Clustering

1. TỔNG QUAN VỀ PHÂN CỤM

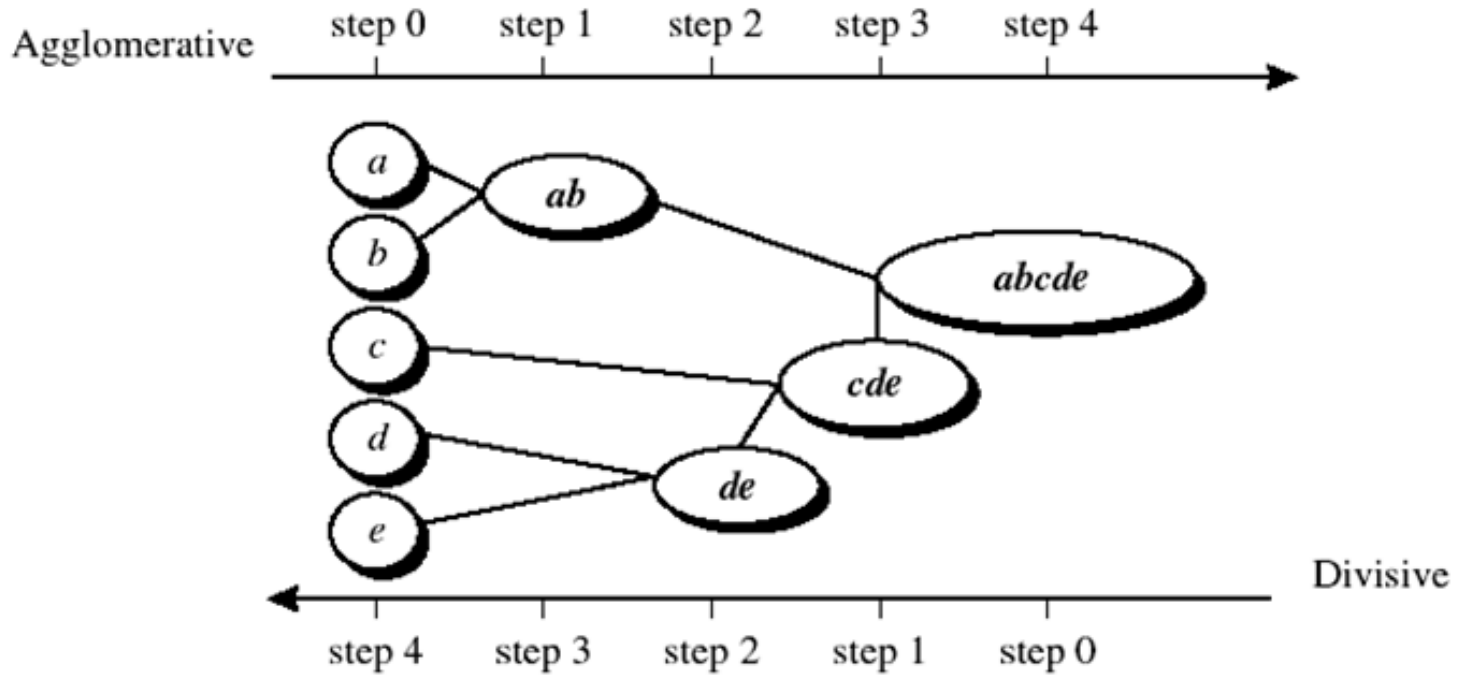
- Định nghĩa: là quá trình phân chia 1 tập dữ liệu ban đầu thành các cụm dữ liệu thỏa mãn:
  - Các đối tượng trong 1 cụm "tương tự" nhau.
  - Các đối tượng khác cụm thì "không tương tự" nhau.
- Giải quyết vấn đề tìm kiếm, phát hiện các cụm, các mẫu dữ liệu trong 1 tập hợp các dữ liệu ban đầu không có nhãn.

1. TỔNG QUAN VỀ PHÂN CỤM

- Các phương pháp phân cụm điển hình:
  - Phân cụm phân hoạch.
  - Phân cụm phân cấp.
  - Phân cụm dựa trên lý thuyết đồ thị.
  - Phân cụm theo hàm tối ưu.
  - Phân cụm dựa trên mật độ.
  - Phân cụm dựa trên lưới.
  - Phân cụm dựa trên mô hình.
  - Phân cụm có ràng buộc

2.1. Tổng quan về HAC

- Khái niệm: HAC là thuật toán phân cụm không giám sát(không cần biết số cụm cần phân vào) nhưng phải cung cấp điều kiện dừng.



Agglomerative and divisive hierarchical clustering on data objects $\{a, b, c, d, e\}$.

