TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA TOÁN - ỨNG DỤNG

BÁO CÁO THỰC HÀNH

CÁC GIẢI THUẬT PHÂN CỤM CƠ BẢN

SVTH: NGUYỄN ĐĂNG TIẾN

MSSV: 3123580050

GVHD: TS. ĐỖ NHƯ TÀI

Năm học: 2025 - 2026

Mục lục

Mục tiêu chung

- Hướng dẫn thực hiện phân cụm dữ liệu với giải thuật K-means
- Hướng dẫn thực hiện phân cụm đa cấp
- Hướng dẫn ôn tập các kiến thức lý thuyết về K-means và phân cụm đa cấp

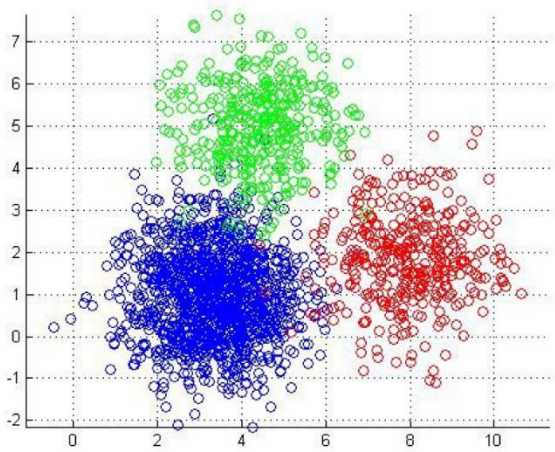
Nội dung bài thực hành bao gồm 2 phần ứng với 2 chương :

- Chương 1: Giải thuật K-means
- Chương 2: Giải thuật Phân cụm đa cấ

CHƯƠNG I: GIẢI THUẬT K-MEANS

1.1. Ôn tập lý thuyết

Giải thuật K-Means hoạt động như thế nào? Hãy giải thích các bước chính trong quy trình phân cụm?



K-Means Clustering

Giải thuật K-Means là một thuận toán phân cụm thuộc nhóm học không giám sát, dùng để chia dữ liệu thành K cụm sao cho:

- Các điểm trong cũng một cụm thì giống nhau (gần nhau)
- Các điểm ở khác cụm thì khác khác (xa nhau)

Thuật toán hoạt động dựa trên việc tối thiểu hỏa tổng bình phương khoảng cách từ mỗi điểm đến tâm cụm

Quy trình thực hiện của giải thuật K-means

1. **Khởi tạo K tâm cụm**
   - Chọn ngẫu nhiên k diểm bất kỳ trong dữ liệu làm tam
2. **Gán mỗi điểm dữ liệu vào cụm gần nhất**
   - Dựa theo công thức tính khoảng cách Euclid:

$$d(x,\mu) = \sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2}$$

- Điểm nào gần tâm nhất thì gom chúng thành một cụm

3. **Cập nhật lại tâm của từng cụm**
   - Trong mỗi cụm vừa được gom tiếp tục lựa chọn tâm mới để gom
     Có thể tính bằng trung bình tất cả các điểm trong cụm

$$\mu_j = \frac{1}{|C_j|}\sum_{x \in C_j} x$$

4. **Lặp lại**
   - Lặp lại các bước cho đến khi tâm giữa các cụm không thay đổi nữa hoặc số vòng lặp đạt ngưỡng

Tại sao cần chọn số lượng cụm (K) trước khi chạy K-Means? Làm thế nào để xác định giá trị K tối ưu?

