Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India *

March 2025

DBSCAN Clustering

Introduction

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised clustering algorithm that identifies dense regions in data. Unlike k-means or hierarchical clustering, DBSCAN does not require the number of clusters to be specified beforehand and efficiently handles noise.

Key Aspects of DBSCAN

1. Core Concepts

DBSCAN defines clusters based on data density using two primary parameters:

- **Epsilon (ε)**: The maximum radius within which points are considered neighbors.
- **MinPts**: The minimum number of points required to form a dense region.

Each data point is classified as one of the following:

- **Core Point**: A point with at least *MinPts* neighbors within the ε radius.
- **Border Point**: A point that is within the ε radius of a core point but has fewer than *MinPts* neighbors itself.
- **Noise (Outlier)**: A point that does not satisfy either condition.

2. Algorithm Steps

The DBSCAN algorithm follows these steps:

1. Mark all points as unvisited.
2. Select an unvisited point p. If p has at least *MinPts* neighbors within ε, mark it as a **core point** and form a cluster.

*Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

3. Expand the cluster by adding all density-reachable points (points that are within ε distance of core points).
4. Repeat until no new points can be added.
5. Points that are neither core nor border points are marked as **noise**.

3. Key Properties

DBSCAN exhibits several important properties:

- **Robust to Outliers:** DBSCAN effectively identifies and separates noise points.
- **Arbitrary Shape Detection:** Since DBSCAN groups points based on density, it can detect non-linear clusters effectively.
- **No Need to Specify Cluster Count:** Unlike k-means, DBSCAN automatically determines the number of clusters.

4. Parameter Sensitivity

While DBSCAN is robust, parameter tuning is crucial:

- **Choosing ε**: A small ε may lead to many noise points, while a large ε may merge distinct clusters.
- **Choosing *MinPts***: A higher *MinPts* value requires denser clusters, reducing noise sensitivity.

5. Complexity Analysis

The time complexity of DBSCAN is:

$$\mathcal{O}(n \log n)$$

where n is the number of data points. This makes DBSCAN efficient for large datasets.

DBSCAN Algorithm

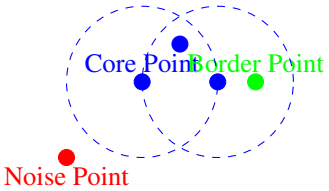
Algorithm 1 DBSCAN Clustering Algorithm

Require: Dataset D, neighborhood radius ε, minimum points $MinPts$
Ensure: Cluster labels for all points in D

Initialize cluster label $C = 0$
for each unvisited point $p \in D$ **do**
  Mark p as visited
  Find neighbors $N = \{q \in D \mid dist(p,q) \leq \varepsilon\}$
  if $|N| < MinPts$ **then**
    Mark p as noise
  else
    Increment cluster label: $C = C + 1$
    Create a new cluster and add p to C
    for each point $q \in N$ **do**
      if q is unvisited **then**
        Mark q as visited
        Find N' = Neighbors of q
        if $|N'| \geq MinPts$ **then**
          $N = N \cup N'$
        end if
      end if
      if q is not yet in any cluster **then**
        Add q to cluster C
      end if
    end for
  end if
end for=0

DBSCAN Visualization

Below is a sample visualization demonstrating core points, border points, and noise points.



DBSCAN is a powerful clustering algorithm that efficiently identifies patterns and outliers in complex datasets. By tuning the parameters ε and $MinPts$, you can optimize cluster detection based on data density.

Step 1: Dummy Data and Clustering

Consider 5 data points clustered by DBSCAN into two clusters:

$$\text{Cluster 1: } \{x_1, x_2, x_3\} \quad \text{Cluster 2: } \{x_4, x_5\}$$

Assume pairwise distances between points are as follows:

$$\begin{bmatrix} 0 & 0.5 & 0.6 & 1.5 & 1.6 \\ 0.5 & 0 & 0.4 & 1.7 & 1.8 \\ 0.6 & 0.4 & 0 & 1.8 & 1.9 \\ 1.5 & 1.7 & 1.8 & 0 & 0.3 \\ 1.6 & 1.8 & 1.9 & 0.3 & 0 \end{bmatrix}$$

Step 2: Compute $a(i)$ for Each Point

The average intra-cluster distance $a(i)$ is calculated as:

$$a(x_1) = \frac{1}{2}\left[d(x_1,x_2) + d(x_1,x_3)\right] = \frac{1}{2}(0.5+0.6) = 0.55$$

$$a(x_4) = \frac{1}{1}d(x_4,x_5) = 0.3$$

Step 3: Compute $b(i)$ for Each Point

The average nearest-cluster distance $b(i)$ is:

$$b(x_1) = \min\left(\frac{1}{2}[d(x_1,x_4) + d(x_1,x_5)]\right) = \frac{1}{2}(1.5+1.6) = 1.55$$

$$b(x_4) = \min\left(\frac{1}{3}[d(x_4,x_1) + d(x_4,x_2) + d(x_4,x_3)]\right) = \frac{1}{3}(1.5+1.7+1.8) = 1.67$$

Step 4: Silhouette Score for Each Point

$$s(x_1) = \frac{b(x_1) - a(x_1)}{\max(a(x_1), b(x_1))} = \frac{1.55 - 0.55}{\max(0.55, 1.55)} = \frac{1}{1.55} \approx 0.645$$

$$s(x_4) = \frac{b(x_4) - a(x_4)}{\max(a(x_4), b(x_4))} = \frac{1.67 - 0.3}{1.67} \approx 0.82$$

Step 5: Overall Silhouette Score

$$S = \frac{1}{5}\sum_{i=1}^{5} s(x_i) \approx \frac{1}{5}(0.645 + 0.7 + 0.6 + 0.82 + 0.8) \approx 0.713$$

Step 6: Interpretation

- Since $S \approx 0.71$, the clustering is fairly well-defined.