2.1. Tổng quan về HAC

- Sau khi phân cụm ta được đồ thị ***dendro*** sau:

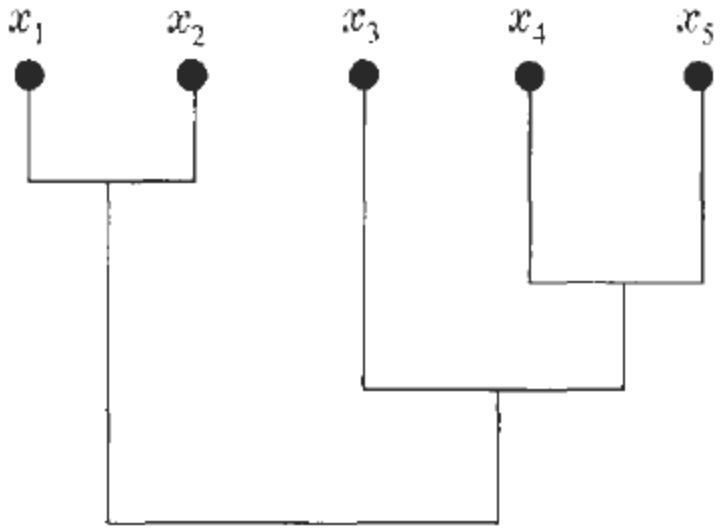
$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4,x_5\}\}$

$\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$

$\{\{x_1,x_2,x_3,x_4,x_5\}\}$



2.1. Tổng quan về HAC

- Thuật toán HAC có 2 phương pháp:

✓ Agglomerative: Đi từ dãy các phần tử và ở mỗi bước gom nhóm các cặp phân vùng(có thể là một phần tử hay là một nhóm có nhiều phần tử đã gom nhóm vào) có khoảng cách giữa 2 phần tử là gần nhau nhất. Cứ như thế cho đến khi chỉ còn lại 1 nhóm.

✓ Divisive: Cách này đi ngược lại với cách làm trên.

2.1. Tổng quan về HAC

- Vấn đề kiểu dữ liệu/đối tượng được gom cụm
  - Ma trận dữ liệu (data matrix)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Với n: số đối tượng (objects)

p: số biến/thuộc tính (variables/attributes)

2.1. Tổng quan về HAC

- Vấn đề kiểu dữ liệu/đối tượng được gom cụm
  - Ma trận khác biệt (dissimilarity matrix):

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

d(i, j) là khoảng cách giữa đối tượng i và j; thể hiện sự khác biệt giữa đối tượng i và j;

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

$d(i,j) \leq d(i,k) + d(k,j)$;

2.1. Tổng quan về HAC

- Khoảng cách(độ khác biệt) giữa 2 đối tượng i và j: được tính bởi độ đo khoảng cách Eulide.

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

- Ví dụ: Có 6 đối tượng A, B, C, D, E, F

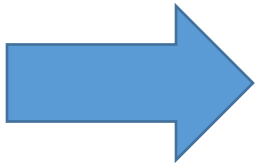
Stt	Object Name	X1	X2
1	A	1.0	1.0
2	B	1.5	1.5
3	C	5.0	5.0
4	D	3.0	4.0
5	E	4.0	4.0
6	F	3.0	3.5

2.1. Tổng quan về HAC

Stt	Object Name	X1	X2
1	A	1.0	1.0
2	B	1.5	1.5
3	C	5.0	5.0
4	D	3.0	4.0
5	E	4.0	4.0
6	F	3.0	3.5

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$\begin{bmatrix} 0 & & & & \\ d(B,A) & 0 & & & \\ d(C,A) & d(C,B) & 0 & & \\ : & : & : & & \\ d(F,A) & d(F,B) & \dots & \dots & 0 \end{bmatrix}$$



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

2.1. Tổng quan về HAC

- Khoảng cách(độ khác biệt) giữa hai cụm:

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

Cách lựa chọn các giá trị ai, aj, b và c cho ta các cách phép đo độ tương đồng d(Ci,Cj) khác nhau: *single link, complete link, weighted pair group method average(WPGMA), unweighted pair group method average(UPGMA), unweighted pair group method centroid(UPGMC)*

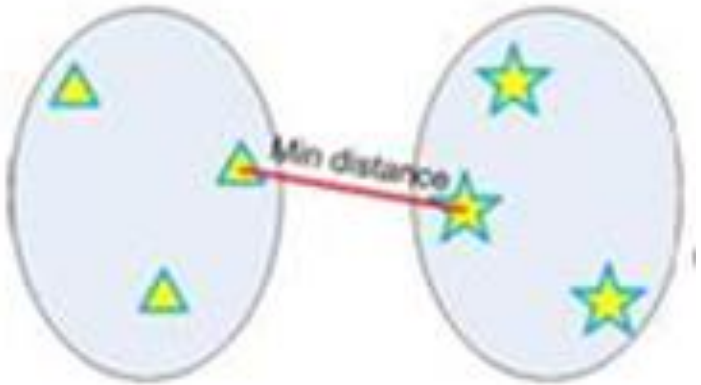
2.1. Tổng quan về HAC

- Single Link: Với a_i=1/2, a_j=1/2, b=0,c=-1/2

→ là khoảng cách nhỏ nhất giữa hai đối tượng thuộc về 2 cụm.

