

**COLE.VN**  
connecting knowledge

**Chủ đề:**

**Học không giám sát (*Unsupervised Learning*)**

# Mục đích buổi học

- Học viên được tiếp cận các khái niệm, kỹ thuật và mô hình học máy cơ bản và phổ dụng với bài toán học không giám sát (Unsupervised Learning)
  - Các hướng tiếp cận
  - Các kỹ thuật học không giám sát

# Nội dung chính (ở đây ghi mục lục)

**Học không giám sát**

Đề bài

Giải bài

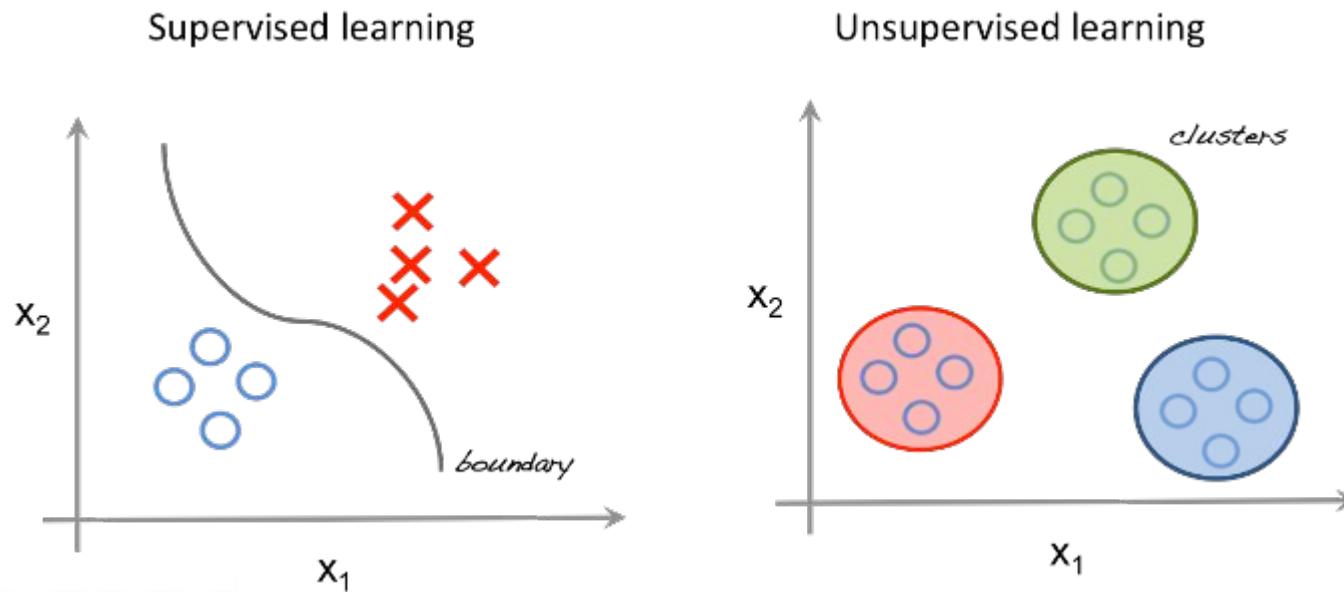
Đáp án

# Học không giám sát

- **Học có giám sát (Supervised learning)** - nhắc lại
  - Tập dữ liệu học (training data) bao gồm các quan sát (examples, observations), mà mỗi quan sát được gắn kèm với một giá trị đầu ra mong muốn.
  - Ta cần học một hàm (vd: một phân lớp, một hàm hồi quy,...) phù hợp với tập dữ liệu hiện có.
  - Hàm học được sau đó sẽ được dùng để dự đoán cho các quan sát mới.

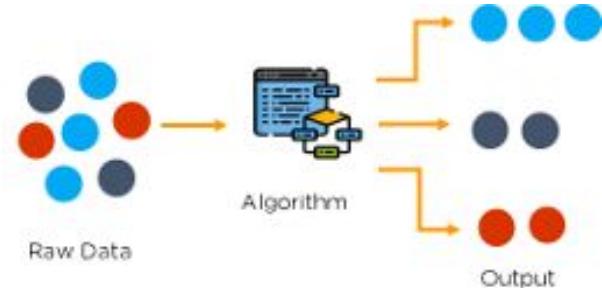
# Học không giám sát

- **Học không giám sát (Unsupervised learning)**
  - Tập học (training data) bao gồm các quan sát, mà mỗi quan sát không có thông tin về nhãn lớp hoặc giá trị đầu ra mong muốn.
  - Mục đích là tìm ra (học) các cụm, các cấu trúc, các quan hệ tồn tại ẩn trong tập dữ liệu hiện có.



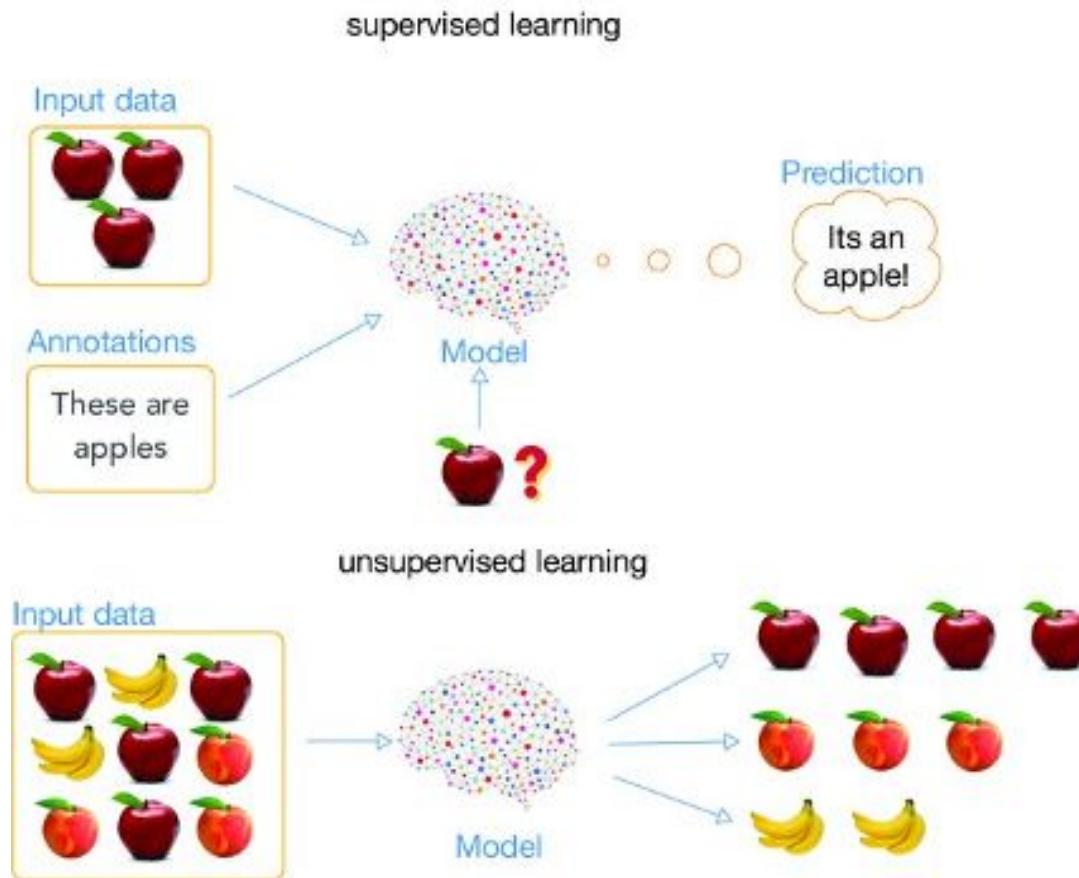
# Học không giám sát

- **Tư tưởng cơ bản của học không giám sát**
  - Nhóm các dữ liệu vào các lớp sao cho:
    - Có sự giống nhau lớn giữa các dữ liệu thuộc cùng lớp
    - Có sự khác nhau cao giữa các dữ liệu của các lớp khác nhau
  - Tìm nhãn (tên của nhóm, định danh của nhóm, ...) và số lượng các nhóm trực tiếp dữ trên dữ liệu (khác với học có giám sát)
- Các thuật toán học không giám sát:
  - Cố gắng khai phá (mining) cấu trúc của dữ liệu (data) data mining
  - Dữ liệu chưa được gán nhãn



# Học không giám sát

- **Supervised learning v.s Unsupervised learning**



# Học không giám sát

## Các lợi ích (điểm mạnh của học không giám sát):

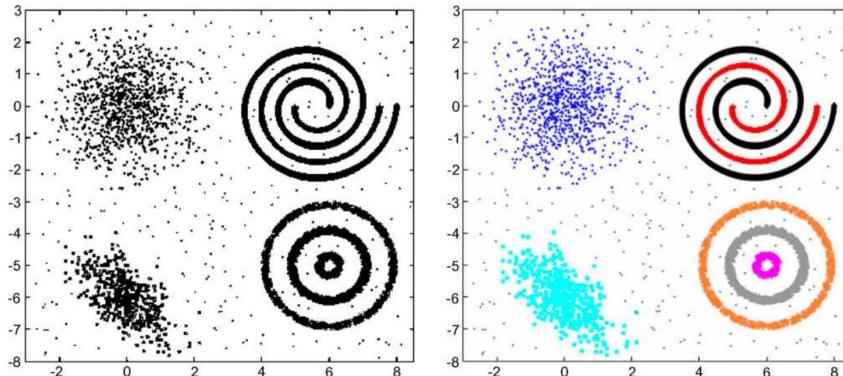
- Việc thu thập và gán nhãn cho một số lượng lớn dữ liệu là một công việc đòi hỏi thời gian, công thức, đôi khi nhảm chán và thiếu chính xác.
  - Ví dụ 1: Thu thập và gán nhãn dữ liệu tiếng nói: xác định đoạn tín hiệu nào tương ứng với âm nào, từ nào.
  - Ví dụ 2: Thu thập và gán nhãn dữ liệu hình ảnh
- Cho phép huấn luyện một tập lớn dữ liệu không gán nhãn, sau đó sẽ thực hiện gán nhãn trên các lớp tìm thấy tiết kiệm thời gian và công sức (hướng tiếp cận data mining)
- Phù hợp với các ứng dụng mà dữ liệu thay đổi theo thời gian theo vết sự thay đổi
- Có thể sử dụng như bộ lựa chọn đặc trưng: do có khả năng đưa ra đặc trưng phù hợp nhất
- Có thể sử dụng trong các bước tìm hiểu và thiết kế hệ thống

# Học không giám sát - Ví dụ

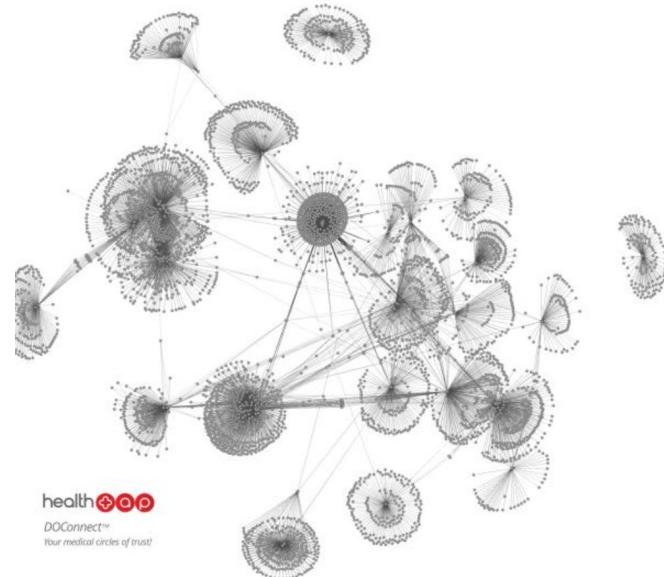
- Ứng dụng:
  - Dimensionality reduction (giảm chiều dữ liệu)
  - Clustering (phân cụm)
  - Anomaly detection (tìm điểm bất thường)

# Học không giám sát - Ví dụ

- Phân cụm (clustering)
  - Phát hiện các cụm dữ liệu, cụm tính chất,...

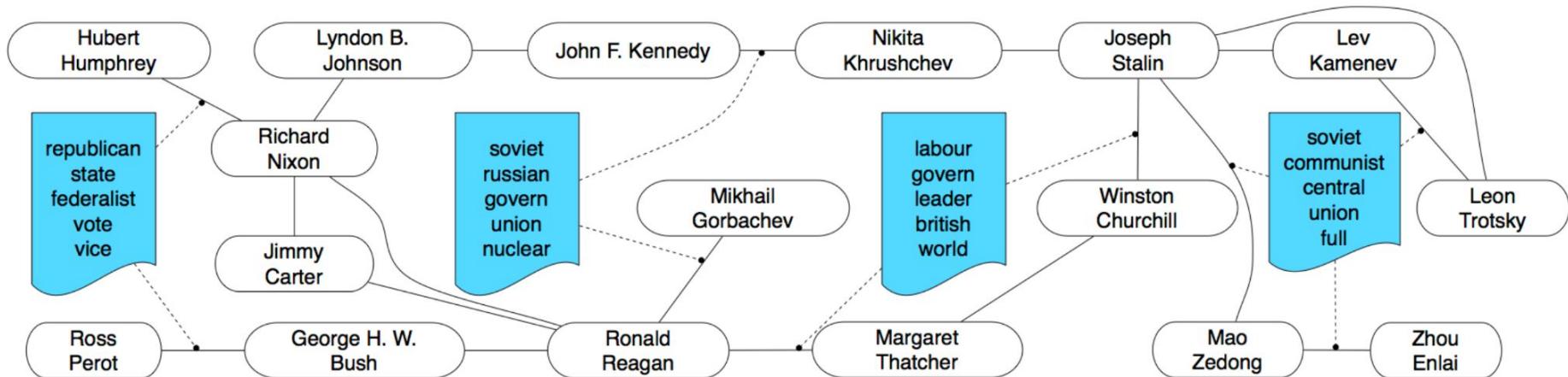
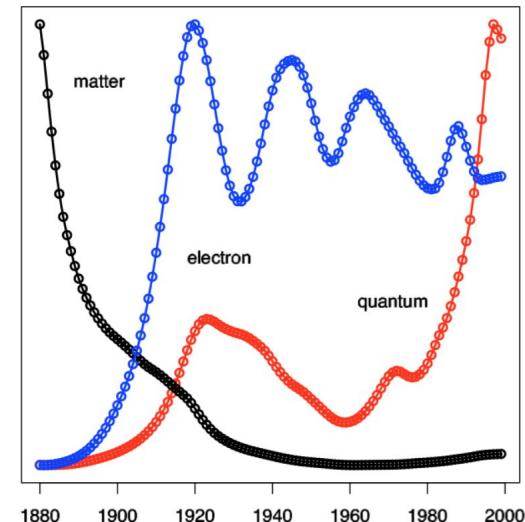


- Community detection
  - Phát hiện các cộng đồng trong mạng xã hội



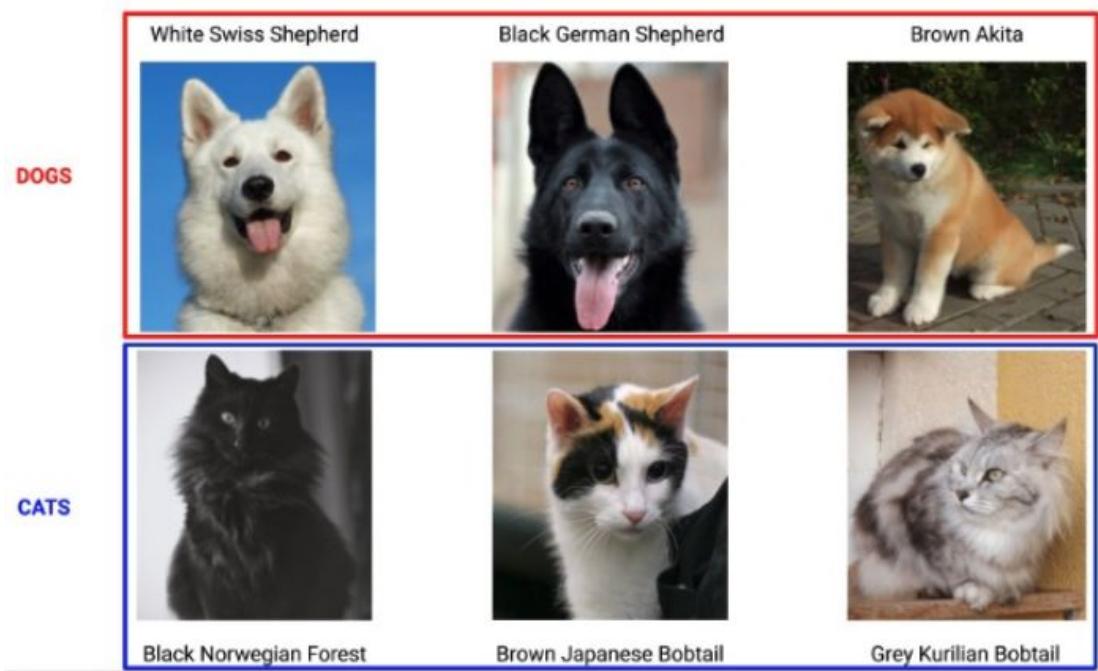
# Học không giám sát - Ví dụ

- Trends detection
  - Phát hiện xu hướng, thị yếu,...
- Entity-interaction analysis



# Clustering - Phân cụm dữ liệu

- Là việc chia dữ liệu thành các nhóm khác nhau nhằm tìm ra cấu trúc, đặc trưng hay các nhóm ẩn trong một tập hợp dữ liệu
- Các quan sát trong mỗi nhóm có nhiều sự tương đồng với các quan sát cùng nhóm hơn là với các nhóm khác.
- Có thể áp dụng học máy cho bài toán này.

