Useful Matlab functions: `linkage, dendrogram, pdist, squareform, cell, bar3`

Introduction to Intelligent Systems
Lab week 3

Assignment 1: Normal distributions Consider the following two sets of observed 1D data (feature values) coming from objects of two different classes:

```
S1 = [-30.8508   23.6509  -15.4613   -0.0466   26.5835   41.1727
         -23.5292   47.8198    29.8043    7.9640    4.1434    3.4718
           0.9068   10.6914   11.5387   34.7819    55.8093   24.0074
           3.7086   28.7557   15.4968  -20.8930   38.4767   19.2119
          14.0552   25.4574   17.8422  -18.2446    5.1299    5.6184]

S2 = [44.6799   66.8210   41.2427   45.1618   42.8800   38.2579
      48.0776   47.2593     65.3007   47.5098   39.3579   66.0346
      62.3468   47.7037   34.9384]
```

and the following test set of data points that need to be classified:

```
T = [5 23 40 70 95].
```

a Plot the elements of the two sets S1 and S2 on the x-axis of a Fig.1: the elements of S1 as blue circles ('bo') and the elements of the set S2 as red circles ('ro'). Plot in the same Fig.1 the points of the test set T as black squares ('ks').

b After a brief consideration of what the underlying distributions that produced the training data sets S1 and S2 might be, decide that they are normal distributions. Compute the parameters (mean and standard deviation) of the two distributions, using maximum estimation. Create a Fig.2 in which you plot the two Gaussian functions, one in blue the other in red, together with the points of the two training data sets S1 and S2. For a moment, contemplate on the elegance of the functions in comparison to the rugged data in the background. The two functions are the class conditional probability densities $p(x|\omega_1)$ and $p(x|\omega_2)$ that the two classes produce a value x of the considered feature.

c Estimate the prior probabilities $P(\omega_1)$ and $P(\omega_2)$. If you have no clue how to do that, look at the lecture slides and follow the steps of the example given there.

d Create a Fig.3 in which you plot the two products $P(\omega_1)p(x|\omega_1)$ and $P(\omega_2)p(x|\omega_2)$, together with the points of the two training data sets S1 and S2 and the test set T.

e Substitute the values of the prior probabilities and the class conditional probability densities determined above in the equation $P(\omega_1)p(x|\omega_1) = P(\omega_2)p(x|\omega_2)$ and solve the resulting equation for x in order to determine the value(s) of the decision criterion that we should use for classification.

f Using the thus obtained value(s) of the decision criterion, determine the classes of the points in the test set `[5 23 40 70 95]`. Create a Fig.4 which copies Fig.3 and adds to it a specification of which domain on the x-axis belongs to class 1 (blue) and which complementary domain corresponds to class 2 (red).

g Evaluate the misclassification rate of the thus created classifier for each of the two classes. (For this, you can for instance use the error function *erf*, familiar from the iris recognition assignment in the first week.)

Assignment 2: *Dendrogram*

The file *dataAEX.mat* contains 19 time series of 19 stocks whose names are specified in *labelsAEX.mat*. You can load the data of these files in your program using *importdata* or *load*.

Create a dendrogram of the 19 stocks using the Matlab functions *linkage* and *dendrogram*. Apply the complete linkage algorithm, using Euclidean distance between the time series.

Some additional Matlab info that may help you: The following information may be useful for your Matlab practice. It is not an assignment and you need not provide any report concerning the following.

Cell data

The cell data format enables the user to create more complex data structures than matrices at the cost of losing matrix operations like multiplication and addition. You can, however, loop over cell data, for example:

```
vals = {'banana', 'apple', 'strawberry'};
for i=1:length(vals)
  disp(vals{i});
end
```

In this way you can loop over string options of Matlab build-in functions, and hence do calculations with different settings. Notice that the strings have different lengths, which is allowed when you use a cell structure.

Storing figures

Another tool Matlab provides is storing plots to a file using *print*. Be very careful though: calling print the *wrong* way will actually print your document to the default printer. Using it in a for-loop, might cause thousands of documents to be printed. You don't want that.

However, you can do this:

```
xx = 0:0.05:1;
functions = {@(x) log(x), @(x) sin(2*pi*x), @(x) (x-1).*x};
labels = {'log', 'sine', 'parabola'};
for i=1:length(functions)
  fig = figure; hold on; %open a new figure window
  fun = functions{i};
```

```
  plot(xx, fun(xx), '-');
  set(gca, 'FontSize', 24); % set the font size of the window to 24
  xlabel('x', 'FontSize', 24);
  ylabel('f(x)', 'FontSize', 24);
  filename = sprintf('%s.eps', labels{i});
  print(fig, '-depsc2', filename); % print the figure to a eps-file
  close(fig); %close the figure window;
end
```

This should have created 3 eps-files.