Yu Jinghuan
5529 0708
CS 5487 Machine Learning
2018年12月2日

CS5487 Programming Assignment 2

Problem 1

a) Implementation Details

This assignment is implemented by Python, chose numpy, matplotlib and scipy as the basic support library. All three clustering method use *do_the_clustering* method as the packaged interface. This method accept a point array as the input source and an integer to control the position of the plot.

b) Algorithms performance comparison

1. K-Means

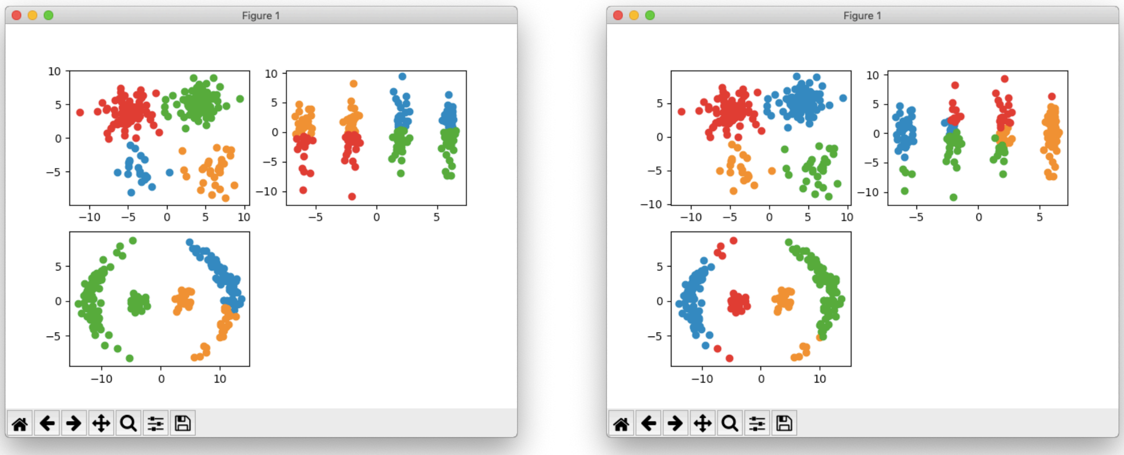

Fig 1. K-means clustering in single loop

Firstly we should notice that K-Means algorithm will lead very different clustering results lot according to the different initialised centroids and iteration steps. And with some extremely situations, the algorithm will deteriorate further due to the bad choice of these centroids.

As for the accuracy level, K-means achieve a good classification result obtained in dataset 1. But as we can see, single processing loop on other datasets are unstable due to the random cluster centres. In the worst case, result are shown in the first subplot of fig1, it can't even divide the dataset C into four clusters, and divide the dataset B into an extremely strange way (though it is reasonable).

To achieve a better result, we can choose better initial centres, or use a loop to search the best clustering result according to the sum of points' distance to each centroids (shown in Fig 2.). But we can see it still has storages in the dataset C.

