Lecture 5: DBSCAN

I. Motivation

- Ta đã học về các thuật toán phân cụm như kmeans hay GMM, tuy nhiên với những bài toán mà data có hình dạng, kích thước lạ, k° theo blob, hoặc có nhiều nhiễu thì các thuật toán này có thể k° work. For example:

II. Introduction

- DBSCAN là một thuật toán phân cụm dựa trên mật độ (density-based clustering), đây là một phương pháp học k⁰ giám sát nhằm xác định cụm phân biệt trong phân phối của dữ liệu, dựa trên ý tưởng rằng một cụm trong k⁰ gian dữ liệu là một vùng có mật độ điểm cao được ngăn cách vs các cụm ≠ bằng các vùng liền kề có mật độ điểm thấp.

III. Basic concepts

Một số định nghĩa mà thuật toán DBSCAN sử dụng:

1. Vùng lân cận Epsilon (Epsilon neighborhood): Eps-neighborhood của một điểm dữ liệu P là tập hợp tất cả các điểm nằm trong phạm vi bán kính ϵ xung quanh điểm P. Kí hiệu tập hợp những điểm này là

$$N_{eps}(P) = \{Q \in D: d(P,Q) \leq \epsilon\}$$

Trong đó D là tập hợp các điểm nằm trong training set.

2. Directly density-reachable Khả năng tiếp cận trực tiếp mật độ để cập tới việc một điểm Q directly-density reachable đến P nếu nó nằm trong neighborhood của P, điều kiện để thỏa mãn điều này là:

+) Q nằm trong vùng lân cận Epsilon của P: $Q \in N_{eps}(P)$

+) Số lượng các điểm nằm trong vùng lân cận Epsilon tối thiểu là minpts ⩾ minPts (core point condition) $|N_\epsilon(Q)| \geq minPts$

Ý nghĩa: Một điểm Q ✓⁰ directly density reachable với điểm P sẽ dựa vào 2 yếu tố đó là khoảng cách giữa chúng và mật độ các điểm trong ϵ neighborhood của Q phải có tối thiểu minPts điểm (với minPts là tham số).

3. Density reachable (khả năng tiếp cận mật độ) liên quan đến cách hình thành một chuỗi điểm trong cụm. Trong một tập hợp chuỗi điểm $\{P_i\}_{i=1}^{n}$

mà bất kỳ một điểm P_i nào cũng directly density reachable bởi P_{i-1} (i.e P_i nằm trong neighborhood ε của P_{i-1}), khi đó ta nói điểm $P = P_n$ là density reachable với điểm $Q = P_1$ (dù hai điểm này ko nằm chung neighborhood nhưng vẫn nằm chung 1 cụm qua sự liên kết về density của 1 chuỗi điểm).

→ Hai điểm bất kỳ P_i và P_j thuộc $\{P_i\}_{i=1}^{n}$ là density reachable vs nhau.

→ Các điểm $\{P_i\}_{i=1}^{n}$ đều sẽ được phân về 1 cụm.

→ Thể hiện sự mở rộng phạm vi của 1 cụm dựa theo liên kết chuỗi.

4. Density-connected : A point p is density connected to a point q if there is a point k such that both p and q are density-reachable from k.

IV, Tham số và 3 loại điểm trong DBSCAN

1) DBSCAN có 2 tham số:
- ε (một số nơi kí hiệu là E): maximum radius of the neighborhood ✓
- minPts: minimum number of point in a neighbh (ko bao gồm điểm ở tâm)

2) 3 loại điểm trong DBSCAN:
- A point is a core point if it has more than a specified number of points (minPts) within neighborhood with radius ε.
- A border point has fewer than minPts points in its neighborh but is in the neighbor hood of core point.
- A noise point is any point that is not a core point or a border point ✓.

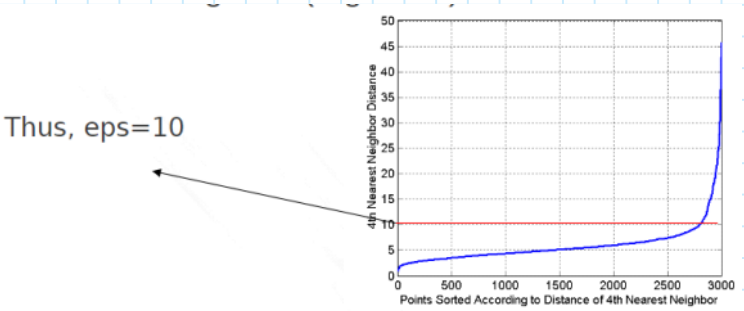
3) Ảnh hưởng của tham số lên performance.
- Eps: larger ε → fewer clusters and smaller ε → more clusters. Nếu ε quá lớn thì nhiều cluster có thể bị gộp lại; trong khi đó ε quá bé khiến tách rời nhiều cluster lớn và có nhiều noise point hơn.
- minPts: higher minPts → neighborhood rộng hơn ⇒ ít cluster hơn và ngược lại (tác động giống ε)

