Pattern Recognition
Classification

Useful matlab functions:

`normpdf`, `normcdf`, `norminv`, `norm`, `mode`

Guidelines for lab reports:

- Always give a (short) explanation of what you are doing.
- Do not forget to include your Matlab programs. Present and discuss the results of your programs, be it a number, a matrix or an image.
- Put large pieces of Matlab code in an appendix.
- One should be able to understand plots independently, be sure to label axes, add a legend for colors, etc.
- Refer to all plots, tables, code blocks, etc. in your report.
- If you print gray-scale make sure the colors used in the plots are distinguishable.

Assignment 1: *classification error, hit/false alarm rates, ROC curve, discriminability.*

Consider two normal distributions with different means $\mu_1 = 5$ and $\mu_2 = 7$ ($\mu_1 < \mu_2$) and equal variances $\sigma^2 = 4$ (see Fig. 2.19 in book or lecture sheets). Let x^* be the value of a decision criterion used to classify an object having a feature with value x in class ω_1 for $x < x^*$ and in class ω_2 for $x \geq x^*$.

The integral of the first distribution (with mean μ_1) for values of $x \geq x^*$ specifies the probability of wrongly classifying an object from class ω_1 into class ω_2 to be referred to as a false alarm. The integral of the second distribution for values of $x \geq x^*$ specifies the probability of correctly classifying an object from class ω_2 into class ω_2, to be referred to as a hit.

1. Choose a value of x^* and compute the probabilities of hit (h) and false alarm (fa). Plot the point (fa, h) in a graph with horizontal axis fa and vertical axis h. Choose a few other values of x^* in the interval $[\mu_1 - 3\sigma; \mu_2 + 3\sigma]$ and plot the corresponding (fa, h) points too. Connect the points with a curve that is called ROC curve. This curve corresponds to discriminability

$$d' = \frac{(\mu_2 - \mu_1)}{\sigma} = \frac{(7-5)}{2} = 1.$$

Repeat the computation of a ROC curve for the cases $\mu_2 = 9$ and $\mu_2 = 11$ and plot all three ROC curves in the same diagram. What is the value of the discriminability d' for each of these cases?

2. Consider the binary vectors given in file `lab3_1.mat`. They specifiy the outcomes of a psychometrical experiment in which a test person is presented a visual stimulus that contains or does not contain a given signal and the test person has to specify whether he/she detected the signal. The vectors code for the following outcomes:

1 1	the signal was presented and detected (hit)
1 0	the signal was presented but the person failed to detect it
0 1	the signal was not presented but the test person indicated to have detected it (false alarm)
0 0	the signal was not presented and the test person indicated that there was no signal

Compute the values of the hit rate h and the false alarm rate fa and plot the point (fa, h) in the plot computed in assignment 1.1 above. This point lies on a ROC curve with a given disciminability value d'. Determine this value by trial and error, i.e. taking different values d' and drawing the corresponding ROC curves until you find a value of d' for which the corresponding ROC curve passes through the experimentally determined point.

hint: Notice that in this case we do not know the values μ_1, μ_2, σ and x^* of the detecting system. And yet we can compute its discriminability d' using the empirically obtained hit and error rates (and under the assumptions that the internal distributions are normal and have equal variances, the difference in their means is caused by the presence of a signal in the visual stimulus).

hint: Note that the first feature encodes whether or not the signal was shown, and the second feature denotes whether or not the test person claims to detect something. So the hit rate is defined as the probability that the test person claims to detect something, given that a signal was presented. And the false alarm rate is the probability that the test person claims to detect something, given that no signal was presented.

Assignment 2: *K-nearest neighbor classification*

We want to do *K-nearest neighbor* classification. Given is the data file `lab3_2.mat` containing 200 2D data points in the space $[0, 1] \times [0, 1]$. The first 100 points belong to class ω_1 and the second 100 points belong to class ω_2.

The program in the file `knn_wrapper.m` uses such a classifier. For each point in the space it determines the class by K-nearest-neighbor classification. The resulting grayscale image shows to which class each point in the space belongs (in the case of two classes, black is ω_1, and white is ω_2).

