Chương 3.
Học không giám sát

Ts. Nguyễn An Tế

Khoa CNTT kinh doanh – ĐH Kinh tế TPHCM

tena@ueh.edu.vn

2025

Tài liệu tham khảo

Brown M.-S., *Data Mining for Dummies*, For Dummies, 2014.

Provost F. and Fawcett T., *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*, O'Reilly Media, 2013.

Shmueli G. et al., *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*, Wiley, 2017.

Vũ Hữu Tiệp, *Machine Learning cơ bản*, 2018.

Xu R., *Survey of Clustering Algorithms*, IEEE Transactions on Neural Networks, 2005.

Nội dung

1. Gom cụm dữ liệu
2. Một số phương pháp gom cụm

1. Gom cụm dữ liệu (Clustering)

- Quản lý khách hàng

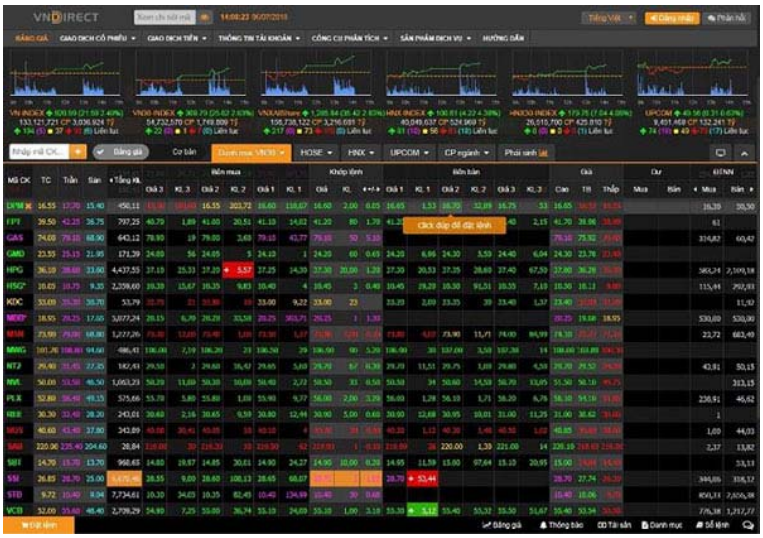
customer profile:
target feature ?

phân lớp ?

1. Gom cụm dữ liệu

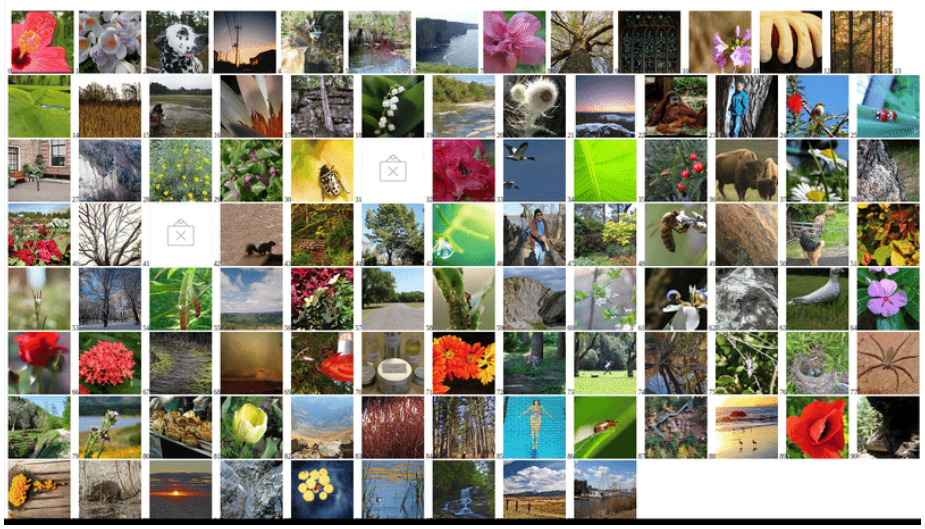
- Thị trường chứng khoán: những nhóm cổ phiếu có xu thế biến động giống nhau ?

phân lớp ?



1. Gom cụm dữ liệu

- Tổ chức album ảnh → phân lớp đa nhãn ?



1. Gom cụm dữ liệu

- Tìm kiếm thông tin: những tài liệu tương tự nhau

http://www.cit.ctu.edu.vn › ~dtnghi › dataminingR ▾ PDF

Giải thuật gom cụm Clustering algorithms - cit.ctu.edu.vn

Dec 2, 2008 — như lớp (nhãn). ▫ gom nhóm : mô hình **gom cụm** dữ liệu (không có nhãn) sao ... có nhiều nhóm giải thuật khác nhau : hierarchical **clustering**,.

Missing: ~~UEH~~ | Must include: UEH

http://scholar.vimaru.edu.vn › files › thinhnv › files ▾ PDF

Chương 5: Gom cụm dữ liệu

Adapting the Right Measures for K-means **Clustering**. KDD'09, Paris, France, July 2009. Page 42. 42. 42.

Missing: ~~UEH~~ | Must include: UEH

https://viblo.asia › hierarchical-cluste... ▾ Translate this page

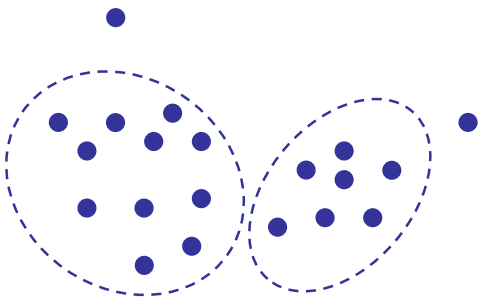
Hierarchical clustering - Phân cụm dữ liệu - Viblo

Feb 17, 2020 — Phân **cụm là** gì? Phân **cụm** dữ liệu **là** bài toán **gom** nhóm các đối tượng dữ liệu vào thánh từng **cụm** (**cluster**) sao cho các đối tượng trong cùng ...

Missing: ~~UEH~~ | Must include: UEH

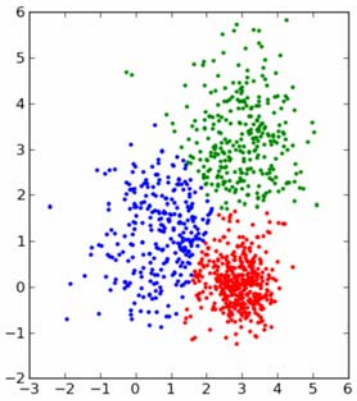
1. Gom cụm dữ liệu

- Nhận diện những dữ liệu bất thường (*outliers*)



1. Gom cụm dữ liệu

- **Cụm** (*cluster*): **tập hợp những đối tượng (dữ liệu)**
  - sự TƯƠNG ĐỒNG cao giữa những phần tử trong cùng cụm
  - sự KHÁC BIỆT lớn với những phần tử trong các cụm khác



1. Gom cụm dữ liệu

- **Biểu diễn các đối tượng $\{x_i\}_{i=1}^{m}$ bằng các vectơ → ma trận**

features

$$\begin{array}{c} \\ x_1 \rightarrow \\ x_2 \rightarrow \\ \cdots \rightarrow \\ x_i \rightarrow \\ \cdots \rightarrow \\ x_m \rightarrow \end{array} \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & & x_{mj} & & x_{mn} \end{pmatrix}$$

