Answers to questions in Lab 3: Segmentation

Benedith Mulongo, master year 2 Machine learning

September 2019

1 Tasks

Question 1

How did you initialize the clustering process and why do you believe this was a good method of doing it?

I have used two different initialization problems, a random initialization and kmeans++ initialization. Kmeans++ is a initialization method proposed by two researchers David Arthur and Sergei Vassilvitskii in 2007. the algorithm is guaranteed to find a solution that is O(log k) competitive to the optimal k-means solution.

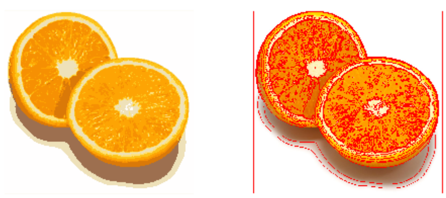

Figure 1: Segmentation with random initialization

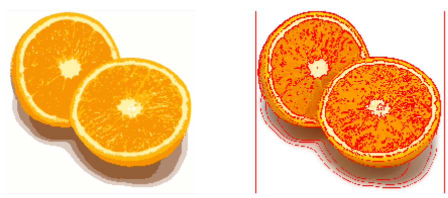

Figure 2: Segmentation with kmeans++ initialization

- **The effect of K**

  The effect of K is to increment the number of segmentation group. By increasing the number of K, we increase the level of details and more groups for each pixels as the figure above show, there is more level curves with increasing number of K components.

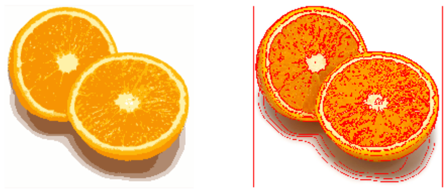

Figure 3: Segmentation with kmeans++ init K = 8

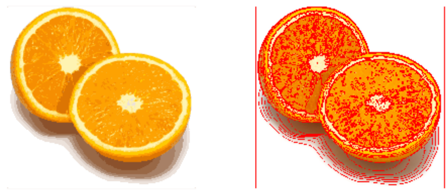

Figure 4: Segmentation with random init K = 16

- **The effect of L**

  The parameter L decides the number of iterations of the K-means algorithm. A smaller K gives a sub-optimal solutions, with a larger L we increase the probability to converge to a stable solution and possibly

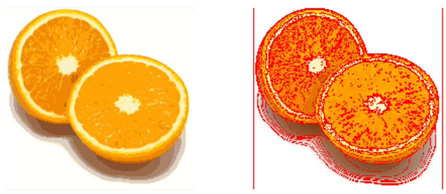

Figure 5: Segmentation++ with K = 8, L = 2

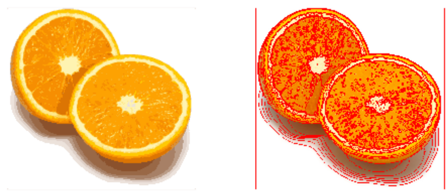

Figure 6: Segmentation++ with K = 8, L = 33

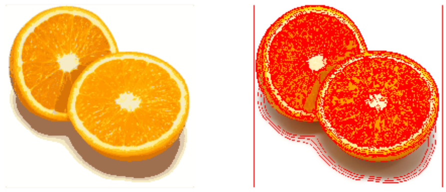

Figure 7: Segmentation random with K = 16, L = 2

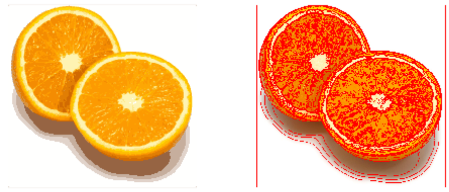

Figure 8: Segmentation++ with K = 16, L = 33

- **The effect of the scale factor**

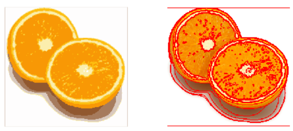

Figure 9: K = 8, L = 3, scale = 0.5

Figure 10: K = 8, L = 3, scale = 1

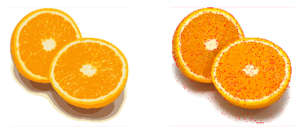

Figure 11: K = 8, L = 3, scale = 5

- **The effect of sigma for blurring**

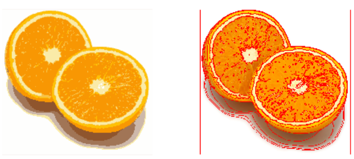

Figure 12: K = 8, L = 3, sigma = 1

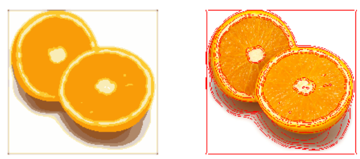

Figure 13: K = 8, L = 3, sigma = 5

Question 2

How many iterations L do you typically need to reach convergence, that is the point where no additional iterations will affect the end results?

