PC5252

Problem set overview

Overview

4 problem sets (each worth 15% of your final grade)

- Will be released on Canvas before Tue lecture in Weeks n = 2, 5, 8, 11
- Will be discussed during Fri tutorials in Weeks n and n+1
- *Due before Tue lecture in Week n+2
- Solutions will be discussed during Fri tutorial in Week n+2
- Solutions will be posted on Canvas after Fri tutorial in Week n+2
- *Will be graded and returned by end of Week n+2

Each set should take < 9 hours - if not, please ask for help

*Except final set

Submitting your solutions

Either written or typeset is fine

Either hard or soft copy is fine

Summarise any numerical work properly: Plots and short explanatory text

Any solutions purely in the form of code (even annotated) will not be graded

Only 1 submission per set - please collate and finalise your solutions before submitting them. Resubmission may be allowed if there is good justification

TAs will provide individual feedback on your graded solutions, and discuss the more common mistakes during tutorials

Late submissions

If you submit late but before solutions are posted + you have asked for an extension, there will be no penalty to your grade

If you submit late but before solutions are posted + you have NOT asked for an extension, there will be a small penalty to your grade (-1.5 marks)

If you submit after solutions are posted, there will be a significant penalty to your grade. Essentially, you will only get “effort” marks that will be awarded up to my discretion (in order to be fair to those who did submit on time)

Symbolic calculations

For proof/derivation-type questions (“Show that…” or “Derive…”): Do by hand

For everything else: Use any method you want. The point is to assess your understanding, not your ability to perform complex mathematical manipulations

Computer algebra software:

- Mathematica
- Maple
- MATLAB (limited), SymPy, etc.

https://nusit.nus.edu.sg/services/software_and_os/software/software-for-student/

Numerical calculations

Some problem sets will require light coding, because Bayesian statistics and machine learning are ultimately about working with data

UNIX terminals: Native on Macs and Linux machines, can use WSL for Windows

Python: Programming language, popular for scientific computing

Anaconda: A distribution of Python (and R)

Jupyter (IPython) notebooks: Execution environment for Python (and R/Julia)

NumPy, Matplotlib, SciPy, scikit-learn, PyTorch: Useful Python libraries

Numerical calculations

No need to learn/use those specific tools if you are familiar with alternatives, e.g.:

- Stan (library for statistical calculations)
- R (popular with statisticians, biologists)
- Any other numerical platform: Julia, MATLAB, even Excel

Again, coding is not the focus of this course. The learning objectives are to:

- Know the practical means of analysing/solving Bayesian and ML problems
- Solve simple problems through any such means
- Learn how to interpret and present the results (mainly with plots)