4) Cách chọn ε và minPts

- Với minPts: thông thường giá trị minPts sẽ có điều kiện minPts ≥ D+1 với D là số chiều của dữ liệu. Tuy nhiên các giá trị lớn hơn thường tốt hơn cho các tập dữ liệu có nhiễu và cũng cho kqua phân cụm hợp lý hơn. Gtri thường chọn là minPts = 2 × D. Nếu dữ liệu có nhiều nhiễu hoặc nhiều quan sát lặp nhau thì nên tăng minPts.
- Với ε: Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance. Noise points thì sẽ có k/c xa hơn hẳn các điểm neighbors, vậy với mỗi điểm ta sẽ chọn ra k điểm

gần nhất đến nó, và chọn k/cách lớn nhất trong k khoảng cách đó sau đó ta sắp xếp các k-neighbor distance đó tăng dần rồi plot ra, khi giá trị distance tăng đột biến ta sẽ chọn k/c đó là ε.

Example:



V, DBSCAN steps

- Let clustercount = 0. For every point p:
  - +) If p is not a core point, assign null label to it.
  - +) If p is a core point, a new cluster is formed. (with cluster count + = 1). Then find all points density reachable to p and classify them in the cluster. (Điểm nào có null label trước đó mà cũng density-reachable thì reassign label cho điểm đó).
  - +) Repeat the process until all points are visited.

VI, DBSCAN: Flaws

- DBSCAN can cluster badly on:
  - +) Varying density data (vì DBSCAN là density-based nên vd nó cluster có density quá loãng sẽ bị phân thành nhiều cluster nhỏ hoặc bị tính là noise)
  - +) High dim data : density bị vary ở các chiều data ≠ nhau.

VI, Advantage and disadvantage of DBSCAN.

1) Advantage

- **Handles irregularly shaped and sized clusters**. One of the main advantages of DBSCAN is its ability to detect clusters that are irregularly shaped. Of all the common clustering algorithms out there, DBSCAN is one of the algorithms that makes the fewest assumptions about the shape of your clusters. That means that DBSCAN can be used to detect clusters that are oddly or irregularly shaped, such as clusters that are ring-shaped.
- **Robust to outliers**. Another big advantage of DBSCAN is that it is able to detect outliers and exclude them from the clusters entirely. That means that DBSCAN is very robust to outliers and great for datasets with multiple outliers.
- **Does not require the number of clusters to be specified**. Yet another advantage of DBSCAN is that it does not require the user to specify the number of clusters. Instead, DBSCAN can automatically detect the number of clusters that exist in the data. This is great for cases where you do not have much intuition on how many clusters there should be.

Advantage (cont.)

- **Less sensitive to initialization conditions**. DBSCAN is less sensitive to initialization conditions like the order of the observations in the dataset and the seed that is used than some other clustering algorithms. Some points that are on the borders between clusters may shift around when initialization conditions change, but the majority of the observations should remain in the same cluster.
- **Relatively fast**. While DBSCAN is not the fastest clustering algorithm out there, it is certainly not the slowest either. There are multiple implementations of DBSCAN that aim to optimize the time complexity of the algorithm. DBSCAN is generally slower than **k-means** clustering but faster than **hierarchical clustering** and spectral clustering.

⊗ Disadvantages :

- **Difficult to incorporate categorical features**. One of the main disadvantages of DBSCAN is that it does not perform well on datasets with categorical features. That means that you are best off using DBSCAN in cases where most of your features are numeric.
- **Requires a drop in density to detect cluster borders**. With DBSCAN, there must be a drop in the density of the data points between clusters in order for the algorithm to be able to detect the boundaries between clusters. If there are multiple clusters that are overlapping without a drop in data density between them, they may get grouped into a single cluster.
- **Struggles with clusters of varying density**. DBSAN also has a difficulty detecting clusters of varying density. This is because DBSCAN determines where clusters start and stop by looking at places where the density of data points drops below a certain threshold. It may be difficult to find a threshold that captures all of the points in the less dense cluster without excluding too many extraneous outliers in the more dense cluster.
- **Sensitive to scale**. Like many other clustering algorithms, DBSCAN is sensitive to the scale of your variables. That means that you may need to rescale your variables if they are on very different scales.
- **Struggles with high dimensional data**. Like many clustering algorithms, the performance of DBSCAN tends to degrade in situations where there are many features. In general, you are better off using dimensionality reduction or features selection techniques to reduce the number of features if you have a high-dimensional dataset.
- **Not as well known**. Another disadvantage of DBSCAN is that it is not as popular and well-studied as other clustering algorithms like k-means clustering and hierarchical clustering. It may not be as easy for collaborators that are not familiar with the algorithm to contribute to a project that uses DBSCAN.