Statistical computing and visualization (Stat 221)

Overview

Term: Spring 2013
Duration: 12.5 weeks, 2 lectures per week
Instructor: Sergiy Nesterko
TF: Alex D'Amour
Twitter: twitter.com/harvardstat221
Website: theory.info/harvardstat221

This is a graduate-level class aimed to equip the PhD graduate students with practical distributed computing and visualization tools to be successful in the program and beyond. The course will also be accessible to motivated undergraduates.

Course material is taught on cases and examples representing recently completed stages of projects in research and industry. Final projects are based on open problems in research and industry, and involve statistical computing OR visualization, or both.

Final project providers contribute project topic descriptions and data for the exercise. Data and confidential information is provided to the engaged students under NDAs if appropriate, not open sourced to the full class. Examples of projects:

- Academic. Example: homophily-based community detection in social networks, helps bridge classical community detection methods and the concept of homophily.
- Industry. Example: the company is introducing a new product, based on market research data need to model market and competitor response to optimally price the product to avoid internal product cannibalization.

Main titles: Maximum likelihood, EM, MCMC, Gibbs, HMC and variants for Bayesian modeling, visualization

Themes and tricks:

- Fast R code
- Interactive visualization
- Odyssey distributed computing (related bash, jobs, Python script to catch disappearing jobs, queues)

- Code aesthetics (generic, well-commented code)
- MPI / rMPI
- Debugging the code (browser(), logging)
- Likelihood computation
- Algorithm tuning
- Model nonidentifiabilities (connection to posterior surface, ways to diagnose/fix them - setting parameters to fixed values, convergence diagnostics)
- Posterior predictive checks
- Frequentist model evaluation
- Parsimony in visualization
- Parsimony in modeling (simplest method that does the job)

Material breakdown: 30-40% visualization, 60-70% computing

Tools:

- Computing related: Odyssey, R, some bash, some Python
- Visualization related: R, Javascript/HTML/CSS, d3.js, jQuery
- A motivated computer literate student will be able to learn as we go

Group work will be allowed and encouraged. Undergraduates will work with graduate students. The format of group work will be formalized upon observing the graduate/undergraduate ration in the class.

Graded assignments structure:

- 6 homeworks, each with computing and visualization tasks. Extra points allow to drop one homework.
- Class participation
- Final project
- No final exam

Course grade breakdown: 10% participation, 55% homework, 35% final project.

Course timeline

When	What	Notes
Week 1 Jan 28 - Feb 3	Bayesian computing example: meet R, Odyssey, d3.js Language definitions - data, parameters, models d3* - basics: html, css, javascript - how it looks and how it debugs	A networks example: fast R code to generate, Odyssey to simulate, d3.js to visualize. Supporting Odyssey bash code.

Week 2 Feb 4 - Feb 10	Modeling intro + Maximum likelihood d3* - enter/exit selection, svg elements and their styling from CSS, basic code to put them on and manipulate	Homework 1 is due Examples: Exper. design, network homophily model, astrostat example
Week 3 Feb 11 - Feb 17	Modeling intro + Maximum likelihood d3* - transitions, intro to event listeners, reinforcing enter/exit selection concept. By this point, student will understand how to create basic visualizations	
Week 4 Presidents day on Feb 18; to Feb 24	EM + missing data d3* - interpolators, working with d3 shapes (lines, areas, arcs)	Homework 2 is due Examples: integration vs maximization
Week 5 Feb 25 - Mar 3	EM + missing data d3* - interpolators, working with d3 shapes (lines, areas, arcs)	
Week 6 Mar 4 - Mar 10	Gibbs d3* - axes component, visualization controls	Homework 3 is due Examples: astrostat
Week 7 Mar 11 - Mar 16, Spring Recess	MCMC + Gibbs d3* - axes component for categorical data, more visualization controls/DOM manipulation	Examples: experiment design
Week 8 Mar 25 - Mar 31	MCMC + Gibbs Guest presentation/lecture d3* - native data manipulation: sorting, keys, nesting	Homework 4 is due
Week 9 Apr 1 - Apr 7	HMC d3* - putting it together, more on event listeners and DOM manipulation	Examples: experiment design, rater clustering, network homophily
Week 10 Apr 8 - Apr 14	HMC vs MLE + Information d3* - resizing, dragging, and zooming behaviors	Homework 5 is due

Week 11 Apr 15 - Apr 21	MPI (tentative) d3* - putting it together, intro code abstraction	Example: Logistic regression MLE via MPI for massive data
Week 12 Apr 22 - Apr 28	HMC (vs point estimation tentative) d3* - more on code abstraction	Homework 6 is due
Week 12.5 Apr 29 - May 1, two lectures	Final lecture Final project info/feedback/celebration	

*all d3.js material is extra credit.

Extended information

Homework grading

Homework grading system is inspired by the way it is treated in CS50.

The work on this problem set will be evaluated by four dimensions, the grades for the dimensions are summed up to form the total grade for the problem set.

Correctness. The code does the needed task, and is consistent with the problem statement.
Design. The extent to which the code and visualizations are well-crafted (clearly, efficiently, logically).
Style. Readability of the code for computing and visualization tasks (good code is like a book, one reads the comments, variable names and language statements, and understands what it does).
Write-up. If the problem set needs written non-code answers, or the visualizations need descriptions, the clarity and correctness of the writing with respect to the task.

Each question will have the maximum points in each dimension displayed. For example, (C5, D5, S5, W5) means the question can contribute a maximum of 5 points in each dimension to the final grade. If a particular dimension is not present in a question, it won't be displayed.

Homework points are transferable - if a student gets over 100% on a homework, the extra points are transferred to other homeworks.

d3.js component

Interactive visualization will be build into each homework assignment along with static visualization component. Interactive visualization will be extra credit and will replace the static component if chosen by the student. The extra credit will allow the student to get over a 100% on each homework.

Final projects

TBA