Affinity Propagation (AP)

*Affinity Propagation (AP) is a clustering algorithm that doesn't require the user to specify the number of clusters beforehand. Instead, **it works by sending messages between data points to discover exemplars,** which are representative points in the dataset.

Similarity Matrix:

AP starts by computing a similarity matrix where each **element represents the similarity between pairs of data points**. The similarity metric could be any measure of similarity or dissimilarity, such as negative Euclidean distance or negative squared distance.

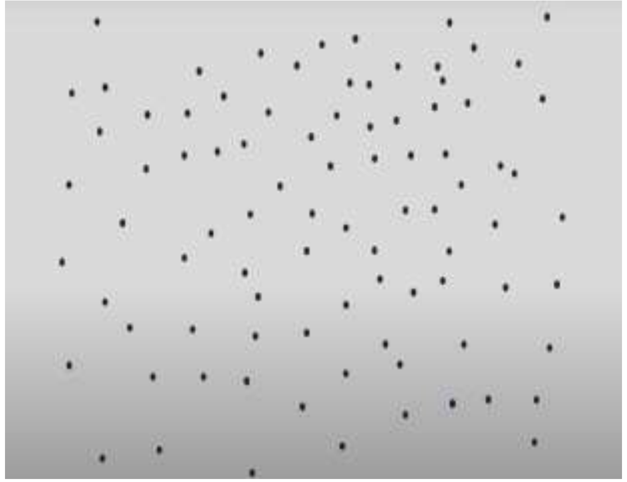
Responsibility Matrix:

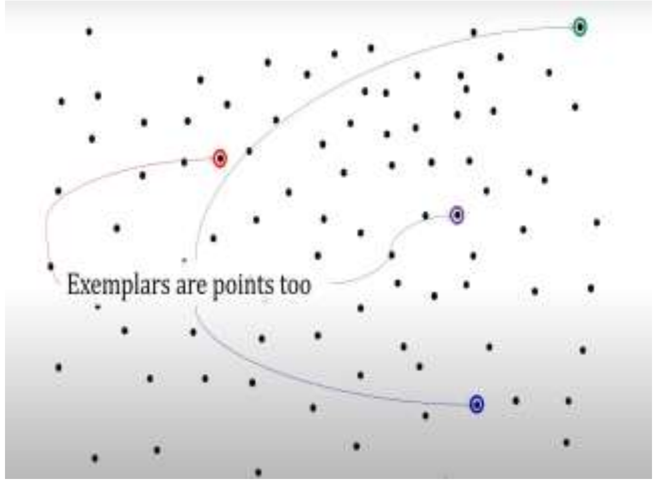
AP iteratively updates a matrix of responsibilities, denoted by ***r*(*i*,*k*),** which represents how well-suited **one data point *i*** is to be an exemplar for another **data point K**.

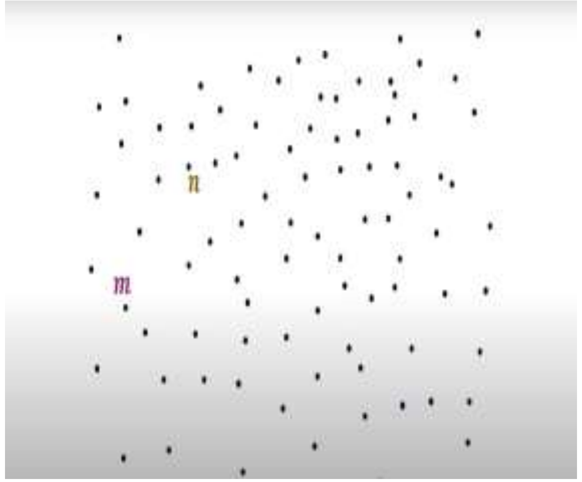
Affinity Propagation (AP)

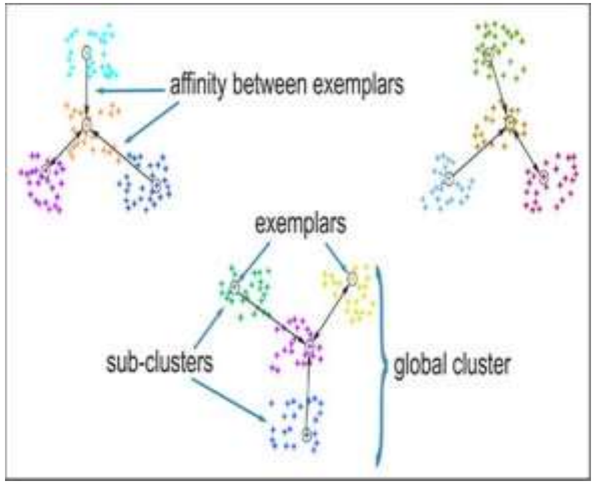
- **Availability Matrix**:
- Similarly, AP updates a matrix of availabilities**, denoted by *a*(*i*,*k*),** which represents the accumulated evidence that data point *k* should **choose data point *i* as its exemplar**.
- **Message Passing**:
- AP iterates between updating the responsibility and availability matrices based on current values. It updates these **matrices by passing messages between data points** according to specific rules.

