Phân tích dữ liệu thông minh

Gom nhóm dữ liệu (clustering)

Kỹ thuật K-means

TS. Nguyễn Tiến Huy

ntienhuy@fit.hcmus.edu.vn

Nội dung

1. Một số điểm chính ở bài trước
2. Giới thiệu gom nhóm
3. Ứng dụng trong nén ảnh
4. Ứng dụng trong phân chủ đề cho văn bản

Một số điểm chính ở bài trước

- Học có giám sát

- SVM, Decision Tree

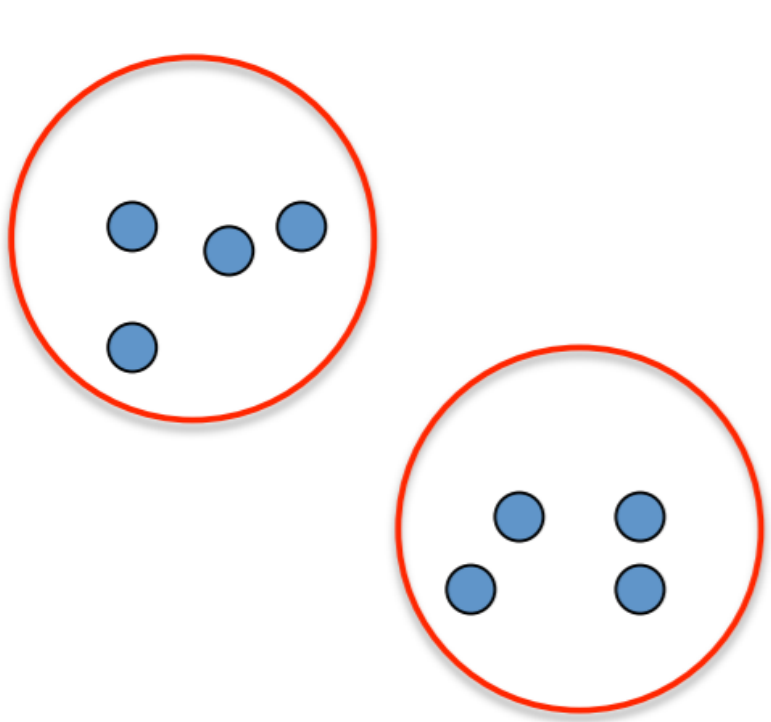
Gom nhóm dữ liệu (clustering)

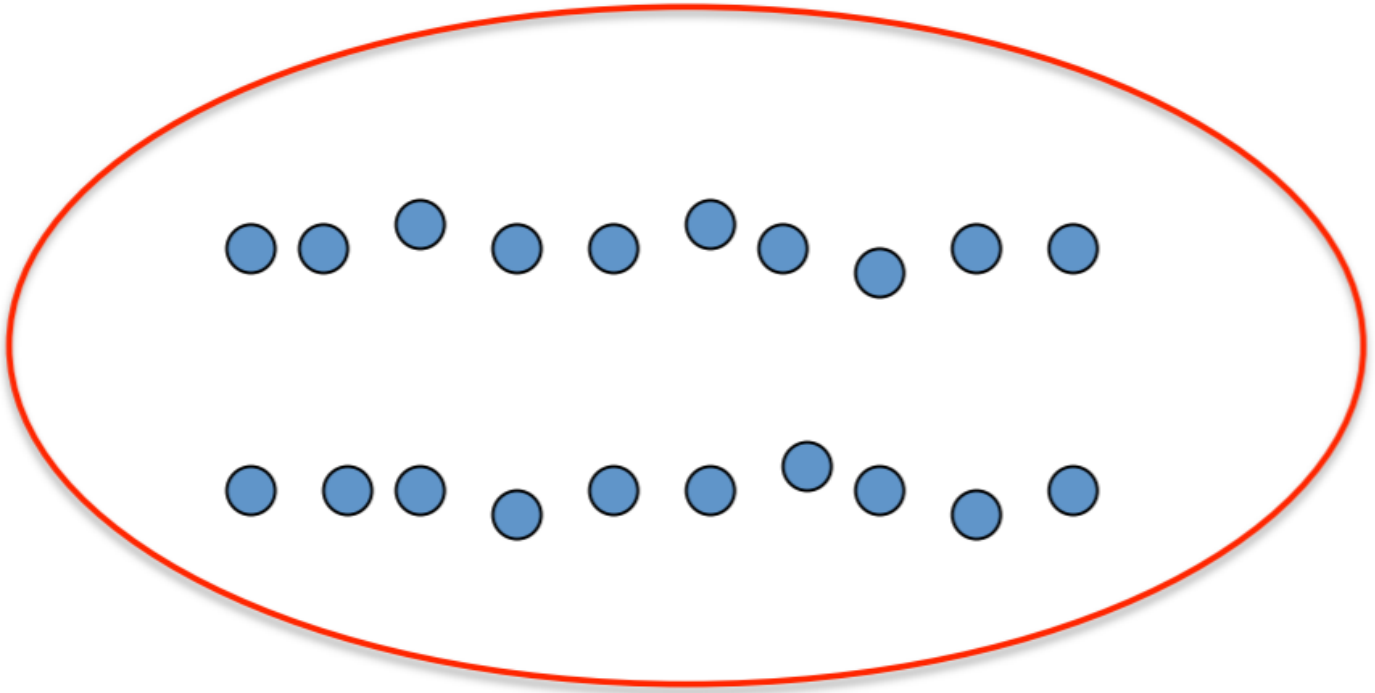
- Clustering là một phương pháp học không giám sát (unsupervised learning)

- Ứng dụng: gom nhóm dữ liệu, email, văn bản, khách hàng,...

Gom nhóm dữ liệu (clustering)

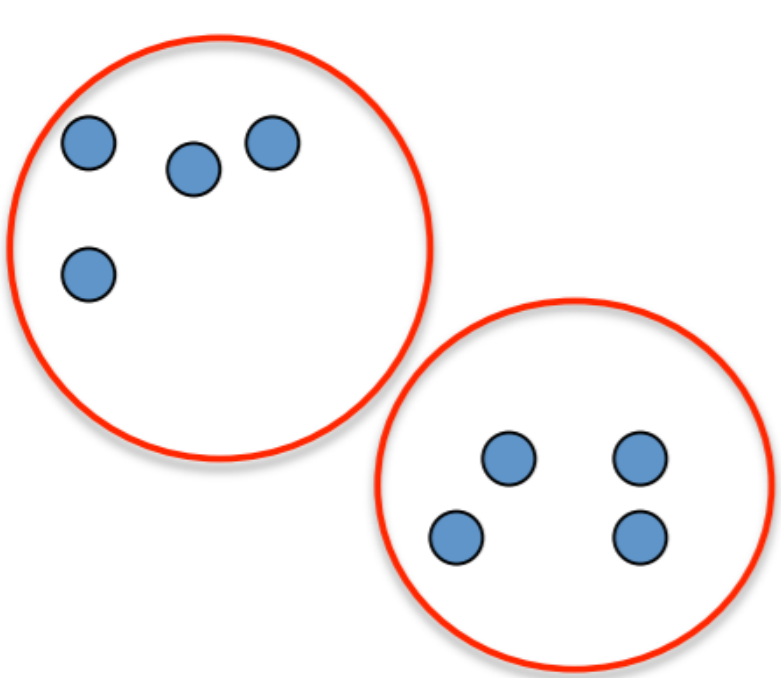
- Ý tưởng: gom nhóm những mẫu dữ liệu giống nhau.