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

Trong đó: C_i,C_j: là 2 đối tượng thuộc cụm C_q



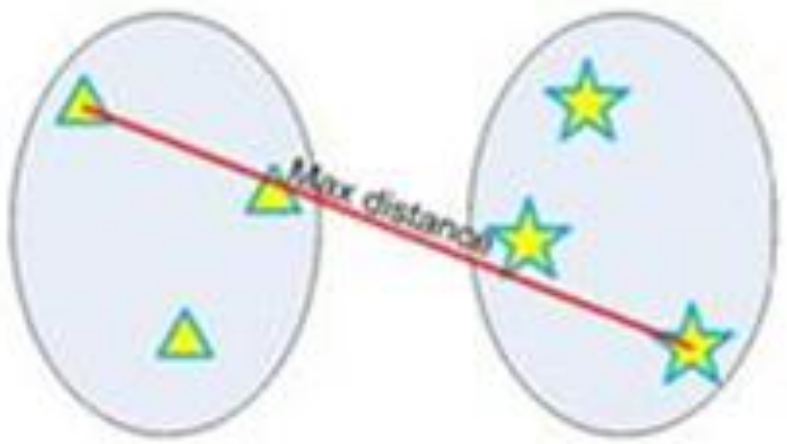
2.1. Tổng quan về HAC

- Complete Link: Với a_i=1/2, a_j=1/2, b=0,c=1/2

→ Là khoảng cách xa nhất giữa hai đối tượng thuộc về 2 cụm.

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

Trong đó: C_i,C_j: là 2 đối tượng thuộc cụm C_q



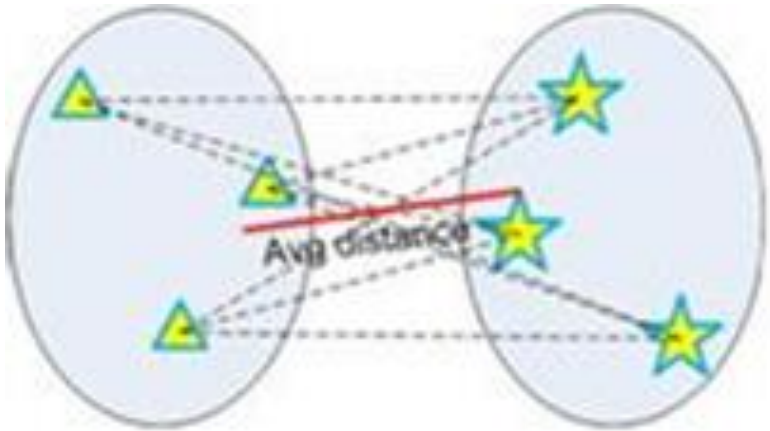
2.1. Tổng quan về HAC

- WPGMA : $a_i = a_j = 1/2$, b = 0, và c = 0

→ Là trung bình khoảng cách giữa các đối tượng trong hai cụm đó.

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

Trong đó: Cq, Cs: là hai cụm.



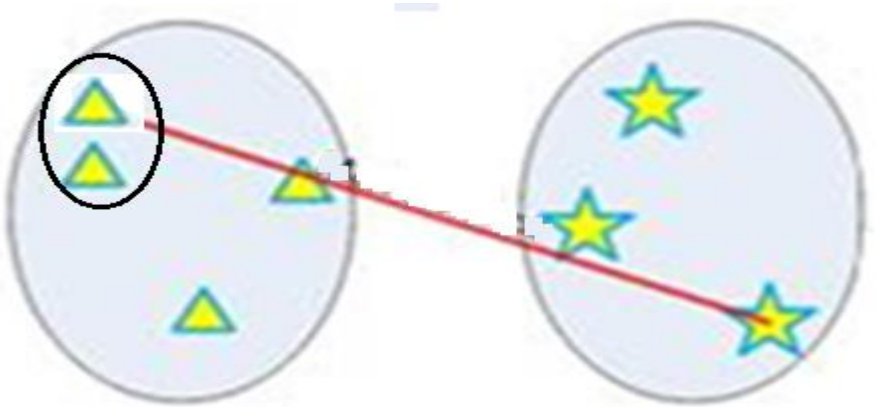
2.1. Tổng quan về HAC

- UPGMA : $a_i = \frac{n_i}{n_i + n_j}$; $a_j = \frac{n_j}{n_i + n_j}$; $b = 0; c = 0$

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

Trong đó: Cq(Ci, Cj), Cs: là hai cụm.

ni, nj lần lượt là số phần tử của Ci, Cj



2.1. Tổng quan về HAC

- UPGMA :

- Gọi mq là tâm của cụm Cq;

$$m_q = \frac{1}{n_q} \sum_{x \in C_q} x$$

- Khi đó, khoảng cách giữa hai tâm của cụm Cq và Cs là:

$$d_{qs} = \|m_q - m_s\|^2$$

2.1. Tổng quan về HAC

- WPGMC: Với $a_i=a_j=1/2$, b=-1/4,c=0

Thì:

$$d_{qs} = \frac{1}{2}d_{is} + \frac{1}{2}d_{js} - \frac{1}{4}d_{ij}$$

Chú ý: $d_{qs} \leq \min(d_{is}, d_{js})$

2.1. Tổng quan về HAC

- So sánh các kết quả tính toán khi áp dụng các cách phép đo độ tương đồng trên:

	SL	CL	WPGMA	UPGMA	WPGMC	UPGMC
$\Re_0$	0	0	0	0	0	0
$\Re_1$	1	1	1	1	1	1
$\Re_2$	1.5	1.5	1.5	1.5	1.5	1.5
$\Re_3$	2	3	2.5	2.5	2.25	2.25
$\Re_4$	16	37	25.75	27.5	24.69	26.46

2.2. Thuật toán phân cụm phân rã

- Quá trình ngược lại với thuật toán Agglomerative, ban đầu chúng ta xem tất cả các đối tượng thuộc cùng 1 cụm, sau đó tiến hành phân thành 2 cụm con. Quá trình này được thực hiện cho đến khi mỗi cụm chỉ còn 1 đối tượng.