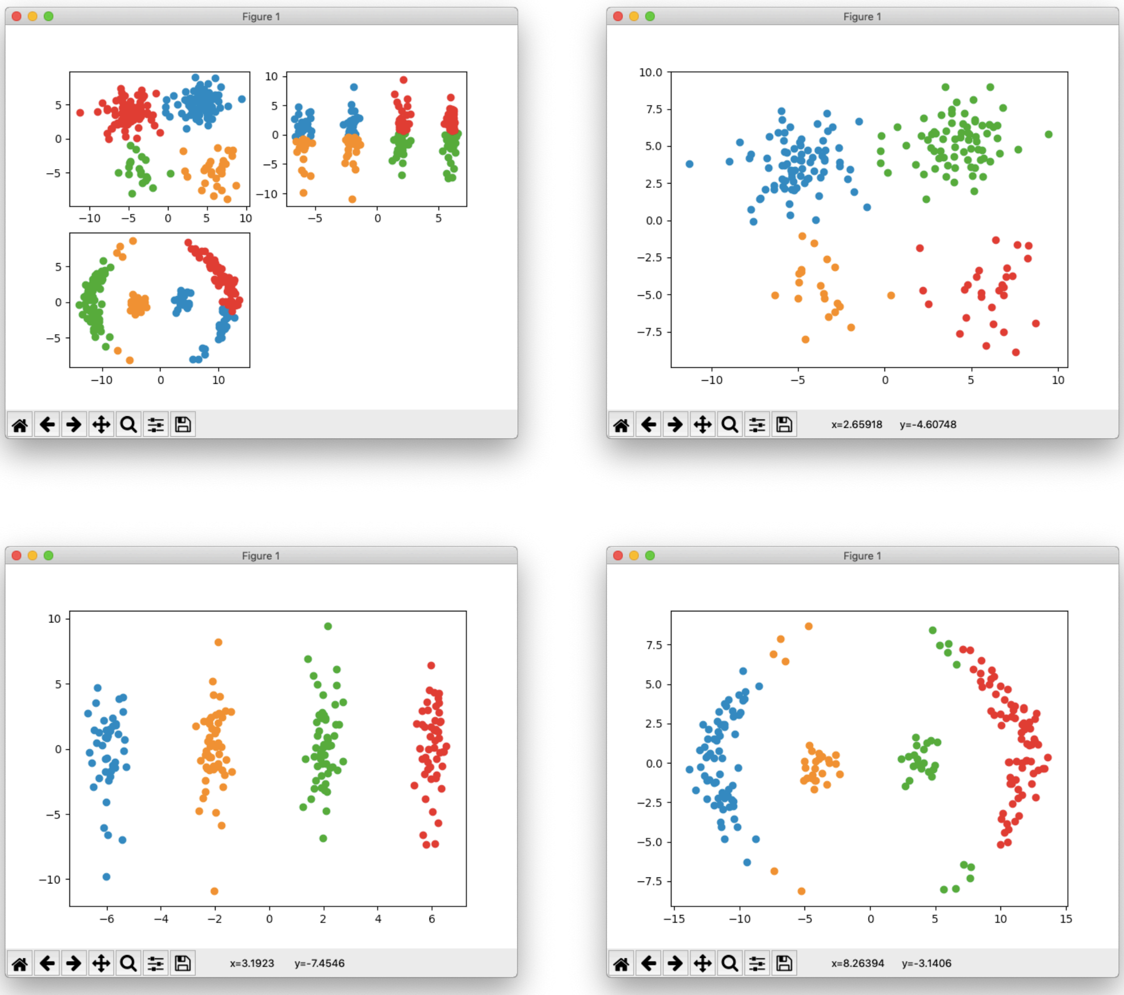

Fig 2. K-means clustering after multiple loop and given initial points

Now we can conclude the K-means algorithm's basic characteristics:

1. This clustering method is very unstable and according its basic principles, result with the least overall distance achieves the best classification.

2. The initial centroids significantly influence the classification result.

3. The classifier can obtain better clustering results in a relatively uniform scatter distributed dataset (like dataset 1), but the aggregation results in other data sets will be greatly deteriorated and extremely unstable (like dataset 2 and dataset 3).

4. The cost of calculation is relatively low, algorithm can get results in dozens of iterations, and in each iteration has less calculating operations than other system.

2. EM-GMM

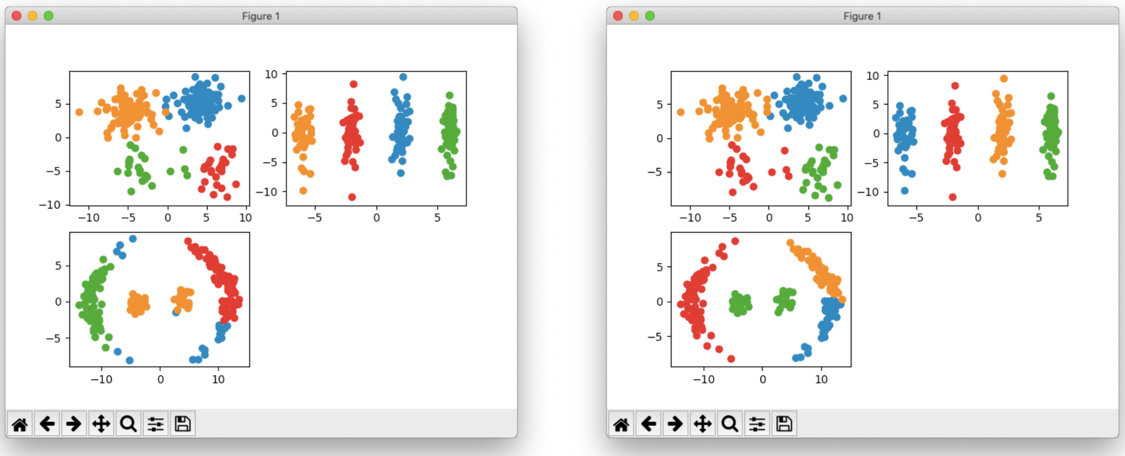

Fig 3 clustering result of EM-GMM

Unlike the K-means, GMM algorithm’s basic idea is to produce some probability density functions instead of simply aggregating the data points. It can gain better performance in obviously separable datasets like dataset2, but will also loss division accuracy in crowed data points due to its original principle is to continually increase the accurateness of the learned functions.

Though EM method will finally lead to the converge result, but it may spending much more than other approaches. Table 4 shows the iteration times to achieve the converge result and the initial centroids. Comparing different datasets, we can see dataset B has better converge speed, but different initial centroids increase about 30% of iteration times. Result is much worse in dataset A and dataset B. In test case 2 and 6, the algorithm can not achieve converge goal in limited iteration times (500 as in design), but with other initial centroids the algorithm give the clustering result within less than half of the limit steps.

The features of EM-GMM clustering can be concluded as following:

1. This clustering method is still unstable, but due to its basic idea of "Soft Assignment", it can produce more information than simply aggregate the points.

2. EM calculates and maximums each sample points’ probability inside different clusters. This make the algorithm very efficient in datasets with obvious margins between each others (like dataset B), but the stability greatly reduced in "crowded" samples like dataset A.

