TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

Tài liệu bài giảng:

KHAI THÁC DỮ LIỆU – IS252

Chương 7:

Gom cụm

ThS. Dương Phi Long – Email: longdp@uit.edu.vn

Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**
  - Supervision: Dữ liệu huấn luyện (quan sát, đo lường, v.v.) được kèm theo nhãn lớp
  - Dữ liệu mới được phân lớp dựa trên tập huấn luyện
- **Unsupervised learning (phân cụm)**
  - Nhãn lớp của dữ liệu huấn luyện không xác định
  - Đưa ra một tập hợp các phép đo, quan sát, ... với mục đích thiết lập sự tồn tại của các lớp hoặc cụm trong dữ liệu

NỘI DUNG BÀI HỌC

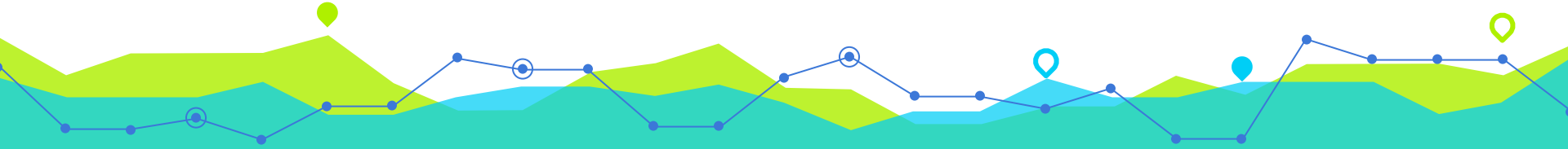
01 Tổng quan về gom cụm dữ liệu

02 Phương pháp phân hoạch

03 Phương pháp phân cấp

04 Phương pháp dựa trên mật độ

05 Phương pháp dựa trên mô hình

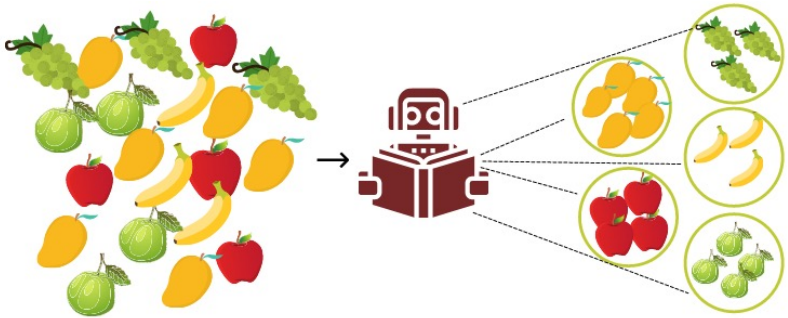


1 Tổng quan về Gom cụm dữ liệu

1. Gom cụm là gì
2. Tiêu chuẩn gom cụm
3. Độ đo khoảng cách
4. Yêu cầu và thách thức
5. Một số phương pháp gom cụm

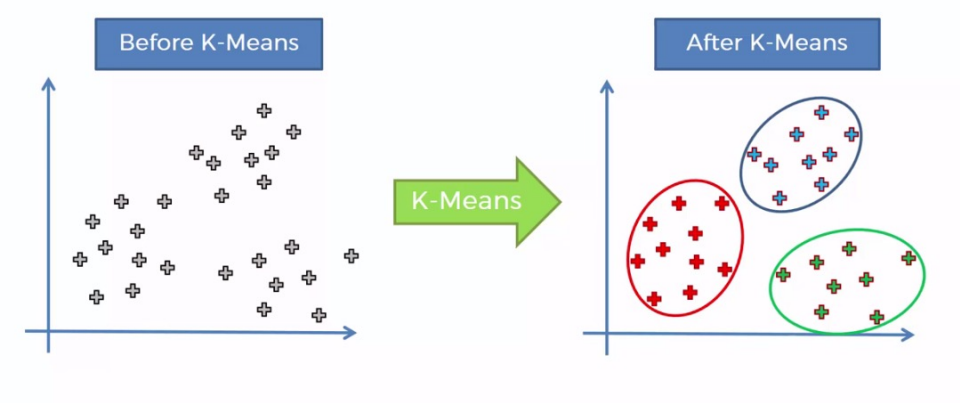
1. Gom cụm dữ liệu

- **Cluster (cụm/ nhóm/ lớp):** tập hợp các đối tượng dữ liệu
  - Tương đồng hoặc liên quan với nhau trong cùng 1 nhóm
  - Không tương đồng hoặc không liên quan với các đối tượng trong các nhóm khác
- **Gom cụm:** Tìm sự tương đồng giữa dữ liệu theo các đặc điểm được tìm thấy trong dữ liệu và nhóm các đối tượng dữ liệu tương đồng thành các cụm



1. Gom cụm dữ liệu

- Cho CSDL D ={t_1, t_2, …, t_n} và số nguyên k.
  Gom cụm là bài toán xác định ánh xạ f: D → {1, …, k}
  sao cho mỗi t_i được gán vào một nhóm K_j với $1 \leq j \leq k$
- **Học không giám sát:** không có lớp được xác định trước (nghĩa là học bằng cách quan sát so với học bằng ví dụ: được giám sát)

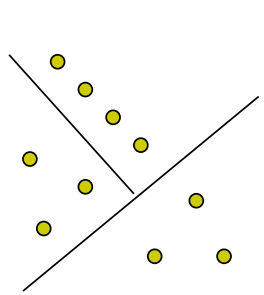


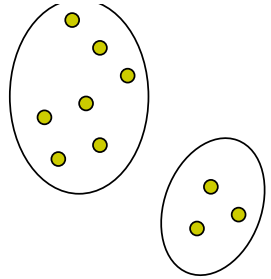
1. Gom cụm dữ liệu

- **Các ứng dụng tiêu biểu**
  - Là một công cụ độc lập để hiểu sâu hơn về phân bố dữ liệu
  - Là một bước tiền xử lý cho các thuật toán khác
- **VD:**
  - Tiếp thị: khám phá các nhóm khách hàng phân biệt trong CSDL mua hàng
  - Y sinh: Phân tích sự tương đồng và gom nhóm gen có cùng chức năng
  - Hoạch định thành phố: nhận dạng các nhóm nhà cửa theo loại nhà, giá trị và vị trí địa lý.

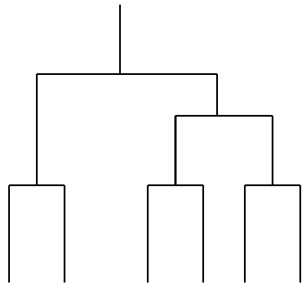
1. Gom cụm dữ liệu

- **Cách biểu diễn các cụm**
  - Các đường ranh giới
  - Các khối cầu
  - Theo xác suất
  - Sơ đồ hình cây
  - …





	1	2	3
I1	0.5	0.2	0.3
I2			
…			
In			



1. Gom cụm dữ liệu

Có bao nhiêu cụm?

6 cụm

2 cụm

4 cụm

Gom cụm vs. Phân lớp

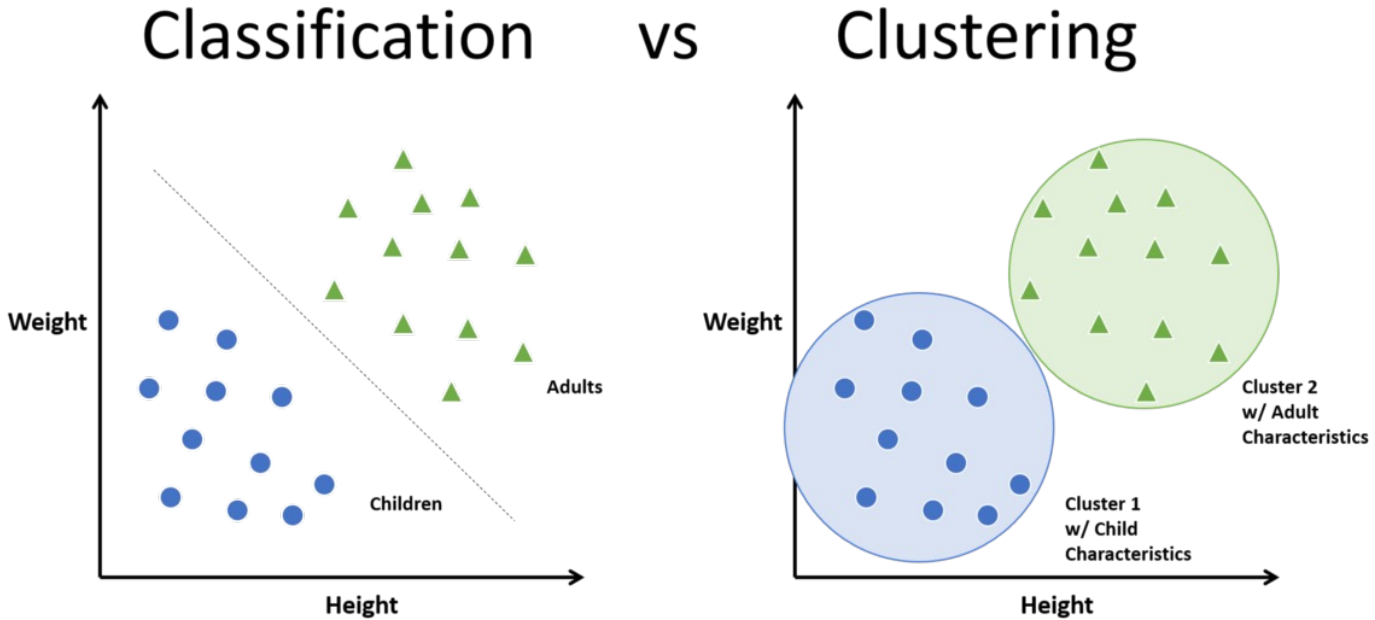
Gom cụm truyền thống

- Mục tiêu: xác định các cụm đối tượng tương tự. Các cụm được phát hiện
- Bộ dữ liệu gồm các thuộc tính
- Không giám sát (nhãn lớp phải học)
- Đánh giá tính tương đồng, "hàm khoảng cách" là rất quan trọng, bởi vì các cụm được phát hiện dựa trên khoảng cách / mật độ.

Phân lớp

- Mục tiêu là dự đoán các lớp từ các thuộc tính/giá trị thuộc tính của đối tượng. Các lớp được xác định trước
- Bộ dữ liệu gồm các thuộc tính và một nhãn lớp
- Được giám sát (nhãn lớp đã biết)
- Bộ phân lớp được học từ các tập ví dụ đã phân lớp
- Bộ phân lớp cần có độ chính xác cao

Gom cụm vs. Phân lớp



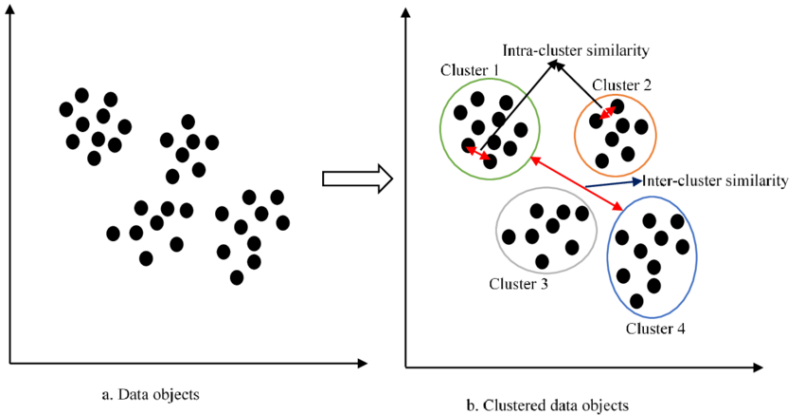
Tìm phương pháp để dự đoán lớp của mẫu mới từ các mẫu đã được gán nhãn lớp trước

Tìm các cụm/ nhóm "tự nhiên" của các mẫu chưa được gán nhãn

2. Tiêu chuẩn gom cụm

- **Phương pháp gom cụm tốt:** tạo ra các cụm có chất lượng
  - Giữa các đối tượng trong cùng 1 cụm sự giống nhau cao (intra-class)
  - Giữa các cụm sự giống nhau thấp (inter-class)

- Chất lượng của kết quả gom cụm phụ thuộc vào:
  - Độ đo sự tương tự
  - Thuật toán gom nhóm
  - Khả năng phát hiện một vài hay tất cả các mẫu bị che (hidden patterns)



3. Độ đo khoảng cách

- Thường được dùng để xác định sự khác nhau hay giống nhau giữa 2 đối tượng.
- **Khoảng cách Minkowski**

$$d(i,j) = \sqrt[q]{\left(\left|x_{i1} - x_{j1}\right|^q + \left|x_{i2} - x_{j2}\right|^q + \cdots + \left|x_{ip} - x_{jp}\right|^q\right)} \quad (1)$$

- $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ và j $= (x_{j1}, x_{j2}, \ldots, x_{jp})$: 2 đối tượng p-chiều
- q: số nguyên dương

3. Độ đo khoảng cách

- **Khoảng cách Manhattan (q=1)**

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \quad (2)$$

- **Khoảng cách Euclide (q=2)**

$$d(i,j) = \sqrt{\left(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2\right)} \quad (3)$$

- **Tính chất của độ đo khoảng cách:**

$$d(i,j) \geq 0 \qquad d(i,i) = 0 \qquad d(i,j) = d(j,i) \qquad d(i,j) \leq d(i,k) + d(k,j)$$

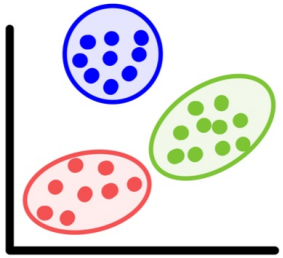
3. Độ đo khoảng cách

- Các kiểu dữ liệu khác nhau: yêu cầu độ đo tương đồng cũng sẽ khác nhau
  - Các biến tỷ lệ theo khoảng: Khoảng cách Euclide
  - Các biến nhị phân: Hệ số so khớp, hệ số Jaccard
  - Các biến tên, thứ tự, tỷ lệ: Khoảng cách Minkowski
  - Các biến dạng hỗn hợp: Công thức trọng lượng

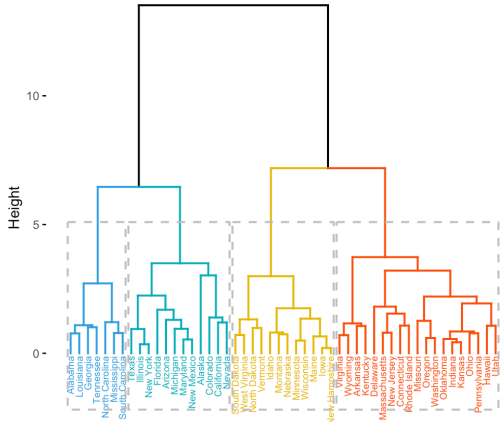
4. Yêu cầu và thách thức

- Khả năng mở rộng (Scalability): Phân cụm tất cả dữ liệu thay vì chỉ trên một số mẫu
- Khả năng xử lý với các loại thuộc tính khác nhau: Numerical, binary, categorical, ordinal, linked hay hỗn hợp
- Phân cụm dựa trên ràng buộc: Người dùng có thể cung cấp đầu vào về các ràng buộc
- Khả năng diễn giải và khả năng sử dụng
- Khác:
  - Khám phá các cụm với hình dạng tùy ý
  - Khả năng xử lý dữ liệu nhiễu
  - Phân cụm tăng dần và không nhạy cảm với thứ tự đầu vào
  - Số chiều lớn

5. Một số phương pháp gom cụm



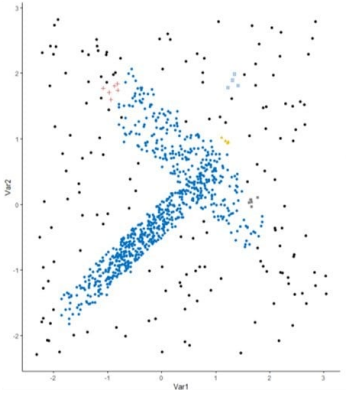
Partitioning approach



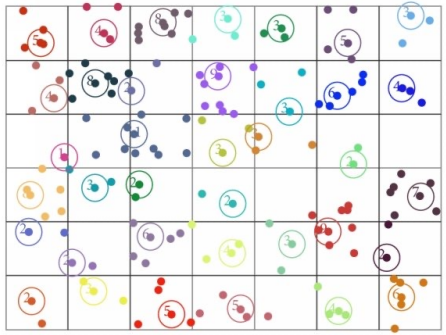
Hierarchical approach

- **Phương pháp phân hoạch (Partitioning approach):**
  - Xây dựng các phân vùng khác nhau và sau đó đánh giá chúng theo một số tiêu chí. VD: giảm thiểu sum of square errors
  - VD: k-means, k-medoids, CLARANS
- **Phương pháp phân cấp (Hierarchical approach):**
  - Tạo phân cấp theo thứ bậc của bộ dữ liệu (hoặc đối tượng)
  - VD: Diana, Agnes, BIRCH, CAMELEON

5. Một số phương pháp gom cụm



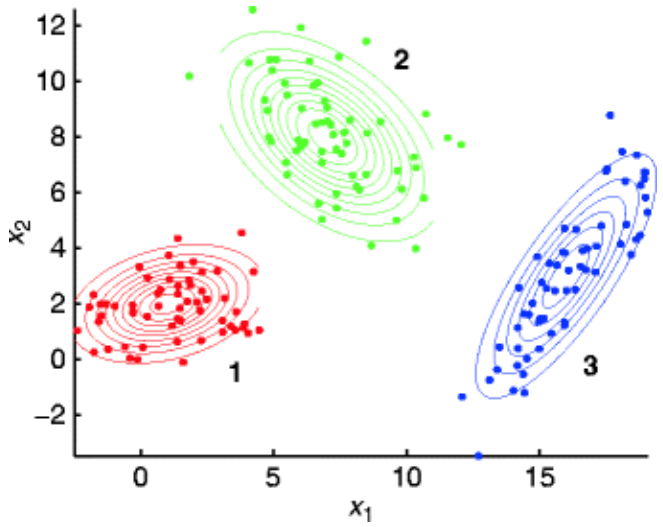
Density-based approach (k=5)



Grid-based approach (k=37)

- **Phương pháp dựa trên mật độ (Density-based approach):**
  - Dựa trên các chức năng kết nối và mật độ
  - VD: DBSCAN, OPTICS, DenClue
- **Phương pháp dựa trên lưới (Grid-based approach):**
  - Dựa trên cấu trúc chi tiết multiple-level
  - VD: STING, WaveCluster, CLIQUE

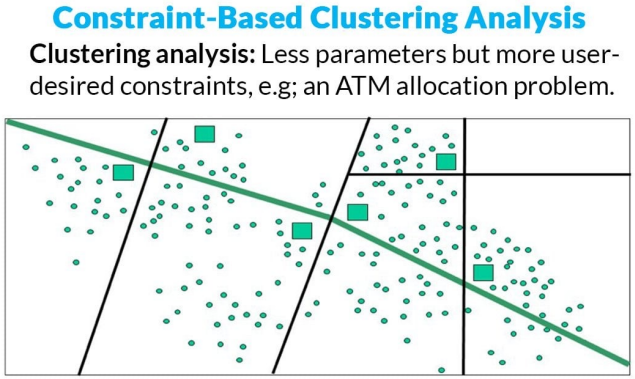
5. Một số phương pháp gom cụm

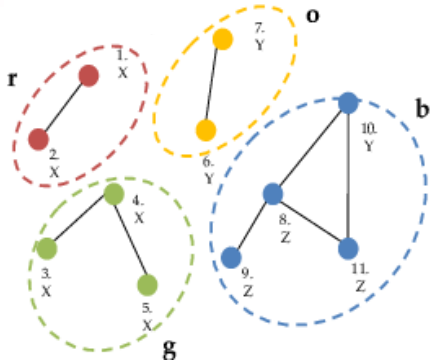


Model-based

- **Phương pháp dựa trên mô hình (Model-based):**
  - Một mô hình được đưa ra giả thuyết cho từng cụm và cố gắng tìm ra sự phù hợp nhất của mô hình đó
  - VD: EM, SOM, COBWEB
- **Phương pháp dựa trên tập phổ biến (Frequent pattern-based):**
  - Dựa trên phân tích các mẫu phổ biến
  - VD: p-Cluster

