Hamza Ali (haal2410)

K-Means Algorithm

The K-means algorithm aims to minimize the sum of squared distances between each data point and the centroid of the cluster to which it belongs. The function to be minimized is:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

where:

- k is the number of clusters.
- Ci represents the set of points in cluster i.
- μ_i is the centroid of cluster i.
- $||x-\mu_i||^2$ is the squared Euclidean distance between a point x and its assigned centroid μ_i.

Assumptions in K-means

1. **Euclidean Space**: The points are assumed to be in a space where Euclidean distance is meaningful.
2. **Spherical Clusters**: K-means assumes that clusters are spherical (equal variance in all directions).
3. **Equal-sized Clusters**: It performs best when clusters have similar sizes and densities.
4. **Independent Features**: K-means assumes that the dimensions of the data are independent.

K-Means Calculation for given points:

$C_1 = (1, 1.5)$, $C_2 = (3, 1)$

For Each point Calculate Euclidean distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Point	d with C_1	d with C_2	Cluster
(0, 0.5)	$\sqrt{(0-1)^2 + (0.5-1.5)^2} = 1.41$	$\sqrt{(0-3)^2 + (0.5-1)^2} = 3.04$	C_1
(0, 0.75)	$\sqrt{(0-1)^2 + (0.75-1.5)^2} = 1.26$	$\sqrt{(0-3)^2 + (0.75-1)^2} = 3.02$	C_1
(1, 1)	$\sqrt{(1-1)^2 + (1-1.5)^2} = 0.5$	$\sqrt{(1-3)^2 + (1-1)^2} = 2$	C_1
(1.25, 0.4)	$\sqrt{(1.25-1)^2 + (0.4-1.5)^2} = 1.1$	$\sqrt{(1.25-3)^2 + (0.4-1)^2} =$	C_1
(1.5, 0.7)	$\sqrt{(1.5-1)^2 + (0.7-1.5)^2} = 0.9$	$\sqrt{(1.5-3)^2 + (0.7-1)^2} = 1.54$	C_1
(2.5, 1)	$\sqrt{(2.5-1)^2 + (1-1.5)^2} = 1.58$	$\sqrt{(2.5-3)^2 + (1-1)^2} = 0.5$	C_2
(3, 2)	$\sqrt{(3-1)^2 + (2-1.5)^2} = 2.06$	$\sqrt{(3-3)^2 + (2-1)^2} = 1.$	C_2
(4, 1.5)	$\sqrt{(4-1)^2 + (1.5-1.5)^2} = 3$	$\sqrt{(4-3)^2 + (1.5-1)^2} = 1.12$	C_2
(4, 2.5)	$\sqrt{(4-1)^2 + (2.5-1.5)^2} = 3.16$	$\sqrt{(4-3)^2 + (2.5-1)^2} = 1.80$	C_2
(5, 2)	$\sqrt{(5-1)^2 + (2-1.5)^2} = 4.03$	$\sqrt{(5-3)^2 + (2-1)^2} = 2.24$	C_2

Now find Centroids:

$$C_1 = \left(\frac{0+0+1+1.25+1.5}{5} , \frac{0.5+0.75+1+0.4+0.7}{5} \right)$$

$C_1 = (0.75, 0.67)$

$$C_2 = \left(\frac{2.5+3+4.4.5}{5} , \frac{1+2+1.5+2.5+2}{5} \right)$$

$C_2 = (3.7, 1.8)$

Centroids are not changing with 2nd Iteration so, the convergence is obtained on second iteration.

4.2 Classification algorithms:

Algorithm	When Applicable	Pros	Cons
K-Nearest Neighbors (KNN)	Small datasets, low-dimensional data, simple classification	Easy to implement, no training required	Slow for large datasets, sensitive to noise
Support Vector Machine (SVM)	Well-separated classes, moderate-dimensional data	Works well with small data, effective in high dimensions	Can be slow on large datasets, sensitive to parameter tuning
Decision Trees (DT)	Interpretable models, categorical & numerical data	Easy to understand, fast training	Prone to overfitting, unstable to small data changes
Random Forest (RF)	Large datasets, high variance datasets	Reduces overfitting, handles missing values well	Computationally expensive, less interpretable
Naïve Bayes (NB)	Text classification, probabilistic models	Fast, simple, works well with small data	Assumes feature independence, may not fit complex data
Template Matching	Well-defined, fixed objects in images	Good for structured patterns, simple	Fails with scale, rotation, occlusion

References:

https://ieeexplore.ieee.org/document/549118

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10562290

https://elitedatascience.com/machine-learning-algorithms

Richard Szeliski, "Computer Vision: Algorithms and Applications," 2nd Edition, Springer, 2022.

4.3

In this task, I used the Mean Shift algorithm to segment a color image contains different colored flowers. I used a five-dimensional (5D) space for segmentation, which includes the color of each pixel (red, green, blue) and its position in the image (x and y coordinates). This helps the algorithm group pixels that are not only similar in color but also close to each other in the image. I tried different values for the bandwidth parameter to see how it affects the result. With a small bandwidth, the image was divided into many small regions, and with a large bandwidth, similar areas were grouped into bigger sections. This showed how changing the settings can give different types of segmentation.

Code:

https://github.com/Hamzaali220/DT084/tree/main/week4