DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

The DBSCAN algorithm identifies clusters of high density and marks points in low-density regions as noise. It has two parameters: ϵ, which specifies the maximum distance between two points for them to be considered as neighbors, and MinPts, which specifies the minimum number of points required to form a dense region.

DBSCAN Algorithm:

RegionQuery Function:

The `RegionQuery`(p, ϵ) function finds all points within a distance ϵ from the point p. It returns the set of neighbors of p.

Algorithm 1 DBSCAN Algorithm

```
Input: A set of points $P$, a distance function Dist$(p, q)$, a distance threshold $\epsilon$, and a minimum number of points MinPts.
Output: A set of clusters $C$.
Initialize all points in $P$ as unvisited.
for each point $p \in P$ do
    if point $p$ is unvisited then
        Mark $p$ as visited.
        $N(p) \leftarrow$ RegionQuery$(p, \epsilon)$
        if size of $N(p)$ ¡ MinPts then
            Mark $p$ as noise.
        else
            Create a new cluster $C$ and add $p$ to it.
            for each point $q \in N(p)$ do
                if point $q$ is unvisited then
                    Mark $q$ as visited.
                    $N(q) \leftarrow$ RegionQuery$(q, \epsilon)$
                end if
                if size of $N(q) \geq$ MinPts then
                    Add $q$ to $N(p)$ (expand the cluster).
                end if
            end for
            for each point $q \in N(p)$ do
                if point $q$ is not in any cluster then
                    Add $q$ to the cluster $C$.
                end if
            end for
        end if
    end if
end for
return clusters $C$.
```

Algorithm 2 RegionQuery function

```
Input: A point $p$, and a threshold $\epsilon$.
Output: A list of points within distance $\epsilon$ of $p$.
Initialize an empty set $N(p)$.
for each point $q \in P$ do
    if Dist$(p, q) \leq \epsilon$ then
        Add $q$ to $N(p)$.
    end if
end for
return $N(p)$.
```