Exercise 5

Comparison of Clustering Algorithms Using WEKA Tool

Objective:

The objective of this lab experiment is to compare the results of K-Means, COBWEB, Canopy, and Hierarchical clustering algorithms using the Iris dataset. The comparison will be based on the following factors:

- Number of clusters.
- Clusters instances.
- Time taken to build the model.

Materials Required:

- Computer with WEKA Tool installed
- Iris dataset

Procedure:

1. **Data Preparation:**

   - Launch the WEKA Tool on your computer.
   - Load the Iris dataset into the WEKA environment.

2. **Selection of Clustering Algorithms:**
   - Navigate to the clustering algorithms section in the WEKA Tool.
   - Select K-Means, CONWEB, Canopy, and Hierarchical clustering algorithms for comparison.
3. **Parameter Configuration:**
   - Set the parameters for each clustering algorithm:

a) K-Means:

- ✓ **Number of clusters:** This parameter depends on the dataset and the desired granularity of clustering. For the Iris dataset, a common choice is 3 clusters since there are three distinct classes in the dataset.

List of Parameters:

Parameter	Sample Suitable Value
canopyMaxNumCanopiesToHoldInMemory	Not applicable for K-Means clustering
canopyMinimumCanopyDensity	Not applicable for K-Means clustering
canopyPeriodicPruningRate	Not applicable for K-Means clustering
canopyT1	Not applicable for K-Means clustering
canopyT2	Not applicable for K-Means clustering
debug	false
displayStdDevs	false
distanceFunction	EuclideanDistance
doNotCheckCapabilities	false
dontReplaceMissingValues	false
fastDistanceCalc	true
initialization Method	kMeansPlusPlus
maxiterations	100
numClusters	Depends on the dataset and desired granularity, common choice: 3 for Iris dataset
numExecutionSlots	1
preserveinstancesOrder	false
reduceNumberOfDistanceCalcsViaCanopies	Not applicable for K-Means clustering
seed	Any integer value

b) COBWEB (COnstrained Nearest NEighBor):

- ✓ COBWEB is a constrained clustering algorithm that requires specific constraints. In the absence of specific constraints, default parameters may not be applicable. However, you can leave the parameters as default if no specific constraints are provided.

List of Parameters:

Parameter	Sample Suitable Value
Acuity	0.5
Cutoff	0.1
Debug	false
doNotCheckCapabilities	false
SaveInstanceData	true
Seed	Any integer value

c) Canopy:

- ✓ T1: The distance threshold for the canopy to start including points.
- ✓ T2: The distance threshold for the canopy to stop including points.

- ✓ These thresholds depend on the dataset and should be set empirically. For the Iris dataset, a common choice could be T1 = 1.0 and T2 = 0.5.

List of Parameters:

Parameter	Sample Suitable Value
T1 (canopyT1)	1.0
T2 (canopyT2)	0.5
debug	false
doNotCheckCapabilities	false
dontReplaceMissingValues	false
maxNumCandidateCanopiesToHoldInMemory	100
minimumCanopyDensity	0.1
numClusters	Depends on the dataset and desired granularity, common choice: 3 for Iris dataset
periodicPruningRate	0.5
seed	Any integer value

d) Hierarchical:

- ✓ Linkage method: Choose between single, complete, average, or Ward's method based on the characteristics of the dataset. For the Iris dataset, Ward's method is often a good choice.
- ✓ Distance measure: Euclidean distance is a common choice for measuring the distance between points in hierarchical clustering.

List of Parameters:

Parameter	Sample Suitable Value
Linkage method (linkType)	Ward
Distance measure	Euclidean
debug	false
distanceFunction	EuclideanDistance
distancesIsBranchLength	false
doNotCheckCapabilities	false
numClusters	Depends on the desired granularity, common choice: 3 for Iris dataset
printNewick	true (if you want to print the Newick representation of the dendrogram)

4. Execution and Model Building:

- ✓ Execute each clustering algorithm individually on the Iris dataset.
- ✓ Record the time taken by each algorithm to build the model.

- ✓ Note down the number of clusters formed by each algorithm.
- ✓ Document the instances present in each cluster.

5. **Data Collection and Analysis:**
   - Collect the data obtained from the executions of clustering algorithms.
   - Analyze the results to compare the performance of each algorithm based on the given factors.
6. **Graphical Representation:**
   - Create graphs to visually represent the comparison between the clustering algorithms.
   - Use bar graphs to compare the number of clusters formed by each algorithm.
   - Use pie charts or stacked bar graphs to illustrate the distribution of instances across clusters.
   - Use a line graph or bar graph to compare the time taken by each algorithm to build the model.
7. **Interpretation of Results:**
   - Interpret the graphs to draw meaningful conclusions.
   - Discuss the implications of the observed differences in the performance of clustering algorithms.
   - Identify any patterns or trends that emerge from the data analysis.
8. **Conclusion:**
   - Discuss the significance of the experiment results.
   - Compare the strengths and weaknesses of each clustering algorithm.
   - Provide recommendations for selecting the most suitable algorithm based on specific requirements.

Web Reference:

https://www.tutorialspoint.com/weka/weka_clustering.htm

https://www.geeksforgeeks.org/k-means-clustering-using-weka/

https://www.geeksforgeeks.org/hierarchical-clustering-using-weka/

Video Reference:

https://youtu.be/--hJXKFLjP0

https://youtu.be/TtBgfXmIDHQ

https://youtu.be/eVtYfH9LBgk

https://youtu.be/MzoZwzkwU-Y

Research Article Reference:

https://ijiset.com/vol7/v7s10/IJISET_V7_I10_31.pdf

https://www.researchgate.net/publication/337676095_Comparison_the_various_clustering_algorithms_of_weka_tools

https://ijcrt.org/papers/IJCRT_196160.pdf