Ví dụ này mô minh họa cho sơ đồ phân rã tổng quát

Trong không gian toạ độ thực 2 chiều, cho 5 điểm có toạ độ sau:

$$X = \{x_1 = (1, 0), x_2 = (3, 0), x_3 = (1, 1), x_4 = (5, 3), x_5 = (5, 4)\}.$$

Sử dụng độ đo khoảng cách Euclid và phân cụm theo điểm đại diện là vector trong bình. Trước hết ta phân đoạn tập X thành hai tập ứng với hai cụm C_1 và C_2 (có $2^{5-1}-1 = 15$ phân đoạn) sau đó tính độ đo gần gũi giữa hai cụm (mức độ không tương tự) vừa tạo ra, kết quả tính toán như bảng sau:

Phân đoạn	Cụm C ₁	Cụm C ₂	Vector đại diện của C ₁	Vector đại diện của C ₂	Khoảng cách giữa hai cụm
1	x_1	x_2, x_3, x_4, x_5	(1, 0)	$(\frac{14}{4}, \frac{8}{4})$	3.20
2	x_2	x_1, x_3, x_4, x_5	(3, 0)	$(\frac{12}{4}, \frac{8}{4})$	2.00
3	x_3	x_1, x_2, x_4, x_5	(1, 1)	$(\frac{14}{4}, \frac{7}{4})$	2.61
4	x_4	x_1, x_2, x_3, x_5	(5, 3)	$(\frac{10}{4}, \frac{5}{4})$	3.05
5	x_5	x_1, x_2, x_3, x_4	(5, 4)	$(\frac{10}{4}, \frac{4}{4})$	3.90
6	x_1, x_2	x_3, x_4, x_5	(2, 0)	$(\frac{11}{3}, \frac{8}{3})$	3.14
7	x_1, x_3	x_2, x_4, x_5	$(1, \frac{1}{2})$	$(\frac{13}{3}, \frac{7}{3})$	3.80
8	x_1, x_4	x_2, x_3, x_5	$(3, \frac{3}{2})$	$(\frac{9}{3}, \frac{5}{3})$	0.17
9	x_1, x_5	x_2, x_3, x_4	(3, 2)	$(\frac{9}{3}, \frac{4}{3})$	0.67
10	x_2, x_3	x_1, x_4, x_5	$(2, \frac{1}{2})$	$(\frac{11}{3}, \frac{7}{3})$	2.48
11	x_2, x_4	x_1, x_3, x_5	$(4, \frac{3}{2})$	$(\frac{7}{3}, \frac{5}{3})$	0.85
12	x_2, x_5	x_1, x_3, x_4	(4, 2)	$(\frac{7}{3}, \frac{4}{3})$	1.80
13	x_3, x_4	x_1, x_2, x_5	(3, 2)	$(\frac{9}{3}, \frac{4}{3})$	0.67
14	x_3, x_5	x_1, x_2, x_4	$(3, \frac{5}{2})$	$(\frac{9}{3}, \frac{3}{3})$	1.50
15	x_4, x_5	x_1, x_2, x_3	$(5, \frac{7}{2})$	$(\frac{5}{3}, \frac{1}{3})$	4.60

Trong các cách phân đoạn trên, chọn hai cụm $C_1 = \{x_4, x_5\}$; $C_2 = \{x_1, x_2, x_3\}$ có khoảng cách $g(C_1, C_2) = 4.60$ là lớn nhất. Do vậy ta có phép phân cụm

$$\mathfrak{R}_1 = \{\{x_4, x_5\}, \{x_1, x_2, x_3\}\}$$

Chọn cụm C_2 để phân đoạn tiếp, ta có bảng sau:

Phân đoạn	Cụm C_{21}	Cụm C_{22}	Vector đại diện của C_{21}	Vector đại diện của C_{22}	Khoảng cách giữa hai cụm
1	x_1	x_2, x_3	(1, 0)	$(7, \frac{1}{2})$	1.12
2	x_2	x_1, x_3	(3, 0)	$(1, \frac{1}{2})$	2.06
3	x_3	x_1, x_2	(1, 1)	(2, 0)	1.41