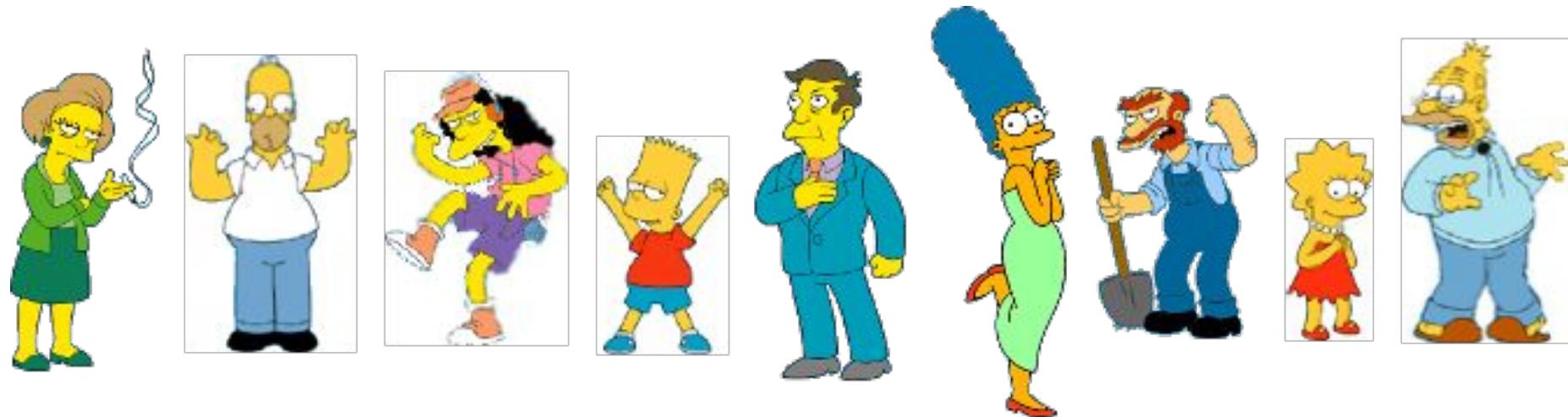


# Clustering - Phân cụm dữ liệu

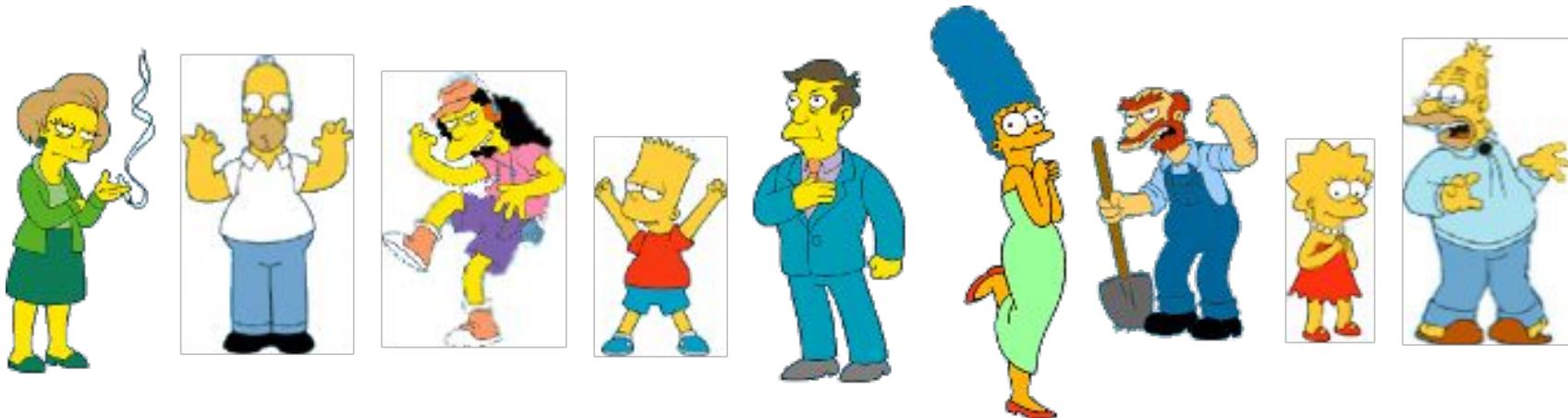
- Input: một tập dữ liệu  $\{x_1, \dots, x_M\}$  không có nhãn (hoặc giá trị đầu ra mong muốn)
- Output: các cụm (nhóm) của các quan sát
- Một cụm (cluster) là một tập các quan sát
  - Tương tự với nhau (theo một ý nghĩa, đánh giá nào đó)
  - Khác biệt với các quan sát thuộc các cụm khác

# Phân cụm bằng học máy (machine learning)

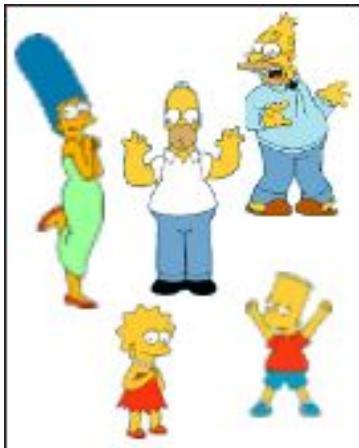
Yêu cầu: Nhóm các đối tượng tương tự nhau?



# Phân cụm bằng học máy (machine learning)



Việc nhóm các đối tượng là **chủ quan**



Thành viên gia đình

Người làm việc



Nữ



Nam



# Phân cụm bằng học máy (machine learning)

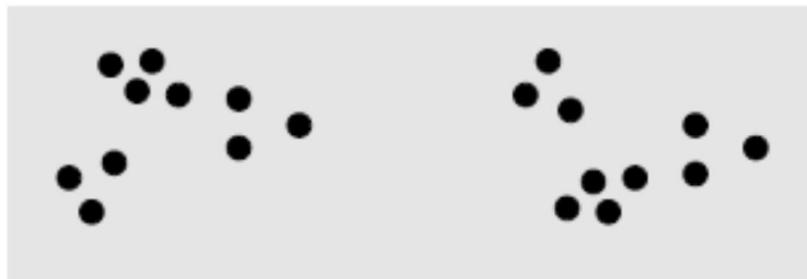
Sự tương tự giữa các đối tượng

- Nhìn thấy, nghe thấy cảm nhận thấy mô tả tường minh cho máy tính
- Liên quan đến các lĩnh vực tâm lý



# Phân cụm bằng học máy (machine learning)

- Mỗi cụm/nhóm nên có bao nhiêu phần tử?
- Các phân tử nên được phân vào bao nhiêu cụm/nhóm?
- Bao nhiêu cụm/nhóm nên được tạo ra?



Bao nhiêu cụm?



6 cụm?



2 cụm?



4 cụm?

# Phân cụm bằng học máy (machine learning)

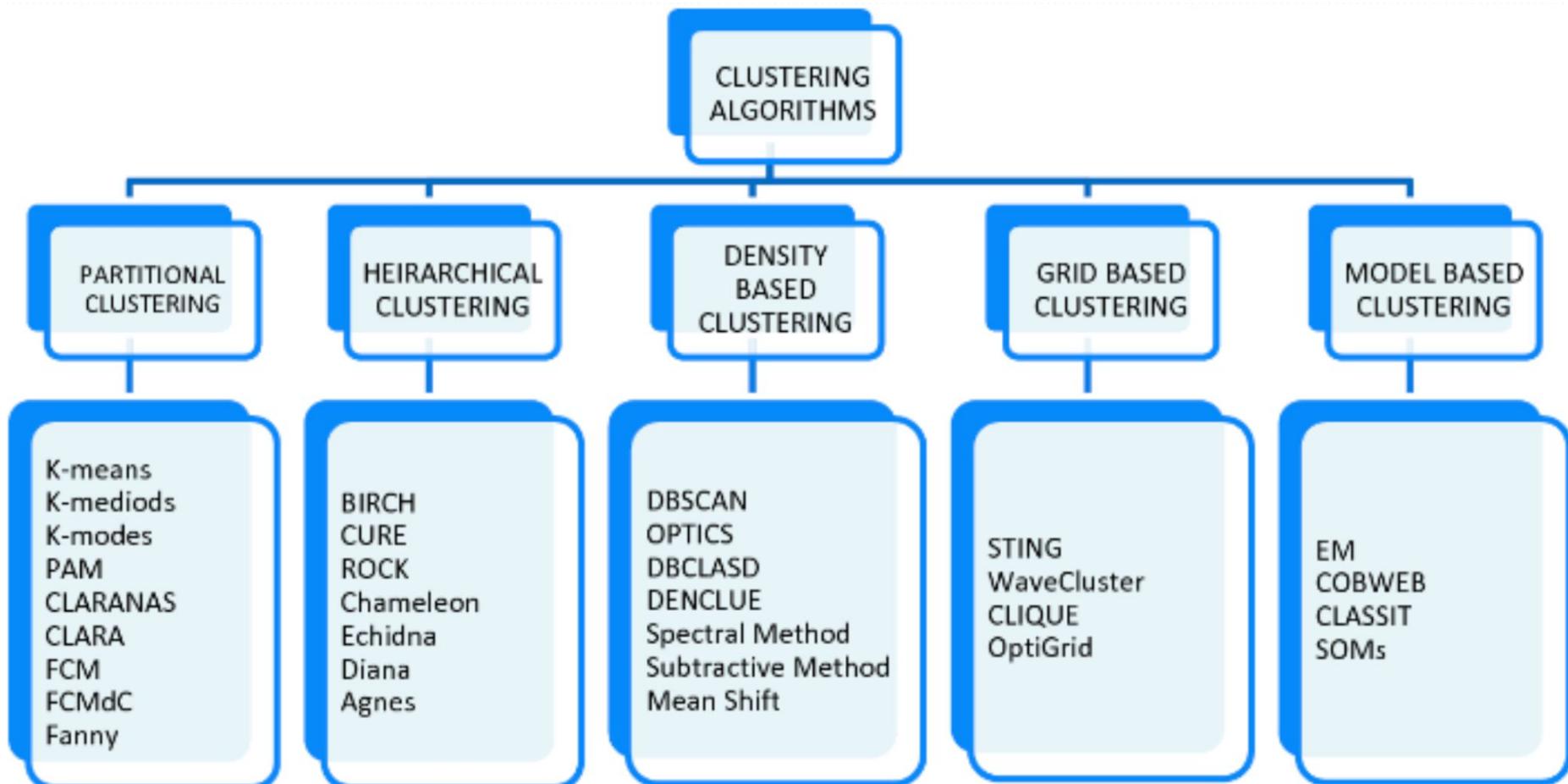
- Các yêu cầu khi thiết kế thuật toán phân cụm dữ liệu:
  - Có thể tương thích, hiệu quả với dữ liệu lớn, số chiều lớn
  - Có khả năng xử lý các dữ liệu khác nhau
  - Có khả năng khám phá các cụm với các dạng bất kỳ
  - Khả năng thích nghi với dữ liệu nhiễu
  - Ít nhạy cảm với thứ tự của các dữ liệu vào
  - Phân cụm ràng buộc
  - Dễ hiểu và dễ sử dụng

# Phân cụm bằng học máy (machine learning)

## Phân loại các phương pháp clustering

- Phân hoạch (partitioning): phân hoạch tập dữ liệu  $n$  phần tử thành  $k$  cụm
  - *Kmeans, Fuzzy C-mean, ...*
- Phân cấp (hierarchical): xây dựng phân cấp các cụm trên cơ sở các đối tượng dữ liệu đang xem xét
  - *AGNES (Agglomerative NESting), DIANA (Divisive ANAlysis) ,...*
- Dựa trên mật độ (density-based): dựa trên hàm mật độ, số đối tượng lân cận của đối tượng dữ liệu.
  - *DBSCAN, OPTICS, MeanShift ,...*
- Dựa trên mô hình (model-based): một mô hình giả thuyết được đưa ra cho mỗi cụm; sau đó hiệu chỉnh các thông số để mô hình phù hợp với cụm dữ liệu/đối tượng nhất.
  - *EM, SOMs ,...*
- Spectral clustering: *phân cụm dựa trên đồ thị*
- ...

# Phân cụm bằng học máy (machine learning)

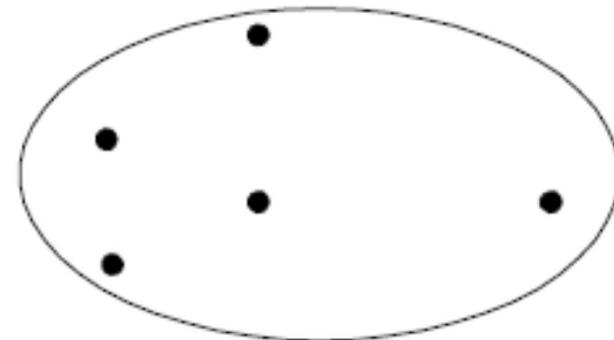
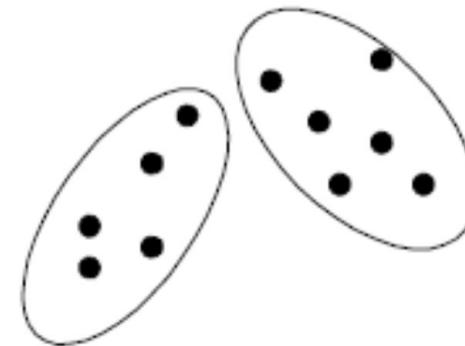


# Phân cụm bằng học máy (machine learning)

Ví dụ: Phân hoạch (partitioning)



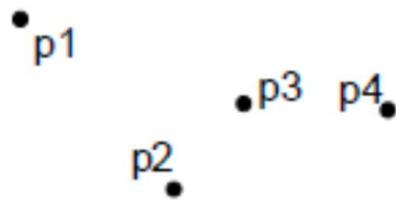
Original Points



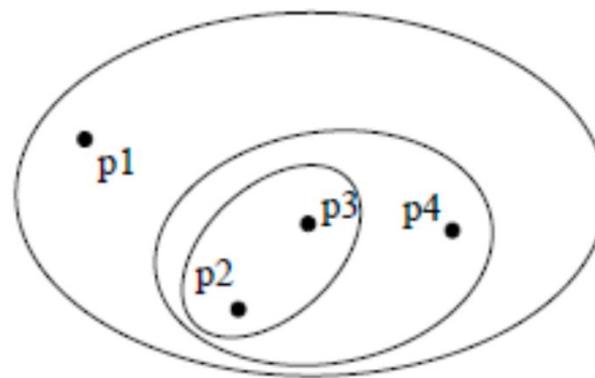
Partitioning

# Phân cụm bằng học máy (machine learning)

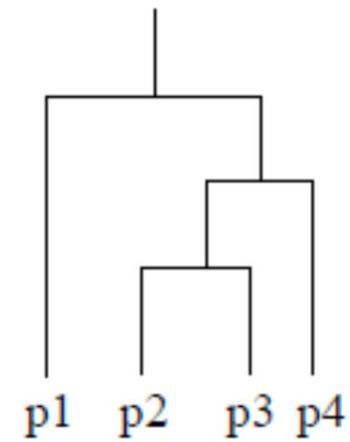
Ví dụ: Phân cấp (hierarchical)



Original Points



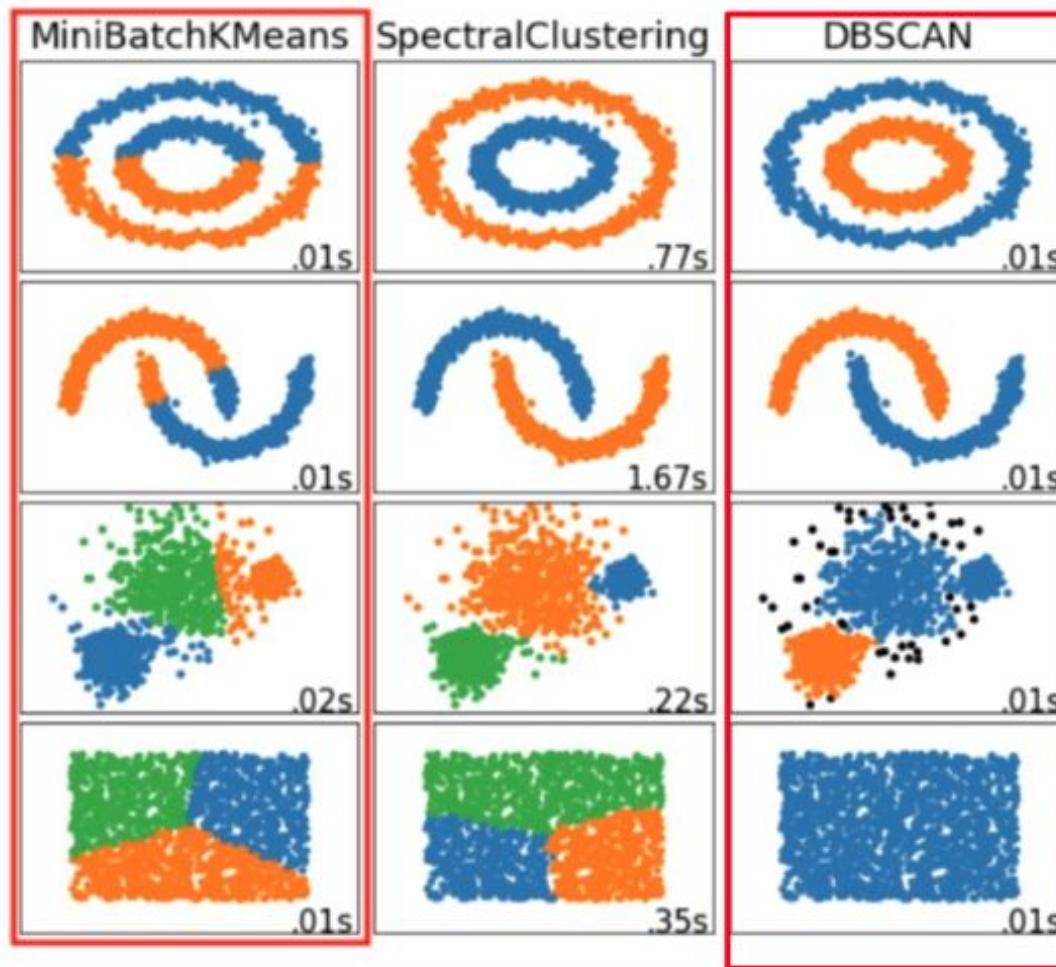
Hierarchical



# Phân cụm bằng học máy (machine learning)

- Đánh giá chất lượng phân cụm (Clustering quality)
  - Khoảng cách/sự khác biệt *giữa các cụm* → *Cần được cực đại hóa*
  - Khoảng cách/sự khác biệt *bên trong một cụm* → *Cần được cực tiểu hóa*