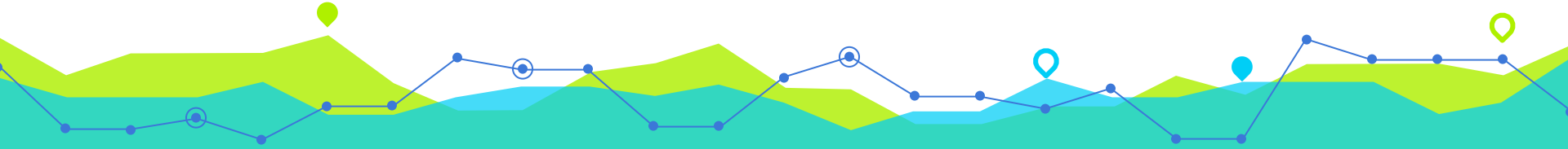
5. Một số phương pháp gom cụm





Link-based

- **Phương pháp dựa trên ràng buộc (User-guided or constraint-based):**
  - Bằng cách xem xét các ràng buộc do người dùng chỉ định
  - VD: COD (obstacles), constrained clustering
- **Phân cụm dựa trên liên kết (link-based):**
  - Các đối tượng thường được liên kết với nhau theo nhiều cách khác nhau
  - VD: SimRank, LinkClus

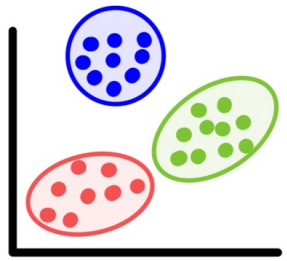


2 Phương pháp phân hoạch

1. Khái niệm cơ bản
2. Thuật toán K-means
3. Thuật toán K-medoids: PAM

1. Giới thiệu

- **Phương pháp phân hoạch (Partitioning approach):**
  - Phân hoạch CSDL D gồm n đối tượng thành k cụm $(k < n)$.
  - Đánh giá cụm theo một số tiêu chí phân hoạch đã chọn



VD: Sum of Squared Error – SSE nhỏ nhất

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x) \quad (4)$$

x: một điểm dữ liệu trong cụm C_i
m_i: trọng tâm (centroid) hoặc trung tâm (medoid) của cụm C_i
k: số cụm
$dist()$: khoảng cách Euclide

1. Giới thiệu

- **Một số thuật toán tiêu biểu:**
  - **k-means** (1): cụm được biểu diễn bằng trọng tâm cụm.
  - **k-medoids** (Partitioning Around Medoids - PAM) (2): cụm được biểu diễn bằng một mẫu dữ liệu nằm gần trung tâm của cụm.
  - **CLARANS** (3): kết hợp tìm kiếm ngẫu nhiên, tối ưu hóa cục bộ để cải tiến k-means (*)

(*): Tìm hiểu và semniar

(1) MacQueen, James B. "On the Asymptotic Behavior of k-means." Defense Technical Information Center 10 (1965).

(2) Rdusseeun, L. K. P. J., and P. Kaufman. "Clustering by means of medoids." Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland. Vol. 31. 1987.

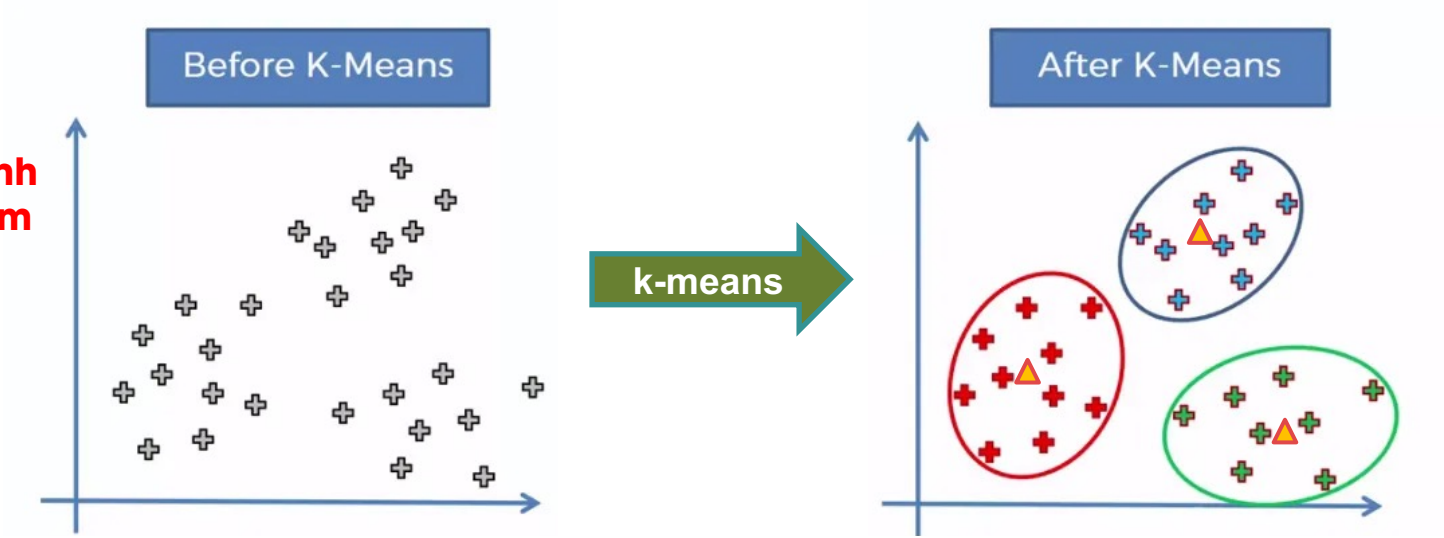
(3) Ng, Raymond T., and Jiawei Han. "CLARANS: A method for clustering objects for spatial data mining." IEEE transactions on knowledge and data engineering 14.5 (2002): 1003-1016.

2. Thuật toán k-means

Mỗi cụm được biểu diễn bằng trọng tâm (centroid) của dữ liệu trong cụm

Cần xác định trước số cụm

k = 3



2. Thuật toán k-means

Các bước thực hiện:

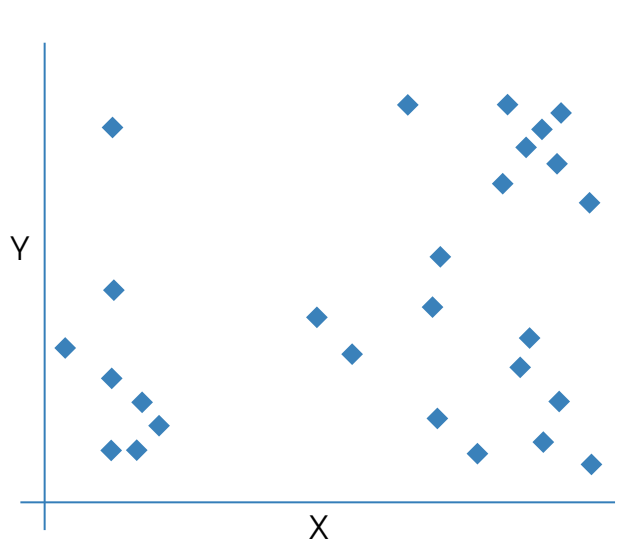
- **B1:** Khởi tạo k centroid ban đầu cho k cụm
- **B2:** Tính khoảng cách Euclide giữa các điểm với centroid.

  Phân cụm cho các điểm dữ liệu
- **B3:** Cập nhật các centroid
- **B4:** Nếu centroid không thay đổi (hoặc thỏa điều kiện dừng): Dừng

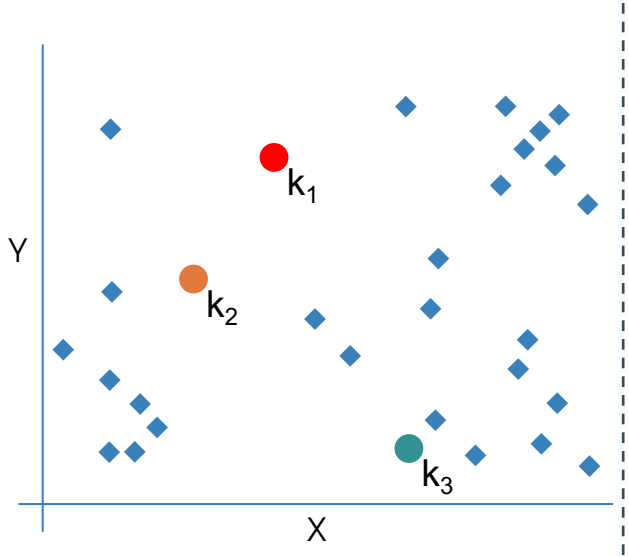
  Ngược lại: quay lại B2.

2. Thuật toán k-means

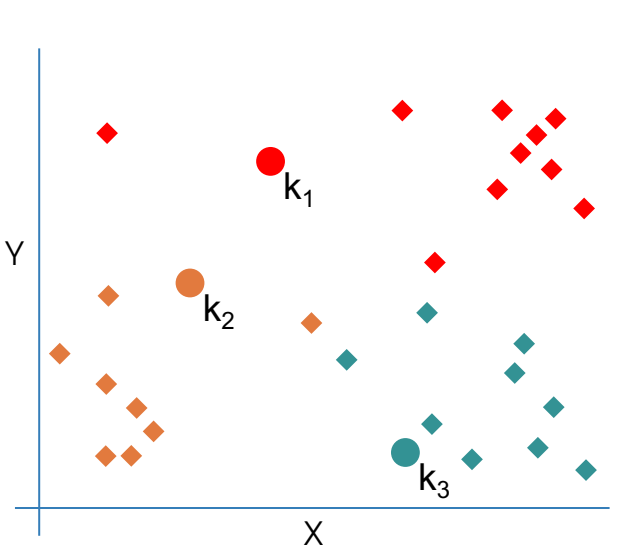
VD2: Minh họa thuật toán



a) Phân 3 cụm



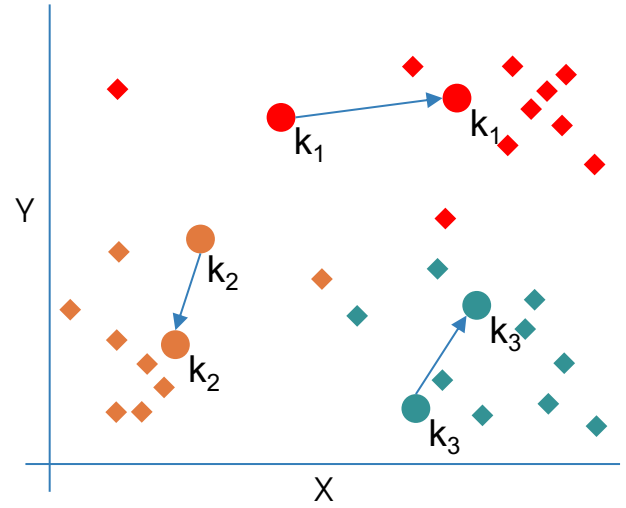
b) Chọn 3 trọng tâm bất kỳ: k_1, k_2, k_3



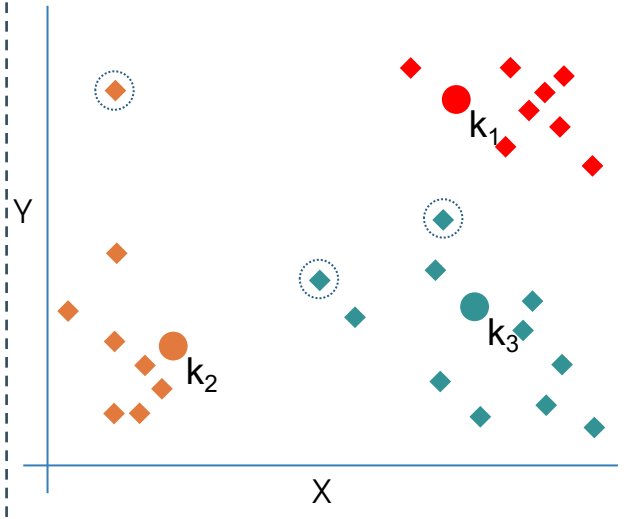
c) Phân cụm cho các điểm

2. Thuật toán k-means

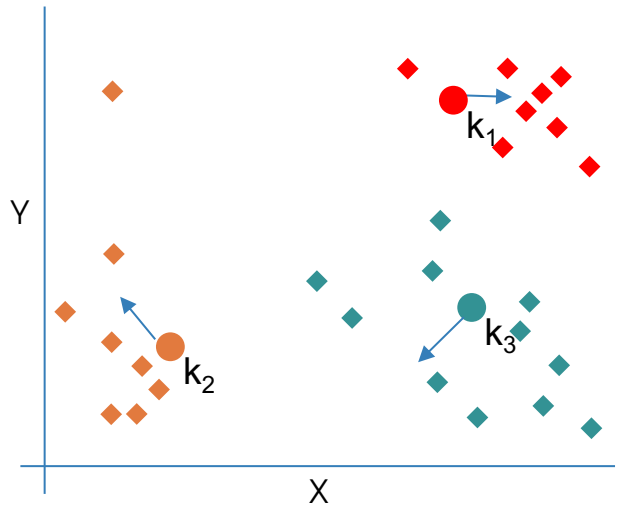
VD2: Minh họa thuật toán



d) Tính lại trọng tâm



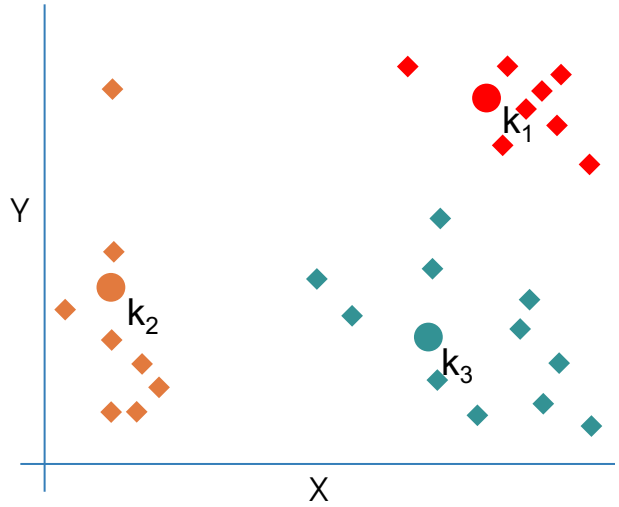
e) Phân cụm cho các điểm



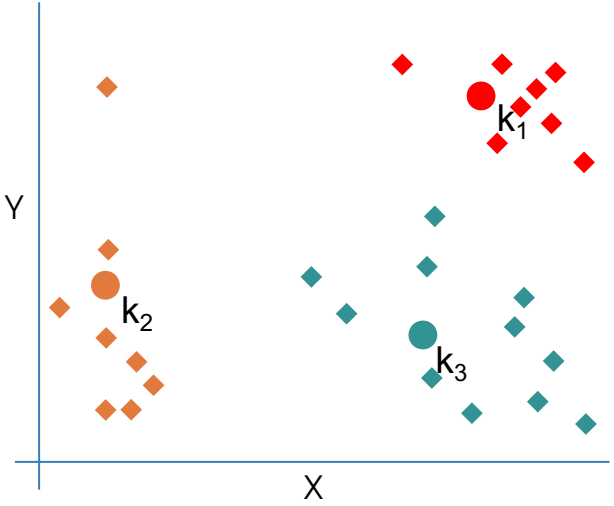
f) Tính lại trọng tâm

2. Thuật toán k-means

VD2: Minh họa thuật toán



g) Phân cụm cho các điểm



h) Dừng khi cụm không đổi

2. Thuật toán k-means

VD3: Cho tập dữ liệu sau:

1. Sử dụng thuật toán k-means để phân cụm với k = 2

   Cách 1 – Khởi tạo centroid

   Cách 2 – Sử dụng ma trận phân hoạch

2. Tính độ lỗi SSE với 2 cụm tìm được ở câu 1.

	x_{i1}	x_{i2}
x_1	1	3
x_2	1.5	3.2
x_3	1.3	2.8
x_4	3	1

2. Thuật toán k-means

VD3: *Cách 1 – Khởi tạo centroid*

- **Bước 1: Chọn centroid**
  - Chọn $\mathbf{X_1}(1, 3)$ là trọng tâm c_1 của C_1
  - Chọn $\mathbf{X_4}(3, 1)$ là trọng tâm c_2 của C_2

	x_{i1}	x_{i2}
x_1	1	3
x_2	1.5	3.2
x_3	1.3	2.8
x_4	3	1

2. Thuật toán k-means

VD3: *Cách 1 – Khởi tạo centroid*

- **Bước 2: Phân cụm cho các điểm với $c_1(1, 3)$ và $c_2(3, 1)$**

	x_{i1}	x_{i2}	$d(x_i, c_1)$	$d(x_i, c_2)$	Cụm
x_1	1	3	**0**		C_1
x_2	1.5	3.2	**0.54**	2.67	C_1
x_3	1.3	2.8	**0.36**	2.48	C_1
x_4	3	1		**0**	C_2

$$d(x_2, c_1) = \sqrt{(1.5-1)^2 + (3.2-3)^2} = \mathbf{0.54}$$

$$d(x_2, c_2) = \sqrt{(1.5-3)^2 + (3.2-1)^2} = 2.67$$

$$d(x_3, c_1) = \sqrt{(1.3-1)^2 + (2.8-3)^2} = \mathbf{0.36}$$

$$d(x_3, c_2) = \sqrt{(1.3-3)^2 + (2.8-1)^2} = 2.48$$

2. Thuật toán k-means

VD3: *Cách 1 – Khởi tạo centroid*

- **Bước 3: Cập nhật centroids**:

	x_{i1}	x_{i2}	Cụm
x_1	1	3	C_1
x_2	1.5	3.2	C_1
x_3	1.3	2.8	C_1
x_4	3	1	C_2

$$c_{11} = \frac{x_{11} + x_{21} + x_{31}}{3} = \frac{1 + 1.5 + 1.3}{3} = 1.27$$

$$c_{12} = \frac{x_{12} + x_{22} + x_{32}}{3} = \frac{3 + 3.2 + 2.8}{3} = 3$$

$$c_{21} = \frac{x_{41}}{1} = 3$$

$$c_{22} = \frac{x_{42}}{1} = 1$$

$\Rightarrow$ $c_1(1.27, 3)$ và $c_2(3, 1)$

2. Thuật toán k-means

VD3: *Cách 1 – Khởi tạo centroid*

- **Quay lại Bước 2: Phân cụm cho các điểm với $c_1(1.27, 3)$ và $c_2(3, 1)$**

	x_{i1}	x_{i2}	$d(x_i, c_1)$	$d(x_i, c_2)$	Cụm
x_1	1	3	**0.27**	2.8	C_1
x_2	1.5	3.2	**0.52**	2.67	C_1
x_3	1.3	2.8	**0.2**	2.48	C_1
x_4	3	1	2.64	**0**	C_2

Cụm không đổi: Dừng

Kết luận:

Cụm C_1: $\{x_1, x_2, x_3\}$

Cụm C_2: $\{x_4\}$

2. Thuật toán k-means

VD3: *Cách 2 – Sử dụng ma trận phân hoạch*

- **Bước 1: Khởi tạo ma trận phân hoạch M_0**

M_0	x_1	x_2	x_3	x_4
C_1	1	0	0	0
C_2	0	1	1	1

- **Bước 2: Tính trọng tâm của C_1, C_2**: $c_1(c_{11}, c_{12})$ và $c_2(c_{21}, c_{22})$

$$c_{11} = \frac{1}{1} = 1 \qquad c_{12} = \frac{3}{1} = 3$$

$$\Rightarrow c_1(1, 3) \text{ và } c_2(1.93, 2.23)$$

$$c_{21} = \frac{1.5 + 1.3 + 3}{3} = 1.93 \qquad c_{22} = \frac{3.2 + 2.8 + 1}{3} = 2.33$$

2. Thuật toán K-means

VD3: *Cách 2 – Sử dụng ma trận phân hoạch*

- **Bước 3: Phân cụm cho các điểm với c_1 (1, 3) và c_2 (1.93, 2.33)**

	x_{i1}	x_{i2}	$d(x_i, c_1)$	$d(x_i, c_2)$	Cụm
x_1	1	3	**0**	1.14	C_1
x_2	1.5	3.2	**0.54**	0.97	C_1
x_3	1.3	2.8	**0.36**	0.78	C_1
x_4	3	1	2.83	**1.70**	C_2

2. Thuật toán k-means

VD3: *Cách 2 – Sử dụng ma trận phân hoạch*

- **Bước 4: Cập nhật ma trận phân hoạch M_1**

M_1	x_1	x_2	x_3	x_4
C_1	1	1	1	0
C_2	0	0	0	1

- **Quay lại Bước 2: Tính trọng tâm của C_1, C_2:**

$$c_{11} = \frac{1 + 1.5 + 1.3}{3} = 1.27 \qquad c_{12} = \frac{3 + 3.2 + 2.8}{3} = 3$$

$\Rightarrow$ **c_1(1.27, 3) và c_2(3, 1)**

$$c_{21} = \frac{3}{1} = 3 \qquad c_{12} = \frac{1}{1} = 1$$

2. Thuật toán K-means

VD3: *Cách 2 – Sử dụng ma trận phân hoạch*

- **Bước 3: Phân cụm cho các điểm với c_1 (1.27, 3) và c_2 (3, 1)**

	x_{i1}	x_{i2}	$d(x_i, c_1)$	$d(x_i, c_2)$	Cụm
x_1	1	3	**0.27**	2.8	C_1
x_2	1.5	3.2	**0.52**	2.67	C_1
x_3	1.3	2.8	**0.2**	2.45	C_1
x_4	3	1	2.64	**0**	C_2

2. Thuật toán K-means

VD3: *Cách 2 – Sử dụng ma trận phân hoạch*

- **Bước 4: Cập nhật ma trận phân hoạch mới M_2:**

M_2	x_1	x_2	x_3	x_4
C_1	1	1	1	0
C_2	0	0	0	1

$M_2 = M_1$: **Dừng**

Kết luận:

Cụm C_1: $\{x_1, x_2, x_3\}$

Cụm C_2: $\{x_4\}$

2. Thuật toán k-means

VD3: 2. Tính SSE với 2 cụm C_1, C_2

	Cụm	$d(c_i, x)$
x_1	C_1	0.27
x_2	C_1	0.52
x_3	C_1	0.2
x_4	C_2	0

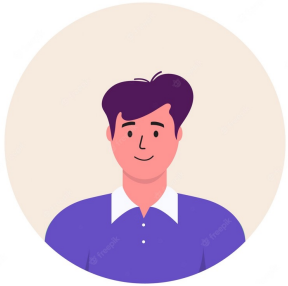
$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(c_i, x)$$

$$= (0.27)^2 + (0.52)^2 + (0.2)^2 + (0)^2$$

$$= 0.3833$$

2. Thuật toán k-means

- **Chuẩn hóa dữ liệu:**



Steven:

Age: 35

Income: 95K

No. of credit card: 3

Ricky:

Age: 41

Income: 215K

No. of credit card: 2

$$D(Steven, Ricky) = \sqrt{(35-41)^2 + (95-215)^2 + (3-2)^2}$$

- Các thuộc tính có miền giá trị khác nhau
- Các thuộc tính có giá trị lớn sẽ ảnh hưởng nhiều đến khoảng cách. VD: income

Cần chuẩn hóa giá trị thuộc tính

2. Thuật toán k-means

- **Chuẩn hóa dữ liệu:** Ánh xạ các giá trị về miền giá trị $[0., 1]$:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i} \quad (17)$$

- v_i :giá trị thực tế của thuộc tính i
- a_i: giá trị đã chuẩn hóa của thuộc tính i

2. Thuật toán k-means

- **Ưu điểm:**
  - Đơn giản, dễ hiểu, tương đối hiệu quả
  - Độ phức tạp: $O(tkn)$, với *n:* số mẫu dữ liệu, *k:* số cụm, t: số lần lặp (k, t << n).
  - Các đối tượng đều được gán vào các cụm
  - Thường đạt được tối ưu cục bộ

2. Thuật toán k-means

- **Nhược điểm:**
  - Áp dụng cho dữ liệu số.
  - Cần xác định trước số cụm k. Cách để xác định k (Hastie et al., 2009)
  - Phụ thuộc vào việc khởi tạo các cụm đầu tiên
  - Nhạy cảm với dữ liệu nhiễu, cá biệt
  - Không phù hợp để khám phá các cụm có kích thước, mật độ khác nhau hoặc có hình dạng không phải là hình cầu

T Hastie, R Tibshirani, JH Friedman, JH Friedman. "*The elements of statistical learning: data mining, inference, and prediction*". Vol. 2. New York: springer, 2009.

2. Thuật toán k-means

BT15

Cho tập dữ liệu

1. Chuẩn hóa dữ liệu về miền giá trị $[0, 1]$

2. Sử dụng thuật toán k-means để phân cụm khách hàng với k=2

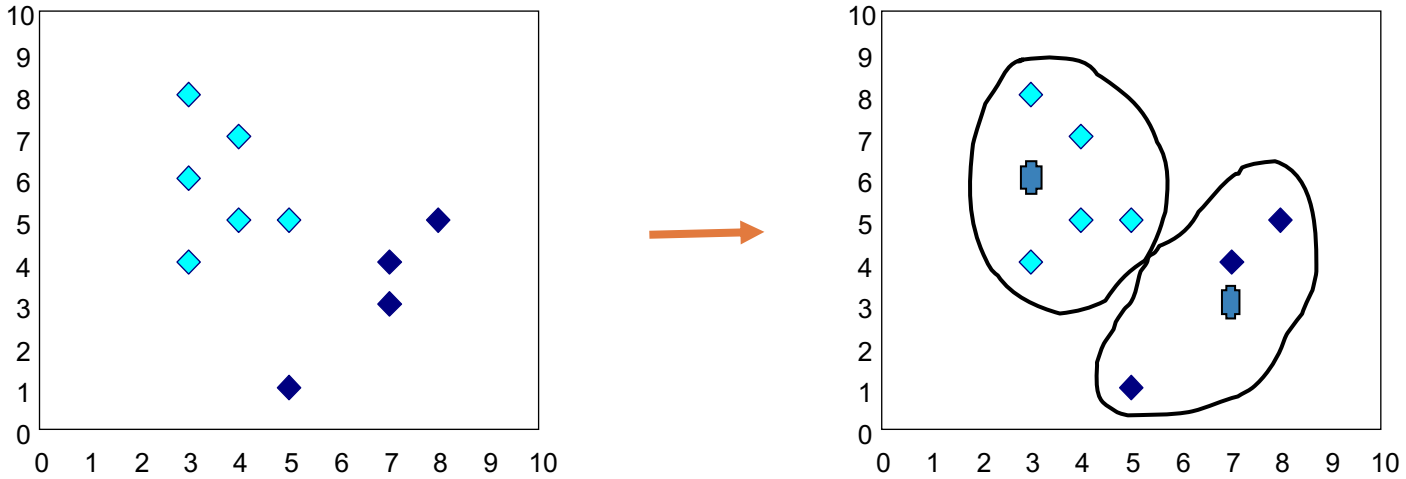
Yêu cầu:

- Nhóm 04 thành viên
- Thời gian: 20 phút
- Nộp bài trên Course

customer	age	income
John	47	23,100
Rachel	39	25,000
Hannah	70	80,000
Tom	66	67,000
Nellie	43	18,000
David	54	25,000

3. Thuật toán K-medoids: PAM

Cho số k, mỗi cụm được biểu diễn bằng một trong các đối tượng gần trung tâm cụm nhất (medoid)



3. Thuật toán K-medoids: PAM

- **B1:** Chọn ngẫu nhiên k đối tượng làm điểm trung tâm medoid ban đầu của k cụm
- **B2:** Gán từng đối tượng vào cụm có medoid gần nó nhất (dựa trên khoảng cách Manhattan). Tính cost cho việc phân cụm.

$$Cost = \sum_{i=1}^{k} \sum_{x_j \in C_i} dist(x_j, c_i) \quad (5)$$

- x_j: một điểm dữ liệu trong cụm C_i
- c_i: medoid của cụm C_i
- dist(): khoảng cách Manhattan

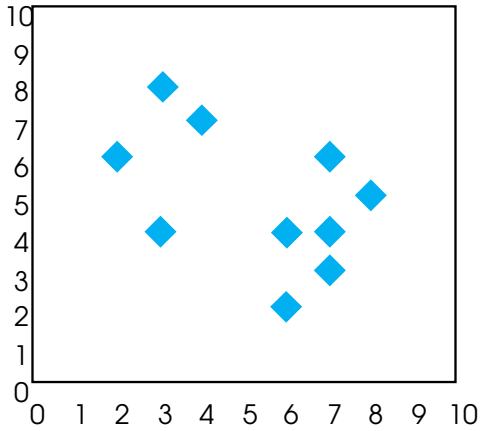
$$dist(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \quad (2)$$

3. Thuật toán K-medoids: PAM

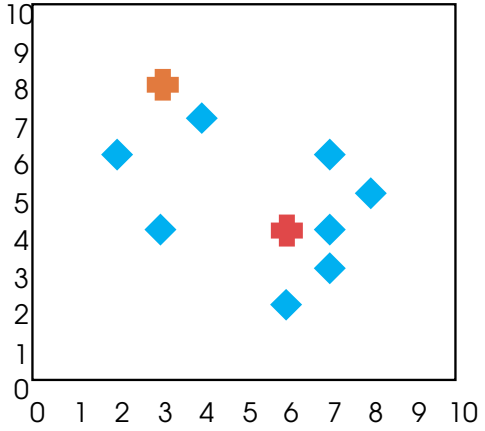
- **B3:** Với mỗi medoid, chọn 1 đối tượng bất kỳ, hoán đổi nó với medoid của cụm.
  - Nếu cost giảm so với trước đó thì quay lại B2.
  - Ngược lại, undo lại bước hoán đổi thực hiện tiếp tục B3 cho đến khi không còn có thay đổi.

3. Thuật toán K-medoids: PAM

VD4: Minh họa thuật toán K-medoids: PAM

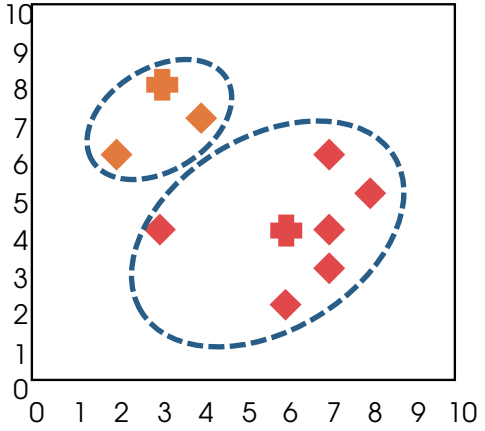


a) Phân cụm với k = 2



b) Chọn k = 2 điểm bất kỳ làm medoid

Total Cost = 19

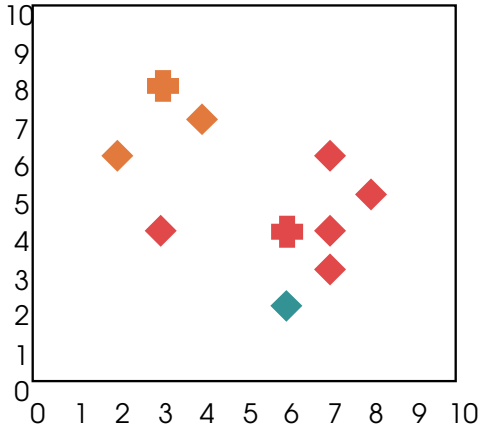


c) Gán từng điểm vào cụm gần medoid nhất.
Total Cost = 19

3. Thuật toán K-medoids: PAM

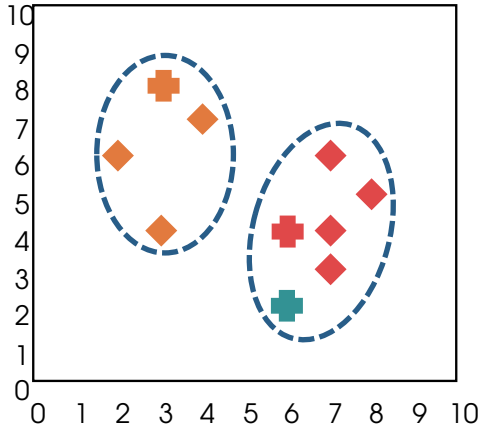
VD4: Minh họa thuật toán K-medoids: PAM

Total Cost = 19



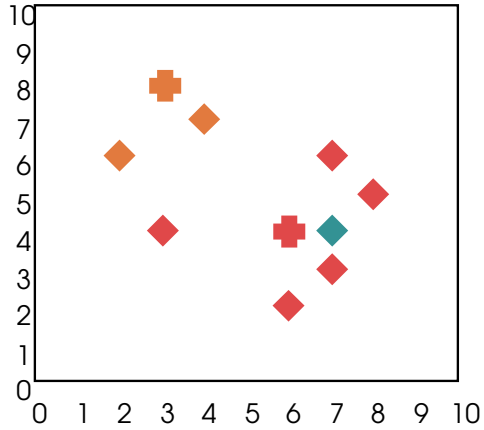
d) Chọn ngẫu nhiên 1 điểm non-medoid.
Swap với medoid

Total Cost = 26



e) Gán cụm lại.
Total Cost = 26 > 19.
Swap lại như trước đó

Total Cost = 19

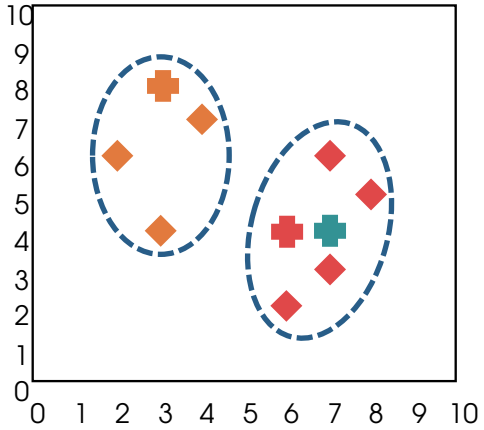


f) Tiếp tục chọn ngẫu nhiên 1 điểm non-medoid.
Swap với medoid

3. Thuật toán K-medoids: PAM

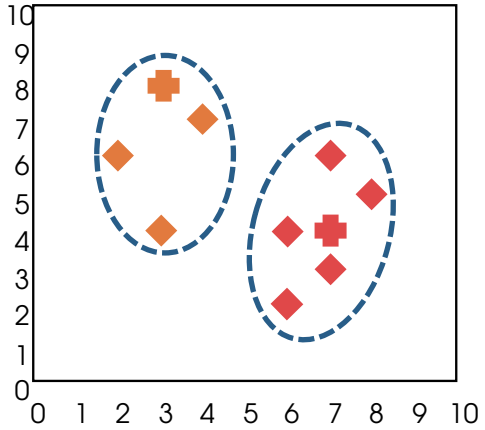
VD4: Minh họa thuật toán K-medoids: PAM

Total Cost = 18



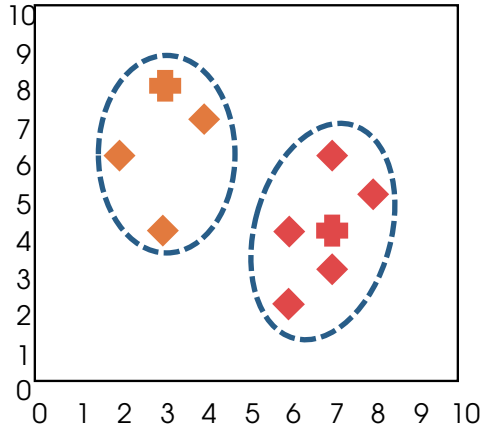
g) Gán cụm lại.
Total Cost = 18 < 19.
Lấy medoid mới

Total Cost = 18



h) Tiếp tục cho tất cả các điểm non-pedoid cho đến khi không có thay đổi

Total Cost = 18



i) Kết quả

3. Thuật toán K-medoids: PAM

VD5: Cho tập dữ liệu các điểm sau:

Sử dụng thuật toán PAM để phân cụm dữ liệu với số cụm là 2

X	x_1	x_2
A	7	6
B	2	6
C	3	8
D	8	5
E	7	4
F	4	7
G	6	2
H	7	3
I	6	4
J	3	4

3. Thuật toán K-medoids: PAM

VD5:

- **Bước 1: Khởi tạo trọng tâm medoids**

Chọn điểm C(3, 8) và I(6, 4) là medoids của 2 cụm C_1 và C_2

X	x_1	x_2
A	7	6
B	2	6
C	3	8
D	8	5
E	7	4
F	4	7
G	6	2
H	7	3
I	6	4
J	3	4

3. Thuật toán K-medoids: PAM

VD5:

- **Bước 2:** Tính khoảng cách của các điểm với medoids và gán cụm gần nhất. Tính tổng cost

X	x_1	x_2	Dist to $C_1(3,8)$	Dist to $C_2(6,4)$	Class
A	7	6	|7-3|+|6-8|=6	1+2=3	C_2
B	2	6	1+2=3	4+2=6	C_1
C	3	8	0	3+4=7	C_1
D	8	5	5+3=8	2+1=3	C_2
E	7	4	4+4=8	1+0=1	C_2
F	4	7	1+1=2	2+3=5	C_1
G	6	2	3+6=9	0+2=2	C_2
H	7	3	4+5=9	1+1=2	C_2
I	6	4	3+4=7	0	C_2
J	3	4	0+4=4	3+0=3	C_2

$$Cost = \sum_{i=1}^{k} \sum_{x_j \in C_i} dist(x_j, c_i)$$

$$= 3 + 3 + 0 + 3 + 1 + 2 + 2 + 2 + 0 + 3 = 19$$

VD5: 3. Thuật toán K-medoids: PAM

- **Bước 3: Chọn từng điểm non-medoids và swap với medoids.**
  - B3.1. Chọn G(6, 2) swap với I(6,4)

X	x_1	x_2	Dist to C_1(3,8)	Dist to C_2(6,2)	Class
A	7	6	4+2=6	1+4=5	C_2
B	2	6	1+2=3	4+4=8	C_1
C	3	8	0	3+6=9	C_1
D	8	5	5+3=8	2+3=5	C_2
E	7	4	4+4=8	1+2=3	C_2
F	4	7	1+1=2	2+5=7	C_1
G	6	2	3+6=9	0	C_2
H	7	3	4+5=9	1+1=2	C_2
I	6	4	3+4=7	0+2=2	C_2
J	3	4	0+4=4	3+2=5	C_1

$$Cost = \sum_{i=1}^{k} \sum_{x_j \in C_i} dist(x_j, c_i)$$

$$= 5 + 3 + 0 + 5 + 3 + 2 + 0 + 2 + 2 + 5 = \mathbf{26}$$

Cost ((3,8), (6,2)) > Cost ((3,8), (6,4))

