DBSCAN

Epoch IIT Hyderabad

Ankit Saha
AI21BTECH11004

30 Jul 2022

1. Introduction

Density-based spatial clustering of applications with noise (DBSCAN) is an unsupervised machine learning algorithm that is used for clustering. As the name suggests, it clusters points based on the densities of points around them. Clusters are closely packed. Points with low densities around them are marked as outliers.

DBSCAN is one of the most popular clustering algorithms. It is especially useful when the training data consists of nested clusters, where algorithms like k-means will not work because the centroids of nested clusters can be very close to each other.

DBSCAN works so well because its clustering principle is consistent with the regular human conception of clusters. If a human had to identify clusters given an arbitrary set of data points, they would do it on the basis of how dense points in a region are. Thus, DBSCAN mimics the human brain when trying to identify clusters.

2. Algorithm

A point is said to be a core point if its ϵ-neighbourhood contains at least δ points where ϵ and δ are both user-defined parameters. A core point is assigned to the first cluster. The core points within the ϵ-neighbourhood of the previous core point are added to the first cluster. All the core points close to the growing first cluster are iteratively added to the first cluster. Finally, we add to the cluster the non-core points that are within the ϵ-neighbourhood of any of the core points in the cluster. At this point, the iteration stops and the non-core points are not used to extend the cluster any further.

Other clusters are formed in a similar fashion. A core point that has not been assigned to a cluster is used to create a new cluster. This process is repeated until all core points are assigned to some cluster. After all clusters have been made, non-core points that are not in any cluster are marked as outliers.

3. Mathematical Formulation

Suppose we are given m samples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ where $x_i \in \mathbb{R}^n \ \forall i \in [m]$

Let $d(\mathbf{x}_i, \mathbf{x}_j)$ be a metric that denotes the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$

A point $\mathbf{x}_i$ is said to be a core point if and only if $|B_\epsilon[\mathbf{x}_i] \cap \mathcal{X}| \geq \delta$

A point $\mathbf{x}_j$ is said to be directly reachable from a core point $\mathbf{x}_i$ if and only if $d(\mathbf{x}_i, \mathbf{x}_j) \leq \epsilon$

A point $\mathbf{x}_j$ is said to be reachable from a core point $\mathbf{x}_i$ if and only if there exists a sequence of core points $\{\mathbf{x}_i, \ldots, \mathbf{x}_k\}$ where each point is directly reachable from the previous and $\mathbf{x}_j$ is directly reachable from $\mathbf{x}_k$. Thus, reachability is the transitive closure of direct reachability

Two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are said to be densely-connected if there exists a point $\mathbf{x}_k$ such that both $\mathbf{x}_i$ and $\mathbf{x}_j$ are reachable from $\mathbf{x}_k$

A cluster is thus defined as a set of mutually densely-connected points such that if a point is reachable from a point in the cluster, then it is a part of the cluster as well.

4. Questions

i) What is a major advantage of DBSCAN over k-means clustering?
ii) What is a drawback of DBSCAN?
iii) What is the time complexity of DBSCAN?
iv) What is the space complexity of DBSCAN?
v) Give a reasonably good value of δ to be used as a parameter.

5. Answers

i) Unlike k-means, we do not have to specify the number of clusters in DBSCAN ourselves. The model computes it directly.
ii) DBSCAN does not work very well when there is a significant variation in densities over the entire dataset. A single combination of ϵ and δ will not be appropriate for all clusters. This also makes the detection of outliers harder.
iii) Iterating over every point in the dataset takes $O(m)$ time.
For every point, we must also compute its ϵ-neighbours. If this is done the naive way, i.e., iterating over all other points and checking if the distance between them is less than or equal to δ, then for each point it takes $O(m)$ time too. Thus, the overall time complexity will be $O(m^2)$
However, if an efficient method like an indexing structure is used that can find a point's ϵ-neighbours in $O(\log m)$ time, then the overall time complexity will be $O(m \log m)$
iv) If the ϵ-neighbours of a point are being computed spontaneously, then no extra memory other than that needed to store the clusters is required. Thus, the space complexity will be $O(m)$
However, if a distance matrix is used to store the distances between each pair of points to avoid recomputation of distances every time which saves time, then $O(m^2)$ memory will be required.
v) $\delta = 2n$ is a reasonably good value of δ that is commonly used in practice.