# Programming Assignment 2 - Clustering

Yang Ji
Student No. 56064832

## 1 Clustering synthetic data

### 1.a Implementation

Source code can be found at `https://github.com/yangji12138/machine-learning/tree/master/Programming2`

**File Structure:**

main.m Main function to display the experiment results

kmean.m kmeans algorithm for $1^{st}$ problem

kmean2.m kmeans algorithm for $2^{nd}$ problem (different distance measurement)

em_gaussian.m EM-GMM algorithm

init_para.m Initialization of parameters in EM-GMM model

calcP.m Calculation of probability function in EM-GMM model

meanshift.m meanshift algorithm

plotf.m Plot Clustering results

### 1.b Running algorithms on the three synthetic data

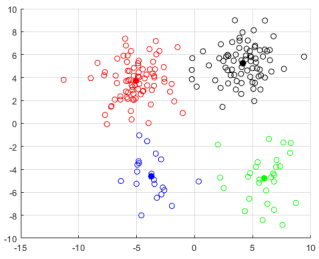


Figure 1: Data_A

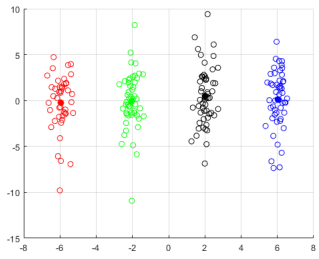


Figure 2: Data_B

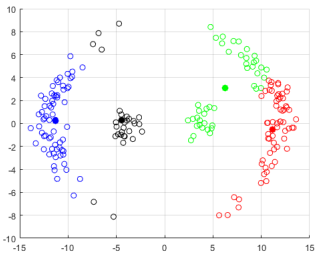


Figure 3: Data_C

Figure 4: K-means Algorithm

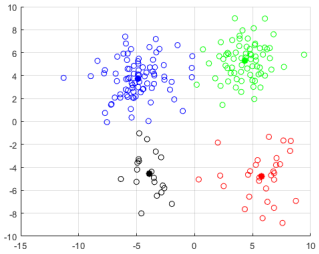


Figure 5: Data_A

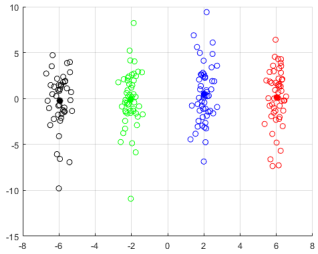


Figure 6: Data_B

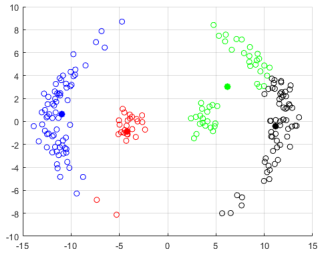


Figure 7: Data_C

Figure 8: EM-GMM Algorithm

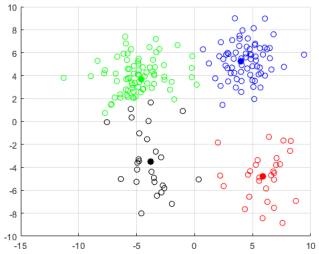


Figure 9: Data_A
$h = 5$

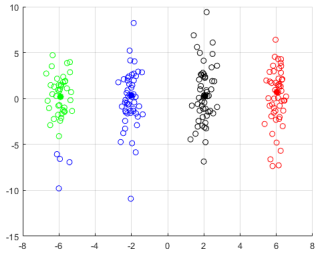


Figure 10: Data_B
$h = 3$

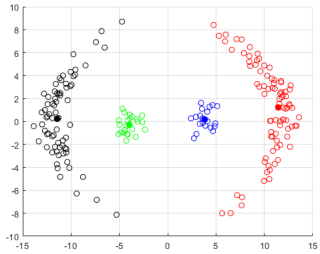


Figure 11: Data_C
$h = 4$

Figure 12: Mean-Shift Algorithm

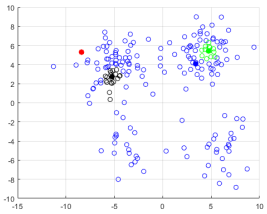
**Conclusion:**

In the experiments, I using the given three algorithms (K-means, EM-GMM and Mean-Shift) to do the clustering tasks. More specifically, K-means and EM-GMM are parametric clustering and mean-shift is non-parametric clustering method. In each figure, I also utilize block spots to mark the center of each cluster.