1. Gom cụm dữ liệu

- **Cụm** (*cluster*) **– Các đại lượng đặc trưng (dữ liệu định lượng)**
  - trọng tâm (*centroid*):

$$C = \frac{1}{m}\sum_{i=1}^{m} x_i$$

  - bán kính (*radius*):

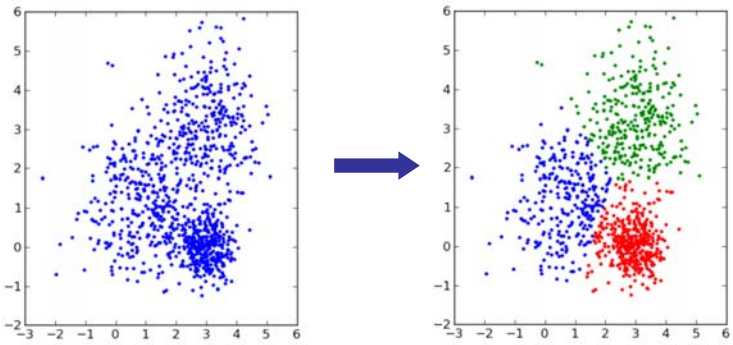
$$R = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - C)^2}$$

  - đường kính (*diameter*):

$$D = \sqrt{\frac{1}{m(m-1)}\sum_{i\neq j}(x_i - x_j)^2}$$

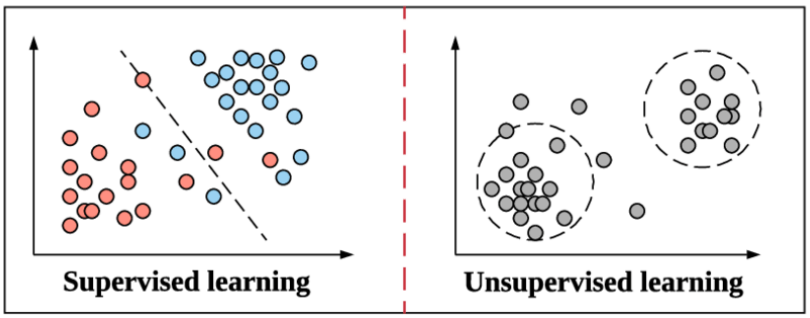
1. Gom cụm dữ liệu

- **Gom cụm** (*Clustering*, *Data Segmentation*)**: tạo "phân hoạch"**
  - dựa trên sự tương đồng (sự khác biệt) giữa các đối tượng
  - *Unsupervised Learning*: không có các lớp được xác định trước (*learning by observations*)



1. Gom cụm dữ liệu

- **Gom cụm** (*Clustering*, *Data Segmentation*)**: tạo "phân hoạch"**
  - các clusters có thể rời nhau hoặc không rời nhau
  - có thể phải cần đến những chuyên gia trong các lãnh vực để diễn giải ý nghĩa của các clusters kết quả



dự đoán mô tả

1. Gom cụm dữ liệu

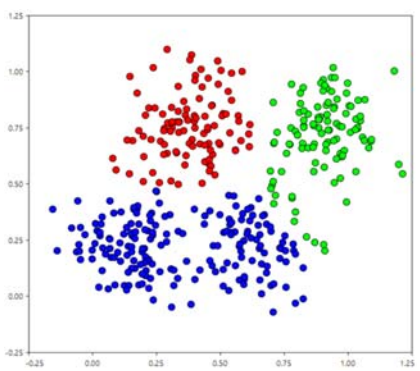
- **Gom cụm** (*Clustering*, *Data Segmentation*) **– Ứng dụng thực tế**
  - kinh doanh, tiếp thị (*Business Intelligence*)
  - nghiên cứu xã hội
  - tìm kiếm thông tin (*Web search, Information Retrieval*): clustering trên tập kết quả hay trên kho dữ liệu
  - sinh học
  - địa chất
  - khí tượng
  - . . .

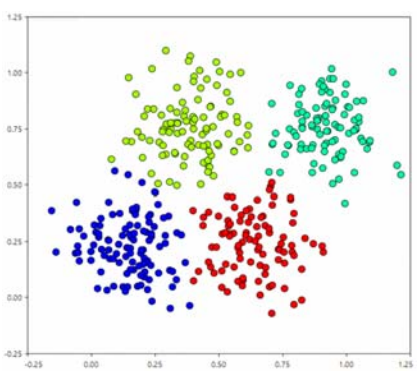
1. Gom cụm dữ liệu

- **Gom cụm** (*Clustering*, *Data Segmentation*) **– Tiền xử lý dữ liệu cho những thuật toán khác**
  - phát hiện outliers
  - summarization: hồi quy, PCA, …
  - compression: xử lý ảnh
  - . . .

1. Gom cụm dữ liệu

- **Chất lượng của clustering: các tiêu chí chung**
  - *intra-class similarity* ↗: sự kết dính (*cohesion*) trong cluster
  - *extra-class similarity* ↘: sự khác biệt (*distinction*) giữa clusters





1. Gom cụm dữ liệu

- **Chất lượng của clustering: các yếu tố**
  - độ đo mức độ tương đồng (*similarity measure*)
  - khả năng phát hiện những dạng thức (*pattern*) tiềm ẩn

1. Gom cụm dữ liệu

- **Sự tương đồng giữa 2 đối tượng**
  - *similarity*
  - *dissimilarity*, *distance*
  - *proximity*: similarity, dissimilarity

1. Gom cụm dữ liệu

- **Sự tương đồng giữa 2 đối tượng: n thuộc tính định lượng**

$$x = (x_1, x_2, ..., x_n) \quad y = (y_1, y_2, ..., y_n)$$

- cosine:

$$sim(x, y) = \frac{\sum_{i=1}^{n} x_i . y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

- tích vô hướng (*scalar product*):

$$sim(x, y) = \sum_{i=1}^{n} x_i . y_i$$

1. Gom cụm dữ liệu

- **Khoảng cách giữa 2 đối tượng: n thuộc tính định lượng**

- khoảng cách *Minkowski*:

$$d(x, y) = \sum_{i=1}^{n} \left(|x_i - y_i|^p \right)^{1/p}$$

- khoảng cách *Manhattan* (*L_1 norm*): (p = 1)

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- khoảng cách *Euclid* (*L_2 norm*): (p = 2)

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}$$

1. Gom cụm dữ liệu

- **Khoảng cách giữa 2 đối tượng: n thuộc tính định lượng**
  - khoảng cách *Chebyshev* (*L_{max} norm*, *Chessboard distance*): ($p \to \infty$)

$$d(x,y) = \max_i |x_i - y_i|$$

  - khoảng cách *Canberra*:

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

1. Gom cụm dữ liệu

- **Sự tương đồng giữa 2 đối tượng: n thuộc tính định danh**
  - xem đối tượng như tập hợp các giá trị thuộc tính

$$sim(x,y) = \frac{|x \cap y|}{n} = \frac{\sum_{i=1}^{n} e(x_i, y_i)}{n} \qquad e(x_i, y_i) = \begin{cases} 1, & x_i = y_i \\ 0, & x_i \neq y_i \end{cases}$$

  - có thể sử dụng trọng số:

$$sim(x,y) = \frac{\sum_{i=1}^{n} w_i . e(x_i, y_i)}{n} \qquad w_i = |DOM(A_i)|$$

