Data Mining and Machine Learning: Coursework 1

Handed Out: 2nd October 2015
What must be submitted (all students): A report of maximum FOUR sides of A4, in PDF format
To be 'Handed in':
Sunday November 1st 2015 23:59pm -- by email to dwcorne@gmail.com with Subject Line: DMML Coursework 1
Worth: 30% of the marks for the module**;**

Marking scheme: 80% for doing what I ask really well; I will look at reasoned argument, insight, presentation, accuracy. 20% 'wow factor' for insightful additional feature: e.g. dazzle me with a particularly interesting and useful way of presenting results, amaze me with some additional work you did associated with these datasets, which is somehow appropriate to this assignment. All within the FIVE sides of A4. If a submission has *n* more than 4 pages, you will automatically lose 10**n* marks out of 100. It will be quite possible to get negative marks for this assignment.
You may be subject to a brief viva in November – this is mainly for me to ensure it is your own work.

The point: I want to make sure you get experience with handling various types of data; this includes applying and understanding methods that explore individual fields of a dataset, and other methods that try to assess the 'big picture'.

WHAT TO DO

Level 10 students: (BSc, and 3rd or 4th yr MEng): PART ONE only

Level 11 students: (MSc and final-year MEng): PART ONE AND PART TWO

PART ONE

Visit the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.html
You will work on **two** datasets from there:

- Spambase
- EEG Eye state

Visit the page for the dataset and the 'Data Folder' link will bring you to the folder. Other information in the data folder (or on the dataset's home page) will indicate what the different fields are, and what the target field means. In both of these cases, the target field is the last column, and is either 0 or 1. **Preparation of the EEG Eye state dataset**: first, you will notice that this is in 'arff; format. It is a plain text file, but 'arff' means that there is some material at the beginning that enables it to be used directly with the 'weka' open source machine learning library. Simply remove those initial lines. Then, since this dataset is rather large and may stretch your resources when doing 1-NN, prepare a reduced version of this dataset by taking a random 1 in 10 of the rows. Using 'awk' for example, you could do this on the linux command line with:

```
awk '{if (rand()<0.1) {print $0}' < dataset.txt > newdataset.txt
```

Now, for each dataset do the following:

1. Produce a version of the data set where each **non-class field** is min-max normalised (in other words, scaled so that each value is between 0 and 1 (inclusive).
2. Calculate the accuracy of 1-nearest-neighbour classification for your dataset.
3. Generate histograms of the distribution of **five fields**, for each of the two classes. For 'spambase' use the first five fields; for EEG, use fields 10,11,12,13 and 14.
4. Write 100—200 words describing how the distributions differ between the two classes, and describing what you think are the two most important fields for discriminating between the classes.
5. Produce a reduced dataset which contains only three fields: the two you considered most important, and the class field.
6. Repeat step 2, but this time for the reduced datasets.
7. Write 100—200 words of insightful discussion about the results of steps 2 and 6.

PART TWO

Visit the UCI Machine Learning Repository again, and learn about the 'Urban Land Cover' dataset. Download a section of this dataset from my site here:
http://www.macs.hw.ac.uk/~dwcorne/Teaching/ulc.csv

In this case, the target field is the *first* field, and, as you can see, there are 147 other fields. Each of the fields is a value obtained from a satellite image for a particular geo position, and the class field provides the land cover category. There are nine possible categories. Going from left to right, after the class field, there is a group of 21 fields which repeats seven times, each for a different image resolution. So, for example, fields 2—22 are the feature values at the highest resolution, and fields 128—148 are the values obtained from the lowest resolution.

Anyway, do this:

1. Produce a version of this data set with 41 fields, where: the first 20 fields are the highest-resolution features, but omitting 'BrdIndx' (column 2 in the original dataset); the next 20 fields are the lowest resolution features, again omitting 'BdrIndx'. The last field is the class field, and this must be changed to a numeric field, as follows:

0 = trees, 1 = grass, 2 = soil, 3 = concrete, 4 = asphalt, 5 = buildings, 6 = cars, 7 = pools, 8 = shadows.