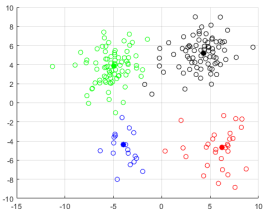
Observing the experiment results, I find that the performance of K-means and EM-GMM is similar. They both can correctly cluster the data points into four classes. And the center point of each cluster can truly reflect the character of each cluster. However, compared to the original figures, these two algorithms have some deviations. They fail to discriminate the last column in Data_C. One reason is from different initial sets of data points. Different Initialization points can lead to totally different clustering results.

For non-parametric method (Mean-Shift), we can adjust the parameter (bandwidth h) to do the clustering. It is worth nothing that the black point in the Figure 12 is the local peak point instead of center point. We find that it works very well when we choose a appropriate parameter (bandwidth h). Although it get rid of the choice of parameter, it suffers from the adjustment on bandwidth h. The details will be shown as follows.

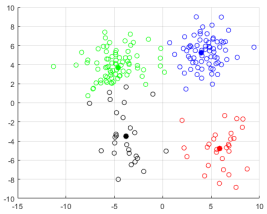
## 1.c Sensitivity of bandwidth h



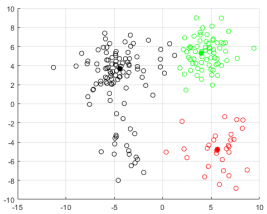
(a) $h = 1$



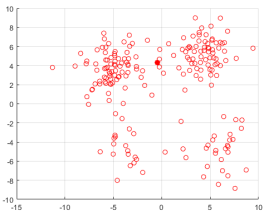
(b) $h = 3$



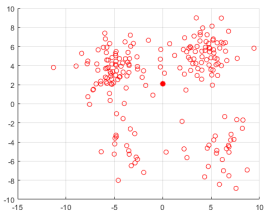
(c) $h = 5$



(d) $h = 6$

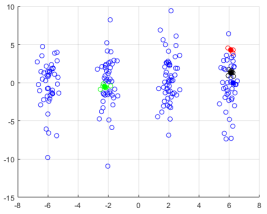


(e) $h = 7$

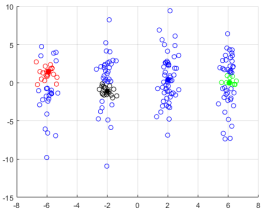


(f) $h = 10$

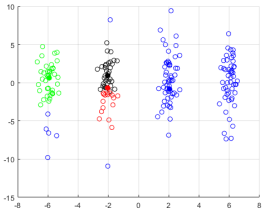
Figure 13: Data_A



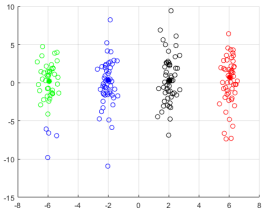
(a) $h = 0.5$



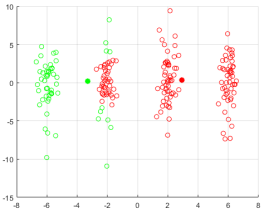
(b) $h = 1$



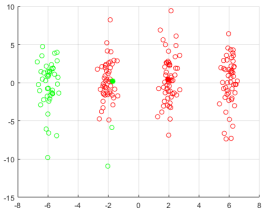