# Phân cụm bằng học máy (machine learning)



# KMeans - Phân cụm bằng học máy

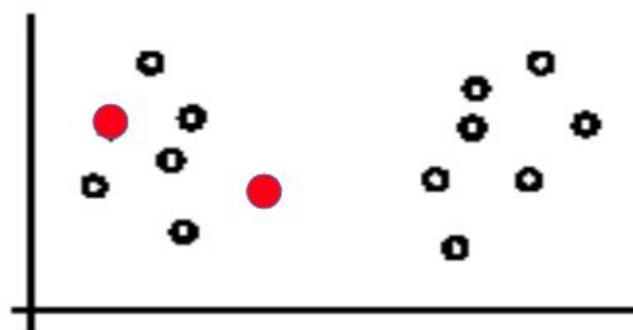
- K-means được giới thiệu đầu tiên bởi Lloyd năm 1957.
- Là phương pháp phân cụm phổ biến nhất trong các phương pháp dựa trên phân hoạch (partition-based clustering)
- Biểu diễn dữ liệu:  $D = \{x_1, x_2, \dots, x_i\}$ 
  - $x_i$  là một quan sát (một vectơ trong một không gian  $n$  chiều)
- Giải thuật K-means phân chia tập dữ liệu thành  $k$  cụm
  - Mỗi cụm (cluster) có một điểm trung tâm, được gọi là centroid
  - $k$  (tổng số các cụm thu được) là một giá trị được cho trước (vd: được chỉ định bởi người thiết kế hệ thống phân cụm)
  - Một đối tượng được phân vào một cụm nếu khoảng cách từ đối tượng đó đến trọng tâm của cụm đang xét là nhỏ nhất
  - Quá trình lặp đi lặp lại cho đến hàm mục tiêu bé hơn một ngưỡng cho phép hoặc các trọng tâm không đổi

# KMeans - Phân cụm bằng học máy

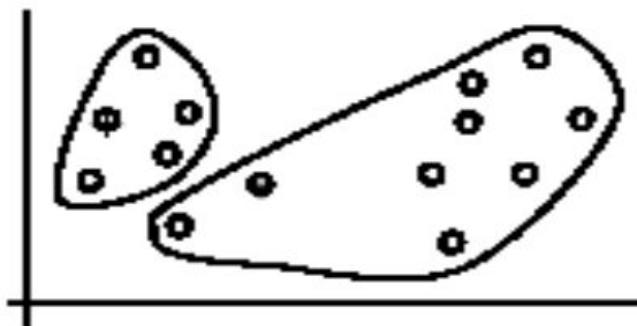
## Thuật toán

- Input:
  - Tập học  $D=\{x_1, x_2, \dots, x_r\}$  ( $x_i$  là một quan sát - một vector trong một không gian  $n$  chiều)
  - Số lượng cụm  $k$
  - Khoảng cách  $d(x, y)$
- Step 1. Chọn ngẫu nhiên  $k$  quan sát **để sử dụng làm các điểm trung tâm ban đầu (initial centroids) của  $k$  cụm.**
- Step 2. Lặp liên tục hai bước sau cho đến khi gặp điều kiện hội tụ (convergence criterion):
  - 2.1. Đối với mỗi quan sát, gán nó vào cụm (trong số  $k$  cụm) mà có tâm (centroid) gần nó nhất.
  - 2.2. Đối với mỗi cụm, *tính toán lại điểm trung tâm của nó dựa trên tất cả các quan sát thuộc vào cụm đó.*

# KMeans - Phân cụm bằng học máy



(A). Random selection of  $k$  centers

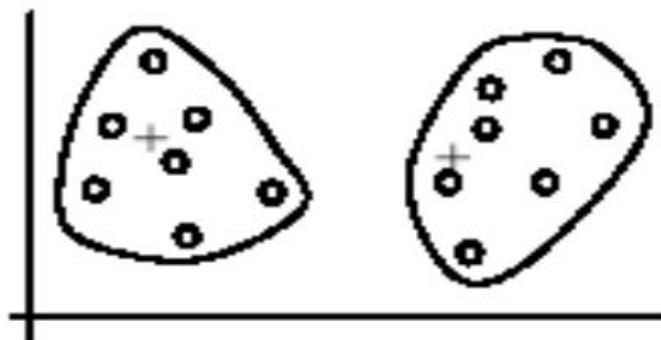


Iteration 1: (B). Cluster assignment

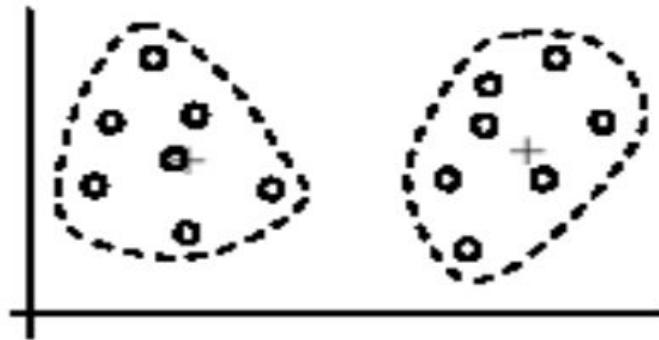


(C). Re-compute centroids

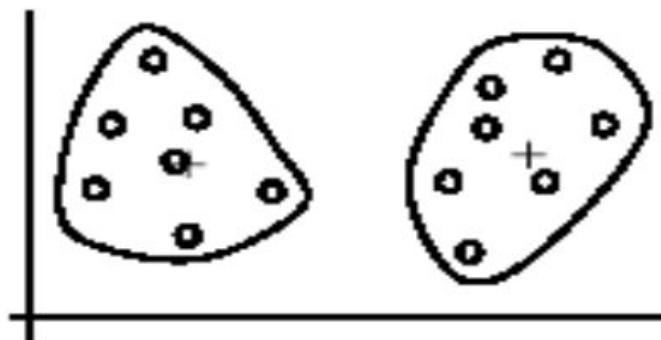
# KMeans - Phân cụm bằng học máy



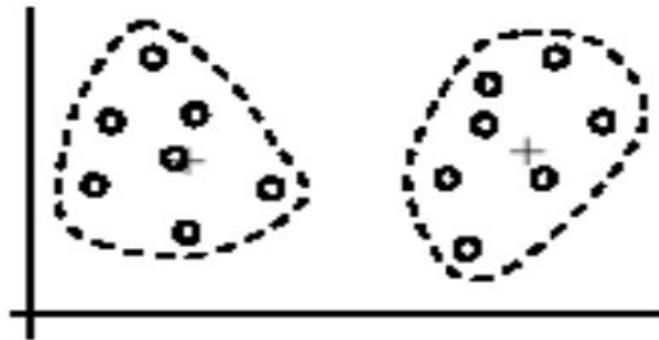
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

# KMeans - Phân cụm bằng học máy

## Điều kiện hội tụ

- Quá trình phân cụm kết thúc, nếu:
  - Không có (hoặc có không đáng kể) việc gán lại các quan sát vào các cụm khác, hoặc
  - Không có (hoặc có không đáng kể) thay đổi về các điểm trung tâm (*centroids*) của các cụm, hoặc
  - Giảm không đáng kể về tổng lỗi phân cụm:

$$Error = \sum_{i=1}^k \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{m}_i)^2$$

$C_i$ : Cụm thứ i

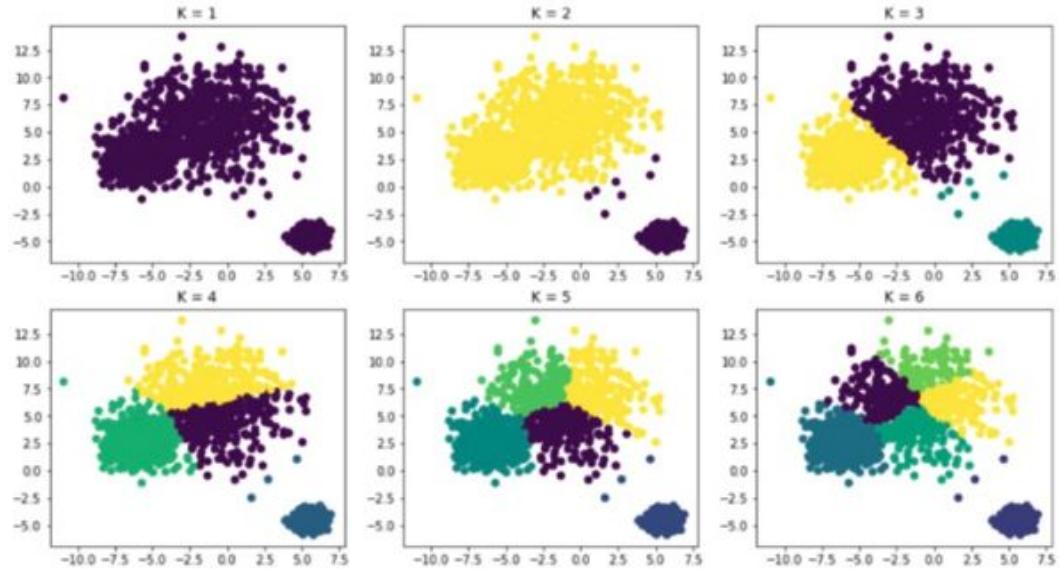
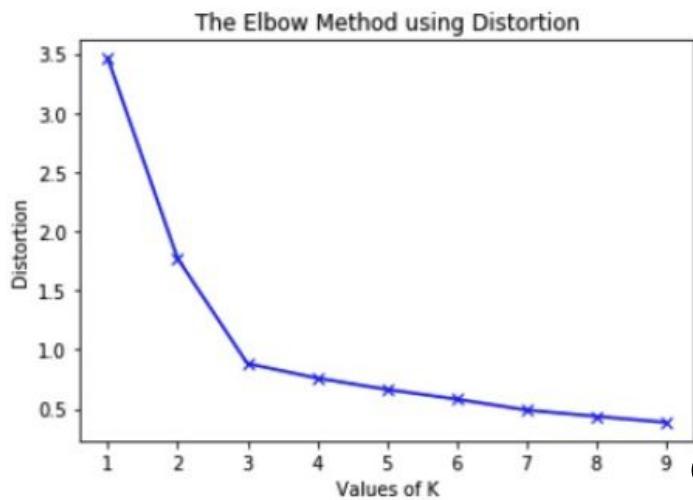
$\mathbf{m}_i$ : Điểm trung tâm (*centroid*) của cụm  $C_i$

$d(\mathbf{x}, \mathbf{m}_i)$ : Khoảng cách (khác biệt) giữa quan sát  $\mathbf{x}$  và điểm trung tâm  $\mathbf{m}_i$

# KMeans - Phân cụm bằng học máy

- Vấn đề: Bao nhiêu cụm là đủ?

→ phương pháp khuỷu tay  
(elbow method)

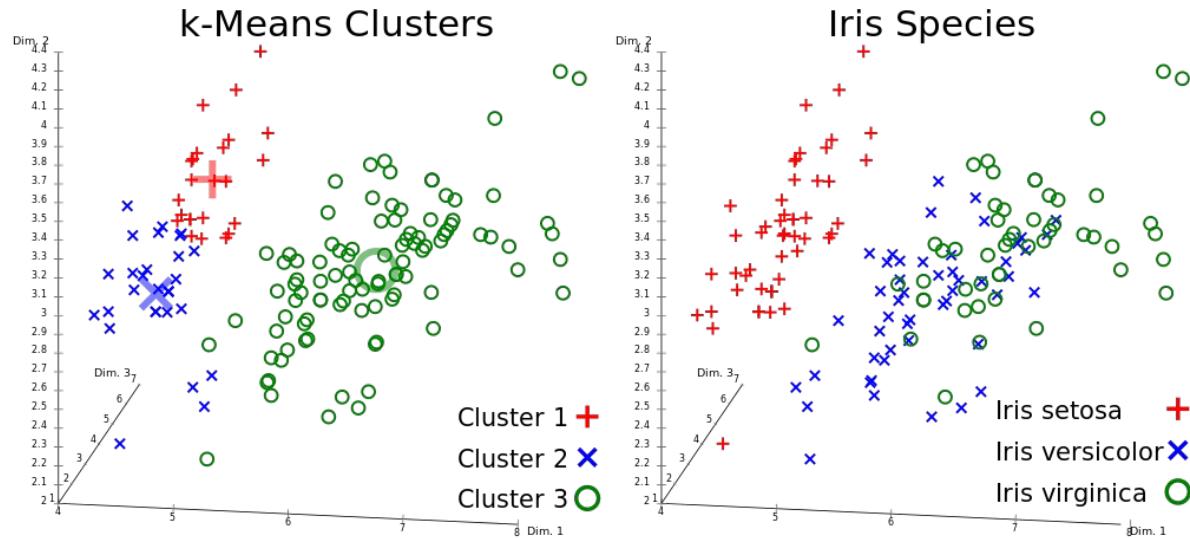


chọn giá trị tại khuỷu tay tức là khi độ biến  
dạng bắt đầu giảm theo tuyến tính

# KMeans - Phân cụm bằng học máy

## ● Minh họa

Kết quả của thuật toán KMeans Clustering trong việc phân loại loài hoa Iris



Ví dụ sử dụng KMeans Clustering



Ảnh gốc



2 clusters



3 clusters

# KMeans - Phân cụm bằng học máy

## Điểm trung tâm và hàm khoảng cách

- Xác định điểm trung tâm: Điểm trung bình (*Mean centroid*)

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

- (vectơ)  $\mathbf{m}_i$  là điểm trung tâm (*centroid*) của cụm  $C_i$
- $|C_i|$  kích thước của cụm  $C_i$  (tổng số quan sát trong  $C_i$ )
- Hàm khoảng cách: Euclidean distance

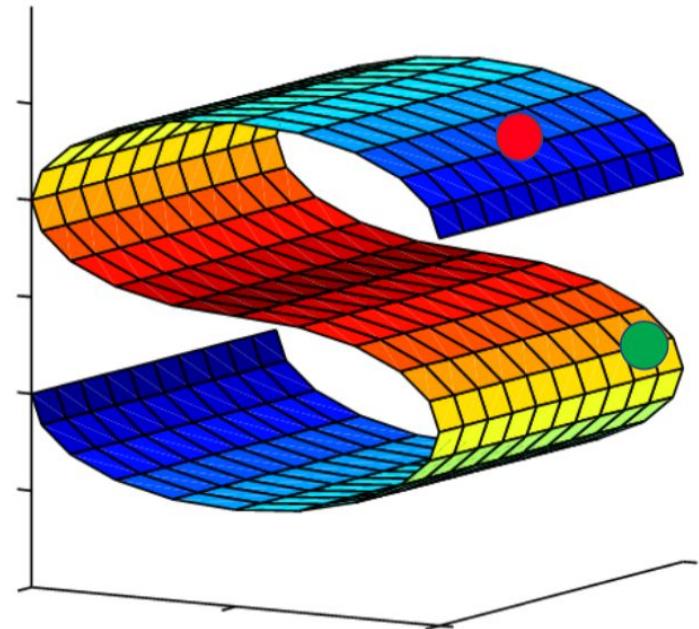
$$d(\mathbf{x}, \mathbf{m}_i) = \|\mathbf{x} - \mathbf{m}_i\| = \sqrt{(x_1 - m_{i1})^2 + (x_2 - m_{i2})^2 + \dots + (x_n - m_{in})^2}$$

- (vectơ)  $m_i$  là điểm trung tâm (*centroid*) của cụm  $C_i$
- $d(x, m_i)$  là khoảng cách giữa  $x$  và điểm trung tâm  $m_i$

# KMeans - Phân cụm bằng học máy

## Hàm khoảng cách

- Xác định điểm trung tâm: Điểm trung bình (*Mean centroid*)
  - Mỗi hàm sẽ tương ứng với một cách nhìn về dữ liệu.
  - Vô hạn hàm!!!
  - Chọn hàm nào?
- Có thể thay bằng độ đo tương đồng (similarity measure)



# KMeans - Phân cụm bằng học máy

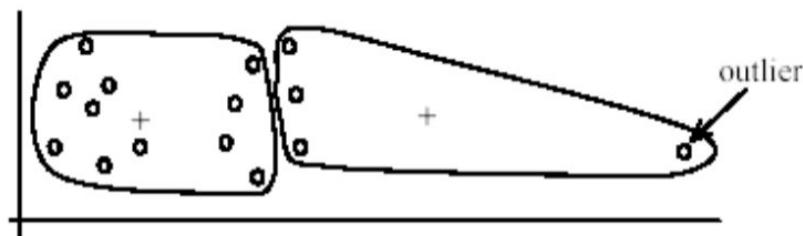
## Các ưu điểm

- Đơn giản, dễ cài đặt, dễ hiểu
- Rất linh động: cho phép dùng nhiều độ đo khoảng cách khác nhau  
→ phù hợp với các loại dữ liệu khác nhau
- Hiệu quả (khi dùng độ đo Euclid)
  - Độ phức tạp tại mỗi bước  $\sim O(r \cdot k)$ 
    - $r$ : Tổng số các quan sát (kích thước của tập dữ liệu)
    - $k$ : Tổng số cụm thu được
  - Thuật toán có độ phức tạp trung bình là đa thức
- K-means là giải thuật phân cụm được dùng phổ biến nhất

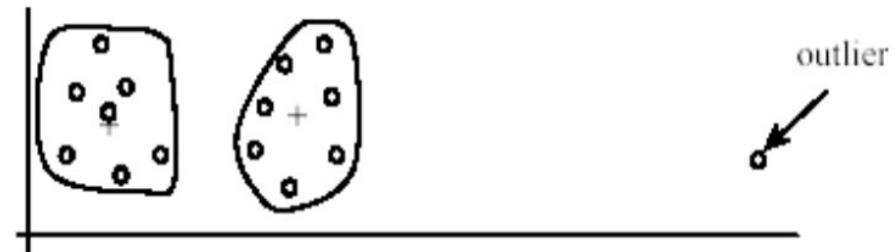
# KMeans - Phân cụm bằng học máy

## Các nhược điểm

- Số cụm  $K$  phải được biết/xác định trước
- Nhạy cảm (gặp lỗi) với các quan sát ngoại lai (nhiễu/outliners)
  - Các quan sát ngoại lai là các quan sát (rất) khác biệt với tất cả các quan sát còn lại
  - Các quan sát ngoại lai có thể do lỗi trong quá trình thu thập/lưu dữ liệu
  - Các quan sát ngoại lai có các giá trị thuộc tính (rất) khác biệt với các giá trị thuộc tính của các quan sát khác



