**Phân cụm K-Means**

Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể 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 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,…). Giả sử công ty đó có rất nhiều dữ liệu của rất nhiều khách hàng nhưng chưa có cách nào chia toàn bộ khách hàng đó thành một số nhóm/cụm khác nhau. Lúc đó ta có thể sử dụng các phương pháp phân cụm, ví dụ như K-means.

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.



*Hình 1: bài toán phân cụm với 3 clusters*

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.

**Phân tích mặt toán học**

**Đặt vấn đề**

Giả sử có điểm dữ liệu là và là số cluster chúng ta muốn phân chia. Chúng ta cần tìm các center và label của mỗi điểm dữ liệu.

Với mỗi điểm dữ liệu đặt là label vector của nó, trong đó nếu được phân vào cluster thì và . Điều này có nghĩa là có đúng một phần tử của vector là bằng 1 (tương ứng với cluster của) các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có label vector là thì nó thuộc vào cluster 1, là thì nó thuộc vào cluster 2,...Cách mã hóa label của dữ liệu như thế này được gọi là biểu diễn **one-hot-encoding**.

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

**Hàm mất mát và bài toán tối ưu:**

Nếu ta coi center là center (hoặc representative) của mỗi cluster và ước lượng tất cả các điểm được phân vào cluster này bởi thì một điểm dữ liệu được phân vào cluster sẽ bị sai số là **( )**. Chúng ta mong muốn sai số này có giá trị tuyệt đối nhỏ nhất nên sai số này có giá trị tuyệt đối nhỏ nhất nên ta sẽ tìm cách để đại lượng sau có giá trị nhỏ nhất:

Hơn nữa, vì được phân vào cluster nên . Khi đó, biểu thức bên trên sẽ được viết lại là:

Sai số cho toàn dữ liệu sẽ là:

Trong đó, lần lượt là các ma trận được tạo bởi label vector của mỗi điểm dữ liệu và center của mỗi cluster. là hàm số mất mát với ràng buộc như trong phương trình .

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

**Thuật toán tối ưu hàm mất mát**

Một cách đơn giản để giải bài toán là xen kẽ giải và khi biến còn lại được cố định. Đây là một thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu. Chúng ta sẽ lần lượt giải quyết hai bài toán sau đây:

1. **Cố định M, tìm Y**

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:

**=**

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

Vì chính là bình phương khoảng cách tính từ điểm tới center , ta có thể kết luận rằng mỗi điểm thuộc vào cluster có center gần nó nhất. Từ đó ta có thể dễ dàng suy ra label vector của từng điểm dữ liệu.

1. **Cố định Y, tìm M**

Giá sử ta đã xác định được label vector vector cho từng điểm dữ liệu, bài toán tìm center cho mỗi cluster được rút gọn lại thành:

Tới đây, ta có thể tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, vì hàm cần tối ưu là một hàm liên tục và có đạo hàm xác định tại mọi điểm. Và quan trọng hơn, hàm này là hàm convex (lồi) theo nên chúng ta sẽ tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng.

Đặt là hàm bên trong dấu , ta có đạo hàm:

Giải phương trình đạo hàm bằng 0 ta có:

Ta thấy là trung bình cộng của các điểm trong cluster

**Thuật toán**

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

**Đầu ra:** Các center và label vector cho từng điểm dữ liệu .

1. Chọn 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.