Lab 1
k-Nearest-Neighbours

7 September 2012

In this exercise you are given an artificial set of data, belonging to two classes. The task is meant to familiarise you further with MatLab and the tools we use this course, as well as with some fundamental issues that we encounter in Machine Learning. The hand-in is a report in PDF format (again, I suggest you write it in LaTeX). Part of the exercise consists of gaining some familiarity with the typesetting program, as this will be useful for your future exercises.

Netlab

In order to perform these exercises, you will need MatLab (or octave, a free MatLab-compatible interpreter) and the Netlab package. Install Netlab, a set of matlab files which can be downloaded from http://www.ncrg.aston.ac.uk/netlab/down.php. Be sure to download release 3.3, and install the package by putting the files where matlab can find them: Either unzip the files to your default matlab directory,[1] or put the files in a directory of your choosing and add that directory to your default matlab path. To do the latter, use the File → Set path menu or type:

```
addpath('my_netlab_dir');
```

on the command line.

The zipped help files, available from the same download page, contain succinct yet complete descriptions of each function, as well as examples of how to use them.

Some Netlab commands you will need

The Netlab package is a very complete suite of matlab functions implementing textbook ML techniques. We will use it repeatedly in the labs of this course. When its functions expect a data set, they expect the multiple data points in the form of a matrix where each row is one multidimensional point and the columns are dimensions. That is,

$$D = \begin{bmatrix} d_1^{(1)} & \cdots & d_N^{(1)} \\ \vdots & \ddots & \vdots \\ d_1^{(M)} & \cdots & d_N^{(M)} \end{bmatrix} \tag{1}$$

where $d_j^{(i)}$ is the jth dimension of the ith data point. This may seem intuitive, but you should keep in mind that this is the transpose of what is conventional in the field.

For this lab, you should check out the following functions:

knn This function creates a k-nearest-neighours classifier, storing all training examples in the resulting structure. Beware that this function expects the training labels to be in 1-of-N encoding: *i.e.*, it expects the label of a data point to be a vector of all zeroes, except for the column that corressponds to its class, which should be one.

[1] `$HOME/matlab` under Linux, or `C:\Program Files\MATLAB_folder\work` under Windows.

knnfwd This function classifies new data points using the structure returned by **knn**. It similarly returns the classification results in 1-of-N format.

confmat This function computes the confusion matrix from two sets of labels: the predictions of the classifier and the ground truth. Again, the expected format is 1-of-N. You can use the confusion matrix to compute the classification error and accuracy, as described in the slides.

The function **conffig** is called in the same way and displays the same information but may be visually more convenient.

Marks

This lab is marked out of 20. The exercises are divided in three groups, with diminishing return on invested effort. You are encouraged to ask questions and discuss the exercises among yourselves, but remember that you are marked on your own insight and understanding — not on anyone else's.

Exercise 1 **(8 marks)**

Download the data file `twoclass.mat` from the lab's blackboard page. This is a matlab file containing 600 two-dimensional data points in two classes.

1. Load the data file (`twoclass.mat`) in matlab.
2. Create a training set containing 75% of both classes, and a test set containing the rest.
3. Plot the data in the training set with the plot command (try help plot for more information on the command). Use different symbols and colours for the different classes.
4. Save the plot as a PDF file, and include it in your report.

Exercise 2 **(6 marks)**

1. Train a kNN classifier, with $k = 1$, on the training data, evaluate its performance on the test data and compute the test error.
2. Try other values for k ($k = 1, 3, \dots$), plot how the test error evolves as a function of k and include the resulting graph in your report. Which value of k would you use, and why? Can you derive any definite conclusions from this experiment?

Exercise 3 **(6 marks)**

1. Discuss briefly (200 words) what the advantage would be of using a method like cross-validation or bootstrapping in this context.
2. Implement 10-fold cross-validation, and give an estimate for a good value of k.
3. Can you use this estimate of k in your final classifier? What is the use of doing cross-validation?
4. How do you compute the test error? What is it an estimate of? What would be the use of a validation set?