(A): Undesirable clusters



(B): Ideal clusters

# KMeans - Phân cụm bằng học máy

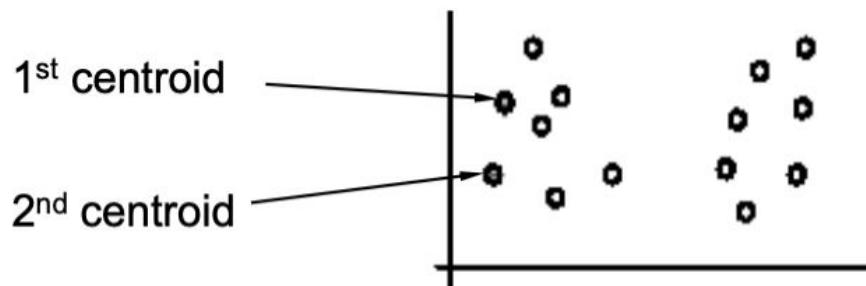
## Giải quyết vấn đề “ngoại lai”

- Giải pháp 1: Trong quá trình phân cụm, cần loại bỏ một số các quan sát khác biệt với (cách xa) các điểm trung tâm (*centeroid*) so với các quan sát khác.
  - Để chắc chắn (không loại nhầm), theo dõi các quan sát ngoại lai (outliners) qua một vài (thay vì chỉ 1) bước lặp phân cụm, trước khi quyết định loại bỏ.
- Giải pháp 2: Thực hành việc lấy ngẫu nhiên (random sampling) một tập nhỏ từ **D** để học *K* cụm
  - Do đây là tập con nhỏ của tập dữ liệu ban đầu, nên khả năng một ngoại lai (outlier) được chọn là nhỏ
  - Gán các quan sát còn lại của tập dữ liệu vào các cụm tùy theo đánh giá về khoảng cách (hoặc độ tương tự)

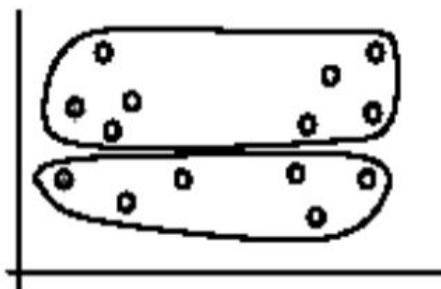
# KMeans - Phân cụm bằng học máy

## Các nhược điểm

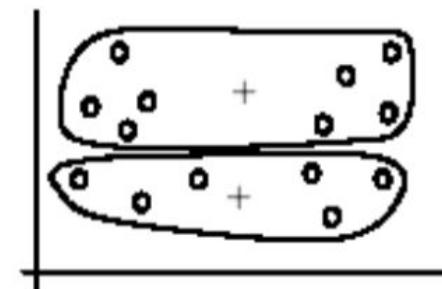
- Giải thuật K-means phụ thuộc vào việc chọn các điểm trung tâm ban đầu (*initial centroids*)



(A). Random selection of seeds (centroids)



(B). Iteration 1



(C). Iteration 2

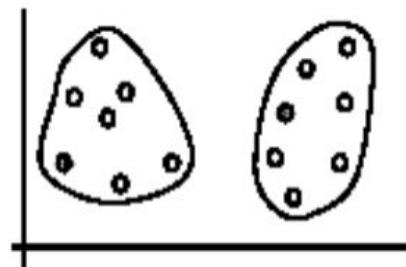
# KMeans - Phân cụm bằng học máy

## Giải quyết vấn đề về *initial centroids*

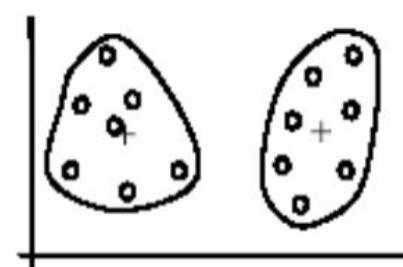
- Kết hợp nhiều kết quả phân cụm với nhau → kết quả tốt hơn
  - Thực hiện giải thuật K-means nhiều lần, mỗi lần bắt đầu với một tập các hạt nhân được chọn ngẫu nhiên



(A). Random selection of  $k$  seeds (centroids)



(B). Iteration 1



(C). Iteration 2

# KMeans - Phân cụm bằng học máy

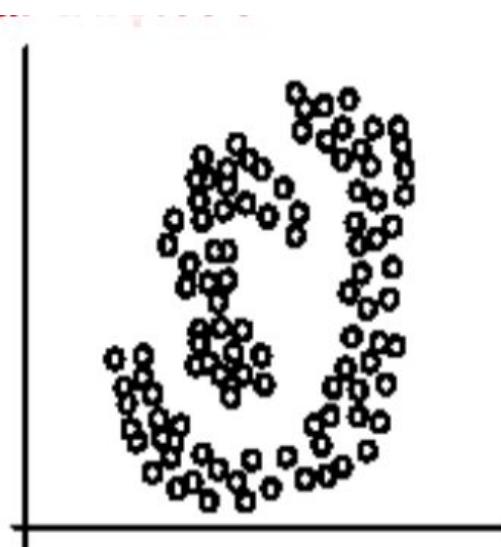
## Giải quyết vấn đề về *initial centroids*

- Một cách chọn hạt nhân nên dùng:
  - Lựa chọn ngẫu nhiên hạt nhân thứ 1 ( $m_1$ )
  - Lựa chọn hạt nhân thứ 2 càng xa hạt thứ 1 càng tốt
  - ...
  - Lựa chọn hạt nhân từ i càng xa càng tốt so với hạt nhân gần nhất trong số  $\{m_1, m_2, m_3, \dots, m_{i-1}\}$
  - ....
- Đây được gọi là phương pháp **K-means++**  
**[Arthur, D.; Vassilvitskii, 2007]**

# KMeans - Phân cụm bằng học máy

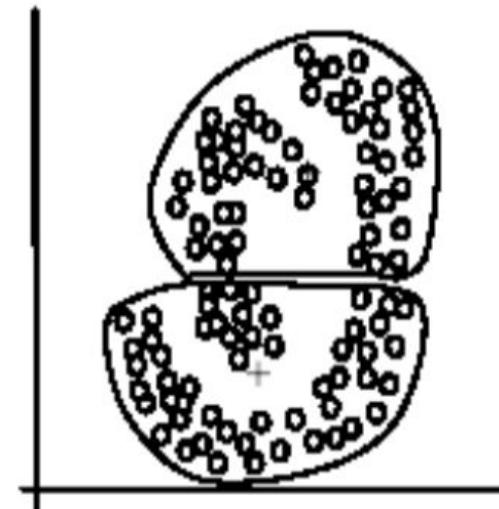
## Các nhược điểm

- K-means (với khoảng cách Euclid) phù hợp với các cụm hình cầu.
- K-means không phù hợp để phát hiện các cụm (nhóm) không có dạng hình cầu.
- Cải thiện??



(A): Two natural clusters

[Liu, 2006]



(B):  $k$ -means clusters

# KMeans - Phân cụm bằng học máy

## Tổng kết

- Mặc dù có những nhược điểm như trên, *k-means* vẫn là giải thuật phổ biến nhất được dùng để giải quyết các bài toán phân cụm – do tính đơn giản và hiệu quả.
  - Các giải thuật phân cụm khác cũng có ưu và nhược điểm riêng
- So sánh hiệu năng của các giải thuật phân cụm là một nhiệm vụ khó khăn (thách thức).
  - Làm sao để biết kết quả thu được là chính xác?
- Link demo KMeans:
  - <https://www.naftaliharris.com/blog/visualizing-k-means-clustering>
  - <http://alekseynp.com/viz/k-means.html>

# Online KMeans - ý tưởng

## K-means

- Cần dùng toàn bộ dữ liệu tại mỗi bước lặp
- Do đó không thể làm việc khi dữ liệu quá lớn (big data)
- Không phù hợp với luồng dữ liệu (stream data, dữ liệu đến liên tục)

**Online K-means** cải thiện nhược điểm của K-means, cho phép ta phân cụm dữ liệu rất lớn, hoặc phân cụm luồng dữ liệu.

- Online learning
- Stochastic gradient descent (SGD)

# Online KMeans - ý tưởng

- K-means tìm K tâm cụm và gán các quan sát  $\{x_1, \dots, x_M\}$  vào các cụm đó bằng cách cực tiểu hóa hàm lỗi sau

$$Q(w) = \sum_{i=1}^M \|x_i - w(x_i)\|_2^2$$

- Trong đó  $w(x_i)$  là tâm gần nhất với  $x_i$ .
- Online K-means cực tiểu hàm Q theo phương pháp leo đồi và dùng thông tin đạo hàm (gradient) của Q.
  - Tuy nhiên tại mỗi bước lặp t ta chỉ lấy một phần thông tin gradient
  - Phần gradient này thu được từ các quan sát tại bước t. Ví dụ:

$$x_t - w_t(x_t)$$

# Online KMeans - thuật toán

- Khởi tạo  $K$  tâm ban đầu
- Cập nhật các tâm mỗi khi một điểm dữ liệu mới đến:
  - Tại bước  $t$ , lấy một quan sát  $x_t$ .
  - Tìm tâm  $w_t$  gần nhất với  $x_t$ . Sau đó cập nhật lại  $w_t$  như sau:

$$w_{t+1} = w_t + g_t(x_t - w_t)$$

$g_t$ : tốc độ học

# Online KMeans - tốc độ học

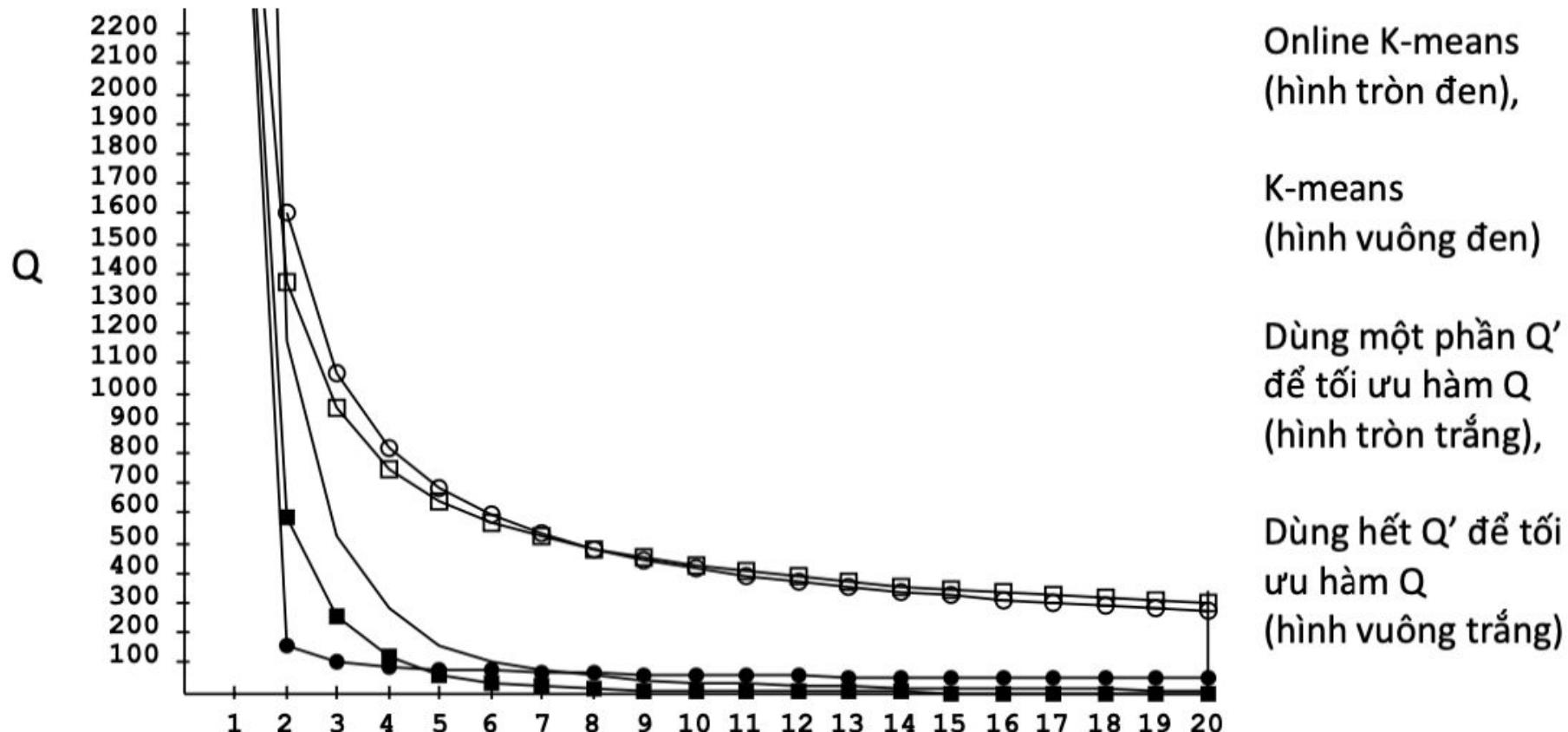
- Một các lựa chọn tốc độ học hay dùng:

$$g_t = (t + t)^{-k}$$

- $\tau, \kappa$  là các hằng số dương.
- $\kappa \in (0.5, 1]$  là tốc độ lãng quên.  $k$  càng lớn thì sẽ nhớ quá khứ càng lâu; các quan sát mới càng ít đóng góp vào mô hình hơn.

# Online KMeans - tốc độ hội tụ

- Hàm Q giảm khi số lần lặp tăng lên (so sánh các phương pháp khác nhau)



# Hierarchical Clustering

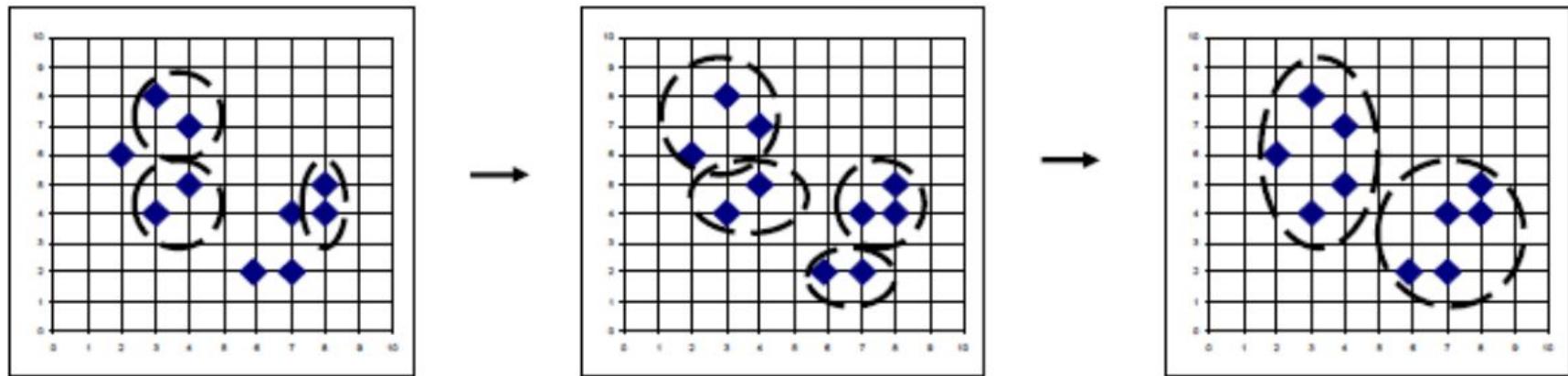
- Ý tưởng:
  - Xuất phát mỗi cụm có một đối tượng (nếu có  $n$  đối tượng thì sẽ có  $n$  cụm).
  - Tiếp theo, tiến hành gộp các cụm cặp hai đối tượng có khoảng cách bé nhất.
  - Quá trình ghép cặp tiến hành lặp cho đến khi các cụm được ghép thành một cụm duy nhất.

# Hierarchical Clustering

- Phân cụm dữ liệu bằng phân cấp (hierarchical clustering): nhóm các đối tượng vào cây phân cấp của các cụm
  - Agglomerative: bottom-up (trộn các cụm)
  - Divisive: top-down (phân tách các cụm)
- Không yêu cầu thông số nhập k (số cụm)
- Yêu cầu điều kiện dừng
- Không thể quay lui ở mỗi bước trộn/phân tách

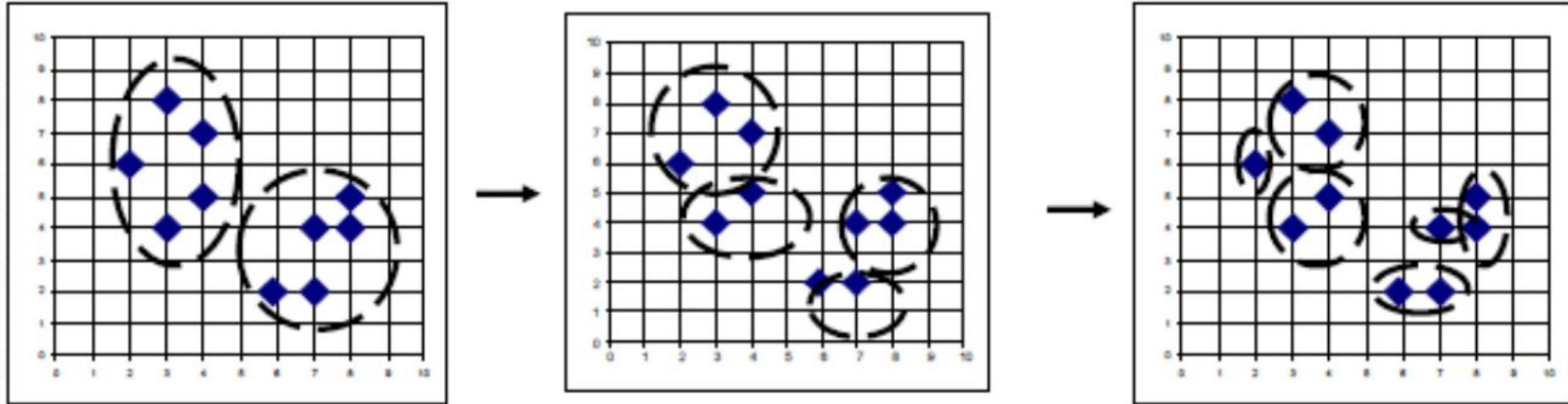
# Hierarchical Clustering

- AGNES (Agglomerative Nesting)
  - Khởi tạo: mỗi đối tượng là một cụm (lá)
  - Đệ quy trộn các nút có sự khác nhau thấp nhất
  - Tiêu chí: min distance, max distance, avg distance, center distance
  - Cuối cùng tất cả các nút thuộc về một cụm (gốc)



