CSE 575 Project2 Unsupervised Learning(K-means) Report

Ziming Dong

10/27/2019

For this project, I write five functions to apply implementation on the given dataset in the coding part:

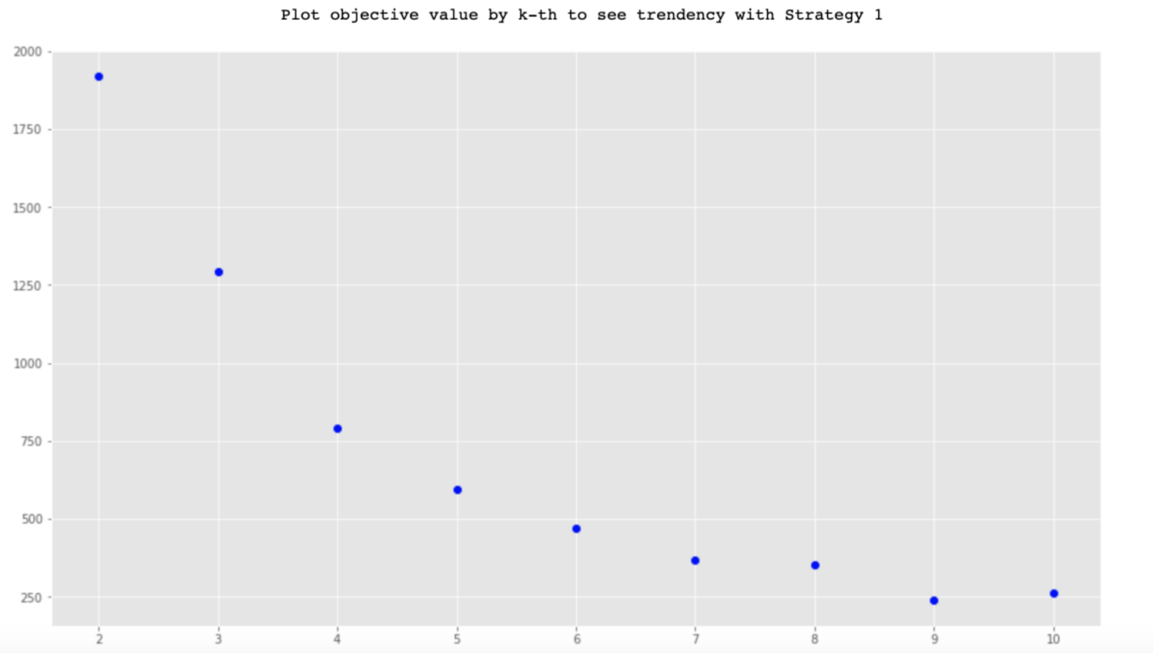
1. Implement the calculation function to calculate the distance between two points.
2. Implement the strategy 1 to randomly picking the initial cluster centers from the given samples.
3. Implement the strategy 2 to pick the first center randomly and pick latter centers by calculating the average distance of the chosen one to all previous(i-1) centers is maximal.
4. Implement the K-means algorithm, which is classified n samples according to the nearest mean of cluster`s points, recompute the mean until it does not change.
5. Implement the calculation function to calculate the objective function value vs. the number of cluster K. The formula is:

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in D_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2 \;)$$