=> Medoids: (3,8), (6,4)

VD5: 3. Thuật toán K-medoids: PAM

- **Bước 3: Chọn từng điểm non-medoids và swap với medoids.**
  - B3.2. Chọn E(7, 4) swap với I(6,4)

X	x_1	x_2	Dist to C_1(3,8)	Dist to C_2(7,4)	Class
A	7	6	4+2=6	0+2=2	C_2
B	2	6	1+2=3	5+2=7	C_1
C	3	8	0	4+4=8	C_1
D	8	5	5+3=8	1+1=2	C_2
E	7	4	4+4=8	0	C_2
F	4	7	1+1=2	3+3=6	C_1
G	6	2	3+6=9	1+2=3	C_2
H	7	3	4+5=9	0+1=1	C_2
I	6	4	3+4=7	1+0=1	C_2
J	3	4	0+4=4	4+0=4	C_1

$$Cost = \sum_{i=1}^{k} \sum_{x_j \in C_i} dist(x_j, c_i)$$

$$= 2 + 3 + 0 + 2 + 0 + 2 + 3 + 1 + 1 + 4 = \mathbf{18}$$

Cost ((3,8), (7,4)) < Cost ((3,8), (6,4))

⇒ **Medoids: (3,8), (7,4)**

⇒ Quay lại bước 2: gán lại cụm

⇒ Tiếp tục bước 3:

VD5: 3. Thuật toán K-medoids: PAM

- **Bước 3: Chọn từng điểm non-medoids và swap với medoids.**
  - Tiếp tục lặp lại với tất cả các điểm non-pedoids
  - Cho đến khi cụm không đổi, thuật toán kết thúc

X	x_1	x_2	Class
A	7	6	C_2
B	2	6	C_1
C	3	8	C_1
D	8	5	C_2
E	7	4	C_2
F	4	7	C_1
G	6	2	C_2
H	7	3	C_2
I	6	4	C_2
J	3	4	C_1

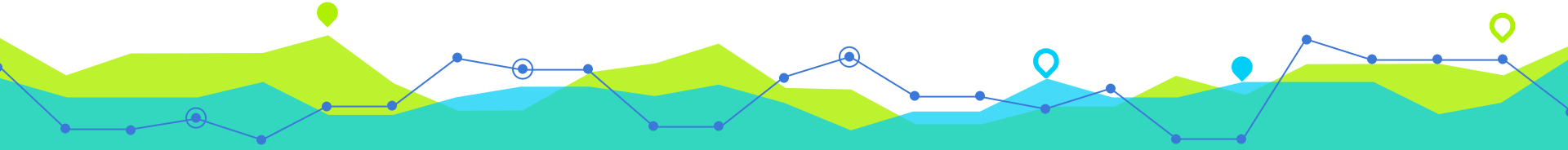
Kết luận:

Cụm C_1 = {C(3,8), B(2,6), F(4,7), J(3,4)}

Cụm C_2 = {E(7,4), A(7,6), D(8,5), G(C,2), H(7,3), I(6,4)}

3. Thuật toán K-medoids: PAM

- PAM hiệu quả hơn so với K-means với dữ liệu nhiễu, cá biệt.
- PAM hiệu quả với tập dữ liệu nhỏ nhưng không mở rộng tốt với tập dữ liệu lớn
- Cải tiến:
  - CLARA – Clustering LARge Applications (Kaufmann & Rousseeuw, 1990): dựa trên phương pháp lấy mẫu
  - CLARANS – Clustering LARge Applications based upon RANdomized Search (Ng & Han, 1994) : lấy mẫu động

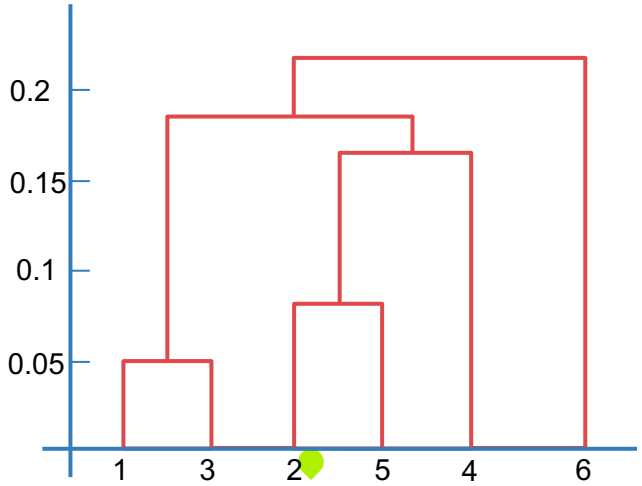


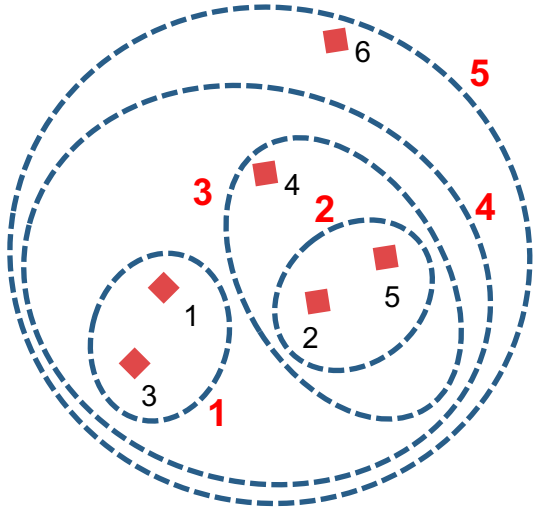
3 Phương pháp phân cấp

1. Giới thiệu
2. Thuật toán AGNES
3. Thuận toán DIANA

1. Giới thiệu

- Xây dựng và tổ chức các cụm như cây phân cấp
- Biểu diễn dưới dạng sơ đồ hình cây, lưu lại quá trình gom, phân cụm
- Sử dụng ma trận khoảng cách

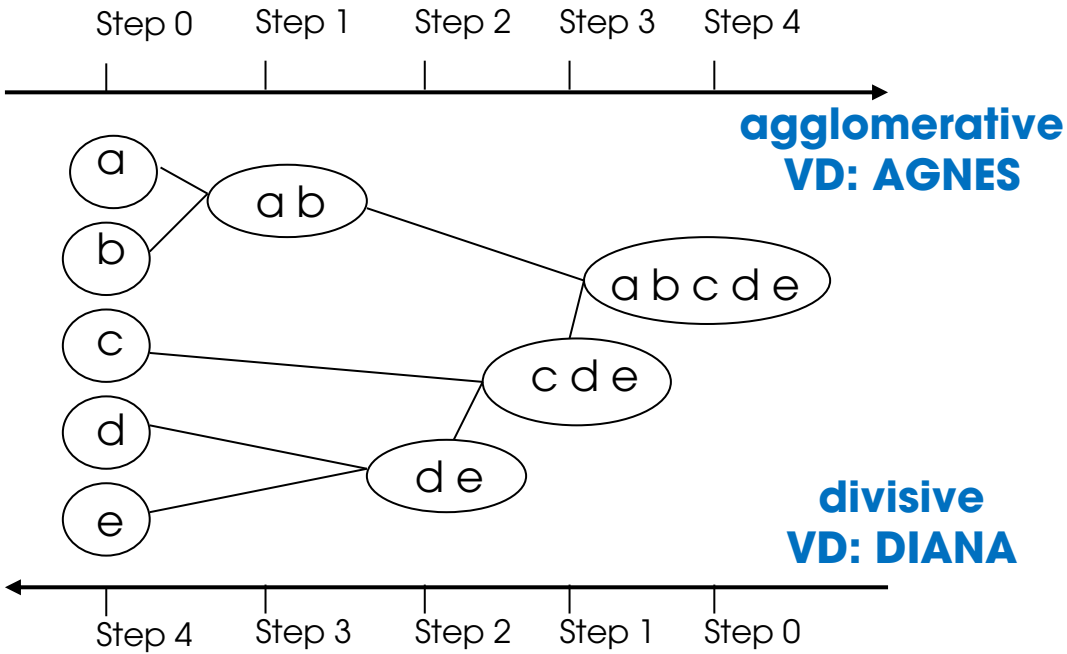




Không cần xác định trước số cụm k

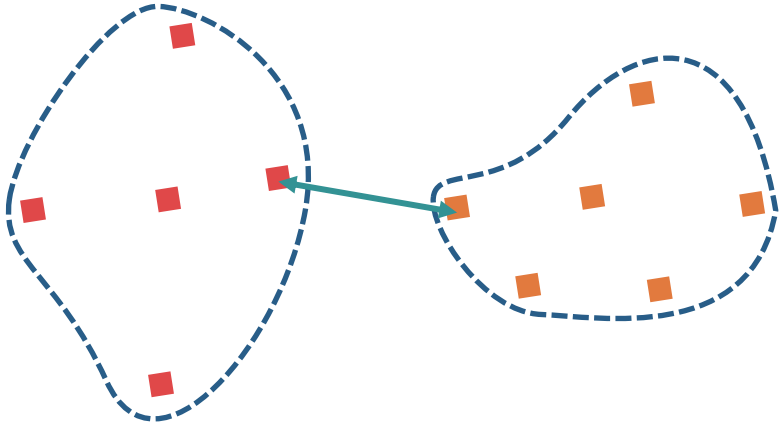
1. Giới thiệu

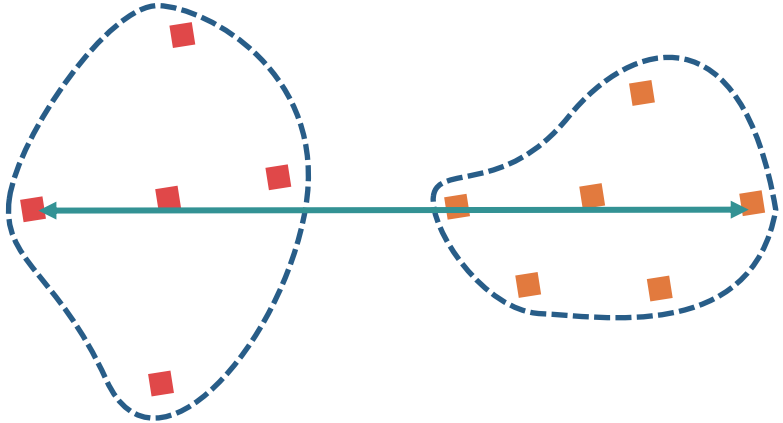
- **02 loại phân cấp:**
  - Tích tụ (agglomerative): từ dưới lên, mỗi đối tượng là một cụm
  - Chia nhỏ (divisive): từ trên xuống, tất cả các đối tượng là một cụm



1. Giới thiệu

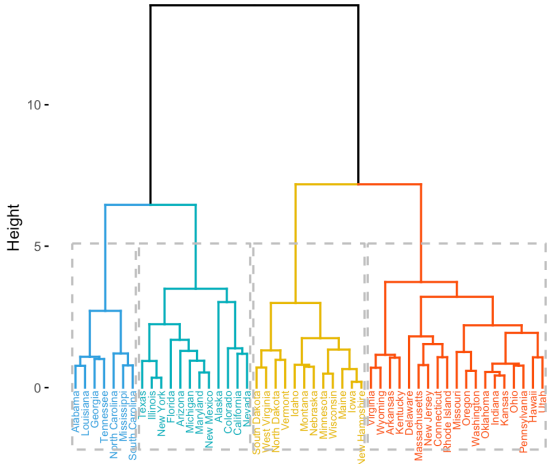
- **Cách xác định khoảng cách giữa các cụm**
  - **Single link:** khoảng cách gần nhất giữa 2 đối tượng thuộc 2 cụm
  - **Complete link:** khoảng cách xa nhất giữa 2 đối tượng thuộc 2 cụm





1. Giới thiệu

- **Một số thuật toán gọm cụm phân cấp**
  - AGNES (Agglomerative Nesting), DIANA (Divisive Analysis)
  - BIRCH (Balance Iterative Reducing & Clustering using Hierarchies) (*)
  - CURE (Clustering Using Representative) (*)
  - ROCK (Robust Clustering using linKs) (*)
  - CHAMELEON (*)

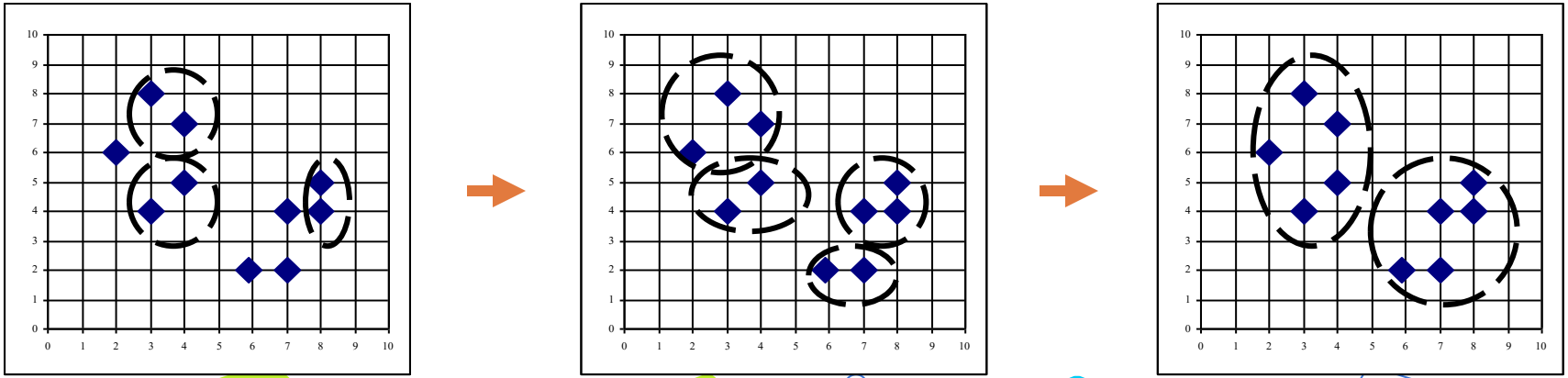


(*): tìm hiểu và semniar

Kaufman, Leonard, and Peter J. Rousseeuw. "Finding groups in data. an introduction to cluster analysis." Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics (1990).

2. Thuật toán AGNES (Agglomerative Nesting)

- **B1:** Mỗi đối tượng là một cụm
- **B2:** Tính khoảng cách giữa các cụm (Single link hoặc Complete link).
  Gom các cụm có khoảng cách giữa các cụm là nhỏ nhất.
- **B3:** Nếu thu được cụm "toàn bộ": Dừng
  Ngược lại: quay lại B2



2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Cho tập dữ liệu:

1. Sử dụng thuật toán AGNES với Single link để gom cụm. Vẽ sơ đồ hình cây.
2. Xác định 3 cụm thu được.

Single link: Khoảng cách cụm là khoảng cách gần nhất giữa 2 điểm của 2 cụm.

Điểm	x	y
P1	0.40	0.63
P2	0.22	0.38
P3	0.353	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

2. Thuật toán AGNES (Agglomerative Nesting)

VD6:

- **Xây dựng ma trận khoảng cách giữa các điểm (Euclide):**

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

$$d(P_1, P_2) = \sqrt{(0.22 - 0.40)^2 + (0.38 - 0.63)^2} = 0.23$$

$$d(P_1, P_3) = \sqrt{(0.353 - 0.40)^2 + (0.32 - 0.63)^2} = 0.22$$

…

2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

1. Mỗi điểm là 1 cụm

2. $dist(P3, P6) = 0.1$ là nhỏ nhất

$\Rightarrow$ Gom {P3}, {P6} thành 1 cụm

$\Rightarrow$ **Các cụm: {P1}, {P2}, {P4}, {P5}, {P3, P6}**

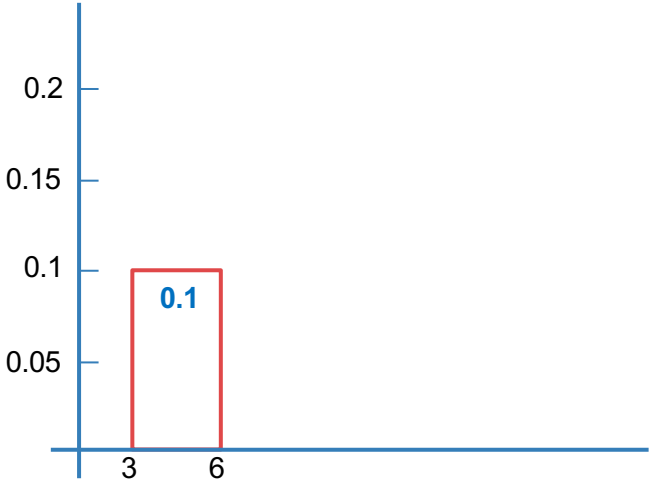
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	**0.10**	0.22	0.39	0

2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

Các cụm sau 2.:

{P1}, {P2}, {P4}, {P5}, {P3, P6}



3. Chưa thu được cụm "toàn bộ": Tiếp tục

2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

4. Tính khoảng cách giữa {P1}, {P2}, {P4}, {P5}, {P3, P6}

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

	P1	P2	P4	P5	P3, P6
P1	0				
P2	0.23	0			
P4	0.37	0.19	0		
P5	0.34	**0.14**	0.28	0	
P3, P6	**0.22**	**0.15**	**0.16**	**0.29**	0

$$dist(\{P1\},\{P3,P6\}) = \boldsymbol{min}\big(dist(\{P1\},\{P3\}), dist(\{P1\},\{P6\})\big)$$
$$= \boldsymbol{min}(0.22, 0.24) = 0.22$$

$$dist(\{P2\},\{P3,P6\}) = \boldsymbol{min}(0.15, 0.24) = 0.15$$

$$dist(\{P4\},\{P3,P6\}) = \boldsymbol{min}(0.16, 0.22) = 0.16$$

$$dist(\{P5\},\{P3,P6\}) = \boldsymbol{min}(0.29, 0.39) = 0.29$$

2. Thuật toán AGNES (Agglomerative Nesting)

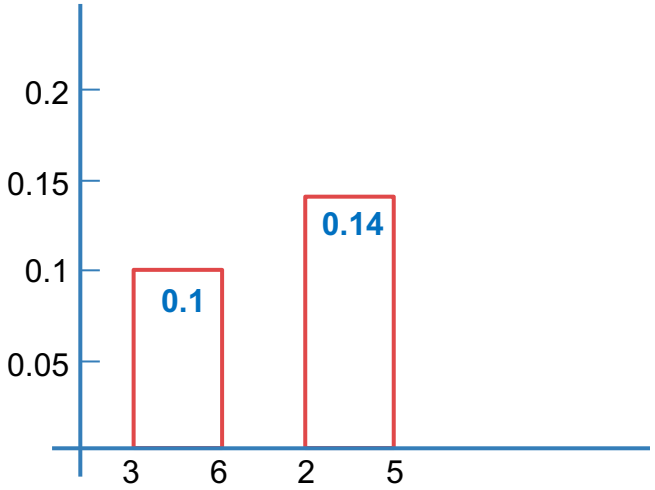
VD6: Sử dụng Single link

5. $dist(\{P2\},\{P5\})$ nhỏ nhất

$\Rightarrow$ Gom {P2} và {P5}

$\Rightarrow$ **Các cụm: {P1}, {P4}, {P2, P5}, {P3, P6}**

6. Chưa thu được cụm "toàn bộ": Tiếp tục



2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

7. Tính khoảng cách giữa {P1}, {P4}, {P2, P5}, {P3, P6}

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

	P1	P4	P2, P5	P3, P6
P1	0			
P4	0.37	0		
P2, P5	**0.23**	**0.19**	0	
P3, P6	0.22	0.16	**0.15**	0

$$dist(\{P1\},\{P2,P5\}) = \boldsymbol{min}(0.23, 0.34) = 0.23$$

$$dist(\{P4\},\{P2,P5\}) = \boldsymbol{min}(0.19, 0.28) = 0.19$$

$$dist(\{P2,P5\},\{P3,P6\}) = \boldsymbol{min}(0.15, 0.24, 0.29, 0.39) = 0.15$$

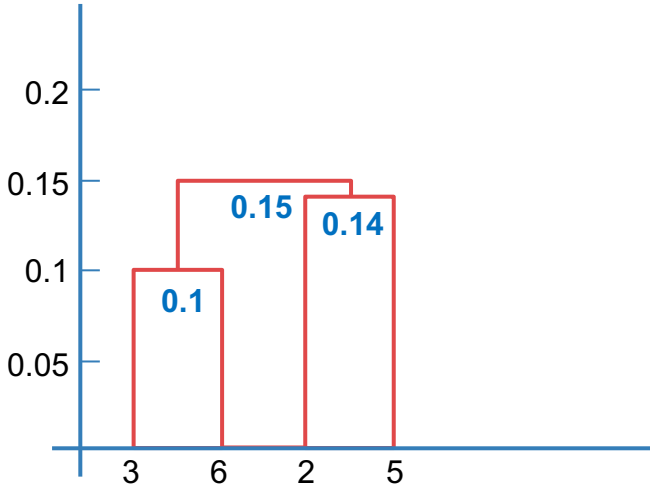
2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