# Hierarchical Clustering

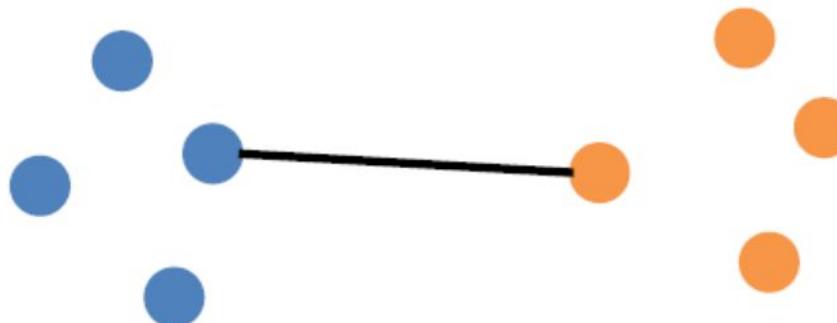
- DIANA (Divisive Analysis)
  - Ngược thứ tự so với AGNES
  - Bắt đầu cụm gốc chứa tất cả các đối tượng
  - Thực hiện Đệ quy chia thành các cụm nhỏ
  - Cuối cùng, mỗi cụm chứa một đối tượng duy nhất



# Hierarchical Clustering

- Khoảng cách giữa hai cụm có thể là một trong các loại sau:
  - Single-linkage clustering: khoảng cách giữa hai cụm là **khoảng cách ngắn nhất giữa hai đối tượng của hai cụm**

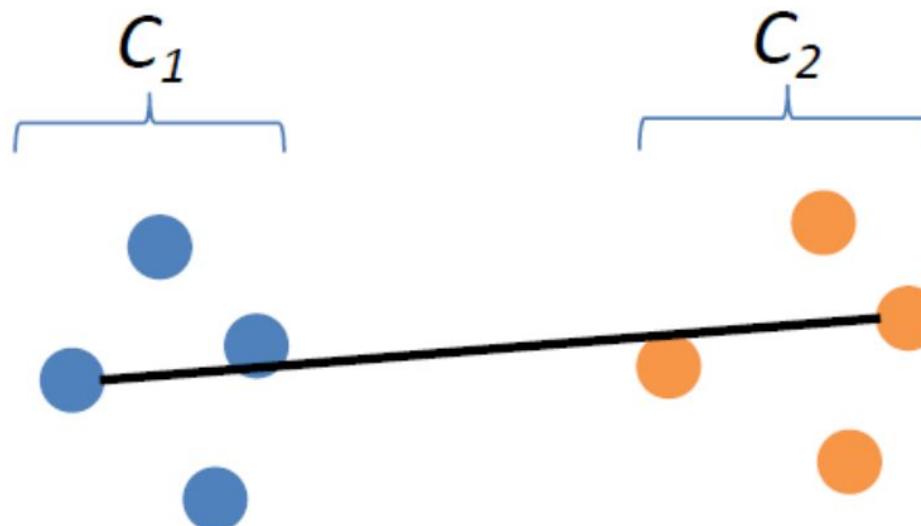
$$D(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} D(x_1, x_2)$$



# Hierarchical Clustering

- Complete-linkage clustering: khoảng cách giữa hai cụm là **khoảng cách lớn nhất giữa hai đối tượng của hai cụm.**

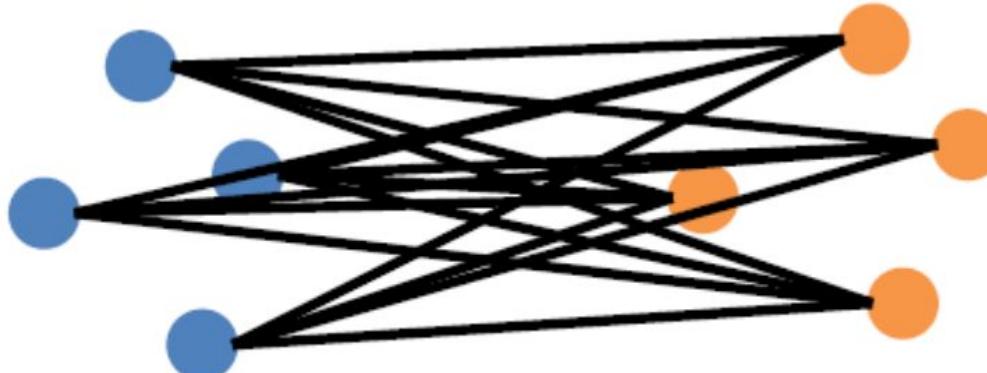
$$D(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} D(x_1, x_2)$$



# Hierarchical Clustering

- Average-linkage clustering: khoảng cách giữa hai cụm là **khoảng cách trung bình giữa hai đối tượng của hai cụm**

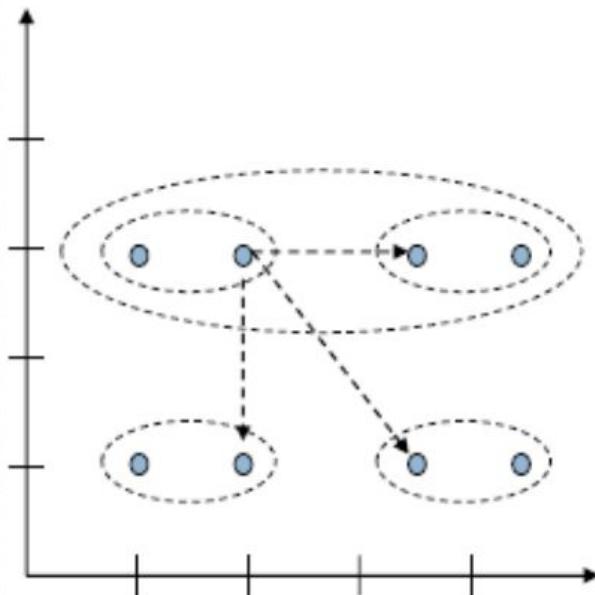
$$D(C_1, C_2) = \frac{1}{|C_1|} \frac{1}{|C_2|} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} D(x_1, x_2)$$



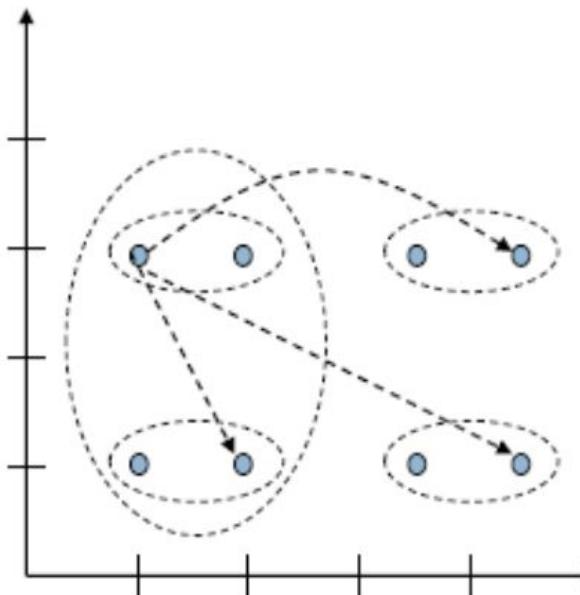
# Hierarchical Clustering

Tiêu chí trộn các cụm: single-linkage, complete-linkage, và average-linkage

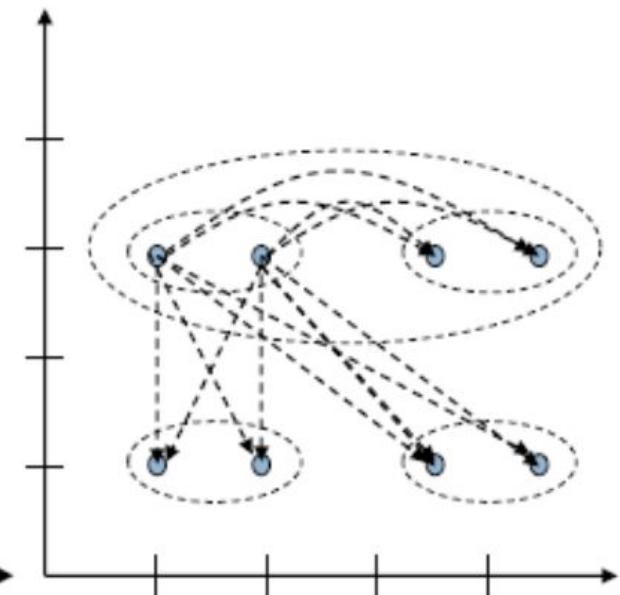
**Single-linkage**



**Complete-linkage**



**Average-linkage**

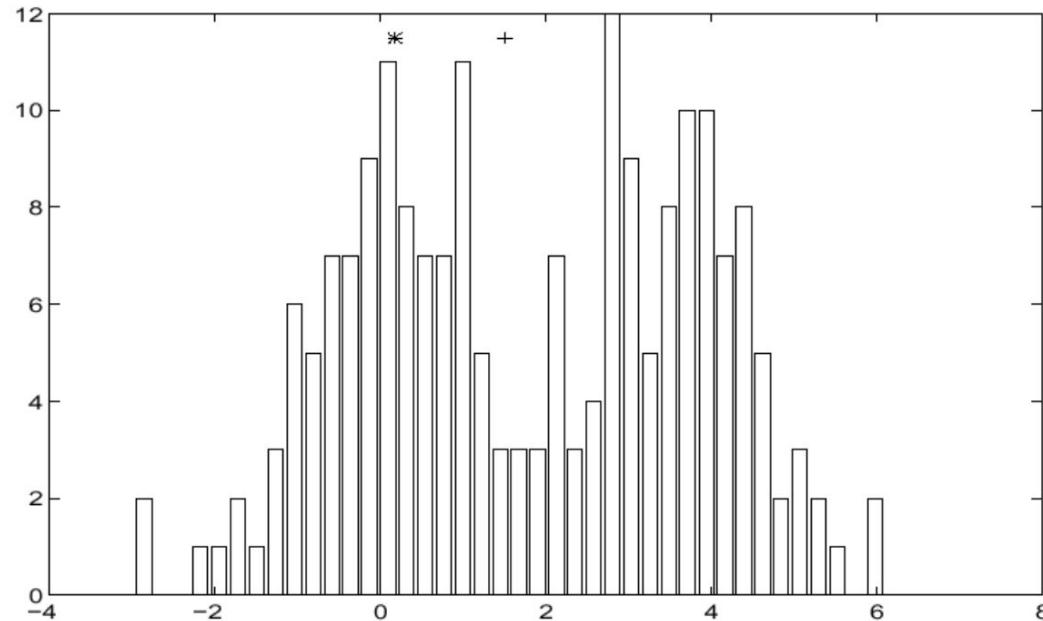


# Density-Based Clustering

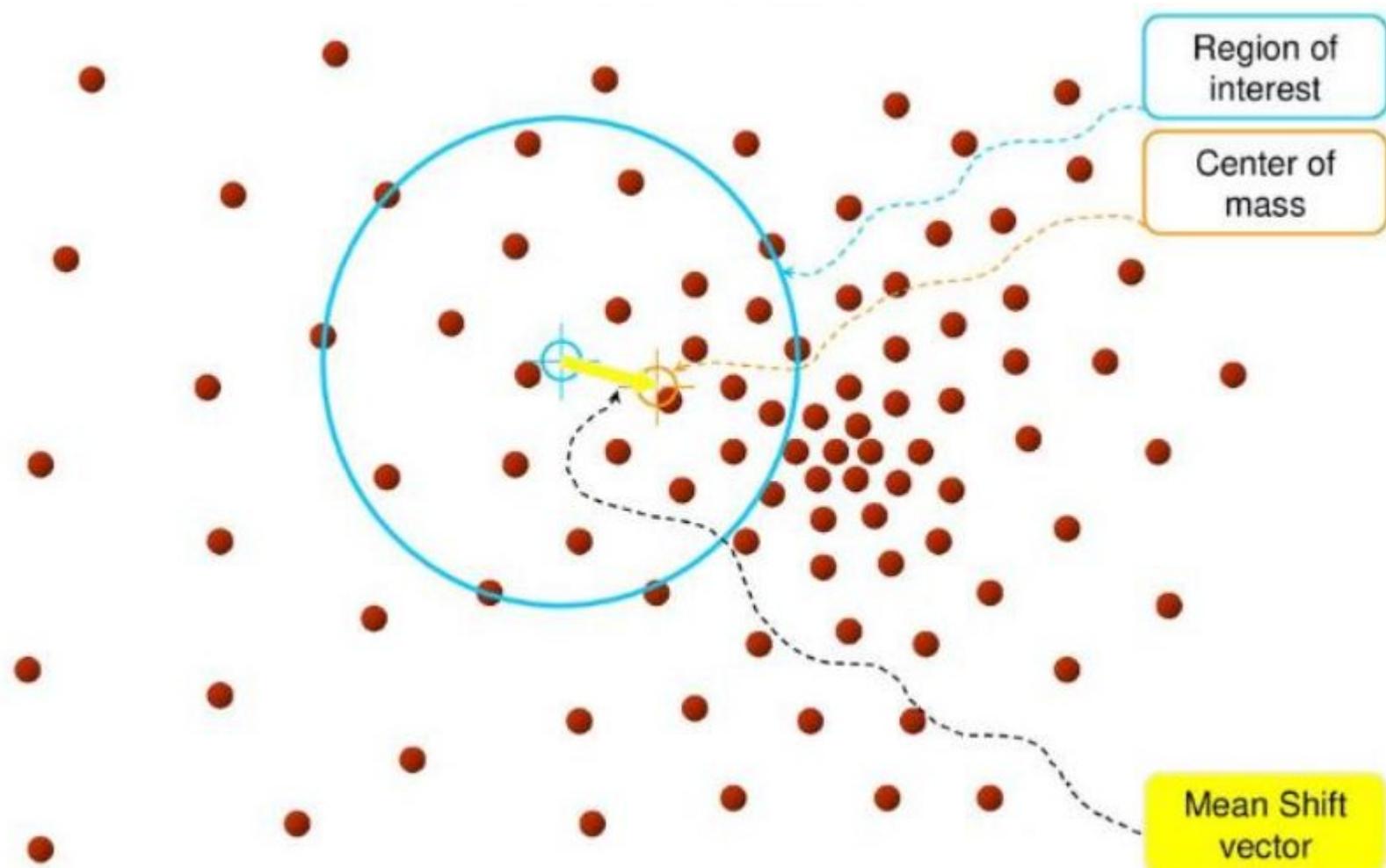
- Phân cụm dữ liệu dựa trên mật độ
  - Mỗi cụm là một vùng dày đặc (dense region) gồm các đối tượng.
  - Các đối tượng trong vùng thưa hơn được xem là nhiễu.
  - Mỗi cụm có dạng tùy ý.
- Giải thuật
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - OPTICS (Ordering Points To Identify the Clustering Structure)
  - DENCLUE (DENsity-based CLUstEring)
  - MeanShift

# MeanShift

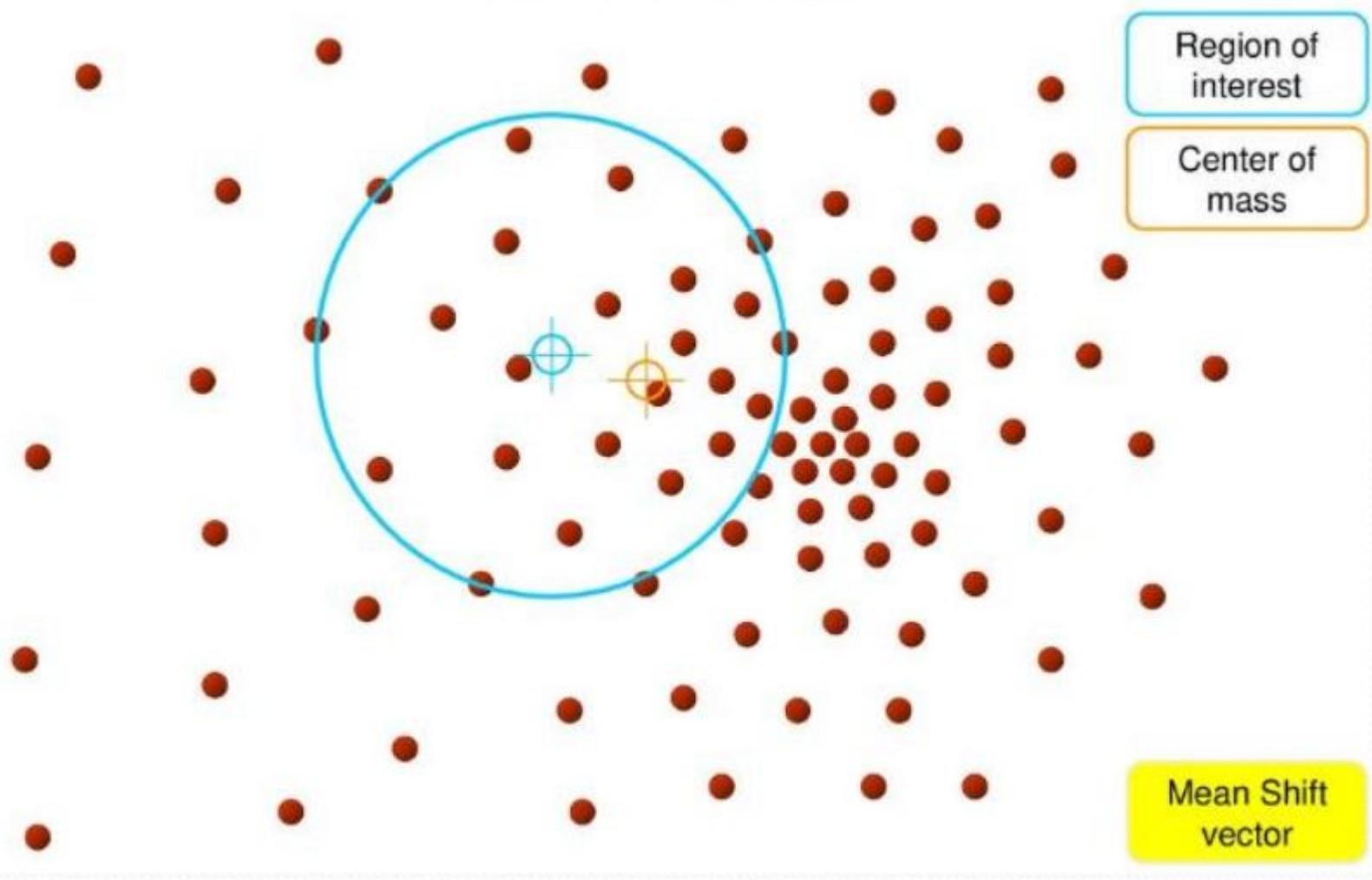
- Tìm kiếm đỉnh (mode) bằng thủ tục lặp
  - Khởi tạo các điểm ngẫu nhiên, và cửa sổ  $W$
  - Tính trọng tâm (“mean”) của cửa sổ  $W$ :  $\sum_{x \in W} xH(x)$
  - Tịnh tiến cửa sổ tìm kiếm tới trọng tâm
  - Lặp lại bước 2 cho tới khi hội tụ



