CSE-575
Project Report- K-means Clustering

Gaurav Kumar-ASU ID-1229081284
gkumar28@asu.edu

Introduction

In this report, we implement and analyze the K-means clustering algorithm on a 2-D dataset using two different strategies for initializing centroids:

1. **Strategy 1 (K-means)**: Randomly pick the initial centers from the given samples.
2. **Strategy 2 (K-means++)**: Pick the first center randomly; for the i-th center (i >1), choose a sample such that the average distance of this chosen one to all previous (i -1) centers is maximal.

Methodology (K- means):

- **Data Source**
  The data used in this project comes from the file **AllSamples.npy** which contains a set of 2-D points.
- **Initialization Strategy**
  For this part, the initialization strategy (Strategy 1) was to randomly pick cluster centers directly from the sample points.
- **Algorithm Implementation**
  The K-means algorithm was implemented with the following steps:
  1. Randomly initialize cluster centroids using Strategy 1.
  2. Assign each data point to the nearest centroid.
  3. Recalculate centroids based on the mean of all data points assigned to that centroid.
  4. Repeat steps 2 and 3 until the centroids no longer change significantly.
- **Objective Function**
  The objective function used to measure the performance of the clustering is:
  $\sum_{i=1}^{k} \sum_{x \in D_i} |x - \mu_i|^2$ Where:
  k is the number of clusters.
  D_i represents the set of samples in the *i*-th cluster.
  μ_i is the centroid of the *i*-th cluster.

Methodology (K-means++):

For the project second part, initialization strategy (**Strategy 2**) was more structured than simply choosing random data points:

- The first center is picked randomly from the dataset.
- For the subsequent centers (i-th center, where i > 1), we choose a data point such that its average distance to all previously chosen centers is maximal.
- The K-means++ inspired strategy, in general, provides more consistent results across different runs compared to the random initialization method.

- The distributed nature of the initialization strategy in Part 2 often results in a better representation of the underlying data distribution, leading to potentially better clustering outcomes.
- As with Part 1, increasing the number of clusters generally resulted in reduced loss. However, the reduction rate might be different due to the improved initialization strategy.

Results

Clustering Outcome

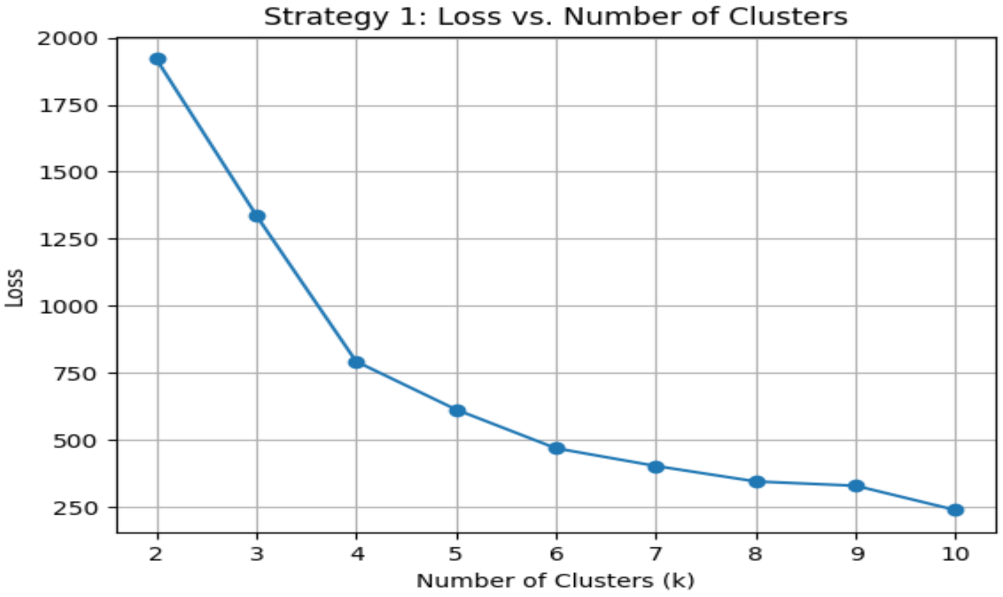
The K-means and K-means++ algorithm was tested with the number *k* of clusters ranging from 2 to 10. For each value of *k*, the algorithm was run until convergence, and the final centroid coordinates were recorded.