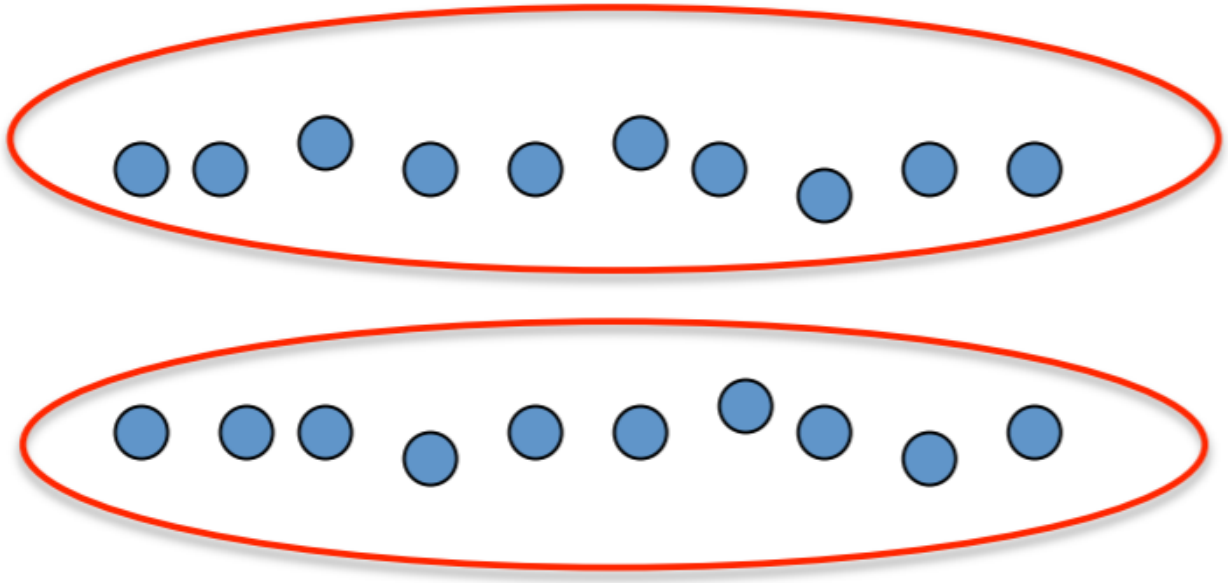




Gom nhóm dữ liệu (clustering)

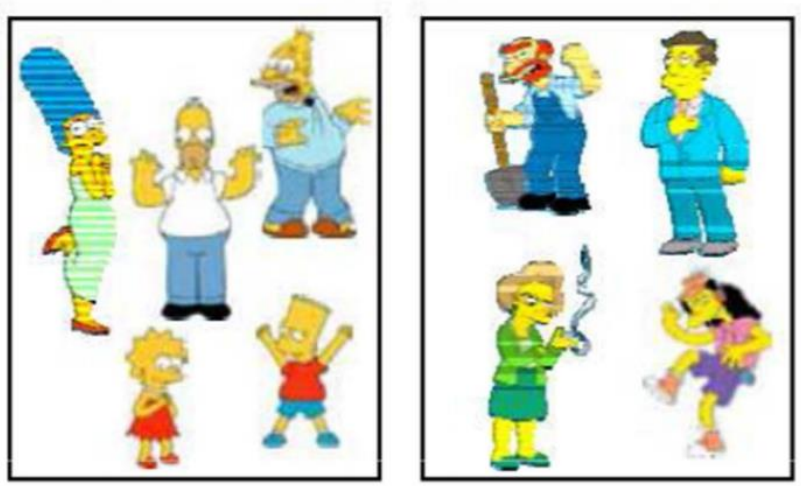
- Định nghĩa thế nào là giống nhau?
  - Phổ biến: khoảng cách Euclidean.



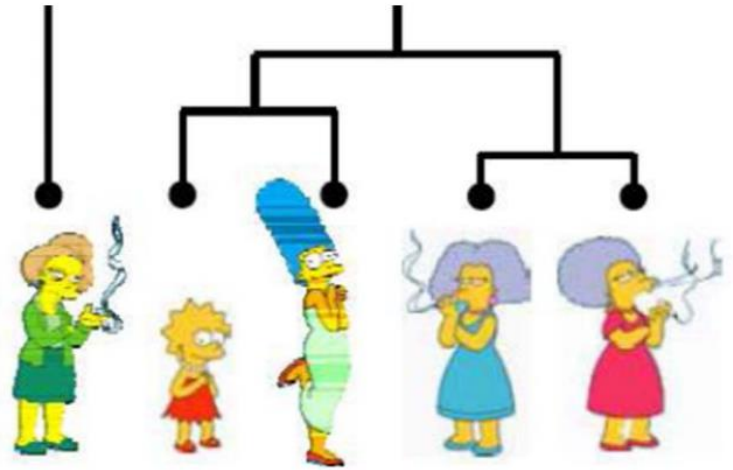


Thuật toán gom nhóm

- Thuật toán chia tách (partition algorithms flat)
  - K-mean
  - Spectral clustering