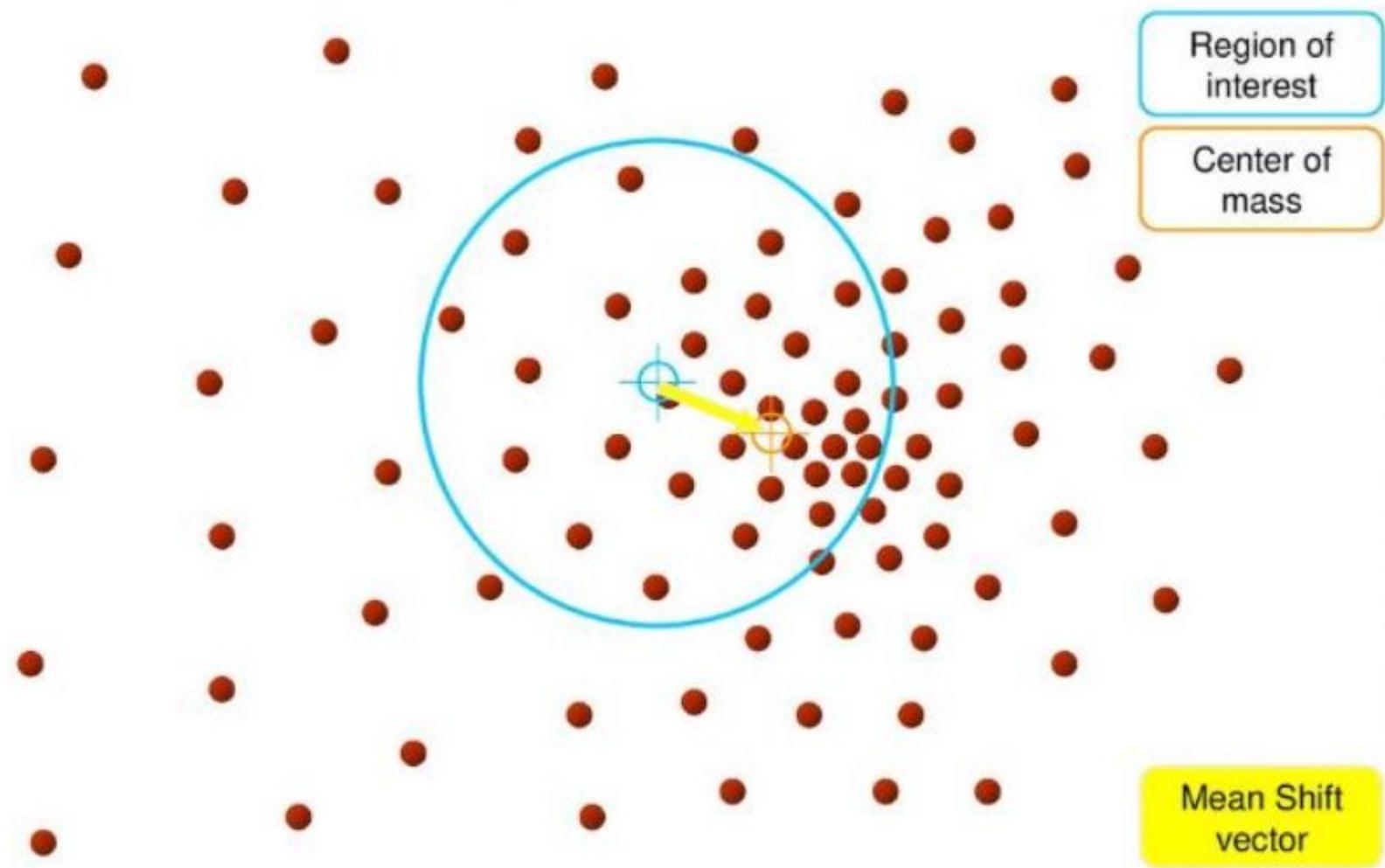
# MeanShift



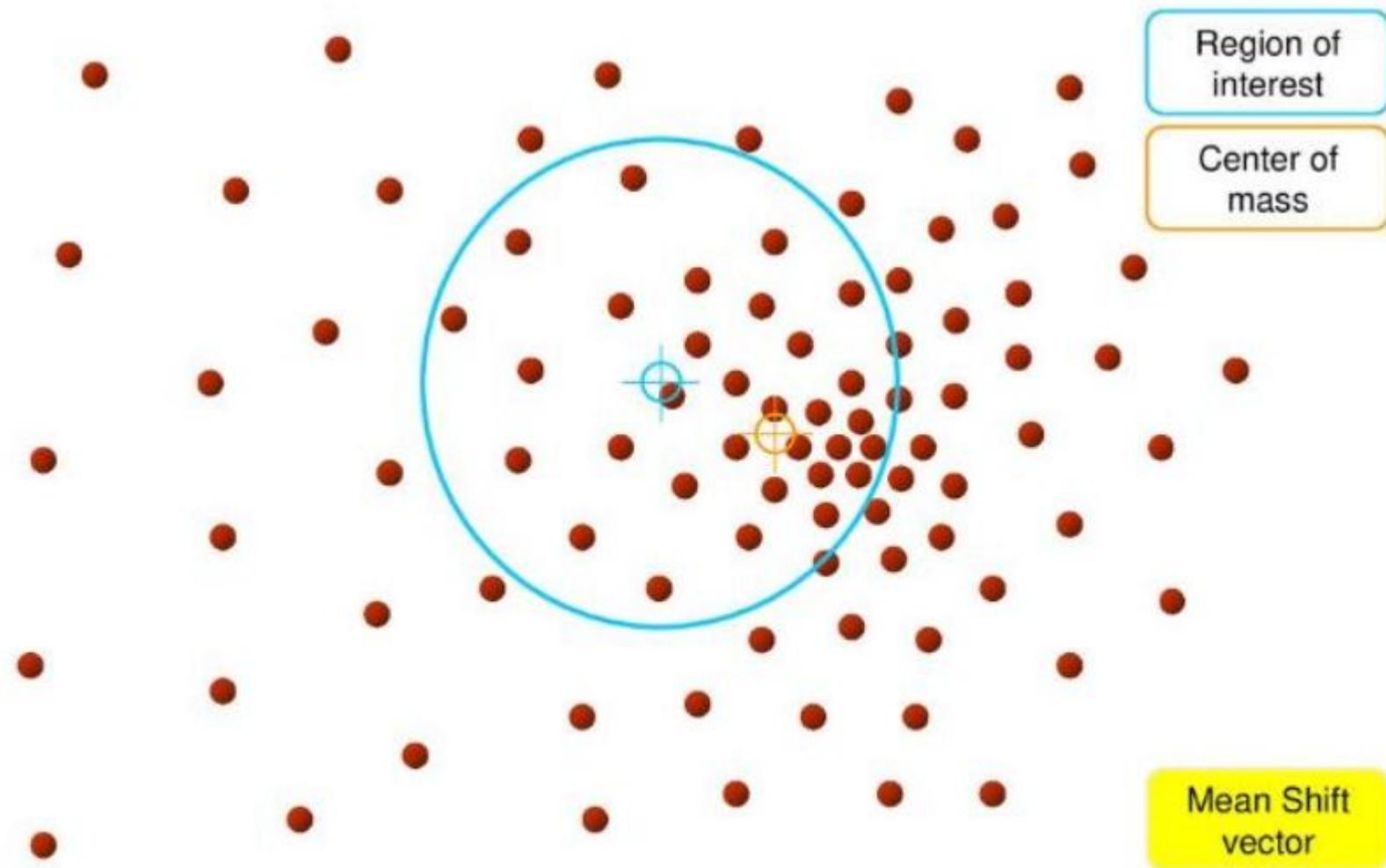
# MeanShift



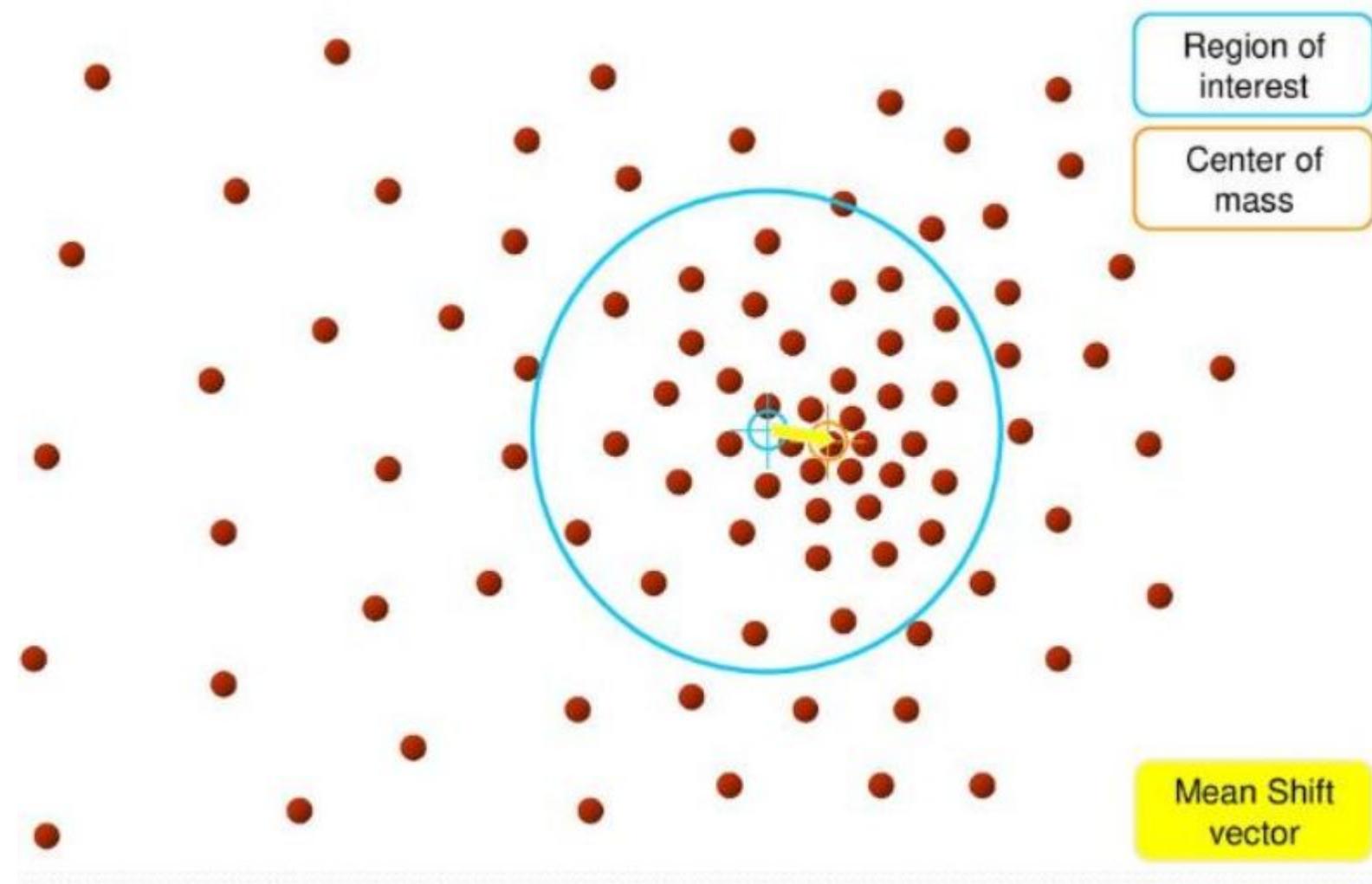
# MeanShift



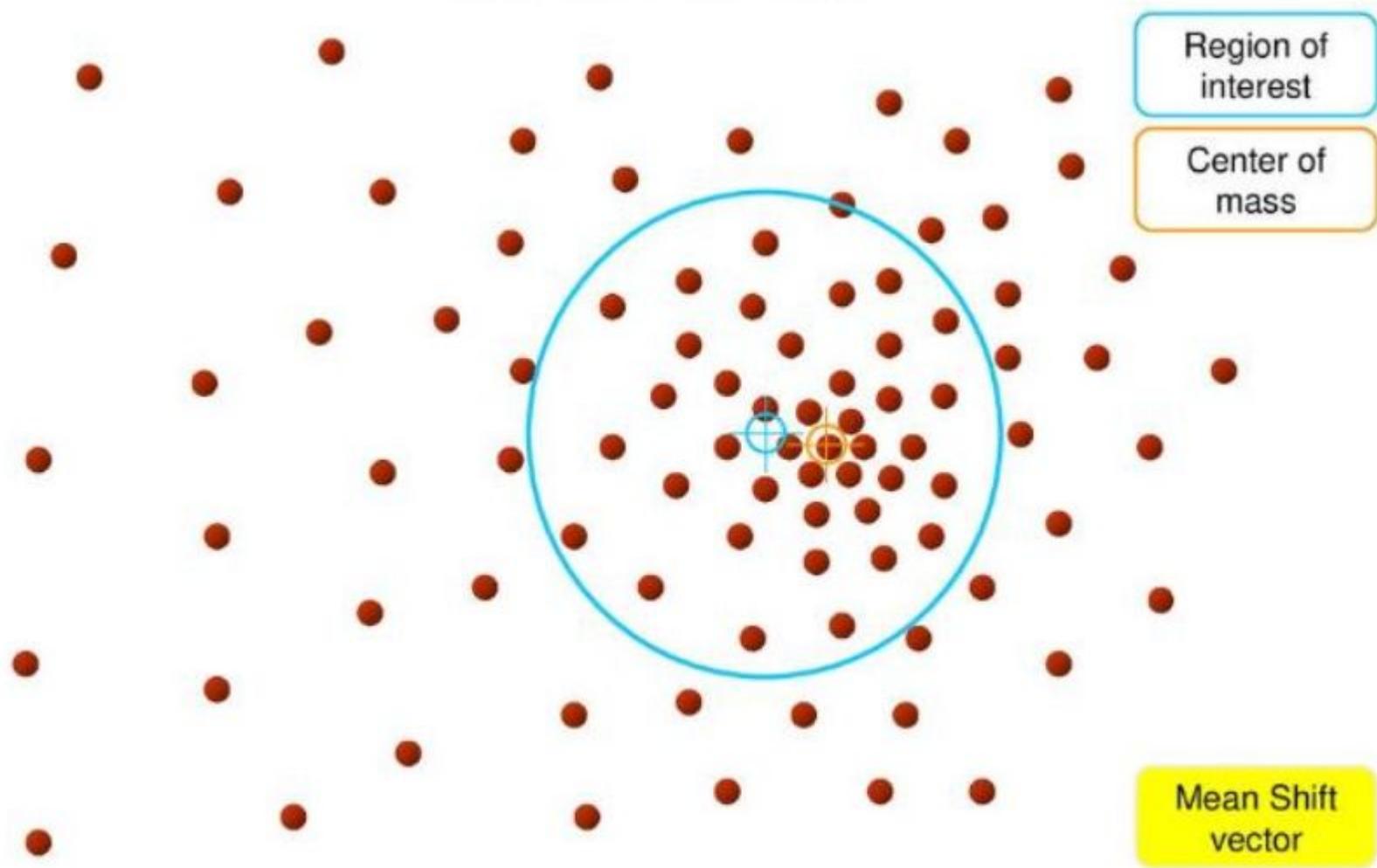
# MeanShift



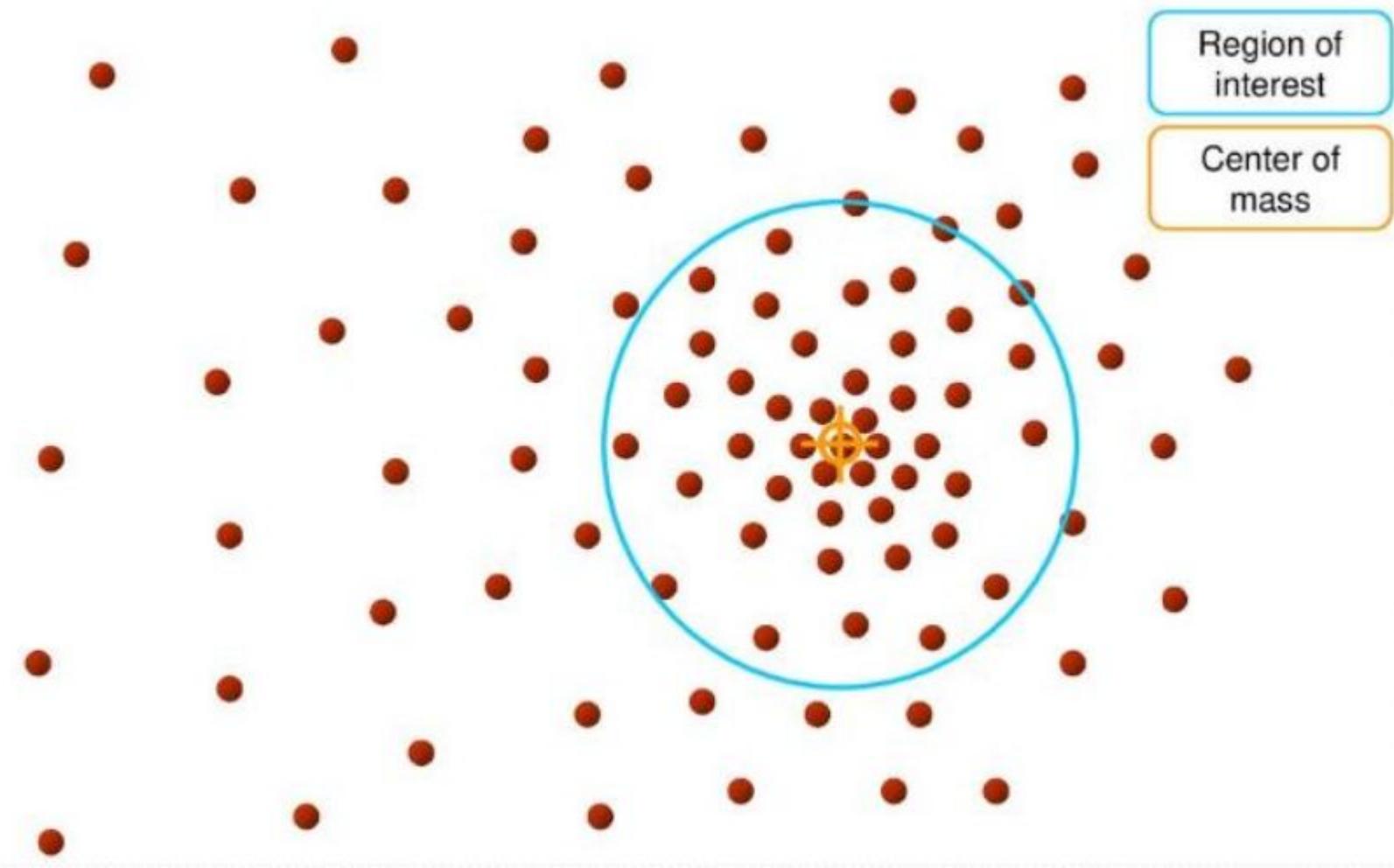
# MeanShift



# MeanShift

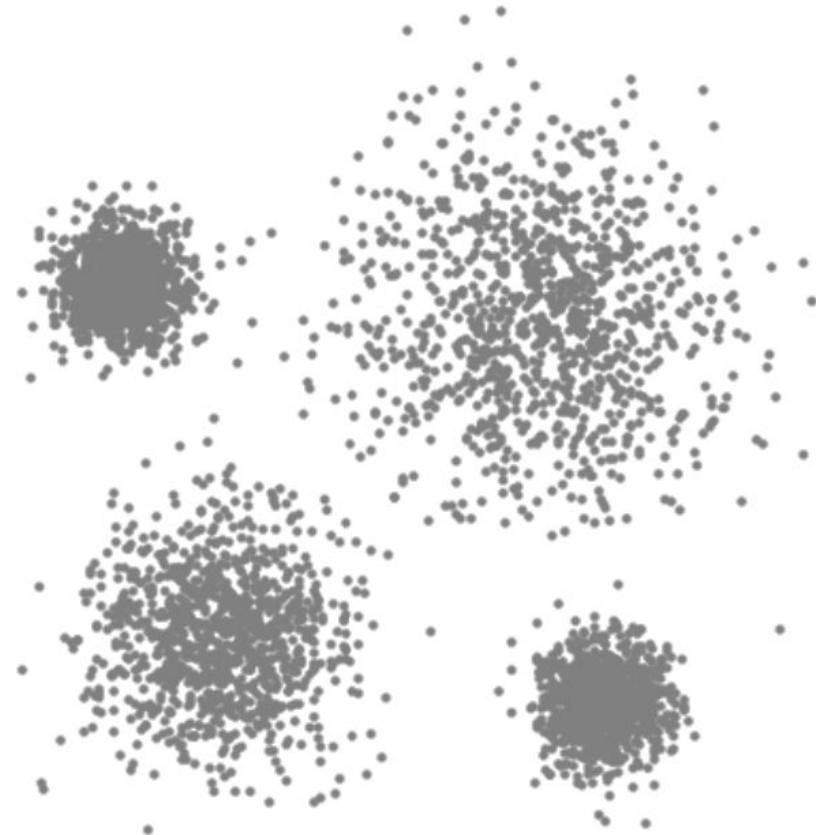


# MeanShift



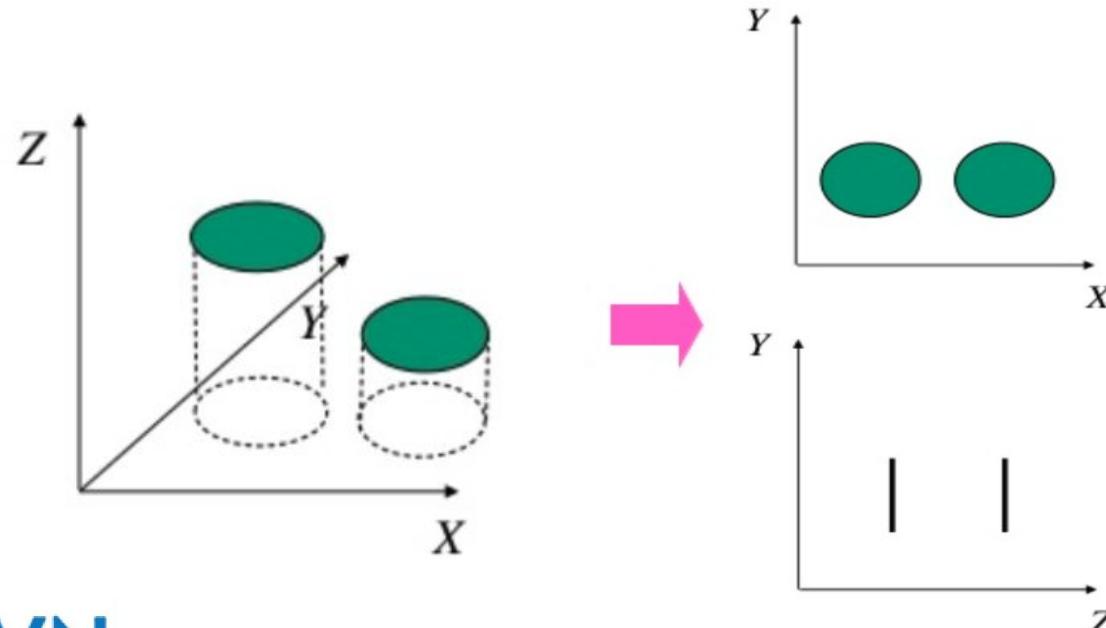
# MeanShift

- MeanShift không cần chọn trước số nhóm cần phân loại. Thuật toán có thể tìm được số nhóm vì chúng sẽ dịch chuyển tự động.
- Vấn đề trong MeanShift là chọn window - bán kính vùng quét để tính mean - là bao nhiêu



# Giảm chiều dữ liệu (dimensionality reduction)

- Là quá trình biến đổi dữ liệu từ không gian chiều cao thành không gian chiều thấp để biểu diễn ở dạng chiều thấp đồng thời giữ lại một số thuộc tính có ý nghĩa của dữ liệu gốc
- Sử dụng như tiền xử lý (preprocessing) khi muốn sử dụng số lượng thuộc tính tít hơn mà giữ lại thông tin nhiều nhất có thể

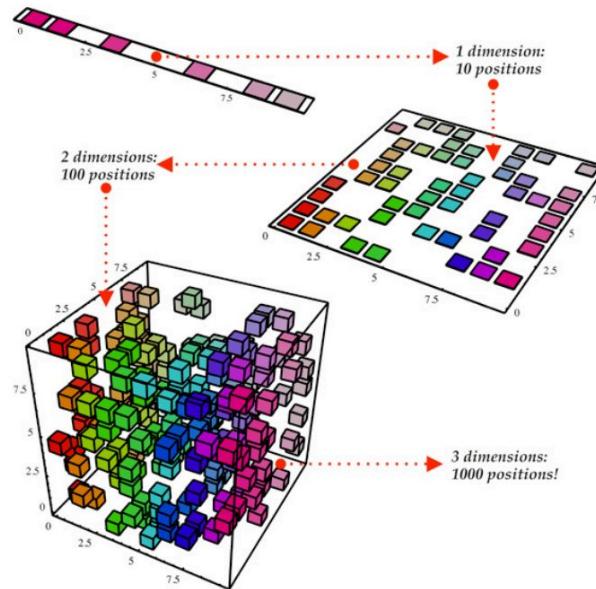


# Giảm chiều dữ liệu (dimensionality reduction)

- Đặc điểm
  - Ưu điểm
    - Giảm overfitting
    - Loại bỏ đặc trưng không cần thiết hoặc mờ nhạt
    - Giảm yêu cầu về tài nguyên cần trong tính toán
    - Dễ dàng trực quan hóa dữ liệu
  - Nhược điểm
    - Mất một lượng thông tin nhất định gây ảnh hưởng đến hiệu quả.
- Gồm 2 dạng chính:
  - Feature selection (lựa chọn đặc trưng)
    - Lựa chọn trực tiếp mà không biến đổi
    - Non-trivial problem: tìm các tổ hợp các đặc trưng tốt nhất.(5 đặc trưng tốt nhất chưa chắc ghép lại thành tổ hợp tốt nhất)
  - Feature extraction (trích chọn đặc trưng)
    - Biến đổi và kết hợp các đặc trưng thành đặc trưng mới

# Giảm chiều dữ liệu (dimensionality reduction)

- Các thuật toán cơ bản
