Name: Jigar Siddhpura **SAPID:** 60004210155

DIV: C/C2 **Branch:** Computer Engineering

DMW - Experiment 10

60004210155
Jigar Siddhpura

DMW - Experiment 10

Aim: Implement Clarans Algorithm.
Write & Explain one algo each on:
1. Spatial association rule.
2. Spatial classification
3. Spatial clustering - DBScan

Theory: 1. Spatial data is data with geometric & location info. such as maps, 3D model, images, etc., can be discrete / continuous & can have spatial relationship with other.
2. Spatial data mining is the process of gathering patterns, relationships, knowledge from spatial data.
3. It is different from regular data mining as it uses spatial attributes & neighbors of objects.
4. Used for spatial characterization & spatial trend analysis.

→ Spatial Association Rule: 1. It refers to relationships btw variables over space.
2. It suggests that variable's value are spatially connected or related.
3. This association can be depicted through maps or mathematically.
4. Scientists use statistical ~~measure to~~ procedure to test & measure the existence of spatial association, confirming that the observed association is valid. One of the algo is — Apriori Algo

Algorithm: This algorithm can be adapted to analyse relationships & pattern in spatial data. Like in geographical dataset, this could discover association btw diff. locations or

events that occur frequently. It is commonly associated with market basket analysis & Healthcare analysis. It uses a BFS & hash tree to calculate itemset association efficiently.

Steps for Apriori : 1. Determine support of itemsets in DB & select min. support & confidence.
2. Take all support in a transaction with higher support value than min. sp value.
3. Find all rules of these subsets that have confidence value more than threshold.
4. Sort rules as dec. order of Lift.

Advantages : 1. Easy to understand & implement.
2. Join & prune steps can be easily implemented on large datasets.

Disadvantages : 1. Slow
2. Less efficients as it scan db multiple times.
3. Time & Space complexity = $O(2^d)$
where d = horizontal width of db.

→ Spatial Classification: 1. It assigns an object to a class from given set of classes based on attribute values of object.

2. It mainly considers distance direction or connectivity relationships among spatial objects.

3. Algo for this is — KNN

4. It assumes the similarity between new cases & available cases & puts the new case into category that is most similar to available categories.

5. KNN stores all available data & classifies new data based on similarity.

6. It is a non-parametric algorithm i.e. does not make any assumptions.

7. It is called lazy learner & it doesn't learn immediately from training set & instead stores dataset & during classification, it performs the action.

8. Eg: Lets say we want to classify an image into cat / dog. KNN will find ~~most~~ similar features of new image btw cats & dogs. The one with most similar will be classified accordingly.

Steps for KNN: 1. Select K neighbors.
2. Calc. Euclidean dist. of K neighbors.
3. Take K nearest neighbors
4. Count no. of categories from these neighbors.
5. Assign new data points to that category for which no. is max.
6. Model is ready.

Note : 1. There is no particular way to find best value for K. Most preferred value is '5'.
2. A very low value like 1 or 2, may be noisy.
3. Large values are good.

Advantages : 1. Simple to implement & robust to noisy data

Disadvantages : 1. 'K' needs to determined formerly.
2. High computational cost as it needs to find dist btw data points.

→ Spatial Clustering – DBScan

1. It is a descriptive task that seeks to identify homogenous groups of objects based on values of their attributes.
2. In spatial datasets, clustering permits a generalization of spatial component like explicit location & extension of spatial objects which define implicit relations of spatial neighborhood.
3. Current spatial clustering techniques can be broadly classified into : ~~Partio~~ Partitional, Hierarchical, locality based.

Algorithm (DBScan) :

1. Density based clustering refers to unsupervised learning method that identify distinct grps in data, based on idea that cluster in data space is a contiguous region of high point density, separated from other clusters by

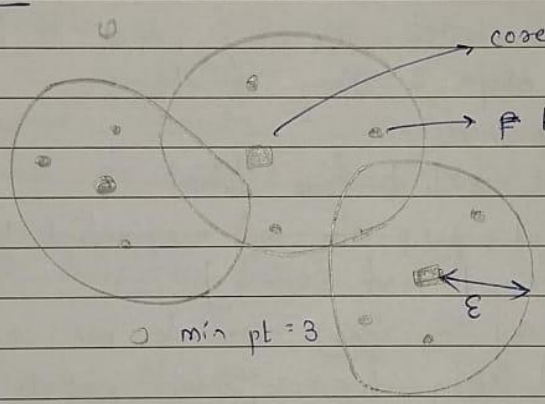
continuous regions of low density point.

2. It can discover clusters of different shapes & sizes from a large amount of data containing noise.

3. It has 2 params:
   a) MinPts: Min. no. of pts. in a clusters for it to be dense.
   b) Eps (ε): Radius of the cluster.

4. There are 3 types of pts after DBScan is complete:
   a) Core: Point that has atleast 'm' pt. ~~this~~ within 'ε'.
   b) Border: Point that has atleast 1 core pt. within 'ε'.
   c) Noise: Point neither core nor border.

Diagram:



Steps: 1. Algo proceeds by arbitrarily picking a point (until all pts. are visited)

2. If there are atleast 'minPts' points within a radius of 'ε', we consider it a cluster.

3. The clusters are then expanded by recursively repeating the neighbor-hood calc. for each neighbor pt.

4. For minPts, a guideline suggests that it should be atleast

'D+1' where D is the no. of dimensions in dataset & min. 3 to avoid trivial clusters.