2.2. Thuật toán phân cụm phân rã

- Thuật toán:

B1: Khởi tạo:

1.1. Choose $\Re_0 = \{X\}$ as the initial clustering.
1.2. $t = 0$

B2: Lặp lại B2:

2.1. $t = t + 1$
2.2. For $i = 1$ to t

* 2.2.1 Among all possible pairs of clusters (C_r, C_s) that form a partition of $C_{t-1,i}$, find the pair $(C^1_{t-1,i}, C^2_{t-1,i})$ that gives the maximum value for g.

Next i

2.2. Thuật toán phân cụm phân rã

2.3. From the t pairs defined in the previous step choose the one that maximizes g. Suppose that this is $(C_{t-1,j}^1, C_{t-1,j}^2)$.

2.4. The new clustering is

$$\mathfrak{R}_t = (\mathfrak{R}_{t-1} - \{C_{t-1,j}\}) \cup \{C_{t-1,j}^1, C_{t-1,j}^2\}$$

2.5. Relabel the clusters of $\mathfrak{R}_t$.

Lặp đến khi tất cả các đối tượng được phân vào các cụm đơn phân biệt.

2.2. Thuật toán phân cụm phân rã

- Ví dụ: Cho ma trận khác biệt của 5 đối tượng A, B, C, D, E như sau:

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

B1: Khởi tạo R0={A, B, C, D, E};

B2: t=1,i=1:

- d(A,(B,C,D,E)) = (d(A,B)+ d(A,C)+ d(A,D)+ d(A,E))/4=(1+2+26+37)/4=16.5
- d(B,(A,C,D,E)) = (1+3+25+36)/4=16.25
- d(C,(A,B,D,E)) = (2+3+16+25)/4 = 11.5
- d(D,(A,B,C,E)) = (26+25+16+1.5)/4 = 17.13
- d(E,(A,B,C,D)) = (37+36+25+1.5)/4 = 24.9

2.2. Thuật toán phân cụm phân rã

Ta có, d(E,(A,B,C,D)) lớn nhất

→ $C^1_{0,1}$={E}; $C^2_{0,1}$={A,B,C,D}

→ R_1={(E),(A,B,C,D)}

- Lặp lại B2: t=2:

- i=1: Duyệt hết tất cả các cặp có thể có trong cụm i=1. Cụm 1 không có cặp đối tượng nào→ chuyển sang xét cụm 2
- i=2: Duyệt hết tất cả các cặp có thể có trong cụm i=2:

d(A,(B,C,D))=(1+2+26)/3=9.3

d(B,(A,C,D))=(1+3+25)/3=9.3

d(C,(A,B,D))=(2+3+25)/3=10

d(D,(A,B,C))=(26+25+16)/3=16.75

2.2. Thuật toán phân cụm phân rã

- Ta có d(D,(A,B,C)) lớn nhất → chọn cụm thứ 2 (A,B,C,D) ở vòng lặp 1 để tiếp tục phân rã thành:

→ $C^1_{1,2}$={D}; $C^2_{1,2}$={A,B,C}

→ R_2={(E),(D),(A,B,C)}

2.2. Thuật toán phân cụm phân rã

- Lặp lại B2: t=3:

- i=1: Duyệt hết tất cả các cặp có thể có trong cụm i=1. Cụm 1 không có cặp đối tượng nào→ chuyển sang xét cụm 2
- i=2: tương tự cụm 1→ chuyển sang cum 3
- i=3: Duyệt hết tất cả các cặp có thể có trong cụm i=3:

d(A,(B,C))=(1+2)/2=1.5

d(B,(A,C))=(1+3)/2=2

d(C,(A,B))=(2+3)/2= 2.5

2.2. Thuật toán phân cụm phân rã

- Ta có d(C,(A,B)) lớn nhất → chọn cụm thứ 3 (A,B,C) ở vòng lặp 2 để tiếp tục phân rã thành:

→ $C^1_{2,3}$={C}; $C^2_{2,3}$={A,B}

→ R_3={(E),(D),(C),(A,B)}

- Lặp lại B2: t=4: làm tương tự như trên ta có:

→ $C^1_{3,4}$={A}; $C^2_{3,4}$={B}

→R_4={(E),(D),(C),(A),(B)}

Khi các đối tượng được phân thành các cụm đơn phân biệt thuật toán dừng.

2.3. Thuật toán phân cụm tích tụ(GAS)

- Thuật toán chung.

B1: Khởi tạo.