Table 4. Iteration Times in different Dataset

Test Case Id	DataSet	Iteration Times	Initial centroids
1	A	172	[0.56303389 0.49626772] [0.32213992 0.58745174] [0.22526338 0.12486429] [0.77015463 0.57366467]
2	A	499 (Reach Iteration Limit)	[0.53934756 0.29409384] [0.28488332 0.3343725] [0.49313483 0.21865764] [0.22877236 0.65093549]
3	B	119	[0.39654836 0.01571868] [0.0620038 0.3434723] [0.45355632 0.03284486] [0.96860628 0.19596907]
4	B	85	[0.37082128 0.39397164] [0.59593785 0.37226715] [0.52349191 0.59755308] [0.18251294 0.68478942]
5	C	227	[0.64554722 0.76558098] [0.57107227 0.07485928] [0.41730654 0.19331524] [0.35236012 0.48982881]
6	C	499 (Reach Iteration Limit)	[0.67293361 0.51979571] [0.60646405 0.98761799] [0.85503201 0.68206309] [0.73421799 0.83747314]

3. This model will finally reach the converge but may cost many steps. And the initial cluster centre will greatly disturb the clustering result and the speed of clustering.

3. Mean-shift algorithm

The most obvious difference between Mean shift algorithm and other approaches above is the algorithm can not be specified the number of clusters to classify. The results are controlled by the "h", the bandwidth parameter. In this implementation, we choose the origin point as the initial point to avoid the influence of initial parameters.

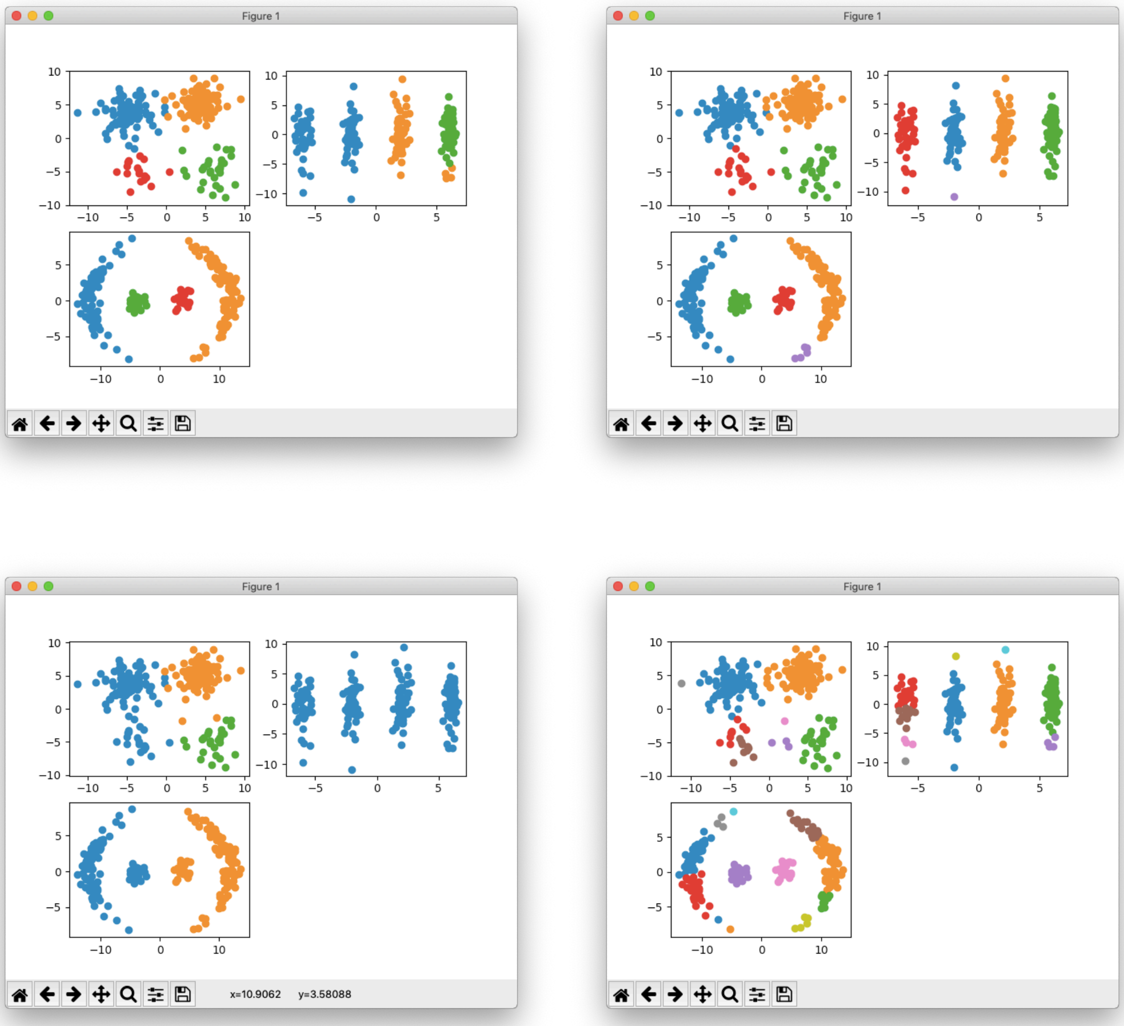

Fig 5. Cluster results with different bandwidth parameters.