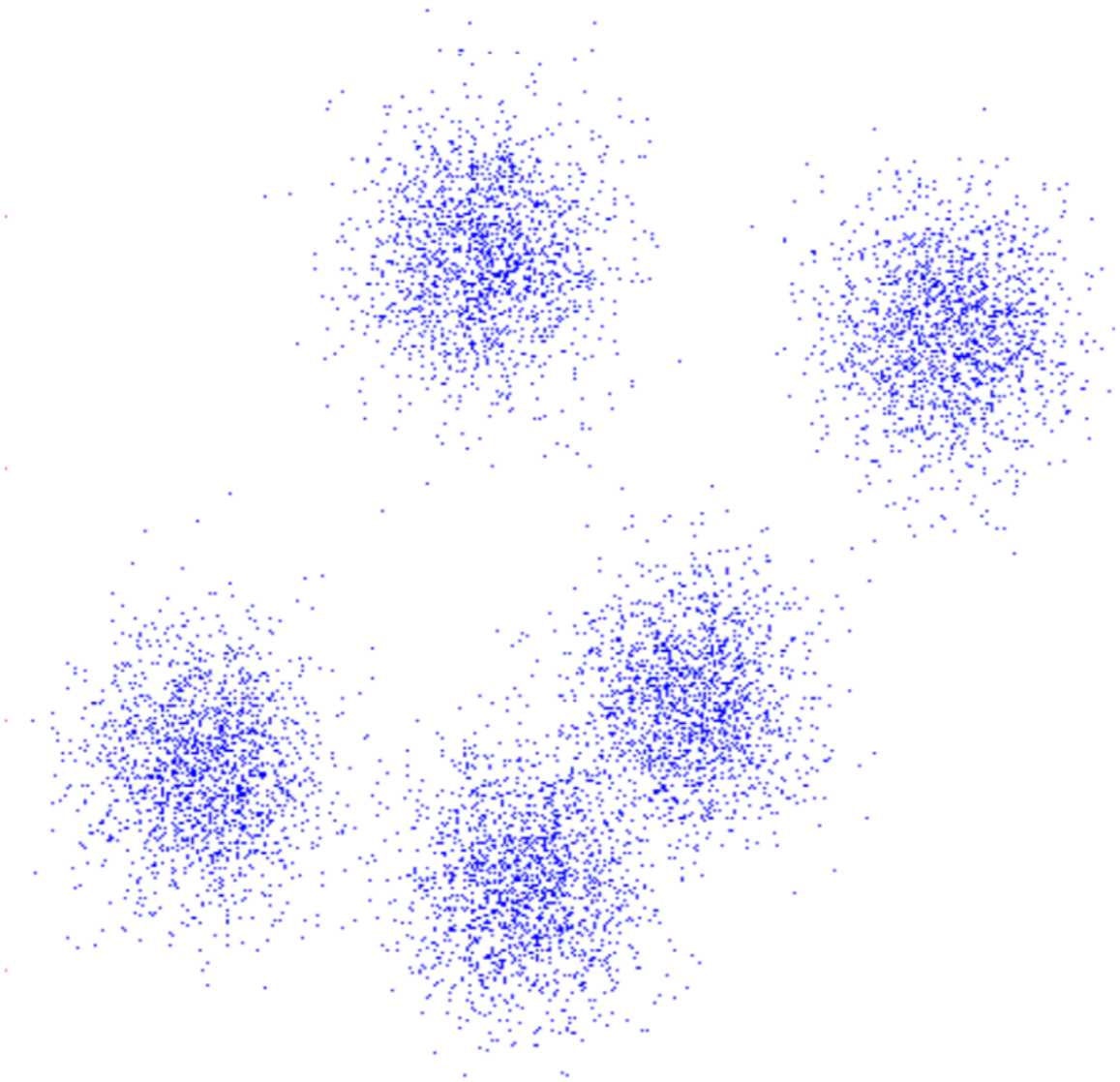


- Thuật toán phân tần
  - Bottom up – agglomerative
  - Top down – divisive



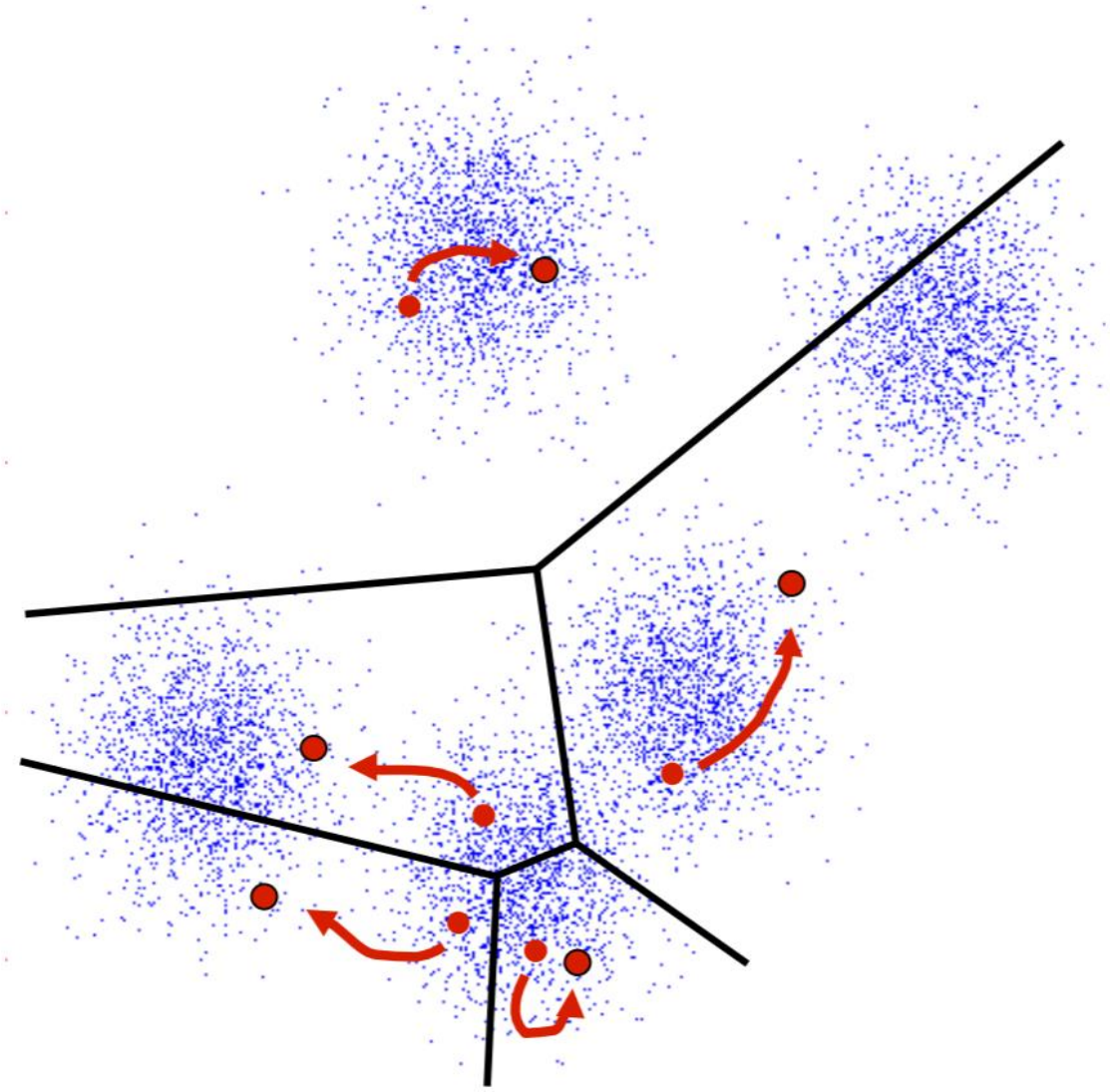
Thuật toán góm nhóm K-means

- **Khởi tạo**: chọn ngẫu nhiên K điểm làm tâm cho K nhóm (cluster)
- **Cập nhật**:
  - Gán các điểm dữ liệu vào nhóm gần nhất.
  - Tính lại tâm của nhóm bằng cách lấy trung bình của các điểm hiện tại trong nhóm.
- *Tiếp tục lặp lại bước cập nhật đến khi không còn thay đổi.*



