Introduction to Intelligent Systems
Lab week 2

Diego Velasco Volkmann   Eduardo Faccin Vernier
S2851059   S3012875

October 31, 2015

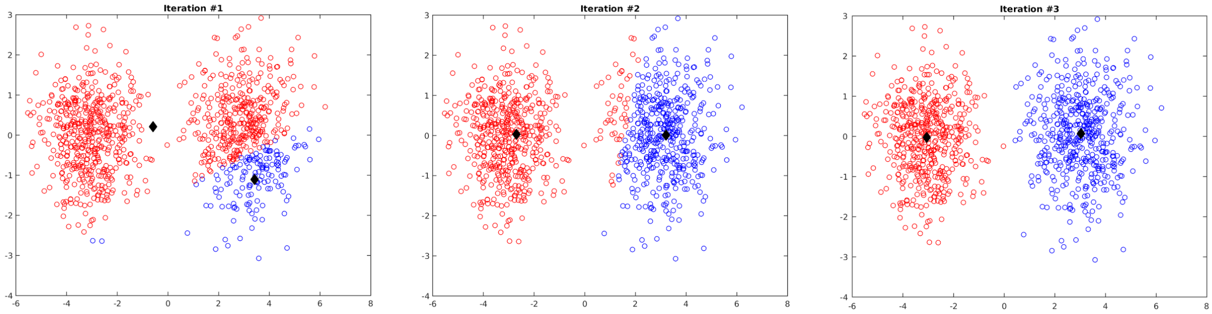
Assignment 1:

Unsupervised learning, K-means clustering algorithm. Implement the K-means algorithm. Show the results in an image (make sure the different clusters can be distinguished easily).

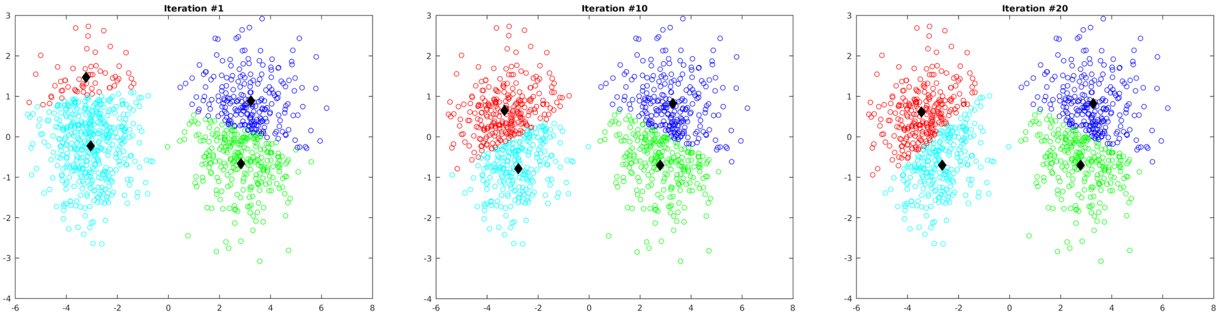
- The file w6_1x.mat contains 2D samples. Apply the k-means clustering algorithm to this data and plot the data in a scatter plot making sure that the different clusters can be easily distinguished. Also plot the cluster-means of the K-means algorithm as it iterates, and make sure the final cluster-means are distinguishable in this plot. Do the steps described above for 2, 4 and 8 means, include the resulting images in your report.

  Answer: These plots were designed so that the black diamonds represent the current cluster mean coordinate in each iteration and each cluster is represented by a different color or marker. The 3 rows of plots below represent 3 executions of the same algorithm in the same dataset with alternating values for the argument k. The first plot of each row shows the initial state of the algorithm, on which k random points are chosen from the dataset to become cluster means. The second plot in each row represents the state of the clusters in the iteration n/2, where n is the number of iterations the algorithm took to stabilize. The last plot in each row represents the final state of the algorithm, which is reached when the set of points that constitutes each cluster doesn't change from iteration n to iteration n+1. The current iteration number is written on top of the plots.