7. $dist(\{P2, P5\}, \{P3, P6\})$ nhỏ nhất

⇒ Gom {P2, P5} và {P3, P6}

⇒ **Các cụm: {P1}, {P4}, {P2, P3, P5, P6}**



8. Chưa thu được cụm "toàn bộ": Tiếp tục

2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

9. Tính khoảng cách giữa {P1}, {P4}, {P2, P3, P5, P6}

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

	P1	P4	P2, P3, P5, P6
P1	0		
P4	0.37	0	
P2, P3, P5, P6	**0.22**	**0.16**	0

$$dist(\{P1\}, \{P2, P3, P5, P6\}) = \boldsymbol{min}(0.23, 0.22, 0.34, 0.24) = 0.22$$

$$dist(\{P4\}, \{P2, P3, P5, P6\}) = \boldsymbol{min}(0.19, 0.16, 0.28, 0.22) = 0.16$$

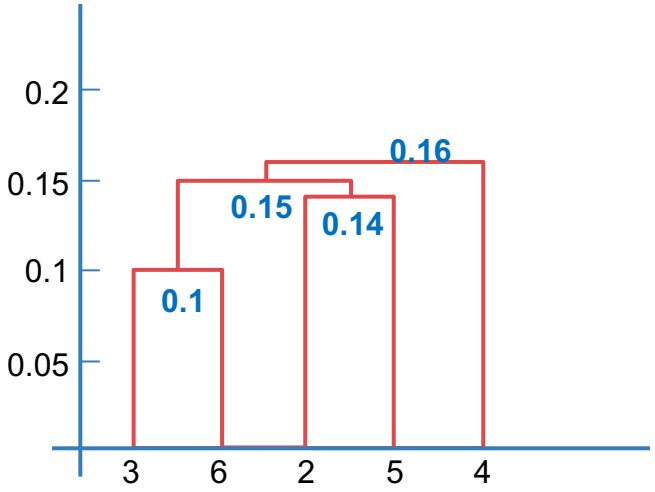
2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

10. $dist(\{P4\}, \{P2, P3, P5, P6\})$ nhỏ nhất.

⇒ Gom {P4} và {P2, P3, P5, P6}

⇒ **Các cụm: {P1}, {P2, P3, P4, P5, P6}**



11. Chưa thu được cụm "toàn bộ": Tiếp tục

2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Sử dụng Single link

9. Tính khoảng cách {P1}, {P2, P3, P4, P5, P6}

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

	P1	P2, P3, P4, P5, P6
P1	0	
P2, P3, P4, P5, P6	**0.22**	0

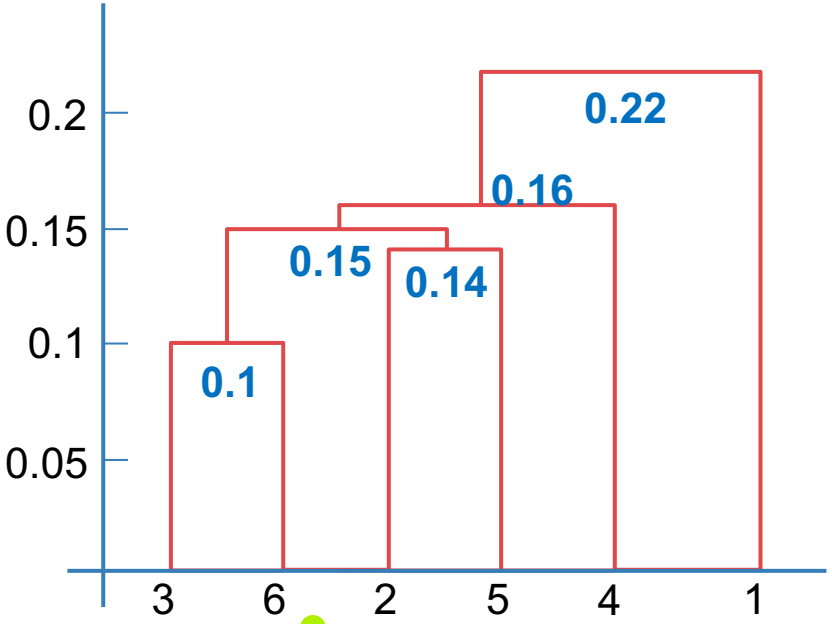
$$dist(\{P1\}, \{P2, P3, P4, P5, P6\}) = \min(0.23, 0.22, 0.34, 0.37, 0.24) = 0.22$$

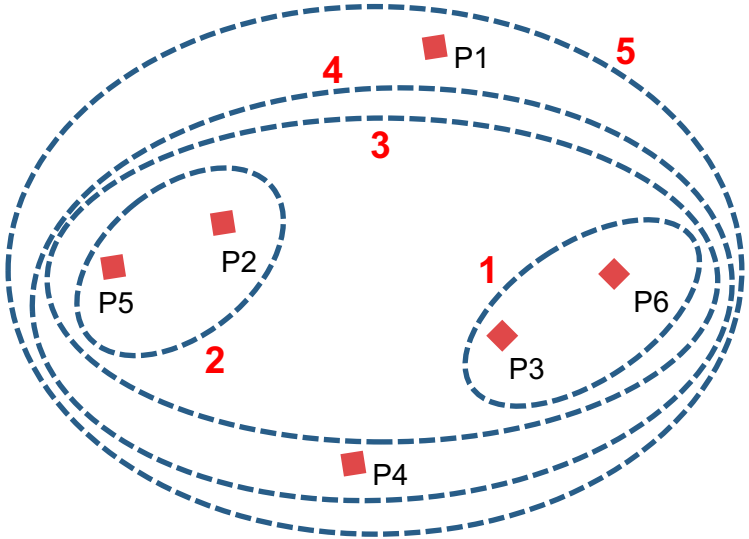
⇒ Gom {P1} và {P2, P3, P4, P5, P6}

⇒ **Cụm: {P1, P2, P3, P4, P5, P6}. Dừng thuật toán**

2. Thuật toán AGNES (Agglomerative Nesting)

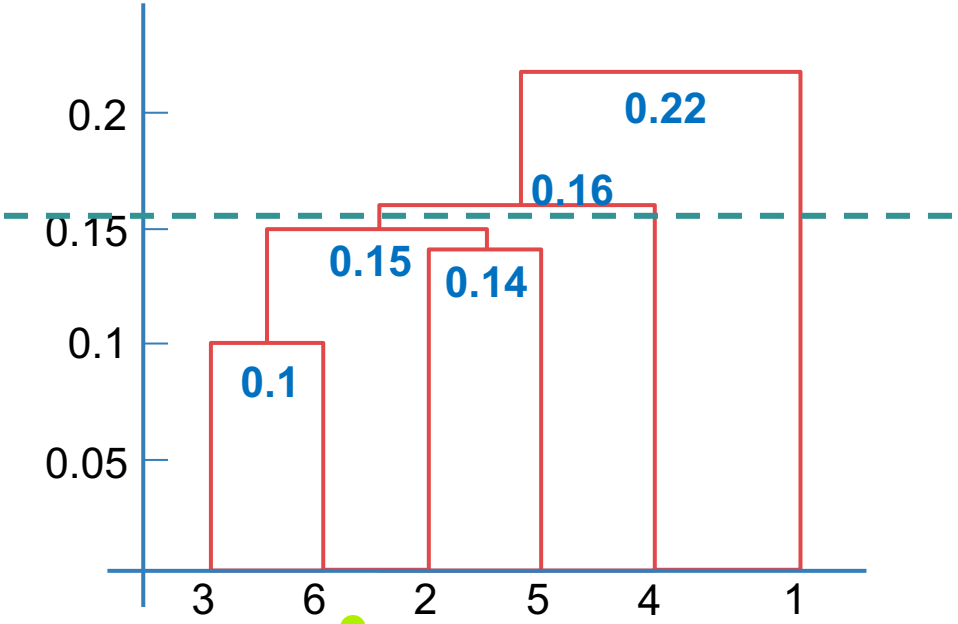
VD6: Sơ đồ hình cây (sử dụng Single link)

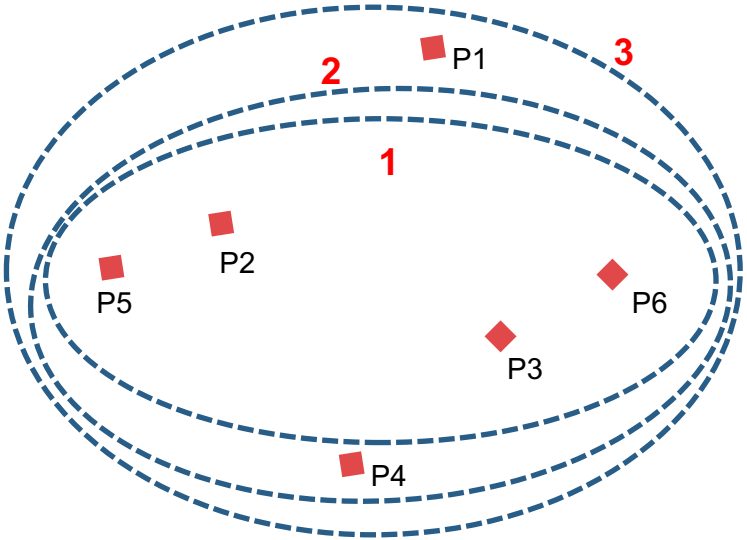




2. Thuật toán AGNES (Agglomerative Nesting)

VD6: Chia 03 cụm





3 cụm: {P3, P6, P2, P5}, {P4}, {P1}

AGNES: Bài tập

BT16

Cho ma trận khoảng cách:

1. Sử dụng thuật toán AGNES với Complete link để gom cụm. Vẽ sơ đồ hình cây.
2. Xác định 3 cụm thu được.

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.19	0.16	0		
P5	0.34	0.14	0.29	0.28	0	
P6	0.24	0.24	0.10	0.22	0.39	0

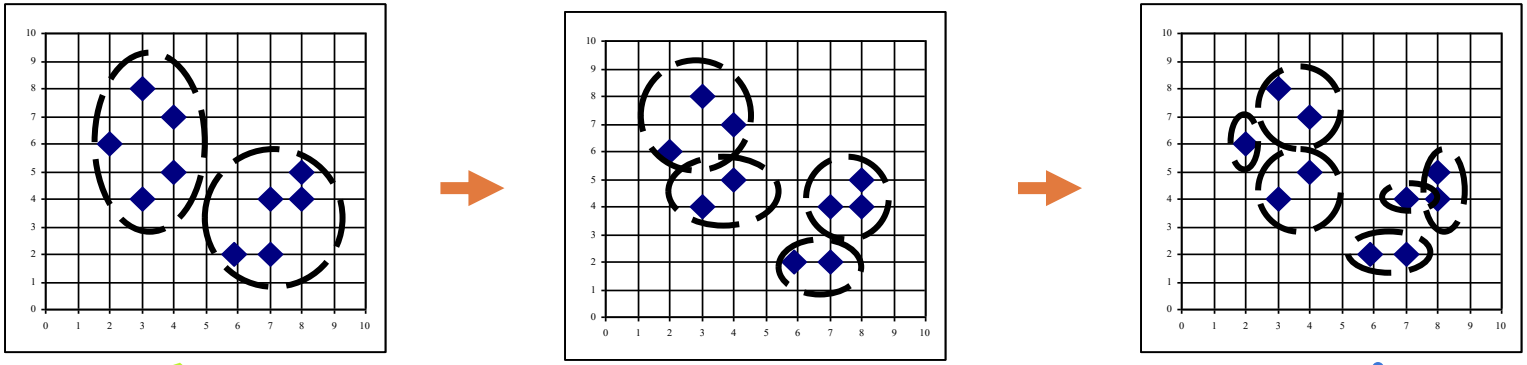
Yêu cầu:

- Nhóm 04 thành viên
- Thời gian: 25 phút
- Nộp bài trên Course

Complete link: Khoảng cách cụm là khoảng cách xa nhất giữa 2 điểm của 2 cụm.

3. Thuật toán DIANA (Divisive Analysis)

- **Ý tưởng:**
  - Tất cả các đối tượng là một cụm
  - Chia nhỏ các cụm có khoảng cách giữa các đối tượng trong nhóm là lớn nhất
  - Nếu thu được cụm chỉ chứa 1 đối tượng thì dừng. Ngược lại, quay lại bước trên.



3. Thuật toán DIANA (Divisive Analysis)

- **B1:** Giả sử rằng cụm C sẽ được chia thành các cụm C_i và C_j

$$C_i = C \text{ và } C_j = \emptyset$$

- **B2:** Đối với từng đối tượng $x \in C_i$

a) Đối với lần lặp đầu tiên, tính khoảng cách trung bình của x đến tất cả các đối tượng khác (D_x)

b) Đối với lần lặp còn lại, tính

$$D_x = avg\{d(x,y): y \in C_i\} - avg\{d(x,y): y \in C_j\}$$

3. Thuật toán DIANA (Divisive Analysis)

- **B3:**

  a) Đối với lần lặp đầu tiên, di chuyển đối tượng có khoảng cách trung bình lớn nhất đến C_j

  b) Đối với lần lặp còn lại, tìm một đối tượng $x \in C_i$ sao cho D_x lớn nhất. Nếu $D_x > 0$ thì chuyển x đến C_j

- **B4:** Lặp lại bước 2b và 3b cho đến khi tất cả D_x đều âm. C_i được tách thành C_i và C_j

3. Thuật toán DIANA (Divisive Analysis)

- **B5:** Chọn cụm nhỏ có đường kính lớn nhất. Sau đó chia cụm này, làm theo các bước 1-5.

- **B6:** Lặp lại cho đến khi tất cả các cụm chỉ chứa một đối tượng

3. Thuật toán DIANA (Divisive Analysis)

VD7: Cho ma trận khoảng cách:

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Sử dụng thuật toán DIANA để gom cụm.

3. Thuật toán DIANA (Divisive Analysis)

VD7:

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

1. C_1= {a, b, c, d, e} và C_2 ={ }

2. Tính Average Dissimilarity giữa từng điểm với điểm còn lại

- $D_a = \frac{1}{4}[d(a,b) + d(a,c) + d(a,d) + d(a,e)] = \frac{1}{4}(9 + 3 + 6 + 11) = 7.25$
- $D_b = \frac{1}{4}(9 + 7 + 5 + 10) = 7.75$
- $D_c = \frac{1}{4}(3 + 7 + 9 + 2) = 5.25$
- $D_d = \frac{1}{4}(6 + 5 + 9 + 8) = 7.25$
- $D_e = \frac{1}{4}(11 + 10 + 2 + 8) = 7.75$

b, e có average dissimilarity cao nhất

$\Rightarrow$ Tách b vào cụm C_2

$\Rightarrow$ C_1 ={a, c, d, e} và C_2 ={b}

3. Thuật toán DIANA (Divisive Analysis)

VD7:

C_1 ={a, c, d, e} và C_2 ={b}

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

3. Tính Average Dissimilarity giữa từng điểm với các điểm còn lại

- $D_a = \frac{1}{3}[d(a,c) + d(a,d) + d(a,e)] - \frac{1}{1}[d(a,b)]$
  $= \frac{1}{3}(3 + 6 + 11) - 9 = -2.33$
- $D_c = \frac{1}{3}(3 + 9 + 2) - 7 = -2.33$
- $D_d = \frac{1}{3}(6 + 9 + 8) - 5 = 2.67$
- $D_e = \frac{1}{3}(11 + 2 + 8) - 10 = -3$

D_d >0 và lớn nhất

⇒ Tách d vào cụm C_2

⇒ C_1 ={a, c, e} và C_2 ={b, d}

3. Thuật toán DIANA (Divisive Analysis)

VD7:

C_1 ={a, c, e} và C_2 ={b, d}

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

4. Tính Average Dissimilarity giữa từng điểm với các điểm còn lại

- $D_a = \frac{1}{2}[d(a,c) + d(a,e)] - \frac{1}{2}[d(a,b) + d(a,d)]$
  $= \frac{1}{2}(3 + 11) - \frac{1}{2}(9 + 6) = -0.5$
- $D_c = \frac{1}{2}(3 + 2) - \frac{1}{2}(7 + 9) = -5.5$
- $D_e = \frac{1}{2}(11 + 2) - \frac{1}{2}(10 + 8) = -2.5$

Tất cả D<0

$\Rightarrow$ C_1 ={a, c, e} và C_2 ={b, d}

$\Rightarrow$ Tách từng cụm C_1 và C_2

3. Thuật toán DIANA (Divisive Analysis)

VD7:

C_1 ={a, c, e} và C_2 ={b, d}

5. Tính đường kính (diameter) của các cụm

- $Diameter_{C_1} = max[d(a,c), d(a,e), d(c,e)] = \max(3, 11, 2) = 11$
- $Diameter_{C_2} = max[d(b,d)\,] = \max(5) = 5$

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Ta có $Diameter_{C_1}$ lớn nhất

$\Rightarrow$ Tách cụm C_1 ={a, c, e}

3. Thuật toán DIANA (Divisive Analysis)

VD7:

6. Tính Average Dissimilarity giữa từng điểm với điểm còn lại trong cụm C_1 ={a, c, e}

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- $D_a = \frac{1}{2}[d(a,c) + d(a,e)] = \frac{1}{2}(3 + 11) = 7$
- $D_c = \frac{1}{2}(3 + 2) = 2.5$
- $D_e = \frac{1}{2}(11 + 2) = 6.5$

a có average dissimilarity cao nhất

$\Rightarrow$ Tách a vào cụm C_3

$\Rightarrow$ Có: C_1 ={c, e}, C_2 ={b, d}, C_3 ={a}

3. Thuật toán DIANA (Divisive Analysis)

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

VD7:

C_1 ={c, e}, C_2 ={b, d}, C_3 ={a}

7. Tính đường kính (diameter) của các cụm

- $Diameter_{C_1} = max[d(c,e)] = \max(2) = 2$
- $Diameter_{C_2} = max[d(b,d)\,] = \max(5) = 5$
- $Diameter_{C_3} = 0$

Ta có $Diameter_{C_2}$ lớn nhất

$\Rightarrow$ Tách cụm C_2 ={b, d} thành C_2 ={b}, C_4 = {d}

$\Rightarrow$ Có: C_1 ={c, e}, C_2 ={b}, C_3 ={a}, C_4 ={d}

3. Thuật toán DIANA (Divisive Analysis)

VD7:

