First an apology for the length of this document! Most of it is instruction rather than work, I promise. Second the legal stuff!

Please remember that plagiarism is a university offence. Do not show your submitted 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 topics surrounding the problems with one another (ideally after you have looked at them yourself). In fact it would be foolish not to. The main point of this exercise is to improve your understanding, not to just dump you with work to do.** 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.

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. You should keep regular backups of your work on your DICE account, even if you are not working on a DICE machine. No sympathy is likely be given in any case of data loss from a laptop or other medium. Laptop hard drives have a very high failure rate. Don’t rely on them. Learn about the Unison synchronisation system or use a VPN if you aren’t already doing this. We don’t provide an impeccable backup service for it to be ignored.

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 MLPR. Presuming a moderate understanding of the course material, this coursework should take *at most* 12 hours, including some time familiarising yourself with MATLAB. 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. See http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html for some helpful documents, especially the getting started section. Many code examples are given in the notes. Don’t use them blindly. MATLAB will also be needed in the second assignment, so use this one as a means to familiarise yourself. However, you are welcome to base your code upon the code in the notes. We don’t want you to reinvent the wheel.

Please “hand in” your submission electronically, using the submit program. Put your answers in a pdf file that is readable on DICE, and is named answers.pdf (and not anything else) in a directory named answers, along with any code you wrote for the project. Include your STUDENT NUMBER (not your exam number) at the top of the file, and the LEVEL YOU ARE TAKING THIS COURSE AT (LEVEL 10 or LEVEL 11). Do not use other formats for your submission.

Then from the DICE directory that contains answers as a subdirectory type

submit msc mlpr-5 1 answers

if you are an MSc student or

submit ai4 mlpr-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.

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 unforeseen ambiguity arises. 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. The total number of marks available is 8. The assignment will be marked out of 8 to the nearest 0.5, and will be converted to a percentage, rounded up to the nearest percent.

The marking will follow the standard University marking scheme. In the context of this assignment that means:

A Well explained description of points above plus extra achievement at understanding or analysis of results. Clear explanations, evidence of creative or deeper thought will contribute to a higher grade.
B Well explained answers to the questions.

C Fairly accurate answers to many questions, but significant deficiencies.

D Evidence that the student has gained some understanding, but not addressed that specified task properly.

E/F/G serious error or slack work.

Answers that are unnecessarily verbose are invariably going to end up being penalised. Don’t forget - unless you really are an expert, writing lots will probably result in writing something really wrong, and writing something really wrong is going to suggest that you don’t really understand what is going on, and you will get zero, or closer to. Just make your best brief stab at each answer and move on. Spend your time trying to understand good answers, rather than writing long and bad ones...

Remember this is worth only 8% of this course. This course is worth 1/12 of your course marks for the year. So the whole assignment is worth 2/3% of the year marks. Keep it in perspective. The main point of this exercise is to try to help you understand aspects of the course. If what you are doing is not improving your understanding, it probably isn’t worth doing. It should also be at least a little bit fun. Please post any questions, whines or suggested improvements to the forum.

Finally, this exercise is designed to be a little challenging at least in parts. Questions that may be more challenging are annotated as such. Typically such questions can be answered with different levels of insight. These will have less weight in the marking for level 10 than for level 11.
1 The Assignment

The data for this assignment is taken from the “UCI Machine Learning Repository” available at http://archive.ics.uci.edu/ml/datasets/Flags (dataset for Question 1) and http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength (dataset for Questions 2 and 3). The second dataset has been downloaded, preprocessed and split into training and test sets for you, and is available in MATLAB format at http://www.inf.ed.ac.uk/teaching/courses/mlpr/mlprassignments.html, and can be loaded into MATLAB using the load command. The first dataset will involve you doing the download and split as instructed in the question.

The second dataset consists of 4 MATLAB variables: the training data and training targets, and the test data and test targets. The test data should only be used for evaluation of the methods you develop.