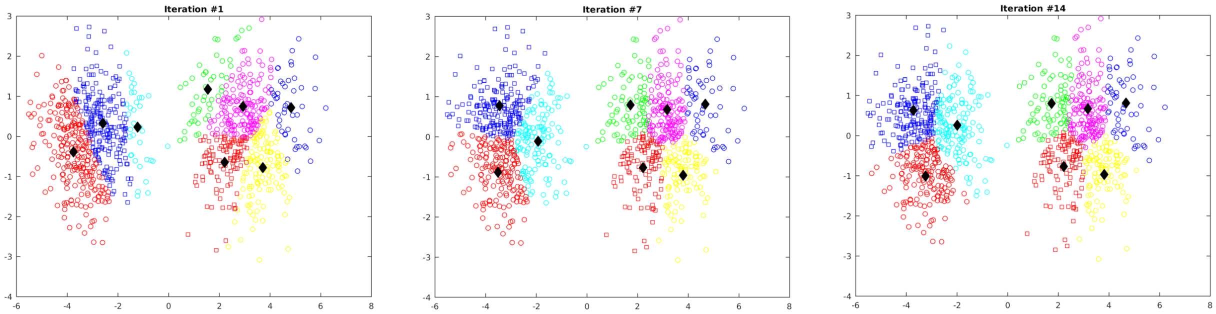
  k = 2 - Stabilized in 4 iterations



k = 4 - Stabilized in 21 iterations

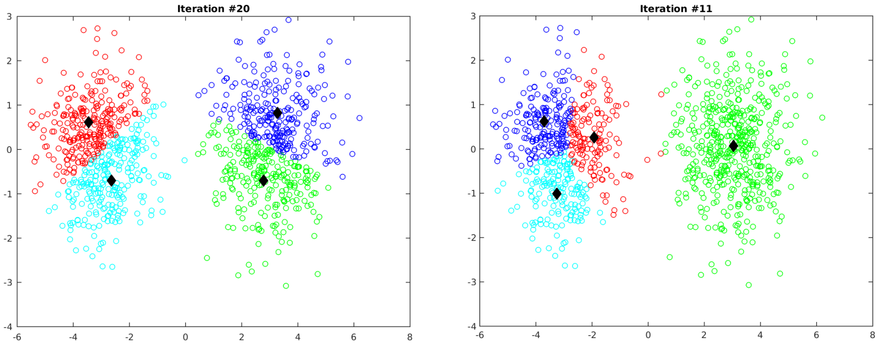


k = 8 - Stabilized in 15 iterations



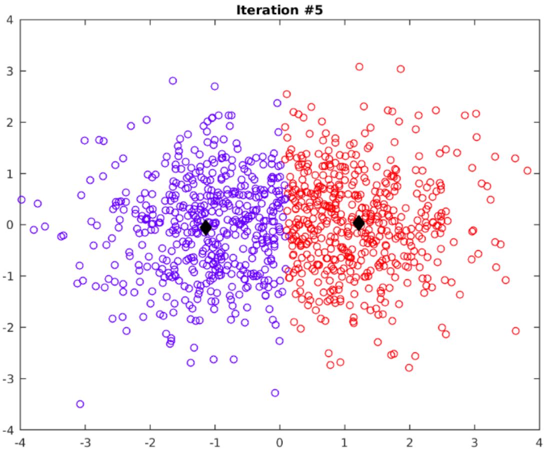
- Run the algorithm again for 2, 4 and 8 means, but this time given the data from w6_1y.mat and w6_1z.mat. You do not have to include figures for all these cases, but run them at least for yourself (a couple of times), try to interpret the results and answer the following questions in your report:
  - a) For w6_1x.mat: explain why the final clusterings are different on each execution, when using more than two clusters (assuming you use different random initial cluster-means each time).

    Answer: Because there are only two very distinguishable clusters in the dataset, the final clustering for k values bigger than two, should be directly related to the placement of the initial seeds. For k equals 4, for example, if two initial seeds land on each of the two distinguishable clusters, each of those should be divided approximately in half and we should have a final clustering similar to the plot on the left. Otherwise, if 3 ou 4 initial seeds land on the same distinguishable clusters the final result should be similar to the one on the right.



