KHAI THÁC DỮ LIỆU

Nguyễn Xuân Việt Đức - 22280012

Bài tập thực hành - Lần 7

Bài 7: SỰ PHÂN TÍCH NHÓM (TIẾP THEO)

III. Nội dung thực hành:

1. Thuật toán hierarchical.

Thuật toán phân cụm phân cấp (Hierarchical Clustering) sử dụng phương pháp hợp nhất từ dưới lên (Bottom-up Agglomerative) với các tiêu chí liên kết: single-linkage, complete-linkage, và average-linkage..

Thuật toán này hoạt động theo các bước sau:

- **Khởi tạo:** Ban đầu, mỗi điểm dữ liệu được coi là một cụm riêng lẻ.
- **Lặp lại:**
  - **Tìm cặp gần nhất:** Xác định hai cụm (clusters) gần nhau nhất dựa trên một tiêu chí liên kết (linkage criterion) đã chọn để đo khoảng cách giữa các cụm.
  - **Hợp nhất (Merge):** Kết hợp hai cụm gần nhất này thành một cụm mới.
  - **(Cập nhật ma trận khoảng cách):** (Ngầm định hoặc tường minh) Cập nhật lại khoảng cách giữa cụm mới vừa tạo với tất cả các cụm còn lại.
- **Kết thúc:** Quá trình này được lặp lại cho đến khi tất cả các điểm dữ liệu thuộc về một cụm duy nhất, hoặc cho đến khi đạt được một số lượng cụm mong muốn. Kết quả là một cấu trúc dạng cây gọi là dendrogram, thể hiện thứ bậc của các lần hợp nhất.

Các Tiêu Chí Liên Kết (Linkage Criteria) Phổ Biến:

Tiêu chí liên kết quyết định cách tính "khoảng cách" giữa hai cụm (ví dụ: Cụm A và Cụm B).

1. **Single-Linkage (Liên kết Đơn / Khoảng cách Tối thiểu):**
    - **Định nghĩa:** Khoảng cách giữa Cụm A và Cụm B được định nghĩa là khoảng cách nhỏ nhất giữa bất kỳ một điểm nào trong Cụm A và bất kỳ một điểm nào trong Cụm B.
    - **Cách hoạt động:** Thuật toán sẽ tìm hai điểm (một từ mỗi cụm) có khoảng cách Euclid (hoặc một thước đo khoảng cách khác) nhỏ nhất, và đó chính là khoảng cách giữa hai cụm.
    - **Đặc điểm:** Có xu hướng tạo ra các cụm dài, giống như "chuỗi". Đôi khi có thể kết nối các cụm riêng biệt nếu có một "cầu nối" gồm các điểm gần nhau giữa chúng (nhạy cảm với các điểm ngoại lệ làm cầu nối giữa các cụm).
2. **Complete-Linkage (Liên kết Đầy đủ / Khoảng cách Tối đa):**
    - **Định nghĩa:** Khoảng cách giữa Cụm A và Cụm B được định nghĩa là khoảng cách lớn nhất giữa bất kỳ một điểm nào trong Cụm A và bất kỳ một điểm nào trong Cụm B.
    - **Cách hoạt động:** Thuật toán sẽ tìm hai điểm (một từ mỗi cụm) có khoảng cách Euclid lớn nhất, và đó chính là khoảng cách giữa hai cụm.
    - **Đặc điểm:** Có xu hướng tạo ra các cụm đặc hơn, có dạng hình cầu. Nó đảm bảo rằng tất cả các điểm trong một cụm được hợp nhất đều tương đối gần nhau. Có thể nhạy cảm với các điểm ngoại lệ nếu chúng làm tăng đáng kể khoảng cách tối đa.
3. **Average-Linkage (Liên kết Trung bình / UPGMA - Unweighted Pair Group Method with Arithmetic Mean):**
    - **Định nghĩa:** Khoảng cách giữa Cụm A và Cụm B được định nghĩa là khoảng cách trung bình giữa tất cả các cặp điểm, trong đó một điểm thuộc Cụm A và điểm còn lại thuộc Cụm B.
    - **Cách hoạt động:** Tính khoảng cách Euclid giữa mọi cặp điểm (một từ A, một từ B), sau đó lấy trung bình của tất cả các khoảng cách này.
    - **Đặc điểm:** Thường được coi là một sự cân bằng giữa single-linkage và complete-linkage. Ít bị ảnh hưởng bởi các điểm ngoại lệ hơn so với single-linkage và có thể hình thành các cụm với nhiều hình dạng khác nhau, nhìn chung mạnh mẽ hơn single-linkage.

