Cài đặt Clustering Using REpresentatives (CURE)

Đầu tiên, đọc dữ liệu từ df_embedding_user được viết xuống file json ở Câu 1.

Cài đặt class CURE, hàm fit class này thực hiện 2 pass như mô tả của thuật toán CURE. Ở **Pass 1**, các bước đều được xử lý ở dạng in-memory, đầu tiên, thực hiện lấy một lượng nhỏ mẫu dữ liệu, mặc định lấy 25% lượng dataset *sample_fraction=0.25*. Bước tiếp theo, sử dụng module sklearn.cluster dùng AgglomerativeClustering để thực hiện *hierarchical clustering* với mặc định số lượng cluster *n_cluster=5*, từ đây, ta có được list các cluster_id (cluster_labels) tương ứng cho mỗi dòng dữ liệu của ta, vì thế ta có thể dễ dàng tạo được một dict với key là cluster_id và value là list các point (embedding, list của 32 số thực). Tiếp theo, từ dict vừa mới tạo thành, ta duyệt qua từ cặp cluster_id và các point thuộc cluster đó, lần lượt tính:

1. Centroid, chính là điểm có giá trị trung bình của các point;
2. Các representative (reps) thuộc cluster đó bằng phương pháp **Maximize minimum distances**, với mỗi point trong cluster, tính khoảng cách ngắn nhất (phép đo Euclidean distance) của nó đến các reps (rep đầu tiên sẽ là centroid, kết quả chọn các rep cuối cùng sẽ không xét centorid), chọn ra khoảng cách ngắn nhất, lưu nó vào một list theo dạng tuple (point, distance), làm tương tự cho các point còn lại, cuối cùng từ list mới vừa tạo, chọn ra point có distance lớn nhất, phương pháp này đảm bảo các point sẽ nằm cách xa nhau, tuy nhiên một số trường hợp rep có thể trở thành outliner;

3. Di chuyển hay "co" các rep lại về centroid một khoảng, mặc định lấy 20% khoảng cách từ rep đến centroid theo công thức:
    4. $new\_rep = rep + shrink\_factor \times (rep - centroid)$
5. Lưu các kết quả vừa tính được vào một list các dict, dict này đại diện cho thông tin của một cluster, gồm cluster_id, centroid, points, reps (đã được co lại);

Cuối cùng, ta merge các cluster có khoảng cách của 2 điểm rep bất kỳ, mỗi điểm thuộc mỗi cluser, nếu khoảng cách này **bé hơn hoặc bằng** *threshold_merge*, ta merge 2 cluster lại bằng cách gom chung các rep của 2 cluster vào 1 list. Cách cài đặt merge cluster của chúng tôi chỉ cần qua 1 nested loop là có thể xác định các cluster nào cần merge, ví dụ, qua nested loop, xác định merge cluser (0, 1) và (1, 2), thì cuối cùng sẽ merge (0, 1, 2) thành một cluster mới. Các cluster được merge và cả không được merge đều được đánh cluster_id lại, bắt đầu 0. Kết quả sẽ lưu thành một list các dict, với dict đó key là cluster_id, value là các rep đã được gom chung khi merge.

