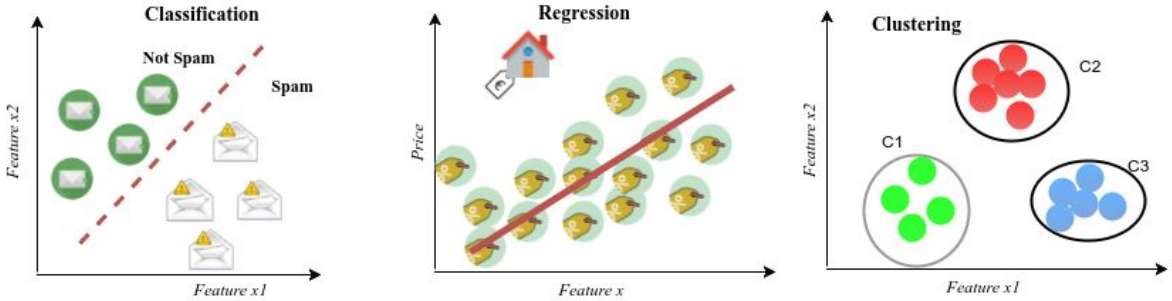
Chương 3
K-means

TS. Phạm Tuấn

ptuan@ute.udn.vn

Giới thiệu

K-means là một thuật toán dùng để phân dữ liệu thành các cụm và nó thuộc loại **học không giám sát** (tức là dữ liệu không có nhãn) và được sử dụng để giải quyết **bài toán phân cụm**.

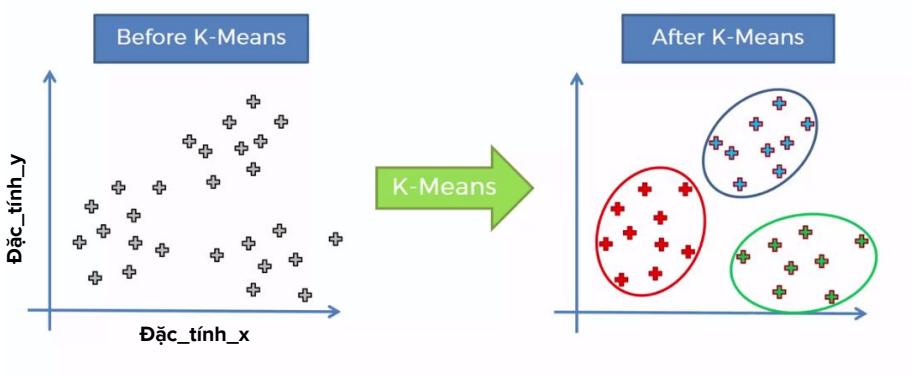


Giới thiệu

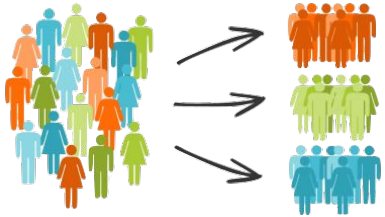
Phân cụm hoạt động dựa trên đặc tính như sau:

Các điểm nằm gần nhau thì có những đặc tính hoặc tính chất giống nhau.

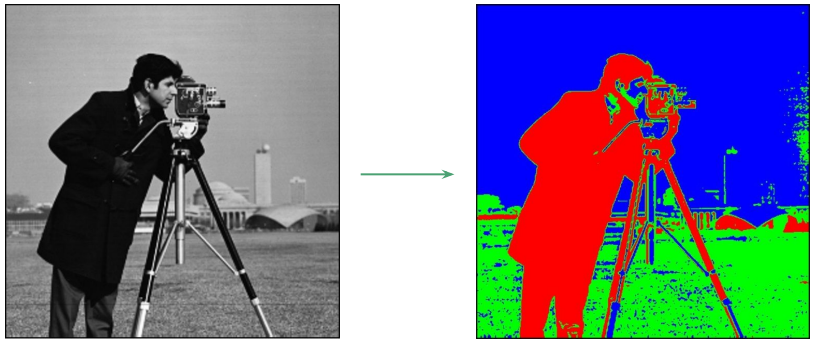
Do đó, những điểm trong cùng 1 cụm thì có cùng 1 số tính chất nhất định.