Tóm tắt cài đặt thuật toán phân cụm phân cấp dạng gom nhóm

1. **Lớp `AgglomerativeClusteringScratch`**:
   - **Khởi tạo (`__init__`)**: Nhận vào số cụm mong muốn (`n_clusters`) và phương pháp liên kết (`'single'`, `'complete'`, `'average'`).
   - **Logic chính (phương thức `fit`)**:
     (a) **Khởi tạo**: Mỗi điểm dữ liệu ban đầu tạo thành một cụm riêng biệt. Dữ liệu gốc được lưu lại.
     (b) **Gộp lặp**: Thuật toán thực hiện `n-1` lần gộp, với `n` là số lượng mẫu. Mỗi bước gồm:
       – **Tính khoảng cách**: Tính khoảng cách giữa tất cả các cặp cụm đang hoạt động, sử dụng phương pháp liên kết đã chọn:
         ∗ `single_linkage_dist_scratch`: Tìm khoảng cách Euclid nhỏ nhất giữa hai điểm trong hai cụm.
         ∗ `complete_linkage_dist_scratch`: Tìm khoảng cách Euclid lớn nhất giữa hai điểm trong hai cụm.
         ∗ `average_linkage_dist_scratch`: Tính khoảng cách Euclid trung bình giữa tất cả các cặp điểm, mỗi điểm từ một cụm (UPGMA).
       – **Tìm cặp gần nhất**: Xác định hai cụm có khoảng cách nhỏ nhất giữa các cụm. Cơ chế phá vỡ ràng buộc đơn giản (ưu tiên cụm có chỉ số gốc nhỏ hơn) được sử dụng.
       – **Gộp**: Hai cụm này được gộp thành một cụm mới.
       – **Cập nhật ma trận liên kết**: Chi tiết về lần gộp này (ID của các cụm được gộp, khoảng cách khi gộp, và số điểm trong cụm mới) được ghi lại trong `linkage_matrix_`. Ma trận này có định dạng tương tự với ma trận được tạo bởi `scipy.cluster.hierarchy.linkage`.
       – **Cập nhật trạng thái**: Danh sách các cụm đang hoạt động và ID của chúng được cập nhật cho vòng lặp tiếp theo.
   - **Gán nhãn (`labels_`)**: Nếu `n_clusters` được chỉ định, phương thức `fit` sử dụng `scipy.cluster.hierarchy.fcluster` trên `linkage_matrix_` để tạo ra nhãn cụm cuối cùng. Đây là phương pháp phổ biến để tạo cụm phẳng từ ma trận liên kết và đảm bảo tương thích với các thư viện. Các nhãn được chuyển thành chỉ số bắt đầu từ 0.
   - **Phương thức `fit_predict`**: Phương thức tiện lợi, gọi `fit` và trả về các nhãn cụm đã tính.
2. **Các hàm hỗ trợ**:
   - `euclidean_dist_manual`: Tính khoảng cách Euclid giữa hai điểm dữ liệu.
   - `get_cluster_points_from_data`: Truy xuất tọa độ các điểm dữ liệu thuộc về một cụm (dựa trên chỉ số gốc của chúng).

- Các hàm tính khoảng cách theo phương pháp liên kết (`single_linkage_dist_scratch`, v.v.) sử dụng các hàm hỗ trợ trên để thao tác với tập điểm trong mỗi cụm.

Cách hoạt động (Phương pháp gom nhóm từ dưới lên)

Thuật toán bắt đầu bằng cách xem mỗi điểm dữ liệu là một cụm riêng biệt. Trong mỗi bước, nó tìm hai cụm "gần nhau nhất" theo tiêu chí liên kết đã chọn và gộp chúng lại thành một cụm lớn hơn. Quá trình này lặp lại cho đến khi tất cả điểm dữ liệu thuộc về một cụm, hoặc đạt đến số cụm được xác định trước. Lịch sử của các lần gộp tạo thành một cấu trúc phân cấp, thường được biểu diễn dưới dạng cây phân cấp (dendrogram), và được lưu lại trong `linkage_matrix_`. Cài đặt "từ đầu" này tính toán trực tiếp khoảng cách liên kết dựa trên các điểm thành phần của cụm tại mỗi bước, thay vì sử dụng công thức cập nhật hiệu quả hơn như Lance-Williams (thường dùng trong các thư viện tối ưu).