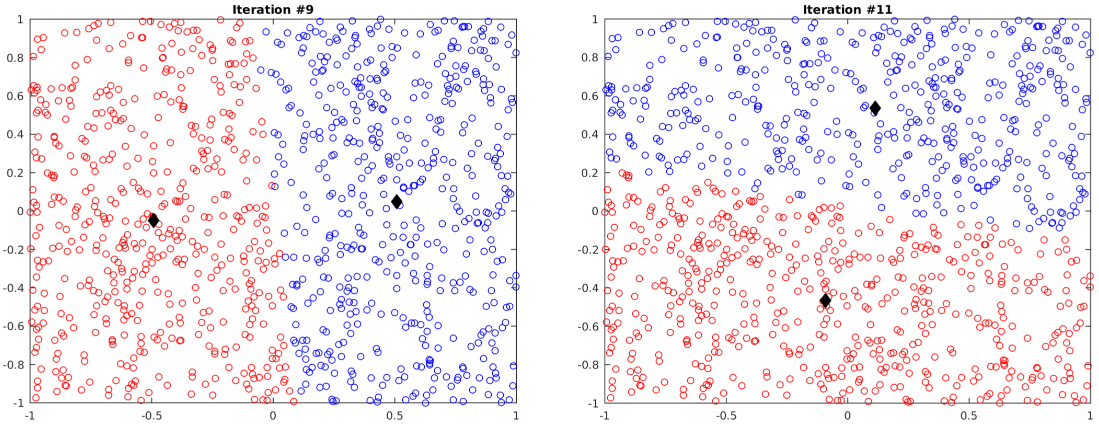
  - b) For w6_1y.mat: for what number of means would you consider the clustering to be stable/predictable? Why? What does this tell you about the distribution that generated these points?

    Answer: The k value that generates me most stable final clustering is 2. Because the points are distributed long along the horizontal axis, so the points are always divided vertically into two clusters, as shown in the plot below. For k values of 4 and 8 the final clusters are not stable and the final result depends on the placement of the initial seeds.

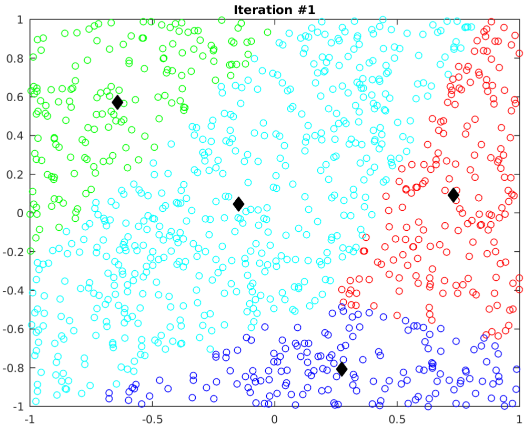


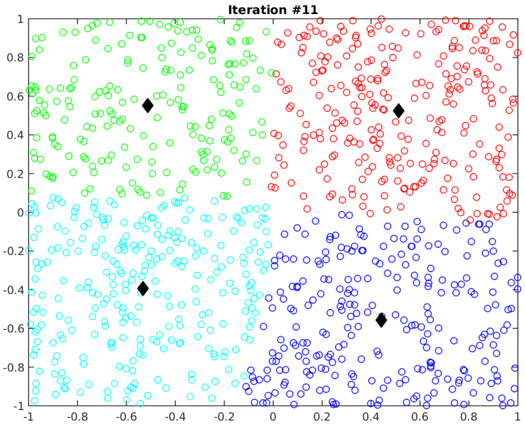
- c) For w6_1z.mat: one could say that the clustering for K = 2 is not predictable, while for K = 4, it is. Explain why and how this is possible.

  Answer: Because the points are so uniformly distributed on the plane, there is no predisposed axis to divides the two clusters, as seen in the previous question. The algorithm behaviour in this dataset is unpredictable, and the axis that divides the dataset in two is determined by the initial seeds.



