**Introduction to Machine Learning**

My name: Pham Van Phuc

Student’s ID: 522H0068

**Homework: Clustering**

**Đánh giá chất lượng cụm trong K-Means**

* Tìm hiểu các phương pháp đánh giá như SSE (Inertia), Silhouette Score, Dunn Index.
  1. SSE (Inertia)

SSE đo tổng bình phương khoảng cách từ các điểm dữ liệu đến tâm cụm.

Công thức: 

Trong đó:

k là số cụm.

Ci là tập hợp các điểm trong cụm

 là tâm cụm của cụm

 là bình phương khoảng cách giữa điểm dữ liệu x và tâm cụm

* 1. Silhouette Score

Đo mức độ gắn kết trong cụm và sự tách biệt giữa các cụm.

Công thức: 

Trong đó :

s(i) là giá trị Silhouette của điểm dữ liệu

a(i) là khoảng cách trung bình từ i đến tất cả các điểm khác trong cùng cụm.

b(i) là khoảng cách trung bình từ i đến các điểm trong cụm gần nhất khác với cụm của i.

Giá trị Silhouette:

+1: Nhóm rõ ràng, thể hiện khả năng phân cụm tốt.

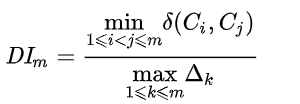
0: Trồng lẫn giữa hai cụm.

-1: Phân cụm kém.

* 1. Dunn Index

Tỉ lệ khoảng cách nhỏ nhất giữa hai cụm so với đường kính lớn nhất của một cụm

Với ký hiệu trên, nếu có m cụm, thì Chỉ số Dunn cho tập hợp được định nghĩa là:



với  δ(Ci,Cj) là khoảng cách giữa các cụm Ci Civaàvà Cj Cjtrong , Δklà khoảng cách trong cụm, ví dụ khoảng cách tối đa trong một cụm khi tuân theo định nghĩa ban đầu của Dunn.

* So sánh ưu nhược điểm của từng phương pháp.
  1. SSE (Inertia)
* Ưu điểm
  + Dễ hiểu, trực quan.
  + Tính toán nhanh, phù hợp với dữ liệu lớn.
* Nhược điểm
  + Dễ bị ảnh hưởng bởi outliers, làm sai lệch cụm.
  + Không đánh giá được sự chồng chéo giữa các cụm.
  1. Silhouette Score
* Ưu điểm
  + Đánh giá được chất lượng phân cụm tốt hơn SSE.
  + Giúp chọn số lượng cụm tối ưu.
* Nhược điểm
  + Tính toán chậm hơn SSE, đặc biệt với dữ liệu lớn.
  + Nhạy cảm với hình dạng cụm không đều.
  1. Dunn Index
* Ưu điểm
  + Đánh giá được sự tách biệt giữa các cụm.
  + Ít bị ảnh hưởng bởi outliers hơn SSE.
* Nhược điểm
  + Tính toán phức tạp, khó áp dụng cho dữ liệu lớn.
  + Hiệu quả giảm khi số lượng cụm lớn.

**Xác định số lượng cụm (k) trong K-Means**

* Nghiên cứu các phương pháp chọn k như Elbow Method, Silhouette Score, Gap Statistics.
  + Elbow Method

Phương pháp này dựa trên việc phân tích tổng bình phương sai số (SSE - Sum of Squared Errors), còn được gọi là **inertia**. SSE đo tổng khoảng cách bình phương từ mỗi điểm dữ liệu đến tâm cụm của nó.

Cách thực hiện:

1. Chạy thuật toán K-Means với nhiều giá trị k khác nhau.
2. Tính SSE cho từng giá trị k.
3. Vẽ biểu đồ SSE theo số cụm k.
4. Xác định điểm mà SSE bắt đầu giảm chậm lại, tạo thành "khuỷu tay" trên đồ thị.

Ví dụ :

Giả sử ta thử các giá trị k từ 1 đến 5 và thu được bảng SSE sau:

|  |  |
| --- | --- |
| k | SSE |
| 1 | 900 |
| 2 | 300 |
| 3 | 275 |
| 4 | 250 |
| 5 | 225 |

ta thấy SSE giảm mạnh từ k=1 đến k=3, sau đó chậm lại. Do đó, điểm gập "khuỷu tay" xuất hiện tại k=3, nên ta chọn k=3 làm số cụm tối ưu.

* + Silhouette Score

Phương pháp này dựa trên **Silhouette Score**, một chỉ số đánh giá mức độ gắn kết trong cụm và sự tách biệt giữa các cụm.

Cách thực hiện :

1. Chạy thuật toán K-Means với nhiều giá trị k khác nhau.
2. Tính Silhouette Score cho từng giá trị k.
3. Vẽ biểu đồ Silhouette Score theo số cụm k.
4. Chọn giá trị k tương ứng với Silhouette Score cao nhất.