C_1 ={c, e}, C_2 ={b}, C_3 ={a}, C_4 ={d}

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

8. Tính đường kính (diameter) của các cụm

- $Diameter_{C_1} = max[d(c,e)] = \max(2) = 2$
- $Diameter_{C_2} = 0$
- $Diameter_{C_3} = 0$
- $Diameter_{C_4} = 0$

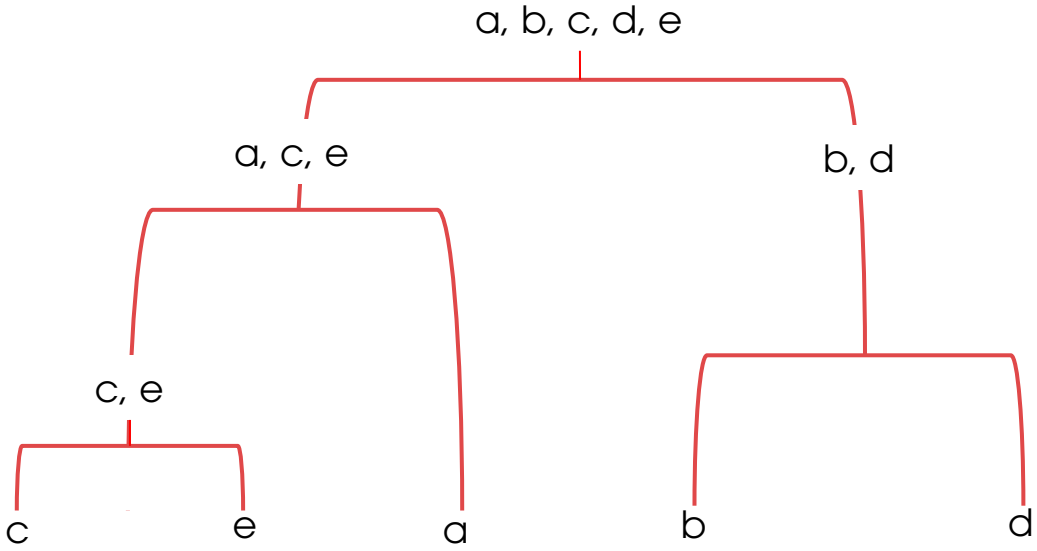
Tất cả $Diameter_{C_1}$ lớn nhất

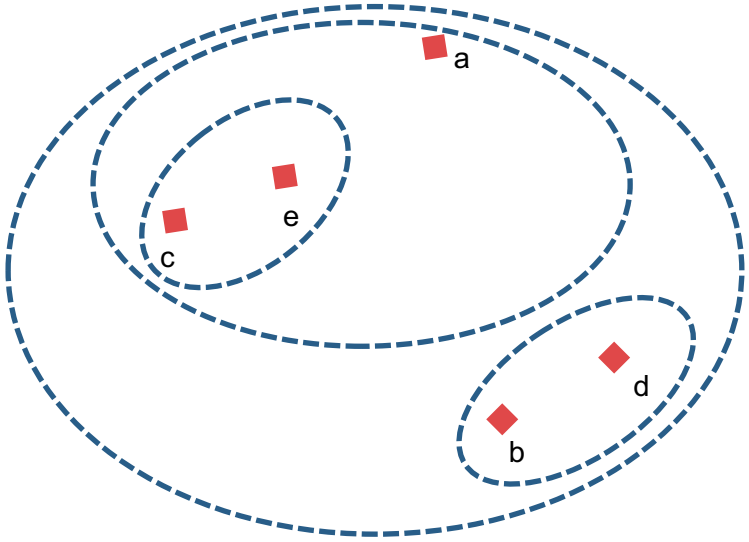
$\Rightarrow$ Tách cụm C_1 ={c, e} thành C_1 ={c}, C_5 = {e}

Có: C_1 ={c}, C_2 ={b}, C_3 ={a}, C_4 ={d}, C_5 = {e} (Dừng)

3. Thuật toán DIANA (Divisive Analysis)

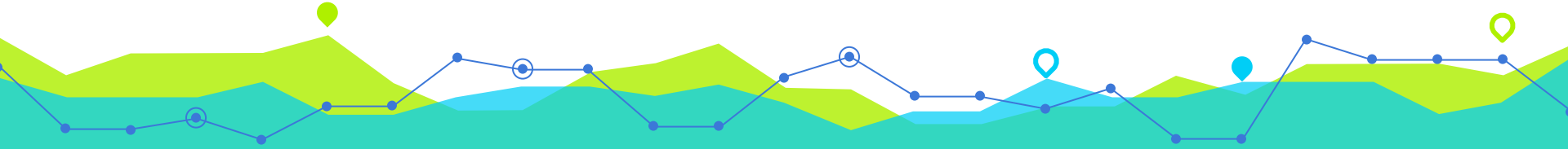
VD7: Sơ đồ hình cây





Phương pháp phân cấp

- **Nhược điểm:**
  - Tính mở rộng thấp. Độ phức tạp O(n2), n: số đối tượng
  - Không thể quay trở lại về bước trước
  - Khó xác định phương pháp tích tụ hay chia nhỏ
  - Nhạy cảm với nhiễu, cá biệt
  - Gặp vấn đề khi các cụm có kích thước khác nhau và có hình dạng cầu
  - Có xu hướng phân chia các nhóm dữ liệu lớn
- Tích hợp phương pháp phân cấp với phân hoạch: BIRCH, CURE, CHAMELEON

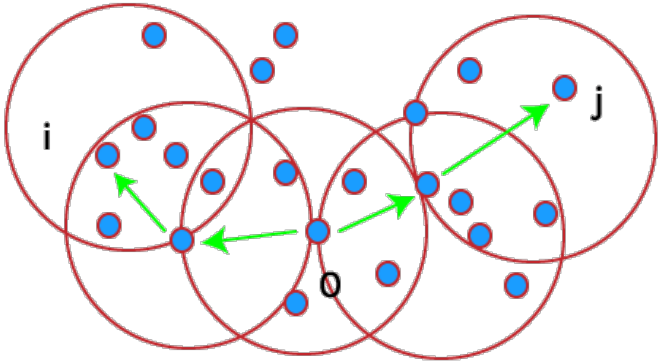


4 Phương pháp dựa trên mật độ

1. Giới thiệu
2. Các khái niệm cơ bản
3. Thuật toán DBSCAN

1. Giới thiệu

- **Phương pháp dựa trên mật độ (density):**
  - Mật độ: số các đối tượng lân cận của một đối tượng dữ liệu theo một ngưỡng nào đó.
  - Khi một cụm dữ liệu đã xác định thì nó có thể tiếp tục được mở rộng thêm các đối tượng mới miễn mật độ của nó lớn hơn ngưỡng.
- Một số thuật toán:
  - DBSCAN Ester, et al. (KDD'96)
  - OPTICS Ankerst, et al (SIGMOD'99).
  - DENCLUE Hinneburg & D. Keim (KDD'98)
  - CLIQUE Agrawal, et al. (SIGMOD'98)

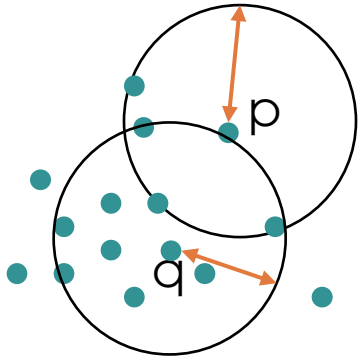


2. Các khái niệm cơ bản

- 2 tham số do người dùng xác định
  - Eps ($\varepsilon > 0$): bán kính lớn nhất vùng lân cận
  - MinPts (ngưỡng): số nhỏ nhất các đối tượng trong vùng lân cận của 1 đối tượng với bán kính Eps

- Mật độ: số đối tượng nằm trong bán kính Eps của một đối tượng

$$N_{\text{eps}}(p): tập \{q \in D | dist(p,q) \leq \text{Eps}\}$$

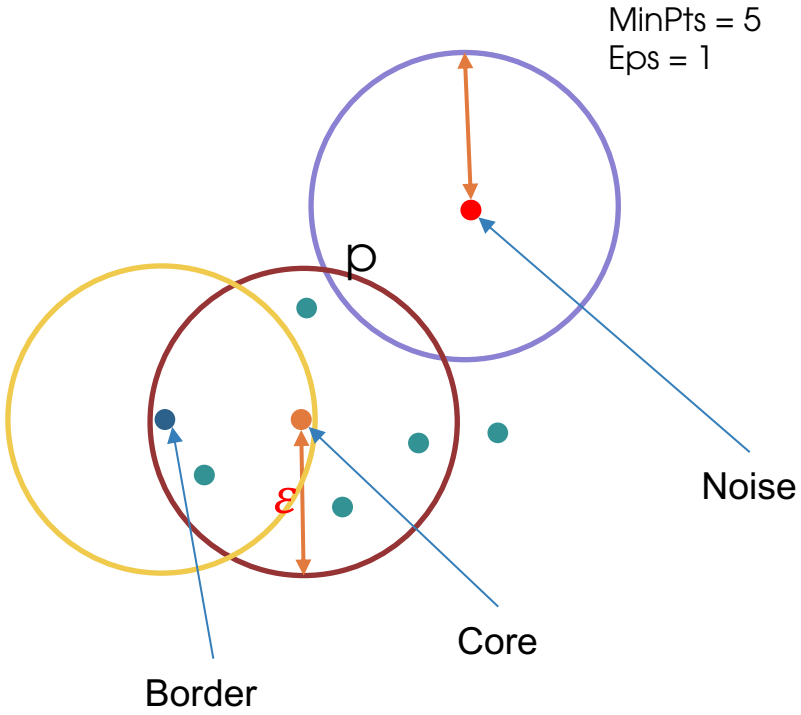


MinPts = 5

Eps = 1 cm

2. Các khái niệm cơ bản

- **Đối tượng nòng cốt (core point):** có số đối tượng lân cận trong bán kính Eps lớn hơn hoặc bằng MinPts
- **Đối tượng biên (borer point):** có số đối tượng lân cận trong bán kính Eps nhỏ hơn MinPts nhưng vẫn nằm trong vùng lân cận của đối tượng core
- **Đối tượng nhiễu (noise point / outlier):** đối tượng không phải core, không phải border



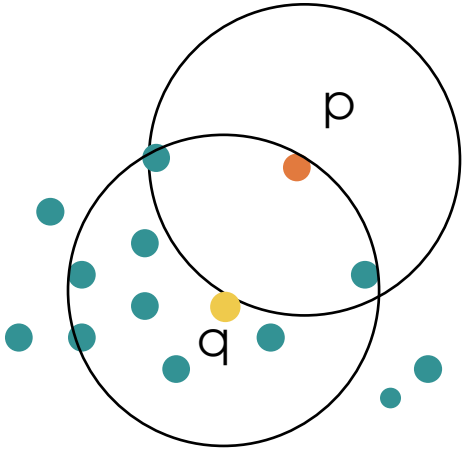
2. Các khái niệm cơ bản

- **Mật độ đạt được trực tiếp (Directly density reachable):**

p: đối tượng có mật độ đạt được trực tiếp từ đối tượng q theo Eps và MinPts nếu:

$$p \in N_{\text{eps}}(q)$$

Và $\left|N_{\text{eps}}(q)\right| \geq MinPts$



MinPts = 5

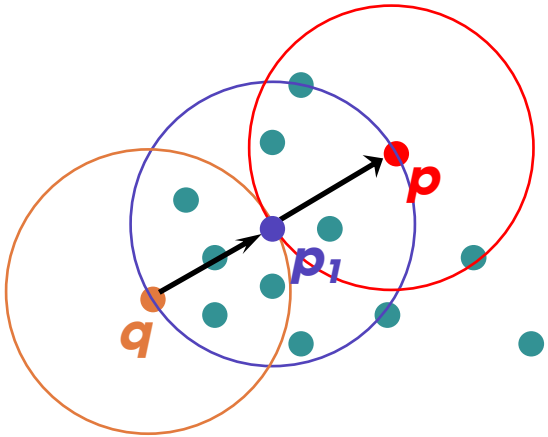
Eps = 1 cm

2. Các khái niệm cơ bản

- **Mật độ đạt được (Density reachable):**

p: đối tượng có mật độ có thể đạt được từ đối tượng q theo Eps và MinPts nếu:

Tồn tại một dây chuyền các đối tượng $p_1 \dots p_n$ với $p_1 = q$, $p_n = p$ sao cho p_{n+1} là đối tượng có mật độ đạt được trực tiếp từ p_i



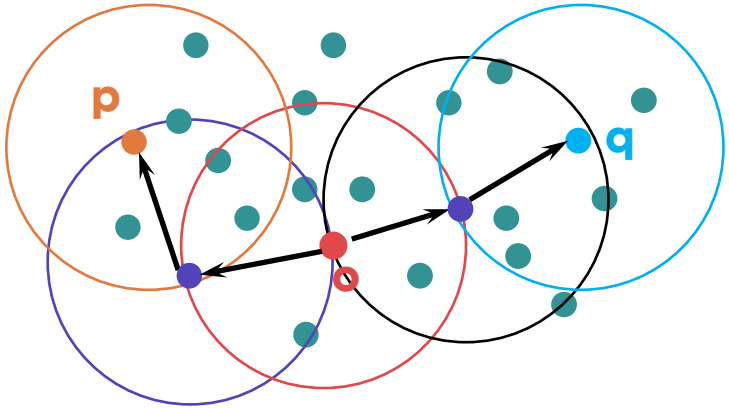
MinPts = 5

Eps = 1 cm

2. Các khái niệm cơ bản

- **Mật độ liên thông (Density connected):**

p: đối đối tượng có mật độ liên thông từ đối tượng q theo Eps và MinPts nếu: Tồn tại một đối tượng o sao cho p, q là đối tượng có mật độ có thể đạt được từ o



MinPts = 5

Eps = 1 cm

3. Thuật toán DBSCAN

- DBSCAN: Density Based Spatial Clustering of Application with Noise
- Một cụm được xác định như tập các đối tượng có mật độ liên thông lớn nhất
- Tìm các cụm hình dạng bất kỳ trong CSDL không gian có nhiễu

3. Thuật toán DBSCAN

- **B1:** Chọn ngẫu nhiên đối tượng p
- **B2:** Tìm tất cả các đối tượng có mật độ có thể đạt được từ p theo Eps, MinPts
- **B3:** Nếu p là core thì hình thành cụm. Nếu p là border thì xem xét đối tượng tiếp theo trong CSDL
- **B4:** Tiếp tục cho đến khi tất cả các đối tượng đều được xử lý

3. Thuật toán DBSCAN

VD8: Cho tập dữ liệu

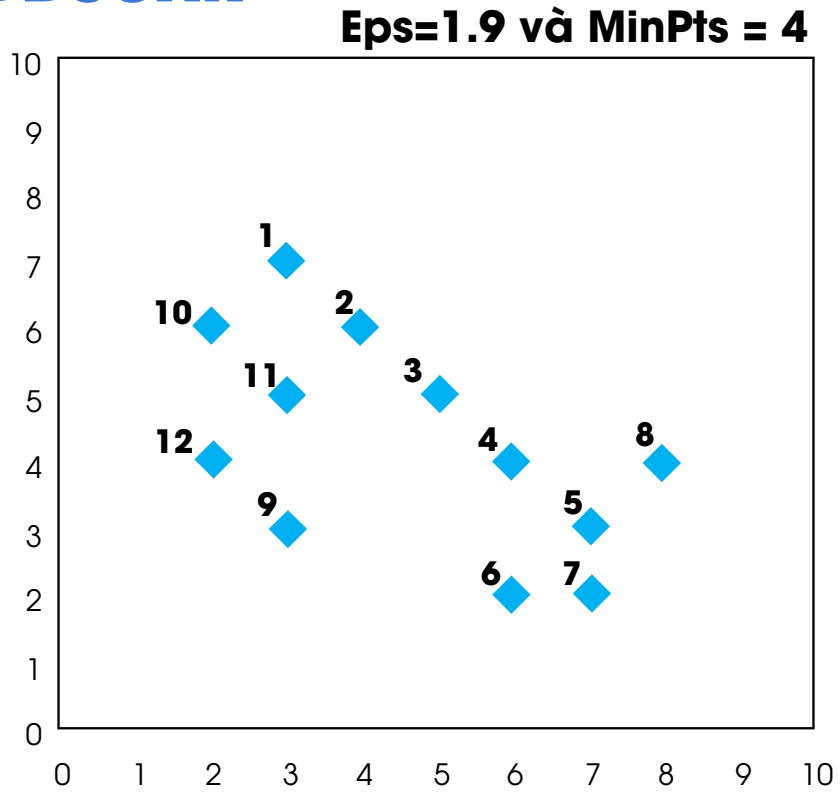
Sử dụng DBSCAN với Eps=1.9 và MinPts = 4 để gom cụm dữ liệu

Point	X	Y
P1	3	7
P2	4	6
P3	5	5
P4	6	4
P5	7	3
P6	6	2
P7	7	2
P8	8	4
P9	3	3
P10	2	6
P11	3	5
P12	2	4

3. Thuật toán DBSCAN

VD8:

Point	X	Y
P1	3	7
P2	4	6
P3	5	5
P4	6	4
P5	7	3
P6	6	2
P7	7	2
P8	8	4
P9	3	3
P10	2	6
P11	3	5
P12	2	4



3. Thuật toán DBSCAN

VD8: Tính ma trận khoảng cách

Eps=1.9 và MinPts = 4

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0	1.41	2.83	4.24	5.66	5.83	6.40	5,83	4.00	1.41	2.00	3.16
P2	1.41	0	1.41	2.83	4.24	4.47	5.00	4.47	3.16	2.00	1.41	2.83
P3	2.83	1.41	0	1.41	2.83	3.16	3.61	3.16	2.83	3.16	2.00	3.16
P4	4.24	2.83	1.41	0	1.41	2.00	2.24	2.00	3.16	4.47	3.16	4.00
P5	5.66	4.24	2.83	1.41	0	1.41	1.00	1.41	4.00	5.83	4.47	5.10
P6	5.83	4.47	3.16	2.00	1.41	0	1.00	2.83	3.16	5.66	4.24	4.47
P7	6.40	5.00	3.61	2.24	1.00	1.00	0	2.24	4.12	6.40	5.00	5.39
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0	5.10	6.32	5.10	6.00
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0	3.16	2.00	1.41
P10	1.41	2.00	3.16	4.47	5.83	5.66	6,40	6.32	3.16	0	1.41	2.00
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	1.41
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

3. Thuật toán DBSCAN

VD8:

Eps=1.9 và MinPts = 4

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0	1.41	2.83	4.24	5.66	5.83	6.40	5,83	4.00	1.41	2.00	3.16
P2	**1.41**	0	1.41	2.83	4.24	4.47	5.00	4.47	3.16	2.00	1.41	2.83
P3	2.83	1.41	0	1.41	2.83	3.16	3.61	3.16	2.83	3.16	2.00	3.16
P4	4.24	2.83	1.41	0	1.41	2.00	2.24	2.00	3.16	4.47	3.16	4.00
P5	5.66	4.24	2.83	1.41	0	1.41	1.00	1.41	4.00	5.83	4.47	5.10
P6	5.83	4.47	3.16	2.00	1.41	0	1.00	2.83	3.16	5.66	4.24	4.47
P7	6.40	5.00	3.61	2.24	1.00	1.00	0	2.24	4.12	6.40	5.00	5.39
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0	5.10	6.32	5.10	6.00
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0	3.16	2.00	1.41
P10	**1.41**	2.00	3.16	4.47	5.83	5.66	6,40	6.32	3.16	0	1.41	2.00
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	1.41
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