1. Gom cụm dữ liệu

- **Sự tương đồng giữa 2 đối tượng: n thuộc tính định tính**
  - "số hóa" các giá trị thuộc tính của x và y:

$$DOM(A_i) = \{a_1, a_2, ..., a_p \,|\, a_i < a_j, \forall i < j\}$$

$$rank(a_i) = i$$

$$x_i = \frac{rank(x_i) - 1}{p - 1} \qquad y_i = \frac{rank(y_i) - 1}{p - 1}$$

1. Gom cụm dữ liệu

- **Khoảng cách giữa 2 đối tượng: nhiều loại thuộc tính**

$$d(x,y) = \frac{\sum_{i=1}^{n} \delta_i(x,y).d_i(x,y)}{\sum_{i=1}^{n} \delta_i(x,y)}$$

  - $\delta_i(x,y) = 0$ nếu (x_i hay y_i bị thiếu) hoặc ($x_i = y_i = 0$)

    Ngược lại: $\delta_i(x,y) = 1$
  - A_i định danh: $d_i(x,y) = 0$ nếu ($x_i = y_i$); ngược lại: $d_i(x,y) = 1$

    A_i định tính: "số hóa" theo rank()

    A_i định lượng: $d_i(x,y) = \frac{|x_i - y_i|}{\max(DOM(A_i)) - \min(DOM(A_i))}$

1. Gom cụm dữ liệu

- **Ma trận khoảng cách** (*distance/dissimilarity matrix*)

$$D_{ij} = d(x_i, x_j)$$

  - ma trận đối xứng hoặc ma trận tam giác
  - ma trận không âm (*non-negative matrix*)
  - $D_{ii} = 0$

$$D_{mxm} = \begin{pmatrix} 0 & d(x_1,x_2) & \cdots & d(x_1,x_j) & \cdots & d(x_1,x_m) \\ & 0 & \cdots & d(x_2,x_j) & \cdots & d(x_2,x_m) \\ & & 0 & \cdots & \cdots & \cdots \\ & & & 0 & \cdots & d(x_i,x_m) \\ & & & & 0 & \cdots \\ & & & & & 0 \end{pmatrix}$$

1. Gom cụm dữ liệu

- **Chuẩn của ma trận** (*matrix norm*)
  - *p-norm*:

$$\|A\|_p = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right)^{1/p}$$

  - *Frobenius norm*: (p = 2)

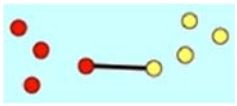
$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

VD: Thuật toán *Fuzzy C-Means*

1. Gom cụm dữ liệu

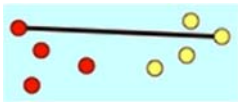
- **Hàm khoảng cách** (*distance function*)**: giữa 2 clusters**
  - *Single-link*: kh.cách ngắn nhất giữa 2 items của 2 clusters

$$D(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



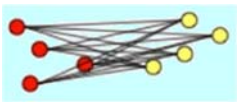
  - *Complete-link*: kh.cách dài nhất giữa 2 items của 2 clusters

$$D(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} d(x_i, x_j)$$



  - *Average-link*: kh.cách trung bình giữa 2 items của 2 clusters

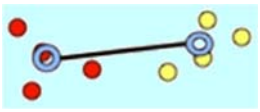
$$D(C_i, C_j) = \frac{1}{|C_i| . |C_j|} \sum_{\substack{x_i \in C_i \\ x_j \in C_j}} d(x_i, x_j)$$



1. Gom cụm dữ liệu

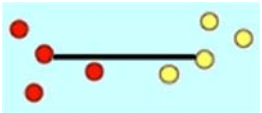
- **Hàm khoảng cách** (*distance function*)**: giữa 2 clusters**
  - *Centroids*: kh.cách giữa 2 trọng tâm của 2 clusters

$$D(C_i, C_j) = d\left(\frac{1}{|C_i|} \sum_{x \in C_i} x, \frac{1}{|C_j|} \sum_{y \in C_j} y \right)$$



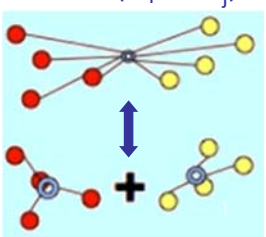
  - *Medoids*: kh.cách giữa 2 đối tượng trung tâm của 2 clusters

$$D(C_i, C_j) = d(M_i, M_j)$$



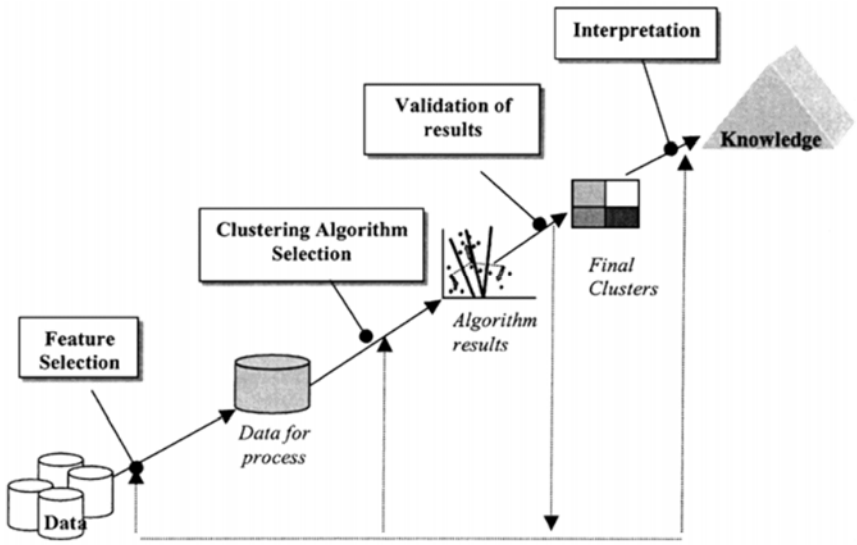
  - *Ward's method*: gộp 2 clusters → xét trọng tâm của ($C_i \cup C_j$)

$$D(C_i, C_j) = \sum_{x \in C_i \cup C_j} d(x, \mu_{C_i \cup C_j})$$



1. Gom cụm dữ liệu

- **Quy trình gom cụm**



1. Gom cụm dữ liệu

- **Gom cụm** (*Clustering*, *Data Segmentation*) **– Một số vấn đề**
  - *scalability*: thuật toán xử lý trên dataset → mức độ chính xác có bị ảnh hưởng bởi quy mô của dữ liệu ?
  - *data types*: text (nominal, ordinal, numerical), images, video, …
  - *complex shapes*: nonconvex, …
  - *visualization*: trực quan hóa (*high dimensionality*)
  - *interpretability*: diễn dịch kết quả gom cụm → tính hữu dụng
  - *noisy data*: dữ liệu nhiễu
  - . . .