(c) $h = 2$



(d) $h = 3$

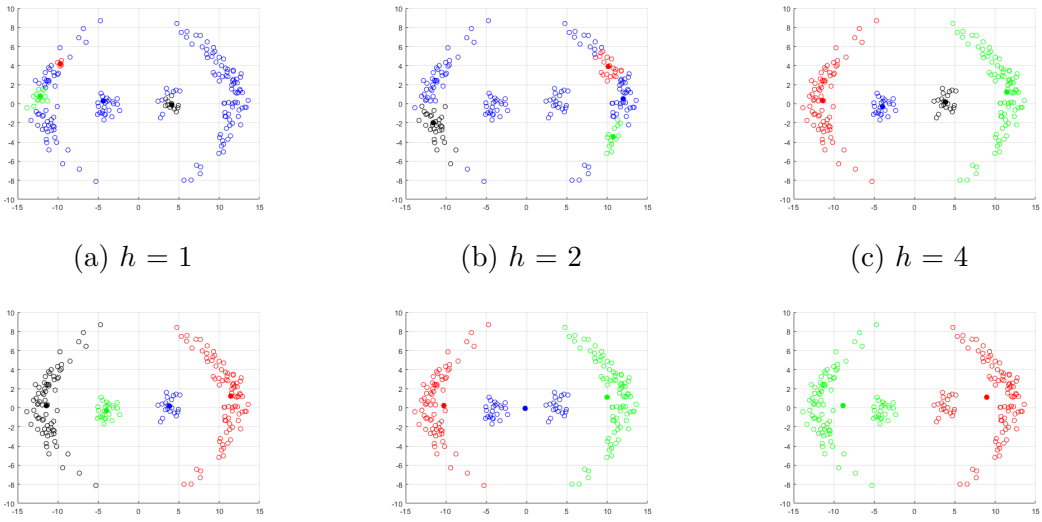


(e) $h = 5$



(f) $h = 7$

Figure 14: Data_B



(a) $h = 1$ (b) $h = 2$ (c) $h = 4$

(d) $h = 5$ (e) $h = 6$ (f) $h = 7$

Figure 15: Data_C

**Conclusion:**

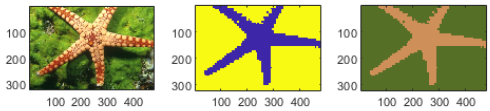
From the perspective of the sensitivity of Mean-Shift bandwidth, I test 6 different values on three datasets. In all three datasets, the choice of bandwidth has the general principles. Take Data_C as an instance. we find that when h = 4 or 5, the clustering results are the best (four distinct results). When the bandwidth is increasing, the cluster number is decreasing into two clusters. Although the cluster number is decreasing, the clustering results are still reflecting the properties of datasets. If the bandwidth is small (e.g. 1,2), it can reflects more details on the data clustering. And it fails to correctly cluster the data sets.

To conclude that, the experiments show that, in order to get better performance on non-parametric clustering, we must know the bandwidth in advance. This is also the limitation of this method.

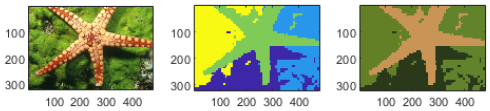
# 2 Image Segmentation

## 2.a Segmentation Examples

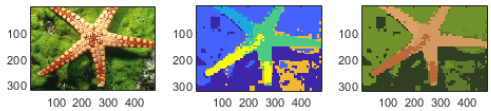
### 2.a.1 Starfish - Image 12033



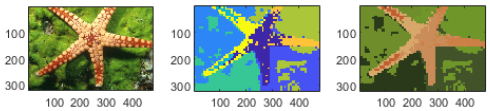

(a) K = 2




(b) K = 4

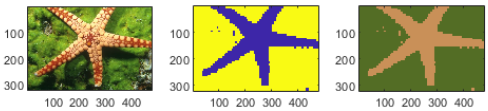



(c) K = 6

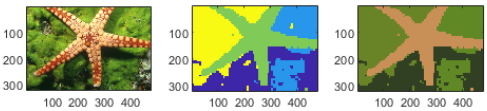



(d) K = 8

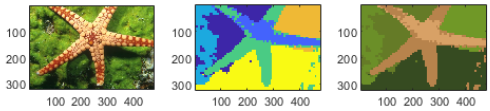
Figure 16: The clustering results using K-means with different k values