Trong các cách phân đoạn trên, chọn hai cụm $C_{21} = \{x_2\}; C_{22} = \{x_1, x_3\}$ có khoảng cách $g(C_{21}, C_{22}) = 2.06$ lớn nhất. Như vậy, phép phân cụm \mathfrak{R}_2 có 3 cụm

$$\mathfrak{R}_2 = \{\{x_4, x_5\}, \{x_1, x_3\}, \{x_2\}\}$$

Làm tương tự ta có phép phân cụm \mathfrak{R}_3 có 4 cụm

$$\mathfrak{R}_3 = \{\{x_4\}, \{x_5\}, \{x_1, x_3\}, \{x_2\}\}$$

và cuối cùng là phép phân cụm \mathfrak{R}_4 có 5 cụm

$$\mathfrak{R}_4 = \{\{x_4\}, \{x_5\}, \{x_1\}, \{x_3\}, \{x_2\}\}$$

Quá trình phân cụm kết thúc. (xem hình 3.15)

3.3.1. Cải tiến sơ đồ GDS

Một điều dễ thấy ở trên là sơ đồ phân rã này được tính toán rất chật chẽ. Đó là hạn chế chính của sơ đồ này so với sơ đồ tích tụ. Do đó, để các sơ đồ đó được sử dụng tốt trong thực tế, cần yêu cầu sự đơn giản hơn trong tính toán. Có một cách là tạo ra sự thoả hiệp mà không cần tìm tất cả các phân đoạn có thể của một cụm. Điều này có thể thực hiện bằng cách loại bỏ một số phân đoạn “không thích hợp” theo một tiêu chuẩn định trước. Mỗi bước trong quá trình phân cụm như sau:

Đặt C_i là cụm vừa được hình thành. Mục đích của là tiếp tục chia C_i thành hai cụm C_i^1 và C_i^2 sao cho mức độ không tương tự giữa hai cụm là lớn nhất. Khởi tạo, $C_i^1 = \emptyset$ và $C_i^2 = C_i$. Sau đó, chọn một vector trong C_i^2 mà mức độ không tương tự trung bình của nó với các vector còn lại là lớn nhất và chuyển nó vào C_i^1 . Tiếp theo với mỗi $x \in C_i^2$, tính mức độ không tương tự trung bình của nó với các vector của $C_i^1, g(x, C_i^1)$ và mức độ không tương tự trung bình của nó với các vector còn lại trong $C_i^2, g(x, C_i^2 \setminus \{x\})$. Nếu với mọi $x \in C_i^2, D(x) = g(x, C_i^2 \setminus \{x\}) - g(x, C_i^1) < 0$ thì

kết thúc. Lúc này C_i được phân thành hai cụm C_i^1 và C_i^2 . Ngược lại, chọn vector $x \in C_i^2$ để $D(x)$ lớn nhất và đưa x vào C_i^1 . Thủ tục được lặp lại cho đến khi gặp tiêu chuẩn kết thúc. Lặp đi lặp lại cách này để tính toán ở bước 2.2.1 của GDS.

Thuật toán cải tiến được viết như sau:

Bước lặp i : Phân cụm C_i thành hai cụm C_i^1 và C_i^2

Thuật toán GDS cải tiến: Phân cụm C_i thành hai cụm C_i^1 và C_i^2

- **Khởi tạo:** $C_i^1 := \emptyset; C_i^2 := C_i;$
- Chọn vector $x \in C_i^2$ mà $g(x, C_i^2 \setminus \{x\})$ đạt giá trị lớn nhất
- $C_i^1 := C_i^1 \cup \{x\}; C_i^2 := C_i^2 \setminus \{x\}$
- **For** < mõi vector $x \in C_i^2$ > **do**
 - Tính $g(x, C_i^1); g(x, C_i^2 \setminus \{x\})$
 - Đặt $D(x) := g(x, C_i^2 \setminus \{x\}) - g(x, C_i^1)$
- **If** $(D(x) < 0, \forall x \in C_i^2)$ **then** <Kết thúc. Ta được 2 cụm C_i^1 và C_i^2 >
- **Else**
 - Chọn vector $x \in C_i^2$ để $D(x)$ lớn nhất
 - $C_i^1 := C_i^1 \cup \{x\}; C_i^2 := C_i^2 \setminus \{x\}$
- Lặp lại quá trình trên với các cụm C_i^1 và C_i^2

Ví dụ 3.10.

Áp dụng thuật toán cải tiến thực hiện ví dụ 3.9

Trong không gian toạ độ thực 2 chiều, cho 5 điểm có toạ độ sau:

$$X = \{x_1 = (1, 0), x_2 = (3, 0), x_3 = (1, 1), x_4 = (5, 3), x_5 = (5, 4)\}.$$

Sử dụng độ đo khoảng cách Euclid và phân cụm theo điểm đại diện là vector trong bình.

Đặt $C_1 = \{x_1, x_2, x_3, x_4, x_5\}$. Phân cụm C_1 thành 2 cụm C_1^1 và C_1^2

- Khởi tạo: $C_1^1 := \emptyset; C_1^2 := C_1;$
- Chọn vector $x \in C_1^2$ mà $g(x, C_1^2 \setminus \{x\})$ đạt giá trị lớn nhất

Để chọn vector x , ta lập bảng sau:

Cụm C_1^1	Cụm C_1^2	Vector đại diện của C_1^1	Vector đại diện của C_1^2	Khoảng cách giữa hai cụm
x_1	x_2, x_3, x_4, x_5	(1, 0)	$(\frac{14}{4}, \frac{8}{4})$	3.20
x_2	x_1, x_3, x_4, x_5	(3, 0)	$(\frac{12}{4}, \frac{8}{4})$	2.00
x_3	x_1, x_2, x_4, x_5	(1, 1)	$(\frac{14}{4}, \frac{7}{4})$	2.61
x_4	x_1, x_2, x_3, x_5	(5, 3)	$(\frac{10}{4}, \frac{5}{4})$	3.05
x_5	x_1, x_2, x_3, x_4	(5, 4)	$(\frac{10}{4}, \frac{4}{4})$	3.90

Ta thấy khoảng cách giữa hai cụm lớn nhất bằng 3.90. Như vậy vector x_5 được chọn để đưa vào cụm C_1^1 , tức là $C_1^1 = \{x_5\}; C_1^2 = \{x_1, x_2, x_3, x_4\}$.

Chọn thêm một vector thuộc C_1^2 để đưa thêm vào C_1^1 . Tức là thực hiện vòng lặp

For \langle mỗi vector $x \in C_1^2 \rangle$ **do**

- Tính $g(x, C_1^1); g(x, C_1^2 \setminus \{x\})$
- Đặt $D(x) := g(x, C_1^2 \setminus \{x\}) - g(x, C_1^1)$

Quá trình tính toán như bảng sau:

x	Toạ độ của x	Tập vector của C_1^1	Vector trung bình cụm C_1^1	Tập vector của $C_1^2 \setminus \{x\}$	Vector trung bình cụm $C_1^2 \setminus \{x\}$	$g(x, C_1^2 \setminus \{x\})$	$g(x, C_1^1)$	$D(x)$
x_1	(1, 0)	x_5	(5, 4)	$\{x_2, x_3, x_4\}$	(3.0, 1.33)	2.4	5.66	-3.26
x_2	(3, 0)	x_5	(5, 4)	$\{x_1, x_3, x_4\}$	(2.33, 1.33)	1.49	4.47	-2.98
x_3	(1, 1)	x_5	(5, 4)	$\{x_1, x_2, x_4\}$	(3, 1)	2.0	5.0	-3.0
x_4	(5, 3)	x_5	(5, 4)	$\{x_1, x_2, x_3\}$	(2.66, 0.33)	4.27	1.0	3.27

