

Phân cụm K-means

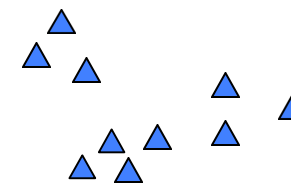
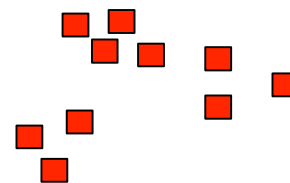
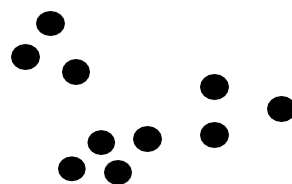
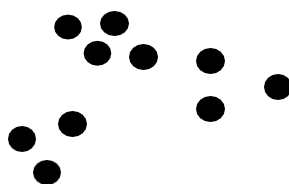
Trình bày: PGS.TS Nguyễn Hữu Quỳnh

Giới thiệu



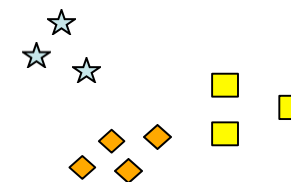
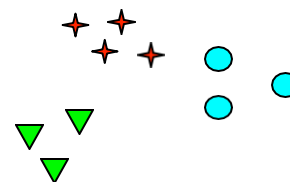
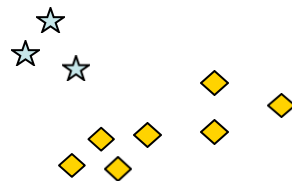
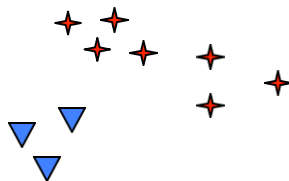
Giới thiệu

Có bao nhiêu clusters



2 clusters

Có bao nhiêu clusters?



4 clusters

6 clusters

Tùy thuộc vào “resolution” !