B1: Chọn **P1**

B2: Các điểm lân cận trong Eps = 1.9 của P1:

$$N_{\text{eps}=1.9}(P1) = \{P2, P10\}$$
$$|N_{1.9}(P1)| < MinPts$$

⇒ P1 không phải core point

⇒ Tiếp tục với các điểm khác

3. Thuật toán DBSCAN

VD8:

Eps=1.9 và MinPts = 4

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0	**1.41**	2.83	4.24	5.66	5.83	6.40	5,83	4.00	**1.41**	2.00	3.16
P2	**1.41**	0	**1.41**	2.83	4.24	4.47	5.00	4.47	3.16	2.00	**1.41**	2.83
P3	2.83	**1.41**	0	**1.41**	2.83	3.16	3.61	3.16	2.83	3.16	2.00	3.16
P4	4.24	2.83	**1.41**	0	**1.41**	2.00	2.24	2.00	3.16	4.47	3.16	4.00
P5	5.66	4.24	2.83	**1.41**	0	**1.41**	**1.00**	**1.41**	4.00	5.83	4.47	5.10
P6	5.83	4.47	3.16	2.00	**1.41**	0	**1.00**	2.83	3.16	5.66	4.24	4.47
P7	6.40	5.00	3.61	2.24	**1.00**	**1.00**	0	2.24	4.12	6.40	5.00	5.39
P8	5.83	4.47	3.16	2.00	**1.41**	2.83	2.24	0	5.10	6.32	5.10	6.00
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0	3.16	2.00	**1.41**
P10	**1.41**	2.00	3.16	4.47	5.83	5.66	6,40	6.32	3.16	0	**1.41**	2.00
P11	2.00	**1.41**	2.00	3.16	4.47	4.24	5.00	5.10	2.00	**1.41**	0	**1.41**
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	**1.41**	2.00	**1.41**	0

Tương tự với các điểm khác

3. Thuật toán DBSCAN

VD8:

Bước 2: Các điểm lân cận trong Eps = 1.9, so sánh MinPts

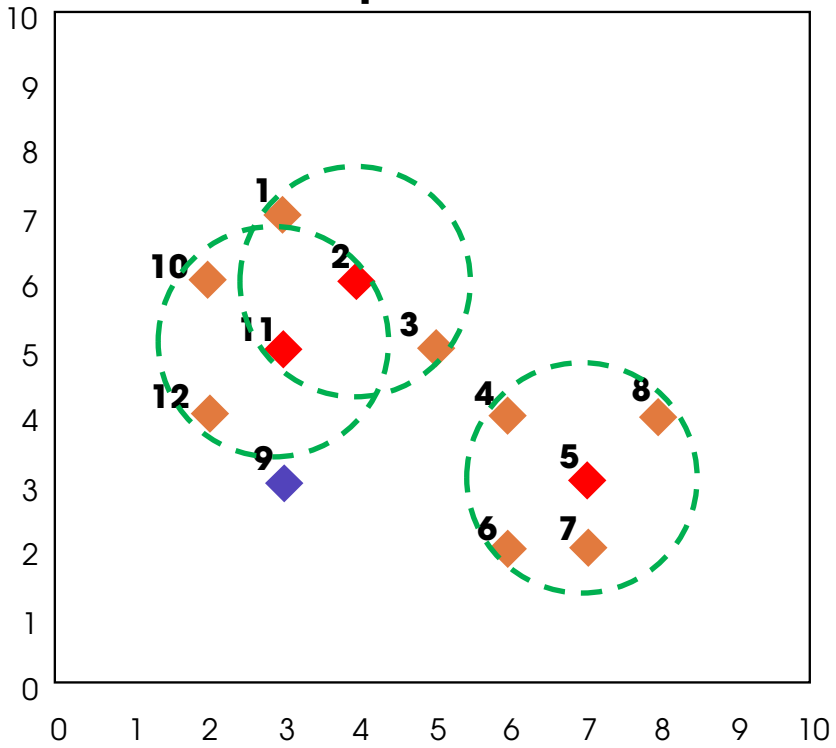
- $N_{eps=1.9}(P1) = \{P2, P10\}$
- $N_{eps=1.9}(P2) = \{P1, P3, P11\}$
- $N_{eps=1.9}(P3) = \{P2, P4\}$
- $N_{eps=1.9}(P4) = \{P3, P5\}$
- $N_{eps=1.9}(P5) = \{P4, P6, P7, P8\}$
- $N_{eps=1.9}(P6) = \{P5, P7\}$
- $N_{eps=1.9}(P7) = \{P5, P6\}$
- $N_{eps=1.9}(P8) = \{P5\}$
- $N_{eps=1.9}(P9) = \{P12\}$
- $N_{eps=1.9}(P10) = \{P1, P11\}$
- $N_{eps=1.9}(P11) = \{P2, P10, P12\}$
- $N_{eps=1.9}(P12) = \{P9, P11\}$

Point	Status
P1	
P2	**Core**
P3	
P4	
P5	**Core**
P6	
P7	
P8	
P9	
P10	
P11	**Core**
P12	

3. Thuật toán DBSCAN

VD8:

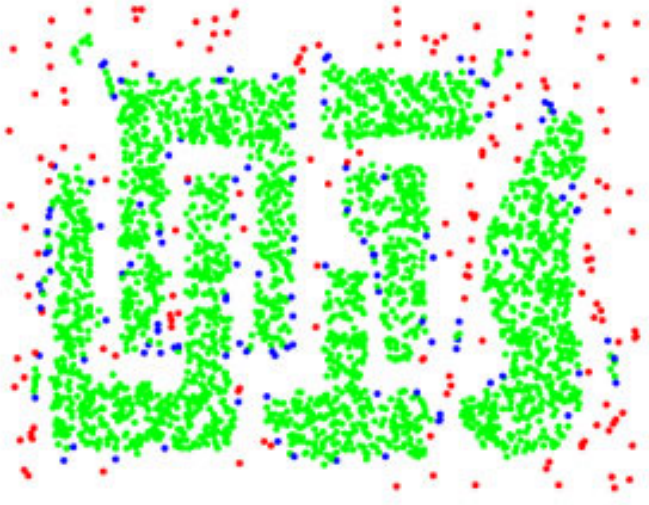
- $N_{1.9}(P1) = \{P2, P10\}$
- $N_{1.9}(P2) = \{P1, P3, P11\}$
- $N_{1.9}(P3) = \{P2, P4\}$
- $N_{1.9}(P4) = \{P3, P5\}$
- $N_{1.9}(P5) = \{P4, P6, P7, P8\}$
- $N_{1.9}(P6) = \{P5, P7\}$
- $N_{1.9}(P7) = \{P5, P6\}$
- $N_{1.9}(P8) = \{P5\}$
- $\boldsymbol{N_{1.9}(P9) = \{P12\}}$
- $N_{1.9}(P10) = \{P1, P11\}$
- $N_{1.9}(P11) = \{P2, P10, P12\}$
- $N_{eps=1.9}(P12) = \{P9, P11\}$



Point	Status
P1	Border
P2	Core
P3	Border
P4	Border
P5	Core
P6	Border
P7	Border
P8	Border
P9	Noise
P10	Border
P11	Core
P12	Border

3. Thuật toán DBSCAN

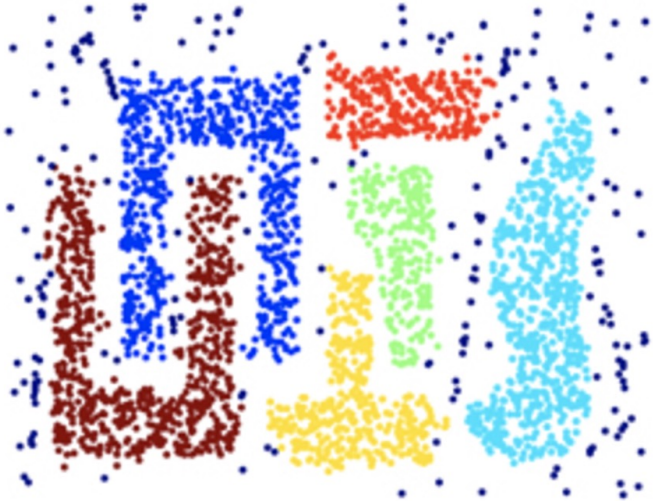
Các đối tượng ban đầu



Các đối tượng:
Core point, border point, noise point

3. Thuật toán DBSCAN

Các đối tượng ban đầu

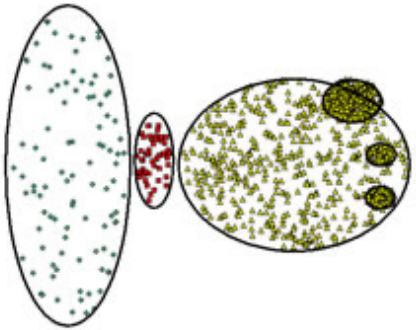


Các cụm

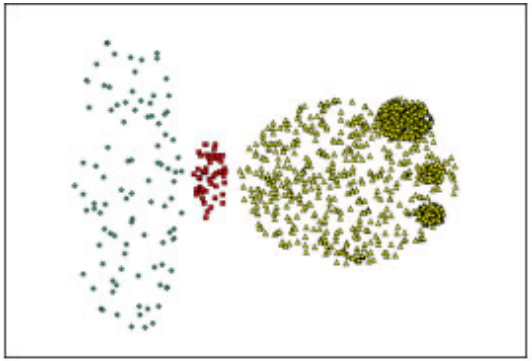
DBSCAN chạy tốt:

- Xử lý được điểm noise, outliers
- Có thể xử lý các cụm có hình dạng và kích cỡ khác nhau

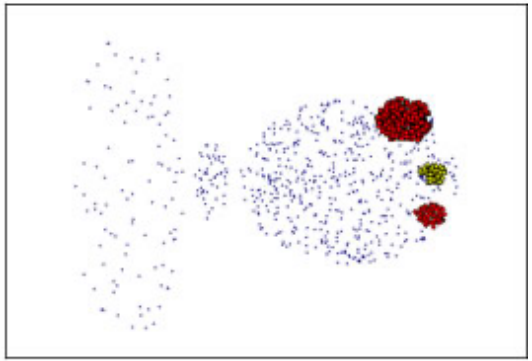
3. Thuật toán DBSCAN



Các đối tượng ban đầu



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

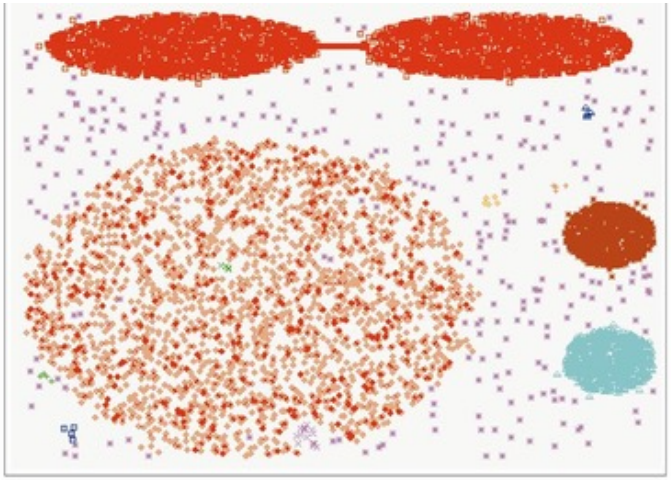
DBSCAN chạy không tốt, gặp vấn đề:

- Mật độ thay đổi, khác nhau
- Dữ liệu nhiều chiều

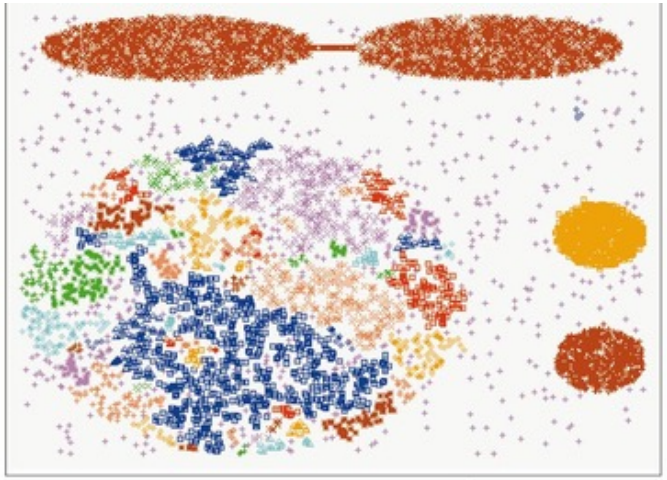
3. Thuật toán DBSCAN

- **Ưu điểm:**
  - Làm việc tốt với dữ liệu nhiễu
  - Có thể giải quyết các trường hợp các cụm có hình dạng và kích thước khác nhau
- **Nhược điểm:**
  - Gặp vấn đề khi các cụm có mật độ khác nhau
  - Độ phức tạp cao đối với dữ liệu nhiều chiều
  - Phụ thuộc vào giá trị Eps, MinPts

3. Thuật toán DBSCAN



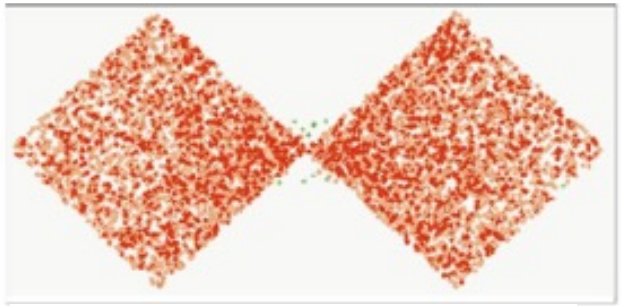
(a) Eps = 0.5



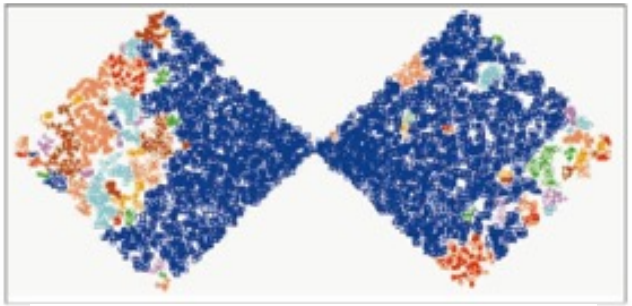
(b) Eps = 0.4

Kết quả DBSCAN cho tập DS1
với MinPts = 4 và Eps = 0.5 (a), Eps = 0.4 (b)

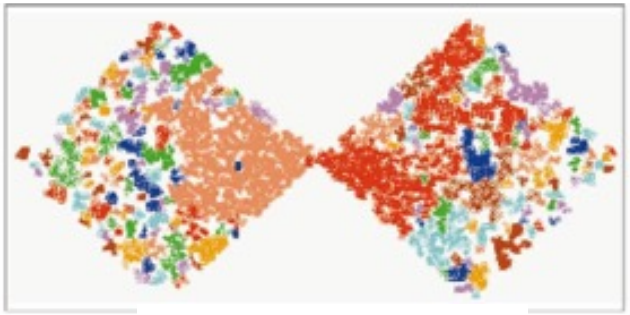
3. Thuật toán DBSCAN



(a) Eps = 5

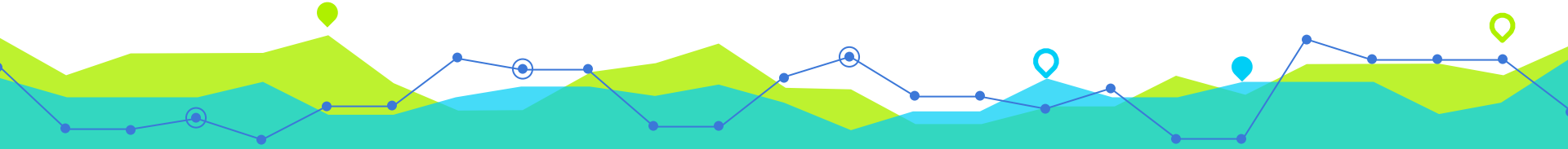


(b) Eps = 3.5



(c) Eps = 3.0

Kết quả DBSCAN cho tập DS1 với MinPts = 4 và Eps = 5.0 (a), Eps = 3.5 (b), Eps = 3.0 (c)

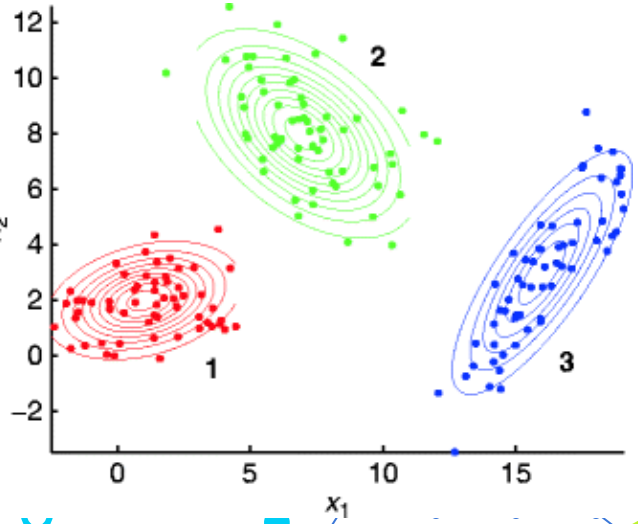


5 Phương pháp dựa trên mô hình

1. Giới thiệu
2. Self-Organizing Map (SOM)

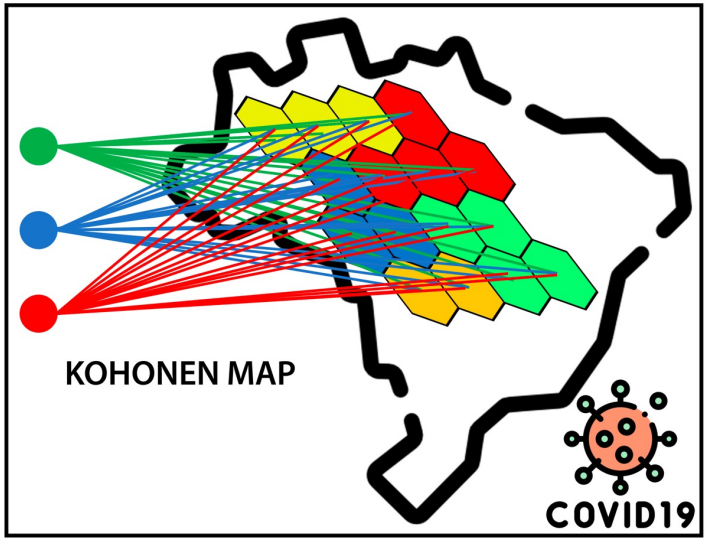
1. Giới thiệu

- Dựa trên sự phù hợp giữa dữ liệu và các mô hình toán học. Một mô hình được đưa ra giả thuyết cho từng cụm và cố gắng tìm ra sự phù hợp nhất của mô hình đó
- Ý tưởng: Dữ liệu phát sinh từ một sự kết hợp nào đó của các phân phối xác suất ẩn
- Một số hướng tiếp cận chính:
  - Thống kê. VD: COBWEB, CLASSIT, AutoClass
  - Xác suất. VD: Fuzzy clustering, EM algorithm
  - Tiếp cận mạng neural. VD: Self-Organizing Map (SOM)