1.1. Choose $\Re_0 = \{C_i = \{x_i\},\ i = 1, \ldots, N\}$
1.2. $t = 0$.

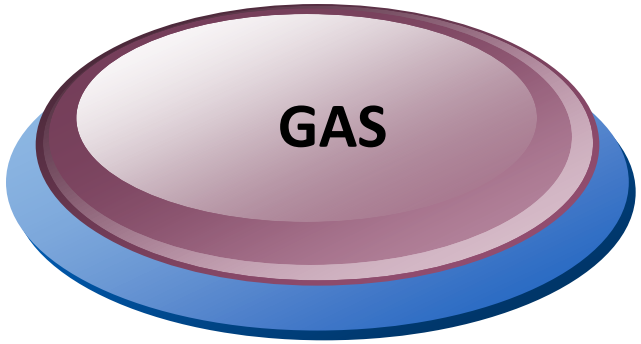
B2: Lặp lại B2:

2.1. $t = t + 1$
2.2. Among all possible pairs of clusters (C_r, C_s) in $\Re_{t-1}$ find the one, say (C_i, C_j), such that

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a similarity function} \end{cases}$$

Lặp đến khi tất cả các đối tượng được phân vào 1 cụm duy nhất.

2.3. Thuật toán phân cụm tích tụ(GAS)



2.3.1. Dựa trên lý thuyết ma trận(MUAS):
MATRIX UPDATING ALGORITHMIC SCHEME

2.3.2. Dựa trên lý thuyết đồ thị(GTAS): GRAPH THEORY BASED ALGORITHMIC

2.3.1. Phân cụm dựa trên lý thuyết ma trận

- Input: ma trận khác biệt NxN; P0=P(X)
- Thuật toán:

B1: Bước khởi tạo

1.1. $\mathfrak{R}_0 = \{\{x_i\}, i = 1, \ldots, N\}$
1.2. $P_0 = P(X)$.
1.3. $t = 0$

B2: Lặp lại B2

2.1. $t = t + 1$
2.2. Find C_i, C_j such that $d(C_i, C_j) = \min_{r,s=1,\ldots,N,\ r \neq s} d(C_r, C_s)$.
2.3. Merge C_i, C_j into a single cluster C_q and form $\mathfrak{R}_t = (\mathfrak{R}_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$
2.4. Define the proximity matrix P_t from P_{t-1} as explained in the text.

Lặp đến khi tất cả các đối tượng được phân về cùng một cụm.

- Output: Một cụm gồm tất cả các đối tượng.

2.3.1. Phân cụm dựa trên lý thuyết ma trận

- Ví dụ:

$$P_0 = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 26 & 37 \\ 1 & 0 & 3 & 25 & 36 \\ 2 & 3 & 0 & 16 & 25 \\ 26 & 25 & 16 & 0 & 1.5 \\ 37 & 36 & 25 & 1.5 & 0 \end{bmatrix} \end{array}$$

B1: Có 5 cụm:R_0= {(A),(B),(C),(D),(E)}

B2: Ta có cụm A và B là gần nhau nhất (khoảng cách là 1) vì vậy ta nhóm A và B vào 1 cluster (A,B).

Tính lại ma trận khoảng cách Dist. Chú ý là khoảng cách giữa các cụm không được nhóm (C,D,E) không thay đổi.

2.3.1. Phân cụm dựa trên lý thuyết ma trận

- Tính lại khoảng cách từ cụm (A, B) đến các cụm khác:

$d((A,B),C) = \min(d_{AC}, d_{BC}) = \min(2,3) = 2$

$d((A,B),D) = \min(d_{AD}, d_{BD}) = \min(26,25) = 25$

$d((A,B),E) = \min(d_{AE}, d_{BE}) = \min(37,36) = 36$

→ $R_1 = \{(A,B),C,D,E\}$

$$P_0 = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 26 & 37 \\ 1 & 0 & 3 & 25 & 36 \\ 2 & 3 & 0 & 16 & 25 \\ 26 & 25 & 16 & 0 & 1.5 \\ 37 & 36 & 25 & 1.5 & 0 \end{bmatrix} \end{array} \Rightarrow P_1 = \begin{array}{c} \\ A,B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccc} A,B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 2 & 25 & 36 \\ 2 & 0 & 16 & 25 \\ 25 & 16 & 0 & 1.5 \\ 36 & 25 & 1.5 & 0 \end{bmatrix} \end{array}$$

2.3.1. Phân cụm dựa trên lý thuyết ma trận

- Ta có: cụm E và D là gần nhau nhất (khoảng cách là 1.5) vì vậy ta nhóm E và D vào 1 cụm (E,D).
- Tính lại khoảng cách từ cụm (E, D) đến các cụm khác ((A,B), C):

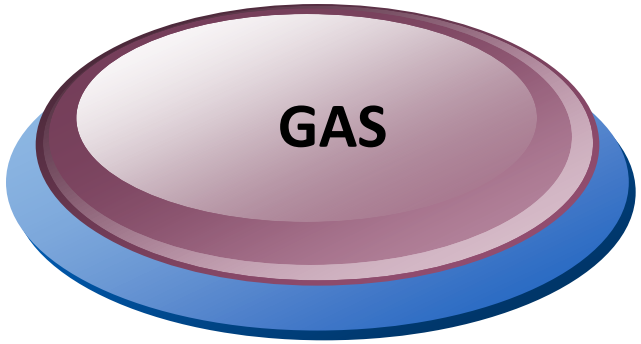
$d((E,D),(A,B)) = \min(d_{EAB}, d_{DAB}) = \min(36,25) = 25$