Giới thiệu

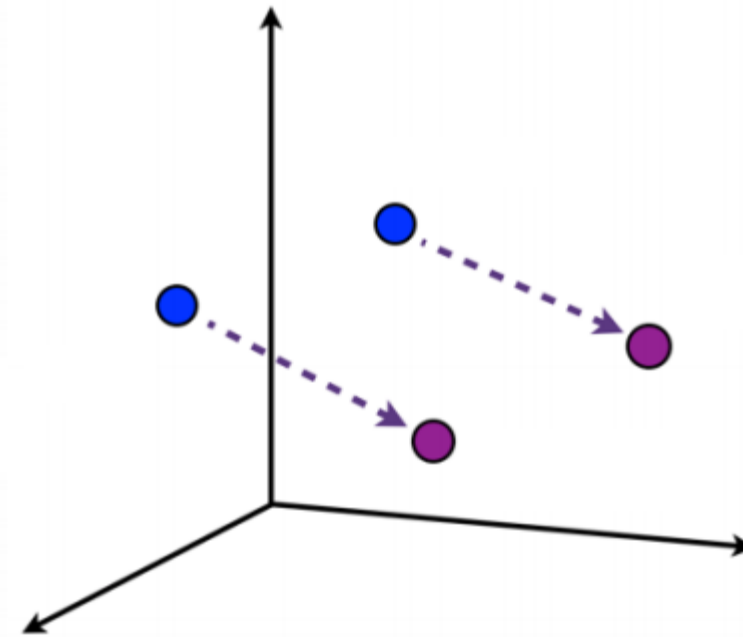
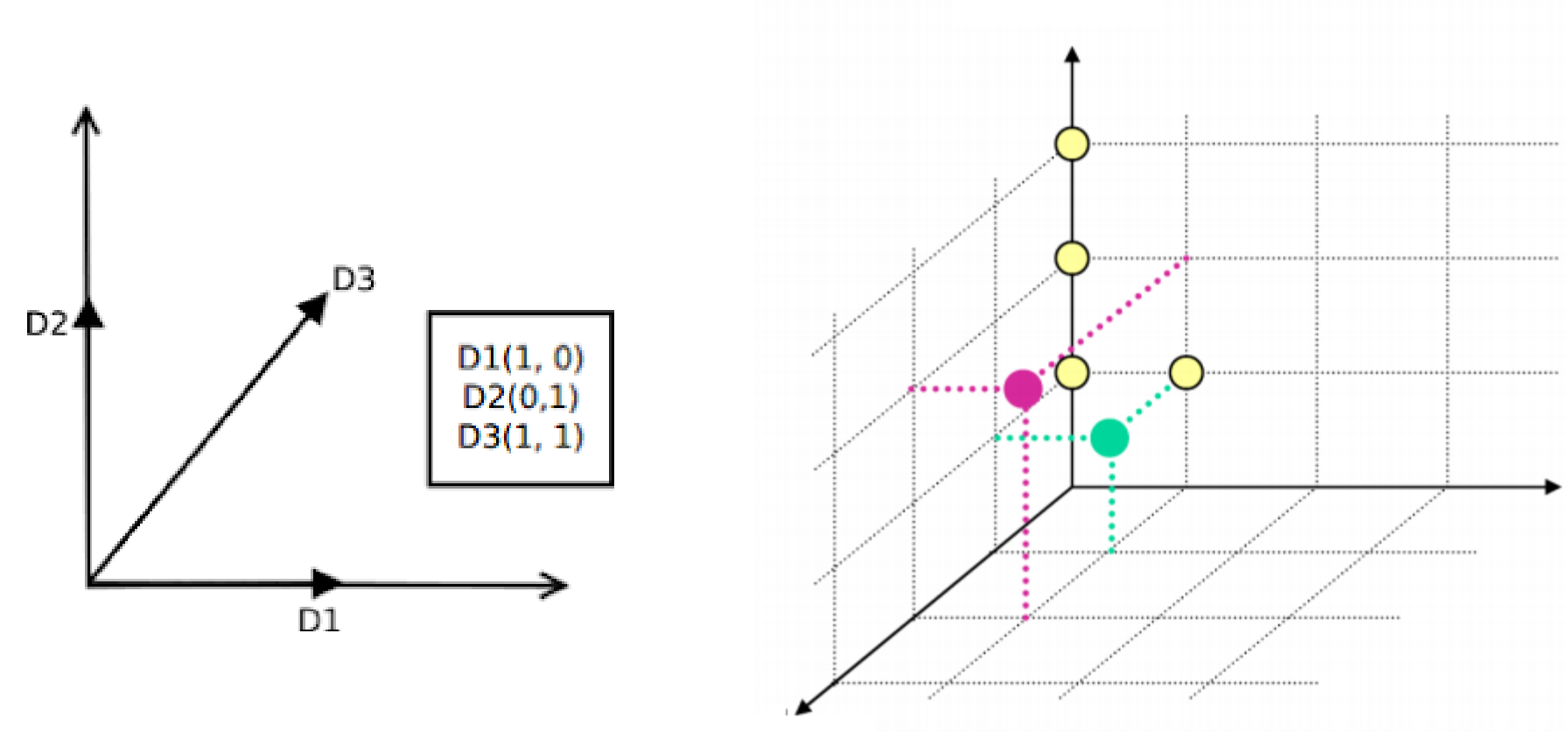
- Thuật toán phân cụm K-means thuộc loại học không giám sát
 - không biết nhãn (label) của từng điểm dữ liệu
 - Mục đích là làm thế nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho *dữ liệu trong cùng một cụm có tính chất giống nhau*.
- **Ví dụ:** Một công ty muốn tạo ra những chính sách ưu đãi cho những nhóm khách hàng khác nhau nhưng chưa có cách phân thành một số nhóm/cụm:
 - phân nhóm dựa trên sự tương tác giữa mỗi khách hàng với công ty: số năm là khách hàng; số tiền khách hàng đã chi trả cho công ty; độ tuổi; giới tính; thành phố; nghề nghiệp;
 - giả sử công ty có nhiều dữ liệu của rất nhiều khách hàng