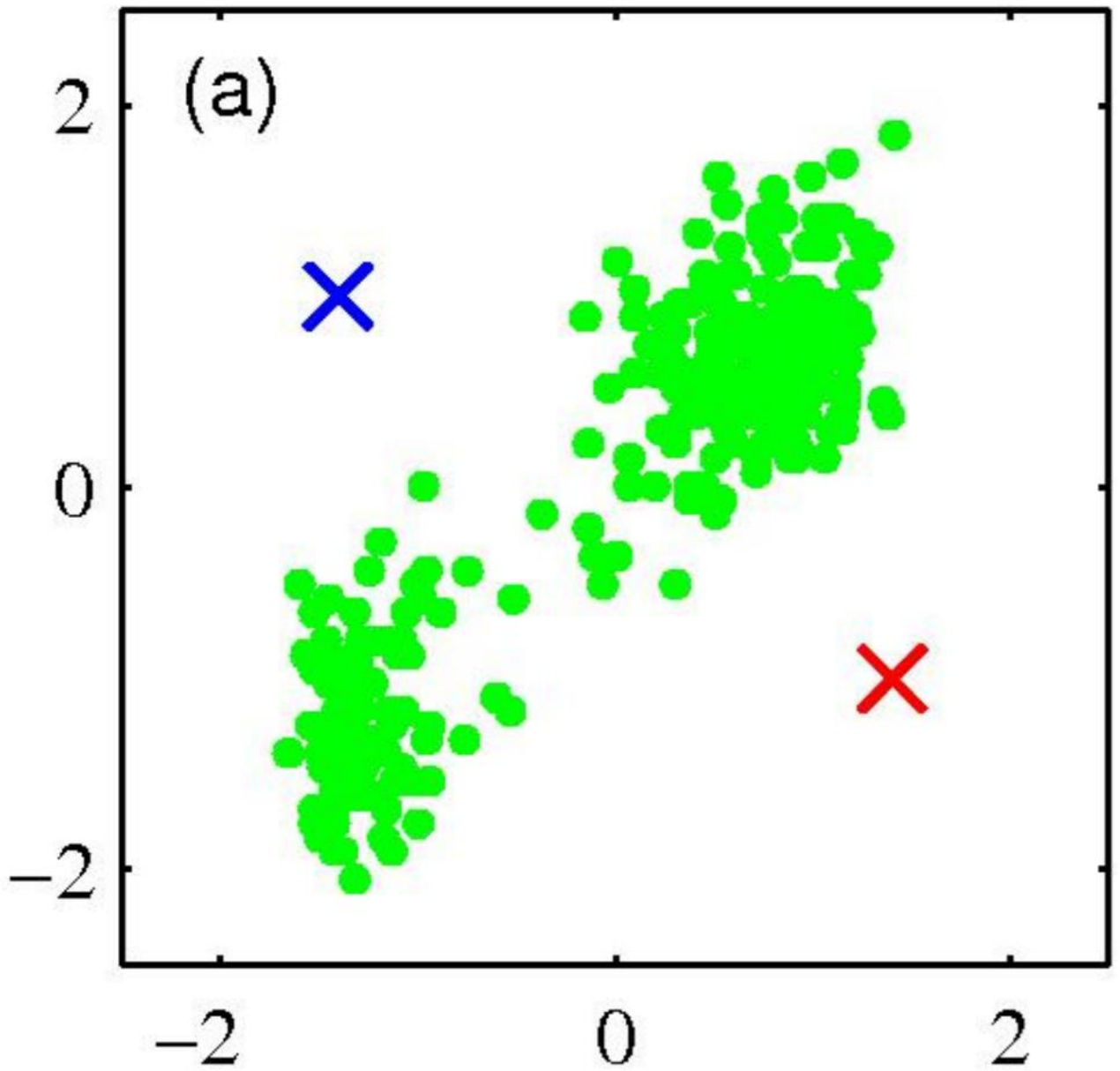
Thuật toán góm nhóm K-means

- **Khởi tạo**: chọn ngẫu nhiên K điểm làm tâm cho K nhóm (cluster)
- **Cập nhật**:
  - Gán các điểm dữ liệu vào nhóm gần nhất.
  - Tính lại tâm của nhóm bằng cách lấy trung bình của các điểm hiện tại trong nhóm.
- *Tiếp tục lặp lại bước cập nhật đến khi không còn thay đổi.*



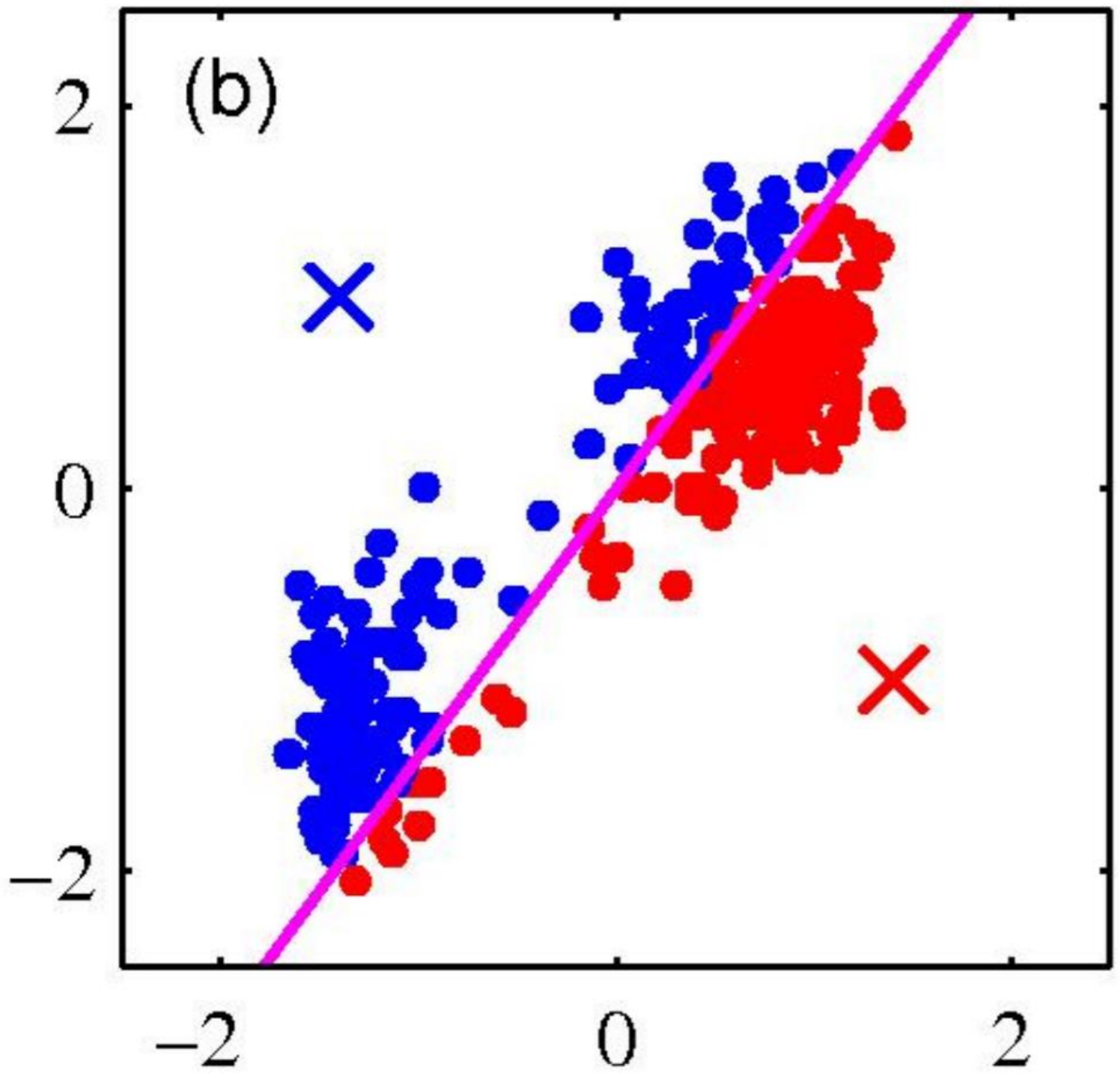
Ví dụ K-means

- **Khởi tạo**: chọn ngẫu nhiên 2 điểm làm tâm (K =2)