- Chọn K trước khi chạy nhằm biết phải sinh trước bao nhiêu điểm tâm cụm, định hình không gian để có thể phân cụm bạn đầu
- Có 3 phương pháp thường được sử dụng để xác định một giá trị K tối ưu
  - Elbow Method
    - Trên biểu đồ WCSS chọn ra các điểm nằm trên "gấp khúc"
  - Silhouette Score
    - Đo độ phân tách giữa các cụm

  Công thức Silhouette Score:

1. **Đo mức độ chặt chẽ của cụm**
   Khoảng cách trung bình đến các điểm trong cùng cụm

$$a(i) = \frac{1}{|C_i| - 1}\sum_{x \in C_i, x \neq i} d(i,x)$$

2. **Đo mức độ tách biệt với các cụm lân cận**

   Khoảng cách trung bình đến các điểm cảu cụm gần nhất khác

$$b(i) = min\frac{1}{|C_k|}\sum_{x \in C_k} d(i,x)$$

3. **Silhouette Score của một điểm**

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)}$$

4. Silhouette Score toàn bộ mô hình phân cụm

$$S = \frac{1}{N}\sum_{i=1}^{N} s(i)$$

- Gap Static
  - So sánh phân tán cụm với dữ liệu giả định chuẩn

Hàm mục tiêu (objective function) của K-Means là gì? Nó đo lường điều gì trong quá trình phân cụm?

Hàm mục tiêu của K-means chính là hàm giảm tối thiểu hóa khoảng cách từ các điểm đến tâm cụm

$$J(\{C_i\},\{\mu_i\}) = \sum_{i=1}^{K}\sum_{x\in C_i} |x - \mu_i|^2$$

Trong đó:

- K: số cụm
- C: mỗi cụm thứ i
- u_i: tâm nằm trong mỗi cụm C
- X: là dữ liệu trong cụm

Hàm mục tiêu nhằm đo lường tổng khoảng cách từ các điểm đến tâm trong cụm

Những hạn chế của K-Means là gì? Trong trường hợp nào K-Means có thể cho kết quả không tốt?

Hạn chế của K-means:

- Giả định cụm có dạng hình cầu
- Nhạy cảm với giá trị khởi tạo
- Nhảy cảm với dữ liệu ngoại lai/nhiễu
- Không tốt khi phân bố cụm chồng lấp
- Chỉ dùng được cho dữ liệu số

Các trường hợp có thể gây ra kết quả không tốt:

- Cụm dài, méo mỏ
- Cụm có kích thước khác nhau
- Dữ liệu không tuyến tính (vòng tròn lồng nhau)

Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai K-Means Clustering không? Hãy mô tả các bước thực hiện

```
from sklearn.cluster import KMeans

from sklearn.datasets import make_blobs
```

```
import matplotlib.pyplot as plt


# Tạo dữ liệu mẫu
X, y = make_blobs(n_samples=300, centers=4,
random_state=42)


# Khởi tạo KMeans
kmeans = KMeans(n_clusters=4, init='k-means++',
random_state=42)
kmeans.fit(X)


# Lấy nhãn và tâm cụm
labels = kmeans.labels_
centers = kmeans.cluster_centers_


# Vẽ kết quả
plt.scatter(X[:, 0], X[:, 1], c=labels)
plt.scatter(centers[:, 0], centers[:, 1], s=200,
marker='X')
plt.show()
```

Sử dụng phương pháp nào trong Python để chọn số cụm K tối ưu (ví dụ: Elbow Method, Silhouette Score)? Hãy chia sẻ một đoạn code mẫu

Phương pháp Elbow Method

```
inertia_values = []
K_range = range(1, 10) # Lựa chọn k từ 1 đến 10


for k in K_range:
    km = KMeans(n_clusters=k, random_state=42)
    km.fit(X)
```

```
    inertia_values.append(km.inertia_)


plt.plot(K_range, inertia_values, marker='o')
plt.xlabel('K')
plt.ylabel('WCSS (Inertia)')
plt.show()
```

Phương pháp Sihouette Score

```
from sklearn.metrics import silhouette_score


scores = []
for k in range(2, 10):
    km = KMeans(n_clusters=k, random_state=42).fit(X)
    scores.append(silhouette_score(X, km.labels_))


plt.plot(range(2, 10), scores, marker='o')
plt.xlabel('K')
plt.ylabel('Silhouette score')
plt.show()
```

K-Means nhạy cảm với giá trị khởi tạo (initial centroids), bạn sẽ làm gì để đảm bảo kết quả ổn định (ví dụ: K-Means++)?

