30540 - Mapping from Aerial and Satellite Images
Unsupervised Classification

Anders Richard Pankoke Holm
s144125

Hand-in date: 27/02-2019

When dealing with aerial and satellite image, we would like to "classify" the objects we see on the ground. For example, we see a lot of water, mountains, vegetation as well as fields, which each could serve as a class. Based on the spectral bands of the individual pixels, we are able to compare these and tell which of them are likely to belong to a certain group. In the following assignment, we will be dealing with 2 different kinds of Unsupervised Classification.

K-means Clustering

We are given the data `igalmss`, which contains 4 images in different bands. Each of them are 512×512 pixels. In the image we see clearly see a what resembles water in the form of a river and well as mountains. The rest of the image features are however difficult to identify. We will initially use the K-means method in order to sort each pixel into classes. We chose a number of classes that we deem plausible of the image. MATLAB has a built in function `evalclusters`, which will given a set of suggested number of classes, compute which is the best number of classes for the given image, which is suggested to be 5 using the K-means algorithm. This will be our number of classes moving forward. For each class, we initially assign class centroids (mean point of class μ_c). We then assign each pixel to the centroid with the smallest Euclidean distance to the pixel:

$$D_{Eic}^2 = (\mathbf{x}_i - \mu_c)^T (\mathbf{x}_i - \mu_c)$$

where $i = 1, \ldots, N$ and $c = 1, \ldots, C$ with N being number of pixels and C being the number of classes. $\mathbf{x}_i$ and μ_c both contain the individual points information for the 4 spectra. Hereafter we reiterate, that is compute new centroids from the new clusters, and once more assign the pixels until the result no longer changes. Given that we have 4 spectra, the data could be represented in a 4D hyperspace. If the points are already not circularly clustered in this 4D space, the K-means could have some problems properly assigning the values, meaning that the algorithm is sensible to outliers.

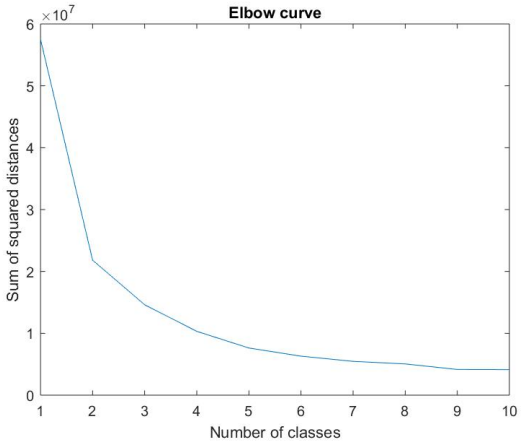

Figure 1: Elbow curve of image data with up to 10 classes

An other measure we could take to determining a proper amount of classes, is to plot the elbow curve (see figure 1). We see the sum of all the squared distances from each iteration with 1 to 10 classes. Every time we add the classes, the sum of squared distances becomes lower, given that there are more points to assign the pixels to. We pick an amount of classes, where adding a new class would hardly reduce the sum of squared distance. Which fits well around 4 or 5 classes. We stick to 5 for the final clustering. Next we will use the MATLAB function `kmeans`, which will assign each pixel to a class using the K-means algorithm. As an extension to this function we will use the "Replicates" option, which will run the K-means algorithm a select number of times and pick the best result, that is the one where the total sum of squared distances is lowest.

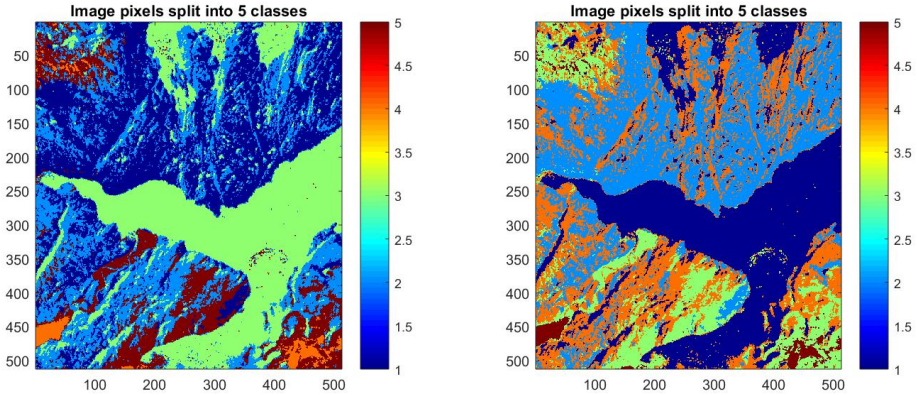

Figure 2: 2 attempts at classifying our image

In figure 2 we see two different classifications of the same image. The same method have been used, that is the `kmeans` function with 5 classes and 10 replicates, but we see that

the results differ. This is likely due to how we initially place our class centroids in model hyperspace. Not only has the order of classes changed, but also some of the pixels assigned to a certain class. Inspecting the classifications, it generally seems that the water and mountains are identified, but for the mountains we have both sides bathed in sunlight and a class for those in the shadow. In the bottom corners we see a clustering that could possibly be clouds.

