Assignment 2 Marking Guide

Assignment 2 is a lot more open-ended than the first one. Therefore, the report is a central piece of the assessment. For this reason, we increased the report mark to 30% of the total mark. The report should address two main topics: a description of the algorithm (item 2 of the marking criteria, see below) and the assumptions/simplifications made in the model (item 3 of the marking criteria).

The remaining of the marks (70%) will be distributed according to: the efficacy of the code (item 1 of the making criteria and 50% of the total mark); the quality of the code (item 4 of the marking criteria and 10% of the full mark); the efficiency of the implementation (item 5 of the marking criteria and 10% of the total mark).

A script will assess code efficacy and efficiency. The quality of the code needs to be marked by manual inspection.

In summary, the marking criteria in the following:

1. 50% of the mark will be determined by the cost incurred by your code after several days of simulated data. The mapping from cost to marks will be determined after the assignment has been submitted.
2. 20% of the mark will be determined by the description of the algorithms used, and a short justification of the methods used.
3. 10% of the mark will be determined by a description of the assumptions and/or simplifications you made in your model, and whether those assumptions would be effective in the real-world.
4. 10% of the mark will be determined by the quality and readability of the code.
5. 10% of the mark will be determined by the efficiency of the code.

In the following, we explain how to assess each of those items.

1. [50 marks] This item is automarked. A script runs each assignment and simulates 10 days of data. We took the distribution of scores and set values for the variables TOP and BOTTOM. The final scores are computed by a linear function defined by these two values described bellow:

   ```
   TOP = 58000
   BOTTOM = 130000
   # linear interpolation between TOP and BOTTOM
   def mark(score):
       if score < TOP:
           return 50
       elif score <= BOTTOM:
           return score*(50/(TOP-BOTTOM))+(BOTTOM*(-50/(TOP-BOTTOM)))
       else:
           return 0
   ```

2. [20 marks] This item consists of a description of the algorithms used, and a short justification of the methods used. Some questions to consider while marking:

   - Is the model a graphical model? We accept any PGM: Bayesian, Markov, Hidden Markov, Conditional Random Fields, etc. If the model is not a PGM, it is not a valid assignment.
   - What sort of query is being posed to the model? Full marginal queries are the best option to minimise expected cost, but MAP or max marginal queries can also be used well. Are they computing exact or approximate answers? Is the inference method efficient?
   - How are they learning the model parameters. Is the learning algorithm appropriate for the model? Are they using some smoothing or regularisation?
   - How is the result of the query being used to compute the action of minimum cost? Is the model incorporating the costs somehow to transform the probabilities into actions?
   - If parts of the inference algorithm are hard coded (not learned), is this well motivated and reasonable?

   Use the following scheme to assess this item:

   1. [10 marks] Learning algorithm, smoothing and regularisation. Be careful since the students are not required to submit the source code of learning part of the assignment (only inference). But the report must address the learning approach.
   2. [10 marks] Inference algorithm, approximations, query and conversion from probability to action.

3. [10 marks] These marks are determined by a description of the assumptions and simplifications made in the model, and whether those assumptions would be effective in the real-world

   - Is the model overcomplicated? For instance, if each variable in the model has large outcome spaces, such as {1,2,3,...,17,18,19,20}. Usually, overcomplicated models tend to overfit the data, so check the score in (1). Overcomplicated models should be marked down unless the report has a good explanation for it.
   - Is the model sound? We have an application with multiple interactions, so time is an important variable. A sound model should incorporate time in some way. Also, is the model a single model or a collection of models, as it happens with CRF? In the case of multiple models, it is important to have an explanation of how these independent models are integrated.
   - Examples of reasonable and approximately true assumptions include: each person has the same preferences and people are just as likely to go through a door in one direction as the other direction.
   - Did the student make assumptions such that nearby rooms are correlated, and far apart rooms are not? (This is important)

4. [10 marks] These marks are determined by the quality and readability of the code.

   - How easy was it to check that their report matches their algorithm?
   - Is the code documented, such as a brief description of each function purpose?
   - What is the general quality of the code and organisation?

5. [10 marks] These marks are automatically computed by the auto marking script. They are based on the running time of the algorithm and are computed according to the following table:

Marks	Time
10	<200s
8	<300s
6	<400s
4	<500s
2	<600s
0	>600s