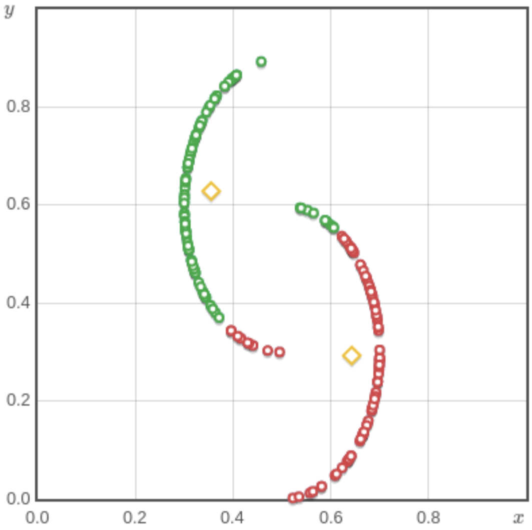
For k equals 4 we can see an interesting behaviour - no matter how the initial seeds are positioned, the end result is always four evenly divided clusters, one in each of the four quadrants. This movement on the seeds is caused by the points in the 4 corners of the plane that pull the mean cluster values towards their directions. If the dataset was distributed in a circular shape, this predisposition wouldn't exist.

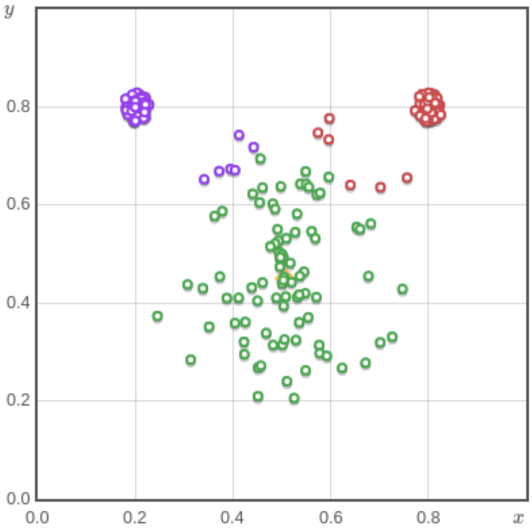




- d) What are your conclusions about the K-means algorithm? When is it useful/not useful?

  Answer: K-means is a very useful algorithm that renders good and consistent results when 2 conditions are met: the clusters must be spherical, and the spread/variance of the clusters is similar. The two images below are examples of bad classification in reason of the breaking of one of these conditions. On the plot on the left, although the clusters have the same scatter (in fact, the same shape), they are not spherical. On the plot on the right, even though the clusters have spherical shapes, they have different scatters.





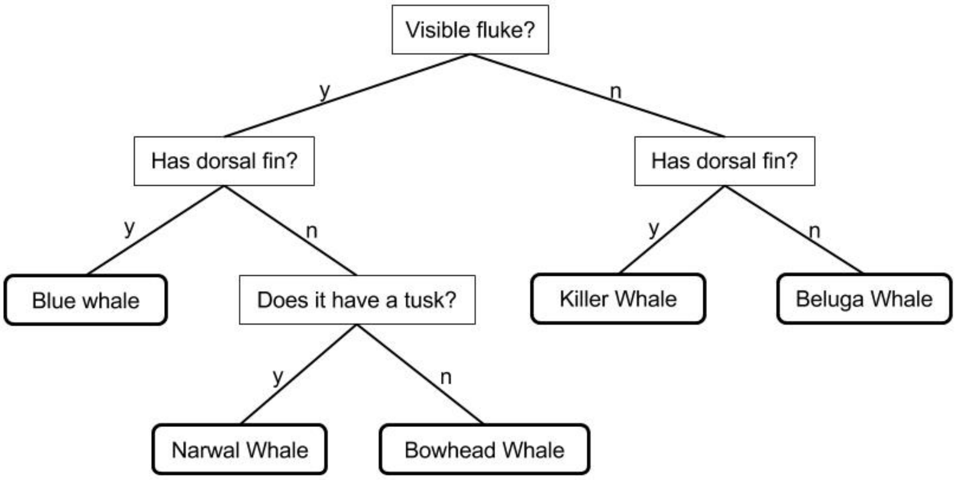
  Other point to this algorithm is that when choosing a k value, the number of clusters should match the data. An incorrect choice of the number of clusters will invalidate the whole process. An empirical way to find the best number of clusters is to try K-means clustering with different number of clusters and measure the resulting sum of squares.