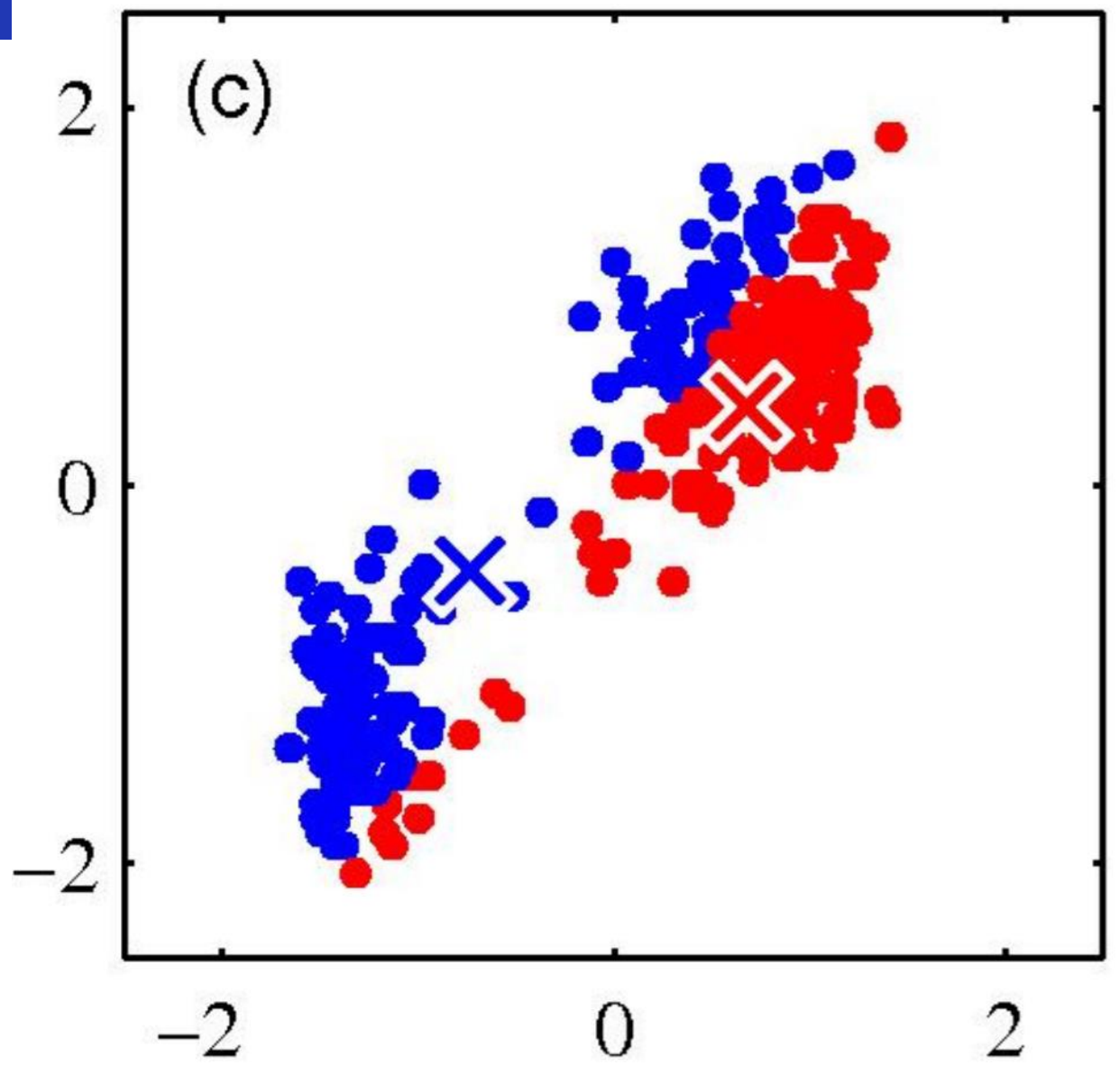
Ví dụ K-means

- **Vòng lặp**: gán các điểm vào nhóm tương ứng.



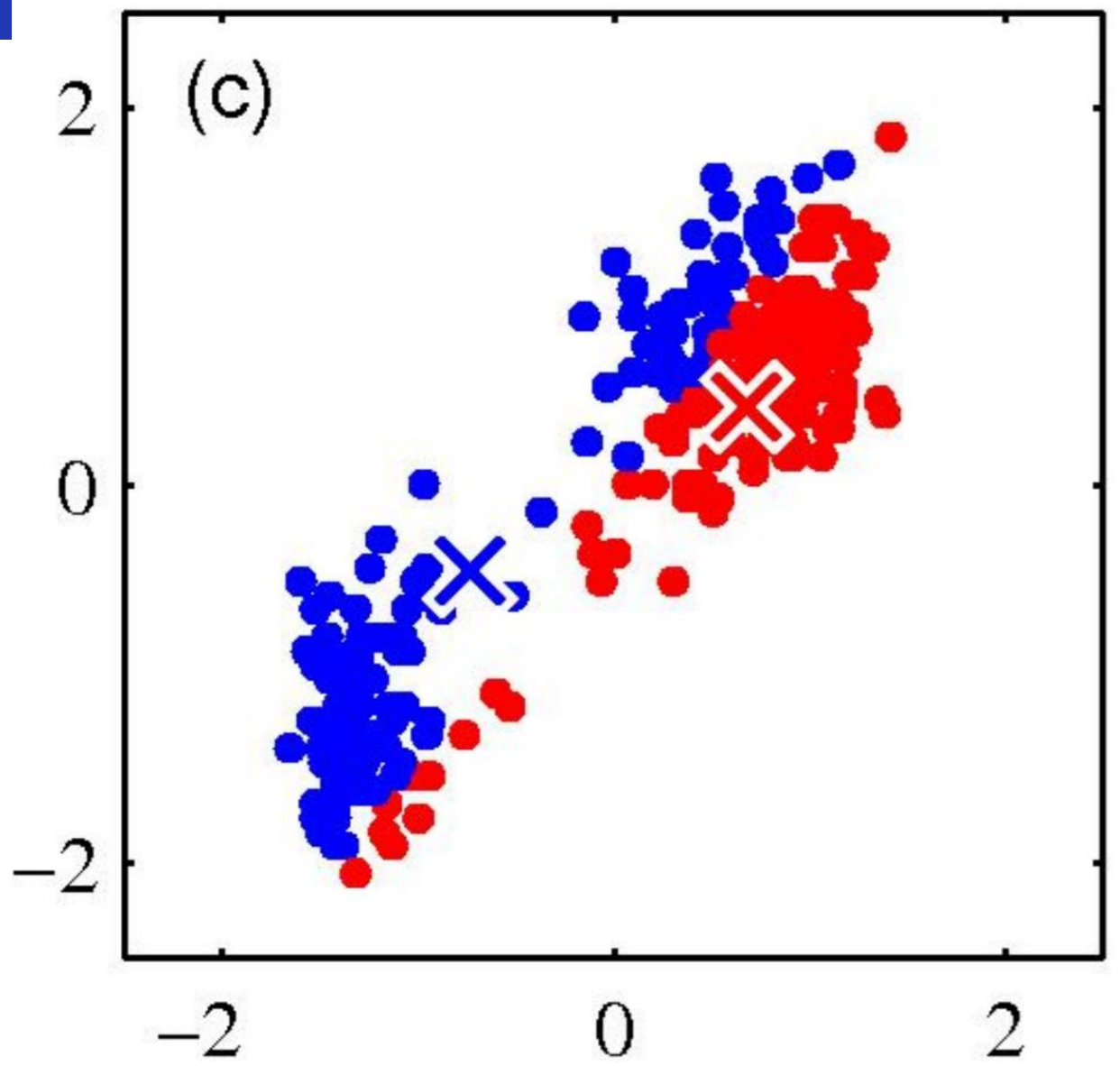
Ví dụ K-means

- **Vòng lặp**: cập nhật lại tâm mới cho các nhóm.