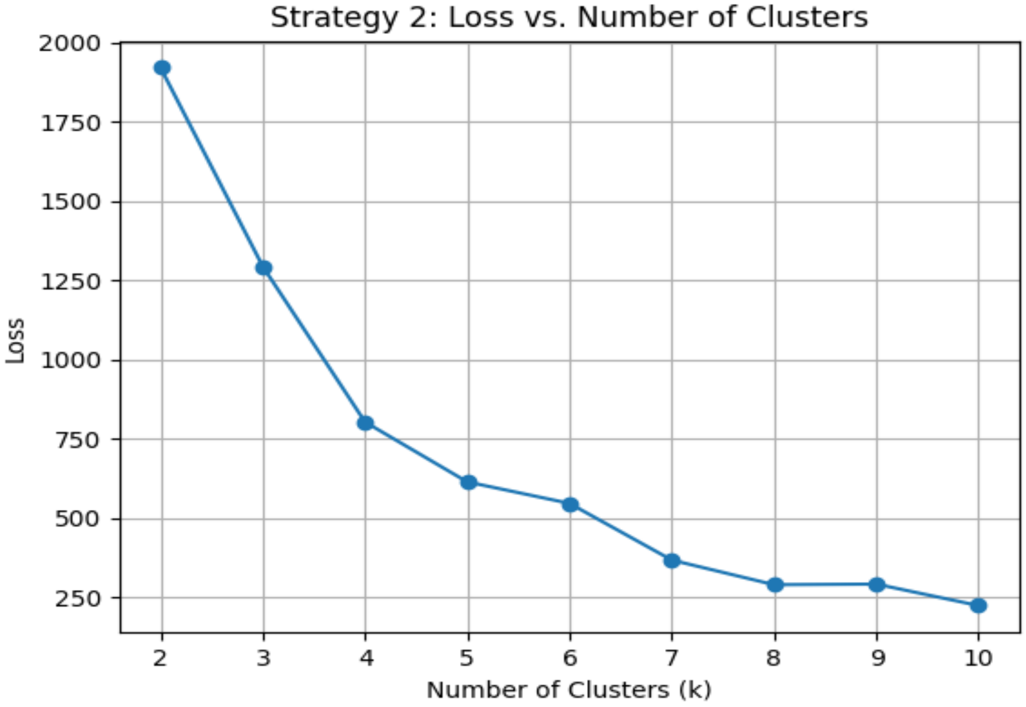
Loss Computation

For each value of *k*, the loss was computed using the objective function, which essentially measures the compactness of the clusters. A lower loss indicates better clustering as it signifies that the points in each cluster are closer to their respective centroids.

Here are the results for both the strategies:

Strategy 1(K-means)	**Strategy 2(k-means++)**
For k=3, Loss=1338.1076016520997, Centroids= [[7.23975119 2.48208269] [3.23489005 2.5530322] [4.83091958 7.29959959]], initial_centroid=[[7.10604472 1.19751007] [6.8950152 0.95350302] [4.05095774 4.05212767]] For k=5, Loss=613.4330781284366, Centroids=[[2.60123296 6.91610506] [5.40252508 6.73636175] [7.34802851 2.35222497] [3.28848611 2.52832379] [7.75648325 8.55668928]], initial_centroid=[[5.68766272 5.38279515] [8.37895231 8.62509614] [7.57805025 3.82487017] [4.74625798 3.54661053] [8.527899 8.55183237]]	For k=4, Loss=803.2167238057567, Centroids=[[6.79532432 2.78778512] [6.92822285 7.92187152] [3.19669343 6.8712608] [2.85235149 2.28186483]], initial_centroid=[[3.79752017 0.69134312] [9.26998864 9.62492869] [1.20162248 7.68639714] [3.85212146 -1.08715226]] For k=6, Loss= 476.29657052696643 , Centroids=[[2.68198633 2.09461587] [5.24028296 7.53131029] [7.55616782 2.23516796] [2.54165252 7.00267832] [7.91430998 8.51990981] [5.23053667 4.2793425]], initial_centroid=[[2.68080913 1.61298226] [9.26998864 9.62492869] [3.85212146 -1.08715226] [2.95297924 9.65073899] [9.26998864 9.62492869] [3.85212146 -1.08715226]]


Strategy 1: Loss vs. Number of Clusters
Loss
2000
1750
1500
1250
1000
750
500
250
2
3
4
5
6
7
8
9
10
Number of Clusters (k)


Strategy 2: Loss vs. Number of Clusters
Loss
2000
1750
1500
1250
1000
750
500
250
2
3
4
5
6
7
8
9
10
Number of Clusters (k)