Assignment 2:

Decision trees. A marine biologist gives you the following descriptions of various whale species:
Killer whale The fluke of this relatively small (6-8m) whale is not visible when it dives, but its tall and pointed dorsal fin is often clearly visible. You can also see the whale blow water quite often.
Beluga whale This whale can be difficult to spot as it does not show its fluke when diving, and does not have a dorsal fin.

Narwhal whale These very small whales usually do not grow above 5 meters and are known for their single, extraordinarily long tusk. Their fluke is clearly visible when they dive, and they do not have a dorsal fin.
Bowhead whale Much like the Narwhal, this whales fluke is visible when it dives, and it does not have a dorsal fin to show off. It is however, a lot larger, reaching sizes up to 20 meters.
Blue whale This whale is believed to be the largest animal ever to have existed. Growing over 30 meters long, its impressive fluke can be seen clearly when it dives. Its dorsal fin, although relatively small, is also often clearly visible.
Build a binary decision tree that can be used by whale spotters. Try to keep the questions as simple as possible, minimize the height of the tree, and explain why you chose this particular tree.

Answer: This particular tree was chosen because it has the lowest height possible and very simple question nodes.



Assignment 3:

K-nearest neighbor classification. We want to do K-nearest neighbor classification. Given is the data file w5_1 .mat containing 100 2D data points in the space [0,1]x[0,1]. The first 50 points belong to class 0 and the second 50 points belong to class 1. Given is the following code (see file w5_1.m):

Listing 1: w5-1.mat

```
clear all;
load w5-1.mat;
K=1;
N=64;
data = w5_1;
nrofclasses = 2;
for i=1:N
    X=(i-1/2)/N;
    for j=1:N
        Y=(j-1/2)/N;
        result(j,i) = KNN([X Y],K,data,nrofclasses);
    end;
end;
imshow(result,[1 nrofclasses], InitialMagnification ,  fit  )
hold on;
data=N*data; % scaling
```

```
% this is only correct for the first question
plot(data(1:50,1), data(1:50,2),    g o   );
plot(data(51:100,1),data(51:100,2),   r +   );
```

For each point in the space it determines the class by K-nearest-neighbor classification. The resulting image shows a black (0 for 0) and white (1 for 1) image showing to which class each point in the space belongs.

- Implement the KNN function and give the code in your report. For a given K it should return the class to which point (X, Y) will belong to based on the data and nrofclasses variables. Write your function in such a way that it works for more than two classes too (see assignment 3.3).

  Answer:

  Listing 2: KNN.mat

```
function a = KNN(cod, k, points, nrofclasses)
count(nrofclasses,2)=0;
dist(100,2) = 0;
for i=1:100
    dist(i,1) = pdist([cod(1) cod(2);points(i,1) points(i,2)],'
        euclidean'); %calculates the distance
    dist(i,2) = i;
end;

dist = sortrows(dist); %sort the matrix in ascending order by the
    first collumn

for i=1:nrofclasses
    count(i,2) = i;
end;

for j=1:k
    for i=1:nrofclasses
        if(dist(j,2)<=i*(100/nrofclasses) & dist(j,2)>(i-1)*(100/
            nrofclasses))
            count(i,1) = count(i)+1;
        end;
    end;
end;

count = sortrows(count); %sort the matrix in ascending order by
    the first collumn
count = flipdim(count,1); %inverts the sorted matrix, making it be
     sorted in descending order

a = count(1,2); %returns the value to be displayed asgray scale by
     the function imshow;
end
```

- Show the results for classification if K = 1,3,5,7;

  Answer: Results in respective order (K = 1, 3, 5 and 7).

