A report on K-means Algorithm

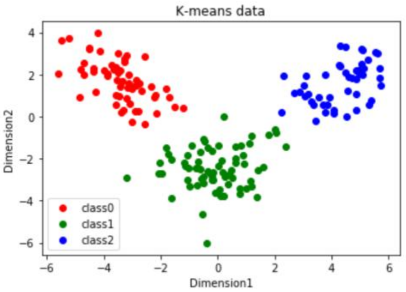
Different initial points

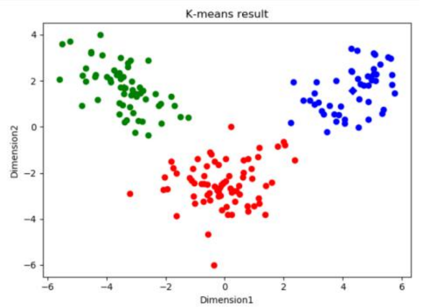
In this part's experiment, I try to set different set of initial points as the start center of the clusters and compare the differences to find some pattern.

I set k=3 as the value of the initial number of cluster centers, and two types of center generation methods, 1) generate from random select points from the data points been given, 2) generate random locations from limited range. I also create a accuracy method by calculating the percentage of the points being correctly clustered.

Generate randomly from data points

I randomly generate 10 sets of initial points from data and all the ACC is equal to 1.0, the ground truth and the cluster result is as follows:





Generate random locations from limited range

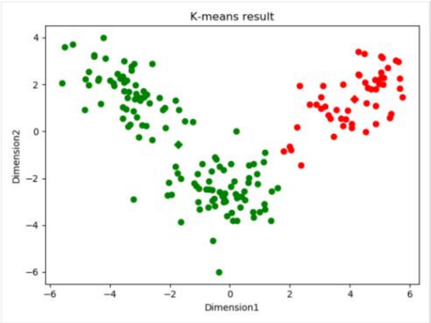
I set the location range to (-10, 10) and random generate 10 sets of points by location. There comes out two kinds of situation:

1) Similar with the result of generating from data points, the ACC is all 1.0 and iteration cycle is similar too.

2) In some cases, there are only two cluster been aggregated while k value is 3. And the accuracy function is fail. For example, the generated center is [6.71528792 -4.76311979], [5.13152893 -9.03344483], [1.02573424 -8.80073907]. The diagram after cluster is shown below.

Once one of the generated center is far from the data points that none of the points will be recognized as this cluster which will cause the location of next iteration will be 'Nan'. And the result will lose one cluster.

After I set the range to (-5, 5), this condition never happened.

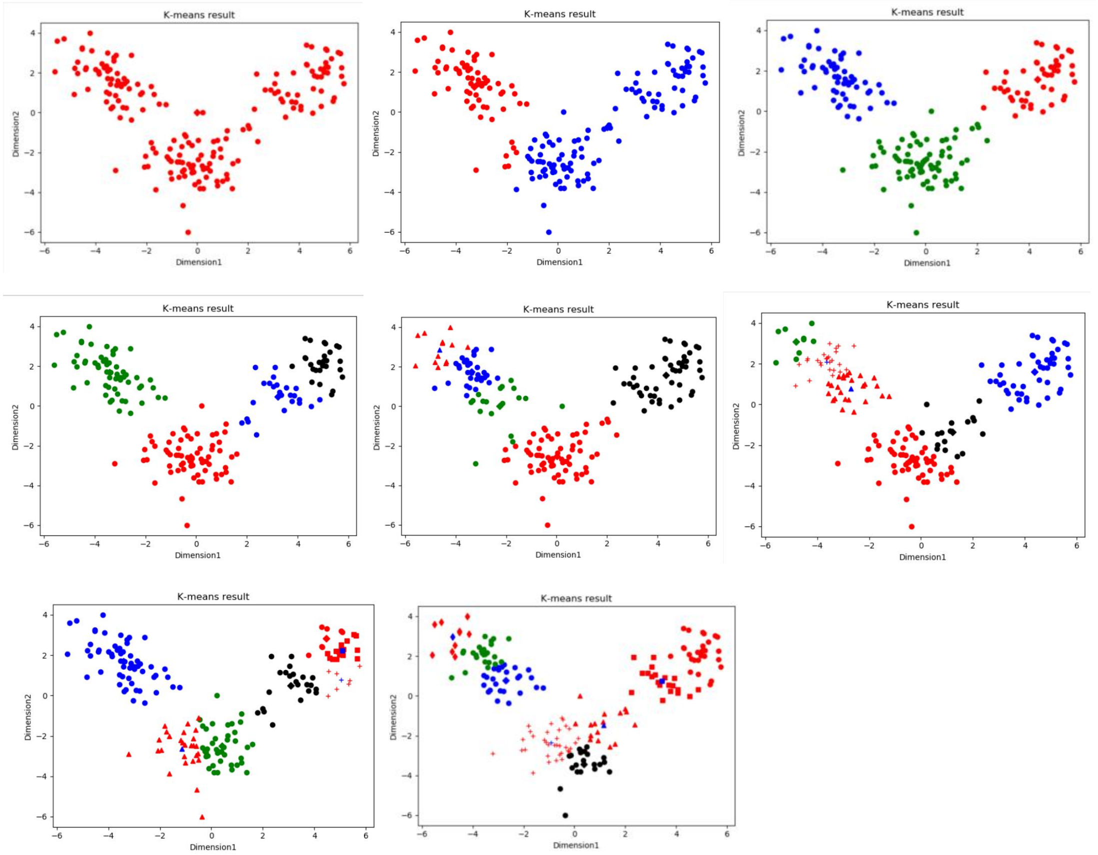


A brief conclusion is it's better to use the nodes in the data as initial points, if we randomly choose location we should not set centers very far from the data points and we should set a good range.

- **Different k value**

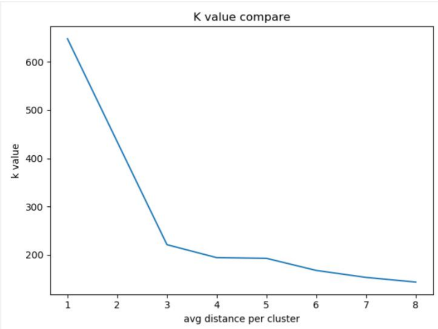
In this part, we try to set different values of k, observe the result and compare it with the ground truth.

✧ I generate the random initial points from data in this part, and I choose the k values from 1 to 8. The results are as follows:



As we can see that with the k goes higher, the big cluster has been cut into more small clusters. Which means the trend of the cluster is the same, they all try to cluster the points who are very close into the same cluster.

✧ Here I calculated the avg distance of each cluster with different k value:



With the k goes higher, the sum of the total distance to the center is getting lower. But as we can see that the clusters are rapidly going tighter as the k values increase from 1 to 3(which is the ground truth), but after 3 the trend is getting slower.

In a conclusion, I think the best k value is the rapid turning point of the tightness. And we shouldn't set the k value too high for the result will be in small broken bits. Not too low for the distance will be large.