$d((E,D),C) = \min(d_{EC}, d_{DC}) = \min(25,16) = 16$

→ $R_2=\{(A,B),C,(D,E)\}$

$$P_1 = \begin{array}{c} \\ A,B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccc} A,B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 2 & 25 & 36 \\ 2 & 0 & 16 & 25 \\ 25 & 16 & 0 & 1.5 \\ 36 & 25 & 1.5 & 0 \end{bmatrix} \end{array} \rightarrow P_2 = \begin{array}{c} \\ A,B \\ C \\ D,E \end{array} \begin{array}{c} \begin{array}{ccc} A,B & C & D,E \end{array} \\ \begin{bmatrix} 0 & 2 & 25 \\ 2 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix} \end{array}$$

2.3. Thuật toán phân cụm tích tụ



2.3.1. Dựa trên lý thuyết ma trận(MUAS):
MATRIX UPDATING ALGORITHMIC SCHEME

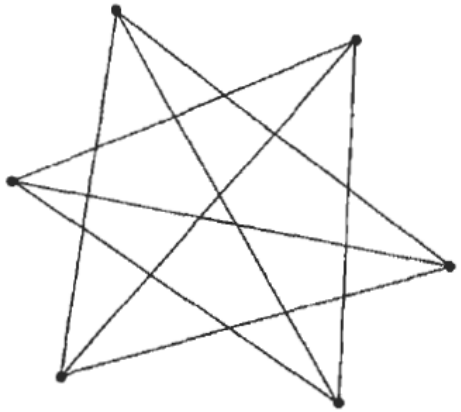
2.3.2. Dựa trên lý thuyết đồ thị(GTAS): GRAPH THEORY BASED ALGORITHMIC

2.3.2. Phân cụm dựa trên lý thuyết đồ thị

- Một số khái niệm:
  - Node degree: là số nguyên k lớn nhất sao cho mỗi node có ít nhất k đường đi.
  - Node connectivity: là số nguyên k lớn nhất sao cho 2 node bất kỳ có ít nhất k đường đi giữa chúng mà không có node chung nào.
  - Edge connectivity: là số nguyên k lớn nhất sao cho 2 cạnh bất kỳ có ít nhất k đường đi giữa chúng mà không có cạnh chung nào.

2.3.2. Phân cụm dựa trên lý thuyết đồ thị

- Đồ thị được gọi là *connected* nếu thỏa ít nhất một trong ba tính chất trên.(gọi là thuộc tính h(k))

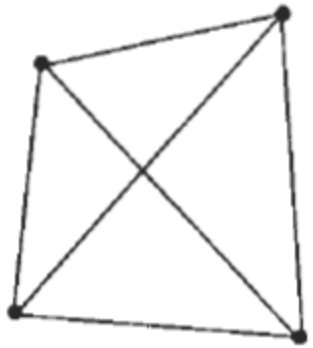


Node connectivity : 3

Edge connectivity : 3

Node degree : 3

- Đồ thị được gọi là complete nếu mỗi đỉnh đều có cạnh nối với tất cả các đỉnh còn lại.



2.3.2. Phân cụm dựa trên lý thuyết đồ thị

- Thuật toán thực hiện theo các bước của thuật toán phân cụm phân cấp chung(GAS), chỉ khác ở bước 2.2, thay đổi $g(C_i,C_j)$ bởi $g_{h(k)}(C_i,C_i)$.

$$g_{h(k)}(C_i, C_j) = \begin{cases} \min_{r,s} g_{h(k)}(C_r, C_s), & \text{for dissimilarity functions} \\ \max_{r,s} g_{h(k)}(C_r, C_s), & \text{for similarity functions} \end{cases}$$

- $g=h_{(k)}$ được xác định như sau:

$$g_{h(k)}(C_r, C_s) = \min_{x_u \in C_r, x_v \in C_s} \{d(x_u, x_v) \equiv a : \text{ the } G(a) \text{ subgraph defined by } C_r \cup C_s \text{ is (a) connected and either (b1) has the property } h(k) \text{ or (b2) is complete}\}$$

G(a): đồ thị mới G được tạo ra khi thực hiện nối 2 đỉnh có khoảng cách(độ khác biệt) là a.

2.3.2. Phân cụm dựa trên lý thuyết đồ thị

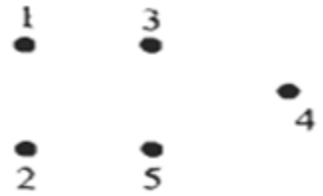
- Ví dụ: Cho ma trận tương đồng sau:

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

- Dựa vào ma trận trên ta lần lượt vẽ được đường đi giữa các điểm.