Ta thấy vector x_4 có khoảng cách đến $C_1^2 \setminus \{x_4\}$ là lớn nhất nên x_4 được chọn để đưa thêm vào C_1^1 . Do vậy $C_1^1 = \{x_4, x_5\}$ và $C_1^2 = \{x_1, x_2, x_3\}$

Lặp lại quá trình trên để chọn một vector x thuộc C_1^2 đưa thêm vào C_1^1 .

Để chọn vector x , ta lập bảng sau:

x	Toạ độ của x	Tập vector của C_1^1	Vector trung bình cụm C_1^1	Tập vector của $C_1^2 \setminus \{x\}$	Vector trung bình cụm $C_1^2 \setminus \{x\}$	$g(x, C_1^2 \setminus \{x\})$	$g(x, C_1^1)$	$D(x)$
x_1	(1, 0)	$\{x_4, x_5\}$	(5, 3.5)	$\{x_2, x_3\}$	(2.0, 0.5)	1.12	5.31	- 4.19
x_2	(3, 0)	$\{x_4, x_5\}$	(5, 3.5)	$\{x_1, x_3\}$	(1.0, 0.5)	2.06	4.03	- 1.97
x_3	(1, 1)	$\{x_4, x_5\}$	(5, 3.5)	$\{x_1, x_2\}$	(2.0, 0)	1.41	4.71	- 3.30

Ta thấy $D(x) < 0$ với mọi $x \in C_1^2$. Không đưa thêm vector nào vào C_1^1 nữa. Do vậy quá trình lặp dừng lại và ta có hai cụm $C_1^1 = \{x_4, x_5\}$ và $C_1^2 = \{x_1, x_2, x_3\}$. Phép phân cụm $\mathfrak{R}_1 = \{\{x_4, x_5\}, \{x_1, x_2, x_3\}\}$

Lặp lại quá trình trên với cụm $C_1^2 = \{x_1, x_2, x_3\}$ ta phép phân cụm

$$\mathfrak{R}_2 = \{\{x_4, x_5\}, \{x_1, x_3\}, \{x_2\}\}$$

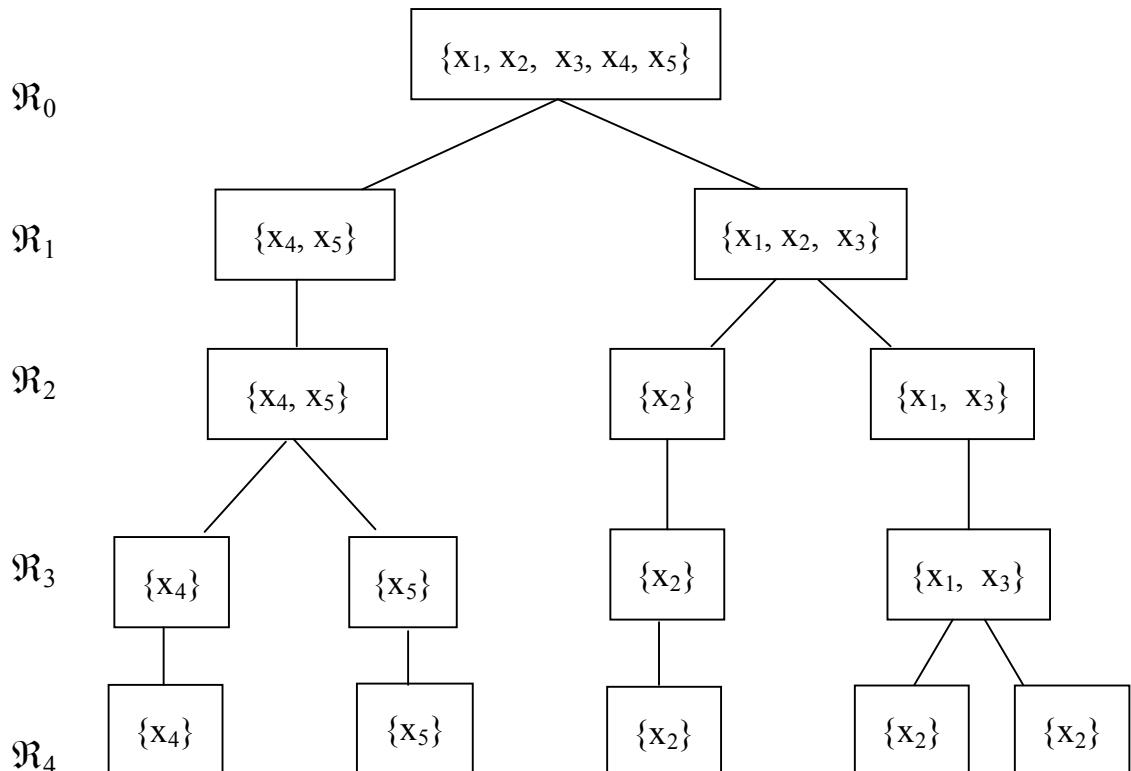
Làm tương tự ta có phép phân cụm \mathfrak{R}_3 có 4 cụm