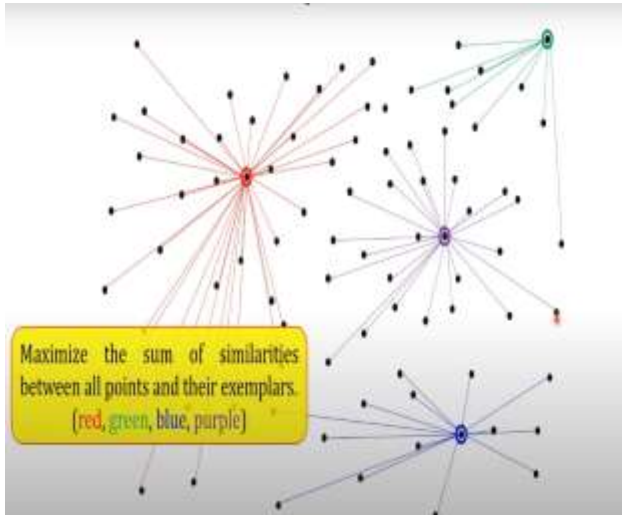
EVALUATION OF AFFINITY PROPAGATION

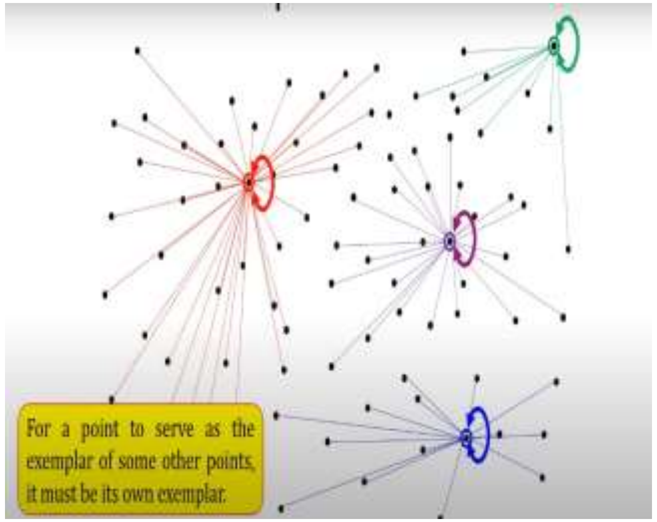


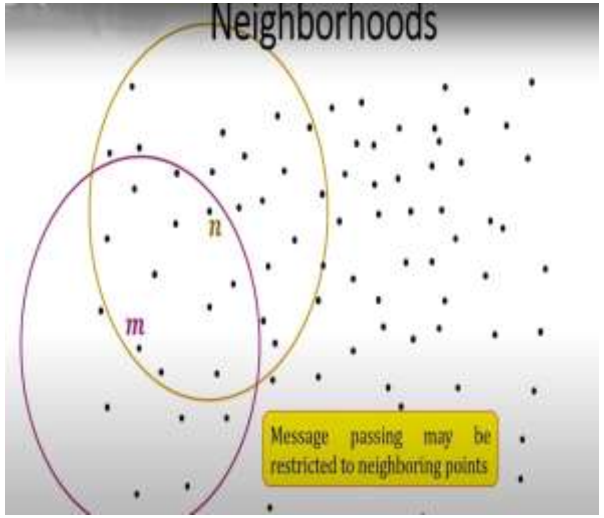


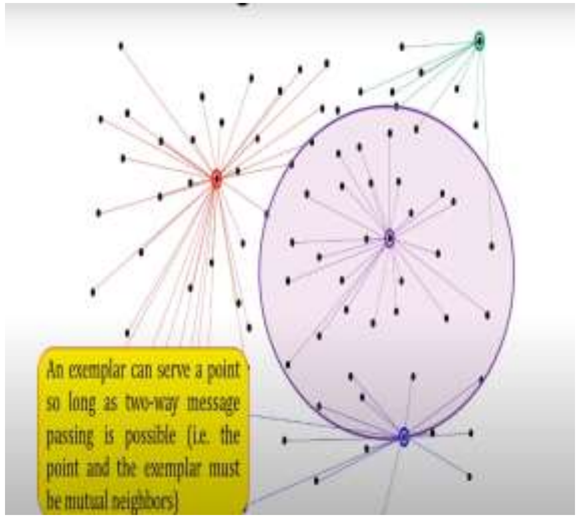












Affinity Propagation (AP)

- **Exemplar Selection**:
- After a certain number of iterations or when the changes in the matrices become small, the algorithm identifies exemplars and assigns data points to these **exemplars based on the values in the responsibility and availability matrices**.
- **Cluster Formation**:
- Finally, clusters are formed based on the exemplars identified. Data points are **assigned to clusters based on which exemplar** they are most strongly associated with.

