ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

KHOA TOÁN - TIN
Faculty of Mathematics and Informatics

PHÂN TÍCH SỐ LIỆU

Chủ đề: Clustering

Giảng viên hướng dẫn: ThS. Lê Xuân Lý

Trần Anh Quân	20227202
Nguyễn Thế Thiện	20227264
Nguyễn Quang Thuận	20227184
Phạm Xuân Trường	20227269
Đào Đình Hân	20227229
Nguyễn Ngọc Toàn	20227185
Lê Thế Quang	20227203
Nguyễn Nam Thắng	20227183
Đỗ Văn Thiện	20227263
Nguyễn Trường Sơn	20227260

Nhóm thực hiện: Nhóm 9 - Lớp 155349 - Học kỳ 2024.1

Ngày 30 tháng 4 năm 2025

Đánh giá thành viên nhóm

Tên	Mã số	Điểm	Nhiệm vụ
Nguyễn Nam Thắng	20227183	1.5	K-Means
Nguyễn Thế Thiện	20227264	1.5	K-Means
Trần Anh Quân	20227202	1.5	Phân cụm phân cấp
Phạm Xuân Trường	20227269	1.5	Thuật toán MDS
Nguyễn Quang Thuận	20227184	1.5	Phân cụm phân cấp
Đỗ Văn Thiện	20227263	1.5	Độ đo tương tự
Đào Đình Hân	20227229	1.5	Thuật Toán DBSCAN
Nguyễn Trường Sơn	20227260	1.5	Phân cụm dựa trên mô hình thống kê
Lê Thế Quang	20227203	1.5	Code MDS
Nguyễn Ngọc Toàn	20227185	1.5	Thuật Toán MDS

Bảng 1: Đánh giá thành viên nhóm 9 - Lớp 155349

Mục lục

1 Giới Thiệu

Phân cụm (clustering) là một trong những bài toán được nghiên cứu rộng rãi nhất trong lĩnh vực khai phá dữ liệu (data mining) và học máy (machine learning). Trong suốt hơn 50 năm qua, bài toán phân cụm đã thu hút sự quan tâm của các nhà nghiên cứu từ nhiều lĩnh vực khác nhau. Ứng dụng của phân cụm rất đa dạng, từ xử lý tin nhắn, đa phương tiện, mạng xã hội cho đến các nghiên cứu trong lĩnh vực sinh học. Hơn nữa, phân cụm còn được áp dụng trong nhiều tình huống khác nhau, chẳng hạn như truyền tải dữ liệu trực tuyến và phân tích dữ liệu không xác định. Đây là một chủ đề phong phú, và các thuật toán phân cụm thường được thiết kế tùy theo đặc thù của dữ liệu và tình huống bài toán cụ thể.

Clustering (Phân cụm) là một kỹ thuật phân tích dữ liệu trong đó các đối tượng hoặc điểm dữ liệu được nhóm lại với nhau dựa trên sự tương đồng hoặc khoảng cách giữa chúng. Kỹ thuật này thường được dùng trong giai đoạn thăm dò dữ liệu để tìm ra các mẫu hoặc mối quan hệ tiềm ẩn mà chưa có giả định hoặc cấu trúc nào được biết trước.

Sự khác biệt giữa phân cụm và phân loại: phân loại (Classification) là kỹ thuật gán các đối tượng vào các nhóm đã biết trước. Chẳng hạn, nếu bạn biết rằng trong dữ liệu của mình có ba nhóm (nhóm A, nhóm B và nhóm C), mục tiêu của bạn là xác định mỗi đối tượng mới thuộc vào nhóm nào. Khác với phân loại, phân cụm không có thông tin trước về số lượng nhóm hoặc cấu trúc của các nhóm. Nói cách khác, phân cụm giúp bạn tự động tìm ra các nhóm tự nhiên có trong dữ liệu mà không có bất kỳ giả định nào về số lượng nhóm hoặc đặc điểm của chúng.

Để minh họa bản chất của sự khó khăn trong việc xác định một nhóm tự nhiên, hãy xem xét việc sắp xếp 16 lá bài trong một bộ bài tiêu chuẩn thành các cụm các đối tượng tương tự. Để tạo thành một nhóm duy nhất từ 16 lá bài hình, có một cách duy nhất. Có 32.767 cách để chia các lá bài hình thành hai nhóm (có kích thước khác nhau), và có 7.141.686 cách để sắp xếp các lá bài thành ba nhóm (có kích thước khác nhau), và cứ thế tiếp tục. Rõ ràng, việc giới hạn thời gian làm cho không thể xác định được các cách nhóm tối ưu nhất cho các đối tượng tương tự từ danh sách tất cả các cấu trúc có thể có. Ngay cả những máy tính nhanh nhất cũng dễ dàng bị quá tải bởi số lượng trường hợp lớn như vậy, vì vậy cần sử dụng các thuật toán để tìm kiếm các nhóm "tốt", nhưng không nhất thiết phải là các nhóm "tối ưu nhất".

Tóm lại, mục tiêu cơ bản của phân tích cụm là tìm ra các nhóm tự nhiên của các đối tượng (hoặc biến số). Để làm được điều đó, trước tiên chúng ta phải phát triển một thang đo định lượng để đo lường sự liên kết (tương tự) giữa các đối tượng.

2 Thước đo sự tương tự

2.1 Lựa chọn thước đo tương tự

Để phân tích và phân cụm dữ liệu, chúng ta cần một phương pháp định lượng để đo lường mức độ tương đồng hoặc gần gũi giữa các đối tượng trong tập dữ liệu. Thước đo sự tương tự giúp xác định các đối tượng nào có đặc điểm gần giống nhau nhất, từ đó nhóm chúng lại thành các cụm có ý nghĩa. Tuy nhiên, việc chọn thước đo sự tương tự không phải lúc nào cũng khách quan và có thể phụ thuộc vào người thực hiện, đặc biệt là khi lựa chọn các biến số hoặc phương pháp đo phù hợp với bản chất dữ liệu. Những cân nhắc quan trọng bao gồm bản chất của các biến số (rời rạc, liên tục, nhị phân), thang đo (danh mục, thứ tự, khoảng cách, tỷ lệ), và kiến thức chuyên môn về lĩnh vực. Khi các mục (đơn vị hoặc trường hợp) được phân cụm, sự gần gũi thường được chỉ ra bằng một loại khoảng cách nào đó. Ngược lại, các biến số thường được nhóm lại dựa trên hệ số tương quan hoặc các thước đo liên kết tương tự.

2.2 Khoảng cách và hệ số tương tự cho cặp các mục

Công thức tính khoảng cách Euclidean giữa hai quan sát p-chiều $x' = [x_1, x_2, \ldots, x_p]$ và $y' = [y_1, y_2, \ldots, y_p]$ là:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_p - y_p)^2} \tag{1}$$

$$= \sqrt{\sum_{k=1}^{p} (x_k - y_k)^2} \tag{2}$$

Công thức tính "Statistical distace" (Tạm dịch là Khoảng cách thống kê) giữa 2 điểm:

$$d(x,y) = \sqrt{(x-y)' A (x-y)} \tag{3}$$

Thông thường, $A = S^{-1}$, trong đó S chứa phương sai và hiệp phương sai mẫu. Tuy nhiên, nếu không có kiến thức trước về các nhóm phân biệt, các giá trị này không thể tỉnh toán được. Vì lý do này, khoảng cách Euclidean thường được ưa chuộng cho việc phân cụm.

Một số cách tính khoảng cách khác

- **Khoảng cách Minkowski**

  Khi không có ý tưởng trước về kiến thức nhóm khoảng cách, chúng ta sử dụng công thức Minkowski. Với $x = (x_1, x_2, \ldots, x_m)$ và $y = (y_1, y_2, \ldots, y_m)$ là các đối tượng dữ liệu m-chiều và m là số nguyên dương.

  Công thức

  $$d(x,y) = \left[\sum_{k=1}^{p} |x_k - y_k|^m \right]^{\frac{1}{m}} \tag{4}$$

  Khi $m = 1$, nó tương đương với *city-block* (hay còn gọi là khoảng cách thành phố) và tính tổng các độ lệch tuyệt đối của các thành phần. Khi $m = 2$, nó tương đương với khoảng cách Euclidean. Khi $p = \infty$, nó tương đương với khoảng cách Chebyshev.

  Hai cách tính khoảng cách tiếp theo là *Canberra metric* và *Czekanowski coefficient*, cả hai công thức đều chỉ được áp dụng cho các biến không âm.

- **Khoảng cách Canbera metric**

  Cho hai điểm $x = (x_1, x_2, \ldots, x_p)$ và $y = (y_1, y_2, \ldots, y_p)$, khoảng cách Canberra được tính như sau:

  Công thức

$$d(x, y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{|x_i + y_i|} \tag{5}$$

  Trong đó x_i và y_i là giá trị của hai điểm trên chiều thứ i, p là số chiều của dữ liệu.

- **Hệ số Czekanowski**

$$S = \frac{2\sum_{i=1}^{p} \min(x_i, y_i)}{\sum_{i=1}^{p}(x_i + y_i)} \tag{6}$$

  Công thức khoảng cách dựa trên hệ số Czekanowski

$$d_(x, y) = 1 - \frac{2\sum_{i=1}^{p} \min(x_i, y_i)}{\sum_{i=1}^{p}(x_i + y_i)} \tag{7}$$

Khi có thể, chúng ta nên sử dụng các khoảng cách metric khỏa mãn ba tính chất sau:

1. Đối xứng: $d(P, Q) = d(Q, P)$ với mọi P, Q trong không gian.
2. Không âm: $d(P, Q) \geqslant 0$, dấu "=" xảy ra khi và chỉ khi $P \equiv Q$.
3. Bất đẳng thức tam giác: $d(P, Q) \leqslant d(P, R) + d(R, Q)$ với mọi P, Q, R trong không gian.

Khi các mục không thể được biểu diễn bằng các phép đo đa chiều có ý nghĩa, các cặp mục thường được so sánh dựa trên sự hiện diện hoặc vắng mặt của một số đặc tính nhất định. Các mục tương tự có nhiều đặc tính chung hơn so với các mục không tương tự. Sự hiện diện hay vắng mặt của một đặc tính có thể được mô tả bằng cách sử dụng một biến nhị phân, nhận giá trị 1 nếu đặc tính có mặt và giá trị 0 nếu đặc tính vắng mặt.

Ví dụ 1: Với $p = 5$, "điểm" cho 2 mục i và k có thể được sắp xếp như sau:

	Đặc trưng 1	Đặc trưng 2	Đặc trưng 3	Đặc trưng 4	Đặc trưng 5
Items i	1	0	0	1	1
Items k	1	1	0	1	0

Bảng 2: Ví dụ 1

Khi đó:

$$(x_{ij} - x_{kj})^2 = \begin{cases} 0 & \text{nếu } x_{ij} = x_{kj} = 1 \text{ hoặc } x_{ij} = x_{kj} = 0, \\ 1 & \text{nếu } x_{ij} \neq x_{kj}. \end{cases} \tag{8}$$

Khi đó khoảng cách Euclidean

$$\sum_{j=1}^{p}(x_{ij} - x_{kj})^2$$

sẽ cung cấp cho ta các cặp không khớp, chính là số các giá trị khác nhau giữa hai đối tượng i và k. Ví dụ bình phương khoảng cách giữa đối hai đối tượng i và k trong ví dụ trên là:

$$\sum_{j=1}^{5}(x_{ij} - x_{kj})^2 = (1-1)^2 + (0-1)^2 + (0-0)^2 + (1-1)^2 + (1-0)^2 = 2$$

Mặc dù công thức tính khoảng cách trên cũng có thể sử dụng để tính toán độ tương đồng, nhưng nó lại có nhược điểm là trọng số của các cặp 1-1, và 0-0 được tính như nhau. Trên thực tê, hầu hết trường hợp, một cặp 1-1 thường có ý nghĩa hay sức ảnh hưởng đến độ tương đồng hơn là một cặp 0-0. Ví dụ, trong một nhóm người, việc hai người cùng đọc một cuốn sách sách Hy Lạp sẽ có nhiều ý nghĩa hơn là cả hai đều không đọc nó. Bởi vậy, chúng ta cần giảm thiểu số cặp 0-0 hay thậm chí loại bỏ nó một cách hoàn toàn. Để chúng ta có thể làm được điều này, một vài cách định nghĩa độ tương đồng khác được gợi ý. Ta sẽ sắp xếp tần số của các kết quả trùng khớp và khác nhau các mục i và k dưới dạng một bảng dự phòng.

Item i	**Item k**		**Totals**
	1	0	
1	a	b	a + b
0	c	d	c + d
Totals	a + c	b + d	p = a + b + c + d

Bảng 3: Bảng dự phòng

Trong bảng trên, a biểu diễn số tần suất xuất hiện của cặp 1-1, b biểu diễn số tần suất xuất hiện của cặp 1-0, và tương tự cho c và d. Tương ứng với ví dụ 1 thì a = 2, b = c = d = 1. Dưới đây chúng tôi đưa ra bảng các hệ số tương tự thường gặp:

Coefficient	**Rationale**
1. $\frac{a+d}{p}$	Trọng số bằng nhau cho các cặp phù hợp 1-1 và 0-0.
2. $\frac{2(a+d)}{2(a+d)+b+c}$	Trọng số gấp đôi cho các cặp phù hợp 1-1 và 0-0.
3. $\frac{a+d}{a+d+2(b+c)}$	Trọng số gấp đôi cho các cặp không phù hợp.
4. $\frac{a}{p}$	Không có cặp 0-0 trong tử số.
5. $\frac{a}{a+b+c}$	Không có cặp 0-0 trong tử số hoặc mẫu số. (Các cặp 0-0 được coi là không liên quan.)
6. $\frac{2a}{2a+b+c}$	Không có cặp 0-0 trong tử số hoặc mẫu số. Không có cặp 0-0 trong tử số hoặc mẫu số.
7. $\frac{a}{a+2(b+c)}$	Không có cặp 0-0 trong tử số hoặc mẫu số. Trọng số gấp đôi cho các cặp không phù hợp.
8. $\frac{a}{b+c}$	Tỷ lệ số cặp phù hợp với số cặp không phù hợp, loại trừ các cặp 0-0.

Bảng 4: Hệ số tương tự dùng cho phân cụm các mục

Những hệ số (*coefficients*) ở hàng 1, 2 và 3 có tính đồng điệu với nhau. Giả sử hệ số thứ 1 được tính toán ở hai bảng thống kê, nếu hệ số thứ nhất ở bảng thống kê 1 hơn bảng thống kê 2 thì hệ số thứ 2 và 3 ở bảng thống kê thứ nhất cũng sẽ hơn hệ số thứ 2, 3 ở bảng thống kê thứ 2. Tương tự, hệ số thứ 5, 6, 7 cũng có tính chất như vậy.

Sự đơn điệu (*monotonicity*) trong việc tính toán độ tương đồng trong bài toán phân cụm đóng vai trò quan trọng vì nó đảm bảo tính nhất quán và đáng tin cậy của kết quả phân cụm.

Khi tính toán độ tương đồng giữa các mục trong bài toán phân cụm, sự đơn điệu đảm bảo rằng khi các mục tương đồng hơn theo một tiêu chí nào đó, thì độ tương đồng tính toán cũng tăng lên hoặc ít nhất là không giảm. Tương tự, khi các mục không tương đồng hơn, độ tương đồng cũng giảm hoặc ít nhất là không tăng.

Sự đơn điệu là một thuộc tính quan trọng trong các hàm đo độ tương đồng vì nó đảm bảo tính chất liên quan đến thứ tự và sự tương quan giữa các mục. Khi các hàm đo độ tương đồng là đơn điệu, ta có thể tin cậy sử dụng chúng để so sánh sự tương đồng giữa các mục và xác định các cụm hoặc nhóm mục tương tự.

Nếu một hàm đo độ tương đồng không đơn điệu, nghĩa là nó không tuân theo tính chất này, thì nó có thể dẫn đến kết quả không nhất quán và không đáng tin cậy trong quá trình phân cụm. Do đó, sự đơn điệu là một yêu cầu quan trọng để đảm bảo tính chính xác và đúng đắn của các phương pháp phân cụm.

Individual	Height	Weight	Eye color	Hair color	Handedness	Gender
Individual 1	68 in	140 lb	green	blond	right	female
Individual 2	72 in	185 lb	brown	brown	right	male
Individual 3	67 in	165 lb	blue	blond	right	male
Individual 4	64 in	120 lb	brown	brown	right	female
Individual 5	76 in	210 lb	brown	brown	left	male

Bảng 5: Ví dụ 2

Xác định 6 biến nhị phân X1,X2,X3,X4,X5,X6 như sau:

$$X_1 = \begin{cases} 1 & \text{height } \geqslant 72 \text{ in} \\ 0 & \text{height } < 72 \text{ in} \end{cases} \qquad X_4 = \begin{cases} 1 & \text{blond hair} \\ 0 & \text{not blond hair} \end{cases}$$

$$X_2 = \begin{cases} 1 & \text{weight } \geqslant 150 \text{ lb} \\ 0 & \text{weight } < 150 \text{ lb} \end{cases} \qquad X_5 = \begin{cases} 1 & \text{right handed} \\ 0 & \text{left handed} \end{cases}$$

$$X_3 = \begin{cases} 1 & \text{brown eyes} \\ 0 & \text{otherwise} \end{cases} \qquad X_6 = \begin{cases} 1 & \text{female} \\ 0 & \text{male} \end{cases}$$

Ta có bảng cho *Individual 1* và *Individual 2* với $p = 6$ như sau:

	X_1	X_2	X_3	X_4	X_5	X_6
Individual 1	0	0	0	1	1	1
Individual 2	1	1	1	0	1	0

Và số lượng các kết quả trùng khớp và khác nhau được chỉ ra trong bảng 2 chiều sau:

	Individual 2 = 1	Individual 2 = 0	Total
Individual 1 = 1	1	2	3
Individual 1 = 0	3	0	3
Totals	4	2	6

Sử dụng hệ số tương tự 1 từ bảng 1 ta có:

$$\text{Hệ số tương tự} = \frac{a+d}{p} = \frac{1+0}{6} = \frac{1}{6}$$

ta tính các hệ số còn lại cho các cặp. Ta được ma trận đối xứng:

Individual	1	2	3	4	5
1	1				
2	$\frac{1}{6}$	1			
3	$\frac{4}{6}$	$\frac{3}{6}$	1		
4	$\frac{4}{6}$	$\frac{3}{6}$	$\frac{2}{6}$	1	
5	0	$\frac{5}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	1

Nhìn vào bảng ta có thể thấy *individual* 5 và 2 có mức độ tương đồng lớn nhất, với giá trị $\frac{5}{6}$ và *individual* 5 và 1 có ít độ tương đồng nhất.

Chúng ta hoàn toàn có thể xây dựng độ tương tự từ khoảng cách, ví dụ chúng ta có thể dùng công thức:

$$s_{ik} = \frac{1}{1 + d_{ik}} \tag{9}$$

trong đó s_{ik} là độ tương đồng của items i và items k, có giá trị từ 0 đến 1.

Tuy nhiên, không phải lúc nào ta cũng có thể xây dựng công thức tính được khoảng cách dựa vào độ tương tự. Chúng ta chỉ có thể làm được chỉ khi ma trận độ tương đồng nữa xác định dương. Với điều kiện này, ta có thể xây dựng công thức tính khoảng cách:

$$\tilde{d}_{ik} = \sqrt{2(1 - \bar{s}_{ik})} \tag{10}$$

2.3 Đo lường sự tương đồng và liên kết cho các cặp biến

Các phép đo tương đồng giúp chúng ta đo lường mức độ tương đồng giữa hai biến, xác định mức độ tương quan giữa chúng dựa trên các thuộc tính và đặc điểm của dữ liệu. Bên cạnh đó, chúng ta cũng quan tâm đến việc đo lường mức độ liên kết giữa các biến. Các phép đo liên kết này giúp chúng ta đo lường mức độ tương quan và mối quan hệ giữa các cặp biến trong phân tích đa biến. Khi các biến được viết dưới dạng nhi phân, nó có thể được sắp xếp vào bảng dự phòng. Lần này, thay vì là các đối tượng (items), các biến (variables) mới chính là thứ quyết định đến việc phân nhóm hay các mục.

	Variable k		**Totals**
Variable i	1	0	
1	a	b	a + b
0	c	d	c + d
Totals	a + c	b + d	n = a + b + c + d

Bảng 6: Bảng dự phòng cho các biến nhị phân

Ví dụ: Variable i bằng 1 và Variable k bằng 0 ở b trong số n items đang xét. Công thức hệ số tương quan của sản phẩm được sử dụng trong trường hợp này để đo độ tương quan tuyến tính giữa hai biến nhị phân. Công thức này có thể được áp dụng trực tiếp vào bảng liên hệ để tính toán hệ số tương quan.

$$r = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \tag{11}$$

Trong đó:

- a, b, c, d là số lượng quan sát trong các ô tương ứng của bảng trên.
- $n = a + b + c + d$ là tổng số quan sát trong bảng.