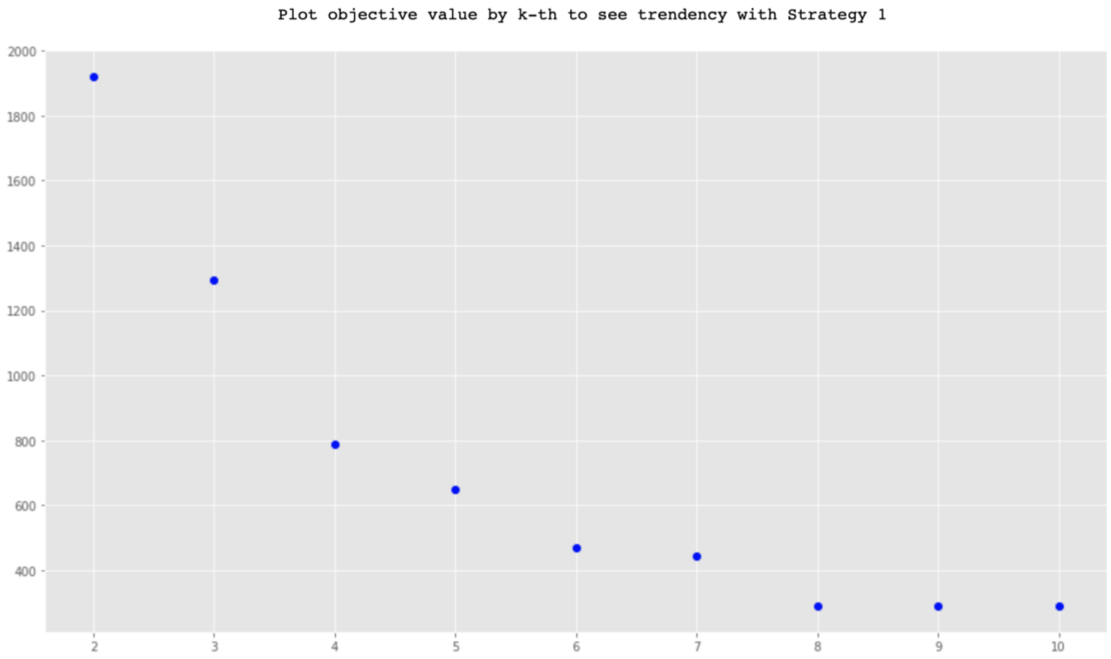
For both strategies, I plot objective value with the number of k.

For strategy 1:

The result for the first initialization:



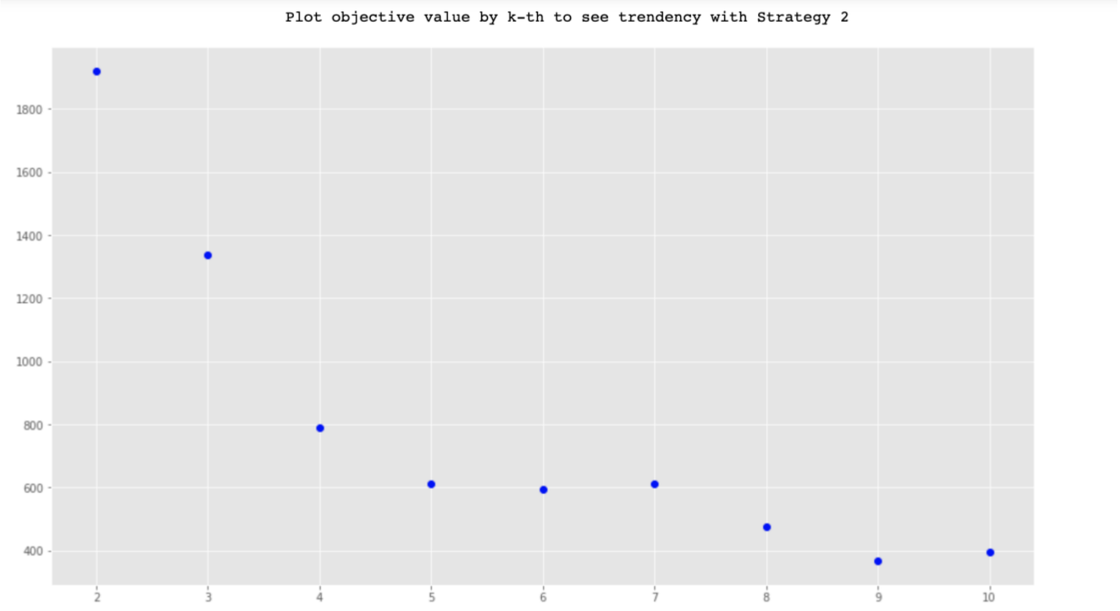
The result for the second initialization:


Plot objective value by k-th to see trendency with Strategy 1

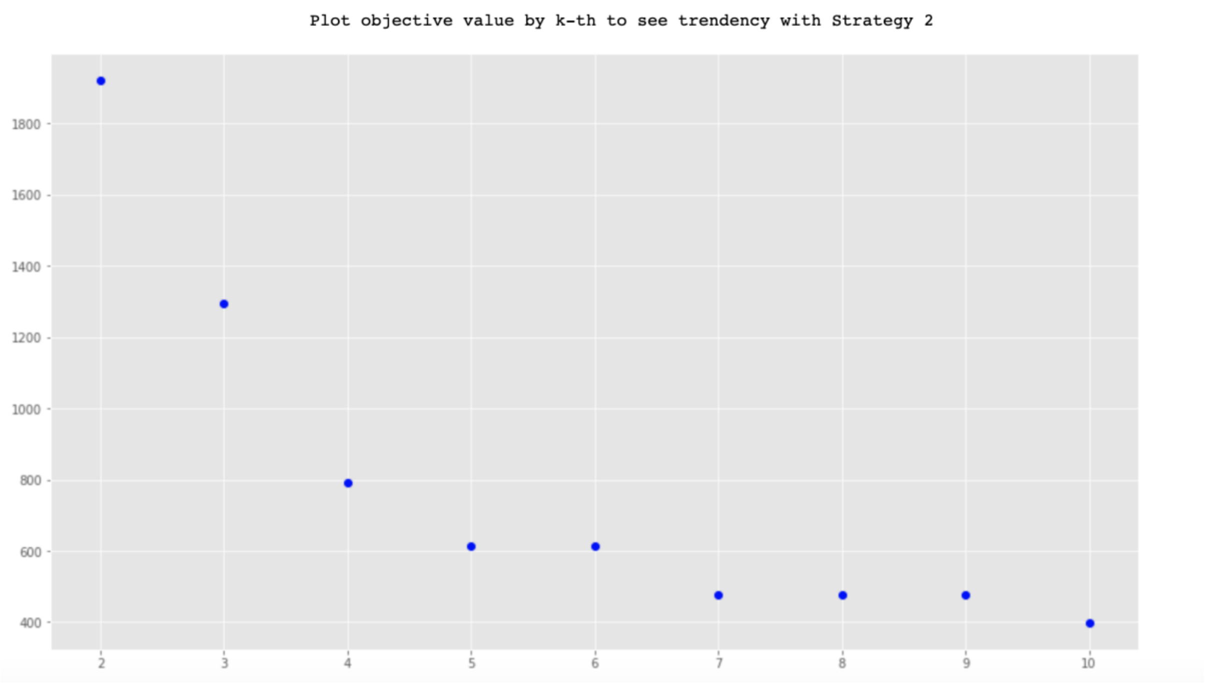
As we see, these two plots are shown that objective value are decreasing when the number of k increase. The objective value dramatically drops when k=3 and k=4. Then the rate of decreasing is lower than before. I think it matches what is said in the lecture video that cost function drops dramatically at some points. If k=1, error is the variance of the samples. If k=n, the error can become 0.

For strategy 2:

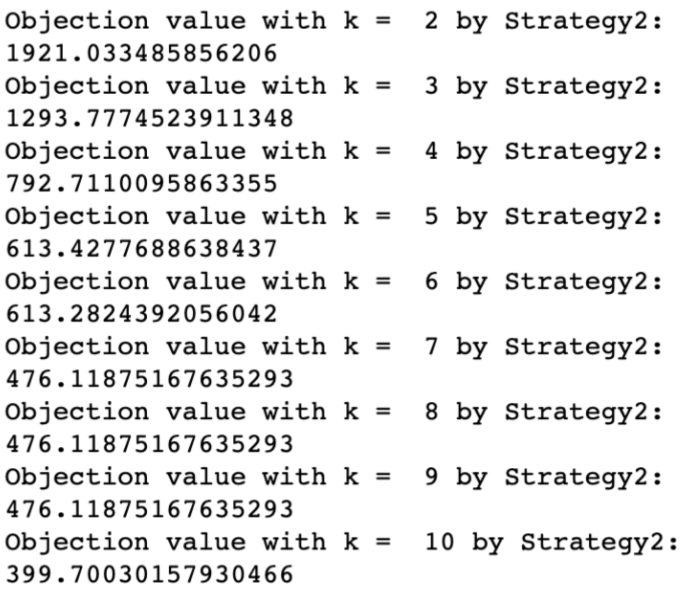
The result for the first initialization:


Plot objective value by k-th to see trendency with Strategy 2

The result for the second initialization:

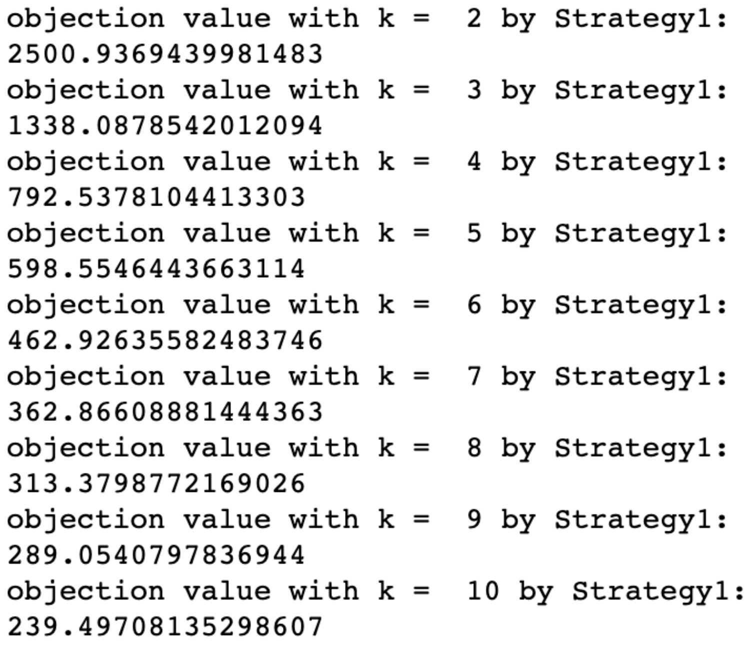


To compare these two plots, we can see the objective value is decreasing when the number of k are increasing. They are both dramatically drops when k=3 and k=4. **However, some objective values are keeping same for strategy 2**. I would like to list the example of objective value of second initialization by strategy2:



```
Objection value with k =  2 by Strategy2:
1921.033485856206
Objection value with k =  3 by Strategy2:
1293.7774523911348
Objection value with k =  4 by Strategy2:
792.7110095863355
Objection value with k =  5 by Strategy2:
613.4277688638437
Objection value with k =  6 by Strategy2:
613.2824392056042
Objection value with k =  7 by Strategy2:
476.11875167635293
Objection value with k =  8 by Strategy2:
476.11875167635293
Objection value with k =  9 by Strategy2:
476.11875167635293
Objection value with k =  10 by Strategy2:
399.70030157930466
```

We can see the value are same when k=7,8,9. After I see this happened, I just go back to check the value from strategy 1:



```
objection value with k =  2 by Strategy1:
2500.9369439981483
objection value with k =  3 by Strategy1:
1338.0878542012094
objection value with k =  4 by Strategy1:
792.5378104413303
objection value with k =  5 by Strategy1:
598.5546443663114
objection value with k =  6 by Strategy1:
462.92635582483746
objection value with k =  7 by Strategy1:
362.86608881444363
objection value with k =  8 by Strategy1:
313.3798772169026
objection value with k =  9 by Strategy1:
289.0540797836944
objection value with k =  10 by Strategy1:
239.49708135298607
```

As the result shows above, there is no same value when k is increasing. So this situation only happened when we calculate the objective value by strategy 2.

What I found:

After I draw the simple test on the paper with 20 discrete points. Firstly, I pick the initial center randomly, then I try to use the alogrithm in the strategy2 to pick points latter. I surprised found that there are some centers are same as of previous centers. Thus, I consider this algorithmcould let me pick the center which occurs before in the record. I think that is the reason why I can get the same obejective values from the different number of k.

To conclude, even the main tendency of cost(objective) function is that value descrease when number of k increase and value will drop dramatically at some points, different ways to pick centers can cause different situtaions. Some values could be same at the adjacent number of ks. Thus, I know the algorithm to choose the initial center for number of k will be an important part in the K-means Unsupervised clusting implementation. I believe there are some ways to improve it, for example, multiple run initial centers and choosing point furthest from the previous centers.