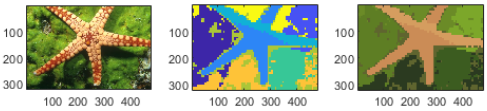

(a) K = 2




(b) K = 4




(c) K = 6




(d) K = 8

Figure 17: The clustering results using EM-GMM with different k values

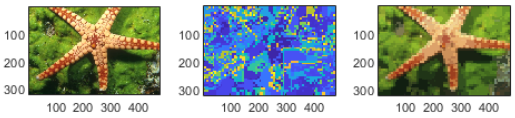



Figure 18: h = 4

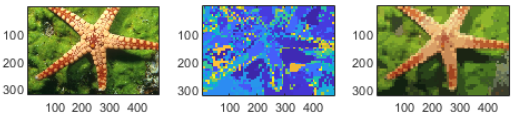



Figure 19: h = 5

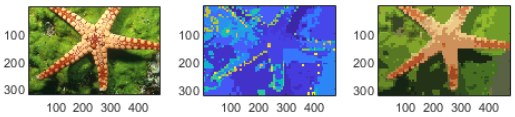
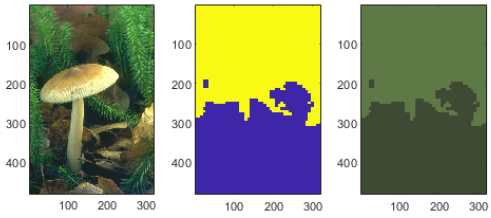


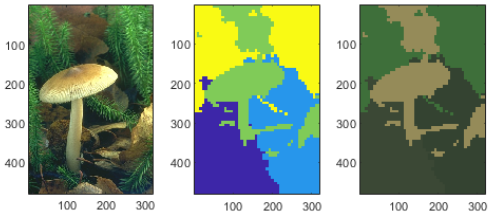

Figure 20: h = 7

Figure 21: The clustering results using Mean-Shift with different bandwidth h values

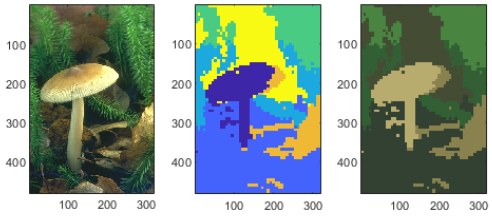
### 2.a.2 Mushroom - Image 208001



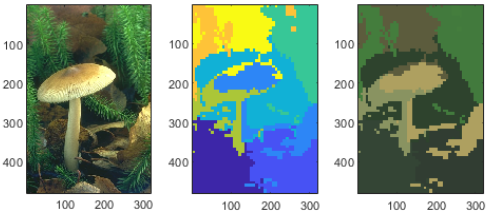
(a) K = 2



(b) K = 4

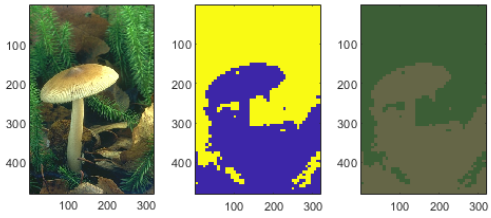


(c) K = 6

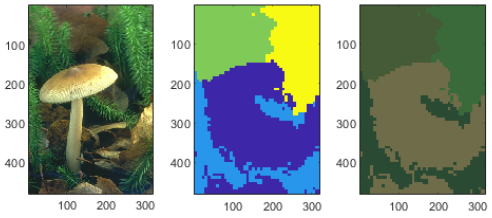


(d) K = 8

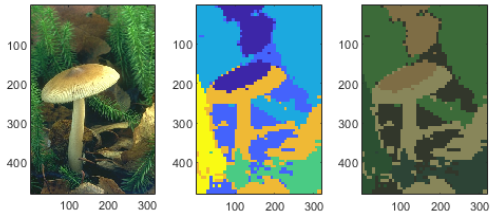
Figure 22: The clustering results using K-means with different k values