Figure 5 shows clustering results applying different bandwidth parameters. The first result uses 0.9 as the bandwidth and the second one uses 1.5. From these two plots we can find out that 1.9 can be a good choice for dataset C, but the step length may be too long for dataset B due to it produces only 3 clusters for the dataset. Relatively, 1.5 is bigger than the ideally band width for it create 5 clusters in dataset B and dataset C. Further, we tired some more extreme bandwidth like 2.9 in the third figure and 0.9 resulting the last one.

Another influencing factor is the initial circle. Fig 6 use four different start circles as the seed of iteration. The first figure is the control groups, the second one shows the result while choosing a big and biased circle, it greatly reduce the clusters and the result has obviously tendency (the first subplot). The next figure reduce the initial circle's radius, it leads a better result but the tendency still exists. The last one uses an unbiased circle, but increases the radius of it. The results are not much different, but points farther away from the core are more likely to be misclassified.

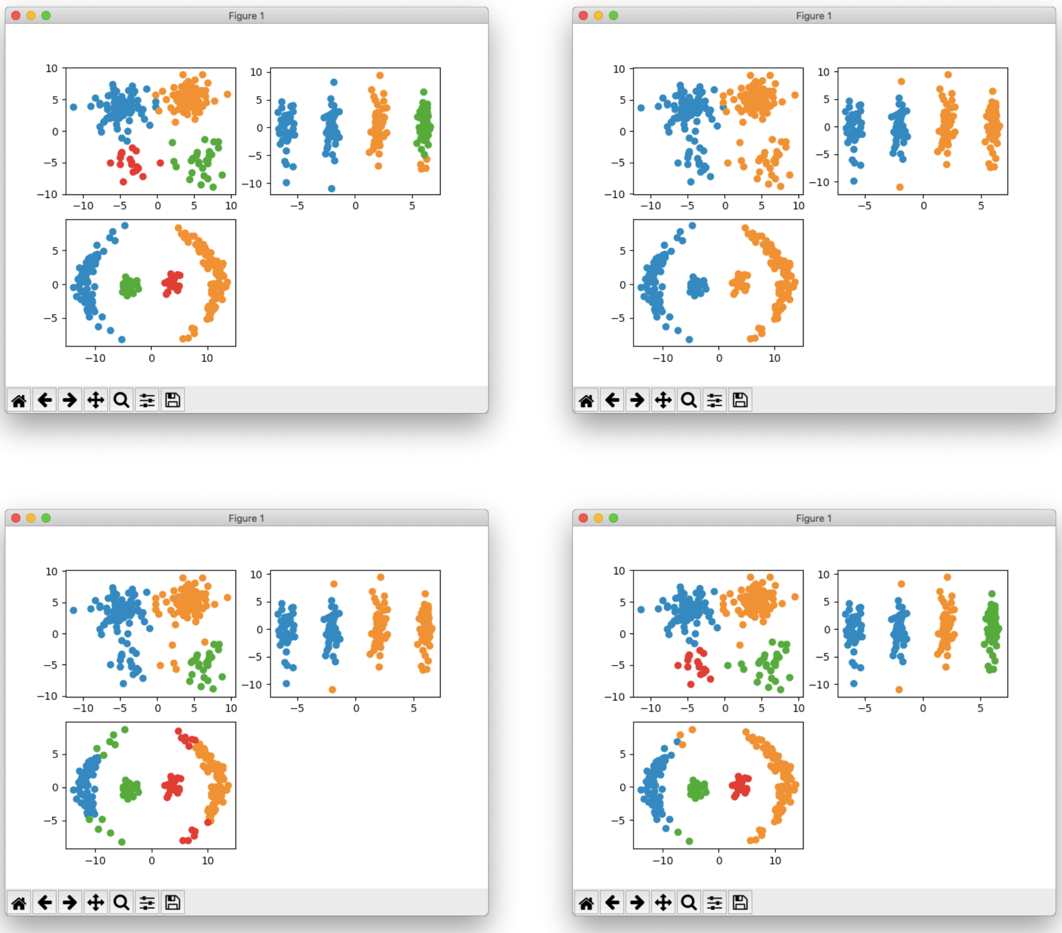
Fig 6. Cluster results with different starting circles

4. Comparison

K-means has the least iteration steps, suits the first dataset most. While giving the correct centroids can perform well on the second dataset, but is hard to classify the third dataset.

EM-GMM is much more stable than the K-means for it will finally converge, different centroid may lead to classifications with different complexity, but the algorithm can still keep relatively consistent results. This algorithm can get the best performance on the second dataset, but can’t behave well on the other two datasets especially when the initial points are chosen incorrectly.