  - Tuyến tính
    - Principal Component Analysis (PCA)
    - Latent Dirichlet Allocation (LDA)
  - Phi tuyến
    - Isomap
    - t-distributed Stochastic Neighbor Embedding (t-SNE)



# PCA - Giảm chiều dữ liệu

- Tuyến tính
- Phổ biến trong việc giảm chiều dữ liệu

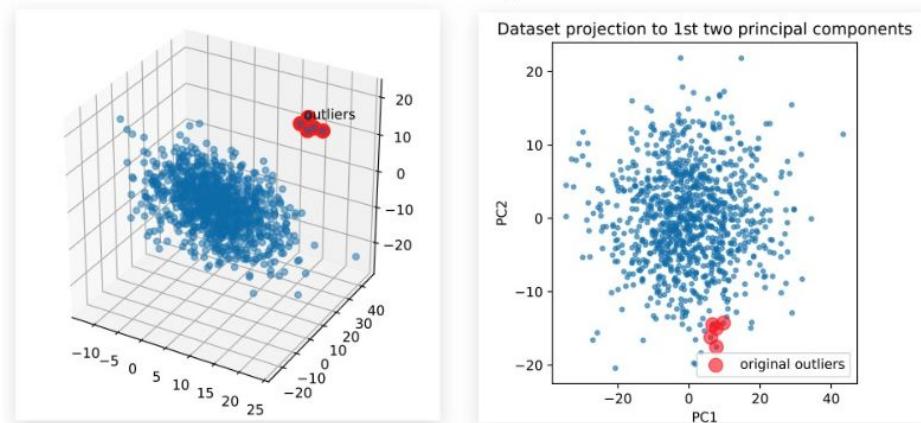
## Ý tưởng:

- Tìm các hướng mà dữ liệu thay đổi nhiều nhất là toạ độ mới
- Chiếu dữ liệu lên toạ độ mới tìm được (ít chiều hơn)

## Đặc điểm

- Hoạt động tốt với dữ liệu có phân phối ổn định
- Không tốt khi có nhiễu (outliers)

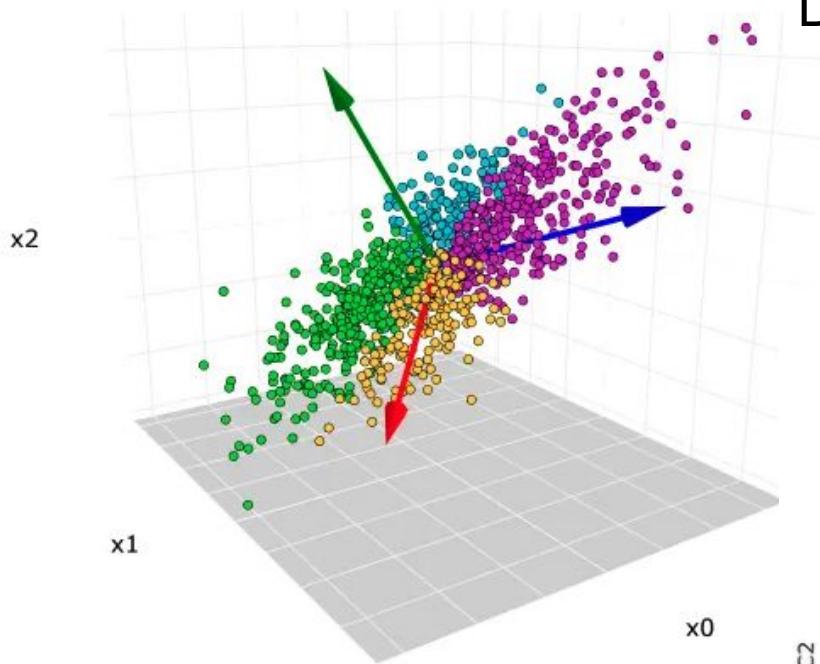
Dataset với 1000 samples có 5 outliers



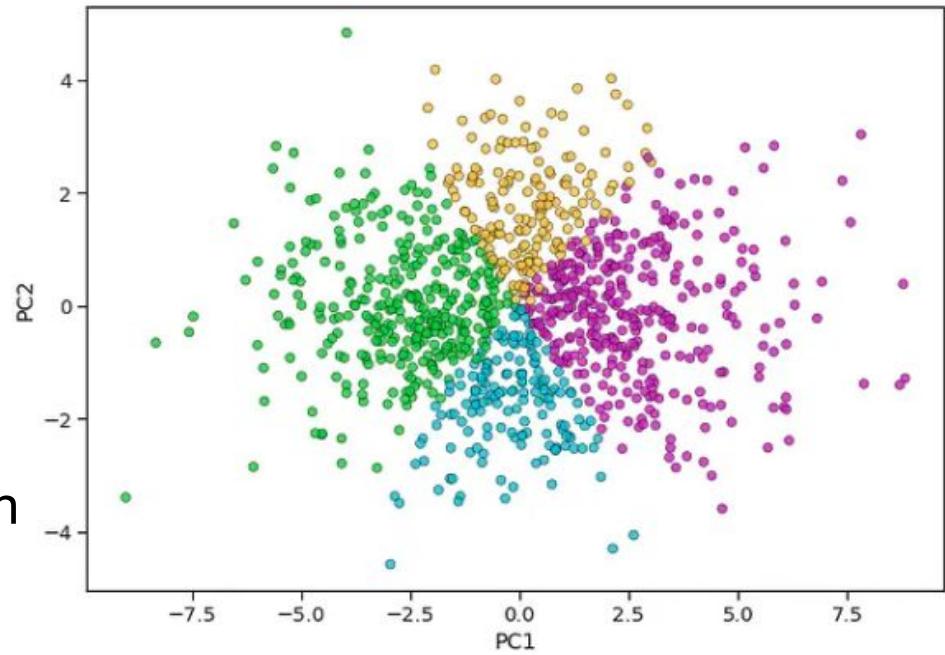
- Lấy một ví dụ về việc có hai camera đặt dùng để chụp một con người, một camera đặt phía trước người và một camera đặt trên đầu.
- Rõ ràng là hình ảnh thu được từ camera đặt phía trước người mang nhiều thông tin hơn so với hình ảnh nhìn từ phía trên đầu.
- Vì vậy, bức ảnh chụp từ phía trên đầu có thể được bỏ qua mà không có quá nhiều thông tin về hình dáng của người đó bị mất.
- PCA chính là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài toạ độ, phần còn lại chỉ mang một lượng nhỏ thông tin.

# PCA - Minh họa

Dữ liệu trong không gian 3D

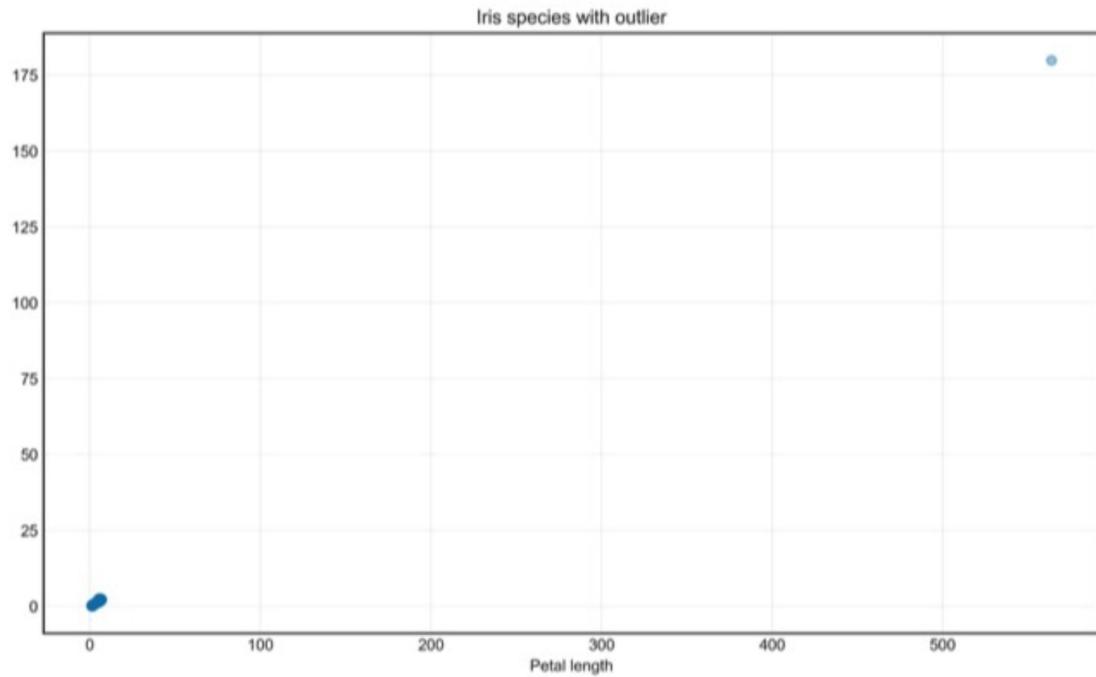


Được xử lý và vẽ lên  
không gian 2D



# Phát hiện bất thường (Anomaly detection)

- Phát hiện điểm bất thường hay điểm ngoại lệ
- Các điểm này rất hiếm và khác các điểm còn lại nên supervised learning là rất khó
- Dùng model để biểu diễn các điểm bình thường
  - Các điểm rất khác với các điểm được biểu diễn bởi model bình thường này được gọi là điểm bất thường
- Mục tiêu: Detect outliers
- Outlier sẽ là quan sát có sự khác biệt lớn so với những quan sát khác