Giới thiệu

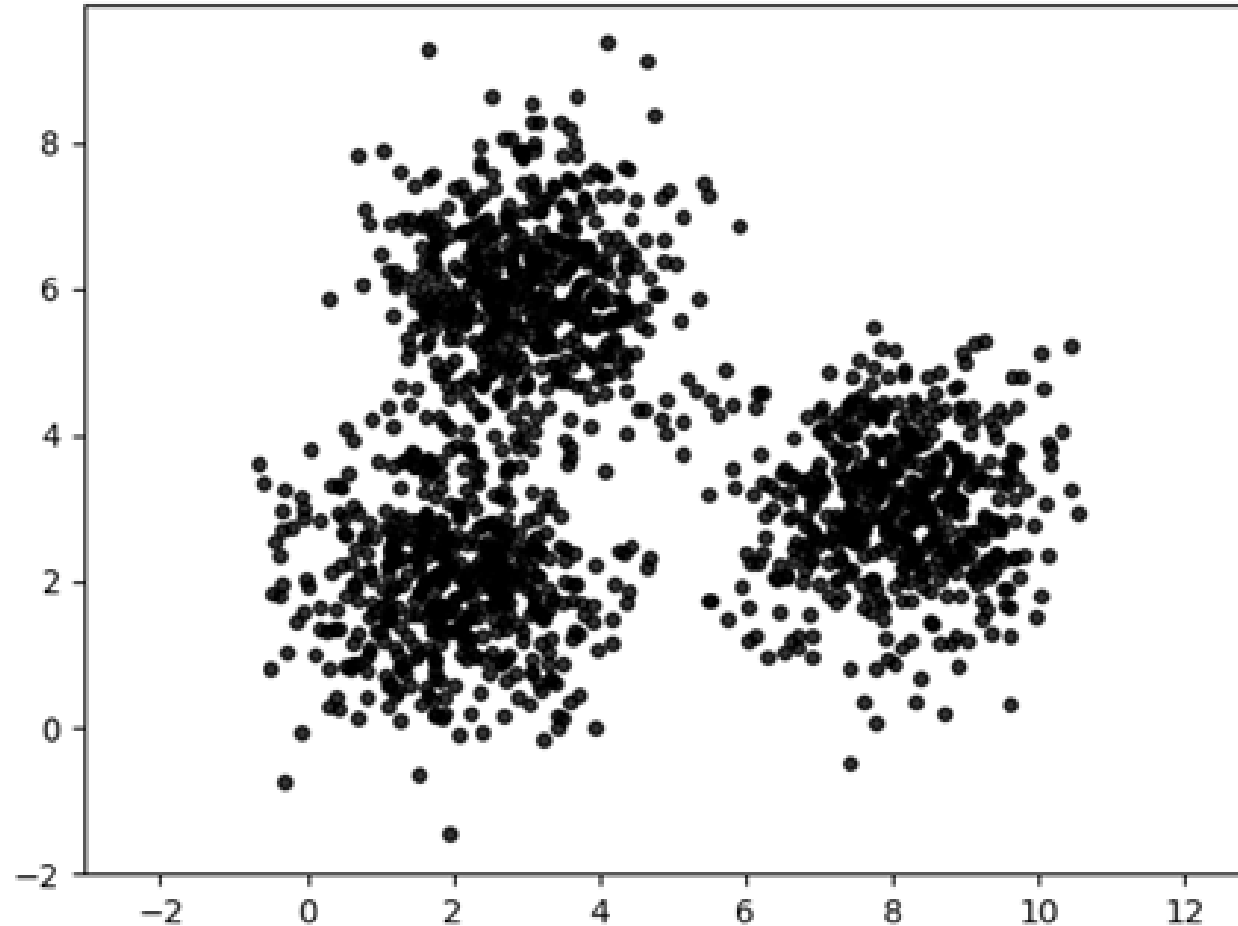
- phương pháp đơn giản là K-means Clustering sẽ được sử dụng để phân thành các cụm
- Sau khi đã có các nhóm, nhân viên công ty lựa chọn ra một vài khách hàng trong mỗi nhóm để quyết định xem mỗi nhóm tương ứng với nhóm khách hàng nào.
- Ý tưởng đơn giản nhất về cluster là tập hợp các điểm ở gần nhau trong một không gian nào đó