Mean-Shift algorithm may be the best choice for the third dataset while choosing the 1.9 as the bandwidth. But this algorithm has the problem of over-fitting and under-fitting, when the bandwidth is too small, it will produce too much clusters and get too few clusters when it is too big.

c) Sensitive of bandwidth parameter

Table 7. Different Band Width on each data set

Test Case Id	DataSet	H	Iteration Times	Clusters
1	A	1.9	25	4
2		1.5	36	4
3		2.9	22	3
4		0.9	78	8
5	B	1.9	169	3
6		1.5	51	5
7		2.9	155	2
8		0.9	99	11
9	C	1.9	74	4
10		1.5	217	5
11		2.9	28	3
12		0.9	93	12

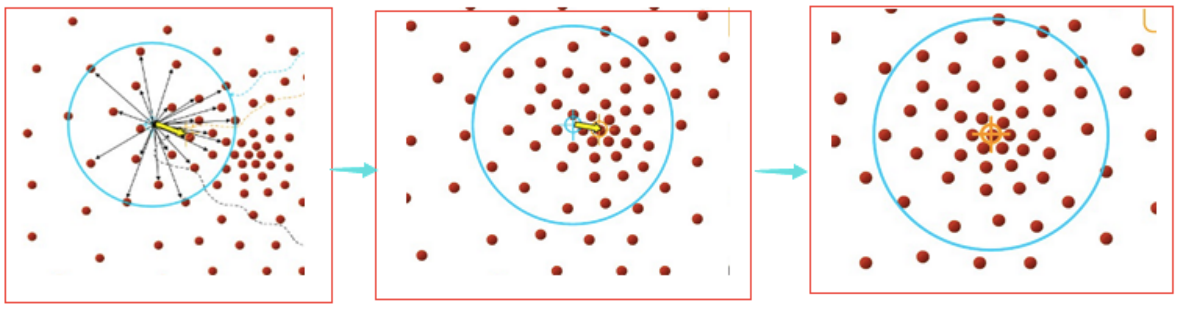

Table 7 counts the iteration times according to different bandwidth parameters and datasets. From the above result, we can conclude the usage of bandwidth parameters, it controls the length of each step when shifting the circle. It can be thought of as a basic unit of clustering in an image. The larger the h is, the easier it is to ignore the details of the image, resulting in under-segmentation; When h is too small, it will lead to excessive segmentation.

Problem 2

a) Use the three clustering algorithms to segment a few of the provided images.

Library Reference

As for the implementation of my program is simple and using single thread to do the entire calculating workload, it can not be applied on these pictures, so I use *sklearn* library as the replacement in some test cases.