Công thức này tính toán hệ số tương quan của sản phẩm r, trong đó giá trị của r nằm trong khoảng từ -1 đến 1. Một giá trị gần 1 cho thấy có mối quan hệ tuyến tính mạnh giữa hai biến nhị phân, trong khi một giá trị gần -1 cho thấy một mối quan hệ tuyến tính âm mạnh giữa hai biến. Một giá trị gần 0 cho thấy không có mối quan hệ tuyến tính đáng kể giữa hai biến nhị phân.

2.4 Kết luận về sự tương tự

Trong dữ liệu, bằng cách định lượng sự tương tự giữa các biến hoặc đối tượng, chúng ta có thể thu được thông tin quý giá về sự liên kết, nhóm hoặc mẫu. Các phép đo tương tự cho phép chúng ta so sánh và so sánh các biến dựa trên đặc điểm, thuộc tính hoặc giá trị của chúng. Chúng cung cấp một khung làm việc để đánh giá mức độ tương tự hoặc khác biệt giữa các biến, giúp chúng ta xác định các điểm tương tự hoặc những điểm chung giữa chúng.

Trong phân tích dữ liệu, các phép đo tương tự có thể được sử dụng trong nhiều lĩnh vực khác nhau, như gom cụm (*clustering*), phân loại (*classification*), hệ thống gợi ý (*recommendation systems*) và nhận dạng mẫu (*pattern recognition*). Chúng giúp chúng ta xác định các nhóm hoặc cụm tương tự, xác định mức độ tương quan hoặc tương tự của các mục, và khám phá các mẫu hoặc cấu trúc ẩn trong dữ liệu.

Hơn nữa, các phép đo tương tự đóng vai trò là nền tảng cho các kỹ thuật phân tích khác nhau, bao gồm phân tích khoảng cách, phân tích tương quan và giảm chiều dữ liệu. Chúng cung cấp cơ sở để tính toán khoảng cách, tương quan hoặc mối liên hệ giữa các biến, giúp chúng ta định lượng các mối quan hệ và sự phụ thuộc có trong dữ liệu.

Nhìn chung, các phép đo tương là công cụ mạnh trong bộ công cụ phân tích dữ liệu, cho phép chúng ta khám phá, hiểu và giải thích các tập dữ liệu phức tạp. Chúng cung cấp một khung số hóa để đánh giá sự tương đồng và có thể dẫn đến thông tin quý giá và kiến thức hữu ích trong nhiều ứng dụng khác nhau.

3 Phân cụm phân cấp

3.1 Giới thiệu về phân cụm phân cấp

Trong các lĩnh vực như sinh học, marketing, phân tích xã hội v.v., phương pháp phân cụm phân cấp (hierarchical clustering methods) được đưa ra để giải quyết nhiều vấn đề nơi mà dữ liệu không có sẵn nhãn và cấu trúc nhóm cần được phát hiện tự động. Tuy nhiên, việc kiểm tra tất cả các khả năng phân nhóm là một thách thức lớn, ngay cả với sự hỗ trợ của các máy tính mạnh mẽ nhất. Để khắc phục khó khăn này, các thuật toán phân cụm cấp bậc đã được phát triển nhằm tạo ra các cụm có ý nghĩa mà không cần phải xem xét toàn bộ các cấu hình có thể

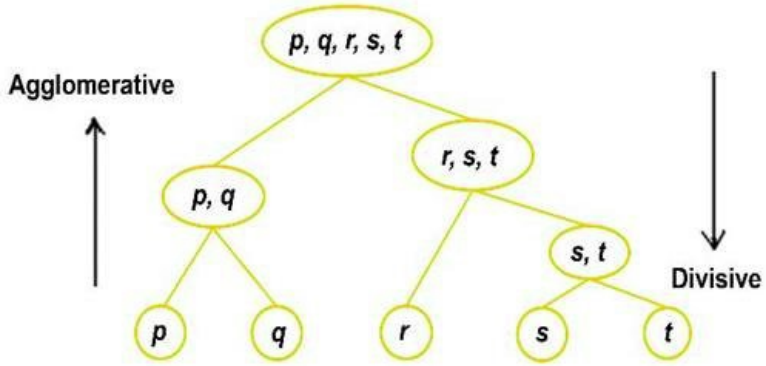
3.1.1 Các phương pháp chính

Phương pháp phân cụm hierarchical clustering được thực hiện bằng cách sử dụng một loạt các phép hợp nhất liên tiếp (Agglomerative) hoặc một loạt các phép chia liên tiếp (Divisive). Các phương pháp tổng hợp bắt đầu bằng việc coi mỗi đối tượng riêng lẻ là một cụm. Sau đó, các cụm giống nhau nhất được hợp nhất lại với nhau dựa trên độ tương tự của chúng. Quá trình này tiếp tục cho đến khi độ tương tự giữa các cụm đã hợp nhất giảm đi đáng kể, khi đó tất cả các cụm con cuối cùng được hợp nhất lại thành một cụm duy nhất.

Ngược lại, các phương pháp chia thành phần hoạt động theo hướng ngược lại. Ban đầu, một cụm đối tượng lớn được chia thành hai cụm con sao cho các đối tượng trong một cụm con "cách xa" các đối tượng trong cụm con kia. Quá trình chia tiếp tục cho đến khi mỗi đối tượng tạo thành một cụm duy nhất.

3.1.2 Biểu đồ dendrogram

Kết quả của cả hai phương pháp tổng hợp và chia thành phần có thể được biểu thị dưới dạng sơ đồ hai chiều được gọi là biểu đồ dendrogram. Biểu đồ dendrogram minh họa quá trình hợp nhất hoặc chia thành phần được thực hiện ở các cấp độ thứ bậc kế tiếp.



Hình 3.1: Biểu đồ dendrogram

3.1.3 Các phương pháp liên kết

Có nhiều thuật toán phân cụm hierarchical clustering khác nhau, và chúng thường được xác định bởi cách tính toán độ tương tự hoặc độ khác biệt giữa các đối tượng. Các phương pháp liên kết (linkage methods) là một phần quan trọng của phân cụm phân cấp, và chúng bao gồm liên kết đơn

(single linkage), liên kết hoàn chỉnh (complete linkage) và liên kết trung bình (average linkage). Mỗi phương pháp này sử dụng cách tính toán khác nhau để quyết định cách hợp nhất các cụm con.

3.1.4 Thuật toán chung

1. Bắt đầu với N cụm, mỗi cụm chứa một phần tử duy nhất và lập ma trận khoảng cách đối xứng NxN
2. Trên ma trận khoảng cách, tìm khoảng cách của các cặp gần nhất (có sự tương đồng nhau nhất). Giả sử khoảng cách giữa hai cụm gần nhất U và V là d_{UV} .
3. Hợp nhất cụm U và V . Gán nhãn cho cụm mới này là (UV). Cập nhập lại ma trận khoảng cách bằng cách:
   - Xóa các hàng và cột tương ứng với cụm U và V
   - Thêm một hàng và một cột gồm các khoảng cách giữ cụm (UV) và các cụm còn lại
4. Lặp lại 'Bước 3' cho đến khi chỉ còn lại một cụm duy nhất

3.2 Phương pháp hợp nhất (Agglomerative)

3.2.1 Phân cụm theo liên kết đơn

Đầu vào của phương pháp này là khoảng cách hoặc sự tương đồng giữa các cặp phần tử. Từ mỗi phần tử riêng biệt, thuật phân cụm phân cấp sẽ tạo ra các cụm lớn hơn bằng cách hợp nhất các cụm nhỏ hơn có khoảng cách nhỏ nhất hoặc độ tương đồng lớn nhất.

Từ ma trận khoảng cách đối xứng NxN là D = d_{ik} ta tìm khoảng cách nhỏ nhất trong D = d_{ik} và hợp nhất các phần tử tương ứng. Giả sử, khoảng cách giữa hai cụm gần nhất U và V là d_{UV}, gộp U với V để có được cụm (UV)

Đối với Bước 3 của thuật toán chung ở trên, khoảng cách giữa (UV) và bất kì cụm W nào khác được tính bằng công thức:

$$d_{(UV)W} = min\{d_{UW}, d_{VW}\}$$

Kết quả của phân cụm liên kết đơn có thể được hiển thị bằng đồ thị dưới dạng biểu đồ dendrogram hoặc sơ đồ cây. Các cành trên cây đại diện cho các cụm. Các nhánh kết hợp với nhau (hợp nhất) tại các nút có vị trí dọc theo trục khoảng cách (hoặc sự tương tự) cho biết mức độ hợp nhất xảy ra.

Ví dụ: Xét ma trận khoảng cách của năm đối tượng như sau:

$$D = \{d_{ik}\} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

Phân cụm thứ nhất

Bước 1: Ta nhận thấy giá trị nhỏ nhất của D hay 2 đối tượng gần nhất là:

$$min\{d_{ik}\} = d_{53} = 2$$

Ta nhóm 2 đối tượng 5 và 3 thánh một cụm (35)

Bước 2:Tính khoảng cách từ cụm (35) tới các đối tượng còn lại là 1, 2 và 4.

$$d_{(35)1} = min\{d_{31}, d_{51}\} = min\{(3, 11)\} = 3$$

$$d_{(35)2} = min\{d_{32}, d_{52}\} = min\{(7, 10)\} = 7$$

$$d_{(35)4} = min\{d_{34}, d_{54}\} = min\{(9, 8)\} = 8$$

Bước 3:Xóa đi các hàng và các cột tương ứng với phần tử thứ 3 và thứ 5, đồng thời thêm một hàng và cột cho cụm (35) ta thu được ma trận khoảng cách mới:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (35) \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} (35) & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

Phân cụm lần 2

Bước 1:Ta có khoảng cách ngắn nhất trong ma trận trên là d(35)1 = 3. Vậy ta ghép (1,3,5) thành cụm (135).

Bước 2: Từ cụm (135) ta tính khoảng cách tới các đối tượng còn lại là 2 và 4, ta có:

Bước 3:Xóa đi các hàng và các cột tương ứng với các chỉ số (35) và 1, sau đó thêm một hàng và cột cho cụm (135) ta được ma trận mới:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (135) \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} (135) & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 6 & 5 & 0 \end{bmatrix} \end{array}$$

Phân cụm cuối

Bước 1:Khoảng cách ngắn nhất trong ma trận trên là $d_{42} = 5$. Vậy ta ghép 2 và 4 thành cụm (24).

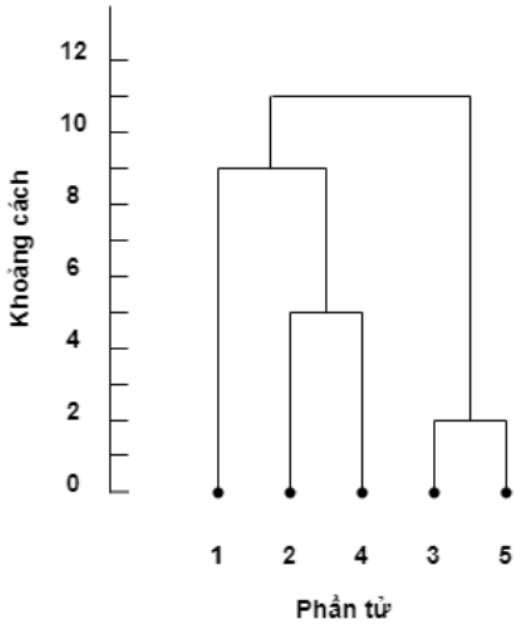
Bước 2:Ta thấy còn lại hai cụm (135) và (24), và

$$d_{\{(24)(135)\}} = min\{d_{\{2(135)\}}, d_{\{4(135)\}}\} = min\{6, 7\} = 6$$

Bước 3:Xóa đi các hàng và các cột tương ứng với các chỉ số 2 và 4, sau đó thêm một hàng và cột cho cụm (24) ta được ma trận mới:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (135) \\ (24) \end{array} \begin{array}{c} \begin{array}{cc} (135) & (24) \end{array} \\ \begin{bmatrix} 0 & \\ 6 & 0 \end{bmatrix} \end{array}$$

Cuối cùng, cụm (24) kết hợp với (135) thành một cụm duy nhất (12345), với khoảng cách gần nhất là 6



Hình 3.2: Biểu đồ 2 chiều của cách phân cụm theo liên kết đơn

3.2.2 Phân cụm theo liên kết hoàn chỉnh

Phương pháp phân cụm liên kết hoàn chỉnh được thực hiện tương tự phương pháp liên kết đơn. Tuy nhiên, cách tính khoảng cách (hoặc sự tương đồng) giữa các cụm lại được xác định bởi khoảng cách giữa hai phần tử, mỗi phần tử từ một cụm có khoảng cách xa nhất

Thay đổi công thức tính khoảng cách giữa các cụm ở Bước 3. Khoảng cách giữa (UV) và bất kỳ cụm W nào khác được tính bằng công thức:

$$d_{(UV)W} = max\{d_{UW}, d_{VW}\}$$

Ví dụ:

Xét ma trận khoảng cách của năm đối tượng như sau:

$$D = \{d_{ik}\} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

Phân cụm lần 1

Bước 1: Ta có 2 đối tượng gần nhất là:

$$min\{d_{ik}\} = d_{53} = 2$$

Bước 2:Tính khoảng cách từ cụm (35) tới các đối tượng còn lại là 1, 2 và 4.

$$d_{(35)1} = max\{d_{31}, d_{51}\} = max\{(3, 11)\} = 11$$

$$d_{(35)2} = max\{d_{32}, d_{52}\} = max\{(7, 10)\} = 10$$

$$d_{(35)4} = max\{d_{34}, d_{54}\} = max\{(9, 8)\} = 9$$

Bước 3: Xóa đi các hàng và các cột tương ứng với phần tử thứ 3 và thứ 5, đồng thời thêm một hàng và cột cho cụm (35) ta thu được ma trận khoảng cách mới:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (35) \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} (35) & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 11 & 0 & & \\ 10 & 9 & 0 & \\ 9 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

Phân cụm lần 2

Bước 1: Ta có khoảng cách ngắn nhất trong ma trận trên là $d_{24} = 3$.Vậy ta ghép (2,4) thành cụm (24).

Bước 2: từ cụm (24) ta tính khoảng cách tới các đối tượng còn lại là (35) và 1, ta có:

$$d_{(24)(35)} = max\{d_{2(35)}, d_{4(35),}\} = max\{11, 9\} = 11$$

$$d_{(24)1} = max\{d_{21}, d_{41}\} = max\{9, 6\} = 9$$

Bước 3: Xóa đi các hàng và các cột tương ứng với phần tử thứ 2 và thứ 4, đồng thời thêm một hàng và cột cho cụm (24) ta thu được ma trận khoảng cách mới:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (35) \\ (24) \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} (35) & (24) & 1 \end{array} \\ \begin{bmatrix} 0 & & \\ 10 & 0 & \\ 11 & 9 & 0 \end{bmatrix} \end{array}$$

Phân cụm lần cuối

Bước 1: Ta có khoảng cách ngắn nhất trong ma trận trên là $d_{(24)1} = 9$.Vậy ta ghép ((24),1) thành cụm (124).

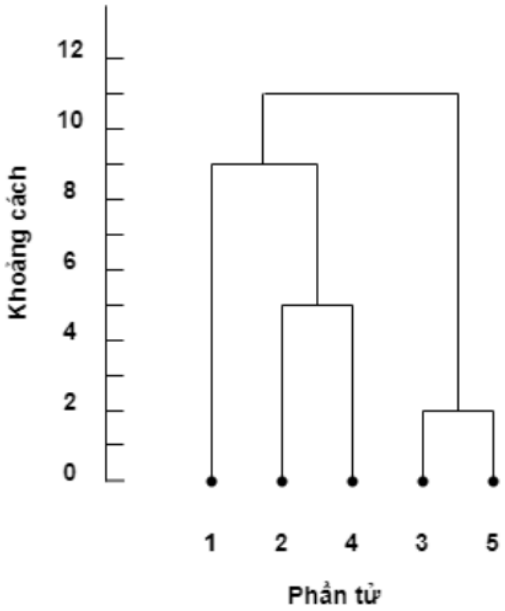
Bước 2: từ cụm (124) ta tính khoảng cách tới các đối tượng còn lại là (35), ta có:

$$d_{(124)(35)} = max\{d_{1(35)}, d_{(24)(35),}\} = max\{11, 10\} = 11$$

Bước 3: Xóa đi các hàng và các cột tương ứng với cụm (24) và chỉ số 4, sau đó thêm một hàng và cột cho cụm (124) ta được ma trận cuối cùng là:

$$D = \{d_{ik}\} = \begin{array}{c} \\ (35) \\ (124) \end{array} \begin{array}{c} \begin{array}{cc} (35) & (124) \end{array} \\ \begin{bmatrix} 0 & \\ 11 & 0 \end{bmatrix} \end{array}$$

Cuối cùng, cụm (124) kết hợp với (35) thành một cụm duy nhất (12345), với khoảng cách gần nhất là 11



Hình 3.3: Biểu đồ 2 chiều của cách phân cụm theo liên kết hoàn chỉnh

3.2.3 Phân cụm theo liên kết trung bình

Average linkage là phương pháp sử dụng khoảng cách trung bình giữa từng cặp điểm (một điểm từ cụm này và một điểm từ cụm kia)
Dầu vào của thuật toán có thể là khoảng cách hoặc hệ số tương đồng đều được. Thuật toán triển khai tương tự như các bước trong thuật toán gốc (12.12). Chúng ta vẫn bắt đầu với ma trận khoảng cách $D = \{d_{ik}\}$ và tìm cặp có khoảng cách gần nhất (giống nhau nhất) và hợp thành 1 cụm. Trong bước 3 của thuật toán thì khoảng cách của cụm UV tới các cụm W khác sẽ được xác định theo công thức:

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W}$$

Ví dụ:

Ta có ma trận khoảng cách đầu vào:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 2 & 4 & 6 \\ 2 & 0 & 5 & 7 \\ 4 & 5 & 0 & 8 \\ 6 & 7 & 8 & 0 \end{bmatrix} \end{array}$$

$d_{1,2}$ =2 là nhỏ nhất, ta ghép 2 cụm này lại.

$$\begin{array}{c} \\ (12) \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} (12) & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 4,5 & 6,5 \\ 4,5 & 0 & 8 \\ 6,5 & 8 & 0 \end{bmatrix} \end{array}$$

$d_{(1,2),3}$= 4,5 là nhỏ nhất, ta ghép 2 cụm này thành cụm 123.

$$\begin{array}{cc} & \begin{array}{cc}(123) & 4\end{array} \\ \begin{array}{c}(123)\\4\end{array} & \begin{bmatrix}0 & 7\\7 & 0\end{bmatrix}\end{array}$$

Chỉ còn cụm (123) và cụm 4 ta ghép lại thành 1 cụm.

Kết quả:
Quá trình phân cụm: [1],[2],[3],[4] -> [1,2],[3],[4] -> [1,2,3],[4]
-> [1,2,3,4] 1.4.3 Kết luận Đối với phân cụm liên kết trung bình, những thay đổi về cách tính khoảng cách(hay hệ số tương đồng) có thể ảnh hưởng đến cấu hình cuối cùng của cụm, ngay cả khi những thay đổi này vẫn giữ nguyên thứ tự tương đối.