5. The parameter 'ε' is determined by examining a k dist. graph, plotting dist to K = minPts | nearest neighbor. A good 'ε' value is where the plot exhibits an 'elbow' indicating an optimal balance.

6. Too small 'ε' results in unclustered data & high values leads to merged clusters.

Clarans : 1. It identifies clusters of ~~similar~~ spatially related data pts. It is designed for large datasets.

2. It uses randomized search technique to explore data space & find clusters, making it robust & flexible in terms of shape & size of clusters.

3. Steps to implement :

a) Initialize k mediods from dataset.

b) For each mediod, swap with each non-mediod to minimize total distance [local search]

c) Randomize by considering alternate non-mediod pts.

d) If perturbed config. improves, update mediods & repeat local search.

e) Repeat steps c) & d)

f) Return final mediods & cluster formed

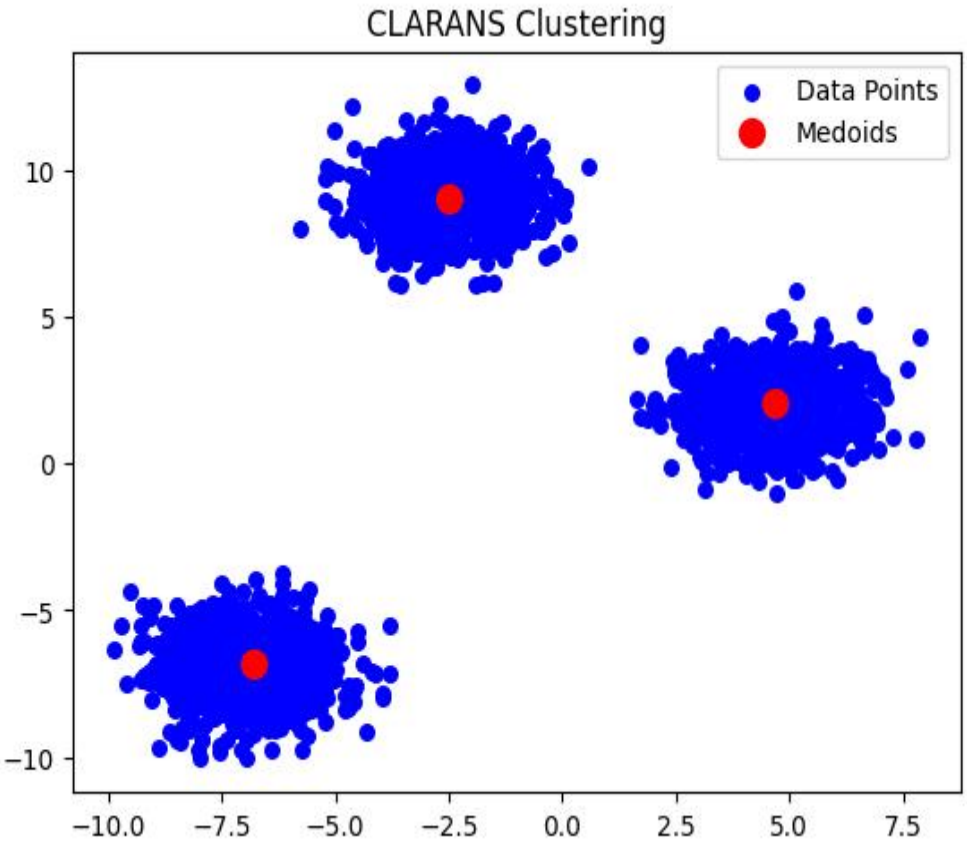
Conclusion : Hence, we implemented CLARANS algo. on a random generated dataset having 3 clusters & implemented same process as mentioned above & obtained mediods & clusters. Also, we learnt about spatial data, association rules, classification & clustering.

Code:

```
 import numpy as np
from sklearn_extra.cluster import KMedoids
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt

X, _ = make_blobs(n_samples=3000, centers=3, cluster_std=1, random_state=42)

# above line of code generate some sample spatial clustered data
# n_samples: The total number of points equally divided among clusters.
# centers: The number of centers to generate, or the fixed center locations.
# cluster_std: The standard deviation of the clusters. Larger values spread out the clusters.
# random_state: Seed for random number generation to ensure reproducibility.

def clarans(X, n_clusters, num_local, max_neighbor):
    kmedoids = KMedoids(n_clusters=n_clusters, method='alternate', max_iter=1)
    best_cost = float('inf')
    best_medoids = None
    for _ in range(num_local):
        kmedoids.fit(X)
        medoids = kmedoids.medoid_indices_
        cost = np.sum(np.min(X[medoids] - X[:, np.newaxis], axis=2), axis=1).mean()
        if cost < best_cost:
            best_cost = cost
            best_medoids = medoids
   return best_medoids

k = 3 #<- no. of cluster
num_local = 10
max_neighbor = 10
medoids = clarans(X, k, num_local, max_neighbor)
# plotting the results
plt.scatter(X[:, 0], X[:, 1], c='blue', marker='o', s=30, label='Data Points')
plt.scatter(X[medoids, 0], X[medoids, 1], c='red', marker='o', s=100, label='Medoids')
plt.title('CLARANS Clustering')
plt.legend()
plt.show()
```

)

Output :


CLARANS Clustering
Data Points
Medoids
10
5
0
−5
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5