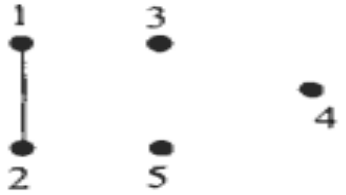
2.3.2. Phân cụm dựa trên lý thuyết đồ thị

- Đầu tiên, ta có G(0):



G(0)

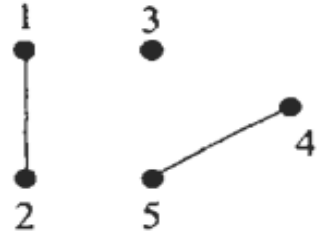
- $\{x_1\},\{x_2\}$: có $g_{h(k)}(x_1,x_2)=1.2$ nhỏ nhất nên ta có đồ thị G(1.2), G(1.2) thỏa complete → chấp nhận G(1.2) → $R_1=\{\{x_1,x_2\}, \{x_3\} ,\{x_4\} ,\{x_5\}\}$



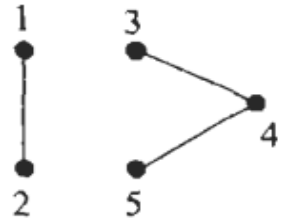
G(1.2)

2.3.2. Phân cụm dựa trên lý thuyết đồ thị

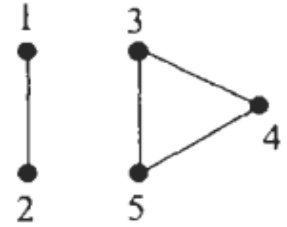
- Lặp lại việc tìm các khoảng cách nhỏ nhất dựa trên thuật toán Single Link, ta lần lượt vẽ được các đồ thị sau:



G(1.5)



G(1.8)



G(2.0)

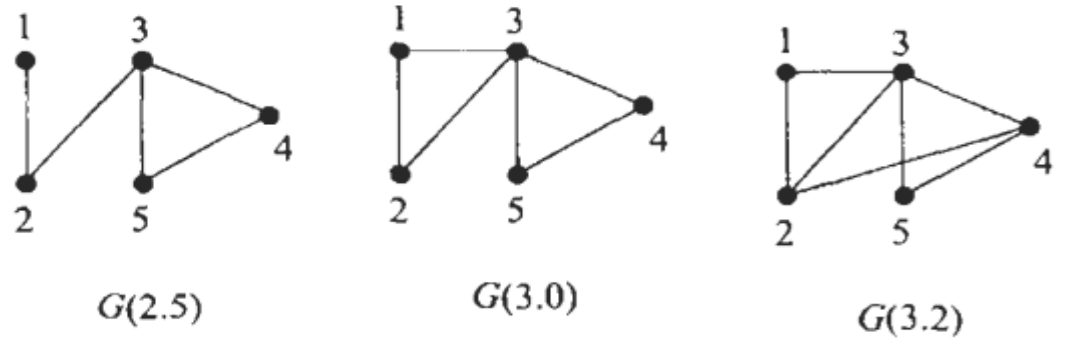
- G(1.5) thỏa là đồ thị connected

→R_2={{x_1,x_2}, {x_3} ,{x_4,x_5}}

- G(1.8) không thỏa là complete hay connected → G(1.8) bị bỏ qua.
- G(2.0) thỏa là đồ thị connected → G(2,0) được chọn →R_3={{x_1,x_2}, {x_3,x_4,x_5}}

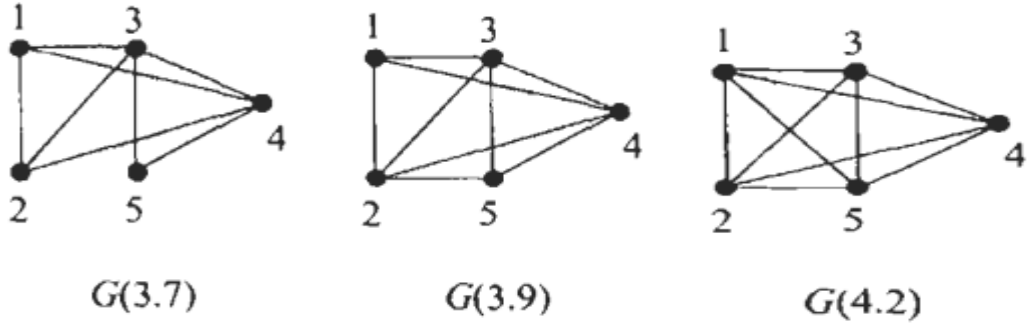
2.3.2. Phân cụm dựa trên lý thuyết đồ thị

- Tương tự như trên nối lần lượt hai đỉnh(chưa được nối)mà khoảng cách giữa chúng là bé nhất trong tất cả các cặp chúng ta xét.



- Nối x_3—x_2=2.5: tạo ra G(2.5) không connected và không complete → G(2.5) bỏ qua.
- Nối x_3—x_1=3: tạo ra G(3) không connected và không complete → G(3) bỏ qua.
- Nối x_4—x_2=3.2 tạo ra G(3.2) không connected và không complete → G(2.5) bỏ qua.

