National

University of Computer and Emerging Sciences

Laboratory Manual

for

Fundamentals of Big Data Lab

Course Instructor	Dr Iqra Safdar
Lab Instructors	Mr. Aiss Shahid Mr. Muhammad Mazarib
Section	
Date	01-Mar-2023
Semester	Spring 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm used in machine learning to group similar data points together.

The algorithm works by defining clusters as areas of high density separated by areas of low density. It does this by starting with a random data point, and then finding all the neighboring points that are within a certain distance of that point. If there are enough neighboring points within that distance, a cluster is formed. If not, the point is labeled as noise and the algorithm moves on to the next point.

DBSCAN is different from other clustering algorithms in that it doesn't require you to specify the number of clusters beforehand. Instead, it automatically determines the number of clusters based on the density of the data points. This makes it useful for datasets where the number of clusters is not known beforehand, or where the clusters are irregularly shaped.

The DBSCAN algorithm has two main parameters: epsilon (ε) and the minimum number of points required to form a cluster (MinPts). The epsilon parameter defines the radius around each data point that the algorithm will use to search for neighboring points. The MinPts parameter specifies the minimum number of neighboring points required to form a cluster.

DBSCAN has a few advantages over other clustering algorithms, such as its ability to identify noise points and its ability to handle clusters of different shapes and sizes. However, it can be sensitive to the choice of parameters, and it can struggle with datasets where the density of points varies widely.

Overall, DBSCAN is a powerful clustering algorithm that is useful for a variety of applications, including customer segmentation, anomaly detection, and image recognition.

Algorithm

1. The algorithm starts by selecting a random data point that has not yet been assigned to a cluster.
2. It then finds all the neighboring points that are within a certain distance (epsilon, ε) of the selected point.
3. If there are enough neighboring points (minimum points, MinPts) within the distance ε, the selected point is assigned to a cluster.
4. The algorithm repeats the above process for each point in the cluster until there are no more points left to add to the cluster.
5. The algorithm then selects a new unassigned data point and repeats the process, creating a new cluster if there are enough neighboring points within ε.
6. The algorithm continues until all points have been assigned to a cluster or labeled as noise.

Code

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs  # import make_blobs dataset from scikit-learn
from sklearn.cluster import DBSCAN  # import DBSCAN clustering algorithm from scikit-learn

# Generate sample dataset
X, y = make_blobs(n_samples=1000, centers=3, n_features=2, random_state=0)
# create 1000 data points in 2 dimensions with 3 clusters using make_blobs function from scikit-learn

# Run DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)  # create an instance of DBSCAN with parameters epsilon=0.5 and min_samples=5
dbscan.fit(X)  # fit DBSCAN model to the data

# Get cluster labels
labels = dbscan.labels_  # get cluster labels from DBSCAN model
```

```
# Plot the results
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='rainbow')  # plot the data points with different colors for different clusters
plt.title('DBSCAN Clustering')  # set the title of the plot
plt.show()  # show the plot
```

Question 1:

Just run the give code.

Question 2:

Write your own dbscan algorithm which should imitate the above output of question 1.

Note: This time you have to complete the task in lab, as it is short so please dont ask for extension.