Using K-means with random initialization for the orange.jpg image the number of iterations before reaching convergence is 46 and for K-means++ the number of iterations before convergence is 33. There is big difference between the two initialization methods.

Question 3

What is the minimum value for K that you can use and still get no super-pixel that covers parts from both halves of the orange? Illustrate with a figure.

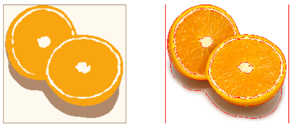

Figure 14: K = 3, L = 3, scale = 0.5

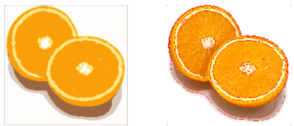

Figure 15: K = 5, L = 3, scale = 1

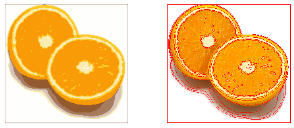

Figure 16: K = 7, L = 3, scale = 5

The effect of parameter K

The best effect is obtained when the number of cluster is between 7 and 8. Experimentally I may argue that the minimum is 7 clusters.

Question 4

What needs to be changed in the parameters to get suitable superpixels for the tiger images as well?

Figure 17: K = 3, L = 4, sigma = 3

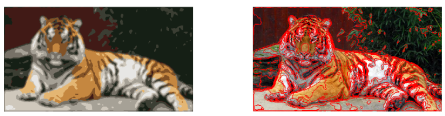

Figure 18: K = 10, L = 34, sigma = 3

The tiger image is more complex with an increasing level of complexity regarding both its structures, variations and color distribution. We need to increase the number of clusters and possibly the number of iterations in order to incorporate the complexity of the image. However as shown in the image above the clusters found are ”correct” but not very clear and distinguishable. Visually a lower number of of clusters give a more clear distinction between clusters.

Question 5

How do the results change depending on the bandwidths ? What settings did you prefer for the different images ? Illustrate with an example image with the parameter that you think are suitable for that image.

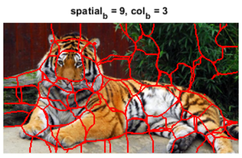

Figure 19: tiger image

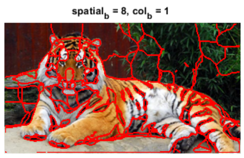

Figure 20: tiger image

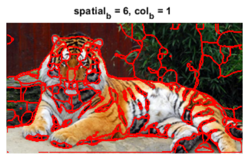

Figure 21: tiger image

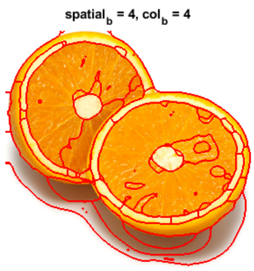

Figure 22: Mean-shift orange

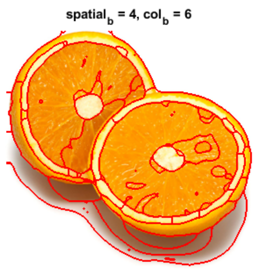

Figure 23: Mean-shift orange

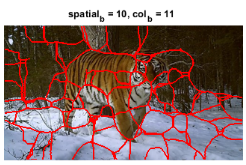

Figure 24: tiger image

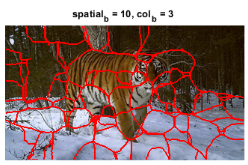

Figure 25: tiger image

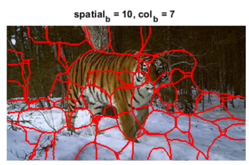

Figure 26: tiger image

What we can observe is that the spatial and colour bandwidths affect the result segmentation image. A smaller bandwidth leads to a more confined and limited number of clusters, a bigger bandwidth creates many more spread clusters.