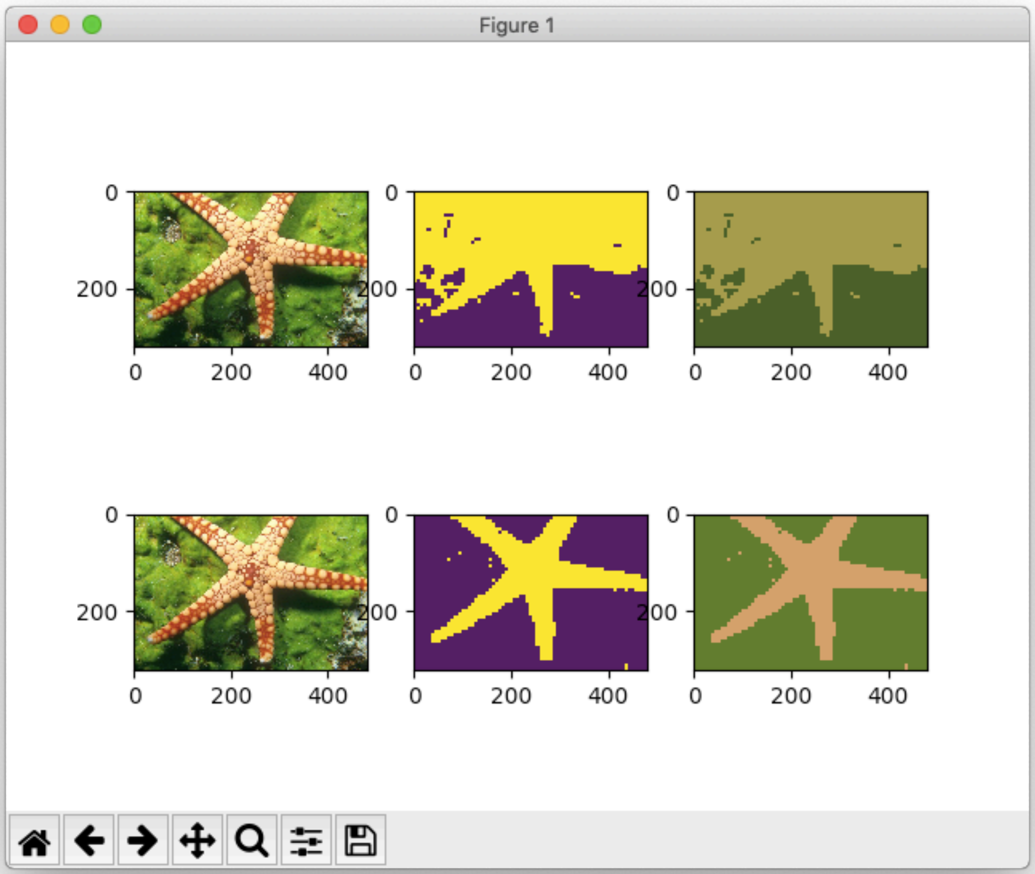

Fig 1. Using K-means and EM-GMM algorithm using K=2

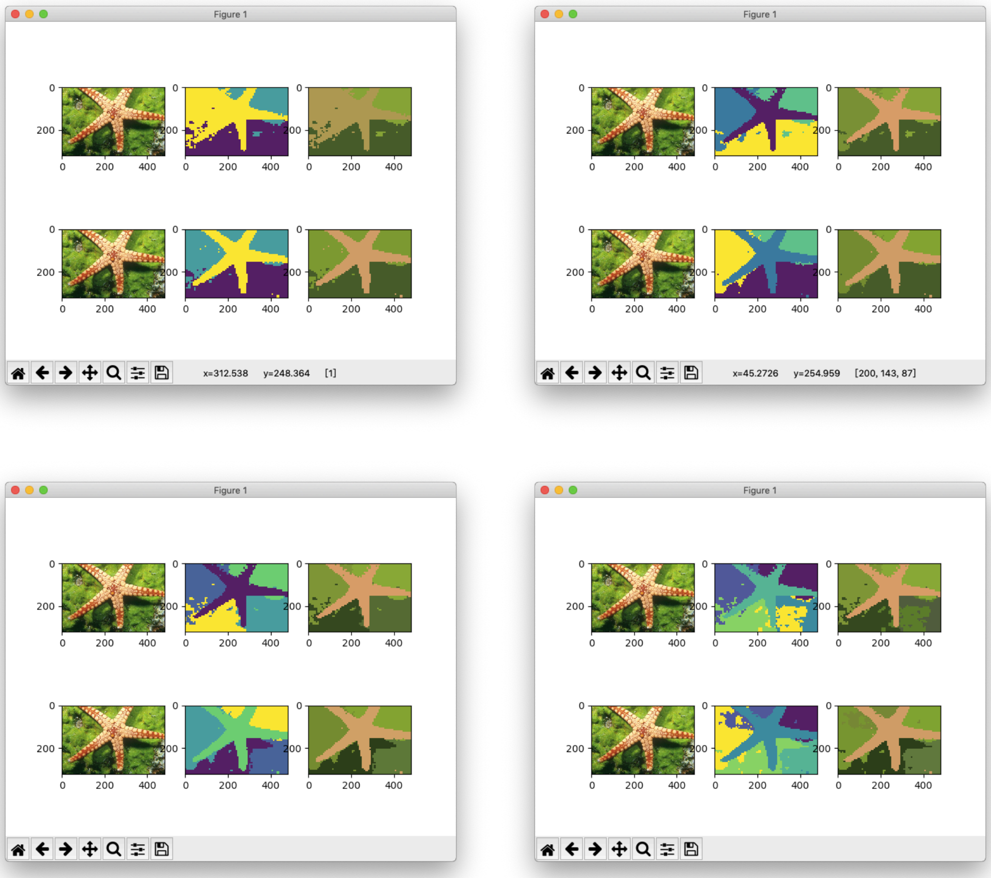

Fig 2. Using K-means and EM-GMM algorithm using K=3~6

As we can see from the Fig 1 and Fig 2, with the increasing of K. Different algorithms achieves different goals.

As for K-means, the boundary of the central starfish has become more apparent when the K transfer 2 to 3. But with the increase of K, the boundary doesn't change much. Relatively, other colour difference due to depth of field become more and more obviously.

As for EM-GMM, the boundary of the starfish is very clear at first (K = 2) and with the increasing K, the model is also keeping it steady. This means the extraction of image's feature is very successful.

As for Mean Shift, the boundary of the starfish is very clear when use 1.8 as the bandwidth. But it is very sensitive. When it changes to 0.9, the boundary is composed of multiple parts, and when it changes to 2.7, the training set can not give out any effective feedback.