1. Implement the `KNN` function with the Euclidean distance function and give the code in your report.

   For a given K it should return the class to which point (X, Y) belongs based on the `data` and `class_labels` variables. The value that your `KNN` function returns should be one of the `class_labels`. Write your function in such a way that it works with more than two classes (see assignment 2.4), and with more than two features as well (i.e. given `[X Y Z]` instead of `[X Y]` as first parameter (the dimensions of `data` will then of course be different as well); you will need this for assignment 3.6).

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

3. Determine the optimal choice of the parameter K in the range $1, 3, \dots, 25$ using leave-one-out cross validation:

   - For each point in the data set, make a copy of the dataset without that point.
   - Classify the point using the reduced dataset.

   Report the error rates for the different values of K (in a plot or a table), which one is the best? And can you explain this result?

hint: Use `array(i)=[]` to remove an element from an array.

4. Repeat assignments 2.2 and 2.3 but now assume that there are 4 classes containing the points with indices (1:50, 51:100, 101:150, 151:200).

   Now in the case where $K = 3$, for instance, it may happen that all three nearest neighbors are of a different class (similar scenarios exist for higher K). In case of such a tie, make an arbitrary choice. You can for example simply choose the class with the lowest class number.

Assignment 3: *Parzen windows, posterior probabilities*

Consider Parzen-window estimates and a classifier for 3-dimensional data sampled from three categories. The data points are given in the files `lab3_3_cat1.mat`, `lab3_3_cat2.mat` and `lab3_3_cat3.mat`.

Let your Parzen window function be the following spherical Gaussian:

$$\phi\left(\frac{x - x^{(j)}}{h}\right) = \exp\left[-\frac{\left(x_1 - x_1^{(j)}\right)^2 + \left(x_2 - x_2^{(j)}\right)^2 + \left(x_3 - x_3^{(j)}\right)^2}{2h^2}\right]$$

where $x_i^{(j)}$ is the i-th component ($i = 1...3$) of the j-th vector ($j = 1...10$) from a given category (file).

1. Set $h = 1$ and compute the densities for each of the three classes for the following three points:

$$\vec{u} = \begin{pmatrix} 0.5 \\ 1.0 \\ 0.0 \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} 0.31 \\ 1.51 \\ -0.50 \end{pmatrix} \qquad \vec{w} = \begin{pmatrix} -1.7 \\ -1.7 \\ -1.7 \end{pmatrix}$$

   The density for a given point $x \in \{\vec{u}, \vec{v}, \vec{w}\}$ and category k ($k = 1, 2, 3$) is computed as follows:

   - Evaluate the function ϕ for the given point x and a data point $x^{(j)}$, $j = 1...10$, from the concerned category. Note that ϕ is a product of three Gaussian functions, one for each of the three components of x and $x^{(j)}$.

- Sum the results obtained for the different data points $x^{(j)}$, $j = 1...10$, of the concerned category.
- Divide the result by a normalization factor $(h\sqrt{2\pi})^3$ which essentially is the volume of the Parzen window.
- Divide the result by the number of data points in the concerned category (10).

Do this for all three categories in order to obtain the three probability densities in the concerned point x.

2. Give an estimate of the priors P_1, P_2 and P_3.

hint: Keep it simple!

3. Using the obtained values of the priors and the densities of the three categories in the concerned point x, compute the posterior probabilities of the categories for that point. Do this for the three points x given above.

4. Using the obtained posterior probabilities, decide which category each of the points x belong to.

5. Repeat the assignments above (3.1 to 3.4) for $h = 2.0$.

6. Classify the three points x using the nearest neighbor rule (re-use your KNN function from assignment 2). Compare the results with the results of 3.4 and 3.5.

7. Classify the three points x using the k-nearest-neighbors rule for $k = 5$. Compare the results with the results of 3.4, 3.5 and 3.6. From your results, how do the parameter h of the Parzen window and the parameter k of the KNN algorithm appear to be related? Can you explain this relation from a theoretical point of view?