- Sử dụng K-Means++ trong sklearn nhằm chọn tâm k cụm tốt hơn
- Chạy nhiều lần
- Scale dữ liệu trước

Làm thế nào để đánh giá chất lượng của các cụm được tạo bởi K-Means? Bạn sử dụng chỉ số nào (ví dụ: Silhouette Score, Within-Cluster Sum of Squares)?

Sử dụng các chỉ số

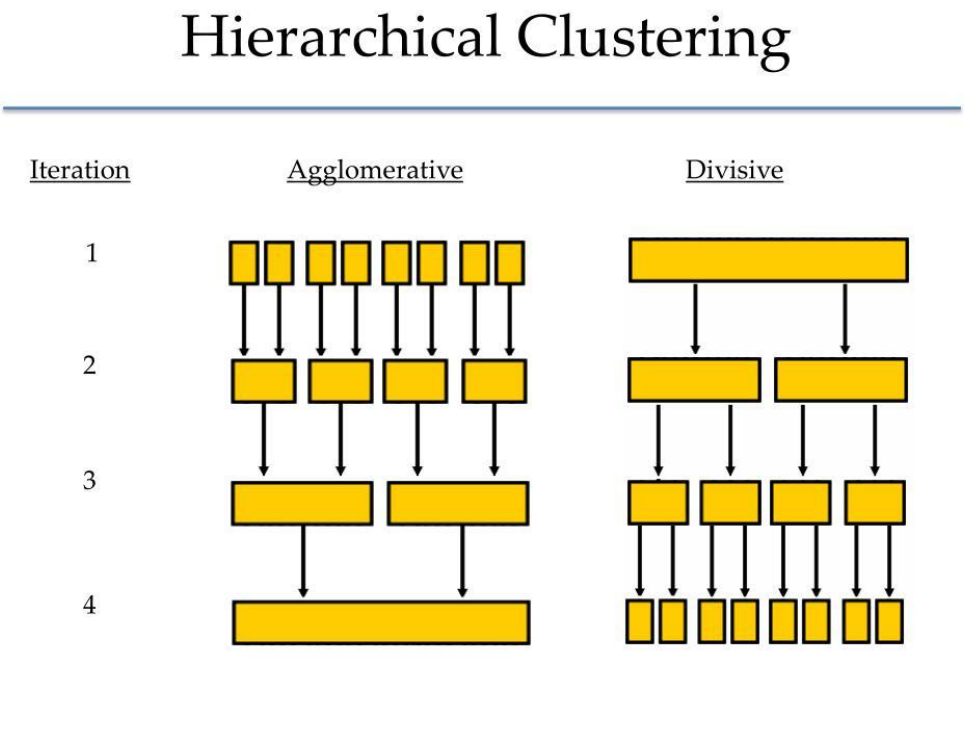
- Silhouette Score - Độ tách biệt giữa các cụm
- WCSS / Inertia - Độ compact trong cụm (càng nhỏ càng tốt)
- Davies-Bouldin Index - Càng nhỏ càng tốt

- Calinski–Harabasz Index - Càng lớn càng tốt

CHƯƠNG II: PHÂN CỤM PHÂN CẤP (HIERARCHICAL CLUSTERING)

2.1. Ôn tập lý thuyết

Giải thuật phân cụm đa cấp hoạt động như thế nào? Hãy giải thích sự khác biệt giữa phân cụm đa cấp hợp nhất (agglomerative) và phân tách (divisive)



Chính là chiến lược chia bài toán phân cụm lớn thành nhiều mức. Thay vì chạy phân cụm trực tiếp trên dữ liệu gốc thuật toán sẽ :

1. Làm thô dữ liệu – Giam kích thước dữ liệu
2. Phân cụm trên mức thô nhất
3. Tinh chính và mở rộng lại dữ liệu – Gán nhãn ngược lên các mức chi tiết hơn và cải thiện dần cụm