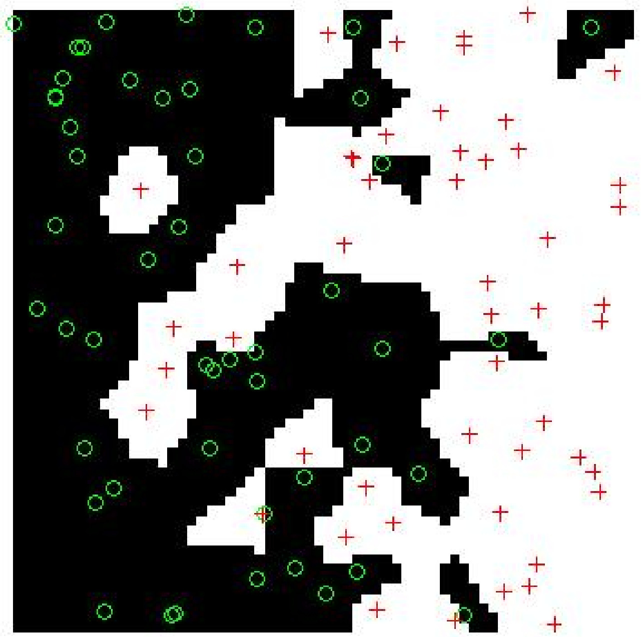


Figure 1: w5_1.mat with K = 1

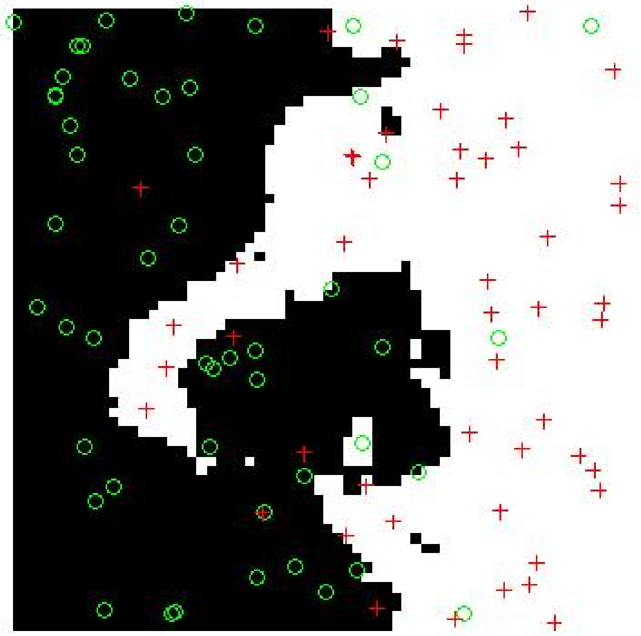


Figure 2: w5_1.mat with K = 3

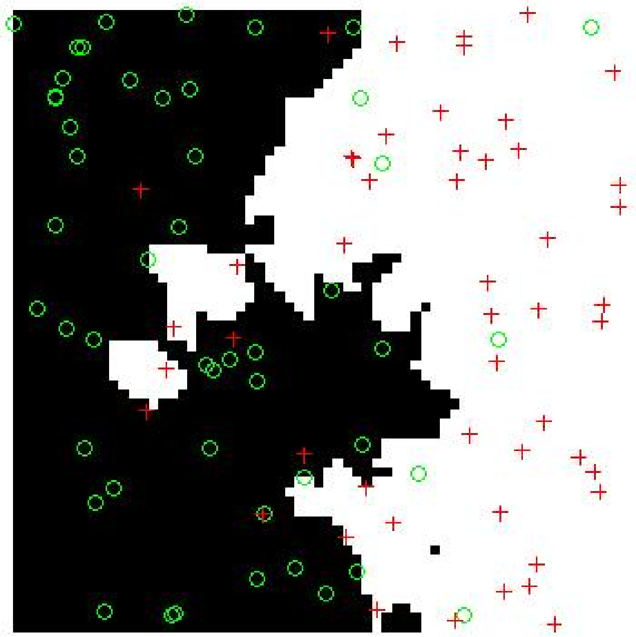


Figure 3: w5_1.mat with K = 5

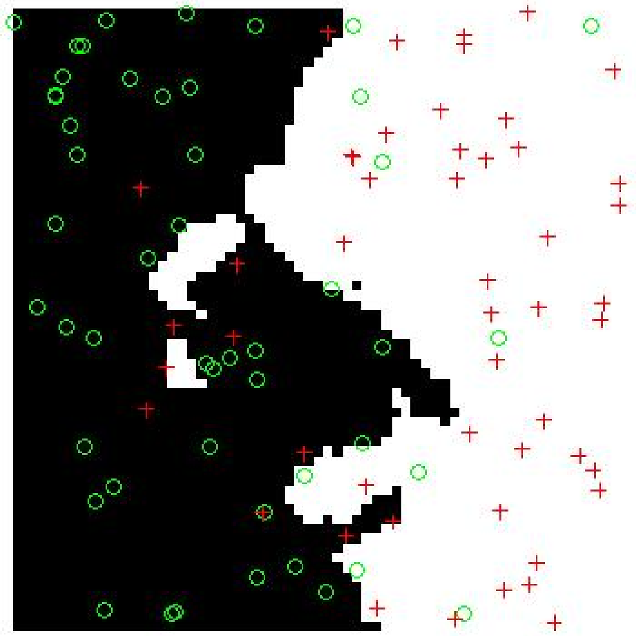


Figure 4: w5_1.mat with K = 7

- Repeat assignment 3.2 but now assume that there are 4 classes containing the points with indices (1 25, 26 50, 51 75, 76 100). In the case where K = 3, it may happen for instance that all three nearest neighbours are of a different class (similar scenarios exist for higher K). There are different approaches to decide which class to choose in such a case (lowest class number, class by closest point, class with lowest average distance within the KNN points, etc.). Explain in your report which approach you used and why.

  Answer: The approach used in the case of a draw was to use the lowest class number, because it’s how the group’s code for question 3.1 already works, therefore requiring less effort.

Listing 3: Modiefied w5-1.mat for the last question

```
clear all;
load w5-1.mat;

K=1;
N=64;
data = w5_1;
nrofclasses = 4;

for i=1:N
  X=(i-1/2)/N;
  for j=1:N
    Y=(j-1/2)/N;
    result(j,i) = KNN([X Y],K,data,nrofclasses);
  end;
end;

imshow(result,[1 nrofclasses],’InitialMagnification’,’fit’);
hold on;
data=N*data; % scaling

% this is only correct for the last question
plot(data(1:25,1),  data(1:25,2),  ’go’);
plot(data(26:50,1),data(26:50,2),’y+’);
plot(data(51:75,1),  data(51:75,2),  ’b*’);
plot(data(76:100,1),data(76:100,2),’rs’);
```

Results in respective order (K =1, 3, 5 and 7).