Nội dung

1. Gom cụm dữ liệu

2. **Một số phương pháp gom cụm**
   - Cách tiếp cận dựa trên phân hoạch
   - Cách tiếp cận dựa trên phân cấp
   - Cách tiếp cận dựa trên mật độ
   - Cách tiếp cận dựa trên lưới

2.1 Cách tiếp cận dựa trên phân hoạch

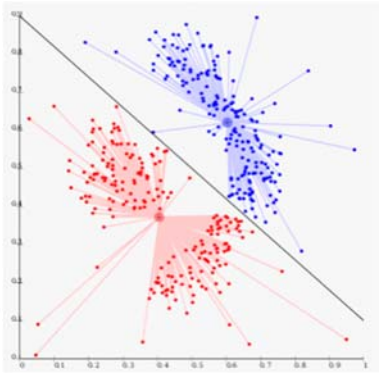
❑ **Cách tiếp cận dựa trên sự phân hoạch** (*Partitioning Approach*)
- dựa trên sự tương đồng, sự khác biệt giữa các đối tượng
- cần xác định số clusters trong phân hoạch
- *k-Means*, *k-Medoids*, *Fuzzy C-Means*
- *CLARA*, *CLARANS*, … (đọc thêm)

2.1 Cách tiếp cận dựa trên phân hoạch

- **Tập dữ liệu D được phân hoạch thành k clusters**
  - xác định một điểm c_i là "đặc trưng" cho mỗi cluster C_i
  - chọn phân hoạch tốt nhất: tổng bình phương khoảng cách (*Sum of Squared Error*) từ c_i đến mọi $x \in C_i$ của tất cả k clusters là nhỏ nhất:

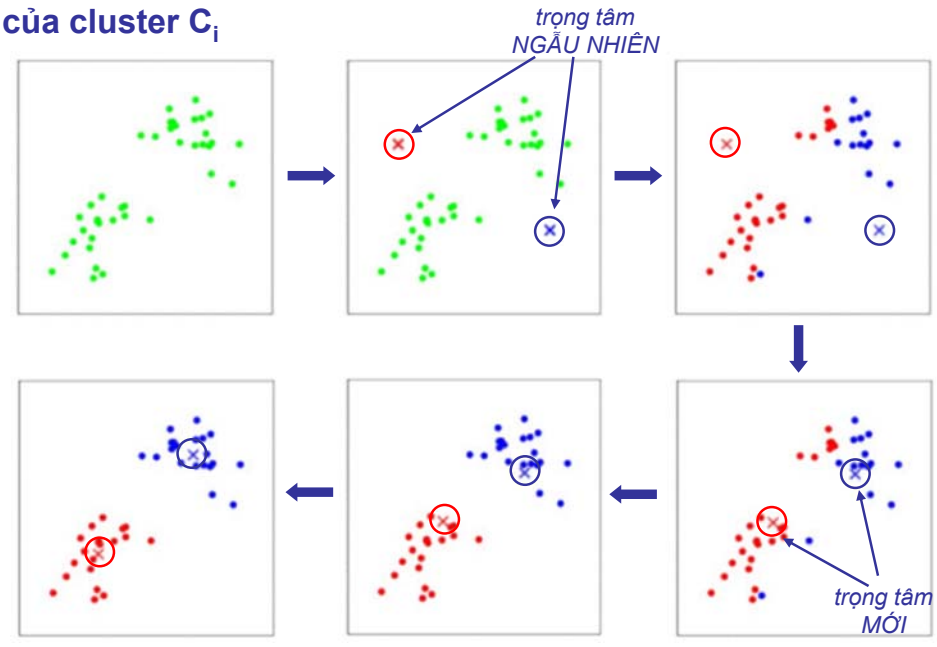
$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)^2$$

  - tối ưu toàn cục: vét cạn mọi cách phân hoạch (khả thi?)
  - dùng *heuristics*: k-Means, k-Medoids, Fuzzy C-Means, …



2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means** (MacQueen, 67; Lloyd, 57)**: c_i là trọng tâm của cluster C_i**



2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means** (MacQueen, 67; Lloyd, 57)**: c_i là trọng tâm của cluster C_i**

  Bước 1. Chọn ngẫu nhiên k trọng tâm $\{c_1, c_2, \ldots, c_k\}$.

  Bước 2. $\forall x \in D$, đưa x vào cluster có trọng tâm gần với x nhất:

$$C(x) = \arg\min_i d(x, c_i)$$

  Nếu phân hoạch mới không đổi so với lần trước: DỪNG.

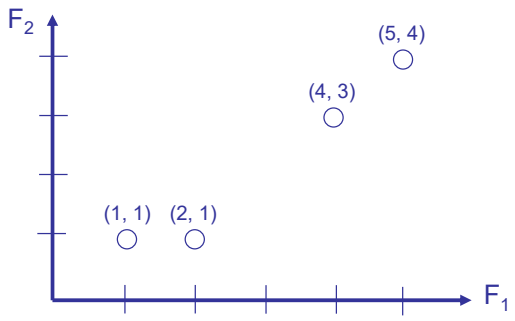
  Bước 3. Xác định lại k trọng tâm $\{c_1, c_2, \ldots, c_k\}$, quay lại Bước 2.

2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means**

  VD: m = 4, không gian 2 chiều $\{F_1, F_2\}$, k = 2 $\{C_1, C_2\}$

Dataset	Feature F1	Feature F2
x1	1	1
x2	2	1
x3	4	3
x4	5	4



2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means**

Vòng lặp t = 1:

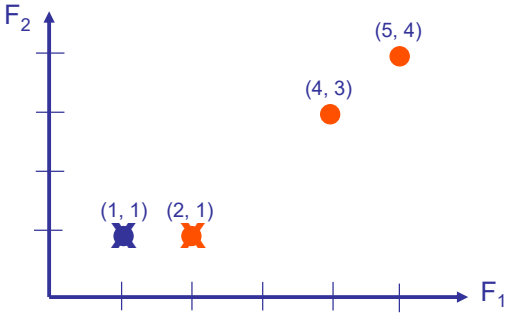
Chọn ngẫu nhiên 2 trọng tâm: $c_1(1, 1)$ và $c_2(2, 1)$

Tính khoảng cách từ các điểm đến từng trọng tâm:

Dataset	Centroid c1	Centroid c2
x1	0	1
x2	1	0
x3	3.606	2.828
x4	5	4.243

Gom các điểm vào cluster:

Dataset	Centroid c1	Centroid c2
x1	0	1
x2	1	0
x3	3.606	2.828
x4	5	4.243



2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means**

Vòng lặp t = 2:

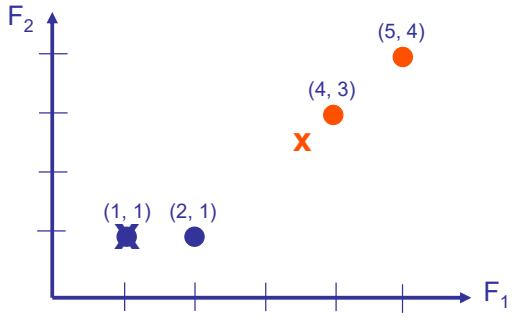
Cập nhật lại 2 trọng tâm: $c_1(1, 1)$ và $c_2(3.67, 2.67)$

Tính khoảng cách từ các điểm đến từng trọng tâm:

Dataset	Centroid c1	Centroid c2
x1	0	3.145
x2	1	2.357
x3	3.606	0.471
x4	5	1.886

Gom các điểm vào cluster:

Dataset	Centroid c1	Centroid c2
x1	0	3.145
x2	1	2.357
x3	3.606	0.471
x4	5	1.886



2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Means**

Vòng lặp t = 3:

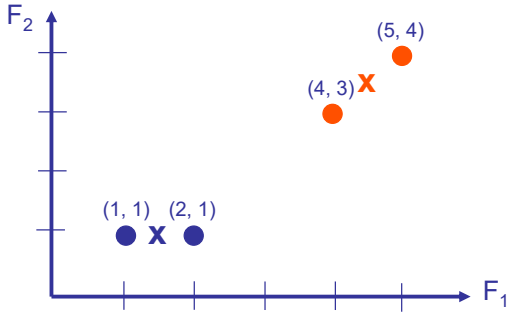
Cập nhật lại 2 trọng tâm: $c_1(1.5, 1)$ và $c_2(4.5, 3.5)$

Tính khoảng cách từ các điểm đến từng trọng tâm:

Dataset	Centroid c1	Centroid c2
x1	0.5	4.301
x2	0.5	3.536
x3	3.202	0.707
x4	4.61	0.707

Gom các điểm vào cluster:

Dataset	Centroid c1	Centroid c2
x1	0.5	4.301
x2	0.5	3.536
x3	3.202	0.707
x4	4.61	0.707



2.1 Cách tiếp cận dựa trên phân hoạch

- **Ưu điểm của k-Means**
  - độ phức tạp O(t.k.n), với n = |D| và t là số vòng lặp (k, t << n)
- **Khuyết điểm của k-Means**
  - xác định *siêu tham số* k → ELBOW, BIC, (Hastie et al., 2009)
  - chỉ áp dụng cho dữ liệu numerical → k-Modes, k-Prototypes
  - nhạy cảm với dữ liệu nhiễu → k-Medoids
  - kém hiệu quả với tập DL không lồi (*non-convex*) → DBSCAN
  - cực trị địa phương (phụ thuộc vào những điểm bắt đầu)

2.1 Cách tiếp cận dựa trên phân hoạch

❑ **Thuật toán k-Modes**

- áp dụng cho dữ liệu kiểu categorical (nominal, ordinal)
- tương tự k-Means nhưng khoảng cách giữa 2 phần tử được tính bằng tổng số giá trị features KHÁC NHAU của 2 phần tử (*Hamming distance*)

$$d(x,y) = \frac{\sum_{i=1}^{n} e(x_i, y_i)}{n} \qquad e(x_i, y_i) = \begin{cases} 0, & x_i = y_i \\ 1, & x_i \neq y_i \end{cases}$$

VD: d((Male, VN, Math), (Female, VN, CS)) = 2/3

2.1 Cách tiếp cận dựa trên phân hoạch

❑ **Thuật toán k-Prototypes: kết hợp giữa k-Means và k-Modes**

- áp dụng cho dữ liệu kiểu bất kỳ
- khoảng cách giữa 2 phần tử có thể được tính toán dựa trên các khoảng cách giữa các numerical và categorical features (có thể có trọng số)

2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán k-Medoids hay Partition Around Medoids (PAM):**
  - *medoid*: phần tử có tổng khoảng cách đến các phần tử khác trong cùng cụm là NHỎ NHẤT
  - tương tự k-Means nhưng chọn c_i là medoid của mỗi cluster C_i

  \+ giảm tác động của noise và outliers

  – độ phức tạp tính toán → CLARANS

2.1 Cách tiếp cận dựa trên phân hoạch

- **Thuật toán Fuzzy C-Means** (Dunn, 73; Bezdek, 81)
  - tập mờ (*fuzzy set*) với hàm thành viên (*membership function*) $F_D : D \rightarrow [0, 1]$ (Zadeh, 65)

    VD: Flip-Flop thuộc 2 nhóm sản phẩm { phần mềm, giải trí }

  - 1 đối tượng có thể thuộc nhiều hơn 1 cluster → *fuzzy clusters* $\{C_1, \ldots, C_k\}$, với mức độ thành viên (*degree of membership*)

2.1 Cách tiếp cận dựa trên phân hoạch

❑ Thuật toán Fuzzy C-Means

- ma trận phân hoạch (*partition matrix*): $W = [w_{ij}]_{n \times k}$, với:

  w_{ij} = membership(d_i, C_j) $\in [0, 1]$ ($\rightarrow$ xác suất)

$$\begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_k \end{array} \\ \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1k} \\ w_{21} & w_{22} & \ldots & w_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nk} \end{pmatrix} \end{array} \longrightarrow \text{tổng giá trị dòng } \boldsymbol{i}: \; ROW_i = \sum_{j=1}^{k} w_{ij} = 1$$

$$\downarrow$$

tổng giá trị cột **j**: $0 < COL_j = \sum_{i=1}^{n} w_{ij} < n$ (*không có cluster rỗng*)

- tối ưu hóa: $SSE(C) = \sum_{j=1}^{k} \sum_{x_i \in C_j} w_{ij}^p . d(d_i, c_j)^2$ với tham số p > 1

2.1 Cách tiếp cận dựa trên phân hoạch

❑ Thuật toán Fuzzy C-Means: cải biên từ k-Means

Bước 1. Khởi tạo (ngẫu nhiên) $W^{(0)}$ với $ROW_i = 1$, $\forall i=1..n$

Bước 2. Xác định k trọng tâm ở vòng lặp thứ t:

$$c_j = \frac{\sum_{i=1}^{n} w_{ij}^p . x_i}{\sum_{i=1}^{n} w_{ij}^p}$$

Bước 3. Cập nhật ma trận phân hoạch $W^{(t)}$ ở vòng lặp thứ t:

$$w_{ij} = \frac{1}{\sum_{s=1}^{k} \left(\frac{d(x_i, c_j)}{d(x_i, c_s)} \right)^{\frac{2}{p-1}}}$$

Bước 4. Nếu $\left\| W^{(t)} - W^{(t-1)} \right\| < \varepsilon$ thì DỪNG; Ngược lại, quay lại Bước 2.

2.1 Cách tiếp cận dựa trên phân hoạch

❑ Thuật toán Fuzzy C-Means

VD: m = 6, không gian 2 chiều $\{F_1, F_2\}$, k = 2 $\{C_1, C_2\}$, p = 2

Dataset	Feature F1	Feature F2
x1	1	6
x2	2	5
x3	3	8
x4	4	4
x5	5	7
x6	6	9

Khởi tạo $W^{(0)}$

Partition	Cluster C1	Cluster C2
x1	0.8	0.2
x2	0.9	0.1
x3	0.7	0.3
x4	0.3	0.7
x5	0.5	0.5
x6	0.2	0.8

2.1 Cách tiếp cận dựa trên phân hoạch

❑ Thuật toán Fuzzy C-Means

Vòng lặp t = 1:

Dataset	Feature F1	Feature F2	Cluster C1	Cluster C2
x1	1	6	0.8	0.2
x2	2	5	0.9	0.1
x3	3	8	0.7	0.3
x4	4	4	0.3	0.7
x5	5	7	0.5	0.5
x6	6	9	0.2	0.8

2 trọng tâm: $c_1 = \left(\frac{5.58}{2.32}, \frac{14.28}{2.32}\right) = (2.4, 6.1)$ $c_2 = \left(\frac{7.38}{2.32}, \frac{10.48}{2.32}\right) = (4.8, 6.8)$

Cập nhật $W^{(1)}$:

Dataset	Feature F1	Feature F2	Cluster C1	Cluster C2
x1	1	6	0.7	0.3
x2	2	5	0.6	0.4
x3	3	8	0.5	0.5
x4	4	4	0.5	0.5
x5	5	7	0.1	0.9
x6	6	9	0.3	0.7

Tiếp tục vòng lặp t … cho đến khi phân hoạch "ổn định".