Affinity Propagation (AP)

Here's a breakdown of the Python code, focusing on the application of Affinity Propagation:

- **import** pandas as pd
- **import** numpy as **np**
- **import** matplotlib.pyplot as **plt**
- from **sklearn**.cluster import **Affinity Propagation**
- from **sklearn**.datasets import **make blobs**
- from **sklearn** import **metrics**

- **Imports**:
- The necessary libraries are imported: pandas for data handling, NumPy for numerical computations, matplotlib for plotting, and various modules from scikit-learn for clustering and metrics evaluation.

Affinity Propagation (AP)

- Dataset = pd.read_csv("Mall_Customers.csv")

Dataset Loading:

The code loads a dataset named "Mall_Customers.csv" using pandas.

X = dataset.iloc[:, 3:5].values

Feature Selection:

It selects specific columns (features) from the dataset, which will be used for clustering.

Affinity Propagation (AP)

- centers = [[1, 1], [-1, -1], [1, -1]]
- X, labels true = make blobs(n_samples=300, centers=centers, cluster_std=0.5, random_state=0)
- **Data Generation**:
- The code generates sample data for clustering using the make blobs function from scikit-learn. This data is used to demonstrate clustering
- af = AffinityPropagation(preference=-50, random_state=0).fit(X).
- **Affinity Propagation**:
- It applies the Affinity Propagation algorithm to the generated data.

Affinity Propagation (AP)

- cluster_centers_indices = af.cluster_centers_indices_
- labels = af.labels_
- n_clusters_ = len(cluster_centers_indices)
- **Cluster Retrieval**: The code retrieves the cluster centers, labels, and the number of clusters identified by the algorithm.
- print("Estimated number of clusters: %d" % n_clusters_)
- print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true, labels))
- **Metrics Evaluation**:
- Various clustering metrics are calculated and printed to evaluate the performance of the clustering algorithm.
- **# Plotting clusters:**
- plt.close("all")
- plt.figure(1)
- plt.clf()
- colors = plt.cycler("color", plt.cm.viridis(np.linspace(0, 1, 4)))

Affinity Propagation (AP)

- **for k, col in zip(range(n_clusters_), colors):**
- **class_members = labels == k**
- **cluster_center = X[cluster_centers_indices[k]]**
- **plt.scatter(X[class_members, 0], X[class_members, 1], color=col["color"], marker=".")**
- **plt.scatter(cluster_center[0], cluster_center[1], s=14, color=col["color"], marker="o")**
- **for x in X[class_members]:**
- **plt.plot([cluster_center[0], x[0]], [cluster_center[1], x[1]], color=col["color"])**
- **plt.title("Estimated number of clusters: %d" % n_clusters_)**
- **plt.show()**

- **For Loop:**
- for k, col in zip(range(n_clusters_), colors): This loop iterates over the range of the number of clusters (n_clusters_) along with the colors generated by plt.cycler.

Affinity Propagation (AP)

- **Cluster Members:**
- class_members = labels == k: This line identifies the data points that belong to the current cluster k. It creates a boolean array where True indicates that the corresponding data point belongs to the current cluster.

- **Cluster Center:**
- cluster_center = X[cluster_centers_indices[k]]: It retrieves the coordinates of the cluster center for the current cluster k from the data X.

- **Scatter Plot:**
- plt.scatter(X[class_members, 0], X[class_members, 1], color=col["color"], marker="."): This line plots the data points belonging to the current cluster k using a scatter plot. It assigns the color col["color"] to the points and uses a dot marker.
- plt.scatter(cluster_center[0], cluster_center[1], s=14, color=col["color"], marker="o"): This line plots the cluster center using a larger circular marker.

- **Connecting Lines:**
- for x in X[class_members]: plt.plot([cluster_center[0], x[0]], [cluster_center[1], x[1]], color=col["color"]): This loop plots lines connecting each data point in the current cluster k to the cluster center. It creates lines from the cluster center to each data point.

- **Title and Display**:
- plt.title("Estimated number of clusters: %d" % n_clusters_): It sets the title of the plot to indicate the estimated number of clusters.
- plt.show(): This command displays the plot with all the clusters and their respective center points, along with the connecting lines.

Affinity Propagation (AP)