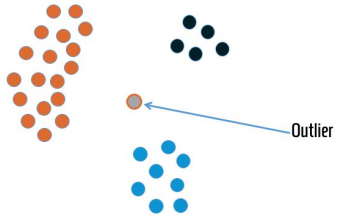
Ứng dụng



Phân nhóm khách hàng



Phân tách chủ thể



Phát hiện nhiễu

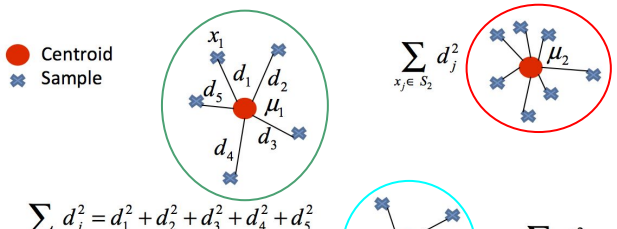
Các khái niệm trong K-means

K: Số cụm

Centroid: Điểm trung tâm của cụm

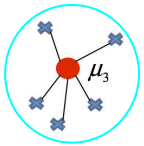
Khoảng cách: Euclidean distance

Chú ý: **Số cụm và số điểm trung tâm là bằng nhau**.



$$\sum_{x_j \in S_1} d_j^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2$$

$$\sum_{x_j \in S_2} d_j^2$$



$$\sum_{x_j \in S_3} d_j^2$$

$$\min_S E(\mu_i) = \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_2} d_j^2 + \sum_{x_j \in S_3} d_j^2$$

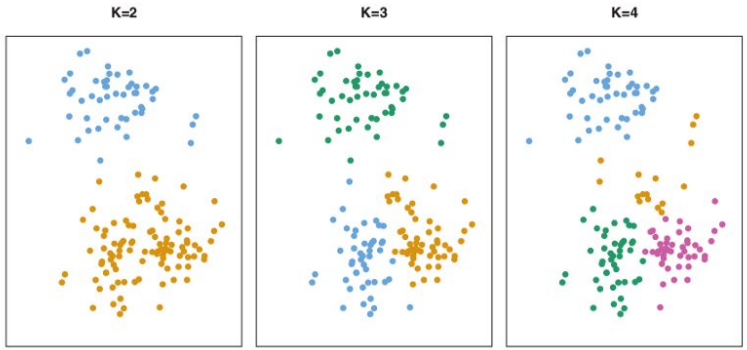
Số cụm (K)

Tùy thuộc vào bài toán thì **số cụm** đã được **xác định trước đó**, hoặc **chưa được xác định**.

Nếu chưa xác định trước, chúng ta có thể **dùng phương pháp** hoặc bằng **thực nghiệm để xác định**.

Số cụm nhỏ thì dữ liệu trong từng cụm sẽ có nhiều điểm ko tương đồng nhau (hỗn tạp, không đồng nhất).

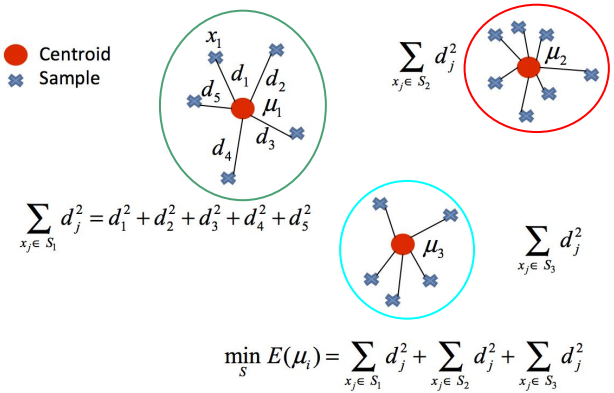
Số cụm lớn thì mô hình không khái quát hóa được dữ liệu.



Điểm trung tâm của cụm (Centroid)

Điểm trung tâm của cụm thường được gọi là Centroid hoặc center. Centroid có các đặc tính sau:

- Centroid là điểm trung tâm của cụm (Cluster), nên 2 từ đó có thể hiểu một cách tương đối là giống nhau và có thể thay thế cho nhau.
- Tọa độ của Centroid chính là trung bình của tất các các điểm thuộc Cluster đó.
- Khoảng cách từ các điểm thuộc Cluster A đến Centroid A là nhỏ nhất so với các Clusters khác.



Khoảng cách Euclidean

Công thức tính khoảng cách thương dùng có tên là Euclidean distance.

Inertia actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster.