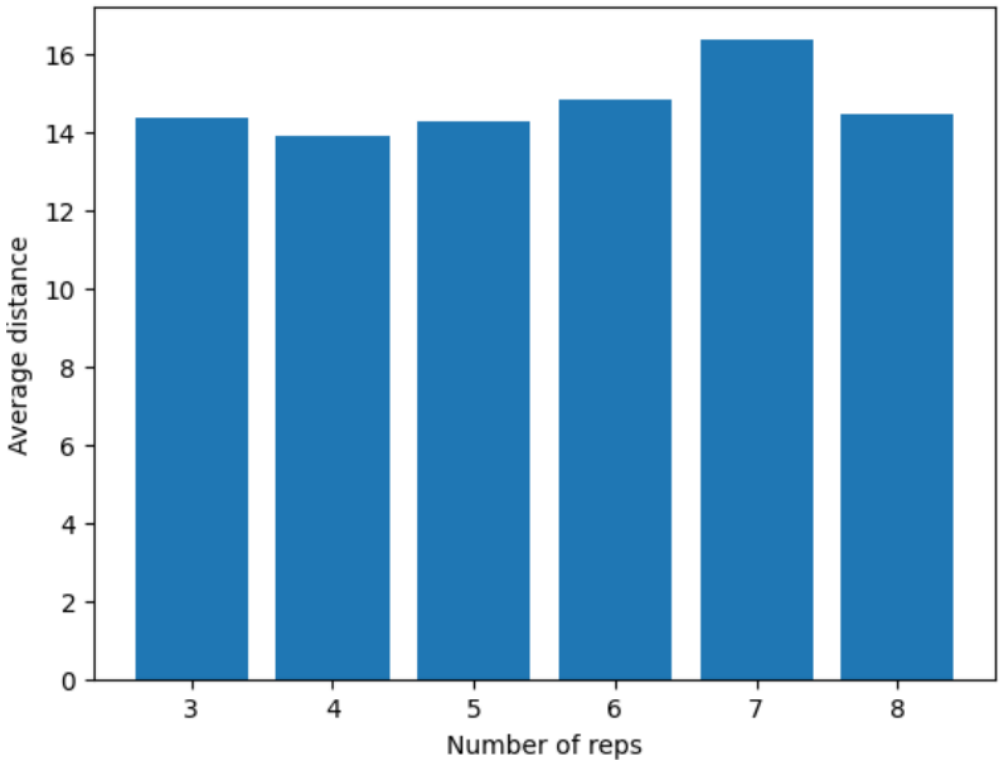
Ở **Pass 2**, từ list các dict vừa mới tìm được, viết một user-defined function để tìm rep có khoảng cách ngắn nhất tới point đang xét, bằng cách với mỗi cluster, tìm khoảng cách ngắn nhất từ point đến rep trong cluser đó, rồi lại tìm xem trong số các khoảng cách đó tìm khoảng cách ngăn nhất (**Minimize minimum distances**). Áp dụng user-defined function vừa nãy cho dataset gốc, ta hình thành một DataFrame với mỗi dòng chứa thông tin cluster id cho point đang xét - **cluster_id**, point đang xét - **point** và representative gần nhất với điểm đang xét và cũng thuộc cluster id – **rep.**

```
+----------+--------------------+--------------------+
|cluster_id|               point|                 rep|
+----------+--------------------+--------------------+
|         4|[-16.796118656868...|[-13.595301929171...|
|         0|[-0.1880540258735...|[0.57158091518525...|
|         0|[-2.3890451791994...|[-2.0696084688059...|
|         0|[-1.9212580037324...|[0.57158091518525...|
|         0|[-4.3522241197278...|[-4.4254231974399...|
|         4|[-9.0957539728547...|[-7.4388935529355...|
|         0|[-4.3722830580566...|[0.57158091518525...|
|         0|[-0.9598459091748...|[0.57158091518525...|
|         1|[-28.912229574723...|[-28.912229574723...|
|         0|[-8.7604340796836...|[0.17583138119460...|
+----------+--------------------+--------------------+
only showing top 10 rows
```

Caption: Kết quả cuối cùng của thuật toán CURE

Biểu đồ cột trung bình khoảng cách

Dựa vào DataFrame trên, chúng tôi cài đặt một user-defined function tính khoảng cách từ point đến rep với cho mỗi dòng dữ liệu, sau đó lấy trung bình tổng các khoảng cách vừa mới tìm được.



Caption: iểu đồ cột thể hiện trung bình khoảng cách từ point đang xét đến rep tương ứng của nó từ bảng DataFrame trên với số lượng rep, n_reps trong đoạn 3 đến 9.