Learning from Data, Assignment Sheet 1

School of Informatics, University of Edinburgh

Instructor: Amos Storkey

Handed out: Thursaday 11 October 2007.
Submission Deadline: 23:59 on Mon 29 October 2007.

Please remember that plagiarism is a university offence. Do not show your work to anyone else. Please also remember that, on any course, you learn as much or more from your peers as you do from your tutors. Please feel free to discuss the general problems with one another (ideally after you have looked at them yourself). But at the end of the day what you write must be yours, and you must understand what you write, and why you didn't write other things. The approach should be one you have chosen to take. If you don't understand it don't write it — it will generally be obvious you don't understand.

The number of marks assigned to each task is given in square brackets. In total, this assignment will contribute 8% to your overall mark for LFD. Please remember that late submissions are not allowed without good reason. Last minute issues (problems of any sort occurring in the last 24 hours) are generally not considered good reason as some contingency should be allowed for. I recommend submitting things 48 hours before the deadline. You can always submit again later if you need to. Presuming a moderate understanding of the course material, this coursework should take *at most* 12 hours, including some time familiarising yourself with Matlab. The first questions are more practical. The last two involve more thinking. Please use the code given here and in the notes to help. Those with absolutely no Matlab background may need a little more time getting familiar with Matlab. Many code examples are given in the assignment. Don't use them blindly. Matlab will also be needed in the second assignment, so use this one as a means to familiarise yourself.

Please "hand in" your submission electronically, using the `submit` program. Put your answers in plain ascii text (no html etc) in a file named `answers.txt` in a directory named `answers`, along with any code you wrote for the project. Then from a DICE directory that contains answers as a subdirectory type

```
submit msc lfd-5 1 answers
```

if you are an MSc student or

```
submit ai4 lfd-4 1 answers
```

if you are a third or fourth year student. Failure to follow these instructions could result in no marks being given. Note the 1 to distinguish this submission from that for the second assignment. Typing a 2 instead will result in your submission not being received and zero marks being given. Using a directory with a different name (other than `answers`) could result in your answers being missed and a zero mark. Be warned. Be careful.

I repeat: Typing a 2 instead will result in your submission not being received and zero marks being given. Using a directory with a different name (other than `answers`) could result in your answers being missed and a zero mark. Be warned. Be careful.

The questions ask you to do some MATLAB programming. It is assumed that some of the time of this assignment will involve familiarising yourself with MATLAB. See `http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html` for some helpful documents, especially the *getting started* section. The routines that you write to accomplish this assignment should be submitted as part of your answer. You are also asked to provide some written answers – these should be written in a plain text file named `answers.txt`. Include your NAME and STUDENT NUMBER at the top of the file, and the LEVEL YOU ARE TAKING THIS COURSE AT (LEVEL 10 or LEVEL 11).

Do not include markup (e.g. html). Keep your file within 80 characters width for ease of printing. Do not include answers in the code. Make it clear what question you are answering at each point. Follow the instructions carefully. You should try to make your code clear and comment it. The code will only be looked at if there are any questions regarding the originality of the answers given, or where an unforseen ambiguity arises. Do not include your answers in the code. The marks will be given on the basis of the written work. The code will not contribute to the marks itself.

The marks associated with each question are given. Good insightful answers to questions can serve to increase the baseline marks by at most a further 10 percent up to a maximum of a hundred percent. Verbosity will be penalised.

This assignment is about Naive Bayes, Principal Component Analysis and Class Conditional Gaussian Modelling.

The data for this assignment is taken from the "Wine Recognition Database" available at `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine`. The data has been doctored (the original problem was too easy to get a very good classification!), preprocessed into training and test sets for you, and is available at `http://www.inf.ed.ac.uk/teaching/courses/lfd/lfdassignments.html`. The original data gives the results of various chemical composition analyses of wines, along with a classification of the wine into three classes.

Please download the data. Load into MATLAB using the `load` command. The test data should only be used for evaluation of the methods you develop. The data file consists of 4 MATLAB variables: the training data and training targets, and the test data and test targets. If you need to ask which variable is which, you should not be doing this course :). Each row corresponds to a different record (i.e. different data point, different sample etc.). The target variables have only one column, which contains the target label. The data variables have one column for each attribute.

In calculating covariances, please use the form provided by the MATLAB function `cov` throughout, even though this is not strictly the maximum likelihood estimate (it normalises by $N-1$ not N for number of samples N).

Visualise histograms of each attribute (Use `hist` in MATLAB). Visualise a few scatterplots (`plot(traindata(:,1), traindata(:,2)),'b.'` will plot attribute 1 against attribute 2). It is always a good idea to do this before tackling any problems.