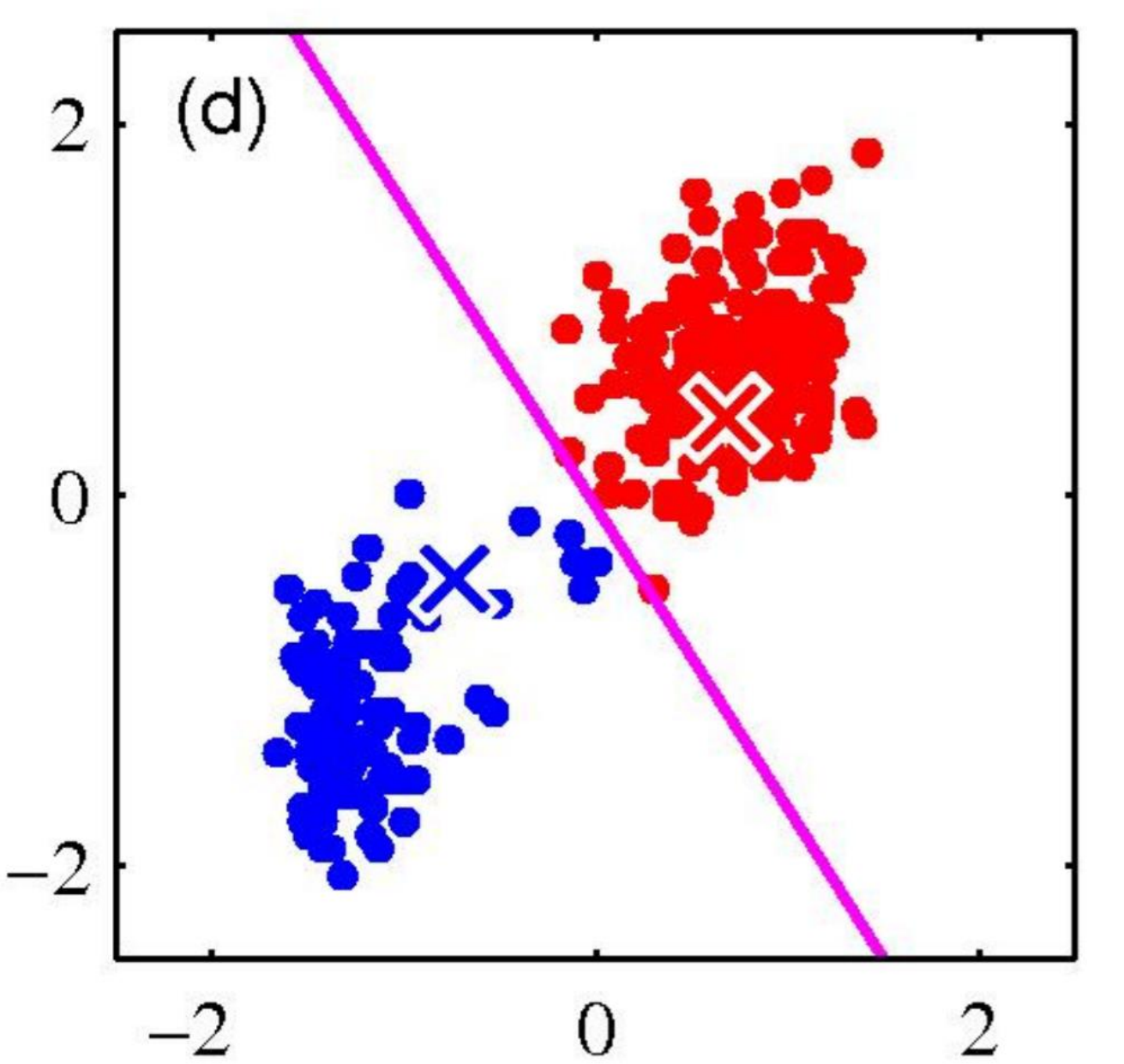
Ví dụ K-means

- **Vòng lặp**: cập nhật lại tâm mới cho các nhóm.