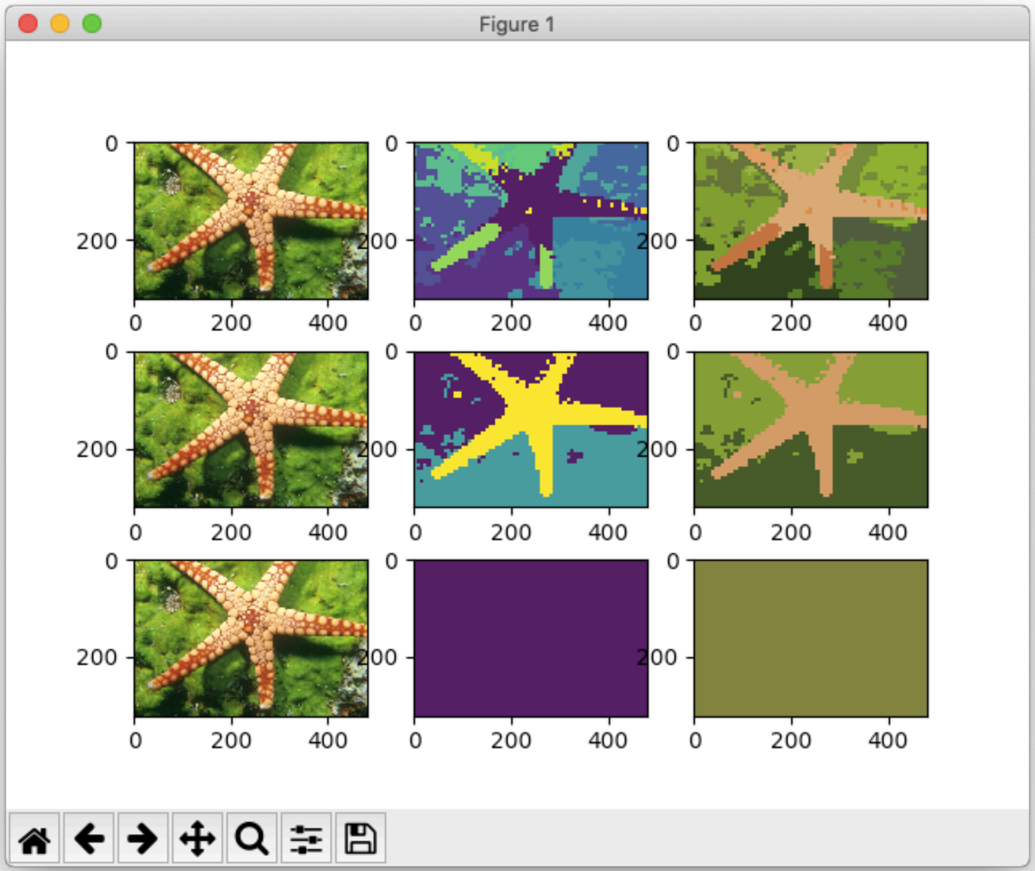

Fig 3. Mean Shift Algorithm's result using 0.9 (first line), 1.8 (second line) and 2.7 (third line) as bandwidth

Now we can give out a conclusion of sensitivity:

K in EM-GMM < K in K-means < h

Other comparison groups to support my opinion:

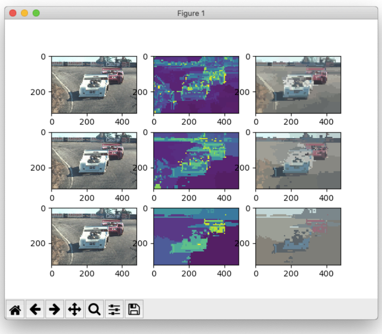
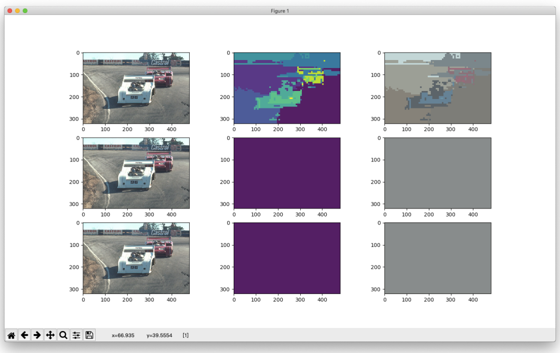

Fig 4. Mean Shift Algorithm's result using 0.3 (first line), 0.6 (second line), 0.9 (third line and fourth line) , 1.8 (fifth line) and 2.7 (sixth line) as bandwidth on 21077.jpg

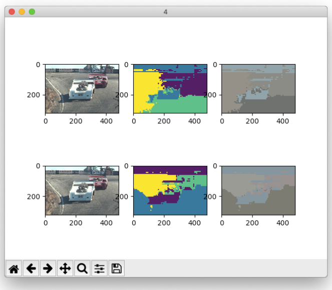
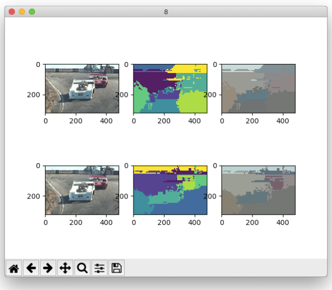
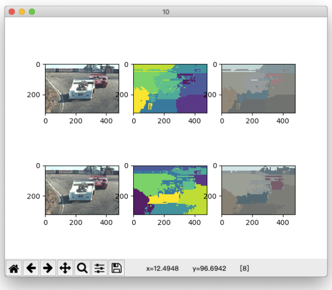