# Phát hiện bất thường (Anomaly detection)

## Bài toán phát hiện bất thường

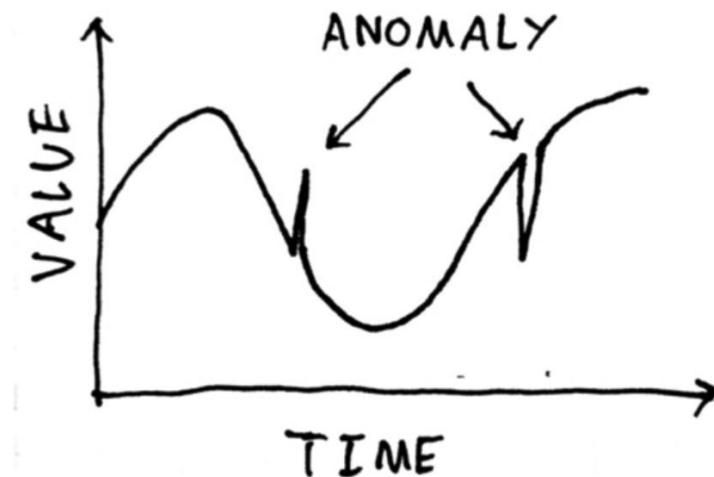
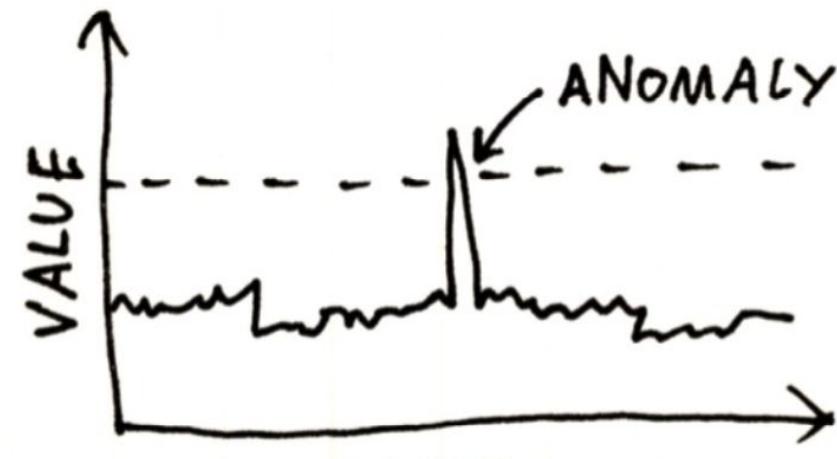
- Tìm ra thiết bị nào hỏng nhanh hơn hoặc tồn tại lâu hơn
- Tìm ra gian lận nào đánh lừa hệ thống
- Bệnh nhân nào đó có khả năng chống lại một căn bệnh nguy hiểm chết người
- Credit card fraud detection
- Network security monitoring
- Heart rate monitoring



shutterstock.com • 379870144

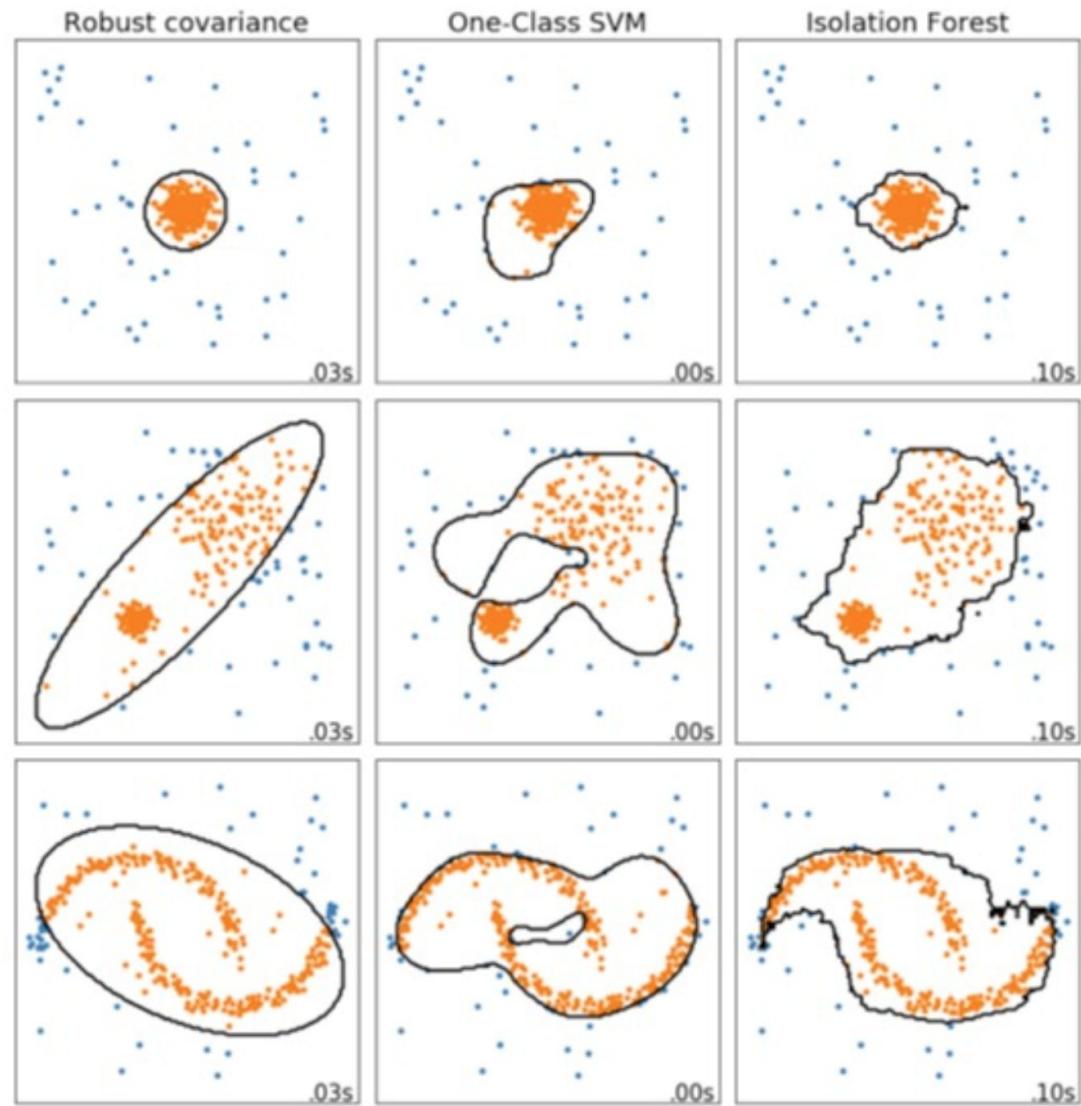
# Anomaly detection

- Cách hoạt động: Xem xét dữ liệu và đưa ra biện pháp
  - Nếu dữ liệu ổn định trong khoảng thời gian đủ dài  
→ Sử dụng ngưỡng
  - Đôi khi sự bất thường không phải ở độ lớn của giá trị mà ở tốc độ thay đổi của nó, như khi nó đột ngột đạt đỉnh hoặc giảm xuống.
  - Trong trường hợp đó cần quan tâm đến đạo hàm của hàm mục tiêu (rate of change)



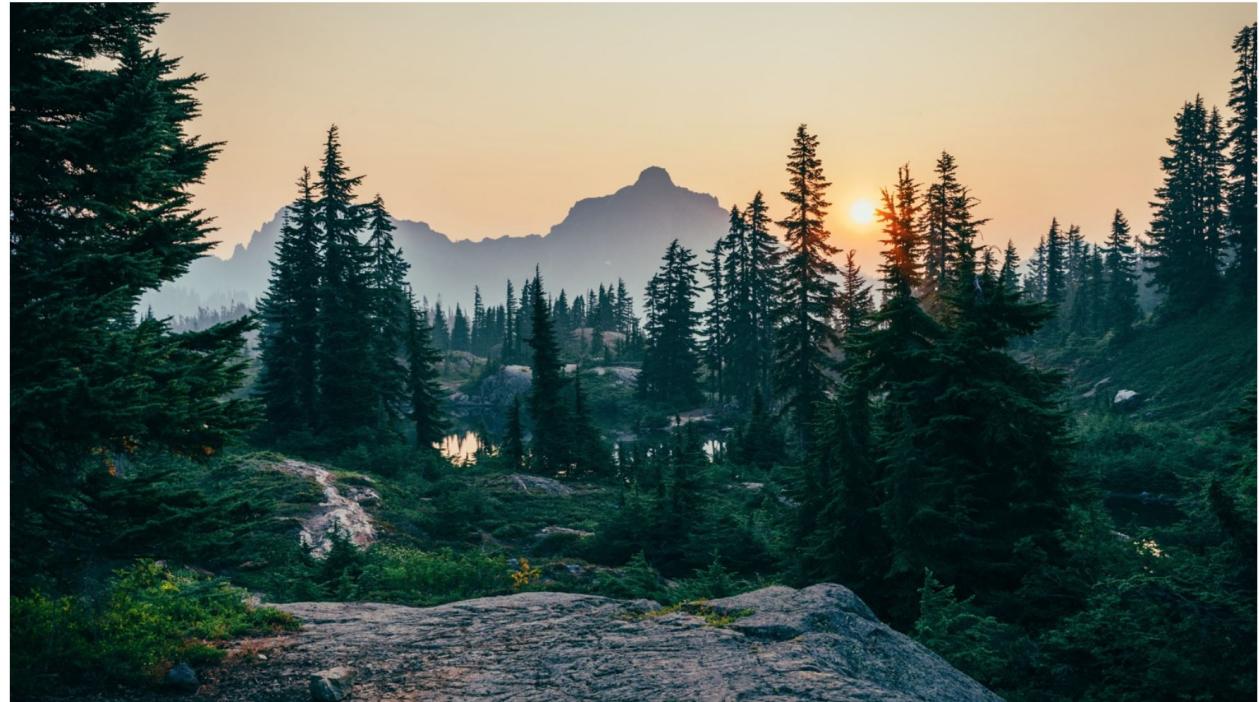
# Anomaly detection

- Thuật toán
  - Isolation Forest
  - Robust covariance
  - One-Class SVM



# Isolation Forest - Anomaly detection

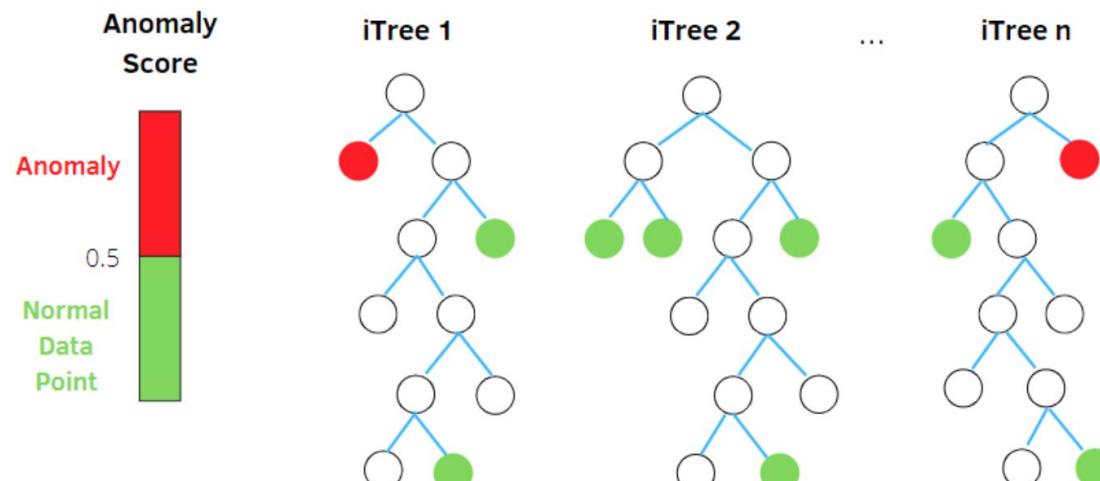
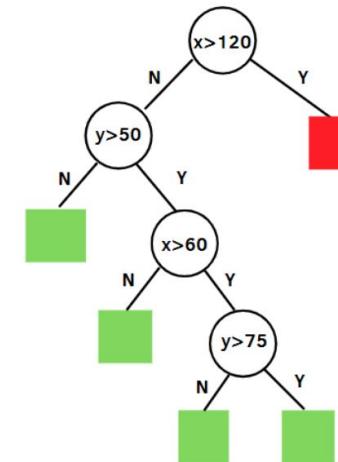
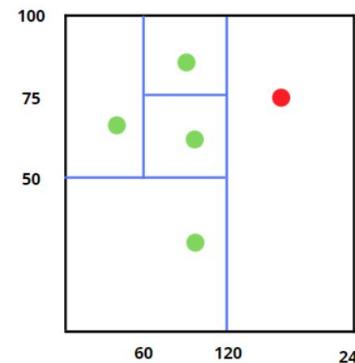
- Ý tưởng:
  - Sử dụng cấu trúc cây được chọn ngẫu nhiên
  - Các điểm dữ liệu nằm sâu phía trong cây: điểm bình thường
  - Các điểm dữ liệu ở các nhánh ngắn: điểm bất thường



# Isolation Forest - Anomaly detection

- Xây dựng một rừng ngẫu nhiên các cây nhị phân
  - Tập ngẫu nhiên dữ liệu được gán cho cây nhị phân
  - Việc phân nhánh cây bằng cách chọn ngẫu nhiên một đặc trưng với một ngưỡng ngẫu nhiên nào đó
  - Nếu giá trị nhỏ hơn ngưỡng: bên trái cây
  - Nếu giá trị lớn hơn ngưỡng: bên phải cây
  - Thực hiện đệ quy cho đến khi tất cả các điểm được cách ly hoàn toàn
  - Lặp lại các bước trên để xây dựng cây nhị phân ngẫu nhiên
- Kiểm tra
  - Một điểm dữ liệu bất thường được đo dựa trên độ sâu cây cần thiết để đến điểm đó

# Isolation Forest - Anomaly detection



# Đánh giá trong bài toán phát hiện bất thường

- Sử dụng supervised metrics (số liệu đo lường được giám sát)
  - Kiểm tra với các điểm bất thường đã được gán nhãn
  - Confusion matrix
- Precision: Bao nhiêu trong số bất thường được phát hiện thực sự là bất thường?
- Recall: Bao nhiêu trong số bất thường thực sự đã được phát hiện?

		PREDICTION	
		NOT OK	OK
THE TRUTH	NOT OK	Actual arrhythmia detected	Failed to detect arrhythmia
	OK	False alarm	Nothing to detect and nothing detected

# Chú ý khi xây dựng model

## LỰA CHỌN ĐÚNG MODEL

- Nếu giá trị đích đã được định nghĩa và đã biết
  - Supervised
    - Classification
    - Regression
- Nếu giá trị đích chưa biết
  - Unsupervised
    - Giảm số chiều
    - Clustering



# Chú ý khi xây dựng model

## Xác định ưu tiên

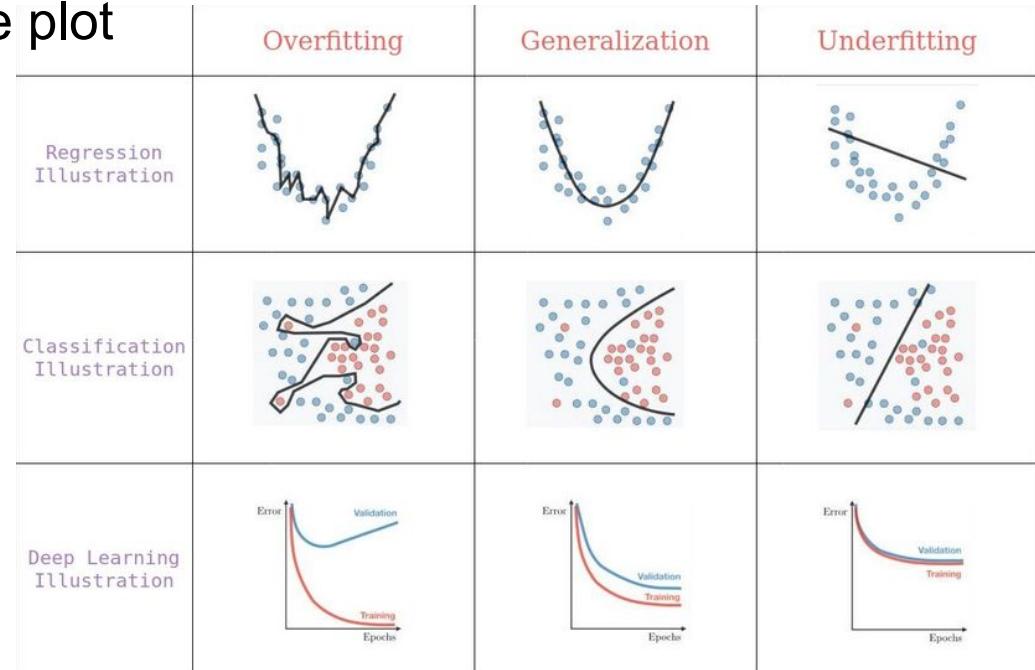
- Khả năng diễn giải mô hình (đơn giản)
  - Hồi quy tuyến tính
  - Cây quyết định
- Mô hình đạt hiệu quả cao (phức tạp)
  - SVM
  - NN
- Chú ý
  - Cây quyết định đơn giản có thể cung cấp cả hiệu quả cần thiết và khả năng diễn giải
  - Ngược lại, có khi SVM và không đạt được cả hai
  - Phụ thuộc vào vấn đề và dữ liệu
  - "Sự đơn giản trước tiên!".

# Sử dụng các chỉ số đo lường (metrics)

- Chỉ số đáp ứng(satisfying metric)
  - Tiêu chí mà mọi mô hình cần phải đáp ứng
  - Ví dụ: độ chính xác tối thiểu, thời gian thực thi tối đa
- Chỉ số tối ưu hoá (optimizing metric)
  - Chỉ số ưu tiên cao nhất dành cho bài toán ví dụ (minimize false positive trong chẩn đoán bệnh, maximize recall trong việc phát hiện gian lận)
- Mô hình cần tìm:
  - Vượt qua tất cả chỉ số đáp ứng và đạt hiệu quả cao nhất ở chỉ số tối ưu hoá
  - Dữ liệu thực tế thay từ liên tục và liên tục kiểm tra sự đáp ứng của model

# Diễn giải mô hình

- Toàn cục
  - Các quy tắc ra quyết định của mô hình là gì
  - Ví dụ:
    - visualize decision tree
    - feature importance plot
- Cục bộ
  - Tại sao mẫu này được phân loại theo cách này?
  - Ví dụ ngân hàng từ chối cung cấp tín dụng nghiên cứu xem các tiêu chí đưa ra là gì



# Tổng kết buổi học

- Các phương pháp học không giám sát
- Các kỹ thuật khai phá dữ liệu

# THANK YOU !

**COLE.VN**  
Connecting knowledge



[www.cole.vn](http://www.cole.vn)