Có 2 loại chính:là:

Agglomerative (Bottom-up — từ dưới lên)

- Đi từ dữ liệu thô gom dần gom dần thành những siêu cục lớn hơn sau đó phân cụ mtreen cấp độ cao rôi gán lại xuống dưới
- Cơ chế: Kết tụ (kết hợp các điểm gần nhau thành các siêu nút, Mỗi bước kích thước dữ liệu giảm mạnh) -> Phân cụm ở mức thô (Áp dụng thuật toán phân cụm trên mức độ nhỏ) -> Mở rộng và tinh chỉnh (Gán nhãn tư mức thô xuống chi tiets, Điều chỉnh để tối ưu hóa mục tiêu)

Divisive (Top-down — từ trên xuống)

- Đi từ tập dữ liệu lớn tách dần thành các cụm nhỏ hơn
- Cơ chế: Chia cụm gốc thành các cụm lớn (Dùng thuật toán tách mạnh) -> Tiếp tục chia thành các cụm con -> Dừng khi đã đạt số cụm mong muốn

So sánh 2 phương pháp

- Kết tụ tốt khi có dữ liệu phan cấu trúc rõ rang
- Phân chia tốt khi dữu liệu có ranh giới tự nhiên rõ
- Chi phsi tính toán phân chia cao hơn
- Việc thực hiện đa cấp của kết tụ dễ hơn
- Tính ổn định của kết tự ổn hơn
- Phân chia nhạy cảm với bước tách đầu tiên

Các phương pháp liên kết (linkage) như single linkage, complete linkage, average linkage, và Ward's method khác nhau ra sao? Khi nào nên sử dụng từng loại

Single linkage (Liên kết đơn - khoảng cách nhỏ nhất)

- Khoảng cách giữa 2 cụm = Khoảng cách nhỏ nhất giữa bất kỳ 2 điểm thuộc mỗi cụm
- Đặc điểm tạo thành chuỗi dài
- Nhạy cảm với điểm nhiễu và lẻ

Dùng khi: Muốn phát hiện hình dạng cụm bấy kỳ

Complete linkage (Liên kết toàn phần – Khoảng cách lớn nhất)

- Khoảng cách giữa 2 cụm = Khoảng cách lớn nhất giữa bất kỳ 2 điểm thuộc mỗi cụm
- Tạo cụm chặt – tròn – nhỏ
- Khắc phục được hiệu ứng – kéo chuỗi
- Nhạy hơn với dữ liệu ngoại lai

Dùng khi: Muốn cụm tách biệt và có danh giới rõ rang

Average linkage (Liên kết trung bình - UPGMA)

- Khoảng cách giữa 2 cụm = trung bình khoảng cách tất cả các cặp điểm thuộc 2 cụm
- Dùng hòa giữa 2 phương pháp trên
- Cụm khá ổn định, không quá rộng hay quá chặt
- Bớt nhạy với ngoại lai

Dùng khi: Dữ liệu thông thường và không có định dạng đặc biệt

Ward's method

- Gộp 2 cụm sao cho tăng trưởng tổng phương sai trong cụm là nhỏ nhất
- Tao cụm rất tròn – rất rõ rang – độ chặt cao
- Ổn dịnh và thường cho kết quả tốt nhất
- Ít nhạy với nhiễu hơn

Dùng khi: Khi dữ liệu dạng số phân bố tương đối chuẩn, phân cụm khách hàng, marketing, dữ liệu xã hội học

So sánh 4 phương pháp

Phương pháp	Tạo cụm	Ưu điểm	Nhược điểm
Single linkage	Dài, lỏng, dễ kéo chuỗi	Bắt hình dạng cụm bất thường	Nhạy nhiễu, cụm xấu
Complete linkage	Nhỏ, chặt	Cụm rõ ràng	Nhạy với outlier
Average linkage	Trung hòa	Ổn định	Chậm, trung bình
Ward's method	Tròn, chặt	Chất lượng cao, phổ biến nhất	Chỉ dùng với dữ liệu số

Dendrogram trong phân cụm đa cấp là gì? Làm thế nào để sử dụng nó để chọn số lượng cụm?