2. Now, repeat PART ONE for your new 41-field version of the dataset. In step 3 of part one, use the first five fields.

HOW TO DO IT?

All of these steps involve processing data files somehow. I don't mind what tools you use to do this. I think the best thing for any student to do is write their own program (in C, C++, Java, whatever) for reading in files and doing the required manipulations. Invariably one finds this is necessary to

do data processing properly – tools like Excel can do quite a lot, but it is overwhelmingly common to realise that there is something you want or need to do with a dataset, which just cannot be done in Excel, but can easily be done if you write your own code. However, if you can do all of this with a spreadsheet program, then that's fine. If you are not a whiz with Excel, note that a good alternative to writing a C program (say) is to take this opportunity to learn to use tools such as awk or Perl. In this handout are several pointers to example awk programs, and explanation of how to run them. These do not do the assignment for you, but by inspecting them you will grasp enough about awk to be able to write awk programs that will do what you need.

MORE DETAIL

Normalising the data: Do Min-Max normalisation of each field, scaling each field of the data (except for the class field) between 0 and 1.

If you need help in producing a min-max normalised version of the dataset, you should be able to do it fairly quickly with an awk program, more easily than with java or C++, etc. Real Excel experts might be able to do it with excel too. I won't provide an awk program (or any other) that does it for you, however you might find it useful, generally for learning about awk, to observe this awk script: http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/znorm.awk; this script is set up for a specific dataset (not one that you are using) and it does 'z-normalisation' which is much more complicated than the normalisation I am asking for here.

Calculating the accuracy of 1-nearest-neighbour: Use this as your distance measure: work out the absolute difference for each (non-class) field, square these differences and add them together. This is the same distance measure I showed you in the second lecture To calculate the accuracy of 1-NN, the basic approach to doing that was given in one of my week 2 slides. Naturally, you can do this with an awk program, or any other way you like.

Generate histograms of the distribution of the data in a field, for each of the two classes. To generate a histogram for field *k* and class *c*: consider only the data instances whose class value is *c*, and consider the set of values in field *k* of these instances. The histogram is a bar chart of frequency (y axis) against value (aligned along the *x* axis). E.g. if 30% of the data instances have value *red* in field *k*, then the bar above *red* will go up to 0.3. If field *k* is numeric (which is true in all cases here), then the x axis should comprise FIVE 'bins', which cover the range of values. E.g. if the values range from 20 to 180, then the first bin represents instances with values in field *k* between 20 and 52, the next bin represents values between 52 and 84, and so on, each covering $1/5^{th}$ of the range. To generate the histograms I showed you in lecture 3, I used (guess what …) an awk script that output the frequency data for each bin, and then imported this into Excel to do the charts. I do not provide this awk script – write your own code, or find a tool that will help you do this.

DATASET PREPARATION

Generally, when I work about with datasets I make much use of the **awk** program. I find it much easier to work with data where the fields are separated with spaces than with commas. So one of the first things I do is convert a csv dataset to a space-separated dataset.

Here: http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/cs2ss.awk

you will find a simple awk program that will convert a csv dataset to a space-separated dataset. Use it like this, at the command line **on a unix or linux machine**:

```
awk –f cs2ss.awk < dataset.csv > dataset.ss
```

where “dataset.csv” can be replaced with the name of your comma-separated dataset, and the new version is in “dataset.ss”, or whatever else you decide to call it.

You don’t have to do this of course – I am merely using this as an excuse to give you another ‘awk’ example.

THE REPORT

- One brief paragraph that tells me how you did it / what tools you used – MAX 50 words. This will not affect the marks – I would just like to know.
- A clear presentation of your histograms, and the associated discussions;
- A clear presentation of your 1-NN results, and the associated discussions;
- Any additional material you feel appropriate;
- That must all be done within 4 **sides** of A4.