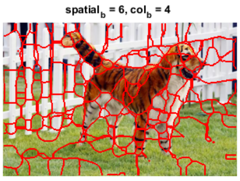

Figure 27: tiger image

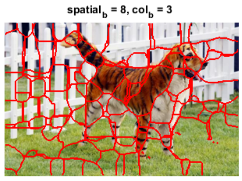

Figure 28: tiger image

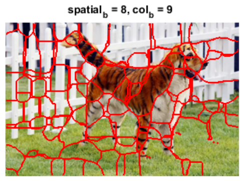

Figure 29: tiger image

The colour bandwidth: The bandwidth determines the radius for the colour space and the resulting smoothness of the segmented image.

Question 6

What kind of similarities and differences do you see between K-means and mean-shift segmentation?

Regarding differences : K-means needs a predefined number of clusters given from the user, however Mean-shift algorithm makes no assumption about the number of cluster this is control by the bandwidth instead. Furthermore K-means does not take into account the spatial domain, only the colour domain, but mean-shit considers both the colour and the spatial domain. K-mean is sensitive to the outliers but a proper initialization method can alleviates the drawback. Mean-shift's output of the algorithm is independent of initialization.

K-means is a parametric methods that assumes that the density functions is super-positions of small Gaussian distributions. However mean-shift is non-parametric and smooths the density function instead in order to finds different peaks and its corresponding feature space regions.

Mean-shit such as K-means is an iterative methods. They are both used for segmentation.

Question 7

Does the ideal parameter setting vary depending on the images? If you look at the images, can you see a reason why the ideal settings might differ? Illustrate with an example image using the parameters you prefer for that image.

Different parameters affect different aspect of the segmented image. The colour bandwidth decides how large the cluster for similar pixels are. A large values decrease the clusters size and leads to many small clusters of similar pixels, small weigth for similar pixels.

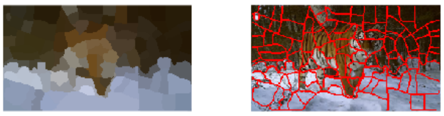

Figure 30: colour bandwidth = 20

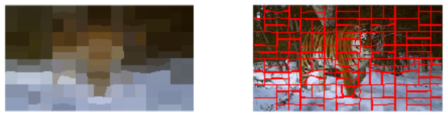

Figure 31: colour bandwidth = 100

Th radius decides the neighbourhood area size, a larger radius will include distant pixels which are less similar leading to a cluster of large size.

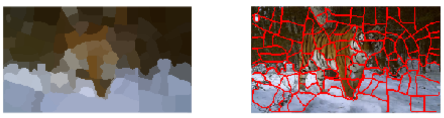

Figure 32: radius = 3

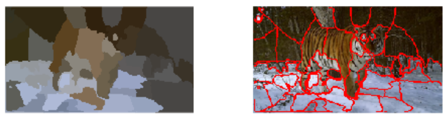

Figure 33: radius = 10

The ncut-thresold The max value (threshold) to keep partitioning. A large value means that we keep partitioning apart further, we lead to a big amount of cluster of small size

The min area sets the minimum limit on how small the cluster can be with respect to the area size. A large value means, cluster of large size.

The max depth sets the amount of recursive calls made by the partition function.

With regarding to the specification of each parameters we can easily conclude that different images may not always match the same parameters setting. Because the cluster size may varies, the colour variation etc.

Best result for tiger image :

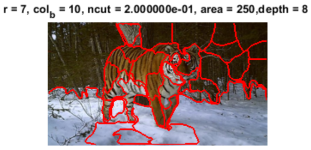

Figure 34: tiger image

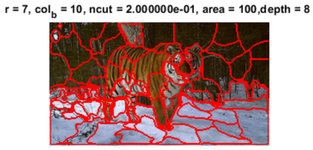

Figure 35: tiger image

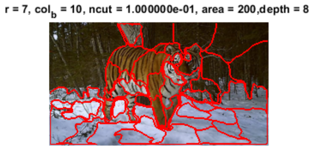

Figure 36: tiger image

Applying the best parameters found for the tiger image, to the the orange image gives a poor result shown in the figure below :

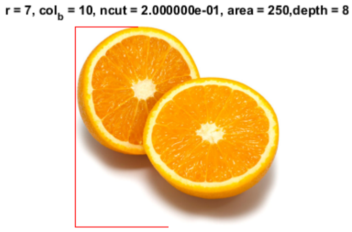

Figure 37: Orange image

Question 8

Which parameter(s) was most effective for reducing the subdivision and still result in a satisfactory segmentation?

