Fall 2020

STAT 221: Computational Tools for Statistical Learning: Approximation, Optimization, and Monte Carlo

Instructor: Jun Liu; Email: jliu@stat.harvard.edu;
Teaching Fellow: Cory McCartan; Email: cmccartan@g.harvard.edu

COURSE DESCRIPTION: Focus on computational tools for statistical inference and learning. It differs from other courses on computatonal methods in its emphases on both comptuational thinking of statistics and statistical implications of inference algorithms. It also differs from traditional in-class teaching in that more hands-on problem solving activities will be organized. Topics include: basic numerical algebraic operations; optimization methods such as Newton-Raphson, bisection, and gradient-based methods; the EM algorithm, variational approximations; Monte Carlo methods including Markov chain Monte Carlo, importance sampling, and sequential Monte Carlo; stochastic gradient descent; neural networks and deep learning tools.

PREREQUISITE & REQUIREMENT: Advanced linear algrebra and advanced calculus; Statistics 110 and 111 or equivalents. Knowledge of programming, especially R, and/or Python, and/or Matlab (or C/C++). Required to implement some of the methods discussed in class as part of the problem sets, and some amount of programming in the final project.

Class Meetings: TTh 10:30-11:45 AM Eastern Time via Zoom. Class will be recorded, except for breakout room sessions.
Office Hours: TTh 9:00-10:00 PM (**Evening** Eastern Time) via Zoom.
Homework policy: You are welcome to discuss the problems with others, but you must write up your solutions yourself and in your own words. Additionally, you must list the names of the students with whom you collaborated (if any). Copying someone elses solution, or just making trivial changes for the sake of not copying verbatim, is not acceptable. Homework must be submitted as a single PDF file via the Canvas course website; no submissions on paper will be accepted. Your homework can be typeset, written using a tablet, or scanned from handwritten work, but must be clear and easily legible (not blurry or faint), and correctly rotated (e.g., not upside down). Occasionally you may be asked to submit your computer codes, which has to be in the language of R or Python, or in an excutable form with clear instructions on how to run it.
Participation: Active participation is expected, through attending class and section regularly and engaging in breakout room and other discussions. Raising questions during the class and section to clarify course-related materials, to generalize concepts, or to correct typos or mistakes of the instructor is highly appreciated and counted favorably towards the final grade. Please keep your camera on during class and section if you are comfortable doing so, though there may be occasional privacy or bandwidth reasons for turning your camera off.
Grading: Course grades will be based on a weighted average of class participation (10%), homeworks (70%), and the final project (20%). The course is letter-graded by default, but you may switch to SAT/UNSAT grading if you prefer, in which case please inform me as soon as you decide.

TOPICS TO BE COVERED (not in chronological order and subject to changes):

- Classic Numerical Methods
  - Mode finding: gradient descent, conjugate gradient, Newton-Raphson, etc.
  - Numerical integration
  - Linear programming, dynamic programming (recursion)
  - Matrix operations: QR, Cholesky, SVD, PCA, etc.
- Monte Carlo Methods
  - Acceptance-Rejection, Antithetic variates, Control variates
  - Importance Sampling, sequential Importance Sampling, sequential Monte Carlo
  - Markov chain Monte Carlo, bootstrap
  - Hamiltonian dynamcis and hybrid Monte Carlo
  - Examples and other advanced MC techniques: collapsing, grouping, multiple-try Metropolis, bridge sampling, Wang-Landau algorithm, etc.
- Modern Tools for Statistical Learning
  - EM algorithm, variational approximation, and others
  - Gradient boosting machines
  - Other topics (optional): neural networks (e.g., FNN, RNN, CNN, backpropagation), stochastic gradient descend, Variational Auto Encoder, FDR controls, etc.

REFERENCES

Fischetti, T. (2015). *Data Analysis with R*, Packt Publishing, ISBN-10: 1785288148.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning (Adaptive Computation and Machine Learning).* MIT Press. Freely available at http://www.deeplearningbook.org/

Liu, J.S. (2001). *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag: New York. ISBN-10: 0387763694.

Nielson, M (2017). *Neural Networks and Deep Learning.* Free on-line book. http://neuralnetworksanddeeplearning.com/index.html

Thisted, R. (1988). *Elements of Statistical Computing.* Chapman and Hall/CRC, ISBN-10: 0412013711.

Wickham , H. and Garrett Grolemund (2018). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data* (1st Edition). O'Reilly Media. ISBN-10: 1491910399