$$\mathfrak{R}_3 = \{\{x_4\}, \{x_5\}, \{x_1, x_3\}, \{x_2\}\}$$

và phép phân cụm \mathfrak{R}_4 có 5 cụm

$$\mathfrak{R}_4 = \{\{x_4\}, \{x_5\}, \{x_1\}, \{x_3\}, \{x_2\}\}.$$

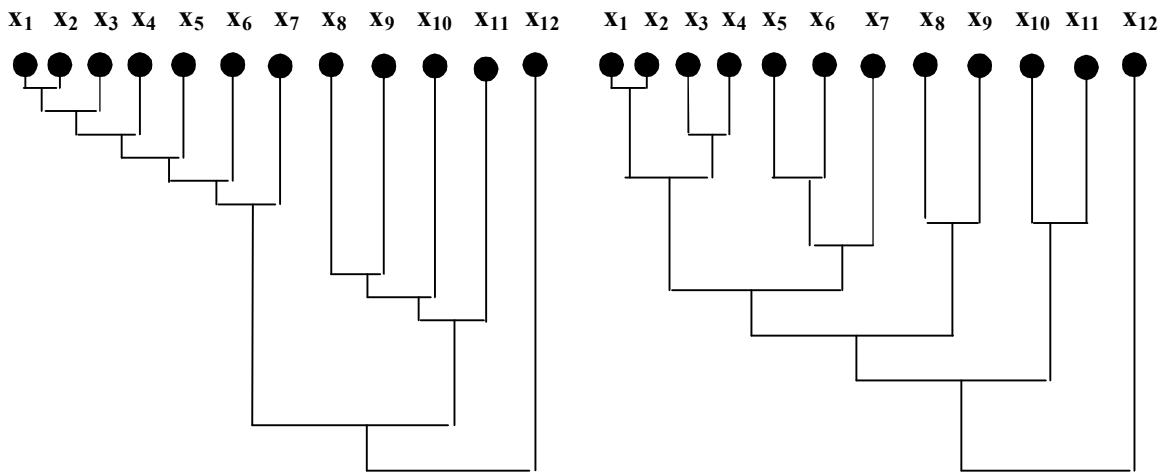
Cả hai ví dụ 3.9 và 3.10 cho cùng kết quả, mô tả bởi sơ đồ sau:



Hình 3-15. Minh họa các bước phân cụm của sơ đồ GDS

3.4. Lựa chọn phân cụm tốt nhất

Trên đây, chúng ta tập trung vào các thuật toán phân cụm phân cấp. Tiếp theo, chúng ta bàn về cách để xác định phép phân cụm tốt nhất trong một cây phân cấp đã cho. Rõ ràng, điều này tương đương với việc chọn ra các cụm phù hợp với dữ liệu. Một cách tiếp cận bằng trực giác là tìm trong sơ đồ gần gũi các cụm có "*thời gian sống*" (lifetime) lớn. "*Thời gian sống*" của một cụm được định nghĩa là giá trị tuyệt đối của hiệu giữa các mức độ gần gũi ở đó cụm đó được tạo ra và mức độ gần gũi ở đó nó bị sáp nhập vào một cụm lớn hơn.



Hình 3-16. Sơ đồ trong trường hợp có hai cụm chính (a) và có cụm duy nhất (b) trong tập dữ liệu.

Ví dụ, sơ đồ 3.16a với hai cụm chính được sinh ra và sơ đồ hình 3.16b có một cụm duy nhất.

Tiếp theo, chúng ta thảo luận hai phương pháp đã được đề xuất trong [5] để xác định phép phân cụm phù hợp với dữ liệu; thích hợp với các thuật toán tích tụ. Thuật toán phân cụm không cần đưa ra toàn bộ cây phân cấp của N cụm, nhưng nó kết thúc khi phép phân cụm phù hợp với dữ liệu đã đạt được theo một tiêu chuẩn.

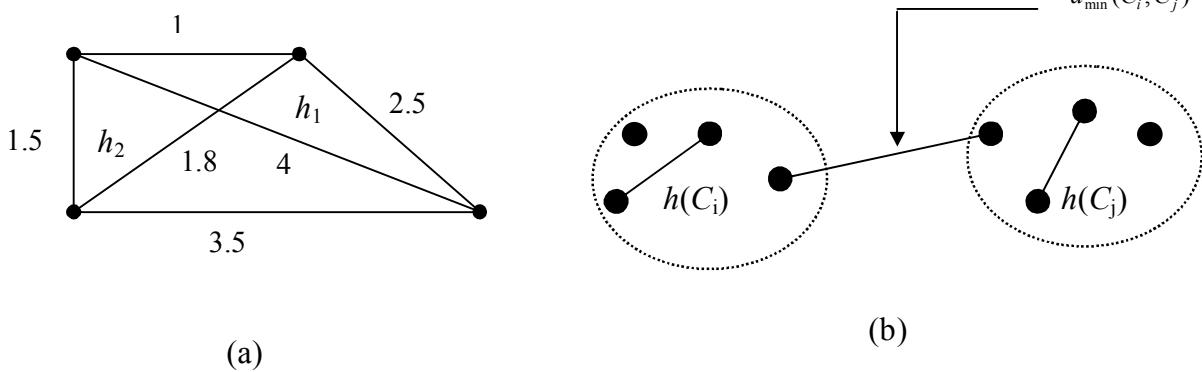
Phương pháp I: Đây là phương pháp không bắn chát, người sử dụng cần:

- Xác định giá trị của một tham số đặc trưng.
- Định nghĩa hàm $h(C)$ đo mức độ không tương tự giữa các vector của cùng cụm C . Nghĩa là, chúng ta có thể coi nó như là một độ đo “*tự - tương tự*”. Chẳng hạn, có thể định nghĩa $h(C)$ là:

$$h_1(C) = \max\{d(x, y) \mid x, y \in C\} \quad (3.28)$$

$$\text{hoặc} \quad h_2(C) = \text{med}\{d(x, y) \mid x, y \in C\} \quad (3.29)$$

xem hình 3.17a



Hình 3-17. Ví dụ về độ đo “Tự - tương tự” (a) và mô phỏng điều kiện kết thúc của phương pháp II (b)

Khi d là khoảng cách Metric, $h(C)$ được định nghĩa là:

$$h(C) = \frac{1}{2n_C} \sum_{x \in C} \sum_{y \in C} d(x, y) \quad (3.30)$$

với n_C là số phần tử của C .

