Useful Matlab functions: `norm, scatter, unique, size, [Y,I]=sort(X), [Y,I]=max(X)`

Introduction to Intelligent Systems
Lab week 2

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).

1. 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.

   (hint: Write your algorithm as a function that takes the 2D array of samples and the number K of means as arguments so that you don't need to change your code to use it for a different number of means, or for a different dataset. Initialize the K means as K distinct points randomly chosen *from the dataset*. If you have problems determining the stopping criterion, ask yourself the questions: when do the means no longer change? What causes a cluster-mean to change? Can you track this? Be careful when using `sum` on `find` results; if only one (or no) match was found, then Matlab might sum over rows, while otherwise it would sum over columns.)

   Some suggestions: you can plot the intermediate cluster-means and the clusters themselves in two different figures of course, but can you also plot them in the same figure (how would you make the cluster-means stand out in such a plot?), using for example `hold on` and several calls to `plot`, varying the third argument to change line/point style and color. Can you perhaps draw lines around between the clusters in some way? Or do you know other ways to represent the clusters?

2. 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).

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?

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.

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

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 whale's 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.

1. 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.

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 $[0, 1] \times [0, 1]$. The first 50 points belong to class ω_0 and the second 50 points belong to class ω_1.

Given is the following code (see file `w5_1.m`):

```
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),  'go');
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 ω_0) and white (1 for ω_1) image showing to which class each point in the space belongs.

1. 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).

2. Show the results for classification if $K = 1, 3, 5, 7$;

3. 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.