2.1 Cách tiếp cận dựa trên phân hoạch

- **Ưu điểm của FCM**
  - gom cụm linh hoạt
  - hiệu quả với tập dữ liệu lớn, các cụm chồng lắp lên nhau
- **Khuyết điểm của FCM**
  - xác định giá trị của p
  - phụ thuộc vào bước khởi tạo → cực trị địa phương
  - nhạy cảm với nhiễu và giá trị bất thường

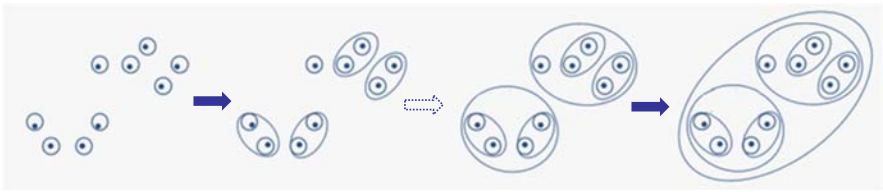
2.2 Cách tiếp cận dựa trên phân cấp

- **Cách tiếp cận dựa trên sự phân cấp** (*Hierarchical Approach*)
  - dựa trên sự cây phân rã các đối tượng theo các tiêu chí
  - clusters sau khi hình thành vẫn có thể phân tách, gộp lại
  - *Agnes*, *Diana*
  - *BIRCH*, *CURE*, *CHAMELEON*, … (đọc thêm)

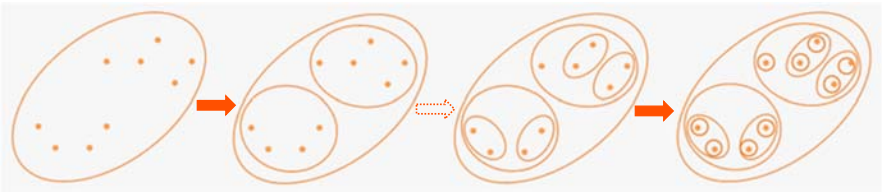
2.2 Cách tiếp cận dựa trên phân cấp

- **Hai phương pháp điển hình**

***Ag**glomerative **Nes**ting* (AGNES)



***Di**visive **Ana**lysis* (DIANA)



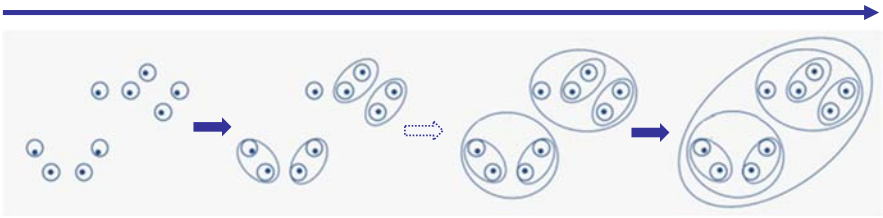
2.2 Cách tiếp cận dựa trên phân cấp

- **Phương pháp AGNES** (Kaufmann & Rousseeuw, 1990)

Bước 1. Khởi tạo m clusters, mỗi cluster chứa 1 đối tượng.

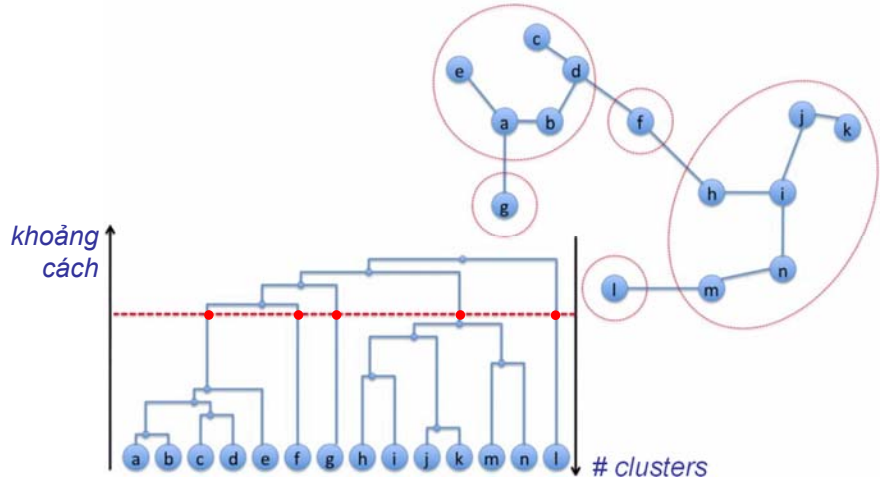
Bước 2. Dựa trên single-link, gộp chung 2 clusters gần nhau nhất thành 1 cluster.

Bước 3. Lặp lại Bước 2 cho đến khi gộp n đối tượng vào 1 cluster



2.2 Cách tiếp cận dựa trên phân cấp

- **Biểu đồ Dendrogram: cây phân cấp các clusters**
  - phân hoạch là một nhát cắt (ngang) ở 1 mức xác định
  - những nút được liên kết với nhau sẽ tạo thành 1 cluster



2.2 Cách tiếp cận dựa trên phân cấp

VD: Phương pháp AGNES (Kaufmann & Rousseeuw, 1990)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

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

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	3.20	2.50	2.24	0.00	1.00
E	4.24	3.54	1.41	1.00	0.00

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	4.95	0	2.24	1.41
(D, F)	2.50	2.24	0	1.00
E	3.54	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	2.50	1.41	0.00

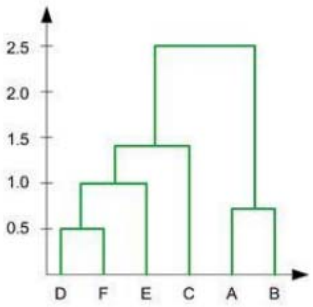
Min Distance (Single Linkage)

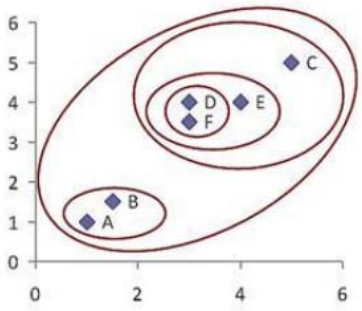
Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	2.50	0.00

2.2 Cách tiếp cận dựa trên phân cấp

VD: Phương pháp AGNES (Kaufmann & Rousseeuw, 1990)

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5





2.2 Cách tiếp cận dựa trên phân cấp

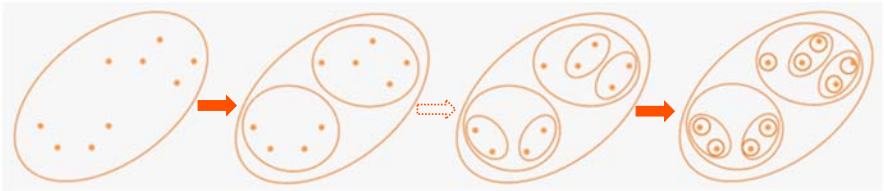
- **Phương pháp DIANA** (Kaufmann & Rousseeuw, 1990)

  Bước 1. Khởi tạo 1 cluster chứa tất cả m đối tượng.