Formula >

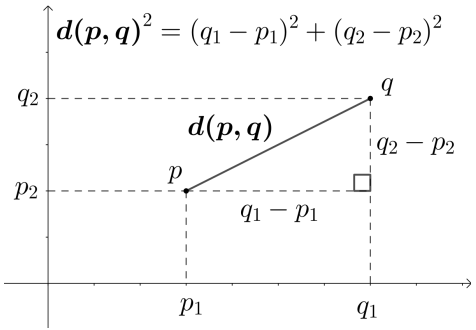
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

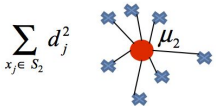
q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

Công thức



Biển diễn



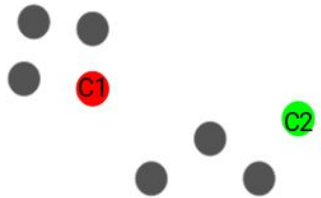
Inertia

Thuật toán

Step 1: Chọn số cluster, hay số centroid

Step 2: Select k random points from the data as centroids

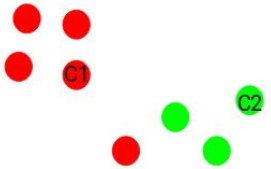
Next, we randomly select the centroid for each cluster. Let's say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:



Thuật toán

Step 3: Assign all the points to the closest cluster centroid

Once we have initialized the centroids, we assign each point to the closest cluster centroid:

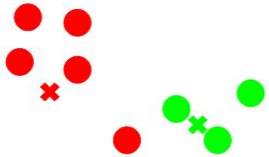


Here you can see that the points which are closer to the red point are assigned to the red cluster whereas the points which are closer to the green point are assigned to the green cluster.

Thuật toán

Step 4: Recompute the centroids of newly formed clusters

Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:

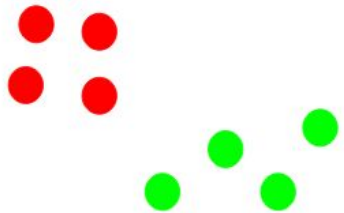


Here, the red and green crosses are the new centroids.

Thuật toán

Step 5: Repeat steps 3 and 4

We then repeat steps 3 and 4:



Pseudo code

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

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

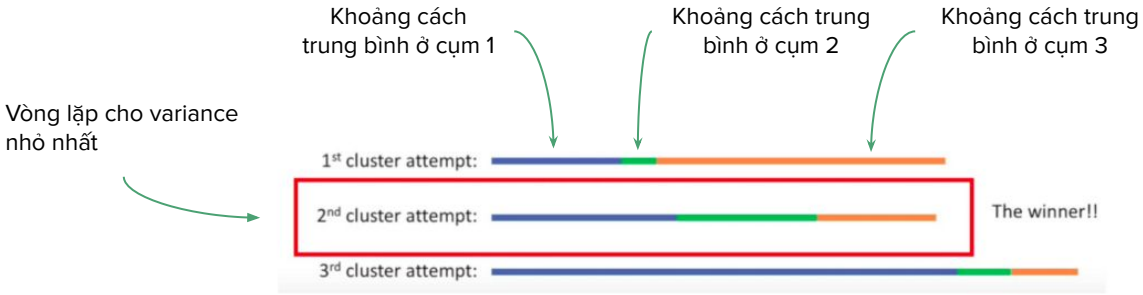
Bước 1. Chọn K điểm bất kỳ làm các centroid ban đầu.
Bước 2. Phân mỗi điểm dữ liệu vào cluster có centroid gần nó nhất.
Bước 3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
Bước 4. Cập nhật centroid cho từng cluster bằng cách lấy trung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