3.3 Phân cụm theo phương pháp liên kết Ward

3.3.1 Thuật toán

Ward tiến hành quá trình phân cụm phân cấp dựa trên việc giảm thiểu tối đa "sự mất mát thông tin" khi ghép 2 cụm
Phương pháp này thường được dùng khi sự mất mát thông tin khi ghép cụm dẫn đến việc gia tăng của ESS. Ban đầu khi chưa ghép cụm thì mỗi điểm dữ liệu là 1 cụm do đó khoảng cách so với tâm là bằng 0 vì vậy, khởi đầu của thuật toán.

$$ESS = \sum_{j=1}^{n} ESS_j = \sum_{j=1}^{n} 0 = 0$$

Giá trị gia tăng của ESS khi ghép 2 cụm A và B lại thành cụm AB

$$\Delta ESS = \frac{N_A N_B}{N_A + N_B} \|\bar{x}_A - \bar{x}_B\|^2$$

3.3.2 Chứng minh công thức ΔESS

$\Delta ESS = \sum_{x_i \in A \cup B} \|x_i - m\|^2 - \sum_{x_i \in A} \|x_i - m_A\|^2 - \sum_{x_i \in B} \|x_i - m_B\|^2$

$\Delta ESS = \sum_{x_i \in A} \left[\|x_i - m\|^2 - \|x_i - m_A\|^2\right] + \sum_{x_i \in B} \left[\|x_i - m\|^2 - \|x_i - m_B\|^2\right]$

$\Delta ESS = \sum_{x_i \in A} (2x_i - m - m_A)^T (m_A - m) + \sum_{x_i \in B} (2x_i - m - m_B)^T (m_B - m)$

$\Delta ESS = (2N_A m_A - N_A m - N_A m_A)^T (m_A - m) + (2N_B m_B - N_B m - N_2 m_B)^T (m_B - m)$

$\Delta ESS = N_A (m_A - m)^T (m_A - m) + N_B (m_B - m)^T (m_B - m)$

$\Delta ESS = N_A \|m_A - \frac{N_A m_A + N_B m_B}{N_A + N_B}\|^2 + N_B \|m_B - \frac{N_A m_A + N_B m_B}{N_A + N_B} 1\|^2$

$\Delta ESS = N_A \|\frac{N_B (m_A - m_B)}{N_A + N_B}\|^2 + N_B \|\frac{N_A (m_B - m_A)}{N_A + N_B}\|^2$

$\Delta ESS = \frac{N_A N_B^2}{(N_A + N_B)^2} \|(m_A - m_B)\|^2 + \frac{N_B N_A^2}{(N_A + N_B)^2} \|(m_B - m_A)\|^2$

$\Delta ESS = \frac{N_A N_B}{N_A + N_B} \|(m_A - m_B)\|^2$

3.3.3 Kết quả cuối cùng

Giá trị cuối cùng của ESS khi tất cả các điểm được ghép lại thành 1 cụm duy nhất là

$$ESS = \sum_{j=1}^{n}(x_j - \bar{x})^T(x_j - \bar{x}) = \sum_{j=1}^{n}\|x_j - \bar{x}\|^2$$

Trong đó:
x_j : phép đo đa biến của phần tử thứ j
$\bar{x}$: giá trị trung bình của tất các các phần tử

Kết quả của phương pháp Ward có thể được biểu thị dưới dạng dendrogram với trục hoành là các phần tử và trục tung là giá trị ESS tại điểm ghép cụm.

Phương pháp Ward dựa trên khái niệm rằng các cụm quan sát đa biến được mong đợi có dạng elip gần đúng, đây là tiền thân của phân cụm không phân cấp.

3.4 Ứng dụng vào bài toán thực tế

3.4.1 Phương pháp liên kết đơn và hoàn chỉnh

Ví dụ 12.4

Language	English	Norwegian	Danish	Dutch	German	French	Spanish	Italian	Polish	Hungarian	Finish
English	0	2	2	7	6	6	6	6	7	9	9
Norwegian	2	0	1	5	4	6	6	6	7	8	9
Danish	2	1	0	6	5	6	5	5	6	8	9
Dutch	7	5	6	0	5	9	9	9	10	8	9
German	6	4	5	5	0	7	7	7	8	9	9
French	6	6	6	9	7	0	2	1	5	10	9
Spanish	6	6	5	9	7	2	0	1	3	10	9
Italian	6	6	5	9	7	1	1	0	4	10	9
Polish	7	7	6	10	8	5	3	4	0	10	9
Hungarian	9	8	8	8	9	10	10	10	10	0	8
Finish	9	9	9	9	9	9	9	9	9	8	0

Bảng 7: sự tương đồng của các ngôn ngữ trên thế giới

Thêm thư viện và đọc dữ liệu từ file csv đồng thời tạo DataFrame từ dữ liệu

```
import pandas as pd
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
###
data = pd.read_csv("D:/Slide/ptsl/data.csv", header=0, index_col=0)
print(data.shape)
data.head(11)
###
df = pd.DataFrame(data)
```

Từ các hàm có sẵn ta xây dựng biểu đồ dendrogram cho dữ liệu đang xét

```
# single linkage
plt.figure(figsize=(10, 5))
dendrogram (linkage(df, method='single'), labels=df.columns, orientation='top',
    distance_sort='descending', leaf_font_size=10)
```

```
plt.title('Dendrogram - Single Linkage')
plt.show()
# complete Linkage
plt.figure(figsize=(10, 5))
dendrogram (linkage (df, method='complete'), labels=df.columns, orientation='top',
    distance_sort='descending', leaf_font_size=10)
plt.title('Dendrogram - Complete Linkage')
plt.show()
```

Kết luận:

- English, mặc dù có sự khác biệt lớn, lại thể hiện sự ảnh hưởng từ các ngôn ngữ Bắc Âu, đặc biệt khi nó có sự tương đồng với Norwegian và Danish, phản ánh sự học hỏi và giao lưu giữa các cộng đồng ngôn ngữ.
- Spanish, French, và Italian có nhiều điểm chung do sự di cư và giao thoa giữa các cộng đồng nói tiếng Romance, điều này giải thích tại sao chúng lại có sự tương đồng đáng kể.
- Mặc dù có ít sự tương đồng với các ngôn ngữ khác, Hungarian và Finnish vẫn duy trì đặc điểm độc lập, cho thấy sự phát triển riêng biệt của các ngôn ngữ này trong các cộng đồng của chúng.

Kết luận: Các phương pháp phân nhóm như Single Linkage Clustering và Complete Linkage Clustering không chỉ giúp chúng ta phân tích mối quan hệ giữa các ngôn ngữ, mà còn mở ra cái nhìn sâu sắc về quá trình di cư, sự giao lưu học hỏi và ảnh hưởng văn hóa giữa các cộng đồng ngôn ngữ khác nhau

3.4.2 Phương pháp liên kết Ward

Import các thư viện cần thiết

```
import numpy as np
import pandas as pd
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster
import matplotlib.pyplot as plt
from sklearn.metrics import jaccard_score
from collections import Counter
```

Lấy dữ liệu từ Kaggle và bỏ các giá trị không cần thiết sau đó cho vào dataframe 'whisky4'
Bộ dữ liệu quan sát 109 loại rượu scotch nguyên chất từ 109 nhà chưng cất khác nhau và 68 biến nhị phân thể hiện những mô ta của người nếm về từng loại rượu

```
whisky4 = pd.read_csv("C:\\Users\\quanm\\Downloads\\scotch.csv")
name_column = whisky4['Unnamed: 0']
whisky4 = whisky4.loc[:, 'color':'FIN.18']
whisky4 = whisky4.iloc[:-2]
whisky4.columns = whisky4.iloc[0]
whisky4 = whisky4[1:].reset_index(drop=True)

name_column = name_column.iloc[1:-2]
name_column = name_column.reset_index(drop=True)

whisky4 = whisky4.apply(pd.to_numeric, errors='coerce')
whisky4['NAME'] = name_column
```

```
whisky4
```

Thực hiện phân cụm phân cấp với thuật toán liên kết Ward với khoảng cách được tính theo hệ số tương đồng Jaccad(S) và khoảng cách giữa 2 cụm được tính theo công thức $d_{i,k} = 1$ - $S_{i,k}$, ma trận khoảng cách sau các lần lặp được cập nhật đệ quy theo thuật toán Lance William

```
num_clt = 12
data = whisky4.drop(columns='NAME').to_numpy()

n_samples = data.shape[0]
jaccard_matrix = np.zeros((n_samples, n_samples))

for i in range(n_samples):
    for j in range(i + 1, n_samples):
        jaccard = jaccard_score(data[i], data[j])
        jaccard_matrix[i, j] = jaccard_matrix[j, i] = jaccard

jaccard_array = jaccard_matrix[np.triu_indices(n_samples, k=1)]

jaccard_array = 1 - jaccard_array

Z = linkage(jaccard_array, method='ward')

plt.figure(figsize=(10, 7))
dendrogram(Z)
plt.show()
clusters = fcluster(Z, t=num_clt, criterion='maxclust')

for i in range(1, num_clt + 1):
    cluster_points = np.where(clusters == i)[0]
    cluster_names = whisky4['NAME'].iloc[cluster_points]
    print(f"Cum {i}: {list(cluster_names)}")
```

Chúng ta cắt ở 12 cụm phân loại rượu và được kết quả như sau:

Cụm 1: ['Glen Albyn', 'Glengoyne', 'Glen Grant', 'Glenlossie', 'Linkwood', 'North Port', 'Saint Magdalene', 'Tamdhu']

Cụm 2: ['Benriach', 'Dailuaine', 'Glen Deveron', 'Glendronach', 'Glenesk', 'Glenkinchie', 'Glen Scotia', 'Inchgower', 'Jura', 'Springbank']

Cụm 3: ['Aberfeldy', 'Glenugie', 'Laphroaig', 'Scapa']

Cụm 4: ['Ardberg', 'Bowmore', 'Dufftown', 'Glenlochy', 'Lagavulin', 'Springbank-Longrow'] Cụm 5: ['Brackla', 'Convalmore', 'Dallas Dhu', 'Glen Keith', 'Glen Ordie', 'Glenrothes', 'Mortlach', 'Tormore']

Cụm 6: ['Banff', 'Benromach', 'Caperdonich', 'Cardhu', 'Craigellachie', 'Dalwhinnie', 'Glencadam', 'Glen Elgin', 'Glen Garioch', 'Glenury Royal', 'Imperial', 'Knockando', 'Knockdhu', 'Lochnagar',

'Miltonduff', 'Teaninich', 'Tomatin']

Cụm 7: ['Balmenach', 'Bruichladdich', 'Bunnahabhain', 'Cragganmore', 'Glenglassaugh', 'Glen Moray', 'Longmorn', 'Rosebank', 'Tamnavulin', 'Tomintoul', 'Tullibardine']

Cụm 8: ['Auchentoshan', 'Ben Nevis', 'Coleburn', 'Speyburn']

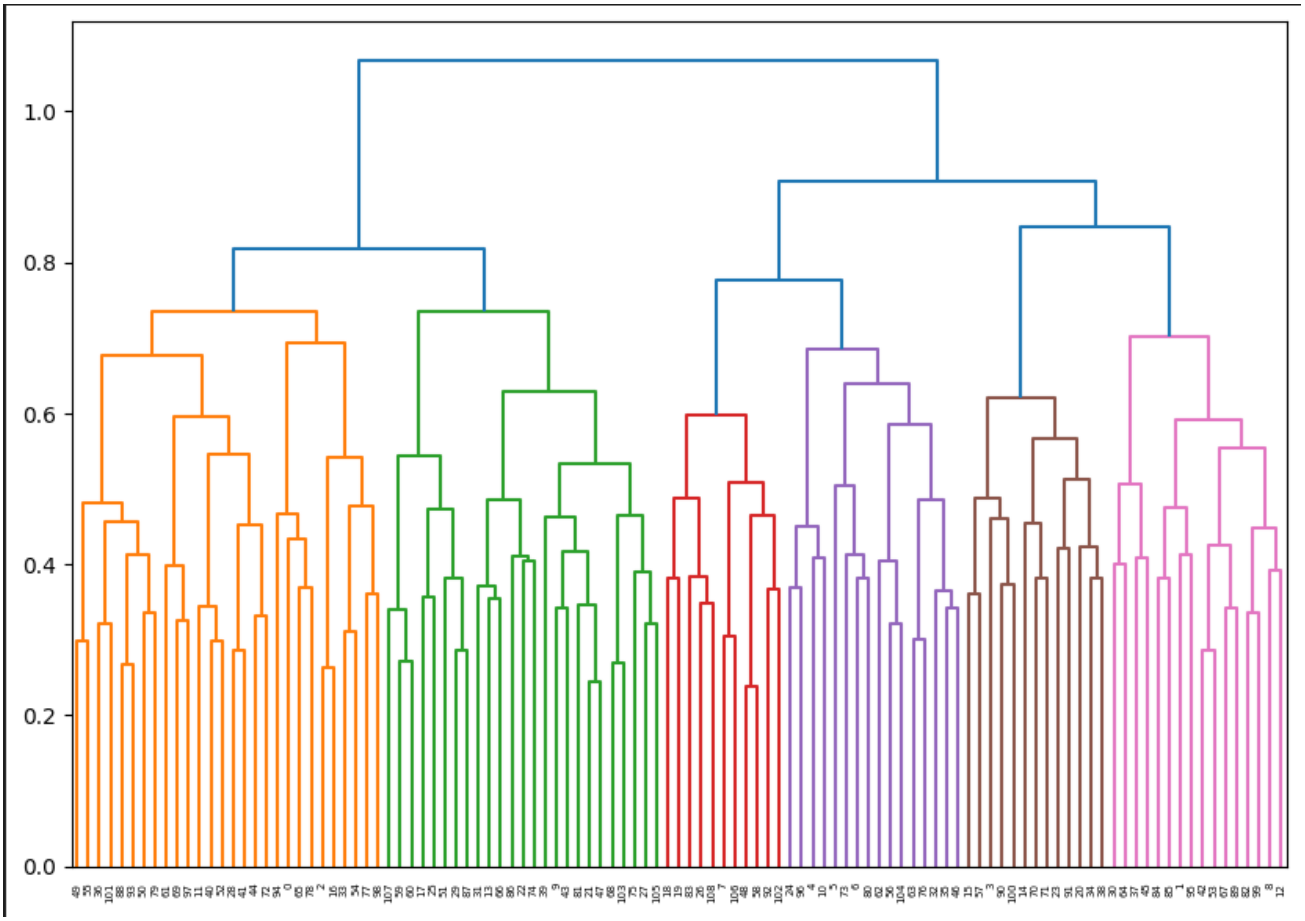
Cụm 9: ['Aultmore', 'Balblair', 'Deanston', 'Fettercairn', 'Glenfiddich', 'Glen Mhor', 'Glen Spey', 'Glentauchers', 'Kinclaith', 'Ladyburn', 'Littlemill', 'Tobermory']

Cụm 10: ['Ardmore', 'Bladnoch', 'Blair Athol', 'Caol Ila', 'Clynelish', 'Edradour', 'Glenburgie', 'Glenmorangie', 'Inchmurrin', 'Inverleven', 'Port Ellen', 'Pulteney', 'Talisker']

Cụm 11: ['Dalmore', 'Glenallachie', 'Glenfarclas', 'Glenturret']

Cụm 12: ['Aberlour', 'Balvenie', 'Benrinnes', 'Glendullan', 'Glenlivet', 'Highland Park', 'Lochside', 'Macallan', 'Millburn', 'Oban', 'Singleton', 'Strathisla']

Dendrogram thu được:



Từ kết quả của phần phân cụm này chúng ta có thể đưa ra đề suất cho khách hàng các loại rượu phù hợp với khẩu vị và sở thích của khách hàng

3.4.3 Kết luận cuối cùng

Mặc dù có nhiều cách tiếp cận khác nhau, tất cả các phương pháp phân cụm phân cấp theo lược đồ cơ bản như đã mô tả trong thuật toán (12-12). Điều này cho thấy sự thống nhất về nguyên lý, dù chi tiết triển khai có thể khác biệt.

Phương pháp phân cụm phân cấp không xét đến nguồn sai số hoặc biến động một cách chính thức. Vì vậy, các phương pháp này nhạy cảm với các điểm ngoại lai (outliers) hoặc các điểm "nhiễu". Điều này có thể làm sai lệch kết quả phân cụm nếu không được xử lý cẩn thận.

Một khi các đối tượng bị gộp nhóm sai ở giai đoạn đầu, không có cách nào để điều chỉnh lại chúng trong phân cụm phân cấp. Do đó, kết quả cuối cùng cần được xem xét kỹ lưỡng để đánh giá tính hợp lý.

Với mỗi vấn đề cụ thể, người dùng nên thử áp dụng nhiều phương pháp phân cụm khác nhau, cũng như các cách đo khoảng cách hoặc độ tương đồng khác nhau. Nếu các kết quả thu được tương đối nhất quán, điều này có thể giúp xác định được các nhóm tự nhiên (natural groupings) trong dữ liệu.

Độ ổn định có thể được kiểm tra bằng cách áp dụng thuật toán phân cụm trước và sau khi thêm các sai số nhỏ (nhiễu) vào dữ liệu. Nếu các nhóm phân cụm vẫn được duy trì, điều này cho thấy kết quả có độ ổn định cao.

Giá trị trùng lặp trong ma trận khoảng cách hoặc độ tương đồng có thể dẫn đến nhiều kết quả khác nhau. Điều này đặc biệt dễ xảy ra ở các mức phân cụm thấp hơn. Tuy nhiên, đây không phải là lỗi của thuật toán mà là do bản chất của dữ liệu. Người dùng cần nhận thức về các giải pháp này để so sánh và đánh giá mức độ trùng khớp giữa các nhóm.

Có trường hợp xảy ra việc lần ghép cụm sau có khoảng cách ghép cụm nhỏ hơn lần ghép cụm trước, nhưng điều này thường liên quan đến thuật toán centroid linkage và median linkage, những phương pháp vừa được giới thiệu thì khỗng dễ xảy ra trường hợp đảo ngược

4 Phân cụm không phân cấp

4.1 Giới thiệu

4.1.1 Giới thiệu phương pháp phân cụm không phân cấp

Kỹ thuật phân cụm không phân cấp là một phương pháp phân tích dữ liệu trong đó các đối tượng được nhóm lại với nhau dựa trên sự tương đồng của chúng mà không tạo ra một cấu trúc phân cấp như trong phân cụm phân cấp. Thay vào đó, nó phân chia dữ liệu thành một số lượng cụm cố định ngay từ đầu.

- Các kỹ thuật phân cụm không phân cấp được thiết kế để nhóm các đối tượng, thay vì các biến, vào một tập hợp gồm K cụm.
- Số lượng cụm K có thể được xác định trước hoặc xác định trong quá trình phân cụm.
- Có thể được áp dụng cho các bộ dữ liệu lớn hơn nhiều so với các kỹ thuật phân cấp.
- K-Means là một phương pháp nổi tiếng nằm trong số các phương pháp phân cụm không phân cấp.

4.1.2 Giới thiệu phương pháp K-means

K-Means là một phương pháp phân cụm đơn giản thuộc loại học không giám sát và được sử dụng để giải quyết bài toán phân cụm.

- Ý tưởng chính của K-Means là phân chia một bộ dữ liệu thành K cụm, K là số lượng cụm biết trước.
- Các điểm dữ liệu trong cùng 1 cụm thì phải có những tính chất tương đồng nhất định.
- Mục tiêu của thuật toán K-Means là với đầu vào là bộ dữ liệu ban đầu và số lượng cụm K, tìm ra trung tâm mỗi cụm và phân các điểm dữ liệu vào cụm tương ứng (Mỗi điểm chỉ thuộc 1 cụm duy nhất).

Ví dụ: Chia nhỏ tệp khách hành thành các nhóm đối tượng khác nhau (để có thể áp dụng các chiến lược kinh doanh cụ thể cho từng nhóm đối tượng như ưu đãi).

4.1.3 Lịch sử hình thành K-means

Mac Queen đưa ra phương pháp K-Means để mô tả thuật toán của ông là phân phối mỗi đối tượng vào cụm có trung tâm gần nó nhất. Quá trình đó được tạo qua 3 bước.

- Phân chia ngẫu nhiên các đối tượng vào K cụm ban đầu.
- Từ toàn bộ danh sách các đối tượng, phân phối từng đối tượng cho cụm có trung tâm gần nó nhất (theo công thức khoảng cách Euclide). Tính toán lại trung tâm cho cụm nhận được đối tượng mới và cụm mất đối tượng
- Lặp lại bước 2 cho đến khi không có sự phân phối lại giữa các cụm

4.2 Tóm tắt thuật toán

- **Đầu vào:** Dữ liệu $\mathbf{X}$ và số cụm K
- **Đầu ra:** Các trung tâm $\mathbf{M}$ và vector nhãn cho từng điểm dữ liệu $\mathbf{Y}$

① Chọn K điểm bất kỳ làm các trung tâm ban đầu

② Phân mỗi điểm dữ liệu vào cụm có trung tâm gần nhất

③ Nếu không có sự thay đổi trong việc gán cụm, dừng thuật toán. (Các điểm trong các cụm không đổi).

④ Cập nhật trung tâm cho từng cụm bằng cách lấy trung bình cộng các điểm trong cụm.

⑤ Quay lại bước 2

Source code thuật toán: `https://github.com/thethien8a/PTSL/blob/main/K-means.ipynb`

4.3 Phân tích toán học phương pháp K-means

Giả sử có N điểm dữ liệu là $X = [X_1, X_2, ..., X_N] \in \mathbb{R}^{d \times N}$ và $K < N$ là số cụm chúng ta muốn phân chia. Chúng ta cần tìm các trung tâm $m_1, m_2, ..., m_K \in \mathbb{R}^d$ và nhãn của mỗi điểm dữ liệu. Với mỗi điểm dữ liệu X_i, đặt $Y_i = [y_{i1}, y_{i2}, ..., y_{iK}]$ là vector nhãn của nó, trong đó nếu X_i được phân vào cụm k thì $y_{ik} = 1$ và $y_{ij} = 0, \forall j \neq k$.
Ràng buộc của Y_i có thể viết dưới dạng toán học như sau:

$$y_{ik} \in \{0, 1\}, \sum_{k=1}^{K} y_{ik} = 1$$

4.3.1 Hàm mất mát và bài toán tối ưu

Nếu ta coi $\mathbf{m}_k$ là trung tâm của mỗi cụm và ước lượng tất cả các điểm được phân vào cụm này bởi $\mathbf{m}_k$, thì một điểm dữ liệu $\mathbf{x}_i$ được phân vào cụm k sẽ có sai số là $|\mathbf{x}_i - \mathbf{m}_k|$. Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ta sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:

$$\|\mathbf{x}_i - \mathbf{m}_k\|_2^2$$

Vì x_i được phân vào cụm k nên $y_{ik} = 1$ và $y_{ij} = 0, \forall j \neq k$. Khi đó, biểu thức bên trên sẽ được viết lại là:

$$y_{ik}\|(x_i - m_k)\|_2^2 = \sum_{j=1}^{K} y_{ij}\|(x_i - m_j)\|_2^2$$

Sai số cho toàn bộ dữ liệu sẽ là:

$$\mathcal{L}(\mathbf{Y}, \mathbf{M}) = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Trong đó:

- $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_N]$ là ma trận được tạo bởi các vector nhãn của mỗi điểm dữ liệu.
- $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K]$ là ma trận được tạo bởi các trung tâm của mỗi cụm.