The second dataset (Questions 2 and 3 - concrete_data.mat) gives the compressive strength of different concrete samples, along with the age of the concrete and the values for 7 of its ingredients (concrete components). 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 value of the concrete compressive strength. The data variables have one column for each attribute (ingredients + age). The original dataset has been pruned and split into a training set with 50 samples and a test set with 500 samples. In a real world application we would use more samples for training our models. However, the number of samples for the datasets in the assignment is chosen for efficiency in training and numerical stability in testing.

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. Other functions are also provided for you. The times given by each question indicate the sort of time that answering the question should take. Extra time may be needed for revision of material and learning MATLAB. If it is taking you a lot longer than the question suggests, you are probably going about your work in a less than ideal way, and you might want to discuss with someone your approach to the work.

Various MATLAB files are referred to in this document. They are all available on the assignment web page, along with this document.

A summary of what is needed for each question is given at the end of the question.

2 Questions

1 [4 Marks, 4 Hours max]. Download the flags data from the UCI Machine Learning Repository to your workspace. Read the information about the data. Load it into MATLAB.
MATLAB is notoriously bad at loading data. For that reason it is worth getting used to it! The function `loadcell.m` will help you to load it in and the function `celltonumeric.m` will help you to convert it to a useful format. You may need to read up on MATLAB cell arrays. Note that the attributes referring to colours have the same colour set, and so you will want a common reference system for those attributes. This will involve passing a suitable `sharing` input to the function `celltonumeric.m`.

Get used to how these work. After converting to a suitable numeric format, save the first 100 data items in the MATLAB variable `flagdatatrain` and the remaining 94 data items into `flagdatatest`. Each row should be a separate data point and each column a separate attribute.

Spend a little time looking at the data.

We will be trying to ascertain the national religion (attribute 7) from the characteristics of the flag. We will use a Naive Bayes model for this purpose.

We wish to use a Bayesian form of Naive Bayes to learn the Bayesian posterior parameter distribution given the training data, and the predictive posterior distribution of the class for new data points. Pay careful attention to the fact that the attributes have many states, i.e. you will need to use the multinomial rather than the binomial distribution for each attribute. This means that you will need to learn a parameter for each of the states that each attribute can take (and for each of the classes). The maximum likelihood solution for the multinomial distribution takes the form:

\[ \theta_k = \frac{m_k}{N} \]

where \( m_k \) is the number of observations that take value \( k \) and \( N \) is the total number of observations (see Bishop 2.2 for more details on the multinomial distribution). The provided Matlab function `datatolofm.m` is useful and is worth studying for helpful methods. Also you should look up about the `find` function, as it will allow you to select particular data items corresponding to each class.

In `donaivebayes.m` I give some matlab code to do the Bayesian naive Bayes learning. This uses a Dirichlet prior for the parameters. It has no comments. Please could you work through the code and briefly comment each line, saying what it is doing and why. If you can, comment also on why the Dirichlet prior results in this form of code, and what the Dirichlet hyper-parameters are set to. What is the simplest way to describe the prior distribution used? You will probably find the notes on the multinomial-Dirichlet useful. They are on the assignment website.

You should now use the learnt model in order to classify the observations in the test set. There are two things you want to obtain. First you want to get the posterior probabilities
of each class, given each data point. Second you want to get the maximum probability class. Add to the provided code some lines that will compute the test probabilities.

There are a few things to bear in mind when writing code for this sort of problem.

- Work in log probability domain whenever possible.
- When converting from log probabilities to unnormalised posterior probabilities first subtract off the maximum log probability from all items. Then exponentiate. These can then be normalised. Probably best to map back to log probability space afterwards. In general this avoids the exponentiated terms being too big or too small. Convince yourself this approach is correct.
- To compute the product of various probabilities instead sum the log probabilities.