  Bước 2. Mỗi cluster có hơn 1 đối tượng được tách thành 2 clusters (top-down, ngược với AGNES).

  Bước 3. Lặp lại Bước 2 cho đến khi có n clusters.

***Di**visive **Ana**lysis* (DIANA)



2.2 Cách tiếp cận dựa trên phân cấp

- **Ưu điểm của cách tiếp cận phân cấp**
  - giải thuật đơn giản
  - kết quả dễ hiểu
  - không cần tham số đầu vào (k)
- **Khuyết điểm cách tiếp cận phân cấp**
  - không thể quay lui
  - độ phức tạp $O(n^2)$, không thích hợp với tập dữ liệu lớn
  - nhạy cảm với nhiễu, dữ liệu bị thiếu
  - không hiệu quả với tập dữ liệu không lồi (*non-convex*)

2.3 Cách tiếp cận dựa trên mật độ

- **Cách tiếp cận dựa trên mật độ** (*Density-based Approach*)
  - dựa trên sự khác biệt về mật độ giữa vùng các đối tượng
  - *DBSCAN*
  - *Mean-Shift*, *OPTICS*, *DenClue*, … (đọc thêm)

2.3 Cách tiếp cận dựa trên mật độ

- **Nguyên tắc dựa trên mật độ**
  - *reachability*: khả năng tiếp cận
  - *connectivity*: sự kết nối
  - *density*: mật độ
  - nhận diện cluster: vùng không gian có mật độ dữ liệu cao và được ngăn cách với clusters gần đó bằng những vùng liền kề có mật độ dữ liệu thấp

2.3 Cách tiếp cận dựa trên mật độ

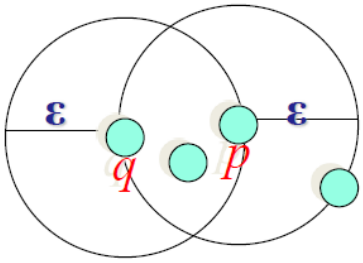
- **Thuật toán DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*)
  - siêu tham số ε: bán kính xác định *neighbors* của điểm p
  - siêu tham số *minPts*: ngưỡng số lượng tối thiểu các điểm của một vùng để được xem như là có mật độ cao
  - *core point*: điểm dữ liệu lõi của một vùng có mật độ cao
  - mở rộng kết nối từ core points, liên kết thêm các neighbors để hình thành các clusters

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

- vùng lân cận (*eps_neighborhood*) của điểm dữ liệu p:

$$N_\varepsilon(p) = \{q \in D \mid d(p,q) < \varepsilon\}$$



[Jing Gao]

Nếu MinPts = 4 thì $N_\varepsilon(p)$ có mật độ cao, còn $N_\varepsilon(q)$ thì không

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

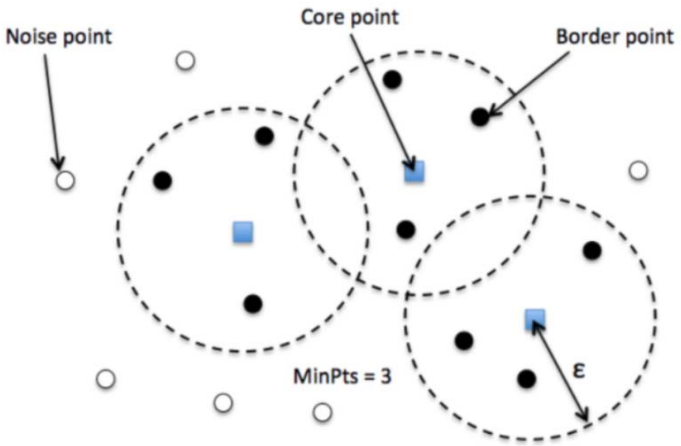
- *điểm lõi* (*core point*) c: $|N_\varepsilon(c)| \geq \text{minPts}$
- *điểm biên* (*border point*) b: không phải core point, nhưng vùng lân cận của b có chứa core point

$$(|N_\varepsilon(b)| < \text{minPts}) \wedge (\exists \text{core\_point c} \in N_\varepsilon(b))$$

- *điểm nhiễu* (*noise point*): không thuộc 2 loại trên (không thuộc cluster nào)

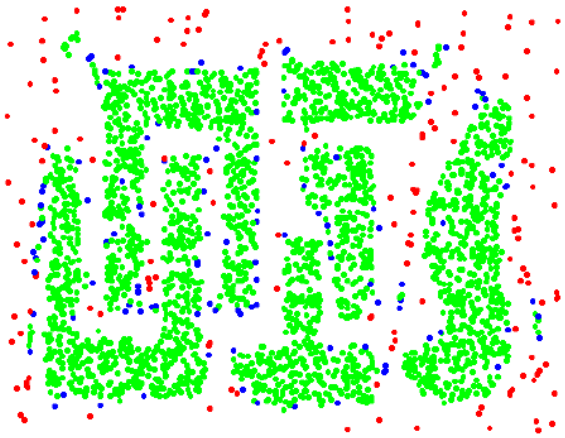
2.3 Cách tiếp cận dựa trên mật độ

- Thuật toán DBSCAN



2.3 Cách tiếp cận dựa trên mật độ

- Thuật toán DBSCAN



core points

border points

noise points

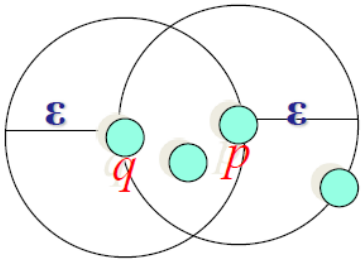
[Jing Gao]

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

- *Direct Density Reachability* (DDR):

q DDR đối với core point p: $(q \rightarrow p) \Leftrightarrow (q \in N_{\varepsilon}(p))$



[Jing Gao]

Với MinPts = 4: q → p

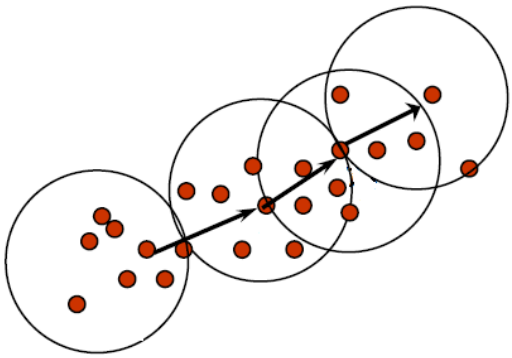
Điều ngược lại không đúng (→ có tính chất *asymmetric*)

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

- *Density Reachability* (DR): chuỗi có hướng các DDR

q DR đối với core point p: $(q \Rightarrow p) \Leftrightarrow (q \rightarrow p_n \rightarrow \ldots \rightarrow p_1 \rightarrow p)$



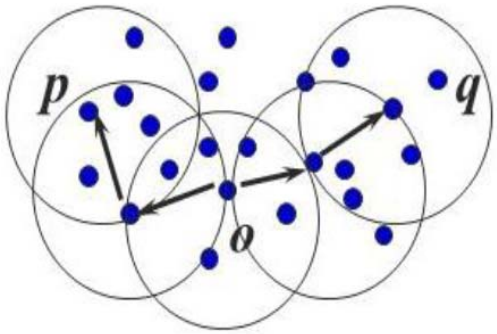
[Jing Gao]