Demo

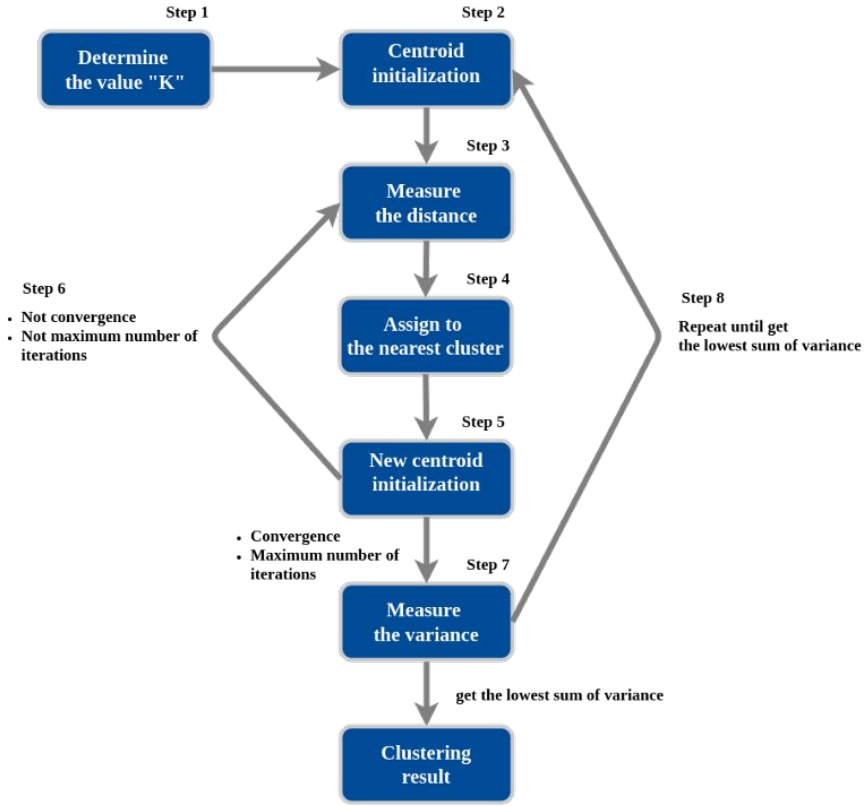
Khởi tạo các Centroid

Việc khởi tạo các điểm Centroids là yếu tố rất quan trọng trong thuật toán Kmeans. Giả dụ, chúng ta một tập hợp điểm dữ liệu phức tạp như hình 1, **việc khởi tạo vị trí ban đầu các Centroid ảnh hưởng đến hiệu suất của mô hình cũng như khái quát hoá dữ liệu**.

Do đó chúng ta cần chạy vài vòng lặp với các **điểm khởi tạo khác nhau** và chọn vòng lặp cho độ khác biệt của khoảng cách trung bình là nhỏ nhất



Flowchart


Step 1
Determine the value "K"
Step 2
Centroid initialization
Step 3
Measure the distance
Step 4
Assign to the nearest cluster
Step 5
New centroid initialization
Step 6
Not convergence
Not maximum number of iterations
Convergence
Maximum number of iterations
Step 7
Measure the variance
Step 8
Repeat until get the lowest sum of variance
get the lowest sum of variance
Clustering result

Pseudo - Python

```
def kmeans(dataSet, k):

    # Initialize centroids randomly
    numFeatures = dataSet.getNumFeatures()
    centroids = getRandomCentroids(numFeatures, k)

    # Initialize var.
    oldCentroids = None

    # Run the main k-means algorithm
    while not shouldStop(oldCentroids, centroids, iterations):
        # Save old centroids for convergence test.
        oldCentroids = centroids

        # Assign datapoint to centroids
        labels = getLabels(dataSet, centroids)

        # Produce new centroids
        centroids = getCentroids(dataSet, labels, k)

    # We can get the labels too by calling getLabels(dataSet,
centroids)
    return centroids
```

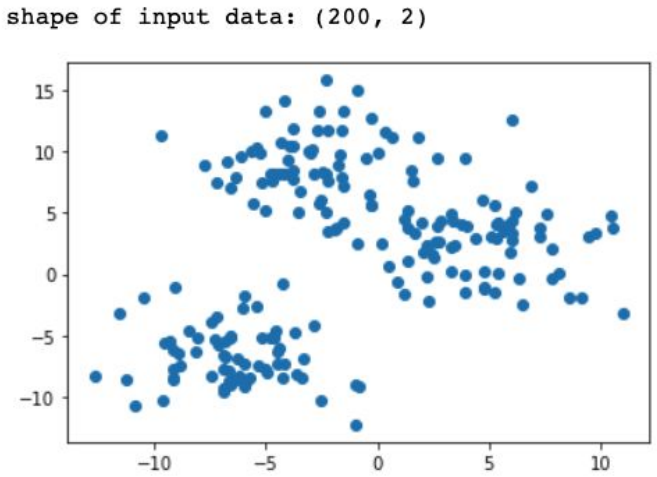
```
# Function: Should Stop
# -------------
# Returns True or False,  if True then k-means is done. it
means that there is no-changing.
def shouldStop(oldCentroids, centroids):
    return oldCentroids == centroids
```

```
# Function: Get Labels
# -------------
# Returns a label for each piece of data in the dataset.
def getLabels(dataSet, centroids):
    # For each element in the dataset, chose the closest
centroid.
    # Make that centroid the element's label.
```

```
# Function: Get Centroids
# -------------
# Returns k random centroids, each of dimension n.
def getCentroids(dataSet, labels, k):
    # Each centroid is the geometric mean of the points that
have that centroid's label.
    # Important: If a centroid is empty (no points have that
centroid's label) you should randomly re-initialize it.
```