Compute the classification accuracy

$$\sum_i \delta(\text{predclass}^i, \text{trueclass}^i)$$

where the sum is over the test cases, \text{predclass}^i is the highest posterior class predicted by the Naive Bayes model for the \text{i}-th test case and \text{trueclass}^i is the corresponding target. Also compute the predictive log probability

$$\sum_i \log(P(\text{class}^i = \text{trueclass}^i|x^i))$$

Comment (in one short sentence) on the results.

What happens if you just use the maximum likelihood parameters and do not introduce the prior distribution?

[Challenging] Suppose, hypothetically, there was also an attribute entitled ‘Number of languages spoken’. In most cases this would be known, but for some countries this would be unknown due to the fact that there are small tribal groups where the exact language breakdown is unknown. For these countries this attribute is missing the data for these cases. You will be using Naive Bayes for your modelling. Discuss and justify how you would handle this missing data for this attribute.

In your answers give the following (typical lengths of answers are indicated):

- My code, your extensions and your comments.
• If you can, give brief comments about how the Dirichlet prior results in the given code, and what the simplest description of the Dirichlet distribution is. (1 or two short paragraphs, ideally with some equations)

• Your computations of the classification accuracy and the predictive log probability for the test set, and your comment. (two or three sentences will be fine)

• A comment about what happens in the maximum likelihood case. (one sentence really)

• Comments about the hypothetical missing data. (a short paragraph or at most two)

2 [ 2 Marks, 2 Hours max]. Download the data concrete.data.mat to your workspace and load it into MATLAB using the load command. Have a look at it. We will not be actually using the data in this question, but the question still relates to a regression problem in which we wish to model the concrete compressive strength as a function of age and ingredients. A linear regression model for this problem will take the form:

\[ t = w_0 + \mathbf{w}^T \mathbf{x} + \epsilon \]

where \( t \) is the target value, \( \mathbf{x} \) is the vector of attributes and \( \epsilon \) is additive Gaussian noise with zero mean and variance \( \beta \).

What is the problem of using maximum likelihood regression methods to compare the two hypotheses:

1. The Concrete Compressive Strength (linearly) depends on all the variables.

2. The Concrete Compressive Strength is independent of the age (attribute 8) and the Super-plasticizer (attribute 5), and is linearly dependent on the rest?

[Challenging] Why would a Bayesian approach potentially allow this comparison to be done? Describe, briefly, but in enough detail to reproduce, a practical approach to doing this comparison for this problem.

In your answers give the following (typical lengths of answers are indicated):

• Comments on the problem of using maximum likelihood for the hypothesis comparison. (1 short paragraph)

• Why a Bayesian method would help and a few lines describing how. (1 paragraph, maybe with an equation or two)
In this section you are asked to perform PCA on the concrete data in 2, excluding the Age attribute (attribute 8). You intend to perform regression on the reduced dataset (note you do not need to do the regression in this exercise).

[Somewhat challenging] Give an advantage and a disadvantage of doing regression after using PCA over doing regression directly. Give examples to make your point.

Standardize the training data to zero mean and unit variance using the following code:

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

Rescale the test data in the same way. Explain why you should not recompute the mean and covariance for the test data.

Perform PCA on the rescaled training data, and re-represent the training data using the linear PCA transformation to obtain a 2 dimensional representation. Re-represent the test data using the same linear transformation. I advise you to write your own PCA function as that will probably help you remember what it does more precisely.

In machine learning we commonly expect that the test scenario would match the training scenario. Presuming the test data here is taken from a typical test scenario, assess the validity of this assumption using your PCA results.

[Challenging] Would there generally be a better way to do PCA for this purpose? Why?

In your answers give the following (typical lengths of answers are indicated):

- Advantage and disadvantage of PCA, with examples. (Two short paragraphs).
- An assessment of validity of test-training match. (A few sentences, a figure)
- Better way of doing PCA. Explanation of why this would improve things (One short paragraph)

For the sake of clarification a short paragraph is probably three or four sentences. If you have time, you might find it beneficial to try out the doing the linear regression in question 2, though it does not form part of the assessment.