Ví dụ:

Giả sử ta thử các giá trị k từ 2 đến 6 và thu được bảng Silhouette Score sau:

|  |  |
| --- | --- |
| k | SSE |
| 2 | 0.6 |
| 3 | 0.72 |
| 4 | 0.75 |
| 5 | 0.68 |
| 6 | 0.65 |

Ta thấy Silhouette Score đạt giá trị cao nhất tại k=4, nên ta chọn k=4 làm số cụm tối ưu.

* + Gap Statistics

Phương pháp này so sánh SSE thu được từ tập dữ liệu thực tế với SSE thu được từ một tập dữ liệu ngẫu nhiên có cùng kích thước. Nếu tập dữ liệu thực có cấu trúc rõ ràng, SSE của nó sẽ thấp hơn so với dữ liệu ngẫu nhiên.

Công thức: 

Trong đó :

 là SSE của dữ liệu thực với k cụm.

là SSE trung bình của B tập dữ liệu ngẫu nhiên.

 là số lần tạo tập dữ liệu ngẫu nhiên.

Cách thực hiện:

1. Chạy K-Means trên dữ liệu thực tế với các giá trị k khác nhau.
2. Chạy K-Means trên tập dữ liệu ngẫu nhiên có cùng số điểm.
3. So sánh SSE của dữ liệu thực với SSE của dữ liệu ngẫu nhiên.
4. Chọn k sao cho Gap(k) lớn nhất.

Ví dụ: Giả sử ta thử các giá trị kkk từ 1 đến 6 và tính được các giá trị Gap như sau:

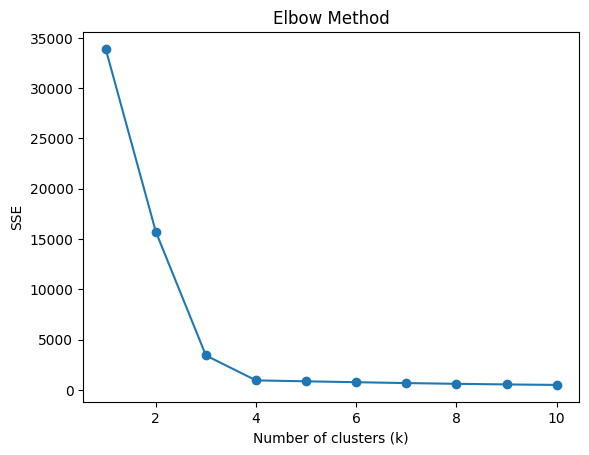
|  |  |
| --- | --- |
| k | Gap(k) |
| 1 | 0.5 |
| 2 | 1.2 |
| 3 | 1.8 |
| 4 | 2.1 |
| 5 | 2.0 |
| 6 | 1.7 |

Ta thấy Gap đạt giá trị cao nhất tại k=4, nên ta chọn k=4 làm số cụm tối ưu.

* Thực hiện ví dụ minh họa bằng Python.

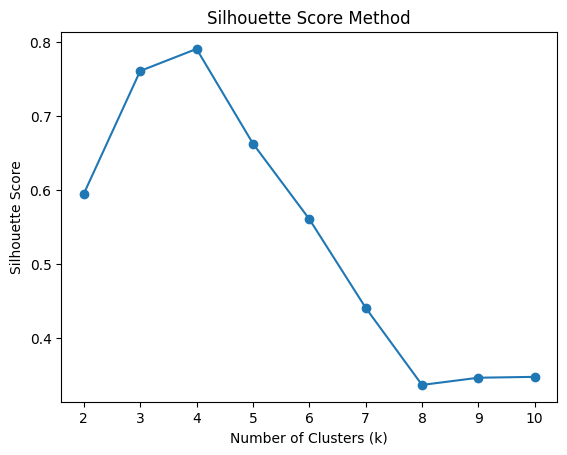
Link mã nguồn : <https://colab.research.google.com/drive/15iAknlXlUoDXLmaDR_zwSjRv6LHRjDVp?usp=sharing>

* + Elbow Method



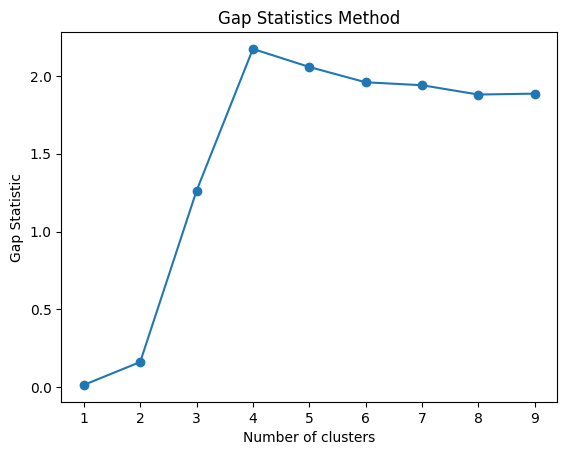
Như trong kết quả của ví dụ minh họa trên: Sau khi hạy thuật toán K-Means với nhiều giá trị k khác nhau và tính toán SSE cho từng giá trị k ta thấy SSE giảm mạnh từ k=1 đến k=3, sau đó chậm lại. Do đó, điểm gập "khuỷu tay" xuất hiện tại k=3, nên ta chọn k=3 làm số cụm tối ưu.