2.3.2. Phân cụm dựa trên lý thuyết đồ thị

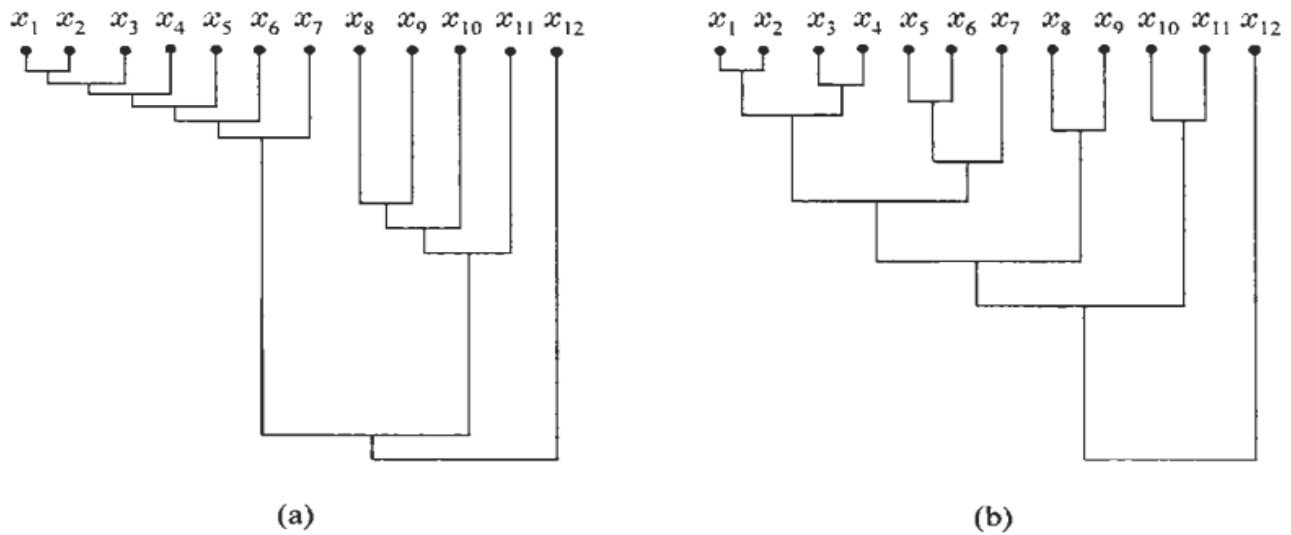


- Nối x_1—x_4=3.7: tạo ra G(3.7) không connected và không complete → G(3.7) bỏ qua.
- Nối x_2—x_5=3.9: tạo ra G(3.9) không connected và không complete → G(3.9) bỏ qua.
- Nối x_5—x_1=4.2: tạo ra G(4.2) complete → G(4.2) được chọn.

→ R_4={x_1,x_2,x_3,x_4,x_5} các đối tượng đã thành 1 cụm→ thuật toán kết thúc

2.4. Lựa chọn số phân cụm như thế nào?

- Thuật toán phân cụm không cần thiết phải tạo ra phân cấp N cụm, việc phân cụm kết thúc khi sự phù hợp nhất của dữ liệu đạt được, phù hợp với tiêu chí.



(a) Đồ thị dendro cho thấy có 2 cụm chính trong dữ liệu

(b) Đồ thị dendro cho thấy có 1 cụm chính trong dữ liệu

2.4. Lựa chọn số phân cụm như thế nào?

- **Cách 1:** có sự tác động bên ngoài. Nó yêu cầu xác định giá trị các tham số cụ thể đối với người sử dụng.
- Sử dụng hàm h(C): phép đo độ "self-similarity"(tự tương đồng)

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

hoặc:

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

Đối với ma trận khoảng cách:

$$h_3(C) = \frac{1}{2n_C} \sum_{x \in C} \sum_{y \in C} d(x, y)$$

n_c: tổng số phần tử của C

2.4. Lựa chọn số phân cụm như thế nào?

- Thuật toán kết thúc khi:

$$\exists C_j \in \Re_{t+1} : h(C_j) > \theta$$

θ: ngưỡng thích hợp để chấp nhận h(C)

R_t : là cụm cuối cùng nếu tồn tại a cụm C trong R_{t+1} với độ khác biệt giữa các vectơ $h(C) > \theta$

- θ thường do người sử dùng đặt ra, đôi lúc được xác định theo công thức :

$$\theta = \mu + \lambda\sigma$$

μ: trung bình khoảng cách giữa hai vecto trong X

σ: phương sai

λ: tham số người dùng xác định

3. Vấn đề tối ưu hóa

- Hầu hết các đề án về clustering dựa trên sự tối ưu đều dựa vào 1 hàm J (cost function), với nhiều cách tính toán khác nhau. Hàm J là sự tối ưu các dữ liệu, và nó là hàm tham số để tìm ra được vector tham biến θ . Trong hầu hết các cách tính thì số lượng các cluster đều được giả định là biết trước.

- Mục tiêu của vấn đề là dự đoán được θ , là đặc trưng tốt nhất của các cluster cơ bản. Vector θ phụ thuộc nhiều vào hình dạng của các cluster.

3. Vấn đề tối ưu hóa

- Cluster đơn giản, do đó có thể bỏ qua các tham số vẫn có thể dễ dàng tìm ra m điểm, trong l chiều, tương ứng với các đặc trưng của cluster.

- Với dạng cluster hình tròn, ta dùng m hypersheres C(Cj,Rj), với Cj là tâm và Rj là bán kính cầu hypersheres để xác định các đặc trưng của cluster.