**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |
| **Course/Section:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Semester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |

**CS-477 Computer Vision**

**Lab#12: Clustering**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **PLO4-CLO4** | **PLO5-CLO5** | **PLO8-CLO6** | **PLO9-CLO7** |
| **Name** | **Reg. No** | **Investigation**  **(5 marks)** | **Modern Tool Usage**  **(5 marks)** | **Ethics**  **(5 marks)** | **Individual and Team Work**  **(5 marks)** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Lab#12: Clustering**

**Objectives**

This laboratory exercise is focused on K-means and DBSCAN clustering which is a widely used unsupervised learning technique. Clustering is used on unlabeled data to look for interesting groups and patterns.

**Lab Instructions**

* This lab activity comprises of following parts: Lab Exercises, and Post-Lab Viva/Quiz session.
* The lab report shall be uploaded on LMS.
* Only those tasks that are completed during the allocated lab time will be credited to the students. Students are however encouraged to practice on their own in spare time for enhancing their skills.

**Lab Report Instructions**

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* Python codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

**K-mean Clustering**

K-means clustering is an unsupervised learning technique that is used to find groups, clusters or patterns in unlabeled. As the dataset is not labelled, only the arrangement of the inputs on the feature space are available. In K-means clustering, K number of clusters are set and then the examples are compared to the cluster centroids. The distance of each feature is used as a metric to define which cluster it belongs to. The cluster centroids are iteratively shifted and the examples belonging to them also change. After enough iterations, useful groups in the feature space are obtained. To determine the best number for clusters, a cost function can be calculated for each K number.

For this lab, you will be provided with some dataset files in .csv format which you will need for the tasks. Additionally, for the final task, you will need to arrange your own dataset by downloading it from the internet. You will need to make use of numpy, pandas and matplotlib libraries for the given tasks.

**Lab Task 1 – 2-Means Clustering \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

In this task, use the provided dataset. Write the code which performs clustering of the dataset into 2 clusters. The pseudocode for the clustering algorithm is provided as follows:

**specify K number of centroids**

**randomly initialize K number of centroids u**

**for j = 1:epochs**

**for i = 1:m**

**c(i) = index of closest cluster to training example**

**for k = 1:K**

**u(k) = mean of all training examples indexed to k**

**plot of x1 and x2 clusters**

To determine the index c(i), you will need to write a function that calculates the Euclidean distance between the points in the feature space. This function will be used to find the closest centroid from each training example. After determining the indexes, the cluster centroids themselves are updated by taking the average of the x values. For k-th cluster, the training examples with index k will be averaged. This completes one iteration of clustering after which a scatter plot is made. The iterations are repeated until interesting groups are obtained in the plots.

Due to the initial randomization of cluster centroids, you may have to repeat the clustering a few times. Also, ensure the random centroids are from within the domain of the feature space.

Your code must generate scatter plots showing the clusters at each iteration. The input values must be colored and marked according to the cluster to which they belong at each iteration. The cluster centroids must also be shown Provide all of the codes and screenshots of the final output. You must include plots from at least 3 different iterations showing the progress of your clustering algorithm.

**Lab Task 2 – K-Means Clustering \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Repeat task 1, however, set value of K = 3, 4 and 5. For all three k values, generate at least three plots.

**Lab Task 3 – Cost Function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Load the given dataset into the python program for this task. In this task, you will modify your code so that it performs clustering from

K = 2, 3, 4… 10.

For each K value, perform about 20 iterations (epochs) of centroid update before moving to the next value of k. Additionally, at the last iteration of each k-value, determine the cost for that K-value:

**for K = 1:10**

**randomly initialize K number of centroids u**

**for j = 1:epochs**

**compute cluster centroids u(k)**

**plot of x1 and x2 clusters**

**compute cost for current K value**

**plot of cost and K**

Store the costs for each k in a list. After the last iteration of the last cluster, make a plot of k vs. cost.

**Lab Task 4 – Your Own Dataset \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Download your own CSV dataset from the internet (e.g. Kaggle). Your dataset must have at least 500 rows and at least 4 feature columns. Perform clustering of your dataset and showcase the plots (the cost function is optional in this task).