Hàm số mất mát trong bài toán K-means của chúng ta là hàm $\mathcal{L}(\mathbf{Y}, \mathbf{M})$ với ràng buộc như được nêu trong phương trình (1). Tóm lại chúng ta cần tối ưu bài toán sau:

$$\mathbf{Y}, \mathbf{M} = \arg\min_{\mathbf{Y},\mathbf{M}} \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2 \quad (2)$$

thỏa mãn điều kiện: $y_{ij} \in \{0, 1\}, \forall i, j; \quad \sum_{j=1}^{K} y_{ij} = 1$

4.3.2 Thuật toán tối ưu hàm mất mát

(i) Cố định M, tìm Y

Giả sử đã tìm được các trung tâm, hãy tìm các vector nhãn để hàm mất mát đạt giá trị nhỏ nhất. Khi các trung tâm là cố định, bài toán tìm vector nhãn cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm vector nhãn cho từng điểm dữ liệu $\mathbf{x}_i$ như sau:

$$\mathbf{y}_i = \arg\min_{\mathbf{y}_i} \sum_{j=1}^{K} y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 \quad (3)$$

thỏa mãn điều kiện: $y_{ij} \in \{0, 1\}, \forall i, j; \quad \sum_{j=1}^{K} y_{ij} = 1 Vì chỉ có một phần tử của vector nhãn $\mathbf{y}_i$ bằng 1$ nên bài toán (3) có thể tiếp tục được viết dưới dạng đơn giản hơn:

$$j = \arg\min_{j} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2.$$

Vì $\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$ chính là bình phương khoảng cách tính từ điểm $\mathbf{x}_i$ tới trung tâm $\mathbf{m}_j$, ta có thể kết luận rằng **mỗi điểm $\mathbf{x}_i$ thuộc vào cụm có trung tâm gần nó nhất!** Từ đó ta có thể dễ dàng suy ra vector nhãn của từng điểm dữ liệu.

(ii) Cố định Y, tìm M

Giả sử đã tìm được cụm cho từng điểm, hãy tìm trung tâm mới cho mỗi cụm để hàm mất mát đạt giá trị nhỏ nhất. Một khi chúng ta đã xác định được vector nhãn cho từng điểm dữ liệu, bài toán tìm trung tâm cho mỗi cụm được rút gọn thành:

$$\mathbf{m}_j = \arg\min_{\mathbf{m}_j} \sum_{i=1}^{N} y_{ij} ||\mathbf{x}_i - \mathbf{m}_j||^2.$$

Ta tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm. Hàm mất mát là hàm lồi theo $\mathbf{m}_j$ nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng.

Đặt $l(\mathbf{m}_j)$ là hàm bên trong dấu arg min, ta có đạo hàm:

$$\frac{\partial l(\mathbf{m}_j)}{\partial \mathbf{m}_j} = 2 \sum_{i=1}^{N} y_{ij} (\mathbf{m}_j - \mathbf{x}_i)$$

Giải phương trình đạo hàm bằng 0 ta có:

$$\mathbf{m}_j \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} y_{ij} \mathbf{x}_i$$

$$\Rightarrow \mathbf{m}_j = \frac{\sum_{i=1}^{N} y_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} y_{ij}}$$

$\Rightarrow \mathbf{m}_j$ chính là trung bình cộng của các điểm trong cụm j.

4.4 Ví dụ về phân cụm k-means

Bài toán : Giả sử ta có tập dữ liệu trong không gian 2 chiều:

$$X = \{(1,1), (1.5,1.5), (1,1.25), (10,11), (11,11), (12,12)\}$$

Giả sử theo kinh nghiệm thực tế, ta biết được có $K = 2$ cụm.
Thuật toán hoạt động như sau:
B1: Chọn ngẫu nhiên 2 tâm cụm trong tập điểm dữ liệu X (2 tâm cụm phải khác nhau, không được trùng nhau). Giả sử rằng ta chọn:

- $m_1 = (1, 1.25)$
- $m_2 = (12, 12)$

VÒNG LẶP $i = 1$
B2: Tính khoảng cách Euclid bình phương từ từng điểm dữ liệu trong X đến các cụm

Điểm dữ liệu	$m_1 = (1, 1.25)$	$m_2 = (12, 12)$
$(1, 1)$	$1/16$	242
$(1.5, 1.5)$	$5/16$	220.5
$(1, 1.25)$	0	236.5625
$(10, 11)$	$2817/16$	5
$(11, 11)$	$3121/16$	2
$(12, 12)$	$3785/6$	0

B3: Xác định xem khoảng cách từ điểm đó đến cụm nào đó là ngắn nhất thì điểm đó sẽ thuộc về cụm ấy.

Điểm dữ liệu	$m_1 = (1, 1.25)$	$m_2 = (12, 12)$	**KẾT LUẬN**
$(1, 1)$	$1/16$	242	$\in$ Cụm m1
$(1.5, 1.5)$	$5/16$	220.5	$\in$ Cụm m1
$(1, 1.25)$	0	236.5625	$\in$ Cụm m1
$(10, 11)$	$2817/16$	5	$\in$ Cụm m2
$(11, 11)$	$3121/16$	2	$\in$ Cụm m2
$(12, 12)$	$3785/6$	0	$\in$ Cụm m2

B4: Cập nhật tâm cụm cho m_1, m_2 bằng cách tính trung bình cộng của các điểm dữ liệu thuộc từng cụm

Ta đang có: $(1, 1), (1.5, 1.5), (1, 1.25)$ thuộc cụm m_1

$$\Rightarrow \text{Cập nhật } m_1 = \left(\frac{1 + 1.5 + 1}{3}, \frac{1 + 1.5 + 1.25}{3}\right) = \left(\frac{7}{6}, \frac{5}{4}\right)$$

Ta đang có: $(10, 11), (11, 11), (12, 12)$ thuộc cụm m_2

$$\Rightarrow \text{Cập nhật } m_2 = \left(\frac{10 + 11 + 12}{3}, \frac{10 + 11 + 12}{3}\right) = (11, 11)$$

B5: Kiểm tra điều kiện dừng:

- Ta thấy m_1 trước đó $m_{1\_trc} = (1, 1.25)$ khác $m_{1\_sau} = \left(\frac{7}{6}, \frac{5}{4}\right)$
- Ta thấy m_2 trước đó $m_{2\_trc} = (12, 12)$ khác $m_{2\_sau} = (11, 11)$

VÒNG LẶP i = 2

B6: Tính khoảng cách Euclide bình phương từ từng điểm dữ liệu trong X đến các cụm.

Điểm dữ liệu	$m_1 = \left(\frac{7}{6}, \frac{5}{4}\right)$	$m_2 = (11, 11)$
(1, 1)	$\frac{13}{144}$	200
(1.5, 1.5)	$\frac{25}{144}$	$\frac{361}{2}$
(1, 1.25)	$\frac{1}{36}$	$\frac{3121}{16}$
(10, 11)	$\frac{24925}{144}$	1
(11, 11)	191.756...	0
(12, 12)	232.923...	2

B7: Xác định xem khoảng cách từ điểm đó đến cụm nào đó là ngắn nhất thì điểm đó sẽ thuộc về cụm ấy.

Điểm dữ liệu	$m_1 = \left(\frac{7}{6}, \frac{5}{4}\right)$	$m_2 = (11, 11)$	**KẾT LUẬN**
(1, 1)	$\frac{13}{144}$	200	$\in$ Cụm m1
(1.5, 1.5)	$\frac{25}{144}$	$\frac{361}{2}$	$\in$ Cụm m1
(1, 1.25)	$\frac{1}{36}$	$\frac{3121}{16}$	$\in$ Cụm m1
(10, 11)	$\frac{24925}{144}$	1	$\in$ Cụm m2
(11, 11)	191.756...	0	$\in$ Cụm m2
(12, 12)	232.923...	2	$\in$ Cụm m2

B8: Cập nhật tâm cụm cho m1, m2 bằng cách tính trung bình cộng của các điểm dữ liệu thuộc từng cụm Ta đang có: $(1, 1)$, $(1.5, 1.5)$, $(1, 1.25)$ thuộc cụm m1

$$\Rightarrow \text{Cập nhật } m1 = \left(\frac{1 + 1.5 + 1}{3}, \frac{1 + 1.5 + 1.25}{3}\right) = \left(\frac{7}{6}, -\frac{5}{4}\right)$$

Ta đang có: $(10, 11)$, $(11, 11)$, $(12, 12)$ thuộc cụm m2

$$\Rightarrow \text{Cập nhật } m2 = \left(\frac{10 + 11 + 12}{3}, \frac{10 + 11 + 12}{3}\right) = (11, 11)$$

B9: Kiểm tra điều kiện dừng:

Ta thấy $m1$ trước đó $m1_{\text{trước}} = \left(\frac{7}{6}, -\frac{5}{4}\right)$ bằng với $m1_{\text{sau}} = \left(\frac{7}{6}, -\frac{5}{4}\right)$

Ta thấy $m2$ trước đó $m2_{\text{trước}} = (11, 11)$ bằng với $m2_{\text{sau}} = (11, 11)$

⇒ **DỪNG VÒNG LẶP**

KẾT LUẬN:

VẬY VỚI TẬP DỮ LIỆU

$$X = \{(1,1), (1.5,1.5), (1,1.25), (10,11), (11,11), (12,12)\}$$

Ta xác định được 2 tâm cụm ở vị trí:

- $m_1 = \left(\frac{7}{6}, -\frac{5}{4}\right)$ gồm các điểm $(1,1)$, $(1.5,1.5)$, $(1,1.25)$
- $m_2 = (11,11)$ gồm các điểm $(10,11)$, $(11,11)$, $(12,12)$

4.5 Bàn luận về sự hội tụ của thuật toán

4.5.1 Bài toán đặt ra

Trong thuật toán phân cụm K-means, bạn được cung cấp một tập hợp gồm n điểm $x_i \in \mathbb{R}^d, i \in \{1,\ldots,n\}$ và cần tìm tâm của k cụm $\mu = (\mu_1,\ldots,\mu_k)$ bằng cách tối thiểu hóa khoảng cách trung bình từ các điểm đến tâm cụm gần nhất. Cụ thể, bạn cần tối thiểu hóa hàm mất mát sau:

$$L(\mu) = \sum_{i=1}^{n} \min_{j\in\{1,\ldots,k\}} \|x_i - \mu_j\|^2.$$

Để gần đúng nghiệm, chúng ta giới thiệu các biến gán mới $z_i = \arg\min_{j\in\{1,\ldots,k\}} \|x_i - \mu_j\|^2$ cho mỗi điểm dữ liệu x_i. Thuật toán K-means lặp lại giữa việc cập nhật các biến z_i (bước gán nhãn - *assignment step*) và cập nhật các tâm cụm $\mu_j = \frac{1}{|\{i:z_i=j\}|}\sum_{i:z_i=j} x_i$ (bước hiệu chỉnh - *refitting step*). Thuật toán dừng khi không có thay đổi nào xảy ra trong bước gán nhãn.

Chứng minh rằng thuật toán K-means được đảm bảo hội tụ (đến một điểm cực tiểu cục bộ).

4.5.2 Chứng minh

Để chứng minh sự hội tụ của thuật toán K-means, chúng ta cần chỉ ra rằng hàm mất mát được đảm bảo giảm đơn điệu trong mỗi lần lặp cho cả bước gán nhãn (*assignment step*) và bước hiệu chỉnh (*refitting step*). Vì hàm mất mát không âm, thuật toán cuối cùng sẽ hội tụ khi hàm mất mát đạt đến cực tiểu (cục bộ).

Gọi $\mathbf{z} = (z_1,\ldots,z_n)$ là các nhãn cụm cho n điểm.

(i) Bước gán nhãn (*Assignment step*)

Chúng ta có thể viết hàm mất mát ban đầu $L(\mu, z)$ như sau:

$$L(\mu, z) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|_2^2$$

Xét một điểm dữ liệu x_i, và gọi z_i là nhãn từ lần lặp trước đó và z_i^* là nhãn mới được gán, được xác định như sau:

$$z_i^* = \arg\min_{j\in\{1,\ldots,k\}} \|x_i - \mu_j\|_2^2$$

Gọi $\mathbf{z}^*$ là các nhãn cụm mới cho tất cả n điểm. Sự thay đổi trong hàm mất mát sau bước gán nhãn được cho bởi:

$$L(\mu, \mathbf{z}^*) - L(\mu, \mathbf{z}) = \sum_{i=1}^{n} \left(\|x_i - \mu_{z_i^*}\|_2^2 - \|x_i - \mu_{z_i}\|_2^2 \right) \leqslant 0$$

Bất đẳng thức này đúng vì quy tắc xác định z_i^*, tức là gán x_i cho cụm gần nhất.

(ii) Bước hiệu chỉnh (*Refitting step*)

Chúng ta có thể viết hàm mất mát ban đầu $L(\mu, z)$ như sau:

$$L(\mu, z) = \sum_{j=1}^{k} \left(\sum_{i:z_i=j} \|x_i - \mu_j\|_2^2 \right)$$

Hãy xét cụm thứ j và gọi μ_j là tâm cụm từ lần lặp trước đó, μ_j^* là tâm cụm mới được tính như sau:

$$\mu_j^* = \frac{1}{|\{i : z_i = j\}|} \sum_{i:z_i=j} x_i$$

Gọi μ^* là các tâm cụm mới cho tất cả k cụm. Sự thay đổi trong hàm mất mát sau bước hiệu chỉnh này được tính như sau:

$$L(\mu^*, z) - L(\mu, z) = \sum_{j=1}^{k} \left(\left(\sum_{i:z_i=j} \|x_i - \mu_j^*\|_2^2 \right) - \left(\sum_{i:z_i=j} \|x_i - \mu_j\|_2^2 \right) \right) \leqslant 0$$

Bất đẳng thức này đúng vì quy tắc cập nhật μ_j^* thực chất đã tối thiểu hóa đại lượng này.

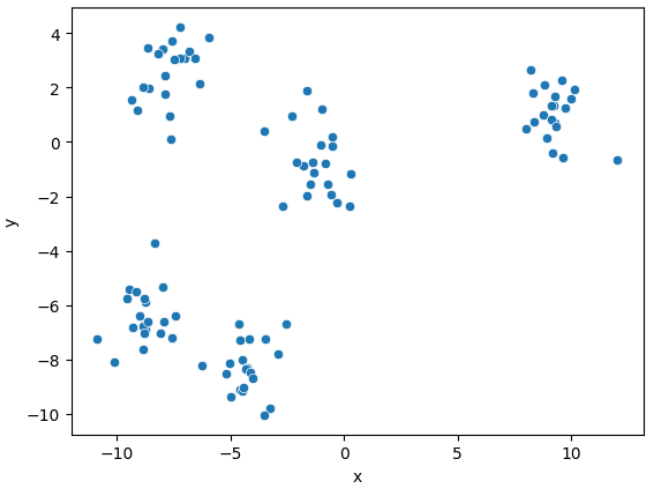
4.6 Thuật toán cải tiến của K-Means: K-means++

4.6.1 Mặt hạn chế của K-Means

Một nhược điểm của thuật toán K-mean là nó nhạy cảm với việc khởi tạo các trọng tâm hoặc các điểm trung bình. Vì vậy, nếu một centroid được khởi tạo là một điểm "xa", nó có thể chỉ kết thúc không có điểm nào được liên kết với nó và đồng thời, nhiều hơn một cụm có thể kết thúc với một centroid duy nhất. Tương tự, nhiều hơn một trung tâm có thể được khởi tạo vào cùng một cụm dẫn đến phân cụm kém. Ví dụ, hãy xem xét các hình ảnh hiển thị bên dưới.

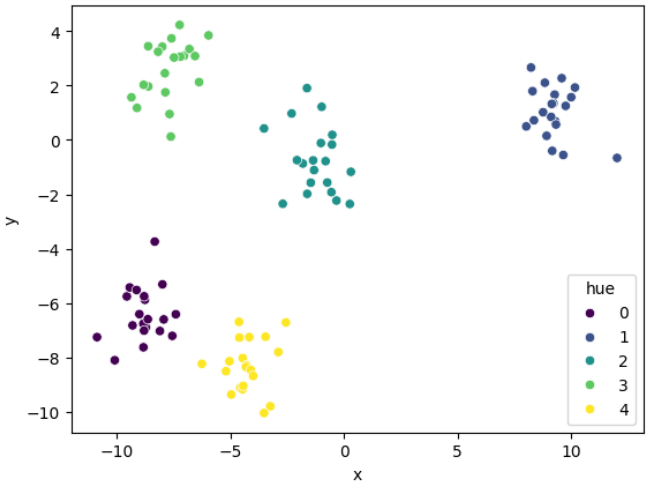
4.6.2 Khởi tạo trung tâm kém dẫn đến phân cụm kém

Vẫn thuật toán K-Means nêu ở trên, giả sử rằng chúng ta có tập dữ liệu như sau:



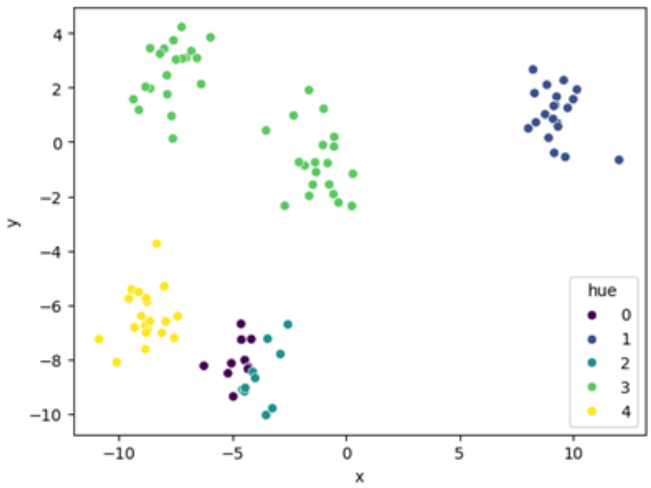
Hình 4.1: Tập dữ liệu minh họa

Nếu phân cụm đúng, tập dữ liệu được chia có dạng như sau:



Hình 4.2: Phân cụm đúng

Tuy nhiên, khi ta chạy thuật toán một vài lần thì việc thuật toán chạy khiến cụm phân bị lỗi không phải là hiếm, không muốn nói là "rất thường xuyên":



Hình 4.3: Phân cụm bị lỗi

4.6.3 Giải quyết vấn đề

Để khắc phục nhược điểm nêu trên, chúng ta sử dụng K-means++. Thuật toán này đảm bảo khởi tạo centroids thông minh hơn và cải thiện chất lượng của phân cụm. Ngoài phần khởi tạo, phần còn lại của thuật toán giống như thuật toán K-means tiêu chuẩn. Đó là K-means++: là thuật toán K-means tiêu chuẩn kết hợp với việc khởi tạo centroids thông minh hơn.