Giới thiệu

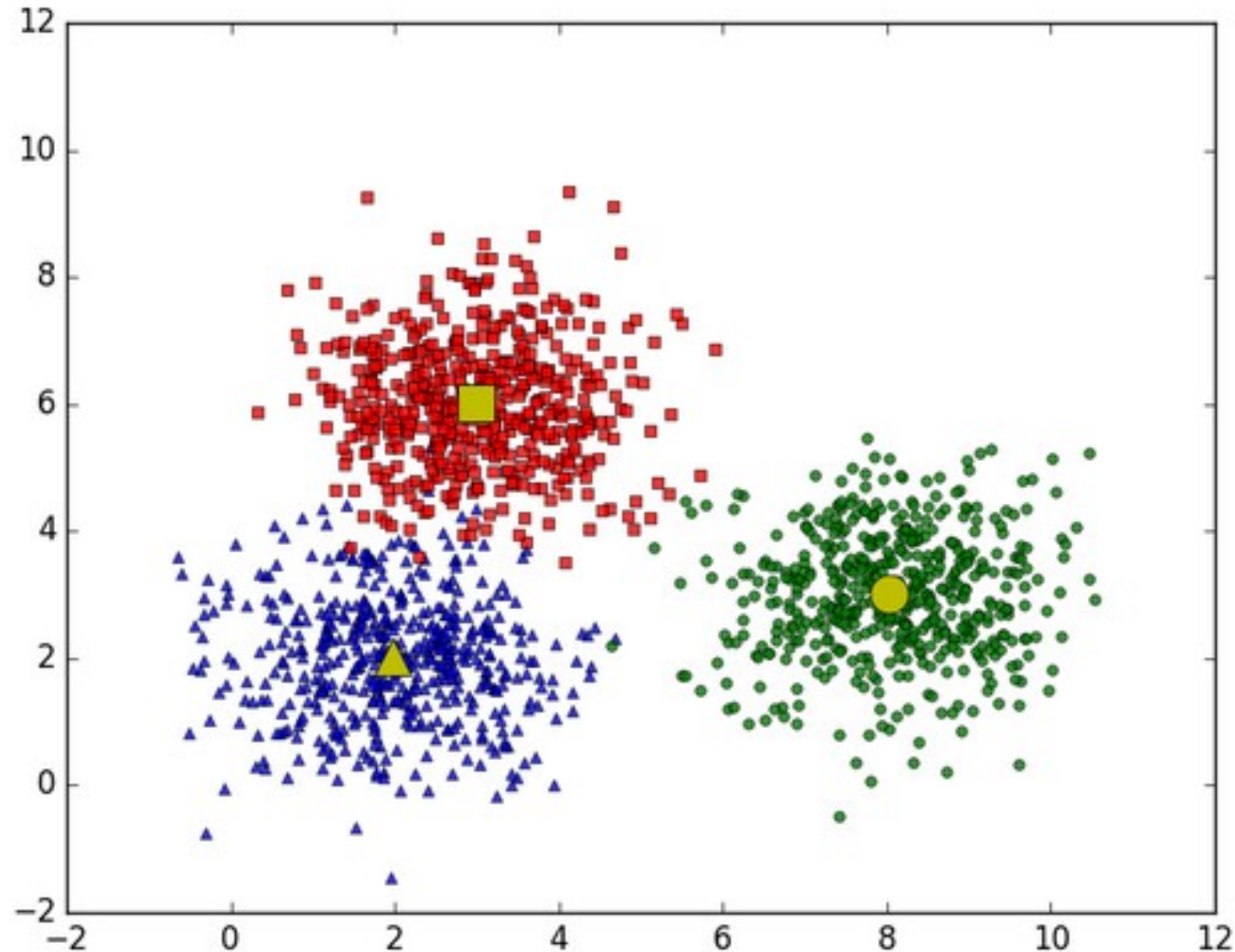
- Điểm trong không gian véc tơ



Giới thiệu



Giới thiệu



Giới thiệu

Bài toán phân nhóm:

Input:

- dữ liệu đầu vào
- số lượng nhóm muốn tìm

Output:

- chỉ ra center của mỗi nhóm
- phân các điểm dữ liệu vào các nhóm

Một số ký hiệu toán học

- Có N điểm dữ liệu $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{d \times N}$ và K là số cụm cần phân chia.
- Chúng ta cần tìm tâm của mỗi cụm $m_1, m_2, \dots, m_K \in \mathbb{R}^{d \times 1}$ và nhãn của mỗi điểm dữ liệu.
- Với mỗi x_i , đặt $y_i = [y_{i1}, y_{i2}, \dots, y_{iK}]$ là véc tơ nhãn
- Nếu x_i được phân vào nhóm k thì $y_{ik} = 1$ và $y_{ij} = 0, \forall j \neq k$
- Ví dụ:
 - Một điểm dữ liệu có véc tơ nhãn là $[1, 0, \dots, 0]$ thì điểm đó thuộc về lớp 1

Một số ký hiệu toán học

- Ràng buộc của y_i có thể viết dưới dạng toán học như sau:

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

Hàm mất mát và bài toán tối ưu

- G/S m_k là center (hoặc representative) của mỗi cluster
 - ước lượng tất cả các điểm được phân vào cluster này bởi m_k
 - một điểm dữ liệu x_i được phân vào cluster k sẽ có sai số là $(x_i - m_k)$
- Chúng ta mong muốn sai số $(x_i - m_k)$ có trị tuyệt đối nhỏ nhất nên đại lượng dưới đây cần có giá trị nhỏ nhất:

$$\|x_i - m_k\|_2^2$$

- Vì x_i được phân vào cụm k nên $y_{ik} = 1, y_{ij} = 0, \forall j \neq k$, do đó chúng ta có:

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

Hàm mất mát và bài toán tối ưu

- Sai số cho toàn bộ dữ liệu sẽ là:

$$\mathcal{L}(Y, M) = \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|x_i - m_j\|_2^2$$

- Trong đó
 - $Y = [y_1; y_2; \dots; y_N]$ là ma trận véc tơ nhãn
 - $M = [m_1, m_2, \dots, m_K]$ là ma trận tâm của mỗi cụm

Hàm mất mát và bài toán tối ưu

- Chúng ta cần tối ưu bài toán để tìm \mathbf{Y} và \mathbf{M}

$$\mathbf{Y}, \mathbf{M} = \arg \min_{\mathbf{Y}, \mathbf{M}} \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$
$$\text{subject to: } y_{ij} \in \{0, 1\} \quad \forall i, j; \quad \sum_{j=1}^K y_{ij} = 1 \quad \forall i \quad (*)$$

- Bài toán (*) là một bài toán khó tìm *điểm tối ưu* vì nó có thêm các điều kiện ràng buộc.
- Bài toán này thuộc loại rất khó tìm nghiệm tối ưu toàn cục (global optimal point, tức nghiệm làm cho hàm mất mát đạt giá trị nhỏ nhất có thể).*

Thuật toán tối ưu hàm mất mát

- Tuy nhiên, trong một số trường hợp chúng ta vẫn có thể tìm được phương pháp để tìm được nghiệm gần đúng hoặc điểm cực tiểu.
 - Lưu ý: *điểm cực tiểu chưa chắc đã phải là điểm làm cho hàm số đạt giá trị nhỏ nhất*
- Giải bài toán (*) là:
 - xen kẽ giải Y và M khi biến còn lại được cố định
 - đây là thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu
- Sau khi giải, chúng ta có được Y và M