so sánh

Đoạn mã còn bao gồm một hàm `run_and_compare` có chức năng:

- Tải dữ liệu (từ `data.csv` hoặc mẫu có sẵn).
- Với mỗi phương pháp liên kết (single, complete, average):
  - Chạy mô hình `AgglomerativeClusteringScratch`.
  - Chạy `scipy.cluster.hierarchy.linkage` để lấy ma trận liên kết từ thư viện.
  - So sánh khoảng cách trong hai ma trận liên kết.
  - Chạy `sklearn.cluster.AgglomerativeClustering` để lấy nhãn cụm từ thư viện.
  - So sánh nhãn từ mô hình viết tay với nhãn từ Sklearn bằng Chỉ số Rand điều chỉnh (Adjusted Rand Index).
  - (Tuỳ chọn) Vẽ cây phân cấp và biểu đồ phân cụm.

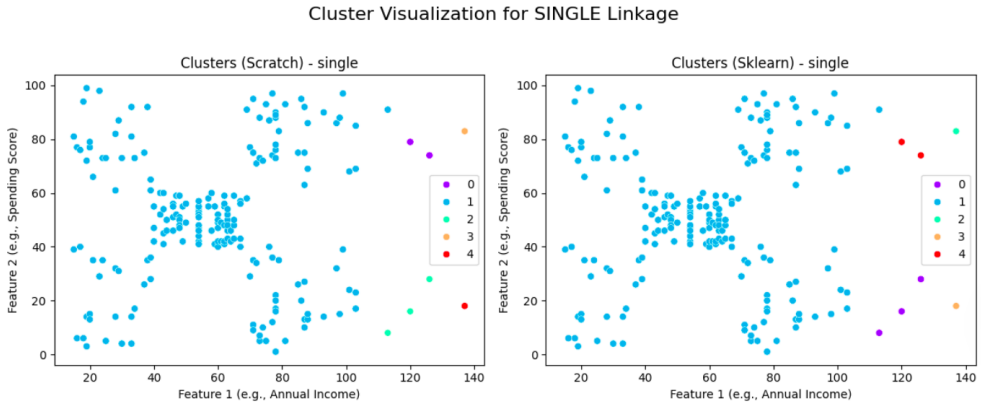
Kết quả so sánh:

```
--- Testing Linkage Method: SINGLE ---

1. From Scratch Implementation:
Linkage Matrix (Scratch) Z_scratch (first 5 rows if large):
     id1    id2  distance  num_points
0   48.0   49.0       0.0         2.0
1   65.0   68.0       0.0         2.0
2  129.0  131.0       0.0         2.0
3  156.0  158.0       0.0         2.0
4   21.0   23.0       1.0         2.0
Cluster Labels (Scratch) (first 20): [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Number of unique labels (Scratch): 5

2. Scipy Implementation (for Linkage Matrix):
Linkage Matrix (Scipy) Z_scipy (first 5 rows if large):
     id1    id2  distance  num_points
0   48.0   49.0       0.0         2.0
1   65.0   68.0       0.0         2.0
2  156.0  158.0       0.0         2.0
3  129.0  131.0       0.0         2.0
4   67.0   69.0       1.0         2.0
Max absolute difference in distances between Z_scratch and Z_scipy: 0.0000
Mean absolute difference in distances: 0.0000
Distances in linkage matrices are very close.
Cluster Labels (from Z_scipy) (first 20): [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

3. Sklearn Implementation (for Cluster Labels):
Cluster Labels (Sklearn) (first 20): [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Number of unique labels (Sklearn): 5
Adjusted Rand Index (Scratch vs Sklearn labels): 1.0000
Cluster labels from Scratch and Sklearn are highly similar.
Skipping dendrogram plotting for larger dataset to maintain clarity.
```