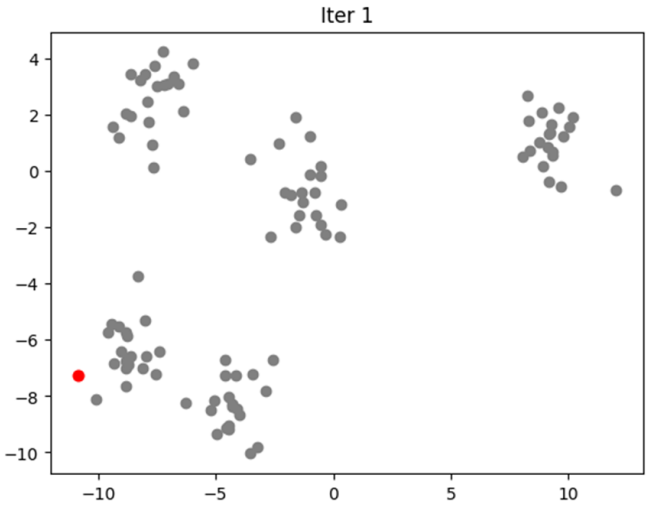
4.6.4 Thuật toán

- B1: Chọn ngẫu nhiên centroid đầu tiên từ các điểm dữ liệu.
- B2: Đối với mỗi điểm dữ liệu chưa được chọn làm tâm cụm, hãy tính khoảng cách giữa điểm đó và tâm cụm gần nhất.
- B3: Chọn trung tâm cụm tiếp theo từ các điểm dữ liệu còn lại với xác suất tỷ lệ thuận với khoảng cách bình phương đến trung tâm cụm gần nhất. Điều này đảm bảo rằng trung tâm mới cách xa các trung tâm hiện có. Điều này giúp đảm bảo rằng các trung tâm cụm ban đầu được phân bổ đều trên toàn bộ tập dữ liệu và không được nhóm lại trong một vùng duy nhất.
  Ở B3 này, để đơn giản khi code thì ta chỉ cần làm như sau: ta tính khoảng cách từ các điểm dữ liệu đến cụm tương ứng của nó. Rồi sau đó ta tìm điểm mà có khoảng cách từ nó đến tâm tương ứng của nó là lớn nhất trong tất cả các khoảng cách ta vừa tính được. Điểm đó sẽ được coi là centroid mới.
- B4: Lặp lại các bước 2 và 3 cho đến khi k centroid được lấy mẫu

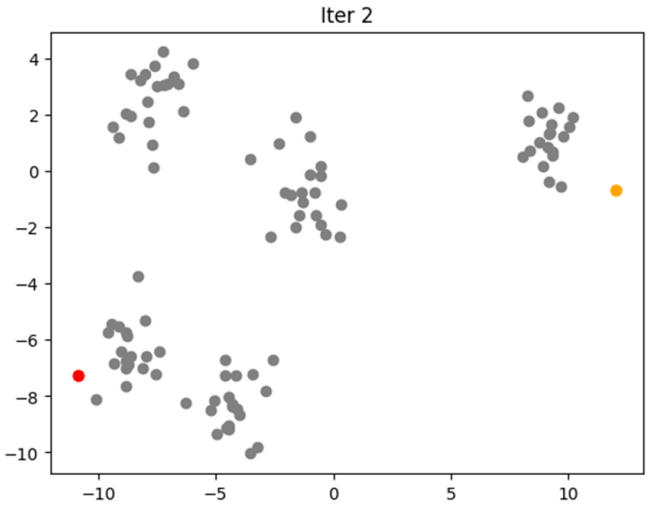
Xem trực quan thuật toán: `https://www.youtube.com/watch?v=HatwtJSsj5Q`

Link source code K-Means++ của nhóm: `https://github.com/thethien8a/PTSL/blob/main/K-means.ipynb`

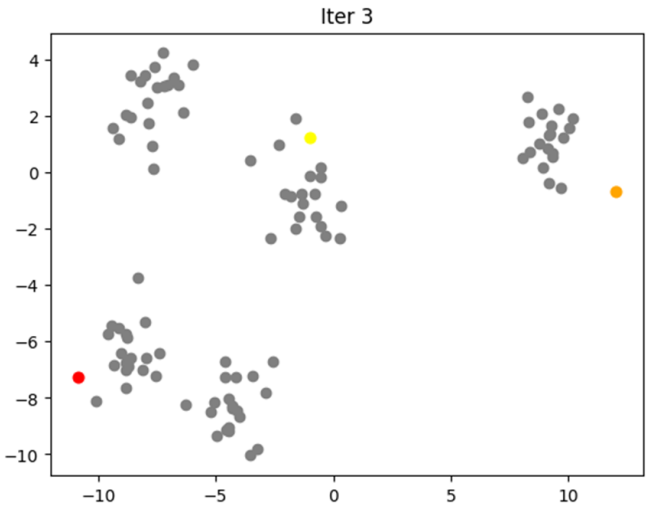
4.6.5 Minh họa trực quan thuật toán K-Means++



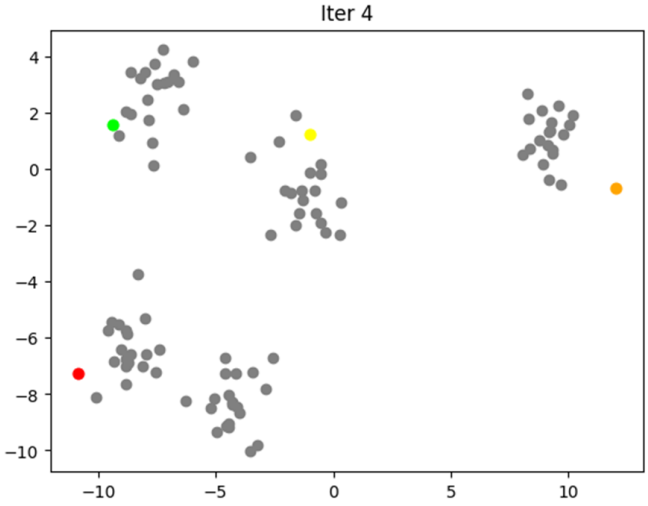
Hình 4.4: Vòng lặp 1



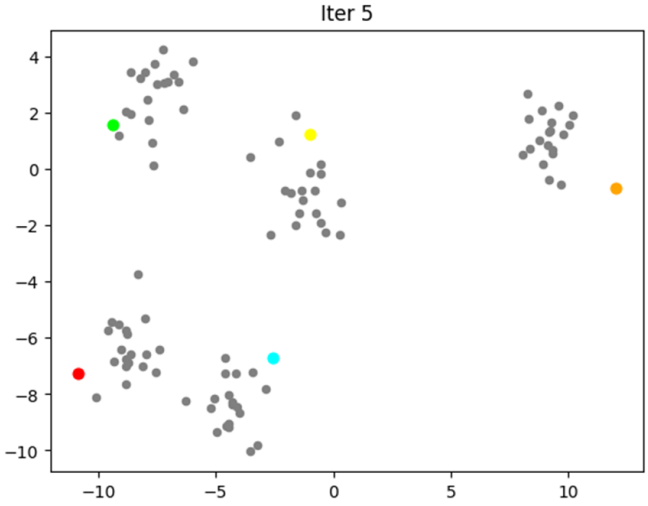
Hình 4.5: Vòng lặp 2



Hình 4.6: Vòng lặp 3



Hình 4.7: Vòng lặp 4



Hình 4.8: Vòng lặp 5

4.6.6 Lợi ích của K-Means++

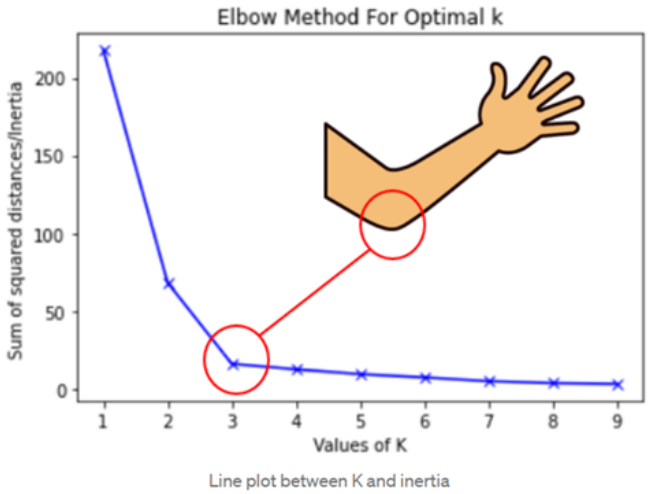
Phương pháp khởi tạo này mang lại sự cải thiện đáng kể trong lỗi cuối cùng của k-means. Mặc dù việc lựa chọn ban đầu trong thuật toán tốn thêm thời gian, nhưng phần k-means tự nó hội tụ rất nhanh sau khi khởi tạo này và do đó, thuật toán thực sự giảm thời gian tính toán. Các tác giả đã kiểm tra phương pháp của họ với các tập dữ liệu thực và tổng hợp, thu được những cải thiện về tốc độ gấp 2 lần và đối với một số tập dữ liệu, cải thiện về lỗi gần 1000 lần. Trong các mô phỏng này, phương pháp mới hầu như luôn hoạt động ít nhất cũng tốt như k-means cơ bản, cả về tốc độ và độ lỗi.
Thêm vào đó, các tác giả tính toán tỷ lệ xấp xỉ cho thuật toán của họ. Thuật toán k-means++ đảm bảo một tỷ lệ xấp xỉ O(log k) trong kỳ vọng (dựa trên tính ngẫu nhiên của thuật toán), trong đó k là số lượng cụm được sử dụng. Điều này trái ngược với k-means cơ bản, có thể tạo ra các phân cụm kém hơn một cách tùy ý so với tối ưu. Một sự tổng quát về hiệu suất của k-means++ với bất kỳ khoảng cách tùy ý nào được cung cấp

4.7 Cách xác định số lượng cụm

4.7.1 Elbow Method

(a) Elbow Method là gì ?

Elbow Method là một kỹ thuật phổ biến được sử dụng để xác định số lượng cụm (k) tối ưu trong bài toán phân cụm (clustering), chẳng hạn như thuật toán K-Means. Nó dựa trên việc quan sát độ biến thiên (inertia) hoặc tổng bình phương sai số (sum of squared distances, SSD) giữa các điểm dữ liệu và trọng tâm của chúng trong từng cụm.



Hình 4.9: Ảnh minh họa phương pháp "khuỷu tay"

(b) Nguyên tắc hoạt động

1. Chạy thuật toán K-Means:
   - Thử nghiệm số lượng cụm k từ $k = 1$ đến $k_{\max}$ (số lượng cụm tối đa mà bạn muốn kiểm tra).
   - Lưu lại tổng bình phương sai số (WCSS - Within-Cluster Sum of Squares) giữa mỗi điểm dữ liệu và trọng tâm gần nhất.
2. Vẽ đồ thị:
   - Trục hoành (x-axis): Số lượng cụm k.
   - Trục tung (y-axis): WCSS hoặc inertia (tổng khoảng cách bình phương).
3. Quan sát "khuỷu tay" (Elbow):
   - Điểm mà độ giảm của WCSS bắt đầu chậm lại, tức là đồ thị chuyển từ giảm nhanh sang giảm chậm, được gọi là "khuỷu tay".
   - Số cụm tại điểm "khuỷu tay" là số cụm k tối ưu.

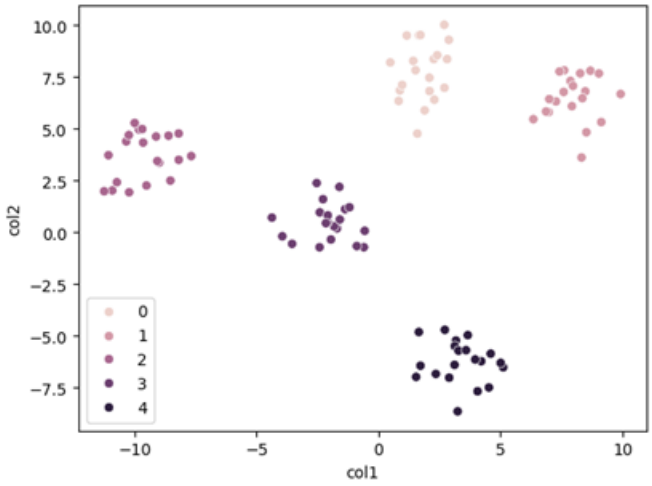
Ở đây

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Trong đó:

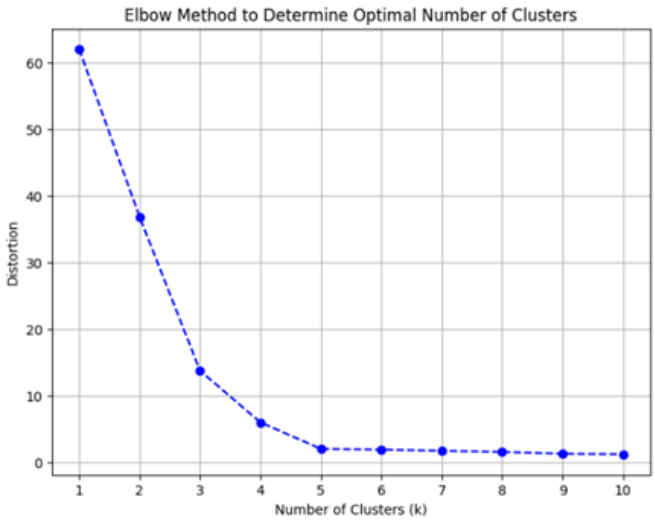
- k: Số lượng cụm.
- C_i: Cụm thứ i.
- x: Điểm dữ liệu thuộc cụm C_i.
- μ_i: Trọng tâm (centroid) của cụm C_i.

(c) Ví dụ minh họa



Hình 4.10: Tập dữ liệu minh họa

Khi ta sử dụng phương pháp "khuỷu tay", kết quả ta có được là hình sau:

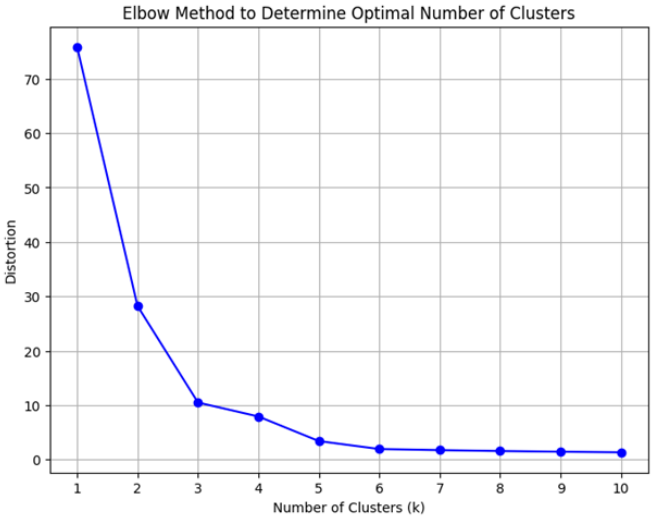


Hình 4.11: Elbow method

Nhận thấy được rằng, tại k=5, dữ liệu bắt đầu có dạng gần giống tuyến tính. Hay nói cách khác, tại k=5 dữ liệu giảm chậm dần, tại đó có dạng giống khuỷu tay khi ta gập vào. Vậy ta có thể kết luận được rằng có 5 cụm trong tập dữ liệu của chúng ta (hay k=5). Đối chiếu lại với hình ảnh tập dữ liệu ở trên, chúng ta có thể dễ dàng thấy có đúng 5 cụm thật.

(d) Mặt hạn chế của Elbow Method

Tuy nhiên, không phải lúc nào thì phương pháp này cũng cho ta kết quả 1 cách trực quan. Ví dụ:



Hình 4.12: Elbow method chưa giải quyết được

Như ở hình ảnh trên, ta có thể nhầm lẫn trong việc kết luận số lượng cụm của tập dữ liệu. Ta có thể kết luận là 6 hoặc 5 hay thậm chí là 4. Thế nên, ta cần tìm thêm một phương pháp khác để xác định được số lượng cụm.

4.7.2 Silhouette Score

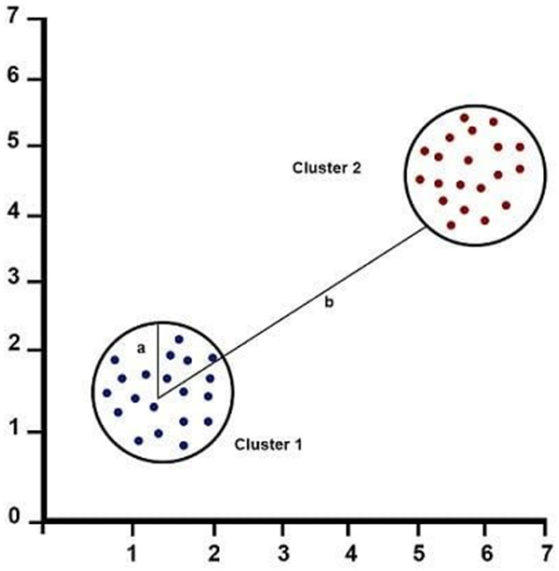
Trong trường hợp như trên, ta cân nhắc việc sử dụng một cách khác để xác định số lượng cụm đó chính là Silhouette Score

(a) Silhouette Score là gì ?

Silhouette Score là một chỉ số đánh giá chất lượng phân cụm trong thuật toán clustering (như K-Means, DBSCAN). Nó đo lường mức độ "chặt chẽ" của các cụm và mức độ "phân tách" giữa các cụm. Silhouette Score cung cấp một giá trị trong khoảng $[-1, 1]$, với ý nghĩa như sau:

- **Silhouette Score gần 1:** Cụm tốt (các điểm dữ liệu gần nhau trong cùng cụm và xa các cụm khác).
- **Silhouette Score gần 0:** Các cụm có thể chồng chéo nhau.
- **Silhouette Score gần -1:** Phân cụm không tốt (các điểm dữ liệu gần cụm khác hơn cụm của chính nó).

(b) Công thức tính Silhouette Score



Hình 4.13: Minh họa các giá trị

Cho điểm dữ liệu i:

- $a(i)$: Khoảng cách trung bình từ i đến tất cả các điểm khác trong cụm của nó (độ "chặt chẽ" của cụm).
- $b(i)$: Khoảng cách trung bình từ i đến tất cả các điểm trong cụm gần nhất (độ "phân tách" giữa các cụm).

Công thức Silhouette Score cho điểm i là:

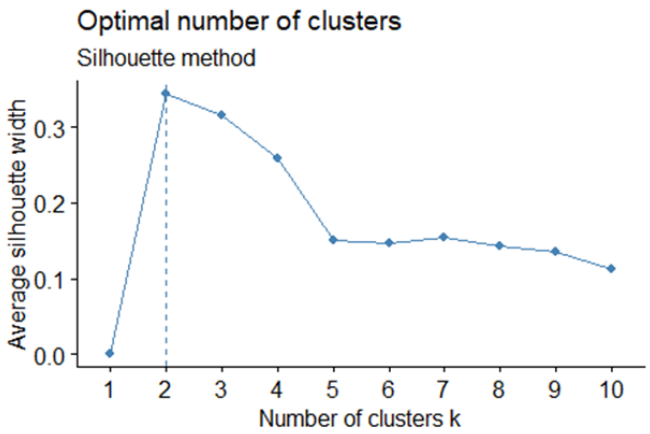
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

- $a(i)$: Càng nhỏ càng tốt (điểm dữ liệu càng gần cụm của nó).
- $b(i)$: Càng lớn càng tốt (điểm dữ liệu càng xa các cụm khác).

Silhouette Score của **toàn bộ tập dữ liệu** là giá trị trung bình của $s(i)$ cho tất cả các điểm dữ liệu i.

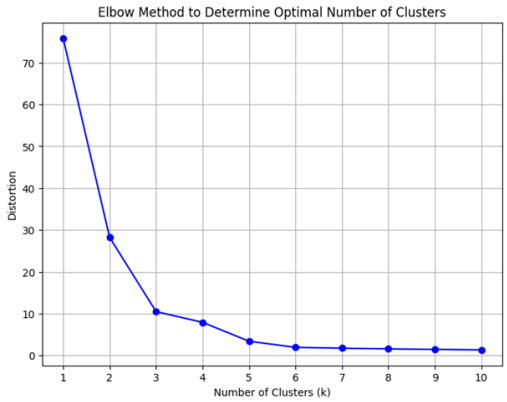
(c) Cách xác định số lượng cụm

Với phương pháp này, ta xác định số cụm tối ưu bằng việc quan sát xem với giá trị nào của k thì Silhoutte Score đạt giá trị lớn nhất thì ta sẽ chọn k đó. Lưu ý rằng ta sẽ vẽ ra đồ thị như hình dưới với trục tung là "Silhouette Score của toàn bộ tập dữ liệu", trục hoành là "giá trị k". Hình dưới đây là minh họa tại k=2 là số cụm được chia phù hợp cho tập dữ liệu:



Hình 4.14: Silhouette Score minh họa

Chúng ta sẽ áp dụng vào bài ban nãy mà thuật toán Elbow Method chưa xử lý được triệt để:



Hình 4.15: Elbow method chưa giải quyết

Hình 4.16: Silhouette Score giải quyết

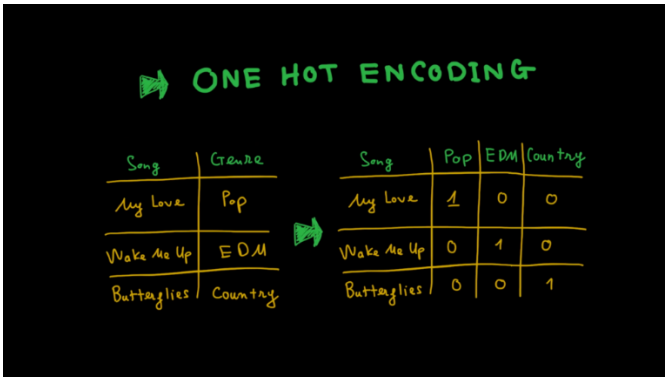
Như ta thấy được rằng, ở hình ảnh của Silhouette Score, với k=6 thì silhouette Score đạt giá trị lớn nhất. Vậy ta chọn k=6

4.8 Những lưu ý khi sử dụng thuật toán K-Means

Để sử dụng thuật toán K-Means một cách hiệu quả, dữ liệu cần thỏa mãn một số điều kiện nhất định nhằm đảm bảo việc phân cụm chính xác và tối ưu. Các điều kiện này bao gồm:

- Dữ liệu số liên tục: K-Means hoạt động tốt với dữ liệu dạng số liên tục. Nếu dữ liệu có các giá trị phân loại (categorical), cần chuyển đổi chúng sang dạng số, ví dụ như sử dụng phương pháp one-hot encoding.

