Introduction to Computation for Statisticians

STAT 598Z, Spring 2013

Instructor: S.V. N. Vishwanathan
co-Instructor: Pinar Yanardag Delul
TA: Jyotiskha Datta
course Email: stat598z@gmail.com

http://learning.stat.purdue.edu/wiki/courses/sp2013/598z/start

08 January 2013

Class Details

- Classes: Tuesdays 10:30 am - 11:45 am UNIV 217
- Labs: Thursdays 10:30 am - 11:45 am HAMP 3144
- Instructor Office Hours: 12:00 noon - 1:00 pm Tue or by appt.
- Instructor Office: at HAAS 232
- TA Office Hours: 12:00 noon - 1:00 pm Thurs or by appt.
- TA Office: MATH G148

Course Description

- This is an **introductory** course in statistical computing
- You will learn about
  - basic algorithms and data structures
  - coding in Python
  - implementing statistical algorithms
- Think of this course as a blend of
  - Programming 101
  - Algorithms 101
  - Statistics 301

This is not a good course for you if you already know how to program!

Why not R?

- R is a *package* for statistical analysis
- Not a full-fledged programming language (IMHO)
- Aim of this course is not to get you to *use* pre-canned routines
- Instead we will learn how to code everything from scratch

Ideal Audience

- Well versed with fundamental statistical concepts such as
  - Probability
  - Random Variables
  - Mean and Variance
  - etc.
- Comfortable with statistical algorithms such as
  - Linear and Logistic regression
  - k-means clustering
  - etc.
- Mild familiarity with a high level programming language is desirable but not necessary
- Interested in learning how to *efficiently* code algorithms for *large scale* data analysis

Prerequisites

Required

- STAT 516: Basic Probability and Applications
- STAT 517: Statistical Inference
- MA 511: Linear Algebra
- Or equivalent . . .

Desirable

- CS 190C: Introduction to Computational Thinking
- Or equivalent . . .

Grading Policy

- Best 5 out of 6 Assignments: 10 points each
- Course project: 15 points
- Midterm Quiz: 10 points
- Finals: 20 points
- Exams are cumulative and will be held in a computer lab
  - will require pen and paper derivations as well as programming
- Class and lab participation: 5 points
  - watch for spot quizzes and programming tasks in the lab

Other policies on the course home page. Please review carefully.

Software Requirements

Python Programming

- Python 2.7 or higher (`http://www.python.org`)
- Numpy (`http://numpy.scipy.org/`)
- Scipy (`http://www.scipy.org`)
- Matplotlib (`http://matplotlib.sourceforge.net/`)
- iPython (`http://ipython.org/`)
- Enthought Python Distribution (`http://www.enthought.com/products/epd.php`)
- Emacs (`http://www.gnu.org/software/emacs/`)

LaTeX(optional)

- TexLive and related packages
- AucTeX mode for emacs

Frequently Asked Questions I

Q: Is this a tough course?

Ans: **YES**. Be prepared to put in at least 10-15 hrs a week on this course if you have never programmed before. The course will progressively get harder as we go along. So try to review the material and finish your assignments on time.

Q: Will I need to do lots of programming?

Ans: **YES**. The homework problems will increasingly become tougher and involve more and more programming. Besides, you are expected to do a non-trivial project.

Q: Will I become a guru Python programmer?

Ans: You may very well become one, but that is not the goal of this course. For most part we will stick to basic language constructs and simple syntax.

Frequently Asked Questions II

Q: Will you teach us how to use standard libraries e.g. for matrix manipulation or sorting?

Ans: Yes. We will learn to code many things from scratch but for some of the basic linear algebra routines we will be using the excellent Numpy and Scipy libraries.

Q: Will I need lots of maths to understand your lectures?

Ans: I expect familiarity with

- Linear Algebra
- Multivariate Calculus
- Probability Theory

as pre-requisites. There will be emphasis on rigor even when learning about algorithms and data structures. After all, Computer Science is all about discrete maths!

Frequently Asked Questions III

Q: Can I meet you anytime I want?

Ans: I will definitely be around during office hours. You are welcome to walk in any other time I am in my office, but do remember that I generally have busy days. To avoid disappointment it is best to book a slot via email.

Q: Do you reply to emails?

Ans: I try to reply to emails as promptly as possible. If you do not hear back from me within 3 - 4 days then please ping me during the class. Your email may have ended up in my junk mail folder!

Frequently Asked Questions IV

Q: Can I solve the HW problems collaboratively?

Ans: **NO**. The course policy clearly says:
Group discussions are encouraged to further understand difficult topics. You may consult with other students about homework problems, provided that you indicate such information (whom you consulted with, which problem, to which extent) on your solution sheet. However, you must refrain from getting direct answers from others.
Any violation will result in zero credit for the assignment.

Q: How do I submit my HW?

Ans: For problems which do not involve coding, neatly type or write the solution and submit in class. I strongly encourage the use of LaTeX and discourage the use of MS Word. For solutions which involve coding, submit a print out in class **and** send your code via email to the TA **before** the class.

Frequently Asked Questions V

Q: Is there a textbook for this course?

Ans: **NO**. This is a graduate level course and is taught without a textbook. I draw upon a number of resources and books to teach this course. Some of them can be found on the homepage. However, I will regularly post handouts with material covered in the class.

Q: Will you post notes for all topics?

Ans: Yes for almost all topics except standard ones for which I will refer you to chapters in a text book or to other standard resources.

Q: Will you use slides (e.g. powerpoint) for your lectures?

Ans: No. I usually prefer to lecture on the blackboard. Class notes will be available for download from the course home page shortly after the class.

Tentative Schedule I

Basics

- 08 Jan: Introduction
- 15 Jan: UNIX commands and Python basics
- 22 Jan: Flow control, functions

Foundations

- 29 Jan: Computational complexity
- 05 Feb: Sorting

Numerical Computing with Python

- 12 Feb: Numpy and Scipy
- 19 Feb: Plotting with Python

Tentative Schedule II

Statistical Algorithms

- 26 Feb: Random variable generation
- 19 Mar: Nearest and K nearest neighbor classifier
- 26 Mar: Kernel density estimation
- 02 Apr: K-means clustering
- 09 Apr: Gaussian mixture models

Convex Functions and Optimization

- 16 Apr: Convex functions and optimization
- 23 Apr: Logistic regression

Background Survey

- Please answer as truthfully as possible
- Can help me tailor the lectures
- Talk to me if you have any concerns or comments

Thank You!

Questions?