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

- *Density Connectivity* (DC): giữa 2 core points

q DC đối với p: $(q \leftrightarrow p) \Leftrightarrow (\exists o : (o \Rightarrow q) \wedge (o \Rightarrow p))$



DC có tính chất *symetric*

2.3 Cách tiếp cận dựa trên mật độ

❑ Thuật toán DBSCAN

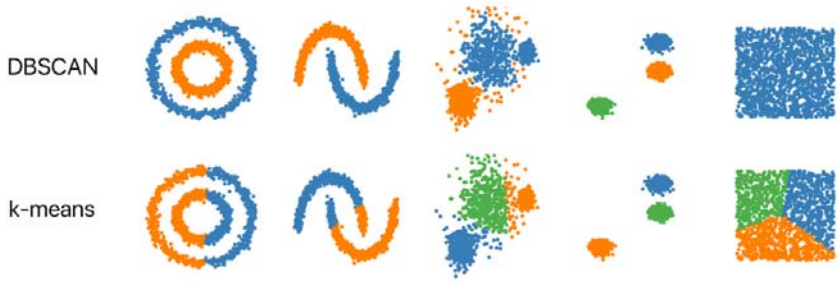
```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
            ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
```

```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
              MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE   // all points in seeds are density-
         // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                        Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN (UNCLASSIFIED, NOISE) THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

2.3 Cách tiếp cận dựa trên mật độ

❑ **Thuật toán DBSCAN**



2.4 Cách tiếp cận dựa trên lưới

❑ **Cách tiếp cận dựa trên lưới** (*Grid-based Approach*)

- dựa trên của lưới
- *CLIQUE*
- *BANG*, *STING*, *WaveCluster*, … (đọc thêm)

2.4 Cách tiếp cận dựa trên lưới

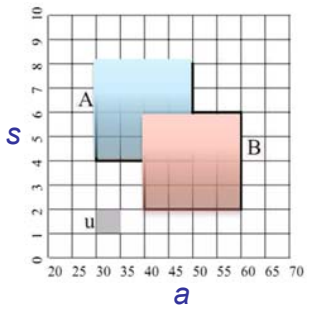
- **Thuật toán CLIQUE** (*CLustering In QUEst*)
  - tìm clusters trên các không gian con (*subspace*)
  - tạo phân hoạch trên từng dimension → hình thành intervals
  - kết hợp các phân hoạch trên các dimensions thành *cells*
  - áp dụng tính chất *Apriori*

2.4 Cách tiếp cận dựa trên lưới

- **Thuật toán CLIQUE** (*CLustering In QUEst*)
  - *unit*: Conjunctive Normal Form trên các dimensions

    VD: $u = (30 \le a < 35) \wedge (1 \le s < 2)$
  - *region*: mở rộng từ các units

    VD: $A = (30 \le a < 50) \wedge (4 \le s < 8)$
  - *selectivity*: mật độ của unit/region, bằng tỷ lệ % số phần tử so với toàn bộ tập dữ liệu
  - *dense*: mật độ vượt qua ngưỡng



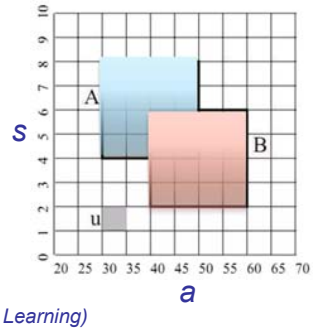
2.4 Cách tiếp cận dựa trên lưới

- **Thuật toán CLIQUE** (*CLustering In QUEst*)
  - *cluster*: phần hội của các dense units → mở rộng tối đa

    VD: dense units A và B, ta có cluster $C = A \cup B$
  - *minimal description*: Disjunctive Normal Form của 1 cluster

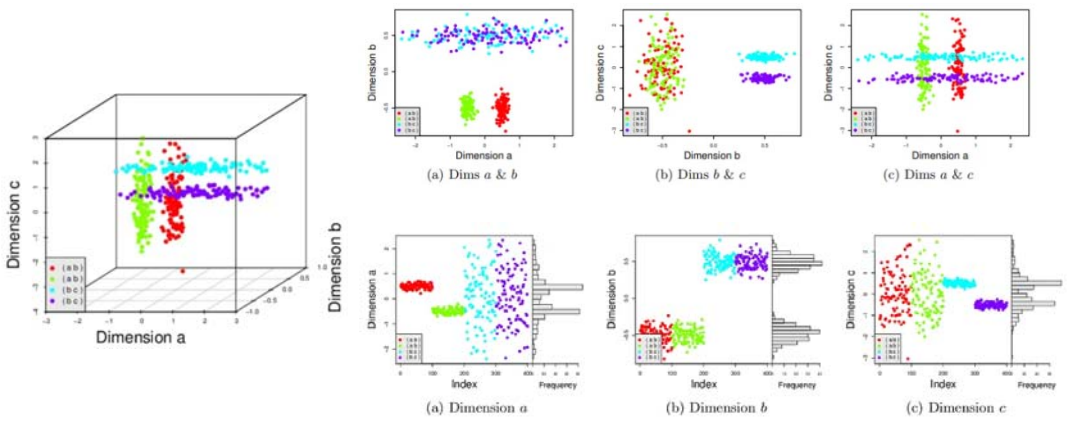
    VD: minimal description của cluster C (*Boolean Algebra*)

    $((30 \le a < 50) \wedge (4 \le s < 8)) \vee ((40 \le a < 60) \wedge (2 \le s < 6))$



2.4 Cách tiếp cận dựa trên lưới

- **Thuật toán CLIQUE** (*CLustering In QUEst*)
  - *subspace clustering*:



2.4 Cách tiếp cận dựa trên lưới

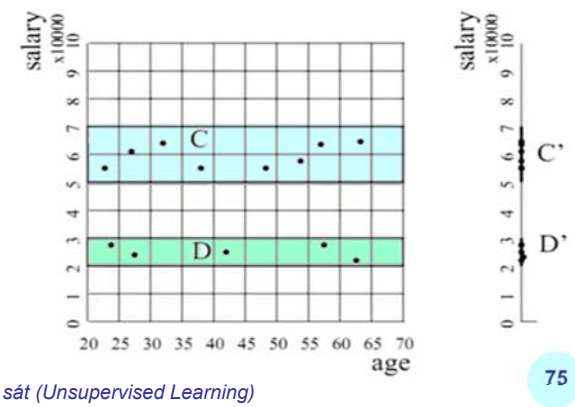
- **Thuật toán CLIQUE** (*CLustering In QUEst*)
  - *subspace clustering*: áp dụng tính chất Apriori

VD: Giả sử ngưỡng mật độ là 20%, $C = (5 \le s < 7)$ $D = (2 \le s < 3)$

C, D: dense units trên (age, salary)

C', D': dense units trên (salary)

Nhưng không có dense unit trên (age)



2.4 Cách tiếp cận dựa trên lưới

- **Thuật toán CLIQUE** (*CLustering In QUEst*)

Bước 1: Nhận diện các subspaces chứa clusters

Bước 2: Nhận diện các clusters trong subspace

Bước 3: Tạo minimal description cho các clusters

2.4 Cách tiếp cận dựa trên mô hình

- **Cách tiếp cận dựa trên mô hình** (*Model-based Approach*)
  - dựa trên mô hình (xác suất) của các clusters
  - *GMM*, *SOM*, … (đọc thêm)

Thảo luận