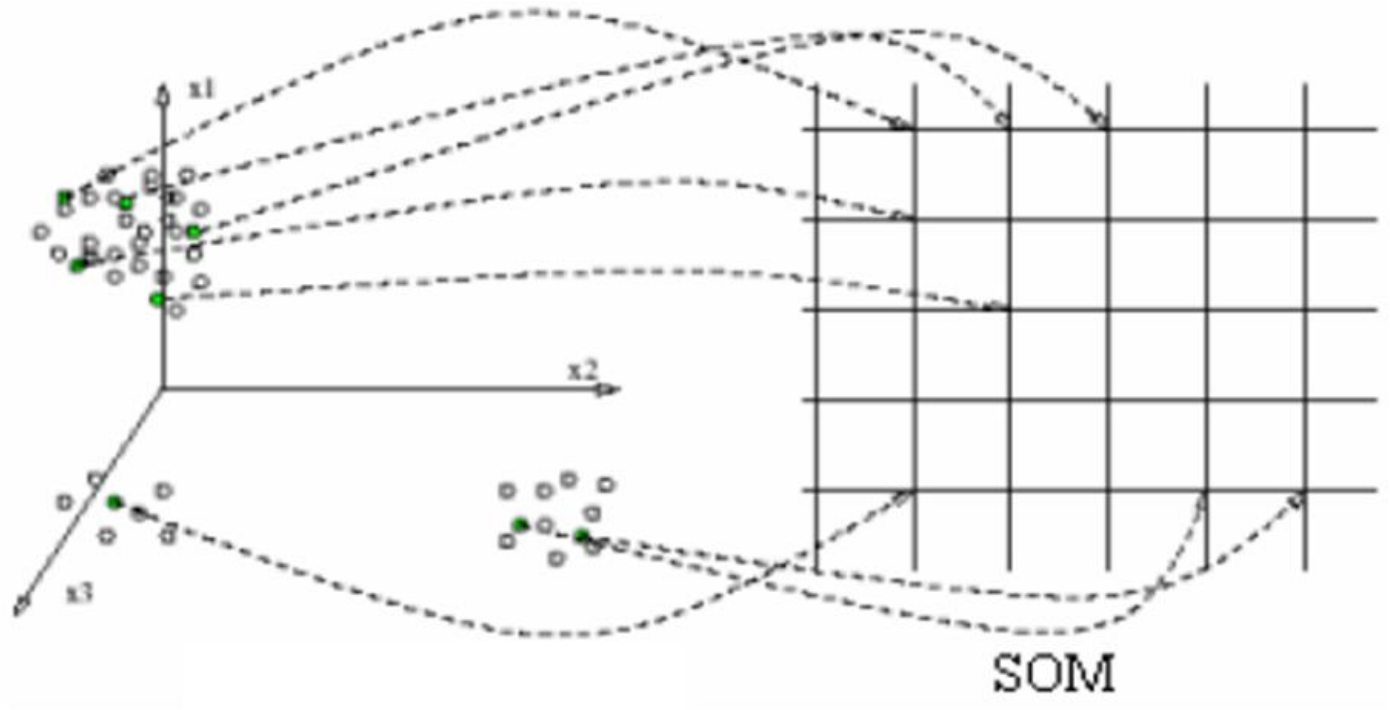


2. Self-Organizing Map (SOM)

- Kohonen giới thiệu những năm1980s. Còn được gọi là mạng Kohonen
- Là 1 ANN truyền thẳng, học không giám sát, thực hiện việc ánh xạ dữ liệu để giảm kích thước dữ liệu đầu vào.
- Từ tập đối tượng nhiều chiều, sử dụng mạng Kohonen có số chiều nhỏ hơn (thường là 2 chiều) để đặc trưng cho tập dữ liệu.

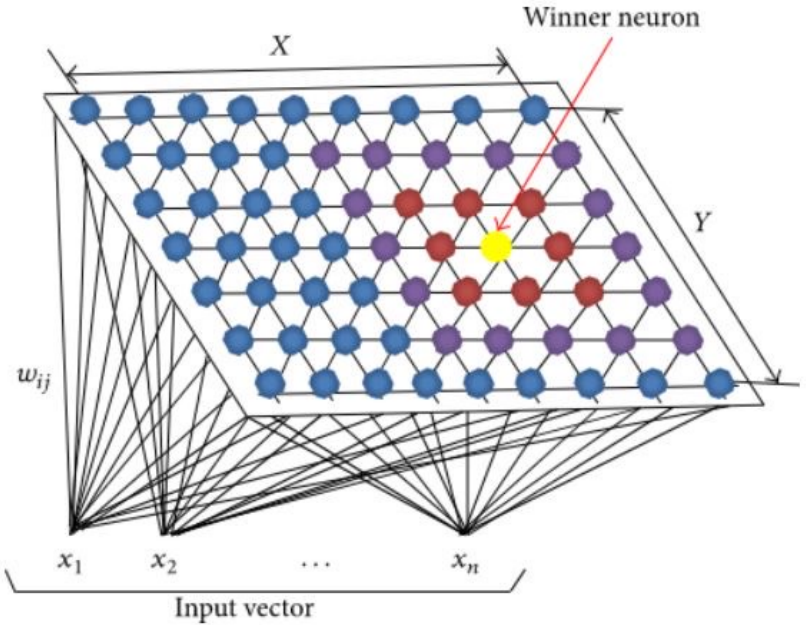


2. Self-Organizing Map (SOM)



2. Self-Organizing Map (SOM)

- **Ý tưởng huấn luyện mạng SOM:** thực hiện n vòng lặp và tại mỗi vòng lặp thực hiện các bước sau:
  - Tìm neural chiến thắng (BMU-Best Matching Unit).
  - Cập nhật trọng số và các lân cận của BMU.
  - Lặp lại cho đến khi không có sự thay đổi nào trên các trọng số hoặc đạt được số lần lặp xác định.



2. Self-Organizing Map (SOM)

- **Input:**
  - Đối tượng gom cụm: tập n các vector đầu vào x (m chiều)
  - Các tham số:
    - Epochs: số lần lặp
    - R: bán kính vùng lân cận
    - α: tốc độ học
- **Output:** Bản đồ Kohonen với k neural (~ k cụm)

2. Self-Organizing Map (SOM)

- **B1:** Khởi tạo các giá trị
  - Khởi tạo giá trị cho k neural
    - Mỗi neural có 1 vector trọng số $w_i = (w_{i1}, w_{i2}, .. w_{im})$
    - k: số cụm, i=1,…, k
    - m: số chiều.
  - Gán giá trị R, epochs và α

2. Self-Organizing Map (SOM)

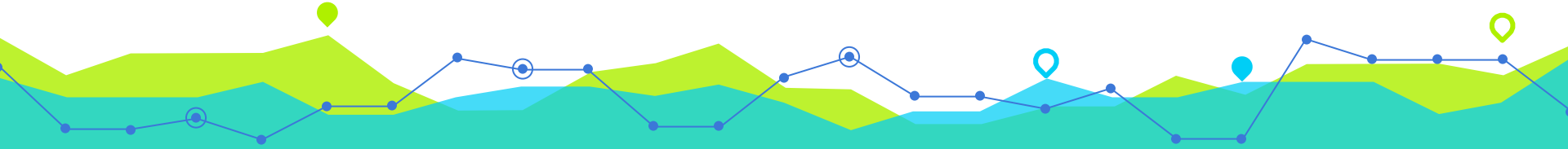
- **B2:** Với mỗi vector đối tượng x trong tập dữ liệu:
  - Tính khoảng cách Euclide của từng neural w_i đến x
  - Tìm neural J sao cho khoảng cách từ x đến J là ngắn nhất
  - Duyệt qua tất cả các vector trọng số thuộc láng giềng của J (trong bán kính R), cập nhật vector trọng số:

$$w_{ij}(new) = w_{ij}(old) + \alpha(x_j - w_{ij}(old))$$

2. Self-Organizing Map (SOM)

- **B3:** Cập nhật lại tốc tốc độ học α và bán kính R.
- **B4:** Dừng nếu thỏa điều kiện dừng (α rất nhỏ hoặc đủ số epoch hoặc vector trọng số hội tụ).

  Ngược lại, lặp lại từ bước 2.



6 Đánh giá chất lượng cụm

1. Đánh giá chất lượng cụm
2. Độ đo, tiêu chuẩn đánh giá chất lượng cụm
3. Phương pháp ước tính độ chính xác

1. Đánh giá chất lượng cụm

- Là nhiệm vụ khó khăn và phức tạp trong phân tích nhóm
- Chất lượng cụm thể hiện qua:
  - Xác định được xu hướng gom cụm của dữ liệu, có cấu trúc trong dữ liệu hay không.
  - Đánh giá ngoài: So sánh kết quả gom cụm với các cấu trúc cụm đã biết. VD: dựa vào thuộc tính phân lớp đã có
  - Đánh giá trong: không sử dụng thông tin bên ngoài, phân tích sự phù hợp của kết quả phân cụm, tìm đặc trưng cụm
  - So sánh kết quả của 2 phương pháp gom cụm khác nhau
  - Xác định chính xác số cụm

Xác định số lượng cụm

- **Thực nghiệm:** VD: k = $\sqrt{n/2}$, với n: số điểm , mẫu trong CSDL
- **Elbow method:** Sử dụng điểm ngoặt trong đường cong của tổng phương sai trong cụm với số cụm khác nhau.
- **Cross validation method:**
  - Chia tập dữ liệu thành m phần
  - Sử dụng m – 1 phần để xây dựng mô hình phân cụm
  - Sử dụng phần còn lại để kiểm thử chất lượng của phân cụm. VD: Đối với mỗi điểm trong tập test, tìm trọng tâm gần nhất và sử dụng SSE giữa tất cả các điểm trong tập test và trọng tâm gần nhất để đo mức độ phù hợp của mô hình với tập test
  - Đối với bất kỳ k > 0, lặp lại m lần, so sánh độ đo chất lượng tổng thể với k khác nhau và tìm số cụm phù hợp nhất với dữ liệu

2. Độ đo, tiêu chuẩn đánh giá chất lượng cụm

- **Chỉ số ngoài (External index):**
  - So sánh các cụm thu được với các lớp đã có sẵn
  - VD: Entropy
- **Chỉ số trong (Internal index):**
  - Đo chất lượng cụm thu được, không sử dụng thông tin bên ngoài.
  - VD: SSE (Sum of Squared Error), Hệ số dáng điệu (silhouette coefficient)
- **Chỉ số tương đối (Relative index):**
  - So sánh 2 phương pháp gom cụm, hoặc so sánh các nhóm
  - VD: SSE, Entropy

2.1. Internal index: SSE và BSS

- **SSE – Sum of Squared Error:**
  - Thường được dùng để so sánh 2 phương pháp gom cụm hoặc 2 cụm
  - Đo sự liên kết, tính gắn kết của các đối tượng trong cùng 1 cụm

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x) \quad (4)$$

  - x: một điểm dữ liệu trong cụm C_i
  - m_i: trọng tâm (centroid) hoặc trung điểm (medoid) của cụm C_i
  - k: số cụm
  - dist(): khoảng cách Euclide

2.1. Internal index: SSE và BSS

- **BSS – Between Cluster Sum of Squares:**
  - Đo sự phân biệt, tính tách biệt của một cụm so với các cụm khác

$$BSS = \sum_{i=1}^{k} |C_i|(m - m_i)^2 \quad (4)$$

  - m_i: trọng tâm (centroid) hoặc trung điểm (medoid) của cụm C_i
  - k: số cụm

2.2. Internal index: Hệ số dáng điệu

- Hệ số dáng điệu (silhouette coefficient) kết hợp cả 2 yếu tố tính gắn kết và tính tách biệt, dùng cho từng điểm, từng cụm và cả phân cụm.
- Cho một điểm i
  - Tính a = trung bình khoảng cách từ i đến các điểm khác trong cụm
  - Tính b = min (trung bình khoảng cách từ i đến các điểm của cụm khác)
  - Hệ số dáng điệu: $$s = \frac{b - a}{\max(a, b)}$$
  - $s \in [0,1]$, s càng gần 1 chất lượng phân cụm càng tốt
- Có thể tính hệ số dáng điều trung bình cho 1 cụm hoặc cho cả một phân cụm

2.2. External index: Entropy, Purity

- Xây dựng bảng thống kê chéo số phần tử của từng cụm thuộc từng lớp đã có sẵn (k cụm, L lớp)
- Với mỗi cụm j tính xác suất để một phần tử thuộc cụm j (cluster j) thuộc về lớp i (class i):

$$p_{ij} = \frac{|m_{ij}|}{|m_j|}$$

- $|m_{ij}|$: Số mẫu của class i thuộc cluster j
- $|m_j|$: Số mẫu của cluster j

2.2. External index: Entropy, Purity

- **Entropy của cluster j:**

$$e_j = -\sum_{i=1}^{L} p_{ij} \log_2(p_{ij})$$

- **Tổng entropy:**

$$e = \sum_{j=1}^{k} \frac{|m_j|}{|m|} e_j$$

- L: Số lớp (class)
- K: Số cụm (cluster)
- $|m_j|$: Số mẫu của cluster j
- $|m|$: Tổng số mẫu

2.2. External index: Entropy, Purity

- **Purity (Độ thuần nhất) của cluster j:**

$$purity_j = \max_i p_{ij}$$

- **Tổng purity:**

$$purity = \sum_{j=1}^{k} \frac{|m_j|}{|m|} purity_j$$

- K: Số cụm (cluster)
- $|m_j|$: Số mẫu của cluster j
- $|m|$: Tổng số mẫu

Đánh giá chất lượng cụm

VD9: Kết quả gom cụm sử dụng k-mean cho tập dữ liệu bài báo LA. Tính tổng etropy và purity

Cluster / class	Entertainment	Financial	Foreign	Metro	National	Sports	**Total Cluster**
1	3	5	40	506	96	27	**677**
2	4	7	280	29	39	2	**361**
3	1	1	1	7	4	671	**685**
4	10	162	3	119	73	2	**369**
5	331	22	5	70	13	23	**464**
6	5	358	12	212	48	13	**648**
Total Class	**354**	**555**	**341**	**943**	**273**	**738**	**3204**

Đánh giá chất lượng cụm

VD9:

Ma trận xác suất

Cluster/ class	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	0.0044	0.0074	0.0591	0.7474	0.1418	0.0399	1.2270	0.7474
2	0.0111	0.0194	0.7756	0.0803	0.1080	0.0055	1.1472	0.7756
3	0.0015	0.0015	0.0015	0.0102	0.0058	0.9796	0.1813	0.9796
4	0.0271	0.4390	0.0081	0.3225	0.1978	0.0054	1.7487	0.4390
5	0.7134	0.0474	0.0108	0.1509	0.0280	0.0496	1.3976	0.7134
6	0.0077	0.5525	0.0185	0.3272	0.0741	0.0201	1.5523	0.5525

VD:

$p_{Financial,\ 2} = 7/361 = 0.0194$

$Entropy_2 = -\sum_{i=1}^{L} p_{i2} \log_2(p_{i2}) = 1.1472$

$Purity_2 = \max_i p_{i2} = 0.7756$

Entropy = $\sum_{j=1}^{k} \frac{|m_j|}{|m|} e_j$ **= 1.1450**

Purity = $\sum_{j=1}^{k} \frac{|m_j|}{|m|} purity_j$ **= 0.7203**

Đánh giá chất lượng cụm

BT17

Cho DL sau

1. Chuẩn hóa dữ liệu, sử dụng k-means với k = 2 và ma trận phân hoạch sau để xác định các nhóm (không dùng cột response). Tính độ đo SSE ở vòng lặp đầu tiên và vòng lặp cuối cùng
2. Tính độ đo entropy và purity cho 2 cụm tạo ra từ câu 1

Customer	Age	Income (k)	No. Card	Response
Lâm	35	35	3	Yes
Hưng	22	50	2	No
Mai	28	40	1	Yes
Lan	45	100	2	No
Thủy	20	30	3	Yes
Tuấn	34	55	2	No
Minh	63	200	1	No
Vân	55	140	2	No
Thiện	59	170	1	No
Ngọc	25	40	4	Yes

M_0	Lâm	Hưng	Mai	Lan	Thủy	Tuấn	Minh	Vân	Thiện	Ngọc
C_1	1	1	1	1	1	0	0	0	0	0
C_2	0	0	0	0	0	1	1	1	1	1

Tổng kết chương

Tổng quan về Gom cụm dữ liệu

1. Gom cụm là gì
2. Tiêu chuẩn gom cụm
3. Độ đo khoảng cách
4. Yêu cầu và thách thức
5. Một số phương pháp gom cụm

Phương pháp phân hoạch

1. Khái niệm cơ bản
2. Thuật toán K-means
3. Thuật toán K-medoids: PAM

Phương pháp phân cấp

1. Giới thiệu
2. Thuật toán AGNES
3. Thuận toán DIANA

Tổng kết chương

Phương pháp dựa trên mật độ

1. Giới thiệu
2. Các khái niệm cơ bản
3. Thuật toán DBSCAN

Phương pháp dựa trên mô hình

1. Giới thiệu
2. Self-Organizing Map (SOM)

Đánh giá chất lượng cụm

1. Đánh giá chất lượng cụm
2. Độ đo, tiêu chuẩn đánh giá chất lượng cụm
3. Phương pháp ước tính độ chính xác

Bài tập chương 7

7.1. Cho tập dữ liệu các điểm sau:

Sử dụng thuật toán K-means để phân cụm dữ liệu với số cụm là 3

	A	B
P1	3	6
P2	5	4
P3	1	8
P4	1	10
P5	7	8
P6	8	8
P7	2	9

Bài tập chương 7

7.2. Cho tập dữ liệu các điểm sau:

Sử dụng thuật toán K-means để phân cụm dữ liệu với số cụm là 3. Tính SSE cho kết quả cuối cùng.

	A	B
X1	0.7	0.45
X2	2.8	1
X3	2.6	1
X4	1	0.8
X5	2.5	1.2
X6	1.3	1.4
X7	0.4	0.7
X8	1.7	1.8
X9	2	2

Bài tập chương 7

7.3. Cho tập dữ liệu các điểm sau:

1. Sử dụng thuật toán K-means để phân cụm dữ liệu với số cụm là 3.

2. Chuẩn hóa CSDL và gom cụm với k=3. So sánh với kết quả câu 1.

Customer	Age	Income (k)	No. Card
Thảo	35	37	3
Hưng	25	51	3
Gia	29	44	1
Thành	45	100	3
Thủy	20	30	4
Đức	33	57	2
Minh	65	200	1
Nhung	54	142	2
Nhật	58	175	1
Tùng	25	40	5

Bài tập chương 7

7.4. Cho tập dữ liệu gồm 5 điểm với ma trận khoảng cách sau:

1. Sử dụng thuật toán AGNES lần lượt với Single Link và Complete Link để gom cụm. Vẽ sơ đồ hình cây.
2. Xác định 3 cụm thu được từ sơ đồ hình cây từ cả 2 cách.

	P1	P2	P3	P4	P5
P1	0				
P2	0.10	0			
P3	0.41	0.64	0		
P4	0.55	0.47	0.44	0	
P5	0.35	0.98	0.85	0.76	0

Bài tập chương 7

7.4. Cho tập dữ liệu gồm 5 điểm với ma trận khoảng cách sau:

1. Sử dụng thuật toán AGNES lần lượt với Single Link và Complete Link để gom cụm. Vẽ sơ đồ hình cây.
2. Xác định 3 cụm thu được từ sơ đồ hình cây từ cả 2 cách.

	P1	P2	P3	P4	P5
P1	0				
P2	0.10	0			
P3	0.41	0.64	0		
P4	0.55	0.47	0.44	0	
P5	0.35	0.98	0.85	0.76	0

Bài tập chương 7

7.5. Cho ma trận hỗn loạn sau. Tính độ đo entropy và purity

Cluster / class	Entertainment	Financial	Foreign	Metro	National	Sports	**Total Cluster**
1	**1**	**1**	**0**	**11**	**4**	**676**	**693**
2	**27**	**89**	**333**	**827**	**253**	**33**	**1562**
3	**326**	**465**	**8**	**105**	**16**	**29**	**949**
Total Class	**354**	**555**	**341**	**943**	**273**	**738**	**3204**

THANKS!

Any questions?