Tóm tắt thuật toán

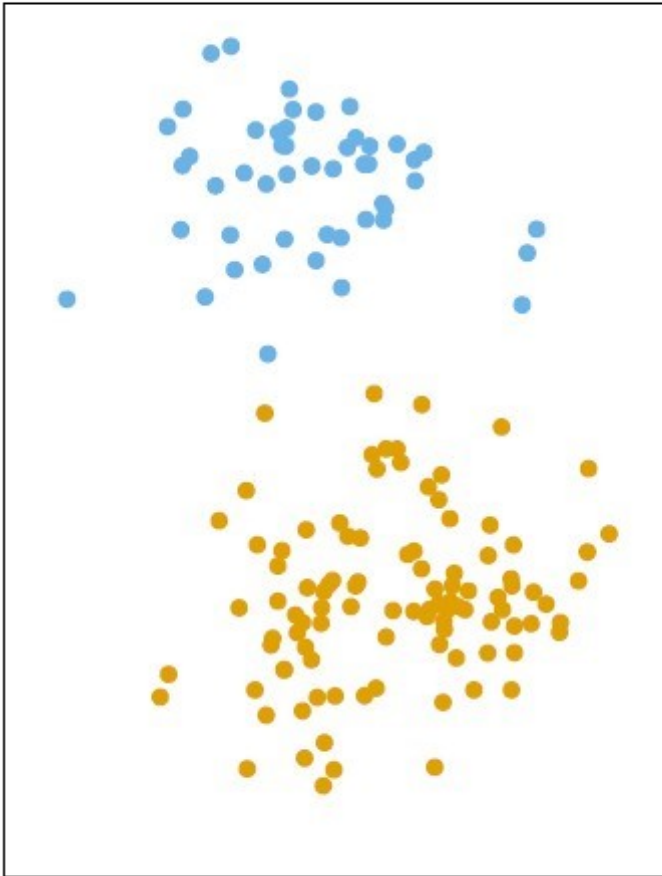
Đầu vào: Dữ liệu X và số lượng cluster cần tìm K

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y

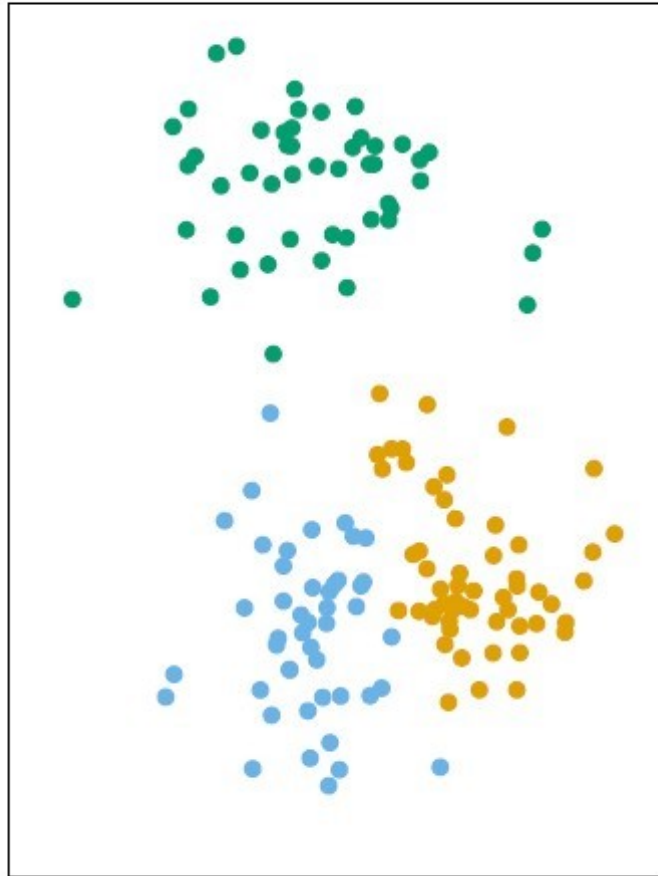
1. Chọn K điểm bất kỳ làm các center ban đầu
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì thuật toán dừng.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

Minh họa

K=2



K=3



K=4