1 [25 Marks, 3 Hours max]. Write and run a MATLAB function to learn a multivariate Gaussian distribution for all the attributes for each of the three classes. Pay careful attention to the code in the notes. You may reuse any code in the notes so long as you understand what it does, and comment it appropriately to show this, and you may use standard MATLAB functions (again the whole point of this exercise is to help you understand things, so don't do this blindly, but the NETLAB function `gauss` may help you). Compute the maximum posterior classes for the unseen test data and, by comparing these predictions with the true targets provided, report the classification accuracy on the test set as a confusion matrix for the three classes. If you are uncertain what a confusion matrix is, then use the web or textbook to familiarise yourself. Give a sentence or two outline of what you did. You will find the matlab command `find` useful:

```
select1 = find( traintarget==1 );
traindata1 = traindata(select1, :);
```

will build `traindata1` which only includes instances of the training data for which the target is 1. Other MATLAB functions you will need include `mean`, `cov`, and `eig`. Always remember to check things are the right way round (columns or rows). For example `eig` returns the eigenvectors as columns.

2 [15 Marks, 2 Hours max]. The number of data points is fairly small for the number of parameters to be learnt, so this might be a poor fit. You consider reducing the dimensionality of the data. Now write MATLAB code to learn a PCA representation for the data, reducing the data to 10 dimensions. However first you will need to rescale the training data to zero mean and unit variance first using

```
meandata = mean(traindata);
stddata = std(traindata);
traindata = traindata - meandata( ones(1, size(traindata, 1) ), : );
traindata = traindata * diag( 1./stddata );
```

Explain why this is important. Rescale the test data in the same way. Explain why you should not recompute the mean and covariance for the test data. Report the first 2 points of the training data in the reduced (PCA) data representation.

3 [5 Marks, 0.5 Hours max] Use the previous code to learn a Gaussian distribution for each class again, but now just using the reduced dimensional PCA data. Once again report the classification accuracy as a confusion matrix.

4 [15 Marks, 2 Hours max] You notice that many of the attributes are not really Gaussian. You decide to re-represent attribute 5 of the *original data* using a class conditional multinomial distribution for different intervals of the attribute values. This can be done by computing normalised histograms. For example for target 1:

```
  boundaries = [-inf 80 90 100 110 120 140 160 inf]
  trainhist1 = histc( traindata(select1, 5), boundaries );
  pvector1 = (trainhist1+1) ./ sum( trainhist1+1 );
```

calculates the (regularised[1]) multinomial probabilities `pvector1` for class 1. Then the function call

```
 mnp1 = mnprob( pvector1, boundaries, testdata(:, 5) );
```

to the function

```
 function mnp = mnprob(pvector, boundaries, valuelist)
 [mg1, mg2] = meshgrid(boundaries, valuelist);
 bddiff = (mg2 >= mg1);
 [dummy, ii] = max( bddiff(:, 1:end-1)-bddiff(:, 2:end), [] , 2);
 mnp = pvector(ii);
```

will return the probabilities of attribute 5 of all the test data, under this multinomial distribution (returned as a vector, with one element for each test record). Again, try to understand these.

Focus on attribute 5 alone. Ensure you understand the code above if you use it. Compute the probabilities of attribute 5 of the test data for each of the three classes. Write code to use Bayes theorem to output the posterior probabilities of attribute 5 of each test point belonging to each class. Write down the posterior probabilities of each class for the first test record. Is this single attribute informative about the class of wine? Discuss.

5 [15 Marks, 1 Hour max] Once again you *could* use your previous code to compute a lower dimensional PCA representation of the rest of the attributes (not including attribute 5), and learn a Gaussian distribution for the this PCA data. Discuss how you could combine the Gaussian distribution for the rest of the attributes with the multinomial distribution for attribute 5. State precisely how you *would* do this. Write a paragraph discussing the advantages and disadvantages of this approach of combining the multinomial distribution with the PCA+Gaussian over doing the PCA+Gaussian alone on all the data. You are not required to implement this as part of the answer, just discuss the issue. However the inquisitive amongst you may want to implement this just for fun :).

6 [15 Marks, 1 Hour max] Most of the attributes are not actually Gaussian. However it may be the case that, for attribute x, x^a is more Gaussian for some a. How might you find a suitable a for each attribute? Write a brief paragraph outlining an approach to this (again implementation of this is not necessary).

[1] Regularisation will be discussed in a later lecture. You do not need to know the details for this exercise.