F21DL Data Mining and Machine Learning: Coursework 1

Handed Out: Thursday September 20th 2012
What must be submitted: A report of maximum FIVE sides of A4, in PDF format
To be 'Handed in':
Friday October 19th 2012 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 5 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'. You will work with three datasets altogether.

WHAT TO DO

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

- Communities and Crime
- Pima Indians Diabetes
- Yeast (you should remove field 1)

Everyone will work on **three** datasets; the three from the UCI repository. In each case, the dataset itself is a text file with comma-separated fields, and the ".names" files gives information about the fields and classes. In all cases the last field is a "class" field of some kind.

For each of your datasets, do this: further details are provided later about each stage.

1. Download it (of course), then **read the Dataset Preparation section of this handout**
2. Produce a version of the data set where each **non-class field** is min-max normalised.
3. Convert into a **two class dataset**; do this for both the original and normalised cases.
4. Calculate the accuracy of 1-nearest-neighbour classification for your dataset; do this for both original and normalised versions
5. Generate histograms of the distribution of the **first five fields**, for each of the two classes.
6. 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.
7. Produce a reduced dataset (two versions: original and normalised) which contains only three fields: the two you considered most important, and the class field.
8. Repeat step 4, but this time for the reduced datasets.

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 – which is the last field in each of these datasets) between 0 and 100. This is equivalent to scaling to the interval [0,1] and then multiplying each value by 100. In the case of the Communities and Crime dataset (already [0,1]-normalised), do z-normalisation instead. I provide an awk script (below) which will do z-normalisation for you.

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, which you can use for the Communities and Crime dataset. This will do *z*-normalisation of a dataset with numeric fields. Min-max normalisation is less complicated.

Converting into a two-class dataset
This simplifies the rest of the assignment; however it is also something that is often sensible to do with some datasets. For example, many machine learning methods can only be applied to two-class datasets. And, whether or not your machine learning method can handle multiple classes, it is often valuable to see what patterns you find when data mining with different clusterings of the classes.
You only need to do this for the Communities and Crime dataset and the Yeast dataset. In the Crime dataset, the last field is the class field, indicating the number of violent crimes per 100,000 in the population. It is a real number between 0 and 1 (this data is already min-max normalised to [0,1]). When converting this into a two-class dataset, class 0 should represent the instances where the violent crimes is between 0 and 0.4, and class 1 should represent everything above 0.4. If you need help with this, I provide an awk script here:
http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/fiddlefield.awk
which simply takes field 12 of a dataset, and converts it into 0 if it is smaller than 0.3 and 1 otherwise. See "Dataset Preparation" below to see how to use awk scripts; note that this one only works when with a space separated version of the dataset.
On the Yeast dataset, the two classes should be CYT and OTHER; so, you need to produce a file that replaces the class field with "OTHER" if it is not "CYT"

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 first five fields, for each of the two classes. Do this only on the non-normalised versions of the datasets. 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 5 '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 otherwise 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.

Communities and Crime dataset

The raw dataset has 128 fields; the first 5 are not useful for data mining, and another 23 of the fields have missing values. You should use only the remaining 100 fields (99 predictive fields and the class field – the amount of violent crime per 100k of the population). You can do this yourself, but I provide awk scripts to do this for you – examining these will also help you learn more about awk, if you wish to. So, you can prepare the communities and crime dataset like this:

1. convert it to a space-separated dataset, as above (otherwise the awk scripts below won't work properly)
2. use http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/fixcommdata.awk to produce a version that misses the first 5 fields, and contains only the fields that have no missing values.

Pima Indians Diabetes

In this case the raw data are all numeric, all fields are useful, and there are no missing values – so there is no special preparation to do. Naturally you can use my awk script to covert it into space separated if you like, but that's up to you.

Yeast dataset

Note that this is already space-separated. However the first field is not useful for prediction. You should remove this field. I provide here a general awk script that will remove specific numbered fields in any (space separated) dataset:
http://www.macs.hw.ac.uk/~dwcorne/Teaching/DMML/removefields.awk
Before running the script, you have to indicate the number of fields in the original dataset, and the specific fields you want to remove, by editing the awk program in obvious places.

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.

- For step 4: A table that tells me the answers for each dataset, followed by a paragraph that attempts to explain any differences in performance between the normalised and original versions, or explains why performance is similar.

- For steps 5 and 6: 1 page per dataset; on each page, the 10 histograms, and the discussion (step 7).

- For step 8: sane as step 4.

- That must all be done within 5 **sides** of A4.