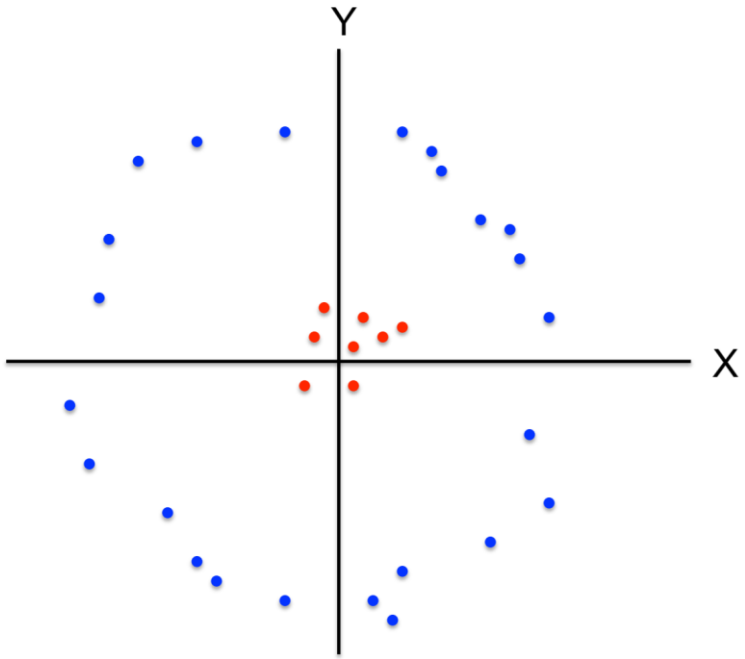
Ví dụ K-means

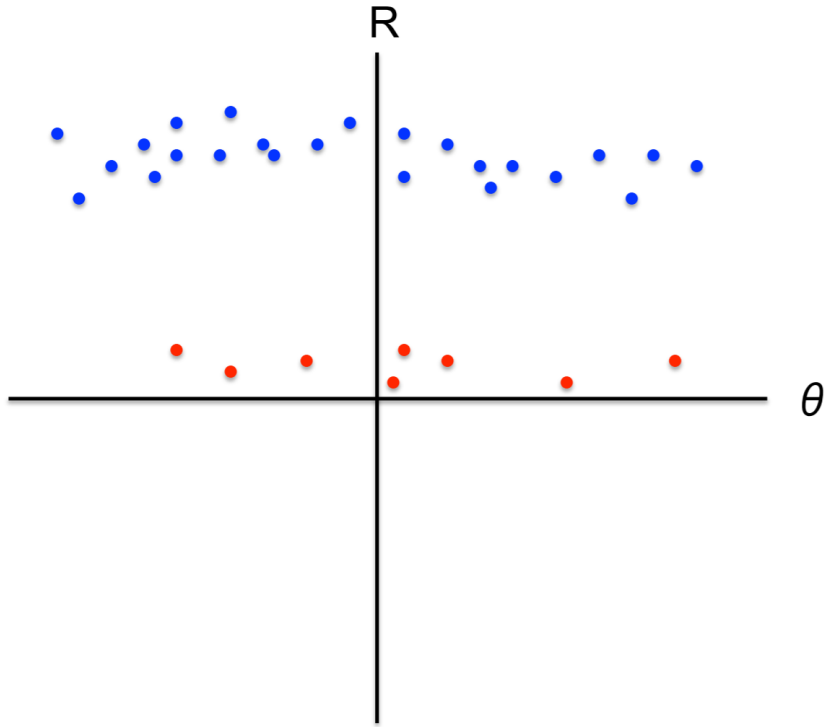
- **Lặp lại 2 bước trên đến khi hội tụ** (không còn thay đổi)



Ưu nhược điểm của K-means

- Bao nhiêu K (nhóm) là hợp lý?
- Phụ thuộc vào việc khởi tạo các tâm.
- Vì các nhóm của K-means có hình tròn, nên không phải lúc nào cũng phù hợp

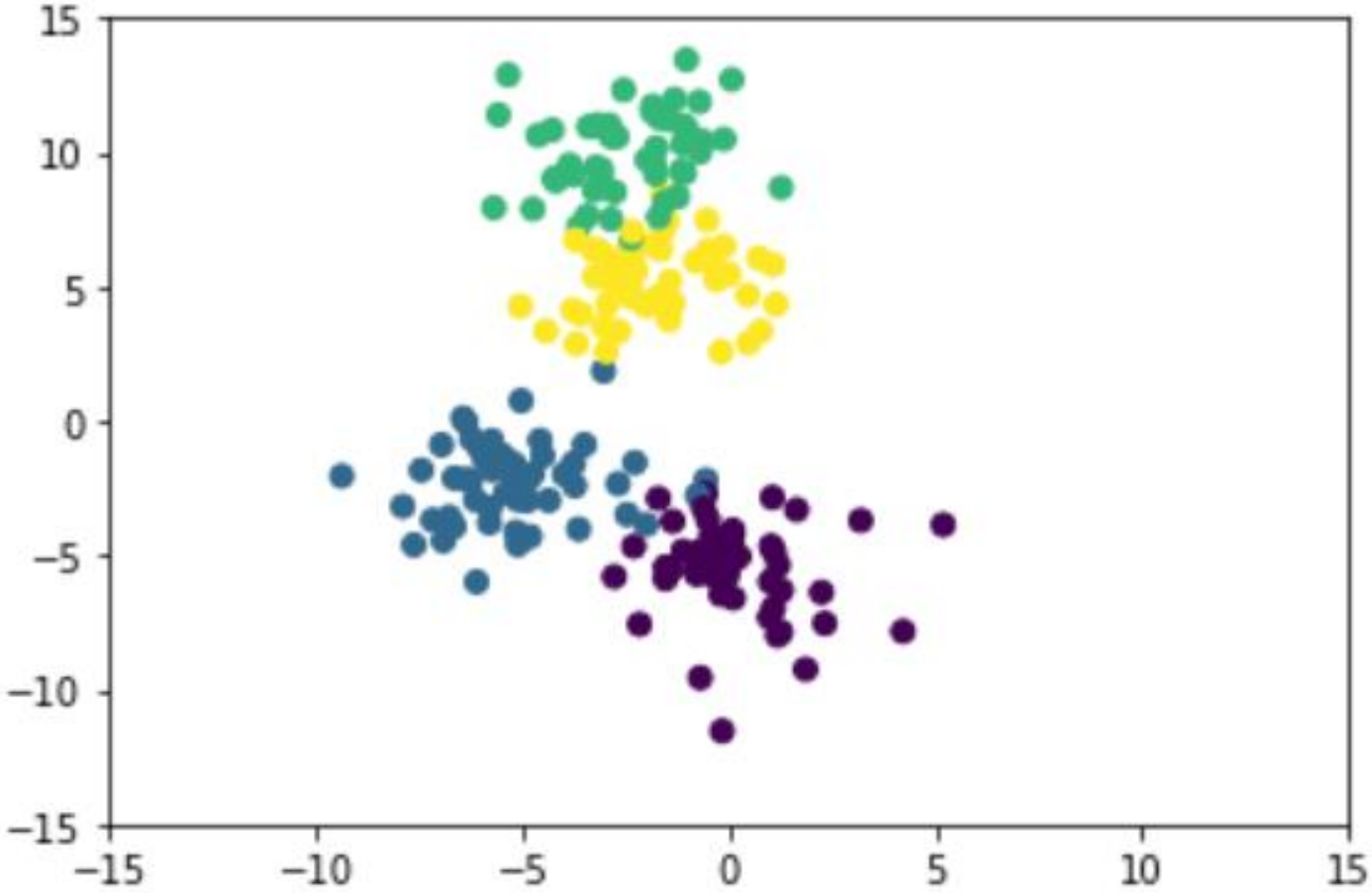




Thực hành với dữ liệu của sklearn.make_blobs

```
# import statements
from sklearn.datasets import make_blobs
import numpy as np
import matplotlib.pyplot as plt
# create blobs
data = make_blobs(n_samples=200, n_features=2,
                  centers=4, cluster_std=1.6, random_state=50)
# create np array for data points
points = data[0]
# create scatter plot
plt.scatter(data[0][:,0], data[0][:,1], c=data[1])
plt.xlim(-15,15)
plt.ylim(-15,15)
```

