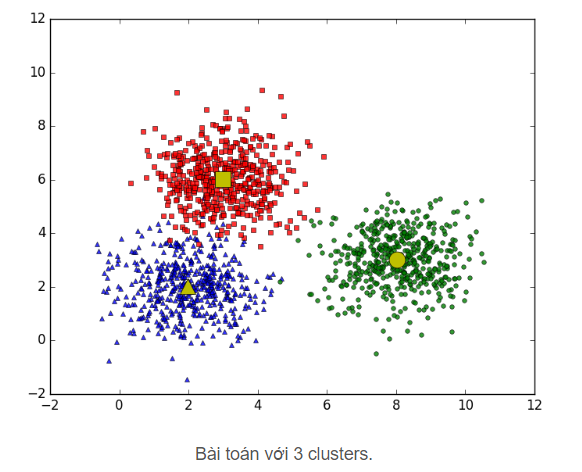
Thuật toán K-means Clustering

1.Giới thiệu

Thuật toán K-means Clustering là thuật toán đơn giản nhất trong bài toán Unsupervised learning. Trong thuật toán K-means clustering, chúng ta 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 cùng 1 tính chất.

**Ví dụ:** Một công ty muốn tạo ra những chính sách ưu đãi cho những nhóm khách hàng khác nhau dựa trên sự tương tác giữa mỗi khách hàng với công ty đó (số năm là khách hàng; số tiền khách hàng đã chi trả cho công ty; độ tuổi; giới tính; thành phố; nghề nghiệp; …).

Ý tưởng đơn giản nhất về cluster (cụm) là tập hợp các điểm ở gần nhau trong một không gian nào đó (không gian này có thể có rất nhiều chiều trong trường hợp thông tin về một điểm dữ liệu là rất lớn). Hình bên dưới là một ví dụ về 3 cụm dữ liệu



Giả sử mỗi cluster có một điểm đại diện (center) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó.

Bài toán:Trên một vùng biển hình vuông lớn có ba đảo hình vuông, tam giác, và tròn màu vàng như hình trên. Một điểm trên biển được gọi là thuộc lãnh hải của một đảo nếu nó nằm gần đảo này hơn so với hai đảo kia . Hãy xác định ranh giới lãnh hải của các đảo.

Hình dưới đây là một hình minh họa cho việc phân chia lãnh hải nếu có 5 đảo khác nhau được biểu diễn bằng các hình tròn màu đen:

Chart

Description automatically generated

Chúng ta thấy rằng đường phân định giữa các lãnh hải là các đường thẳng (chính xác hơn thì chúng là các đường trung trực của các cặp điểm gần nhau). Vì vậy, lãnh hải của một đảo sẽ là một hình đa giác.

2.Phân tích toán học

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

Với mỗi điểm dữ liệu xi đặt yi=[yi1,yi2,…,yiK] là label vector của nó, trong đó nếu xi được phân vào cluster k thì yik=1và yij=0,∀j≠k. Điều này có nghĩa là có đúng một phần tử của vector yi là bằng 1 (tương ứng với cluster của xi), các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có label vector là [1,0,0,…,0][1,0,0,…,0] thì nó thuộc vào cluster 1, là [0,1,0,…,0][0,1,0,…,0] thì nó thuộc vào cluster 2, ……. Cách mã hóa label của dữ liệu như thế này được gọi là biểu diễn [one-hot](https://en.wikipedia.org/wiki/One-hot).

A picture containing schematic

Description automatically generated

Hàm mất mát và bài toán tối ưu

Nếu ta coi center mk là center (hoặc representative) của mỗi cluster và ước lượng tất cả các điểm được phân vào cluster này bởi mk, thì một điểm dữ liệu xi được phân vào cluster k sẽ bị sai số là (xi−mk). Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ([giống như trong bài Linear Regression](https://machinelearningcoban.com/2016/12/28/linearregression/#sai-so-du-doan)) ta sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:

Hơn nữa, vì xi được phân vào cluster k nên yik=1,yij=0, ∀j≠k. Khi đó, biểu thức bên trên sẽ được viết lại là:

A screenshot of a computer

Description automatically generated with low confidence

Sai số cho toàn bộ dữ liệu sẽ là:

A picture containing text, watch

Description automatically generated

Trong đó Y=[y1;y2;…;yN], M=[m1,m2,…mK] lần lượt là các ma trận được tạo bởi label vector của mỗi điểm dữ liệu và center của mỗi cluster. Hàm số mất mát trong bài toán K-means clustering của chúng ta là hàm L(Y,M) với ràng buộc như được nêu trong phương trình (1).

Tóm lại, chúng ta cần tối ưu bài toán sau:

Text, letter

Description automatically generated

Thuật toán tối ưu hàm mất mát

#### **Cố định M, tìm Y**

**Giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất.** Điều này tương đương với việc tìm cluster cho mỗi điểm dữ liệu.

Khi các centers là cố định, bài toán tìm label vector cho toàn bộ dữ liệu có thể được chia nhỏ thành bài toán tìm label vector cho từng điểm dữ liệu xi như sau:

Text, letter

Description automatically generated

Vì chỉ có một phần tử của label vector yi bằng 1 nên bài toán (3) có thể tiếp tục được viết dưới dạng đơn giản hơn:

Diagram

Description automatically generated with low confidence

#### **Cố định Y, tìm M**

**Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.**

Một khi chúng ta đã xác định được label vector cho từng điểm dữ liệu, bài toán tìm center cho mỗi cluster được rút gọn thành:

Text, schematic

Description automatically generated

Đặt l(mj) là hàm bên trong dấu argmin, ta có đạo hàm:

Text

Description automatically generated

Giải phương trình đạo hàm bằng 0 ta có:

Text, letter, schematic

Description automatically generated

mj **là trung bình cộng của các điểm trong cluster** j.

### 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 ta dừng thuật toán.
4. Cập nhật center 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.
5. Quay lại bước 2.

## **3. Ví dụ trên Python**

Chart, scatter chart

Description automatically generated

Trong đồ thị trên, mỗi cluster tương ứng với một màu. Có thể nhận thấy rằng có một vài điểm màu đỏ bị lẫn sang phần cluster màu xanh.

Hình ảnh sau khi phân loại:

Chart, scatter chart

Description automatically generated

Text

Description automatically generated