COMP3702: Assignment 3 Discussion Paper

Group: FAKE BRAINS
Members: Jonathan Holland, Michael Ball, Xavier Casley

TASK 2 Ouput:	Data 1	Data 2	Data 3
Likelihood	2.11510783532E-11	4.63925755586E-59	0.0
Log Likelihood	-24.5793302257656	-134.317966142725	-23764.7163014954

Task 3 Discussion:
As the amount of training data increases the CPTs tend to be more accurate. The likelihood though tends toward 0, as each new dataset makes it's value smaller. In cases where there is surplus data, such as in CPTNoMissingData-d3.txt which has 5000 sets, this results in such a small number that java computes it as 0. Log-likelihood however can still cope with values in these cases and shows a high magnitude number, as expected. This means that something other than likelihood needs to be used to value a network/dataset pair, as the likelihood is too dependant upon dataset size.

Increasing the number of nodes in the network also decreases the likelihoods. Similarly to increasing the datasets, increasing the number of nodes means more multiplication of a value < 1, and likelihood values of closer to 0. It should be noted that increasing the number of nodes has less of an effect than changing the size of the dataset.

As the possible values of each variable increase it would be expected that the final likelihood would also increase. This can be expected to some degree, however for each additional probability due to either additional nodes or data sets this will still decrease the likelihood and log likelihood dramatically. For larger datasets (eg. CPTNoMissingData-d3) the log-likelihood should be used over the general likelihood to ensure an accurate answer.

Task 5 Discussion:
The likelihood and log likelihood equations don't take into account the size of the dataset in a manageable way, despite the effect a different sized dataset has on the values returned (The accuracy of the fir for smaller datasets is questionable). This is why the Score(G) is important. It provides a counterbalance to account for the size of the data being examined via a constant adjustable parameter.
When changing the parameter C we found that largely changed the magnitude of the scores.

Task 6 Discussion:
Given enough time, the structural complexity between the network from the best tree initialisation method and the no edge method (adding and subtracting edges at random and then comparing the resultant log-likelihood) is minimal. The random chain method however is

significantly different as it makes the assumption that each parent has only one child. This is extremely unlikely in the case of most networks where elements are dependent upon more than one factor. Because of this, the resultant scoring of the random chain method is worse than the other two for all cases bar those unique systems where a chain is applicable. Despite this, it is still sometimes a viable option even for networks where it obviously isn't the best fit.This is because it is a much faster and less computationally expensive option; for every extra edge in the other two methods, the combination of all directions needs to be picked to optimise the network's fit as well and this adds significant complexity.

Of the two remaining options, the trade-off is largely dependent on the size of the network/nodes to be assessed. For a larger number of nodes, attempting to randomly add and subtract edges to reach an optimal solution requires a significant amount of computational time. On top of this, if the balance between adding and subtracting is not ideal, the optimally fitted network may never be reached using this method. However, on smaller data sets, this method is more likely to achieve good results as it can simply run through and compare all of the possibilities. This makes it superior in fitting smaller data sets as the best tree init method does not guarantee the optimal result. On the other hand for larger networks the best tree init method is superior as there is less random activity involved and it can reasonably quickly find a well fitted solution.