  ⇒ Mã hóa one-hot encoding



Hình 4.17: Phương pháp one hot endcoding

- Phân phối các cụm (clusters) hình cầu và tương đối đều đặn: K-Means giả định rằng các cụm có dạng hình cầu và có kích thước tương tự nhau. Nếu các cụm có hình dạng khác, ví dụ như dạng elip, hoặc phân bố không đồng đều về kích thước, K-Means có thể cho kết quả không chính xác.

  ⇒ Sử dụng biểu đồ phân tán

- Khoảng cách Euclidean phù hợp: K-Means sử dụng khoảng cách Euclidean để đo lường độ tương tự giữa các điểm dữ liệu. Do đó, dữ liệu phải có mối quan hệ tuyến tính và khoảng cách Euclidean phải phản ánh được sự khác biệt giữa các điểm dữ liệu.

  ⇒ Chuẩn hóa dữ liệu

- Không gian đa chiều hợp lý: Trong không gian có nhiều chiều (high-dimensional space), thuật toán K-Means có thể gặp khó khăn vì hiện tượng "curse of dimensionality". Việc giảm số chiều (dimensionality reduction) bằng các phương pháp như PCA (Principal Component Analysis) có thể giúp cải thiện hiệu quả.

  ⇒ Giảm chiều dữ liệu bằng PCA

- Dữ liệu không có nhiễu: K-Means nhạy cảm với nhiễu (outliers). Các điểm dữ liệu nhiễu có thể ảnh hưởng đến trung tâm cụm (centroid) và dẫn đến kết quả không chính xác. Do đó, nên loại bỏ hoặc xử lý các điểm nhiễu trước khi áp dụng thuật toán.

  ⇒ Loại bỏ ngoại lai

- Số cụm k đã biết trước hoặc có thể ước tính: Thuật toán K-Means yêu cầu biết trước số cụm k. Nếu chưa biết số cụm phù hợp, có thể sử dụng phương pháp như Elbow Method hoặc Silhouette Score để xác định giá trị k tốt nhất.

  ⇒ Dùng Elbow hoặc Silhouette Score

Tóm lại, dữ liệu sử dụng K-Means nên có dạng số liên tục, các cụm có hình cầu, không chứa quá nhiều nhiễu và có thể đo lường bằng khoảng cách Euclidean.

4.9 Ứng dụng của phân cụm không phân cấp vào 1 số bài toán thực tế

4.9.1 Nén ảnh

Mục tiêu khi xây dựng thuật toán:

- Giảm số lượng màu sắc trong ảnh: Mục đích chính của nén ảnh bằng K-Means là giảm số lượng màu trong ảnh mà vẫn giữ được chất lượng hình ảnh ở mức chấp nhận được.

- Giảm dung lượng tệp: Bằng cách đại diện các màu tương tự bằng một số cụm màu chính, ảnh nén sẽ chiếm ít bộ nhớ hơn.

  Code thực thi bằng cách nhận đầu vào là 1 ảnh.

- Ta sử dụng thuật toán Kmeans để xác định số cụm chính là số lượng màu ta mong muốn sau khi nén và thay thế các điểm ảnh bằng màu của tâm cụm gần nhất.

Link source code: `https://github.com/thethien8a/PTSL/blob/main/K-means.ipynb`

4.9.2 Lựa chọn bảng màu

Mục tiêu khi xây dựng thuật toán:

- K-Means có thể giúp trích xuất bảng màu (color palette) từ một bức ảnh bằng cách nhóm các màu tương tự nhau và chọn ra các màu đại diện của từng nhóm. Điều này có thể giúp các nhà thiết kế đồ họa hoặc nhà thiết kế thời trang dễ dàng lựa chọn màu sắc cho sản phẩm của họ.

Đầu vào của code là import ảnh, nhập số lượng màu cần lấy.

Link source code: `https://github.com/thethien8a/PTSL/blob/main/K-means.ipynb`

4.9.3 Phân cụm khách hàng

Mục tiêu khi xây dựng thuật toán:

- Phân nhóm khách hàng có đặc điểm tương đồng: Nhóm các khách hàng có hành vi hoặc đặc điểm mua sắm tương tự như thu nhập, mức chi tiêu, sở thích, hoặc lịch sử mua hàng vào cùng một cụm.
- Xác định chiến lược tiếp thị: Giúp doanh nghiệp thiết kế chiến lược tiếp thị hiệu quả hơn, tập trung vào từng nhóm khách hàng cụ thể (ví dụ: khách hàng chi tiêu cao, khách hàng tiềm năng).
- Cá nhân hóa trải nghiệm khách hàng: Đề xuất sản phẩm hoặc dịch vụ phù hợp cho từng cụm khách hàng dựa trên sở thích và hành vi đã quan sát.
- Phân tích hành vi khách hàng: Tìm hiểu và khám phá xu hướng tiêu dùng, giúp doanh nghiệp dự đoán nhu cầu khách hàng.
- Tối ưu hóa quản lý nguồn lực: Hỗ trợ trong việc phân bổ nguồn lực hợp lý hơn, như ưu tiên chăm sóc nhóm khách hàng có giá trị cao hoặc cải thiện trải nghiệm cho nhóm khách hàng có nhu cầu đặc biệt.

Dưới đây là một ví dụ sử dụng thuật toán K-means trong phân tích khách hàng bằng cách phân nhóm dựa trên các đặc điểm như thu nhập và điểm chi tiêu

Link source code: `https://github.com/thethien8a/PTSL/blob/main/Cum_KH.ipynb`

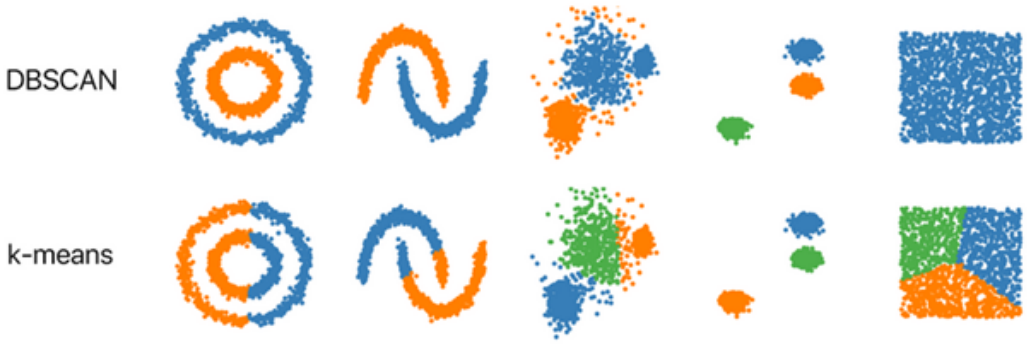
4.10 Giới thiệu thuật toán học không giám sát khác – DBSCAN

4.10.1 Khái niệm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán cơ sở để phân nhóm dựa trên mật độ. Nó có thể phát hiện ra các cụm có hình dạng và kích thước khác nhau từ một lượng lớn dữ liệu chứa nhiễu.

4.10.2 So sánh giữa DBSCAN với K-Means

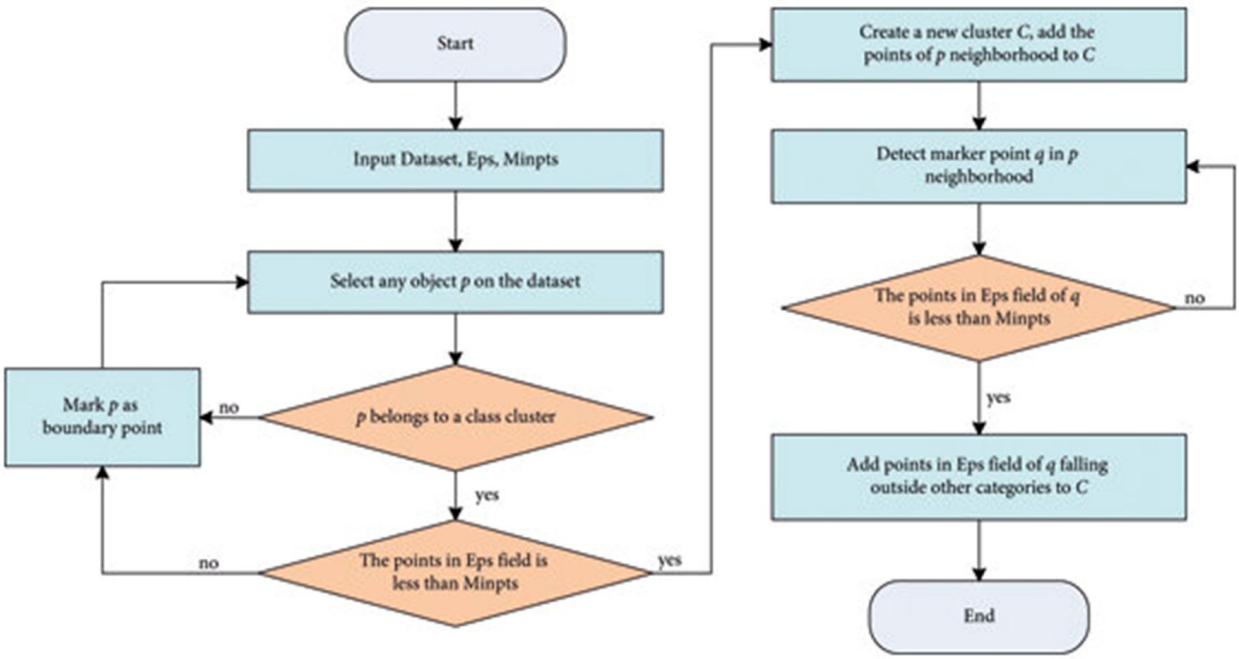
Thuật toán k-Means có thể phân cụm các quan sát có sự tương đồng một cách khá lỏng lẻo. Sau mỗi vòng lặp của thuật toán thì mỗi một quan sát đều được phân vào một cụm nhất định, thậm chí đó là những quan sát nhiễu (noise data) phân bố cách xa tâm cụm. Do đó trong thuật toán k-Means mọi điểm đều ảnh hưởng tới tâm cụm. Chính vì điều này nên dẫn tới khi xuất hiện outliers sẽ ảnh hưởng tới độ chính xác của thuật toán cũng như chất lượng của cụm. Trong DBSCAN thì vấn đề này được khắc phục nhờ cơ chế hình thành cụm đặc biệt mà ở đó các điểm dữ liệu nhiễu sẽ được tách thành một phần riêng mà chúng ta sẽ tìm hiểu cơ chế này ở phần tiếp theo. Thậm chí là đối với những phân phối có hình dạng đặc biệt mà k-Means không phân cụm tốt thì DBSCAN cũng có thể phân cụm được như hình minh hoạ bên dưới:



Hình 4.18: Thuật toán DBSCAN và K-means

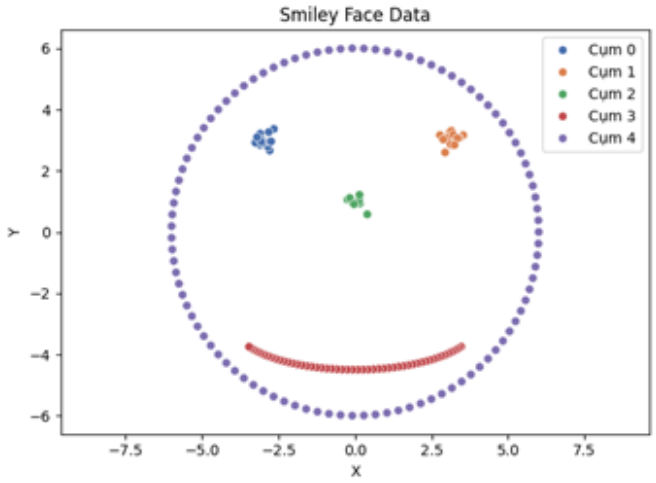
Trong thuật toán DBSCAN cũng không cần khai báo trước số lượng cụm cần phân chia. Đây là một ưu điểm lớn của DBSCAN so với k-Means bởi vì đôi khi chúng ta sẽ không thể biết trước số lượng cụm cần phân chia bao nhiêu là hợp lý, đặc biệt là trên những bộ dữ liệu hoàn toàn mới mà chúng ta chưa từng có kinh nghiệm về chúng. Trong DSCAN chúng ta chỉ cần xác định hàm tính toán khoảng cách và bán kính khoảng cách bao nhiêu được coi là gần nhau để thuật toán tự động thực hiện quá trình phân cụm.

Bên cạnh ưu điểm không cần xác định số lượng cụm thì DBSCAN là thuật toán có tốc độ tính toán rất nhanh, bạn đọc có thể đọc về chủ đề này tại: `https://en.wikipedia.org/wiki/SIGKDD`



Hình 4.19: Mô tả thuật toán DBSCAN

Một ưu điểm nữa của thuật toán DBSCAN là giải quyết bài toán mặt cười mà KMEANS ko làm được:



Hình 4.20: Bài toán mặt cười được giải quyết bằng DBSCAN

Để tìm hiểu rõ hơn thuật toán người đọc có thể tham khảo thêm các nguồn tài liệu sau:

`https://www.youtube.com/watch?v=_A9Tq6mGtLI`

`https://phamdinhkhanh.github.io/deepai-book/ch_ml/DBSCAN.html`

`https://www.cit.ctu.edu.vn/~dtnghi/rech/dir_0/Duy-et-al-New.pdf`

5 Phân cụm dựa trên mô hình thống kê

5.1 Giới Thiệu

Các phương pháp phân cụm phổ biến được thảo luận trước đó bao gồm:

- Liên kết đơn,
- Liên kết hoàn chỉnh,
- Liên kết trung bình,
- Phương pháp Ward,
- Phân cụm K-means.

Các phương pháp này hợp lý theo trực giác, nhưng không dựa trên mô hình để giải thích cách các quan sát được tạo ra. Phân cụm dựa trên mô hình thống kê sẽ khắc phục hạn chế này. Những tiến bộ lớn trong các phương pháp phân cụm đã được thực hiện thông qua việc giới thiệu các mô hình thống kê chỉ ra cách thu thập $(p \times 1)$ phép đo x_j từ N đối tượng được tạo ra. Mô hình phổ biến nhất là mô hình trong đó cụm k có tỷ lệ kỳ vọng p_k của các đối tượng và các phép đo tương ứng được tạo ra bởi hàm mật độ xác suất $f(x)$.
Sau đó nếu có K cụm, vectơ quan sát cho một đối tượng duy nhất (gồm p chiềun ví dụ như chiều cao,cân nặng...) có thể được mô hình hóa như phát sinh từ một phân phối hỗn hợp:

$$f_{\text{Mix}}(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}) \tag{12.1}$$

trong đó:

- p_k: Xác suất để một đối tượng thuộc về cụm k, với $p_k \geqslant 0$ và $\sum_{k=1}^{K} p_k = 1$,
- $f_k(\mathbf{x})$: Hàm mật độ của cụm k.

Phân phối $f_{\text{Mix}}(x)$ này được gọi là hỗn hợp của K phân phối $f_1(x), f_2(x), \ldots, f_K(x)$, vì các quan sát được tạo ra từ phân phối thành phần $f_k(x)$ với xác suất p_k. Bộ sưu tập N vectơ quan sát được tạo ra từ phân phối này sẽ là hỗn hợp các quan sát từ các phân phối thành phần.

5.2 Mô hình hỗn hợp chuẩn đa biến (Gaussian Mixture Model - GMM)

Mô hình hỗn hợp chuẩn đa biến (GMM) là một loại mô hình phổ biến trong đó các phân phối thành phần $f_k(x)$ được giả định là các hàm mật độ chuẩn đa biến $\mathcal{N}_p(\mu_k, \Sigma_k)$.
Trong đó mô hình chuẩn đa biến cho một quan sát x là:

$$\begin{aligned} f_k(x) &= \sum_{k=1}^{K} p_k \mathcal{N}_p(x \mid \mu_k, \Sigma_k) \\ &= \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right). \end{aligned} \tag{12.2}$$

Các cụm được tạo ra bởi mô hình hỗn hợp K chuẩn được mô hình hóa như các phân phối chuẩn đa biến, nên có hình elip với mật độ quan sát lớn nhất ở gần tâm của mỗi cụm (tức là gần vector trung bình μ_k).
Hình dạng và kích thước của các cụm được xác định bởi ma trận hiệp phương sai Σ_k, có thể khác nhau giữa các cụm, giúp mô hình hóa linh hoạt hơn so với các phương pháp phân cụm dựa trên khoảng cách.

5.3 Suy luận dựa trên khả năng (Likelihood-based Inference)

Suy luận dựa trên khả năng, đối với N đối tượng và một số cụm cố định K, được biểu diễn dưới dạng hàm hợp lý :

$$\begin{aligned} L(p_1, p_2, \ldots, p_K, \mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K) &= \prod_{i=1}^{N} f_{\text{mix}}(x_i | \mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K) \\ &= \prod_{i=1}^{N} \left(\sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)} \right) \end{aligned} \tag{12.3}$$

Trong đó các tỷ lệ $p_1, ..., p_k$, các vectơ trung bình $\mu_1, \ldots, \mu_K$ và các ma trận hiệp phương sai $\Sigma_1, \ldots, \Sigma_K$ là các tham số chưa biết. Các phép đo cho các đối tượng khác nhau được coi là các quan sát độc lập và phân phối giống hệt nhau từ phân phối hỗn hợp.
Thông thường có quá nhiều tham số chưa biết cho các tham số để đưa ra suy luận khi số lượng đối tượng được nhóm lại ít nhất là vừa phải. Tuy nhiên, có thể đưa ra một số kết luận nhất định liên quan đến các tình huống mà phương pháp nhóm lại theo phương pháp Heuristic sẽ hoạt động tốt. Đặc biệt, quy trình dựa trên khả năng xảy ra theo mô hình hỗn hợp chuẩn với tất cả Σ_k là cùng một bội số của ma trận đồng nhất, gần giống với phương pháp nhóm lại K-means và phương pháp Ward. Cho đến nay, chưa có mô hình thống kê nào được đưa ra mà sự hình thành nhóm quy trình gần giống như liên kết đơn, liên kết hoàn chỉnh hoặc liên kết trung bình.
Quan trọng nhất là, theo trình tự các mô hình hỗn hợp (12.2) cho K khác nhau, các vấn đề về việc lựa chọn số lượng cụm và lựa chọn phương pháp cụm thích hợp đã được giảm xuống thành vấn đề lựa chọn trạng thái thích hợp mô hình thống kê. Đây là một tiến bộ lớn.
Một cách tiếp cận tốt để lựa chọn một mô hình là trước tiên thu được các ước tính tối đa của hàm hợp lý $\hat{p}_1, \ldots, \hat{p}_k, \hat{\mu}_1, \ldots, \hat{\mu}_k, \hat{\Sigma}_1, \ldots, \hat{\Sigma}_k$ cho một số cụm cố định K. Giá trị của hàm hợp lý tối đa cho biết mức độ phù hợp của dữ liệu với mô hình có số cụm K đã chọn. Các ước tính này phải được thu được bằng số bằng phần mềm chuyên dụng. Giá trị kết quả tối đa của hàm hợp lý:

$$L_{\max} = L(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_K, \hat{\mu}_1, \hat{\Sigma}_1, \ldots, \hat{\mu}_K, \hat{\Sigma}_K)$$

cung cấp cơ sở cho việc lựa chọn mô hình.Làm thế nào để quyết định giá trị hợp lý cho số lượng cụm K? Để so sánh các mô hình với số lượng tham số khác nhau người ta sử dụng một tiêu chí phạt, một hình phạt được trừ đi từ gấp đôi giá trị tối đa của hàm hợp lý để đưa ra.

$$-2 \cdot \ln L_{\max} - \text{Penalty} \tag{12.4}$$

Trong đó, hình phạt phụ thuộc vào số lượng tham số ước tính và số lượng quan sát N. Điều này giúp tránh việc chọn các mô hình quá phức tạp (với số cụm lớn) mà có thể không thực sự cần thiết.
Vì tổng xác suất p_k bằng 1, nên chỉ có $K - 1$ xác suất cần được ước tính. Còn lại, có $K \times p$ giá trị trung bình và $K \times p \times \frac{(p+1)}{2}$ giá trị phương sai và hiệp phương sai cần được ước tính. Tổng số tham số cần ước tính trong mô hình là:

$$(K - 1) + K \cdot p + K \cdot p \cdot \frac{(p + 1)}{2} = (K.\frac{1}{2}(p + 1)(p + 2) - 1)$$

5.4 Tiêu chuẩn AIC và BIC

AIC (Akaike Information Criterion) là một thước đo giúp chọn mô hình tốt nhất bằng cách cân bằng giữa độ phù hợp của mô hình với dữ liệu và độ phức tạp của mô hình.
Tiêu chuẩn thông tin Akaike (AIC), hình phạt là 2N X (số lượng tham số) nên

$$\text{AIC} = 2 \cdot \ln L_{\max} - 2N \cdot (K\frac{1}{2}(p+1)(p+2) - 1) \tag{12.5}$$

Tiêu chuẩn thông tin Bayesian (BIC) tương tự nhưng sử dụng logarit của số tham số trong hàm phạt,được sử dụng khi số lượng quan sát N lớn.

$$\text{BIC} = 2 \cdot \ln L_{\max} - 2 \cdot \ln(N) \cdot (K\frac{1}{2}(p+1)(p+2) - 1) \tag{12.6}$$

Đôi khi vẫn có khó khăn với quá nhiều tham số trong mô hình hỗn hợp nên các cấu trúc đơn giản được giả định cho Σ_k. Đặc biệt, các cấu trúc phức tạp hơn được phép nhưng được chỉ ra trong bảng sau.