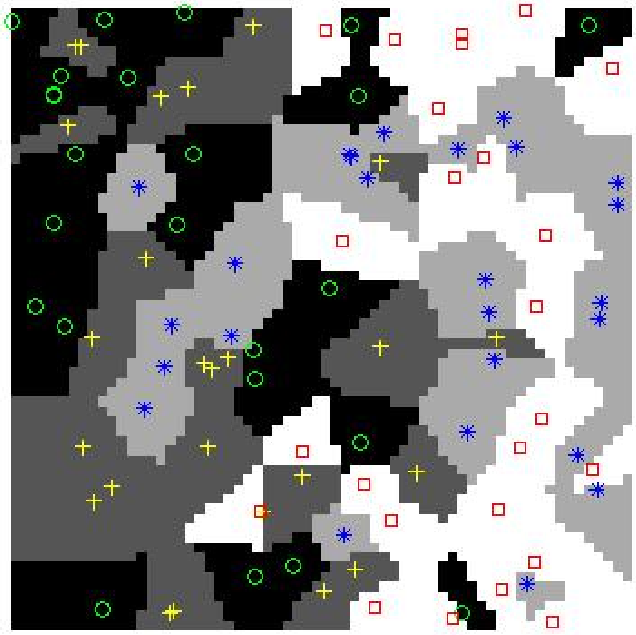


Figure 5: w5-1.mat with K = 1

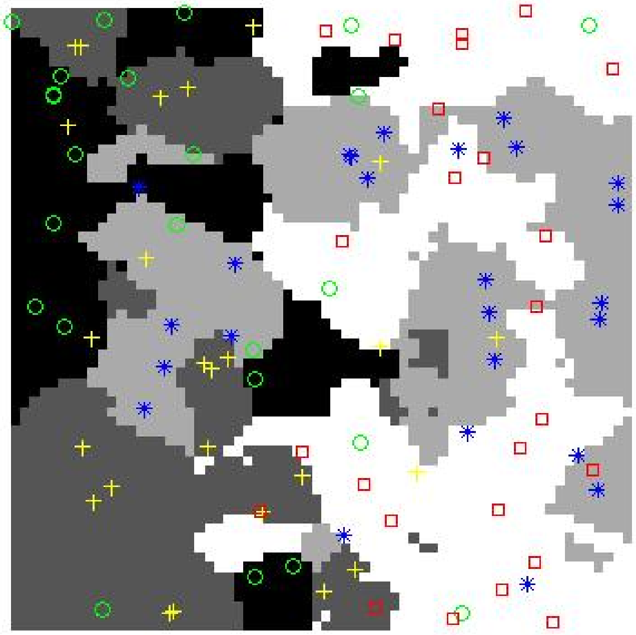


Figure 6: w5-1.mat with K = 3

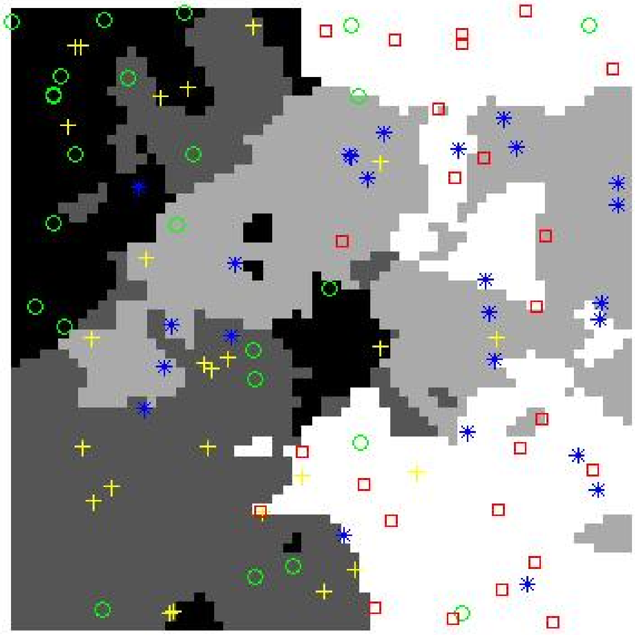


Figure 7: w5-1.mat with K = 5

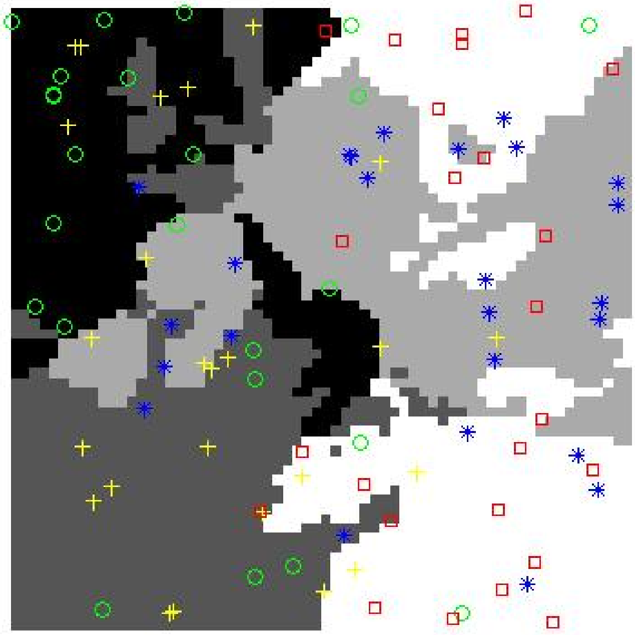


Figure 8: w5-1.mat with K = 7

Division of work

The first and second assignments were done as by Eduardo. The last assignment was done by Diego. All the scripts used have been included in .zip file e-mailed.