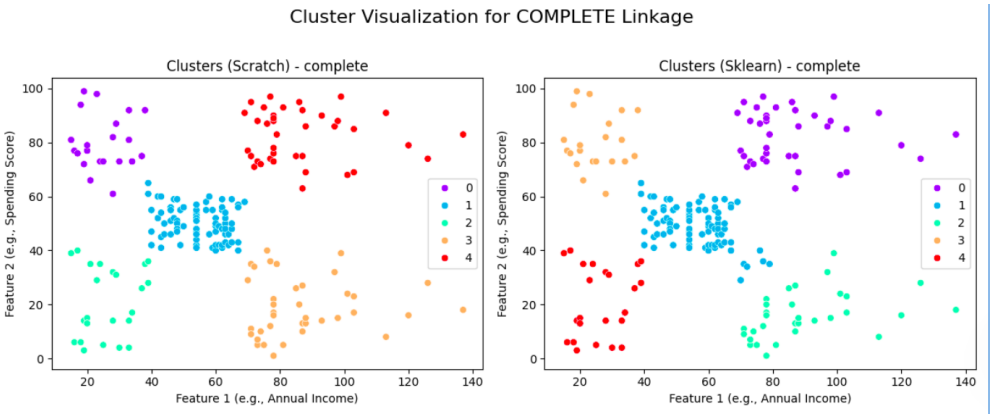
Kết quả so sánh:

```
--- Testing Linkage Method: COMPLETE ---

1. From Scratch Implementation:
Linkage Matrix (Scratch) Z_scratch (first 5 rows if large):
     id1    id2  distance  num_points
0   48.0   49.0       0.0         2.0
1   65.0   68.0       0.0         2.0
2  129.0  131.0       0.0         2.0
3  156.0  158.0       0.0         2.0
4   21.0   23.0       1.0         2.0
Cluster Labels (Scratch) (first 20): [2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0]
Number of unique labels (Scratch): 5

2. Scipy Implementation (for Linkage Matrix):
Linkage Matrix (Scipy) Z_scipy (first 5 rows if large):
     id1    id2  distance  num_points
0   65.0   68.0       0.0         2.0
1   48.0   49.0       0.0         2.0
2  129.0  131.0       0.0         2.0
3  156.0  158.0       0.0         2.0
4   21.0   23.0       1.0         2.0
Max absolute difference in distances between Z_scratch and Z_scipy: 4.3820
Mean absolute difference in distances: 0.1869
Distances in linkage matrices have some differences (expected due to precision/tie-breaking).
Cluster Labels (from Z_scipy) (first 20): [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]

3. Sklearn Implementation (for Cluster Labels):
Cluster Labels (Sklearn) (first 20): [4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3]
Number of unique labels (Sklearn): 5
Adjusted Rand Index (Scratch vs Sklearn labels): 0.9126
Cluster labels from Scratch and Sklearn show some differences.
Skipping dendrogram plotting for larger dataset to maintain clarity.
```



Kết quả so sánh:

```
--- Testing Linkage Method: AVERAGE ---

1. From Scratch Implementation:
Linkage Matrix (Scratch) Z_scratch (first 5 rows if large):
     id1    id2  distance  num_points
0   48.0   49.0       0.0         2.0
1   65.0   68.0       0.0         2.0
2  129.0  131.0       0.0         2.0
3  156.0  158.0       0.0         2.0
4   21.0   23.0       1.0         2.0
Cluster Labels (Scratch) (first 20): [4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3]
Number of unique labels (Scratch): 5

2. Scipy Implementation (for Linkage Matrix):
Linkage Matrix (Scipy) Z_scipy (first 5 rows if large):
     id1    id2  distance  num_points
0   65.0   68.0       0.0         2.0
1   48.0   49.0       0.0         2.0
2  156.0  158.0       0.0         2.0
3  129.0  131.0       0.0         2.0
4   21.0   23.0       1.0         2.0
Max absolute difference in distances between Z_scratch and Z_scipy: 0.4842
Mean absolute difference in distances: 0.0215
Distances in linkage matrices have some differences (expected due to precision/tie-breaking).
Cluster Labels (from Z_scipy) (first 20): [4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3]

3. Sklearn Implementation (for Cluster Labels):
Cluster Labels (Sklearn) (first 20): [1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3]
Number of unique labels (Sklearn): 5
Adjusted Rand Index (Scratch vs Sklearn labels): 1.0000
Cluster labels from Scratch and Sklearn are highly similar.
Skipping dendrogram plotting for larger dataset to maintain clarity.
```