Dendrogram là một biểu đồ dạng cây dùng trong phân cụm phân cấp (Hierarchical Clustering) để biểu diễn quá trình gộp cụm hoặc tách cụm.

Cho thấy:

- Những điểm nào được ghép trước
- Khoảng cách (độ khác biệt) giữa các cụm khi ghép
- Bao nhiêu cụm phù hợp (nhờ nhìn các "khoảng cách nhảy vọt")

Một dendrogram có 3 phần:

Lá (leaves)

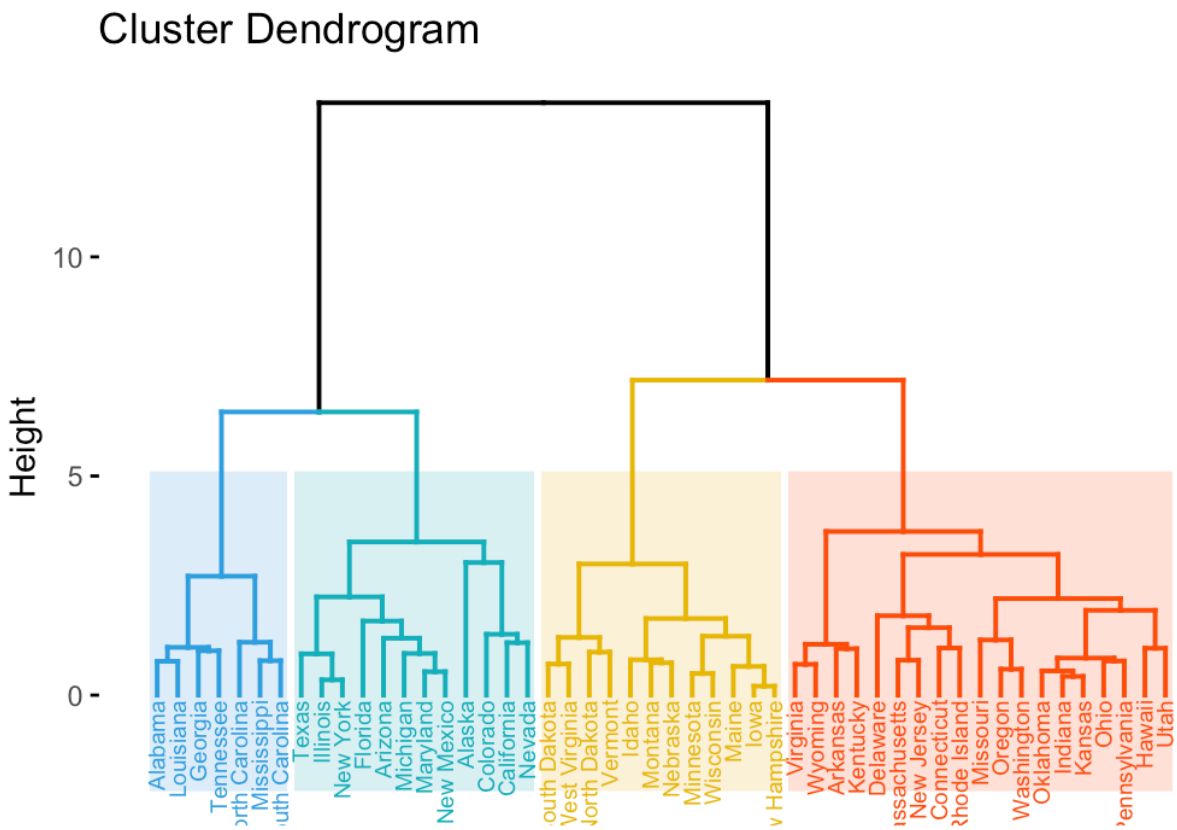
- Là các điểm dữ liệu ban đầu.