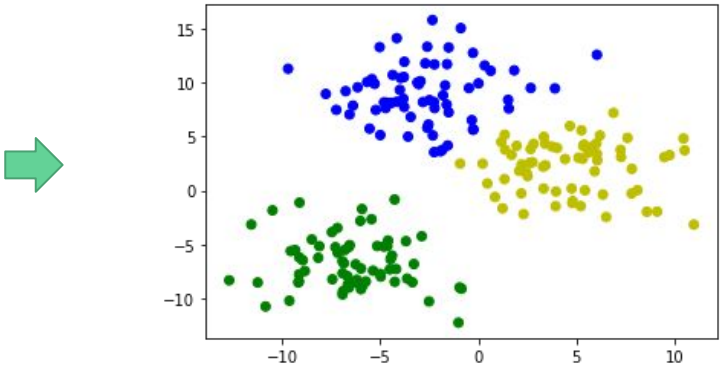
Ví dụ - Python

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
import numpy as np

features, true_labels = make_blobs(n_samples=200,n_features=2,
centers=3,cluster_std=2.75,random_state=42)

print('shape of input data:', features.shape)
plt.scatter(features[:,0],features[:,1])
plt.show()
```



Ví dụ - Python

```
kmeans = KMeans(n_clusters=3, random_state=0).fit(features)
color_dict = {0:'b', 1:'g', 2:'y'}
colors = [color_dict[i] for i in kmeans.labels_]
plt.scatter(features[:,0],features[:,1], c = colors)
plt.show()
```

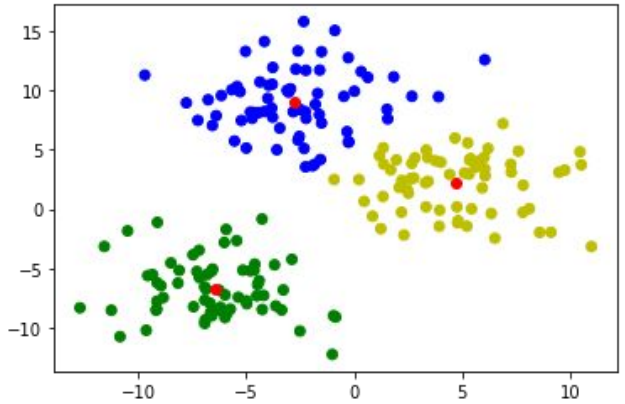



```
kmeans.cluster_centers_
```

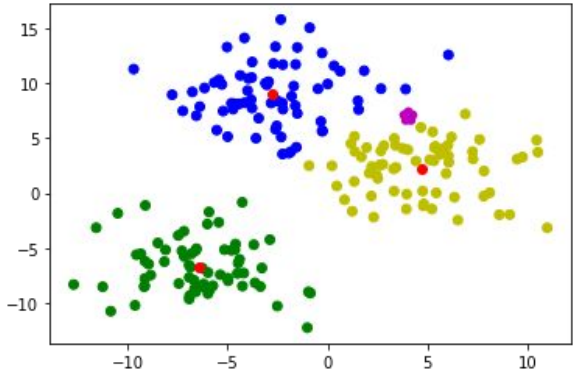
```
array([[-2.75094934,  8.97602228],
       [-6.47709913, -6.71236134],
       [ 4.69396233,  2.15039643]])
```

Ví dụ - Python

```
plt.scatter(features[:,0],features[:,1], c = colors)
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_
[:,1], color = 'r')
plt.show()
```




```
input_test = np.array([4, 7]).reshape(-1,2)
y_pre = kmeans.predict(input_test)[0]

plt.scatter(features[:,0],features[:,1], c = colors)
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_
[:,1], color = 'r')
plt.scatter(input_test[:,0],input_test[:,1], c = 'm', marker = '*',
linewidths = '6')
plt.show()
```



Bài tập nhóm số 3.

Tạo **300 điểm dữ liệu** với **4 phân cụm**, dùng thư viện sklearn để tìm điểm centroid của 4 phân cụng đó và vẽ lên trục toạ độ. (1 điểm)

Viết **hàm xác định centroid của 4 phân cụm** đó bằng thuật toán. (In toạ độ của các centroid) (5 điểm)

- Trong đó yêu cầu phải chạy thử 3 vòng lặp với các centroid được khởi tạo ngẫu nhiên khác nhau.
- In độ chênh lệch khoảng cách trung bình giữa các cụm.
- In ra lần phân cụm tốt nhất, số in ra là độ chênh lệch khoảng cách trung giữa các cụm

Vẽ tất cả các điểm và các centroid đã xác định lên trục toạ độ. (1 điểm)

Viết **hàm dự đoán nếu có input là 1 điểm dữ liệu** thì xác định nó thuộc centroid nào. (In centroid đó ra) (2 điểm)

Vẽ tất cả các điểm bao gồm điểm dữ liệu nhập vào và các centroid lên trục toạ độ. (1 điểm)