Module Code
COM00032H

BEng, BSc, MEng and MMath Degree Examinations 2019–20

DEPARTMENT OF COMPUTER SCIENCE

Machine Learning & Probabilistic Graphical Models

Open Individual Assessment

Issued: 4 March 2020 (12 noon)

Submission due: 22 April 2020 (12 noon)

Feedback and marks due: 20 May 2020

All students should submit their answers through the electronic submission system: http://www.cs.york.ac.uk/student/assessment/submit/ by 22 April 2020 (12 noon). An assessment that has been submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty.

Your attention is drawn to the section about Academic Misconduct in your Departmental Handbook: https://www.cs.york.ac.uk/student/handbook/.

Any queries on this assessment should be addressed by email to James Cussens at james.cussens@york.ac.uk. Answers that apply to all students will be posted on the VLE.

Your exam number should be on the front cover of your assessment. You should not be otherwise identified anywhere on your submission.

Rubric

- All data and any other additional files are available at the MLPG VLE site in the **Assessment** section.
- **Your submission should be a single zip file. If your submission is not a single zip file 5 marks will be deducted from your overall mark.** This zip file should contain a single PDF which contains what we call your *report*. In addition it should contain a number of files (Python programs and text files). The questions below specify what should be in your report and which files should be included in your zip file.
- All Python programs should be in Python3.
- Your Python code can assume that a working installation of PyStan is available.
- Your Python code can assume that a working installation of scikit-learn is available.
- Your Python code must run correctly when run under Linux on the Ubuntu distribution which is used in our department's software labs (i.e. the distribution you used during practicals).
- Unless otherwise indicated there are no word limits on your answers.
- All Stan models should be included as strings inside Python programs. (This is just to cut down on the number of files you have to submit.)

1 Conditional independence in Bayesian networks (10 marks)

Consider the Bayesian network structure (DAG) shown in Figure 1 on page 3.

1. List all pairs of variables which are independent (3 marks).
2. List all pairs of variables (excluding C and G) which are independent given C and G (3 marks).
3. Draw a DAG which is Markov equivalent to the DAG in Figure 1, but is not the same as it. (2 marks)
4. Draw a DAG which is not Markov equivalent to the DAG in Figure 1, but which has the same undirected skeleton. (2 marks)

Notes: You are not asked to provide any explanation of your answers. Marks for this question are determined entirely by the accuracy of your answers.

What to submit: Just put your answers in your report.

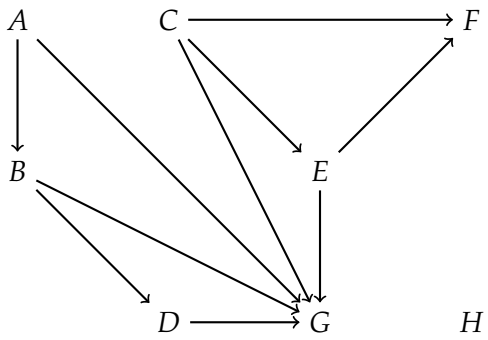

Figure 1: Bayesian network structure for Question 1.

2 House prices with STAN (50 marks)

This question asks you to use STAN to analyse fictitious data on the factors influencing house prices in two localities. Download the file `q2dat.py` from the VLE. `q2dat.py` is a Python program which defines a dictionary called `datadkt` containing the fictitious house price data. There are 4 variables for this data: L, S, A and P. L is a binary variable indicating whether the house is in locality 0 or 1. S is the size of the house (in square metres), A is the age of the house (in years) and P is the price (in £10,000) of the house. There are 90 datapoints. The first 50 datapoints have $L = 0$, the last 40 have $L = 1$.

2.1 A simple model (15 marks)

1. Write a Python program called `q21.py` which contains a STAN model which implements a standard linear regression model for predicting P from L, A and S. You do not need to define priors for any of your parameters (so STAN will just give them default priors for you.) Your program should import the data `datadkt` from the file `q2dat.py`. You should include your STAN model as a string in `q21.py`. (10 marks).

2. Use Pystan to perform Bayesian inference for this model and data. Include in your report the textual summary of posterior distributions produced by PyStan (i.e. the output which starts with the header `mean se_mean...` and ends with a line for `lp__`). Include in your report plots of the relevant posterior distributions. Explain what steps you have taken to ensure that you have computed reasonable approximations to the true posterior distributions over your parameters. Justify your choice of Bayesian inference (sampling vs variational). (5 marks)

What to submit: Submit `q21.py`. Put summary, plots, explanation and justification in your report.