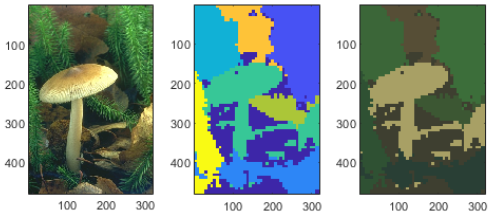

(a) K = 2




(b) K = 4




(c) K = 6




(d) K = 8

Figure 23: The clustering results using EM-GMM with different k values

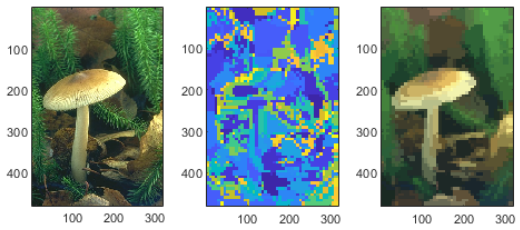



Figure 24: h = 4

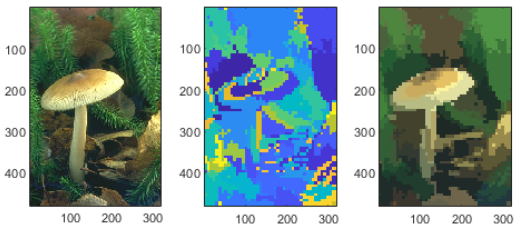



Figure 25: h = 6

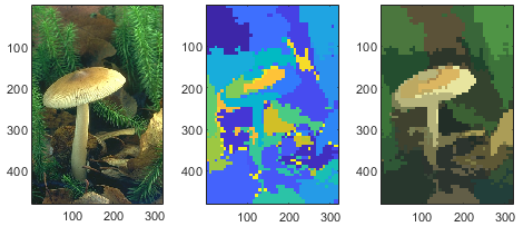



Figure 26: h = 8

Figure 27: The clustering results using Mean-Shift with different bandwidth h values

**Conclusion:**

The picture above shows the evaluation results of three algorithms. What's more, I also adjust the model parameters to make the clustering performance better. Qualitatively, Mean-Shift algorithm has a better performance than EM-GMM and K-means. By adjusting the bandwidth, we can get the best evaluation results using mean-shift algorithm. For Em-GMM and k-means, when we increase the cluster number k, the evaluated results will show more details in both datasets (starfish & Mushroom).

When considering the sensitivity of three algorithms, the mean-shift algorithm is the most sensible. For EM-GMM and k-means, even when cluster number k is very small, the segmentation still can represent the shape of original figures. But when bandwidth h is very small or large,the segmentation totally loses the properties of original pictures.

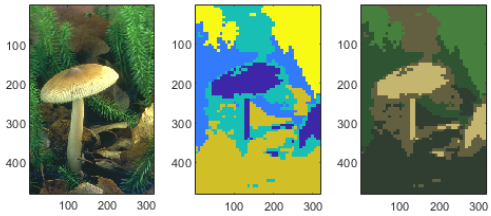
This example could represent the advantages and limitations of mean-shift algorithm. It doesn't rely assume shape on clusters and has only one parameter choice (bandwidth). Therefore, it is a generic technique for clustering and fit into many different models. The main limitation lies on the selection of window size.

For k-means algorithm, despite of its efficiency and simplicity, it lacks consistency, that means clustering results mainly rely on Initialization sets. When I run the k-means program, the running time tends to be very long if the data set is very large.