- **Min-area**
- **Max-depth**
- **ncut-thresold**

Question 9

Why does Normalized Cut prefer cuts of approximately equal size? Does this happen in practice?

The definition of the objective function used in the Normalized Cut algorithm is following :

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \tag{1}$$

where $assoc(A,V) = assoc(A,A) + cut(A,B)$, minimizing the objective given in equation (1) is the same as making $assoc(A,V) \rightarrow \infty, assoc(A,V) \rightarrow \infty$ and $Ncut(A,B) \rightarrow 0$. Making both association as big as follow will necessarily leads to them have similar size in order to meet the objective function that balance the both values.

It does not alwas happen in pratice because we may have clusters with overlapping vertices and outliers, an assymetric affinity in the graph.

Question 10

Did you manage to increase radius and how did it affect the results?

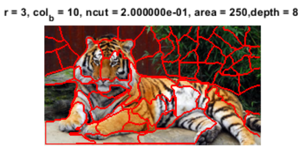

Figure 38: tiger image

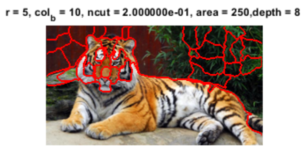

Figure 39: tiger image

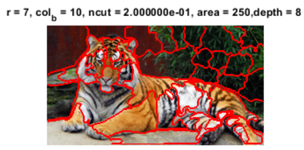

Figure 40: tiger image

Figure 41: tiger image

The radius size of the neighbouring area of a pixel, when we increase the size of the radius we get bigger cuts that includes distant and non-similar pixels. If radius is very big, we get a cut that includes the whole image as shown above.

Question 11

Does the ideal choice of alpha and sigma vary a lot between different images? Illustrate with an example image with the parameters you prefer.

Figure 42: tiger image alpha = 15, sigma = 20

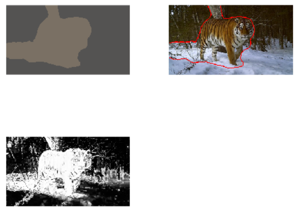

Figure 43: tiger image alpha = 20, sigma = 40

Figure 44: tiger image alpha = 10, sigma = 40

Yes, the choice of alphas and sigmas vary a lot for different images,

Question 12

How much can you lower K until the results get considerably worse?

Figure 45: tiger image K = 1

Figure 46: tiger image K = 2

Figure 47: tiger image K = 3

The lowest we can for the tiger is K = 2. After that the resulting is not good for the tiger data. But that also depends on the image, for the tiger3.jpg K = 1 still gives a good result.

Question 13

Unlike the earlier method Graph Cut segmentation relies on some input from a user for defining a rectangle. Is the benefit you get of this worth the effort? Motivate !

Using a predefined bounding box, we make it easier to initialize the foreground and background, with speed up the convergence of the algorithm and increase the accuracy of the EM-algorithm. However if we have a image that is difficult to segment such as there is no clear foreground and background, then the initialization may not help, but we can still find a better bounding box in order to find a good segmentation.

Question 13

What are the key differences and similarities between the segmentation methods (K-means, Mean-shift, Normalized Cut and energy-based segmentation with Graph Cuts) in this lab? Think carefully!

All the methods are clustering image based on the colour features. Their respective goal is to group similar pixels in one and the same clusters. Graph-cut, K-means are parametric methods and make a Gaussian assumption about the density function. Normalized cut and graph cut are both graphical methods, although normalized cut are a pure application of graph spectral clustering in image segmentation.

Graph needs a predefined user initialization of the foreground and background. Normalized cut analyze the similarity of the affinity and cluster the graph image by cutting edges that have weak affinities. Mean-shift find cluster by analyzing peaks, modes in the density function. K-means makes a Gaussian assumption about the density function and iteratively improve the estimation of the each means.

References