COMP9334 Capacity Planning of Computer Systems and Networks

Assignment – Part A

Objective:

Learn statistical techniques to characterise workload of a system.

In the lecture of week-1, you learnt the minimum spanning tree algorithm. The matlab software has support for a number of functions in its Statistical toolbox. For this task, you are going to use a data file called *resource.dat*. This contains three columns (Disk1 usage, Disk2 Usage and CPU Usage).

You are going to accomplish the following tasks:

1. Read and execute the code example1.m to learn a few matlab functions that will be useful for this task. At this point, you should learn *pdist()* function and various methods such as *city block , Mahalanobis, Minkowski* etc. used for distance calculation. Create a scatter plot of first two columns of your data to get a visual feedback.

2. Read the documentation for a function called *linkage()* for clustering. It uses a number of methods. Become familiar with these methods.

3. Find pairwise distances using the default Euclidean distance.

4. Find two type of linkages '*single*' and '*complete*'. Read documentation on *dendrogram()* function. Plot dendrogram for each of these cases using the function *dendrogram* and add proper title.

5. Find the value of *y* for which you can achieve three clusters. Which of the two methods provides better clustering and why?

6. Get the membership for 3 clusters using function *cluster()*. Include these values in your report. You may wish to use *find()* function to extract these values from array storing membership information.

7. Learn about *cophenetic coefficient*. Calculate and compare the cophenetic coefficient for single and complete linkage using *cophenet()*. Which method provides better representation?

Please include various outputs and figures in your report.

There are many more functions that you can learn to perform clustering. I encourage you to read and play with another function kmeans() in your own time. It paritions the data into k groups such that within-group sum of squares is minimised.
Note that same techniques are used for classification purposes in AI and other fields.

Submission Instructions:

1. The submission deadline is 23:59:59 Wednesday 28th March 2012 .
2. The total marks for this part is 10 marks. Please note that other parts will be released in due course.
3. You are required to submit a written report detailing the work that you have done. It must be in Acrobat "pdf" format. It must be called Ass1PartA.pdf". : Your report should complete matlab script as an appendix in this report. When you are ready to submit, type 9334 at the bash prompt and then the command: give cs9334 Assignment Ass1PartA.pdf (please check notice board for any change in these instructions)
4. Please note that the system will only accept "Ass1PartA.pdf " as the filename for submission. Also, note that the total size of your submission should be smaller than 2 MBytes. If you still have difficulty, email cs9334@cse.unsw.edu.au for instructions.
5. You can submit multiple times before the deadline. The latest submission overrides the earlier submissions, so make sure you submit the correct file. Do not leave until the last moment to submit, as there may be technical or communications error and you will not have time to rectify that.

Late Submission Penalty: Late penalty will be applied as follows:

- 1 day after deadline: 10% reduction
- 2 days after deadline: 20% reduction
- 3 days after deadline: 30% reduction
- 4 days after deadline: 40% reduction
- 5 or more days late: NOT accepted

NOTE: The above penalty is applied to your final total. For example, if you submit your assignment 1 day late and your score on the assignment is 10, then your final mark will be 10 – 10*(10% penalty) = 9.

Plagiarism: You are to write all of the code for this assignment and produce the report yourself. The LIC will decide on appropriate penalty for detected cases of plagiarism. The most likely penalty would be to reduce the assignment mark to ZERO. We are aware that a lot of learning takes place in student conversations, and don't wish to discourage those. However, it is important, for both those helping others and those being helped, not to provide/accept any programming language code in writing, as this is apt to be used exactly as is, and lead to plagiarism

penalties for both the supplier and the copier of the codes. Write something on a piece of paper, by all means, but tear it up/take it away when the discussion is over.