COMP4702/COMP7703 - Machine Learning

Prac 1 – Introduction to Data Exploration, WEKA and Matlab

Aims:

- To reinforce some of the basic concepts of machine learning from lectures.
- To gain familiarity with the WEKA software package.
- To gain familiarity with Matlab.
- To produce some assessable work for this subject.

Part 1 - WEKA

Procedure:

Weka is a freely available machine learning software package that is very well-known and widely used. The Weka homepage is:

http://www.cs.waikato.ac.nz/~ml/weka/index.html

The software is written in java and full source code and code documentation is provided. There is also an accompanying book (available in the library):

http://www.cs.waikato.ac.nz/~ml/weka/book.html

While the book is quite good, we won't be following it in the course and it isn't really a users' guide to the software anyway. The attachments to this prac are from this book however.

Weka contains implementations of many machine learning algorithms, and has many other features such as the ability to run large batches of experiments, data visualization and preprocessing and so on. We will start by using some of the basic components in Weka - try not to be distracted by all the features that we won't be using!

→ Run Weka and select **Explorer** from the **GUI Chooser** window.
→ Scan through the Explorer User Guide document (course materials webpage) and compare with what you see on the screen, to get a feel for the software. Don't worry about understanding all points yet.

Weather dataset

→ To start, open one of the datasets that comes with Weka (Open file, data folder, weather.arff). A scanned table of the data is available on the course site. The **Preprocess** tab displays information about the data (list of attributes, number of instances, etc).
→ Go to the Visualize tab. What you see is a "draftman's display" – i.e. scatterplots of the data with respect to all combinations of pairs of attributes. Some of these aren't very useful – e.g. a plot of an attribute with itself (humidity-humidity): 1-D data in a 2-D space. Another example is looking at "windy" and "play" – these attributes are binary, so many datapoints "sit on top of each other" in the scatterplot. A better plot is "temperature" v's "humidity" – click on this scatterplot to get a closer look at this plot.

Q1: for a dataset with n attributes, how many unique scatterplots are in a draftsman's display?

c(n,2)+n

Iris Dataset

One of the most widely used datasets is the Iris dataset, which originates from a famous statistician – R. A. Fisher:

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics 7* (part II): 179-188. Reprinted in *Contributions to Mathematical Statistics*, 1950. New York: John Wiley.

山鸢尾 变色鸢尾

维吉尼亚鸢尾

The dataset contains fifty examples of each of three types of plant (Iris setosa, Iris versicolor, Iris virginica), with four numerical measurements for each observation: sepal length, sepal width, petal length and petal width. This data is included as one of the Weka example datasets.

→ Load the iris data and have a look at it via the Visualize tab. It should be clear that there is structure in the data which has a close relationship with the different plant species (classes).

Mystery Dataset

Q2: On the course website you will find a real-world dataset "mystery.csv". Use Weka's Visualize and the histogram/statistical summary information on the Preprocess tab, do some "Exploratory Data Analysis" of this dataset. Write up your findings as a ½ - 1 page summary. You should attempt to make your summary as technically accurate as possible and avoid any vague language or using terminology without first defining what you mean. You should also try and make your summary informative and insightful (e.g. stating how many variables and observations is nice but is not a particularly deep insight). Finally, if you have any ideas you could make a guess as to what you think the dataset might be about.

Part 2 - Matlab

Procedure:

Matlab is a very powerful numerical computing environment, and we will be using it throughout the practicals in this course.

If you are not familiar with Matlab, you should work through the 'Matlab Primer' available on the course website (under 'Reference Material'). The following questions are not specifically related to Machine Learning; rather they are to ensure you have the skills that are required in future practicals.

The following questions must be conducted in Matlab.

Q3: Given the dataset 'prac1_q3.dat' (available on the course website), what is the mean and standard deviation of the data? Provide the commands/code you used to find this.

Q4: Given the dataset 'prac1_q4.dat' (available on the course website), plot (as blue circles) the first column of the data against the second column of the data. On the same graph, plot (as red

```
Q3  >> load('prac1_q3(2).dat')
    >> mean = mean(prac1_q3_2_);
    >> std = std(prac1_q3_2_) ;
```

```
load('prac1_q4(2).dat')
c1 = prac1_q4_2_(:,1);
c2 = prac1_q4_2_(:,2);
c3 = prac1_q4_2_(:,3);
c4 = prac1_q4_2_(:,4);
plot(c1,c2,'b')
hold on
plot(c3,c4,'r')
xlabel('Input')
ylabel('Output')
```

squares) the third column of the data against the fourth column of the data. Label the x and y axes ‘Input’ and ‘Output’ respectively. Provide your graph.

Q5: Create a histogram of 1000 random numbers from a normal distribution with mean equal to 2 and standard deviation equal to 4. The histogram should have 30 bins. Label your x and y axes ‘Random Variable’ and ‘Frequency’ respectively. Provide your histogram.

```
y = normrnd(2,4,[1000,1]);
histogram(y,30)
xlabel = "Random Variable"
ylabel = "Frequency"
```

Q6. Create a function that has two inputs; 1) a vector (here we will call ‘in’) and 2) an integer (here we will call ‘n’). The function must return a vector as output (here we will call ‘out’). The purpose of the function is to reverse the input vector in chunks of size ‘n’ - the first ‘n’ entries of ‘out’ are the last ‘n’ entries in ‘in’ (and so forth). If the value of ‘n’ causes a chunk less than ‘n’ to be left in ‘in’, please append the remaining entries to out. Examples of appropriate input and output are given in the table below.

‘in’	**‘n’**	**‘out’**
[1, 2, 3, 4, 5]	1	[5, 4, 3, 2, 1]
[1, 2, 3, 4, 5]	2	[4, 5, 2, 3, 1]
[1, 2, 3, 4, 5, 6]	2	[5, 6, 3, 4, 1, 2]
[1, 2, 3, 4, 5, 6]	3	[4, 5, 6, 1, 2, 3]
[1, 2, 3, 4, 5, 6]	4	[3, 4, 5, 6, 1, 2]
[19, 34, 59, 2, 45, 83, 20]	5	[59, 2, 45, 83, 20, 19, 34]

Provide your function.

```
function out=test(in,n)
[p,q]= size(in);
out = zeros(p,q);
time = fix(q/n);
for i = 1:time
   out((i-1)*n+1:(i-1)*n+n)= in(q-(i-1)*n-(n-1): q-(i-1)*n);
end
out(n*time+1:q)= in(1: q-time*n)
%return out
end
```