Fig 5. Using different K(4,8,10) on clustering the 21077.jpg

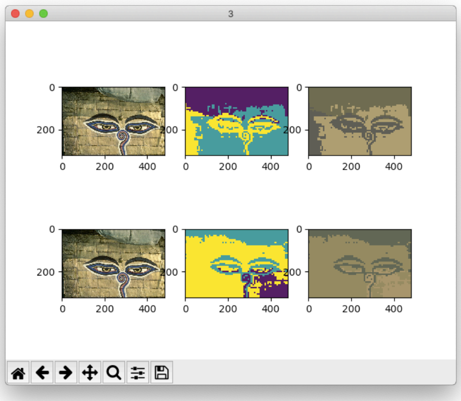
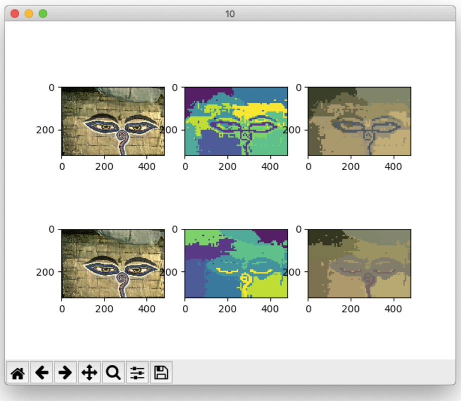

Fig 6. Using different K on clustering the 56028.jpg

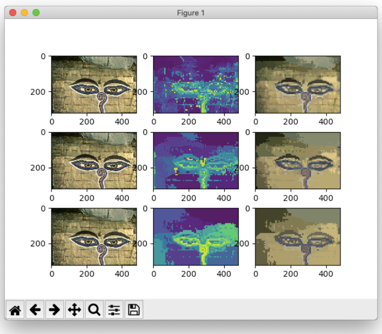
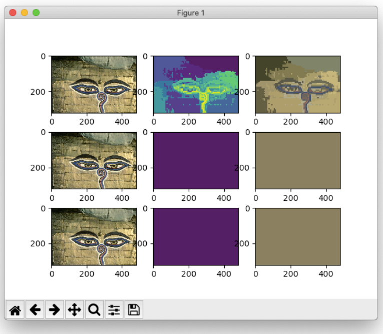

Fig 7. Using different bandwidth on clustering the 56028.jpg

Comments:

There is a strange result when I was applying my implementation to the training set as K = 2, it gave a result like Fig 8. Though I can not recurrent it no matter how I try, it is still an interesting result. It means even in low cluster numbers, K-means algorithm still can provide a precise result only if there is a good starting point

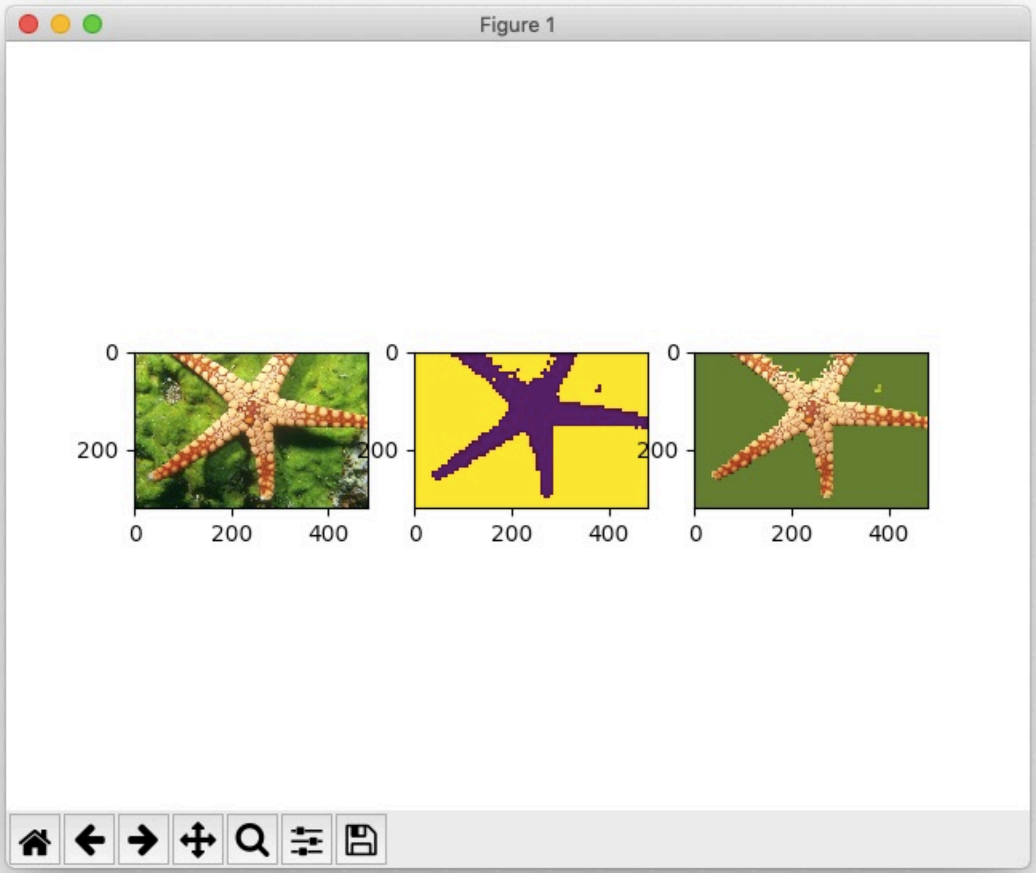

Fig 8. An interesting Clustering result using my K-means Implementation