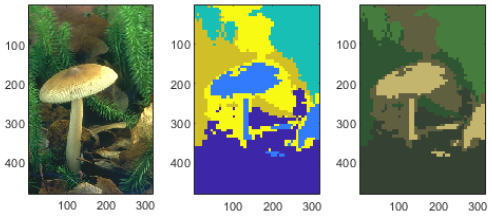
For EM-GMM, it provides more information, including latent information, than k-means algorithm. But it needs more computation time and could easily fall into the local maximum. Also it needs us to specify the clustering number K in advance.

## 2.b Allowing Different Scaling of the Features

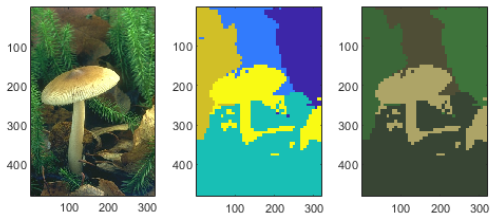
### 2.b.1 K-Means



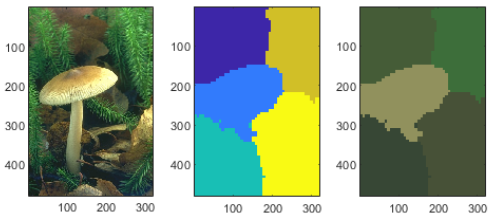

(a) $\lambda = 0.1$




(b) $\lambda = 0.5$



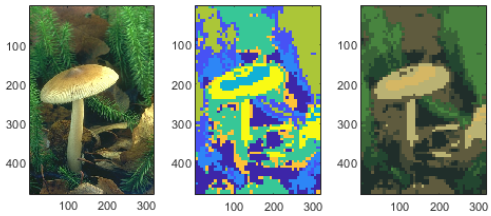

(c) $\lambda = 1$



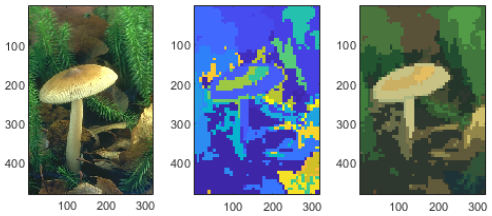

(d) $\lambda = 3$

Figure 28: The clustering results using K-means with different Scaling values, Clustering number k = 5

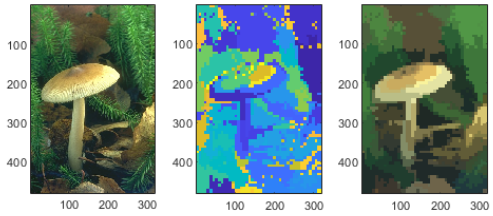
### 2.b.2 Mean-Shift



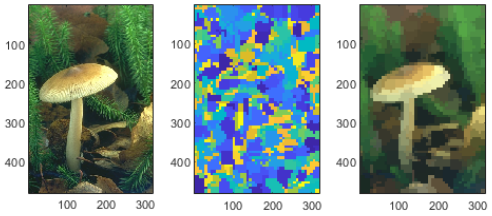
(a) $\lambda = 0.1$



(b) $\lambda = 0.5$



(c) $\lambda = 1$



(d) $\lambda = 3$

Figure 29: Clustering results using Mean-Shift with different Scaling values, bandwidth h = 6

**Conclusion**

Here, $\lambda$ is defined as the ratio of pixel location and chrominance values.

For k-means algorithm, if we add more weights on chrominance values than pixel location, the clustering results could get better. It provides more details, which can't be distinguished when $\lambda = 1$.

For mean-shift algorithm, when we decrease the weights on chrominance values, outputs tend to be more sensitive on pixel locations, which leads to the clustering of nearby data points. When chrominance values raise the weights, it tends to group the data points with the same color (ignoring the location effects). Observing the experiment figures, we can find that the sketch of Mushroom is very clear when $\lambda = 3$, while the sketch tends to be blur when $\lambda = 0.1$.