Đặt θ là ngưỡng của $h(C)$. Khi đó, thuật toán kết thúc ở phép phân cụm \mathfrak{R}_t nếu:

$$\exists C_j \in \mathfrak{R}_{t+1} : h(C_j) > \theta \quad (3.31)$$

Tức là: \mathfrak{R}_t là phép phân cụm cuối cùng nếu tồn tại một cụm C trong \mathfrak{R}_{t+1} mà sự không tương tự giữa các vector của nó ($h(C)$) lớn hơn θ .

Đôi khi, ngưỡng θ được định nghĩa là:

$$\theta = \mu + \lambda\sigma \quad (3.32)$$

ở đó μ là khoảng cách trung bình giữa hai vector bất kỳ trong X và σ là dung sai của θ . Tham số λ là tham số do người dùng định nghĩa. Vì vậy, nhu cầu cần chỉ rõ giá trị thích hợp của θ được chuyển thành việc lựa chọn λ . Tuy nhiên, λ có thể được ước lượng hợp lý hơn θ .

Phương pháp II: Đây là phương pháp bản chất; nghĩa là, trong trường hợp này chỉ xem xét cấu trúc của tập dữ liệu X . Theo phương pháp này, phép phân cụm cuối cùng \mathfrak{R}_t phải thoả quan hệ sau:

$$d_{\min}^{ss}(C_i, C_j) > \max\{h(C_i), h(C_j)\}, \forall C_i, C_j \in \mathfrak{R}_t \quad (3.33)$$

Nghĩa là: trong phép phân cụm cuối cùng, mức độ không tương tự giữa hai cụm lớn hơn mức độ “tự - tương tự” của mỗi cụm (xem hình 3.16b). Ở đây d_{\min}^{ss} là độ đo giàn gũi đã định nghĩa trong chương 1. Chú ý rằng, đây chỉ là điều kiện cần.

Cuối cùng, phải thấy rằng tất cả các phương pháp đó dựa theo kinh nghiệm (heuristic) và chúng chỉ biểu thị phép phân cụm tốt nhất. Kết quả phân cụm cuối cùng phụ thuộc nhiều vào tính chủ quan của các chuyên gia.

Bài tập chương 3

Bài 3.1. Xét khoảng cách Euclid là độ đo gần gũi giữa 2 vector. Chỉ xét các vector theo thang khoảng (ratio-scaled)

Chứng minh rằng:

- (a) Một ma trận mẫu có duy nhất một ma trận gần gũi tương ứng
- (b) Một ma trận gần gũi không có duy nhất một ma trận mẫu.

Hơn nữa, có nhiều ma trận gần gũi không tương ứng với bất kỳ ma trận mẫu nào.

Gợi ý: (b) Xét phép tịnh tiến các điểm trong tập dữ liệu X.

Bài 3.2.

Từ công thức (3.10) suy ra công thức (3.8)

(Derive Eq(3.10) from Eq (3.8))

Gợi ý: Sử dụng đồng nhất sau

$$n_3m_3 = n_1m_1 + n_2m_2 \quad (3.34)$$

Và

$$n_1\|m_1 - m_3\|^2 + n_2\|m_2 - m_3\|^2 = \frac{n_1n_2}{n_1 + n_2}\|m_1 - m_2\|^2 \quad (3.35)$$

Ở đây C_1 và C_2 là 2 cụm bất kỳ và $C_3 = C_1 \cup C_2$

Bài 3.3. Chứng minh rằng với thuật toán WPGMC $d_{qs} \geq \min(d_{is}, d_{js})$

Bài 3.4. Chứng minh

$$d'_{qs} = \frac{n_q \cdot n_s}{n_q + n_s} d_{qs} \quad (3.36)$$

Gợi ý: Nhân 2 vế

$$\|m_q - m_s\|^2 = \frac{n_i}{n_i + n_j} d_{is} + \frac{n_j}{n_i + n_j} d_{js} - \frac{n_i n_j}{(n_i + n_j)^2} d_{ij}$$

với

$$\frac{(n_i + n_j)n_s}{n_i + n_j + n_s} \quad (\text{this equation holds from problem 3.2})$$

