DBSCAN

Saturday, December 7, 2019 19:28

For the glory of God

What is DBSCAN ?

- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester in 1996.
- DBSCAN is one of the most popular unsupervised machine learning techniques that still remains as one of the highest cited Science papers.
- The DBSCAN algorithm is based on the intuitive notion of clusters and noises.
  - The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
  - In other words, the main concept is to locate regions of high density that are separated from one another by regions of low density.
  - In short, given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions whose nearest neighbors are too far away.

Why DBSCAN ?

- Fundamentally, all clustering methods use the same approach, i.e. calculate similarities and use it to cluster the data points.
- Then, what is the difference between DBSCAN and other clustering methods ?
- The authors (Martin ester et al) mentioned the following reasons in their paper ;
  - DBSCAN requires minimum domain knowledge.
  - DBSCAN can discover clusters of arbitrary shape.
  - DBSCAN is efficient for large database.
- In other words,
  - The other clustering algorithms (e.g. K-means and EM) are suitable only for compact and well-separated clusters.
  - They also require users to specify the number of clusters before users run the algorithm.
  - They are sometimes prone to noisy data points.
  - However, real life data may contain noise as well as irregularity.
  - The key concept of DBSCAN actually helps us to identify the concerns mentioned above.

How does the DBSCAN algorithm work ?

· The DBSCAN algorithm can be described in the following steps ;

· Step 1 :

- The algorithm starts with an arbitrary point which has not been visited.

· Step 2 :

- Its neighborhood information is retrieved from the ε parameter.

↳ This is one of hyperparameters for the DBSCAN algorithm.

- Here ;

§ If the distance between two points $\leq$ eps (ε), they are considered as neighbors.

If the distance between two points $\geq$ eps (ε), they are not considered as neighbors.

Note !

· If eps is chosen too small, then large part of the data will be considered as outliers.

· If eps is chosen too large, the clusters will merge and majority of the data points will be in the same clusters.

⇒ One way to find the optimum eps value is based on the k-distance graph.

- Then, what is the K-distance graph ? The idea is described as follows ;

1) Calculate the distance to the nearest K points for each data point (e.g. K=5, from the point, find 5 nearest point)

↳ Here, K corresponds to MinPts in the DBSCAN algorithm. ↳ And calculate the average of distances

↳ This is another hyperparameter for the DBSCAN algorithm.

2) Sort the distance results in an ascending order

↳ Make sure to normalize the values $[0,1]$

§ Larger the dataset, the larger value must be chosen.

The minimum value = 3

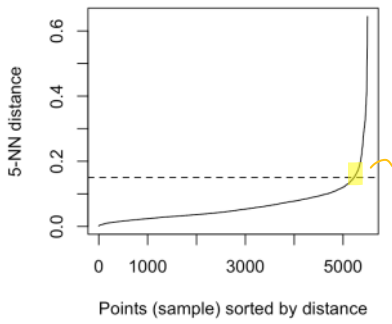
Rule of thumb : MinPts $\geq$ Dimension + 1

↓

In practice, $\ln(n)$

; n = # of data points

3) Plot the results such as ;



X axis : Number of data points

Y axis : Normalize distance results

↳ The slope is extremely changed.

4) Look to see where the change is most pronounced

5) Select it as the optimal epsilon

· Step 3 :

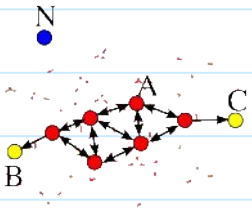
- If there are a sufficient number of points (according to MinPts) within ε neighborhood, cluster formation starts.

  ↳ When the process starts, the current data point becomes the first point in the new cluster.

- Otherwise, the point is labeled as a noise point.

  ↳ Note that this point can be considered as a cluster of different point.

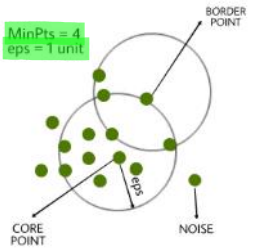
e.g.



A : Core point

B, C : Border point

N : Noise point



- As can be seen, the DBSCAN algorithm contains three different types of data points.

  Core point : A point is defined as a core if it has more than MinPts points within ε.

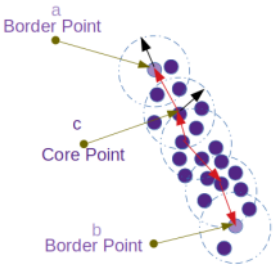
  Border point : A point is defined as a border if it has fewer than MinPts within ε but it's in the neighborhood of a core point

  Noise point : A point is defined as a noise if it is neither core nor border point. (Not assigned to a cluster)

· Step 4 :

- From the first point in the new cluster (step 3), add all points within ε neighborhood of the first point into the same cluster.

- This is then repeated for all of the new points that have been just added to the cluster group.



↑ represents the recursive process

a, b are Density Reachable from a core point c.

a, b are called Density Connected points.

↳ Actually, the process is repeated until the density connected cluster is completely found.

- The recursive process is repeated until all points in the cluster are determined.

  ↳ In other words, all points of the cluster have been visited and labeled.

· Step 5 :

- Once we are done with the current cluster, a new unvisited point is selected and processed.

  (This is also repeated until all points of the cluster are marked as visited)

What is Pseudocode for the DBSCAN algorithm ?

· The pseudocode can be expressed as follows: (Time complexity = $O(n^2)$)

```
DBSCAN(DB, distFunc, eps, minPts) {
   C = 0                                                  /* Cluster counter */
   for each point P in database DB {
      if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
      Neighbors N = RangeQuery(DB, distFunc, P, eps)      /* Find neighbors */
      if |N| < minPts then {                              /* Density check */
         label(P) = Noise                                 /* Label as Noise */
         continue
      }
      C = C + 1                                           /* next cluster label */
      label(P) = C                                        /* Label initial point */
      Seed set S = N \ {P}                                /* Neighbors to expand */
      for each point Q in S {                             /* Process every seed point */
         if label(Q) = Noise then label(Q) = C            /* Change Noise to border point */
         if label(Q) ≠ undefined then continue            /* Previously processed */
         label(Q) = C                                     /* Label neighbor */
         Neighbors N = RangeQuery(DB, distFunc, Q, eps)   /* Find neighbors */
         if |N| ≥ minPts then {                           /* Density check */
            S = S ∪ N                                     /* Add new neighbors to seed set */
         }
      }
   }
}
```

What are advantages and disadvantages in terms of the DBSCAN algorithm?

a) Advantages

· It does not require one to specify the number of clusters in the data a priori.

· It can find arbitrarily shaped clusters.

· It is robust to outliers because it has a notion of noise.

· It only asks one to specify two hyperparameters (i.e. ϵ and MinPts)

b) Disadvantages

· It does not work well when dealing with clusters of varying density.

· It usually struggles with high-dimensional datasets. (i.e. curse of dimensionality)

· If the data and scale are not well understood, choosing a meaningful distance threshold ϵ can be difficult.