2.2 A less simple model (10 marks)

1. Write a Python program called `q22.py` which differs from `q21.py` only in the following respect: there is a constraint that states that size has a positive effect on price. (5 marks)

2. Use Pystan to perform Bayesian inference for this model and data. Include in your report the textual summary of posterior distributions produced by PyStan. Include in your report plots of the relevant posterior distributions. In fewer than 40 words analyse the difference, if any, between these results and those you obtained for (2.1). (5 marks)

What to submit: Submit `q22.py`. Put summary, plots and analysis in your report.

2.3 Two models (10 marks)

1. Write a Python program called `q23.py` which has a separate linear regression model for each of the two localities. Both models should use the constraint that size has a positive effect on price. Neither of these two models should use the L variable. (5 marks)

2. Use Pystan to perform Bayesian inference for this model and data. Include in your report the textual summary of posterior distributions produced by PyStan. Include in your report plots of the relevant posterior distributions. In fewer than 40 words analyse the difference, if any, between these results and those you obtained for (2.2). (5 marks)

What to submit: Submit `q23.py`. Put summary, plots and analysis in your report.

2.4 A compromise model (15 marks)

The problem with the approach used in (2.2) is that a single model is used for both localities—it could be that the two localities are so different that having only one variable (L) to distinguish between them is too crude. On the other hand, the problem with the approach used in (2.3) is that the two localities are treated entirely separately which is also problematic since it is reasonable to assume that factors affecting house prices are not entirely different in different localities.

1. Write a Python program called `q24.py` which contains a STAN model that effects a compromise between the models of (2.2) and (2.3). There should be separate regression parameters for the two localities (like in (2.2)) but no regression parameter should be independent of any of the 90 datapoints (unlike in (2.2). (5 marks)

2. Use Pystan to perform Bayesian inference for this model and data. Include in your report the textual summary of posterior distributions produced by PyStan. Include in your report plots of the relevant posterior distributions. (5 marks)

3. Include in your report a Bayesian network representation of the STAN model you have created for this question and (in fewer than 50 words) a justification for the model you have chosen. Note that there are a number of equally reasonable models for this question. (5 marks)

What to submit: Submit `q24.py`. Put summary, plots, Bayesian network and justification in your report.

3 VB vs MCMC (10 marks)

In fewer than 200 words overall: (i) describe Hamiltonian MCMC, (ii) describe variational inference as done in Stan and (iii) discuss the pros and cons of both approaches. (Any equations or figures do not count towards the word count.)

What to submit: Put your answer to this question in your report.

4 Hidden Markov models (15 marks)

Download the file `q4hmm.txt` from the VLE. This file provides transition and emission probabilities for an HMM which has 4 states: $(0,0), (0,1), (1,0), (1,1)$, each of which can emit one of 3 outputs: -1, 0 or 1. You can think of each state as a cell in a 2×2 grid where it is possible to transition from any cell to any other cell. The initial state distribution is represented as a transition from a dummy state `(None,None)`.

Download the file `q4.dat`. This file has 7 sequences for this HMM (of lengths 5,7,10,3,12,5 and 7).

Write a Python program called `q4.py` to compute both forward and backward probabilities for each of these sequences using the transition and emission probabilities from the file `q4hmm.txt`. Your computed probabilities should be written to a file called `forward.txt` and your backward probabilities to a file called `backward.txt`.

The format for `forward.txt` should be as follows:

```
Sequence 0, length 5:
(0,0) * * * * *
(0,1) * * * * *
(1,0) * * * * *
(1,1) * * * * *

Sequence 1, length 7:
```

```
(0,0) * * * * * * *
(0,1) * * * * * * *
(1,0) * * * * * * *
(1,1) * * * * * * *

...
```

where each $*$ should be replaced by the relevant forward probability. So output for each sequence is kept separate and each sequence gets a header as indicated. The forward probabilities for a given state and sequence should be written in order with the forward probability for timepoint $t = 1$ coming first. The format for backward probabilities should be the same. (10 marks)

Include in your report an explanation of how forward and backward probabilities are used in the EM algorithm for learning the parameters of a HMM. Use fewer than 60 words for this. (5 marks)

What to submit: Submit `q4.py`, `forward.txt` and `backward.txt`. Put your explanation in your report.

End of examination paper