Giả định Σ_k	Tổng số tham số	BIC
$\Sigma_k = \eta I$	$K(p+1)$	$\ln L_{\max} - 2\ln(N)(K(p+1))$
$\Sigma_k = \eta_k I$	$K(p+2)-1$	$\ln L_{\max} - 2\ln(N)K(p+2-1)$
$\Sigma_k = \eta_k \text{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$	$K(p+2)+p-1$	$\ln L_{\max} - 2\ln(N)(K(p+2)+p-1)$

Ngay cả đối với một số cụm cố định, việc ước tính mô hình hỗn hợp cũng phức tạp. Một gói phần mềm hiện tại, MCLUST, có trong thư viện phần mềm R, kết hợp cụm phân cấp, thuật toán EM và tiêu chí BIC để phát triển một mô hình cụm phù hợp.

5.5 Thuật toán EM (Expectation Maximization)

Thuật toán EM (Expectation-Maximization) là một phương pháp lặp để tìm ước lượng hợp lý cực đại (maximum likelihood estimates) của các tham số trong các mô hình thống kê, đặc biệt khi dữ liệu bị thiếu hoặc có những biến ẩn (hidden variables) không được quan sát trực tiếp. Thuật toán EM rất phổ biến trong nhiều lĩnh vực như phân tích dữ liệu, học máy, và mô hình hỗn hợp.
Thuật toán EM bao gồm hai bước chính:
1.Bước Kỳ vọng (Expectation - E-step)
Trong bước E (Expectation - Kỳ vọng) của thuật toán EM, mục tiêu là tính giá trị kỳ vọng của hàm hợp lý đầy đủ, có tính đến sự không chắc chắn của các biến ẩn. Cụ thể, bạn cần tính xác suất có điều kiện mà mỗi điểm dữ liệu thuộc về từng cụm hoặc phân phối thành phần.
Xác suất có điều kiện mà quan sát x_i thuộc về cụm hoặc phân phối Gaussian thành phần k, được gọi là trọng số trách nhiệm γ_{ik}. Công thức:

$$\gamma_{ik} = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)} \tag{12.7}$$

trong đó:

- π_k: Xác suất tiên nghiệm của thành phần Gaussian thứ k,
- $N(x_i \mid \mu_k, \Sigma_k)$: Hàm mật độ xác suất của phân phối Gaussian với trung bình μ_k và ma trận hiệp phương sai Σ_k,
- K: Tổng số thành phần Gaussian.

Trong bước E của thuật toán, một ma trận $(N \times K)$ được tạo có hàng thứ j chứa các ước lượng của xác suất có điều kiện (trên các ước lượng tham số hiện tại) mà quan sát x; thuộc cụm $1, 2, \ldots, K$. Vì vậy, tại hội tụ, quan sát thứ j, được gán cho cụm k mà xác suất có điều kiện:

$$p(k \mid x_j) = \hat{p}_j f(x_j \mid k) / \sum_{i=1}^{K} \hat{p}_i f(x_i \mid k) \tag{12.8}$$

2. Bước Tối đa hóa (Maximizatiọn - M-step)

Mục tiêu của bước này là tối đa hóa hàm hợp lý để ước tính lại các tham số của phân phối Gaussian trong mô hình hỗn hợp.

Cập nhật các tham số μ_k, Σ_k, và π_k (trung bình, phương sai và tỷ lệ của mỗi cụm) sao cho hàm hợp lý đạt giá trị cực đại.

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_{i=1}^{N} \gamma_{ik}}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} \gamma_{ik}}$$

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$$

Quá trình lặp lại

- Bước E và bước M được thực hiện lặp đi lặp lại cho đến khi hội tụ, tức là khi sự thay đổi trong các ước lượng tham số giữa hai lần lặp liên tiếp là rất nhỏ:

  $$\|\hat{\theta}^{(t+1)} - \hat{\theta}^{(t)}\| < \epsilon$$

  Trong đó, $\hat{\theta}^{(t+1)}$ và $\hat{\theta}^{(t)}$ là các tham số ước tính trong lần lặp thứ $t+1$ và t, và ϵ là một ngưỡng rất nhỏ, xác định mức độ hội tụ của thuật toán.

- Khi thuật toán hội tụ, kết quả là một tập hợp các ước lượng hợp lý cực đại (MLE) của các tham số, tức là các tham số $\hat{\theta}$ tối ưu mà không còn sự thay đổi lớn giữa các lần lặp.

5.6 Ví dụ: Mô hình phân cụm dữ liệu loài hoa

Xem xét dữ liệu loài hoa trong bảng dưới đây:

Table 11.5 *(continued)*

π_1: *Iris setosa*				π_2: *Iris versicolor*				π_3: *Iris virginica*			
Sepal length x_1	Sepal width x_2	Petal length x_3	Petal width x_4	Sepal length x_1	Sepal width x_2	Petal length x_3	Petal width x_4	Sepal length x_1	Sepal width x_2	Petal length x_3	Petal width x_4
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

Hình 5.1: Dữ liệu các loài hoa

5.6.1 Giới thiệu về tập dữ liệu Iris

Tập dữ liệu Iris là một trong những tập dữ liệu phổ biến và được sử dụng rộng rãi trong lĩnh vực học máy và thống kê. Tập dữ liệu này thường được sử dụng để thực hành và kiểm thử các thuật toán phân loại và gom cụm. **Các Đặc Trưng**

- **(a) Sepal Length (Chiều Dài Đài Hoa):** Đo lường từ đỉnh đài hoa đến đáy đài hoa.
- **(b) Sepal Width (Chiều Rộng Đài Hoa):** Đo lường chiều rộng của đài hoa.
- **(c) Petal Length (Chiều Dài Cánh Hoa):** Đo lường từ đỉnh cánh hoa đến đáy cánh hoa.
- **(d) Petal Width (Chiều Rộng Cánh Hoa):** Đo lường chiều rộng của cánh hoa.

Loài Hoa Tập dữ liệu Iris chứa thông tin về ba loài hoa Iris khác nhau:

- **(a) Setosa:** Iris setosa.
- **(b) Versicolor:** Iris versicolor.
- **(c) Virginica:** Iris virginica.

Ứng Dụng Tập dữ liệu Iris thường được sử dụng để kiểm thử và so sánh hiệu suất của các thuật toán học máy, đặc biệt là trong các nhiệm vụ phân loại và gom cụm. Nó cũng thường được sử dụng trong giảng dạy và hướng dẫn để giới thiệu các khái niệm cơ bản của học máy và thống kê.

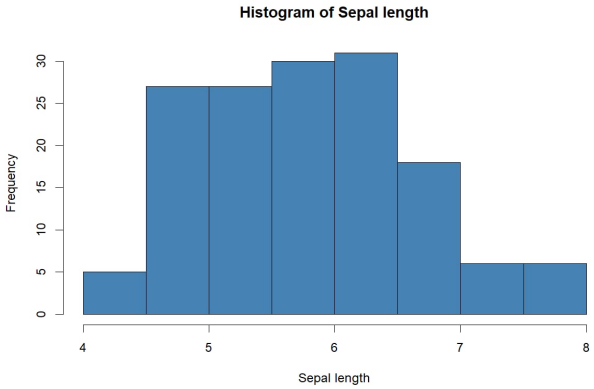
5.6.2 Áp dụng mô hình phân cụm thống kê (GMM) và thuật toán EM với tập dữ liệu Iris sử dụng ngôn ngữ R

a, Trực quan hóa dữ liệu
Đầu tiên, chúng ta sẽ lấy bộ dữ liệu đã cài đặt sẵn trong ngôn ngữ R.

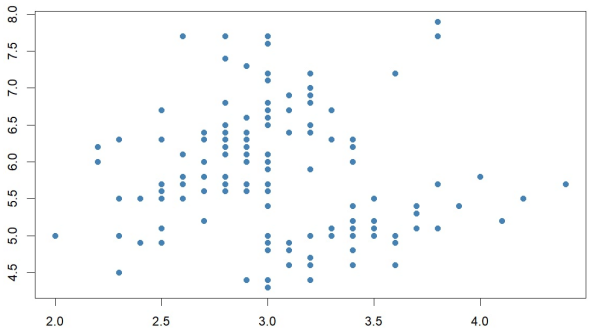
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa :50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicolor:50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	

Hình 5.2: Bảng thông tin về dữ liệu

Biểu diễn dữ liệu của Sepal Length dưới dạng các biểu đồ.



Hình 5.3: Biểu đồ cột



Hình 5.4: Biểu đồ phân tán

b, Giả định thứ nhất:

$$\Sigma_k = \eta_k I,\ k = 1, 2, 3.$$

Tiêu chí chọn số cụm:

- Sử dụng MCLUST trong R.
- Số cụm $K = 3$ được chọn với các trung tâm ước tính:

$$\mu_1 = \begin{bmatrix} 5.01 \\ 3.43 \\ 1.46 \\ 0.25 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 5.90 \\ 2.75 \\ 4.40 \\ 1.43 \end{bmatrix}, \quad \mu_3 = \begin{bmatrix} 6.85 \\ 3.07 \\ 5.73 \\ 2.07 \end{bmatrix}.$$

Các thông số của mô hình K=3

Hệ số tỷ lệ phương sai-hiệp phương sai:

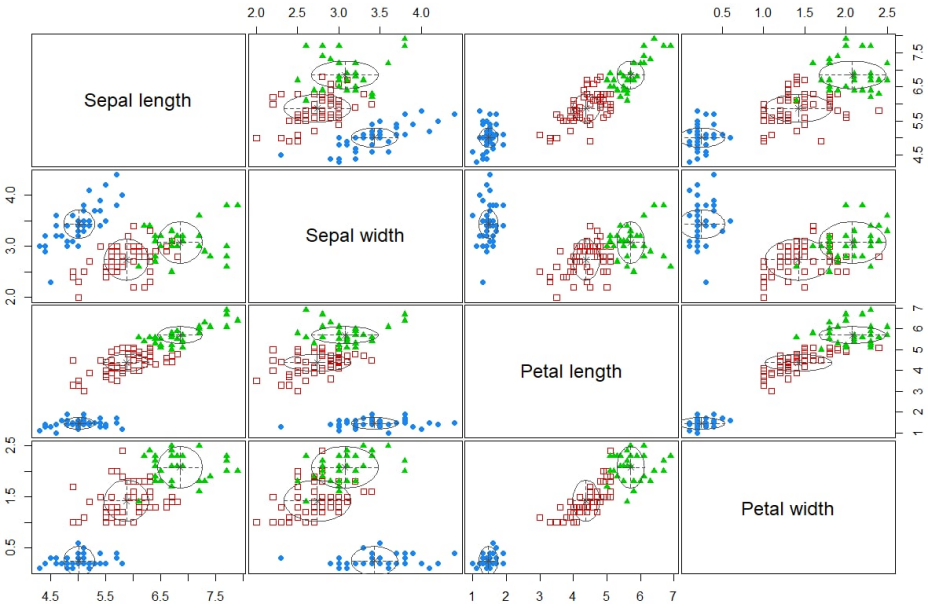
$$\hat{\eta}_1 = 0.076, \quad \hat{\eta}_2 = 0.163, \quad \hat{\eta}_3 = 0.163.$$

Tỷ lệ pha trộn ước tính:

$$p_1 = 0.3333, \quad p_2 = 0.4133, \quad p_3 = 0.2534.$$

Chỉ số BIC:

$$\text{BIC} = \ln L_{\max} - 2\ln(N)K\big(p + 2 - 1\big) = -853.8.$$



Hình 5.5: Biểu đồ trực quan với K=3

c, Giả định thứ hai: Thực hiện phân cụm tối đa K=9 với các cấu trúc ma trận hiệp phương sai Σ_k khác nhau.

```
Bayesian Information Criterion (BIC):
          EII        VII        EEI        VEI        EVI        VVI        EEE        VEE        EVE
1 -1804.0854 -1804.0854 -1522.1202 -1522.1202 -1522.1202 -1522.1202 -829.9782 -829.9782 -829.9782
2 -1123.4117 -1012.2352 -1042.9679  -956.2823 -1007.3082  -857.5515 -688.0972 -656.3270 -657.2263
3  -878.7650  -853.8144  -813.0504  -779.1566  -797.8342  -744.6382 -632.9647 -605.3982 -666.5491
4  -893.6140  -812.6048  -827.4036  -748.4529  -837.5452  -751.0198 -646.0258 -604.8371 -705.5435
5  -782.6441  -742.6083  -741.9185  -688.3463  -766.8158  -711.4502 -604.8131        NA -723.7199
6  -715.7136  -705.7811  -693.7908  -676.1697  -774.0673  -707.2901 -609.8543 -609.5584 -661.9497
7  -731.8821  -698.5413  -713.1823  -680.7377  -813.5220  -766.6500 -632.4947        NA -699.5102
8  -725.0805  -701.4806  -691.4133  -679.4640  -740.4068  -764.1969 -639.2640 -654.8237 -700.4277
9  -694.5205  -700.0276  -696.2607  -702.0143  -767.8044  -755.8290 -653.0878        NA -729.6651
          VVE        EEV        VEV        EVV        VVV
1 -829.9782 -829.9782 -829.9782 -829.9782 -829.9782
2 -605.1841 -644.5997 -561.7285 -658.3306 -574.0178
3 -636.4259 -644.7810 -562.5522 -656.0359 -580.8396
4 -639.7078 -699.8684 -602.0104 -725.2925 -630.6000
5 -632.2056 -652.2959 -634.2890        NA -676.6061
6 -664.8224 -664.4537 -679.5116        NA -754.7938
7 -690.6108 -709.9530 -704.7699 -809.8276 -806.9277
8 -709.9392 -735.4463 -712.8788 -831.7520 -830.6373
9 -734.2997 -758.9348 -748.8237 -882.4391 -883.6931

Top 3 models based on the BIC criterion:
    VEV,2     VEV,3     VVV,2
-561.7285 -562.5522 -574.0178
```

Hình 5.6: Bảng các kết quả thu được

Với tối đa $K = 9$, lựa chọn tốt nhất là mô hình hỗn hợp hai nhóm với hiệp phương sai không bị ràng buộc là $K = 2$ với BIC = -561.7285. Xác suất hỗn hợp ước tính là $\hat{p}_1 = 0.3333$ và $\hat{p}_2 = 0.6667$.

Các trung tâm nhóm ước tính:

$$\mu_1 = \begin{bmatrix} 5.01 \\ 3.43 \\ 1.46 \\ 0.25 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 6.26 \\ 2.87 \\ 4.91 \\ 1.81 \end{bmatrix}$$

Hai ma trận hiệp phương sai ước tính:

$$\hat{\Sigma}_1 = \begin{bmatrix} 1.1218 & 0.0972 & 0.016 & 0.0101 \\ 0.0972 & 0.1408 & 0.0115 & 0.0091 \\ 0.0160 & 0.0115 & 0.0296 & 0.0059 \\ 0.0101 & 0.0091 & 0.0059 & 0.0109 \end{bmatrix}, \quad \hat{\Sigma}_2 = \begin{bmatrix} 0.4530 & 0.1209 & 0.4489 & 0.1655 \\ 0.1209 & 0.1096 & 0.1414 & 0.0792 \\ 0.4489 & 0.1414 & 0.6748 & 0.2858 \\ 0.1655 & 0.0792 & 0.2858 & 0.1786 \end{bmatrix}.$$

6 Thuật toán MDS

6.1 Giới thiệu

6.1.1 Vấn đề

Khi chúng ta có một tập dữ liệu lớn với nhiều thông tin (ví dụ: dữ liệu về khách hàng, sản phẩm, thị trường...), việc trực quan hóa và phân tích toàn bộ dữ liệu này trở nên rất khó khăn vì con người chỉ có thể hình dung được không gian 2D hoặc 3D.
Để giải quyết vấn đề này, các nhà khoa học dữ liệu đã tìm ra cách "nén" dữ liệu nhiều chiều này vào một không gian có số chiều thấp hơn (ví dụ: 2D hoặc 3D) mà vẫn giữ được tối đa thông tin ban đầu.

6.1.2 Giải pháp

Mở rộng đa chiều (Multidimensional scaling): Đây là một nhóm các kỹ thuật giúp chúng ta vẽ các điểm dữ liệu lên một biểu đồ 2D hoặc 3D sao cho khoảng cách giữa các điểm trên biểu đồ càng gần với khoảng cách thực tế giữa các điểm dữ liệu ban đầu càng tốt.
Có hai loại mở rộng đa chiều:
• Mở rộng đa chiều phi tuyến tính: Chỉ sử dụng thứ tự của các khoảng cách để vẽ biểu đồ.
• Mở rộng đa chiều tuyến tính (hay phân tích tọa độ chính): Sử dụng cả thứ tự và độ lớn của các khoảng cách để vẽ biểu đồ.

6.1.3 Mục tiêu

Giảm thiểu biến dạng: Khi "nén" dữ liệu từ không gian nhiều chiều xuống không gian 2D hoặc 3D, chúng ta không thể tránh khỏi việc mất một phần thông tin. Mục tiêu của các kỹ thuật mở rộng đa chiều là giảm thiểu sự mất mát thông tin này, tức là giữ cho các điểm dữ liệu trên biểu đồ vẫn giữ được mối quan hệ tương đối như trong dữ liệu ban đầu.

6.1.4 Ví dụ

Phân tích dữ liệu về các loại rượu vang
Giả sử chúng ta có một tập dữ liệu lớn về các loại rượu vang khác nhau. Mỗi loại rượu được mô tả bởi nhiều đặc trưng như:
• Thành phần: tỷ lệ tannin, acid, đường,...
• Cảm quan: mùi hương, vị, độ đậm đặc,...
• Nguồn gốc: quốc gia, vùng, giống nho,...
• Đánh giá: điểm số của các chuyên gia, giá cả,...
Với nhiều đặc trưng như vậy, chúng ta khó có thể trực quan hóa và so sánh các loại rượu một cách dễ dàng. Làm thế nào để tìm ra các nhóm rượu vang có hương vị tương đồng hoặc các yếu tố nào ảnh hưởng đến chất lượng của rượu?
Giải pháp:
Sử dụng mở rộng đa chiều: Chúng ta có thể sử dụng kỹ thuật mở rộng đa chiều để "nén" dữ liệu về các loại rượu vang vào một không gian 2D hoặc 3D.

Kết quả:
• Tìm ra các nhóm rượu: Các loại rượu có hương vị tương đồng sẽ nằm gần nhau trên biểu đồ. Ví dụ, các loại rượu vang đỏ đậm đà, tannin cao sẽ tập trung ở một khu vực, trong khi các loại rượu

vang trắng nhẹ nhàng, thanh mát sẽ tập trung ở một khu vực khác.
• Xác định các yếu tố ảnh hưởng: Bằng cách phân tích vị trí của các điểm dữ liệu trên biểu đồ, chúng ta có thể tìm ra mối liên hệ giữa các đặc trưng của rượu và hương vị của chúng. Ví dụ, chúng ta có thể thấy rằng tỷ lệ tannin có liên quan chặt chẽ đến độ đậm đà của rượu.