* + Silhouette Score



Như trong kết quả của ví dụ minh họa trên: Sau khi hạy thuật toán K-Means với nhiều giá trị k khác nhau và tính toán Silhouette Score cho từng giá trị k ta thấy Silhouette Score đạt giá trị cao nhất tại k=4, nên ta chọn k=4 làm số cụm tối ưu.

* + Gap Statistics



Như trong kết quả của ví dụ minh họa trên: Chạy K-Means trên dữ liệu thực tế với các giá trị k khác nhau. Sau đó chạy K-Means trên tập dữ liệu ngẫu nhiên có cùng số điểm. Cuối cùng so sánh SSE của dữ liệu thực với SSE của dữ liệu ngẫu nhiên. Ta thấy giá trị của Gap(k) lớn nhất tại k = 4 vậy nên ta chọn k=4 làm số cụm tối ưu.

**Giải thuật DBSCAN và loại dữ liệu phù hợp**

* DBSCAN là thuật toán phân cụm dựa trên mật độ, có thể xác định các cụm có hình dạng bất kỳ và phát hiện được nhiễu (outliers).

Cách hoạt động:

DBSCAN sử dụng hai tham số chính:

* + - ϵ (Epsilon - Bán kính lân cận): Xác định khoảng cách tối đa để một điểm được coi là hàng xóm của điểm khác.
    - minPts (Minimum Points - Số điểm tối thiểu): Số lượng điểm tối thiểu cần có trong một khu vực bán kính ϵ để được coi là cụm dày đặc.

Các loại điểm trong DBSCAN:

* + Core point (Điểm lõi): Có ít nhất minPts điểm trong bán kính ϵ.
  + Border point (Điểm biên): Không phải điểm lõi, nhưng nằm trong vùng ϵ của một điểm lõi.
  + Noise point (Điểm nhiễu): Không phải điểm lõi cũng không phải điểm biên.

Các bước của thuật toán:

* + Chọn một điểm chưa được phân loại.
  + Nếu điểm đó là core point, mở rộng cụm bằng cách thêm các điểm lân cận vào cụm.
  + Nếu điểm đó là border point, nó được thêm vào cụm của core point gần nhất.
  + Nếu điểm không thuộc cụm nào (noise point), nó được đánh dấu là nhiễu.
  + Lặp lại cho đến khi tất cả các điểm đều được phân loại.
* So sánh với K-Means, xác định loại dữ liệu phù hợp.

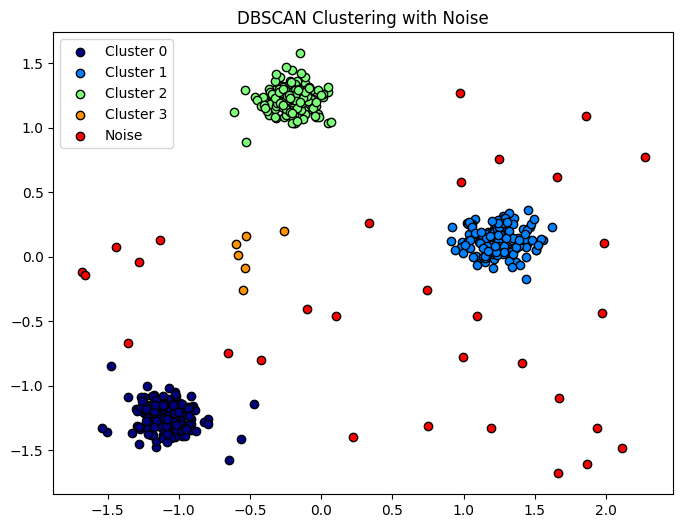
|  |  |  |
| --- | --- | --- |
| Nội dung | DBSCAN | K-Means |
| Nguyên lý | Dựa vào mật độ điểm dữ liệu | Dựa vào khoảng cách giữa điểm dữ liệu và tâm cụm |
| Số cụm k | Không cần xác định trước | Cần xác định trước k |
| Xử lý nhiễu | Có thể xác định điểm nhiễu | Không phát hiện được nhiễu |
| Dạng cụm | Nhận dạng cụm có hình dạng bất kỳ | Cụm có dạng hình cầu |
| Thích hợp cho | Dữ liệu có nhiễu, cụm không đối xứng | Dữ liệu có cụm rõ ràng, dạng hình cầu |
| Nhược điểm | Hoạt động kém với cụm có mật độ thay đổi | Không xử lý tốt dữ liệu có nhiễu |

DBSCAN phù hợp với dữ liệu có nhiễu và không cần xác định số cụm trước.