Các nhánh (branches)

- Cho biết **hai điểm hoặc hai cụm** nào được gộp với nhau.

Chiều cao (height)

- Biểu thị **khoảng cách hoặc độ khác biệt** khi hai cụm được ghép.



Chọn số lượng cụm thông qua nguyên tắc nhảy vọt

- Tìm chỗ có một nhảy vọt lớn nhất về chiều cao
- Rồi cắt dendrogram tại một mức ngay trước khi khoảng nhảy vọt xảy ra

Quy tắc chọn số cụm

- Chọn ngay trước bước ghép lớn nhất
- Elbow rule trên dendrogram
- Tối đa hóa khoảng cách giữa các cụm
- Hãy cắt dendrogram ở nơi mà khoảng cách giữa các mức ghép tăng đột biến
- Nếu dendrogram giống cái thang → chọn số bậc thấp nhất trước khi bậc cao xuất hiện
- Đừng chọn cắt ở phần trên cùng — đó là lúc bạn đang ép những cụm rất khác nhau phải ghép lại với nhau.

Phân cụm đa cấp có thể áp dụng cho dữ liệu phi số (non-numeric data) như thế nào? Hãy giải thích

Phân cụm đa cấp thường đươc dùng cho dữ liệu số vì

- Dễ tính khoảng cách Euclid
- Dễ gom và tách cụm
- Có thể dùng K-means

Có thể áp dụng các chiến lược để có thể áp dụng phân cụm đa cấp cho dữ liệu phi số

- Embedding chuyển dữ liệu phi số thành dạng số
- Sử dụng khoảng cách chuyên biệt
- Xây dựng đồ thi rồi phân cụm đa cấp trên đồ thị

Viết đoạn code mẫu bằng Python (sử dụng Scikit-learn) để triển khai phân cụm đa cấp hợp nhất (agglomerative clustering) không? Hãy mô tả các bước thực hiện

```
from sklearn.cluster import AgglomerativeClustering


model = AgglomerativeClustering(n_clusters=4,
linkage='ward')
labels = model.fit_predict(X)
```

Làm thế nào để vẽ dendrogram trong Python sử dụng thư viện như scipy hoặc matplotlib? Hãy chia sẻ một đoạn code mẫu

```
from scipy.cluster.hierarchy import dendrogram,
linkage
import matplotlib.pyplot as plt


Z = linkage(X, method='ward')
```

```
plt.figure(figsize=(10, 5))
dendrogram(Z)
plt.show()
```

Các lớp trong gói Scipy hỗ trợ phân cụm đa cấp? và so sánh giữa cách tiếp cận scikit-learn và cách tiếp cận sử dụng Scipy

SciPy class	***Công dụng***
`scipy.cluster.hierarchy.linkage`	*Xây ma trận liên kết*
`dendrogram`	*Vẽ cây phân cấp*
`fcluster`	*Cắt cây để lấy nhãn cụm*
`distance.pdist`	*Tính khoảng cách*

So sánh giữa sklearn và Scipy

Scipy

- Làm việc sâu với dendrogram
- Nghiên cứu các liên kết single/complete/average/ward chi tiết
- Phân tích phân cụm phân cấp như trong khai phá dứx liệu
- Học làm về lý thuyết phân cụm đa cấp
- Cần truy cập sâu vào cấu trúc cây phân cấp
- Viết thuật toán và nghiên cứu phương pháp

Sklearn

- Phân cụm để dung trong các pipeline học máy đánh giá triển khai, những gì Scipy không có về phân cụm thì sklearn đều có
- Không có đa cấp đúng nghĩa nhưng có nhiều thuật toán có thể được dùng theo từng cấp
- Dùng khi chạy phân cụm đánh giá nhiều thuật toán
- Cần nhiều thuật toán phân cụm khác
- Chuẩn hóa dữ liệu, giảm chiều, đánh giá cum, ..