6.1.5 Bài toán ví dụ

Bài toán gốc (5D):
Có 5 điểm trong không gian 5 chiều (dimension).
Tọa độ của các điểm:
Điểm 1: (1, 2, 3, 4, 5)
Điểm 2: (2, 3, 4, 5, 6)
Điểm 3: (5, 5, 5, 5, 5)
Điểm 4: (7, 6, 2, 2, 1)
Điểm 5: (6, 4, 3, 7, 8)

Yêu cầu:
Giảm số chiều từ 5D xuống 2D hoặc 3D bằng MDS.
Bước 1 - Tính toán ma trận khoảng cách
Khoảng cách Euclid giữa hai điểm:

$$D_{ij} = \sqrt{\sum_{k=1}^{5} (x_{i,k} - x_{j,k})^2}$$

Ví dụ:
• Khoảng cách giữa Điểm 1 và Điểm 2:

$$D_{12} = \sqrt{(1-2)^2 + (2-3)^2 + (3-4)^2 + (4-5)^2 + (5-6)^2} = \sqrt{5} \approx 2.236$$

• Tiếp tục tính các khoảng cách giữa các cặp điểm.

Bước 2: Ma Trận Khoảng Cách (D)
Ma trận khoảng cách (D):

$$D = \begin{bmatrix} 0 & 2.236 & 6.480 & 8.888 & 9.380 \\ 2.236 & 0 & 4.690 & 8.775 & 9.110 \\ 6.480 & 4.690 & 0 & 7.071 & 8.062 \\ 8.888 & 8.775 & 7.071 & 0 & 6.855 \\ 9.380 & 9.110 & 8.062 & 6.855 & 0 \end{bmatrix}$$

Bước 3: Áp Dụng MDS
MDS (Multidimensional Scaling):
• MDS cố gắng ánh xạ các điểm từ không gian 5D vào không gian 2D/3D.
• Đảm bảo khoảng cách giữa các điểm được bảo toàn tối đa.

Thực hiện bằng Python:
• Khởi tạo MDS

• Kết quả: Tọa độ của các điểm trong không gian 2 chiều.

Kết Quả Và Ý Nghĩa

Kết quả:

Ví dụ tọa độ sau khi giảm chiều xuống 2D:

• Điểm 1: (0.5, 1.2)
• Điểm 2: (1.1, 0.8)
• Điểm 3: (4.0, 3.9)
• Điểm 4: (5.2, 6.1)
• Điểm 5: (6.0, 5.7)

Ý nghĩa:

• Giảm số chiều giúp trực quan hóa và phân tích dễ dàng hơn.
• Khoảng cách trong không gian mới phản ánh tương đối chính xác khoảng cách trong không gian gốc.

6.2 Một số khái niệm dùng trong chia tỷ lệ đa chiều

6.2.1 Khoảng cách và độ tương đồng

Khoảng cách:

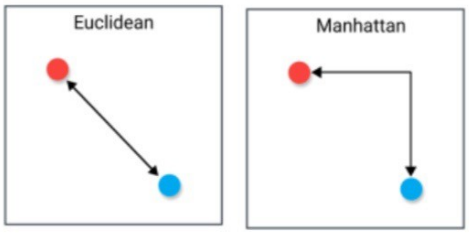
Khoảng cách là một giá trị số đo lường độ dài hay độ tách biệt giữa hai điểm trong không gian. Nó cung cấp thông tin về cách mà hai đối tượng được phân biệt với nhau và có thể làm cơ sở cho việc so sánh và phân loại các đối tượng trong các bài toán phân tích dữ liệu.

Có nhiều loại khoảng cách khác nhau được sử dụng tùy thuộc vào loại dữ liệu và bài toán cụ thể. Một số ví dụ phổ biến về khoảng cách bao gồm:

• **Khoảng cách Euclide**: Là khoảng cách đo bằng đoạn thẳng giữa hai điểm trong không gian Euclidean (không gian hai chiều hoặc nhiều chiều).
• **Khoảng cách Manhattan**: Là tổng của các khoảng cách giữa các cặp đối tượng trong không gian hai chiều hoặc nhiều chiều.

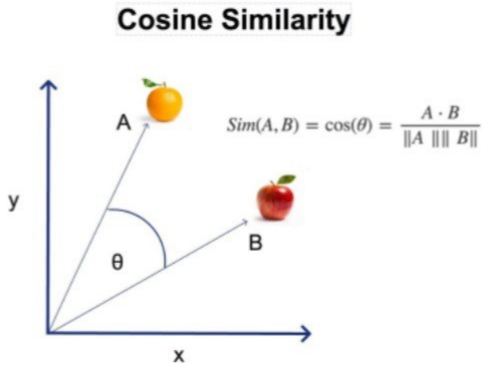
Độ tương đồng:

Độ tương đồng (hoặc độ tương tự) là một giá trị số đo lường mức độ giống nhau hoặc tương tự giữa hai đối tượng. Đối với độ tương đồng, giá trị càng lớn thì hai đối tượng càng giống nhau. Thường thì độ tương đồng là nghịch đảo của khoảng cách, tức là khi khoảng cách tăng thì độ tương đồng giảm và ngược lại. Độ tương đồng thường được sử dụng trong các bài toán so sánh, gom nhóm, và xếp hạng.



Hình 6.1: Khoảng cách Euclidean và Manhattan

Ví dụ:
Độ tương đồng Cosine: Đo lường mức độ tương tự hướng của hai vector. Giá trị gần 1 thể hiện hai vector cùng hướng (tương đồng), giá trị gần -1 thể hiện hai vector đối diện nhau (tương phản), và giá trị gần 0 thể hiện hai vector vuông góc nhau (không tương đồng).



Hình 6.2: Độ tương đồng Cosine

Lựa chọn phương pháp khoảng cách và độ tương đồng phụ thuộc vào loại dữ liệu và mục tiêu của bài toán. Một phép đo lường tốt sẽ giúp ta hiểu và phân tích dữ liệu một cách hiệu quả, đồng thời áp dụng các phương pháp phân tích dữ liệu thích hợp. Phương pháp đo lường khoảng cách và độ tương đồng đóng vai trò quan trọng trong việc xác định mức độ tách biệt hoặc tương tự giữa các điểm, đối tượng trong dữ liệu. Dưới đây là một số phương pháp phổ biến để đo lường khoảng cách và độ tương đồng.

6.2.2 Phương pháp đo lường khoảng cách và độ tương đồng

Phương pháp đo lường khoảng cách
• Khoảng cách Euclidean
Phương pháp này dựa trên định lý Pythagoras trong không gian Euclidean.
Cho 2 điểm $A(x_1, y_1)$ và $B(x_2, y_2)$.
Trong không gian hai chiều., khoảng cách Euclidean giữa chúng là:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

.
Trong không gian nhiều chiều, khoảng cách Euclidean được mở rộng thành:

$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + ... + (z_n - z_m)^2}$$

.

• Khoảng cách Manhattan
Còn gọi là khoảng cách cung đường, là tổng của khoảng cách giữa các điểm theo mỗi chiều.
Trong không gian hai chiều:

$$d(A,B) = |x_2 - x_1| + |y_2 - y_1|$$

.
Trong không gian nhiều chiều:

$$d(A,B) = |x_2 - x_1| + |y_2 - y_1| + ... + |z_2 - z_1|$$

.

• Khoảng cách Cosine
Phương pháp này đo lường góc giữa hai vector trong không gian đa chiều. Cho hai vector A và B, khoảng cách cosine được tính bằng:

$$d(\mathbf{A},\mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

A ·**B** là tích vô hướng của hai vector và $\|\mathbf{A}\|$ và $\|\mathbf{B}\|$ là độ dài của từng vector.

Phương pháp đo lường độ tương đồng
• Độ tương đồng Cosine: Đã được đề cập trong phần trước, đo lường độ tương đồng hướng giữa hai vector trong không gian đa chiều.
Giá trị độ tương đồng cosine nằm trong khoảng [-1, 1], với 1 thể hiện hai vector cùng hướng, 0 thể hiện hai vector vuông góc nhau và -1 thể hiện hai vector đối diện nhau.

6.3 Thuật toán cơ bản

6.3.1 Cơ sở toán học

Với N phần tử, có những mức độ tương đồng (khoảng cách) giữa các cặp phần tử khác nhau. Những mức độ tương đồng này là dữ liệu cơ bản. (Trong những trường hợp không thể dễ dàng định lượng mức độ tương đồng, chẳng hạn như sự tương đồng giữa hai màu, thứ tự xếp hạng của các mức độ tương đồng là dữ liệu cơ bản).
Giả sử có N phần tử, các mức độ tương đồng có thể được sắp xếp theo thứ tự tăng dần nghiêm ngặt như sau:

$$s_{i_1k_1} \leqslant s_{i_2k_2} \leqslant \cdots \leqslant s_{i_Mk_M} \tag{12-21}$$

Ở đây, $s_{i_1 k_1}$ là mức độ tương đồng nhỏ nhất trong số M mức độ tương đồng. Chỉ số dưới biểu thị cặp phần tử ít tương đồng nhất — tức là, các phần tử có xếp hạng 1 trong thứ tự mức độ tương đồng. Các chỉ số khác cũng được diễn giải theo cách tương tự.

Chúng ta muốn tìm một cấu hình q-chiều của N phần tử sao cho các khoảng cách giữa các cặp phần tử khớp với thứ tự trong (12-21). Nếu các khoảng cách được bố trí theo thứ tự tương ứng, thì sẽ có một sự khớp hoàn hảo khi:

$$d^{(q)}_{i_1 k_1} \geqslant d^{(q)}_{i_2 k_2} \geqslant \cdots \geqslant d^{(q)}_{i_M k_M} \tag{12-22}$$

Nghĩa là, thứ tự giảm dần của các khoảng cách trong không gian q-chiều hoàn toàn tương tự với thứ tự tăng dần của các mức độ tương đồng ban đầu. Miễn là thứ tự trong (12-22) được bảo toàn, độ lớn của các khoảng cách không quan trọng.

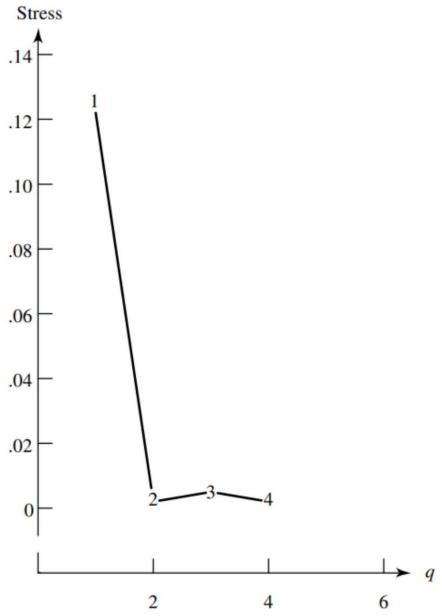
Với một giá trị q cho trước, có thể không tìm được một cấu hình của các điểm mà các khoảng cách theo cặp tuân theo thứ tự ban đầu của các mức độ tương đồng. Kruskal đề xuất một phép đo mức độ mà một biểu diễn hình học không đạt được sự khớp hoàn hảo. Phép đo này, gọi là **stress**, được định nghĩa là:

$$\text{Stress}^{(q)} = \sqrt{\frac{\sum\sum_{i<k}\left(d^{(q)}_{ik} - d'_{ik}\right)^2}{\sum\sum_{i<k}\left(d^{(q)}_{ik}\right)^2}} \tag{12-23}$$

Các số d'_{ik} trong công thức stress là các số thoả mãn (12-22); tức là, chúng có mối quan hệ đơn điệu với các mức độ tương đồng. Các giá trị này không phải là khoảng cách theo nghĩa rằng chúng thoả mãn các thuộc tính khoảng cách thông thường. Chúng chỉ đơn giản là các số tham chiếu được sử dụng để đánh giá mức độ không tuân thủ tính đơn điệu của các giá trị quan sát được.
Ý tưởng là tìm một biểu diễn của các phần tử dưới dạng các điểm trong không gian q-chiều sao cho stress nhỏ nhất có thể. Kruskal đề nghị Stress được giải thích như sau:

Stress	Goodness of fit
20%	Poor
10%	Fair
5%	Good
2.5%	Excellent
0%	Perfect

Bảng 8: Mức độ phù hợp các mức



Hình 6.3: Minh họa điểm "khuỷu tay"

Mức độ phù hợp đề cập đến mối quan hệ đơn điệu giữa các điểm tương đồng và khoảng cách cuối cùng.
• Với mỗi q, cấu hình sẽ dẫn đến theo giá trị Stress nhỏ nhất mà nó có thể đạt được.
• Khi q tăng, Stress nhỏ nhất trong phạm vi sai số làm tròn, sẽ giảm và bằng 0 đối với $q = N$ - 1.
• Bắt đầu với $q = 1$, có thể xây dựng đồ thị của các số Stress(q) này so với q. Giá trị của q mà biểu đồ này bắt đầu chững lại có thể được chọn là giá trị "tốt nhất" của kích thước. Từ đó ta tìm được điểm "khuỷu tay" trên biểu đồ số chiều Stress

6.3.2 Các bước thực hiện

Bước 1
• Với N phần tử ta sẽ thu được $M = N(N$ - $1)/2$ điểm tương đồng của mỗi cặp phần tử.
• Việc cần làm là tìm khoảng cách tương ứng với mỗi cặp phần tử ở trên rồi sắp xếp chúng lại từ lớn nhất đến nhỏ nhất.
• Nếu khoảng cách không tính toán được thì sắp xếp phải được chỉ định.

Bước 2
Sử dụng cấu hình thử nghiệm trong q-chiều, xác định khoảng cách $d_{ik}^{(q)}$ và d'_{ik} thỏa mãn phép đo (12-22) và phép đo Stress(12-23)

Bước 3
Sử dụng d'_{ik}, di chuyển các điểm xung quanh để có được cấu hình cải tiến, Một cấu hình mới sẽ có $d_{ik}^{(q)}$ với d'_{ik} mới và giá trị Stress nhỏ hơn. Quá trình sẽ được lặp đi lặp lại cho đến khi đạt được biểu diễn tốt nhất (Stress tối thiểu).

Bước 4
Vẽ đồ thị Stress(q) nhỏ nhất so với q và chọn số chiều tốt nhất là q từ việc kiểm tra biểu đồ này.

Chú thích:
Giả định rằng các giá trị tương đồng lúc đầu là đối xứng ($s_{ik} = s_{ki}$), không có ràng buộc và không có quan sát nào bị thiếu.

6.4 Ứng dụng trong giảm chiều dữ liệu

6.4.1 Giới thiệu

Một trong những ứng dụng lớn nhất của chia tỷ lệ đa chiều là dùng trong giảm chiều dữ liệu. Khi đó, chia tỷ lệ đa chiều (MDS) được chia làm 2 loại:
• Classic MDS
• Non - Metric MDS

6.4.2 Classic MDS

Trong phương pháp Classic MDS ta sẽ sử dụng khoảng cách Euclidean giữa 2 điểm A và B.

$$d(A, B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Vấn đề: Chỉ có khoảng cách, làm sao để có thể biểu diễn các đối tượng trên tọa độ. Trong phương pháp này, ta giả định các khoảng cách đều là Euclidean.
Classic MDS, được giới thiệu lần đầu bởi Torgerson (1952) giúp giải quyết bài toán này. Trong phương pháp này, ta giả định các khoảng cách đều là Euclidean.
Các bước tiến hành
• **Bước 1:** Lập ma trận bình phương $D^2 = [d_{ij}^2]$ (Bình phương từng giá trị trong ma trận khoảng cách ban đầu)
• **Bước 2:** Áp dụng định tâm kép. Ta tính ma trận $B = -1/2 * C * D^2 * C$ (Ma trận C được tính bằng $C = I - n^{-1} * J_n$ với n là số lượng đối tượng, I là ma trận đơn vị kích thước nxn và J_n là ma trận toàn số kích thước nxn. Quá trình này gọi là quá trình định tâm kép.
• **Bước 3:** Tìm ra **m** lớn nhất giá trị riêng λ_1, λ_2, λ_3, ...,λ_m và các vector riêng tương ứng e_1, e_2, e_3, ...,e_m của ma trận B, ở đây **m** là số chiều mong muốn cho kết quả đầu ra.
• **Bước 4:** Xây dựng ma trận **X** từ các giá trị riêng và vector riêng đã tìm được: $X = E_m * \Lambda_m^{1/2}$, trong đó E_m là ma trận các vector riêng kích thước **m x m** và Λ_m là ma trận đường chéo của **m** giá trị riêng của ma trận **B**

6.4.3 Non - Metric MDS

Non - Metric MDS là một biến thể của MDS, khi không thể đo khoảng cách chính xác giữa các điểm dữ liệu. Trong Non - Metric MDS, chúng ta chỉ quan tâm đến thứ tự các tương đối của các khoảng cách thay vì giá trị chính xác của chúng. Điều này giúp giảm bớt áp lực trong việc thu thập dữ liệu, vì chỉ cần biết thứ tự tương đối của khoảng cách thay vì các giá trị cụ thể.

Mục đích: Tìm ra các tọa độ của các điểm trong không gian p chiều, sao cho có sự tương thích tốt giữa các sự tương đồng quan sát và khoảng cách giữa các điểm.
Chia tỉ lệ phi số liệu tìm mối quan hệ không tham số đồng biến giữa các sự không giống nhau trong ma trận các mục và các khoảng cách Euclidean giữa các mục, cũng như vị trí của từng mục

trong không gian thấp chiều. Mối quan hệ này thường được tìm thấy bằng cách sử dụng hồi quy isolotic: giả sử x là các vector các độ gần gũi, $f(x)$ là biến đổi đồng biến của x, và d là khoảng cách giữa các điểm; sau đó, ta phải tìm tọa độ sao cho chúng tối thiểu hóa "STRESS":

$$STRESS = \sqrt{\frac{\sum(f(x) - d)^2}{\sum d^2}}$$

Có nhiều các biến thể của hàm STRESS này, các chương trình MDS tự động tối thiểu hóa STRESS để có được kết quả tốt nhất.
Giá trị của hàm STRESS cho thấy mức độ phù hợp của MDS, giá trị nhỏ cho thấy giải pháp tốt, giá trị càng lớn cho thấy càng sai số lớn.
Kruskal (1964) đã cung cấp một số giá trị để diễn giải giá trị của STRESS với mức độ phù hợp của nó:

Stress	Goodness of fit
20%	Poor
10%	Fair
5%	Good
2.5%	Excellent
0%	Perfect

Bảng 9: Mức độ phù hợp các mức

Chú ý
Nhân tố cốt lõi của thuật toán chia tỷ lệ đa chiều phi số liệu là quá trình tối ưu hóa hai chiều. Trước tiên, ta phải tìm ra biến đổi đồng biến tối ưu của các độ gần gũi. Tiếp theo, các điểm của một cấu hình phải được sắp xếp một cách tối ưu, để các khoảng cách của chúng phù hợp càng chính xác càng tốt với các độ gần gũi đã được chia tỷ lệ.
Các bước cơ bản trong thuật toán Non - Metric:
• **Bước 1:** Tìm một cấu hình ngẫu nhiên của các điểm, ví dụ bằng cách lấy mẫu từ phân phối chuẩn.
• **Bước 2:** Tính toán các khoảng cách d giữa các điểm.
• **Bước 3:** Tìm biến đổi đồng biến tối ưu của các độ gần gũi để thu được dữ liệu được điều chỉnh tối ưu $f(x)$.
• **Bước 4:** Tối thiểu hóa STRESS giữa dữ liệu đã được điều chỉnh tối ưu và các khoảng cách bằng cách tìm một cấu hình mới của các điểm.
• **Bước 5:** So sánh STRESS với một tiêu chí. Nếu STRESS đủ nhỏ thì thoát khỏi thuật toán, nếu không thì quay lại **Bước 2**

7 Tổng kết

Trong báo cáo này, nhóm đã trình bày một cách chi tiết các phương phápvà thuật toán phân cụm dữ liệu, một công cụ quan trọng trong lĩnh vực khoahọc dữ liệu. Báo cáo không chỉ giới thiệu về các phương pháp phân cụm cơbản như K-means và phân cụm phân cấp (Hierarchical Clustering), mà còn mởrộng thảo luận đến các mô hình thống kê phức tạp hơn như Gaussian MixtureModels (GMM).Các phân tích này được hỗ trợ bởi các công thức toán học chính xác và các ví dụ thực tiễn, cho phép người đọc có cái nhìn sâu sắc về cách thức ứng dụng các thuật toán này trong thực tế.

Một điểm nhấn trong báo cáo là việc áp dụng lý thuyết vào các bài toán thực tế, đặc biệt là trong lĩnh vực phân tích khách hàng và sinh học, nơi phân cụm dữ liệu đóng vai trò trung tâm trong việc phát hiện và phân tích các mẫu hành vi. Những ứng dụng này chứng minh rằng phân cụm dữ liệu không chỉ là một công cụ lý thuyết mà còn có giá trị thiết thực cao.

Báo cáo cũng nhấn mạnh đến sự cần thiết của việc lựa chọn đúng phương pháp phân cụm dựa trên bản chất của dữ liệu và mục tiêu của bài toán phân tích. Sự khác biệt giữa các phương pháp, từ phân cụm kết tụ đến phân cụm dựa trên mô hình, mở ra nhiều lựa chọn cho nhà khoa học dữ liệu trong việc tối ưu hóa kết quả phân tích