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, Masters, and motivated undergraduate students with practical distributed computing and visualization tools for scientifically rigorous work on data-intensive problems.

In this class, we will build a way of thinking about quantitative problems regardless of their seeming complexity. The class covers all stages of a data analysis problem: (1) setting up the problem, (2) designing a method to solve it, (3) implementing the related computation, (4) presenting the findings. We will uncover the components of each stage, demonstrate the interaction between all stages (1-4), and discuss their practical implementation.

In the lectures, we will apply this way of thinking to computing algorithms, traditional statistical models, latest advances in statistical research, and case studies from industry. We will discuss how efficient programming enables problem solving for data analysts. We will showcase the recent advances in visualization and their connection to the statistical world.

Problem sets guide students through practical computing problems, inviting to fill in missing parts of the code and highlighting the fundamental statistical questions along the way. There is also an interactive visualization component in each homework for extra credit.

Final projects are based on data-driven problems in research and industry. Working on the final project, the students will go through all 4 stages of a data analysis problem. Course partners view final projects as high priority and look forward to the solutions generated by the students.

Keywords

Main titles: Maximum likelihood, EM, MCMC, Gibbs, HMC and variants for Bayesian modeling, MPI, 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
- Visualization related: R, Javascript/HTML/CSS, d3.js, jQuery
- A motivated computer literate student will be able to learn as we go

Course timeline

When	What	Notes
Week 1 Jan 28 - Feb 3	Introduction of the class. Examples of Visualization+Modeling+Computing (VMC) combination and its importance in data analysis. d3* - basics: html, css, javascript - how it looks and how it debugs	
Week 2 Feb 4 - Feb 10	Guest lecture, more on VMC, a rigorous look at interactive visualization	

	d3* - enter/exit selection, svg elements and their styling from CSS, basic code to put them on and manipulate	
Week 3 Feb 11 - Feb 17	Modeling and Likelihood with examples: probit regression, Hidden Markov Models. Guest appearance. Feb 17 - start work on final projects d3* - transitions, intro to event listeners, reinforcing enter/exit selection concept. By this point, student will understand how to create basic visualizations	Homework 1 is due
Week 4 Presidents day on Feb 18; to Feb 24	Likelihood principle, Maximum Likelihood estimator, parallel computing. d3* - interpolators, working with d3 shapes (lines, areas, arcs)	
Week 5 Feb 25 - Mar 3	MLE, Likelihood and latent variables/missing data d3* - interpolators, working with d3 shapes (lines, areas, arcs)	Homework 2 is due
Week 6 Mar 4 - Mar 10	Expectation-Maximization (EM) algorithm for missing data d3* - axes component, visualization controls	
Week 7 Mar 11 - Mar 16, Spring Recess	EM with MCMC, data augmentation, Gibbs sampler; conditional independence logic d3* - axes component for categorical data, more visualization controls/DOM manipulation	Homework 3 is due
Week 8 Mar 25 - Mar 31	Hamiltonian Monte Carlo (HMC) algorithm for continuous parameters d3* - code abstraction, building visualization controls	
Week 9 Apr 1 - Apr 7	MLE, MAP, MCMC estimation and decision theory - how they fit together; non-embarrassingly parallel (NEP) computing, MPI d3* - putting it together, more on event listeners and DOM manipulation	Homework 4 is due
Week 10 Apr 8 - Apr 14	Parallel Tempering with MPI, other NEP techniques d3* - resizing, dragging, and zooming behaviors	
Week 11 Apr 15 - Apr 21	Troubleshooting complicated situations in computing and modeling; more on static and interactive visualization	Homework 5 is due

	d3* - putting it together, intro code abstraction	
Week 12 Apr 22 - Apr 28	Good statistical programming; summary of tradeoffs in modeling/computing/visualization d3* - more on code abstraction	
Week 12.5 Apr 29 - May 1, two lectures	Effective presentation of findings; final project info/feedback/ celebration	

*all d3.js material is extra credit.

Course assignments

Grading

Graded assignments structure:

- 5 homeworks, each with computing and visualization tasks. Extra points allow to drop one homework via using rollover of extra points to other homeworks.
- Class participation
- Final project
- No final exam

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

Group work is 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.

Homework grading

Homework grading system is inspired by 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 a student to get over a 100% on each homework. Extra points can be carried over to other problem sets.

The languages used in for interactive visualization component are Javascript, HTML, and CSS. Students are encouraged to brush up on the corresponding technical skills as the in-class learning curve for programming languages may be steep for some students.

Good introductory resources for the used languages include the Codeacademy's Javascript, HTML, and CSS tracks.

Final projects

Final projects are an important component of the class. They are intended to cover the full cycle of the process to solve a quantitative challenge in all rigor. The final projects are intended for the students to apply and further enhance the skills they have acquired or improved during the course of the class in the areas of statistical modeling, computing, or data visualization.

The work on the final projects is expected to be carried out for the time period of 10 weeks starting February 17 and ending April 28.

The work on the final projects can be performed individually or in teams of up to 4-5 students, with the expected average team size of 3. In the process of the project, the teams will produce 2 updates on the progress, the final write-up, and a short presentation of its outcome.

Students have an option to work on one of final projects class by class partners and collaborators, or use their own project as the final project. The assignment of students to class

projects will be performed based on the rankings they assign to project descriptions that will be made available to the students at the start of the term. Only students taking the class for credit can participate in the final projects offered by the class partners and collaborators.

Alphabetical list of class partners and collaborators providing one or more project by institution:

- Athena Health
- Caesars Entertainment
- Deloitte
- Diffeo
- Ebay
- Harvard
- Hubway/MAPC
- IBM
- KBA/DARPA
- MIT
- Nationwide
- Risk Management Services
- Sense
- Siemens
- Starbucks

Final project timeline

Date	Event
Week of Jan. 28	Public final project descriptions distributed to students
February 10	Deadline to submit rankings for final projects, or to propose the student's own project
February 17	Team assignment is released, students start working on the projects
March 4	Deadline to submit problem statement status update
April 4	Deadline to submit design of the solution status update
April 29	Deadline to submit final project write-up
TBA	Final project presentations