Investigating Principal Components

As an extra attempt at classifying, we will investigate the Principal Components. We will only look into the first 2 components, since these are most likely to tell us something about the data. From the elbow curve, we get that 4 classes would be a good estimate. The results of the classification can be seen below.

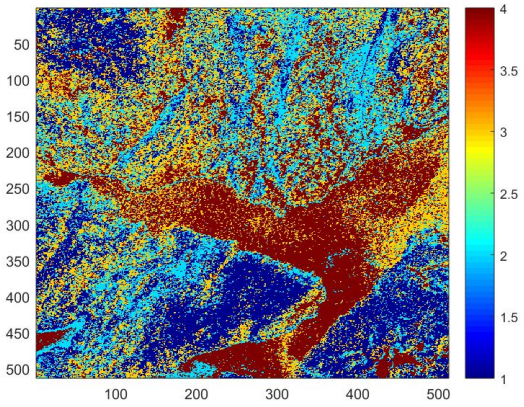

Figure 3: K-means classification using the 2 first Principal Components

We see that using the Principal Components, will give us a very grainy and noisy image, i.e. a poor classification.

Gaussian Mixture Models (GMM)

Just like with the K-means, we use the (GMM) in order to classify each pixel. GMM uses Bayesian principles in order to classify each pixel. Every class is given by a Gaussian distribution and whenever we assign a new pixel to a group, this pixel comes with new information of the group, hence the distribution is updated. Different from the K-means algorithm, (but similar to the Fuzzy C-means, which sorts like the K-means approach), GMM assigns each pixel proportionately to each group. For example we could have a sea shore, that consisted both of water and sand, hence the pixel would partly be assigned to both groups.

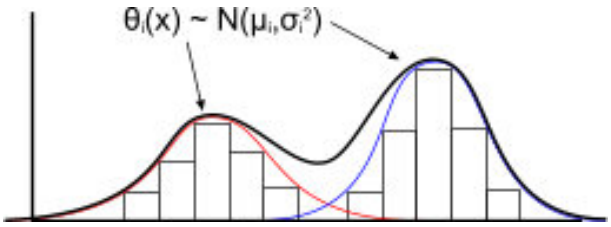

Figure 4: Gaussian distributions for two different classes

On figure 4 we see two Gaussian distributions from a red and blue group. We get the probability density function (pdf) of the thick black line from:

$$f(\mathbf{x}_i) = \pi_1 \phi\theta_1(\mathbf{x}_i) + \pi_2 \phi\theta_2(\mathbf{x}_i)$$

where π_1 and π_2 are the proportions to which the pixel belong to each group. The probability of the pixel being in class c given the observation $\mathbf{x}_i$ is given:

$$P(\omega_c|\mathbf{x}_i) = \frac{\pi_c \phi\theta_i}{\sum_{i=1}^{c} \pi_i \phi\theta_i}$$

Using `evalclusters`, we get that the optimal number of classes is 2, which when plottet is the mountains and the water, which was initially hypothesized. For the sake of displaying the proportional classification, we plot the result using 5 classes.

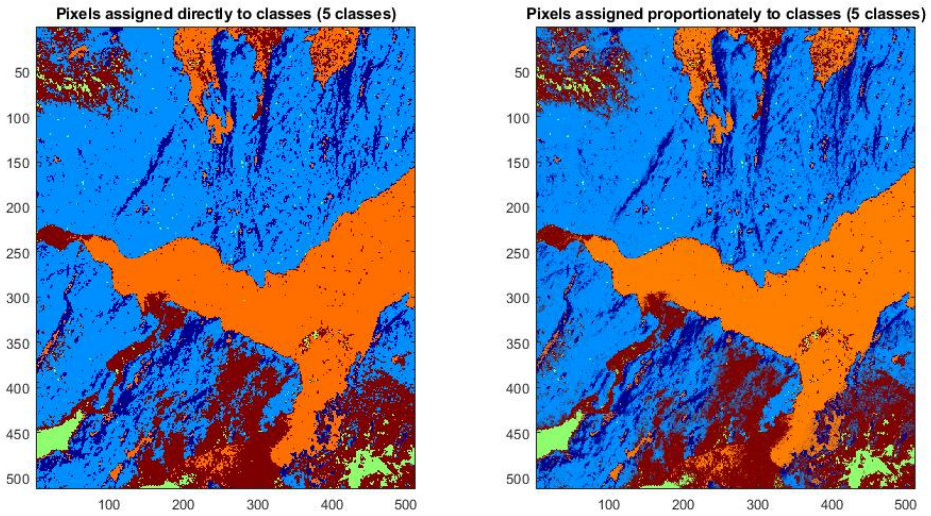

Figure 5: Classification using GMM with 5 classes

In figure 5 on the left we see the pixels assigned directly to the class they are closest to. On the right they are assigned proportionally. The effect of proportional assignment is clearly seen between the dark red and light blue classes.