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

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm in machine learning and data mining. It is particularly useful for identifying clusters of data points in a dataset based on the density of points in the feature space. Unlike k-means or hierarchical clustering, DBSCAN doesn't require specifying the number of clusters beforehand and can discover clusters of arbitrary shapes.

Here is the basic algorithm for DBSCAN:

1. **Input:**
   * **Dataset:** *D*={*x*1​,*x*2​,...,*xn*​}, where *xi*​ is a data point in the feature space.
   * **Parameters:**
     + **Epsilon (*ε*):** The maximum distance between two points for one to be considered as in the neighborhood of the other.
     + **MinPts:** The minimum number of points required to form a dense region.

Exploring two concepts known as Density Reachability and Density Connectivity helps in understanding these parameters.

**Density Reachability,** with respect to density, defines a point as reachable from another if it is within a specific distance (epsilon) from it.

**Density Connectivity,** on the other hand, employs a transitivity-based chaining approach to ascertain if points belong to a specific cluster. For instance, points p and q may be connected if p->r->s->t->q, where a->b signifies that b is in the neighborhood of a.

1. **Algorithm:**
   * For each data point *p* in the dataset *D*:
     + If *p* is not visited:
       - Mark *p* as visited.
       - Find all points in the *ε*-neighborhood of *p* (including *p*).
       - If the number of points in the neighborhood is less than MinPts, mark *p* as noise.
       - Otherwise, create a new cluster and add *p* to the cluster.
       - Expand the cluster by adding all reachable points in the *ε*-neighborhood to the cluster.
2. **Output:**
   * The algorithm identifies clusters of data points and marks some points as noise if they don't belong to any cluster.

In the algorithm, a point *q* is considered to be in the *ε*-neighborhood of *p* if the distance between *p* and *q* is less than or equal to *ε*. The algorithm classifies points into three categories:

* **Core points:** Points with at least MinPts points in their *ε*-neighborhood.
* **Border points:** Points with fewer than MinPts points in their *ε*-neighborhood but are reachable from a core point.
* **Noise points:** Points that are neither core nor border points.

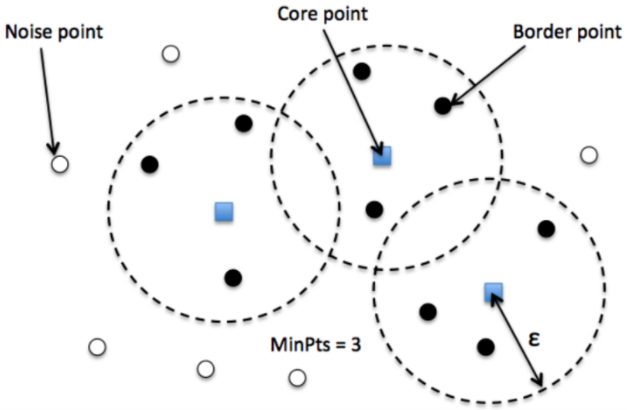


Figure :Credit https://www.theaidream.com/post/dbscan-clustering-algorithm-in-machine-learning

DBSCAN has advantages such as being robust to outliers and capable of discovering clusters of arbitrary shapes. However, it may struggle with datasets of varying densities, and choosing appropriate values for *ε* and MinPts can be challenging.

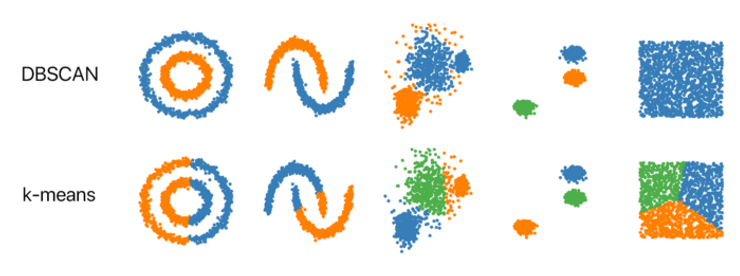


Figure : Credits: https://github.com/NSHipster/DBSCAN

**Lab Task 5 – Your Own Dataset \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Download your own CSV dataset from the internet (e.g. Kaggle). Perform DBSCAN clustering of your dataset and showcase the plots .

**Lab Task 6 – Take home(optional)**

Download your own CSV dataset from the internet e.g heatmap. Perform Hierarchical clustering of your dataset and showcase the plots .