K-Means thích hợp khi dữ liệu có cụm hình cầu và số cụm đã biết trước.

* Viết chương trình Python áp dụng DBSCAN trên dữ liệu có nhiễu.

Link mã nguồn: <https://colab.research.google.com/drive/15iAknlXlUoDXLmaDR_zwSjRv6LHRjDVp?usp=sharing>



Dữ liệu ban đầu có 3 cụm chính và thêm 50 điểm nhiễu.

DBSCAN đã tự động xác định số cụm, thay vì phải chỉ định như K-Means.

Các điểm nhiễu được tô màu đỏ, chứng tỏ thuật toán có khả năng phát hiện outliers tốt.

Các điểm thuộc cùng một cụm được gom chặt chẽ, phản ánh khả năng nhận diện mật độ tốt.

**Feature Selection trong Clustering**

* Vai trò của lựa chọn đặc trưng trong phân cụm.
  + **Cải thiện độ chính xác** của phân cụm
  + **Giảm thời gian tính toán**
  + **Tăng khả năng giải thích** kết quả
* Mô tả các kỹ thuật như PCA, Feature Importance, Correlation-based Selection.
  + PCA (Phân tích thành phần chính - Principal Component Analysis)

Là phương pháp giảm chiều dữ liệu, giúp chuyển đổi dữ liệu sang hệ tọa độ mới dựa trên phương sai.

Giữ lại những thành phần quan trọng nhất và loại bỏ các thông tin dư thừa.

Giúp tăng hiệu suất phân cụm trên dữ liệu có số chiều lớn.

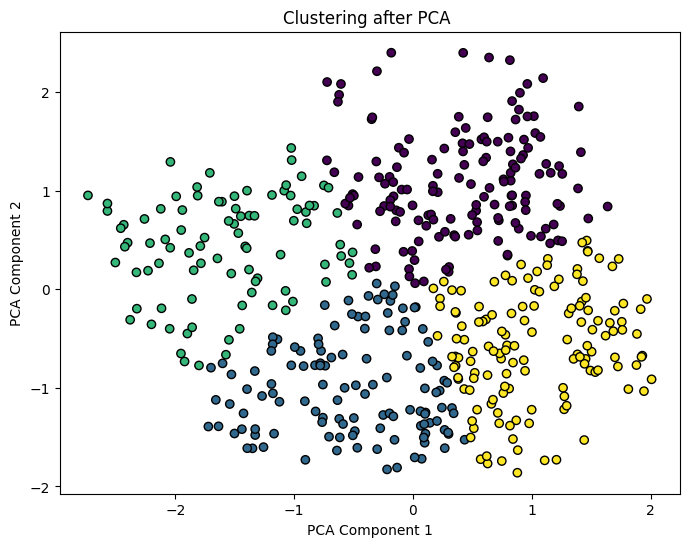
* + Feature Importance (Tầm quan trọng của đặc trưng)

Một số thuật toán như Random Forest hoặc Gradient Boosting có thể tính toán mức độ quan trọng của từng đặc trưng.

Các đặc trưng có mức độ đóng góp thấp có thể bị loại bỏ để cải thiện hiệu suất phân cụm.

* + Correlation-based Feature Selection (Lựa chọn đặc trưng dựa trên tương quan)
  + Các đặc trưng có độ tương quan cao với nhau có thể chứa thông tin trùng lặp.
  + Giữ lại đặc trưng đại diện nhất để giảm dư thừa dữ liệu.
* Áp dụng PCA để giảm chiều dữ liệu trước khi phân cụm.
  + Dữ liệu ban đầu có 5 chiều (3 đặc trưng từ cụm gốc + 2 đặc trưng nhiễu).
  + PCA giảm số chiều từ 5 xuống 2, giúp trực quan hóa dữ liệu dễ dàng hơn.
  + PCA giữ lại thông tin quan trọng nhất, giúp K-Means phân cụm chính xác hơn.

Link mã nguồn: <https://colab.research.google.com/drive/15iAknlXlUoDXLmaDR_zwSjRv6LHRjDVp?usp=sharing>



Mỗi cụm được biểu diễn bằng một màu khác nhau.

Kết quả phân cụm khá rõ ràng, chứng tỏ PCA giúp loại bỏ nhiễu tốt.