Stat 362 Project

For the final project you get to pick a Monte Carlo topic of interest to you, do a deep dive into it, and then write a report. It is ok, even preferable, for you to pick something that you plan to use in a future project on campus, or something connected to an area you would like to work in when you graduate. NB, you cannot get credit for the same work in two courses, so no recycling other projects.

Some of the suggested topics below will be touched on in class, but there is no way to get completely to the bottom of them in class.

Mechanics

You can work in teams of one or two people. There should be a single report that can be up to 6 pages long. In a two person project, page 7 should describe who did what. More is expected from a team than an individual.

If you want to use narrow margins, then use a two column format. In addition to your six pages, you may tack on an appendix with source code, tables or other supporting material.

Due on the last day of class. Email a PDF to owen@stanford.edu.

Topics

Below are some suggested topics. The work can be reading a handful of papers, explaining what is in them, then implementing an algorithm, extending the theory, or doing an extended example. Sometimes there is more to be learned by doing a connected set of examples where you vary aspects of the problem to see what happens. If a literature survey is your main contribution, then be sure to include works by multiple authors, not just one.

1. Get to the bottom of Hamiltonian MCMC. Read the key articles, synthesize what they have to say, form your own opinion. Do an example, either writing your own code by hand or using Stan.

2. Same as the previous one but for variational Bayes. The issue with VB is that there is always doubt about whether sampling VB is close to what you get sampling the thing you've approximated.

3. Same as above but with multilevel Monte Carlo.

4. Same again but for approximate Bayesian computation.

5. There is some interesting work by Michael Mahoney (now at UC Berkeley) on randomized linear algebra. Survey, implement, compare and summarize some of the methods. See also Achlioptas' work on random projections.

6. Sometimes people solve differential equations by running Monte Carlo samples. One cool idea there is called 'walks on spheres'.

7. Read about the equi-energy sampler and compare it to parallel and multiple tempering. (Read about those too!)

8. Sampling statistical tables with given margins. Start with work by Yuguo Chen, Persi Diaconis and Susan Holmes.

9. There are numerous good papers by Roberts and Rosenthal on convergence rates of MCMC algorithms.

10. The best way to find the volume of a high dimensional set is often by Monte Carlo, especially the hit and run algorithm. Also Mark Huber has a tootsie-pop algorithm.

11. Become very good at using a probabilistic programming language such as Stan or Edward or pymc. In order to show that you're good at it, you should solve an ambitious MC problem with your chosen tool, not just that language's version of 'hello world'.

12. Develop a theory of gain coefficients for randomized Halton points. These are randomized quasi-Monte Carlo concepts. Can you get a universal bound for them?

13. The course notes mention some ways to simulate networks. There are many more.

14. Sparse grids.

15. Sequential Monte Carlo and sequential quasi-Monte Carlo.