Thực hành với dữ liệu của sklearn.make_blobs



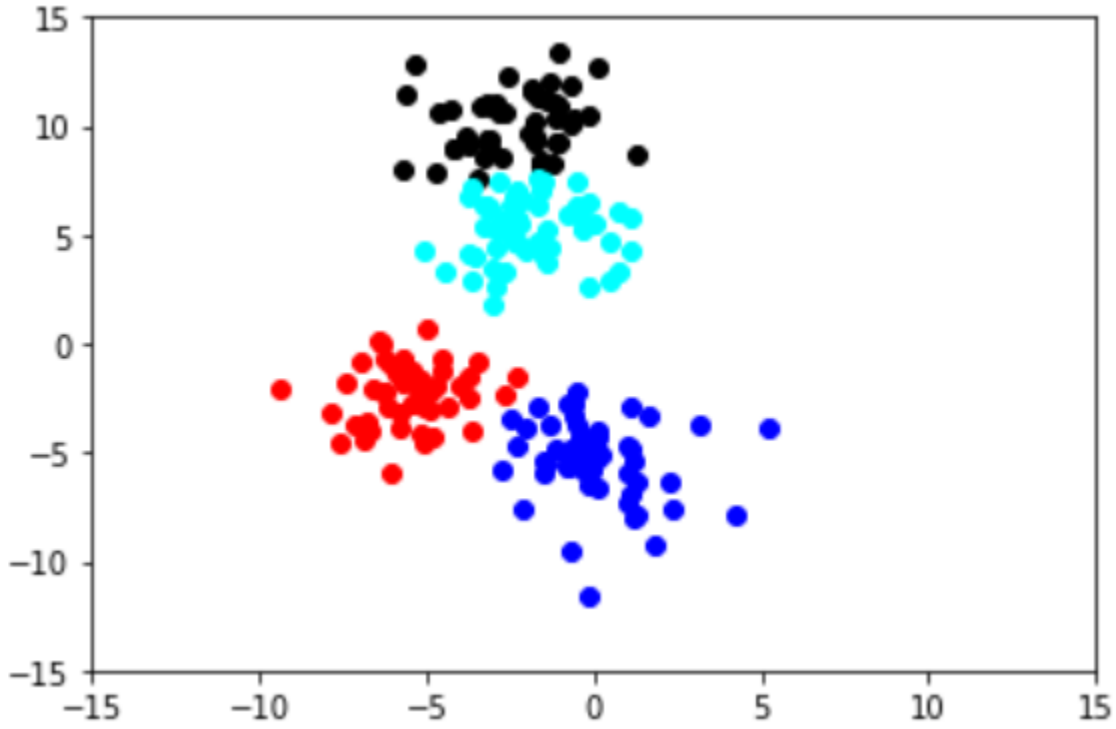
Thực hành với dữ liệu của sklearn.make_blobs

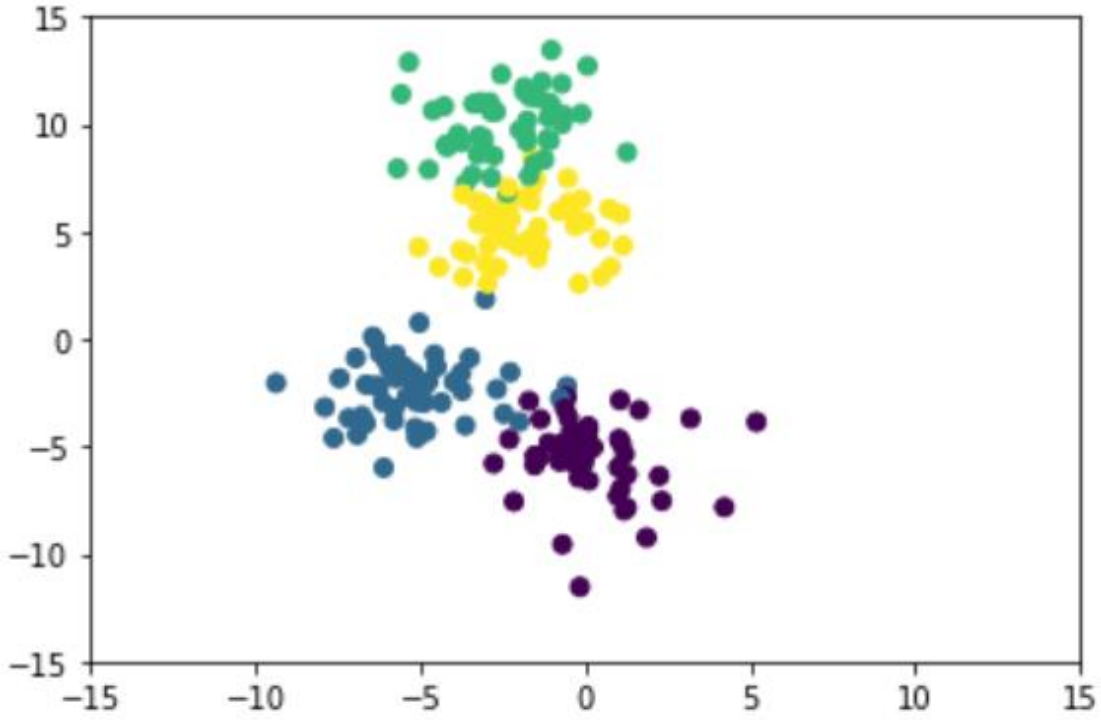
```
# import KMeans
from sklearn.cluster import KMeans

# create kmeans object
kmeans = KMeans(n_clusters=4)
# fit kmeans object to data
kmeans.fit(points)
# print location of clusters learned by kmeans object
print(kmeans.cluster_centers_)
# save new clusters for chart
y_km = kmeans.fit_predict(points)

plt.scatter(points[y_km ==0,0], points[y_km == 0,1], c='red')
plt.scatter(points[y_km ==1,0], points[y_km == 1,1], c='black')
plt.scatter(points[y_km ==2,0], points[y_km == 2,1], c='blue')
plt.scatter(points[y_km ==3,0], points[y_km == 3,1], c='cyan')
plt.xlim(-15,15)
plt.ylim(-15,15)
```

Thực hành với dữ liệu của sklearn.make_blobs





Thảo luận

- Giả sử số lượng bệnh nhân nghi nhiễm ncovid-19 quá lớn, không thể xét nghiệm hết các bệnh nhân trong thời gian ngắn.

=> Hãy đề xuất sử dụng AI giúp hỗ trợ xác định người có khả năng nhiễm cao dựa vào triệu chứng, thông tin dịch tễ (đã đi qua những đâu, tiếp xúc ai v.v....)

Bài tập - Nén ảnh sử dụng K-means:

- Ý tưởng:
  - Gom nhóm các màu giống nhau lại thành 1 nhóm, và sử dụng 1 màu chung cho nhóm đó => giảm số lượng màu phải dung.
- Link ảnh: https://drive.google.com/file/d/16WnYBtjsFGVyA7qfdkxaO-on2cQQ-gWF/view?usp=sharing

Ảnh gốc

Ảnh nén

Nén ảnh

```
from skimage import io
from sklearn.cluster import KMeans
import numpy as np

image = io.imread('tiger.png')
io.imshow(image)
io.show()

rows = image.shape[0]
cols = image.shape[1]

image = image.reshape(image.shape[0]*image.shape[1],4)
kmeans = KMeans(n_clusters = 128, n_init=10, max_iter=200)
kmeans.fit(image)

clusters = np.asarray(kmeans.cluster_centers_,dtype=np.uint8)
labels = np.asarray(kmeans.labels_,dtype=np.uint8 )
labels = labels.reshape(rows,cols);

np.save('codebook_tiger.npy',clusters)
io.imsave('compressed_tiger.png',labels);
```

Giải nén ảnh

```
from skimage import io
import numpy as np

centers = np.load('codebook_tiger.npy')
c_image = io.imread('compressed_tiger.png')

image = np.zeros((c_image.shape[0],c_image.shape[1],4),dtype=np.uint8 )
for i in range(c_image.shape[0]):
    for j in range(c_image.shape[1]):
            image[i,j,:] = centers[c_image[i,j],:]
io.imsave('reconstructed_tiger.png',image);
io.imshow(image)
io.show()
```

Bài tập – Gom nhóm văn bản

- Link: https://colab.research.google.com/drive/1geh5illX5RQ2jHIXwzsv-jKsbpoffQ80?usp=sharing

Cảm ơn đã theo dõi!