Bài 3.5. (a) Chứng minh công thức (3.19)

(b) Hoàn thành chứng minh công thức (3.18)

Gợi ý: Bình phương 2 vế công thức (3.18)

Bài 3.6. Xét ma trận gân gùi trong ví dụ 3.5. Tìm sơ đồ gân gùi xuất phát từ thuật toán GTAS khi $h(k)$ là:

- (a) Thuộc tính liên kết nút (node connectivity property)
- (b) Thuộc tính liên kết cạnh (edge connectivity property) với $k = 3$

Bài 3.7. Chứng minh rằng khoảng cách giữa 2 cụm C_r và C_s , $d(C_r, C_s)$ mà 2 cụm này có cùng mức của sự phân cấp sinh ra bởi thuật toán liên kết đơn, cho bởi công thức (3.4) có thể viết như sau:

$$d(C_r, C_s) = \min_{x \in C_r, y \in C_s} d(x, y) \quad (3.37)$$

Nghĩa là, các thuật toán liên kết đơn dựa trên lý thuyết ma trận và lý thuyết đồ thị là tương đương.

Gợi ý: Quy nạp theo mức của sự phân cấp t. Xét các phép phân cụm \mathfrak{R}_t và \mathfrak{R}_{t+1} có chung $N-t-2$ cụm.

Bài 3.8. Chứng minh rằng khoảng cách giữa 2 cụm C_r và C_s , $d(C_r, C_s)$ mà nó có cùng mức của sự phân cấp sinh ra bởi thuật toán liên kết đầy đủ cho bởi công thức (3.5) có thể viết như sau:

$$d(C_r, C_s) = \max_{x \in C_r, y \in C_s} d(x, y) \quad (3.38)$$

Gợi ý: Xem gợi ý trong bài tập trên

Bài 3.9. Trình bày và chứng minh các định đề của 2 bài toán trước khi sử dụng độ đo tương tự giữa 2 vector.

Bài 3.10. Xét ma trận gân gùi sau:

$$P = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 \\ 4 & 0 & 1 & 8 & 7 \\ 9 & 1 & 0 & 3 & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$

Áp dụng các thuật toán liên kết đơn và liên kết đầy đủ với ma trận P và giải thích các sơ đồ kết quả.

Bài 3.11. Xét ma trận không tương tự P trong ví dụ 3.5. Thay $P(3,4) = 6$ ($P(4,6)$ cũng bằng 6). Đặt $h(k)$ là thuộc tính bậc của nút với $k = 2$. Thực hiện thuật toán lý thuyết đồ thị tương ứng khi

- (a) Cạnh (3,4) xét đầu tiên
- (b) Cạnh (4,6) xét đầu tiên

Giải thích các sơ đồ kết quả

Bài 3.12. Xét ma trận không tương tự sau:

$$P = \begin{bmatrix} 0 & 4 & 9 & 6 & 5 \\ 4 & 0 & 3 & 8 & 7 \\ 9 & 3 & 0 & 3 & 2 \\ 6 & 8 & 3 & 0 & 1 \\ 5 & 7 & 2 & 1 & 0 \end{bmatrix}$$

- a) Sử dụng thuật toán liên kết đơn và thuật toán liên kết đầy đủ hãy xác định tất cả các sơ đồ kết quả có thể, với ma trận P và giải thích kết quả.
- b) Đặt $P(3, 4) = 4$, $P(1,2) = 10$, và đặt P_1 là ma trận gàn gũi mới. Chú ý rằng P_1 không chứa ràng buộc. Xác định tất cả các sơ đồ kết quả có thể từ việc ứng dụng thuật toán UPGMA với P_1 .