LECTURE 1: INTRODUCTION

STAT 545: Introduction to Computational Statistics

Vinayak Rao
Department of Statistics, Purdue University

August 23, 2016
Logistics

- **Class**: Tue/Thur 1030-1145AM, Rec 114
- **Class Website**: [www.stat.purdue.edu/~varao/teach.html](http://www.stat.purdue.edu/~varao/teach.html)
- **Class Email**: purduestat545@gmail.com
  Send all homework here
- **Instructor**: Vinayak Rao (varao@stat.purdue.edu)
  If you email me, include STAT545 in the subject
  E.g. ‘STAT545: My dog ate my homework’
- **Office**: Math236
- **Office Hours**: 1 - 2 PM Tuesdays or by appointment
- **TA**: Cheng Li
- **Office Hours**: 2-3 PM Wednesdays (MATH G171)
To report an emergency, call 911. To obtain updates regarding an ongoing emergency, sign up for Purdue Alert text messages, view www.purdue.edu/ea.

There are nearly 300 Emergency Telephones outdoors across campus and in parking garages that connect directly to the PUPD. If you feel threatened or need help, push the button and you will be connected immediately.

If we hear a fire alarm during class we will immediately suspend class, evacuate the building, and proceed outdoors. Do not use the elevator.

If we are notified during class of a Shelter in Place requirement for a tornado warning, we will suspend class and shelter in [the basement].

If we are notified during class of a Shelter in Place requirement for a hazardous materials release, or a civil disturbance, including a shooting or other use of weapons, we will suspend class and shelter in the classroom, shutting the door and turning off the lights.

Grading

Homework: 30%
Midterm 1: 20%
Midterm 2: 20%
Project: 25%
Class participation: 5%
Seven assignments involving reading, writing and programming are vital to doing well in the exams. Late homework will not be accepted. One (worst) homework will be dropped. You may discuss problems with other students, but must:

- write your own solution independently
- name students you had significant discussions with

Academic integrity:
www.purdue.edu/odos/osrr/academicintegritybrochure.php
A nontrivial real world problem
Read and implement an algorithm from a paper
Groups of 2 (contact me first for groups of size 3)
Must submit:

- A proposal explaining problem, goals and distribution of work (mid-October)
- A report (each group member must submit their own report)
- A short presentation

Start thinking about this early!
We will use R for homework/project (more later)

- Need to know R or a language like Matlab/Python etc
- Don’t need to be an expert but willing to learn as you go

Some (undergrad engineering-level) math

- Probability and statistics: conditional densities, Bayes rule, maximum likelihood.
- Linear algebra.
- Basic multivariate calculus
We will not use a fixed textbook for this course. Will link to relevant documents over the course. For now:

- Math cribsheets:
  http://www.cs.nyu.edu/~roweis/notes/matrixid.pdf
- Old and new matrix algebra (T. Minka):
- R-manual
  http://cran.r-project.org/doc/manuals/R-intro.pdf
That said, there are relevant textbooks for reference:


http://www-bcf.usc.edu/~gareth/ISL/
PROGRAMMING IN R:

- Don’t need to be an expert, but be willing to learn.
- Won’t be graded for programming elegance.
- Reading assignments and homeworks will guide you.
- But you need to experiment yourself!
- If you’re stuck, the internet is your friend.

Setting up your computing environment:

- Important you have access to R/a text-editor/a compiler.
- Your Purdue account should have all of this.
Probability, statistics and computation

Probability:

• analysis of random phenomena (properties of probability distributions and models).

Statistics:

• the study of the collection, organization, analysis, interpretation and presentation of data

Computation:

• Vital for stat. analysis of large datasets/complex models
• Storage/representation/manipulation of data
• Development and analysis of algorithms

• One view: ‘Who cares?’ But¹:

• Statistical Computing: ‘Application of Comp. Sci. to Statistics’
  Tools: programming, software, data structures and their manipulation, hardware (GPUs, parallel architectures)
  E.g. Releasing software to the world?

• Computational Statistics: ‘Design of algorithms for implementing statistical methods on computers’
  Statistical methodology/algorithms E.g. Writing a paper?

This course: a bit of both (but mostly latter)

¹C. Lauro. Computational statistics or statistical computing, is that the question?
We will look at algorithms to:

- optimize loss/utility functions
- integrate functions
- generate/use random numbers (Monte Carlo methods)

Often these attempt to solve the same statistical problems

We will look at

- general purpose algorithms
- algorithms that exploit specific structure (e.g. Gaussianity, conditional independence etc.)
- frameworks for developing new algorithms
TOPICS COVERED (TENTATIVE)

• Numerical integration (Newton-Cotes, Gaussian quadrature) and differentiation.
• Combinatorial Optimization
• The EM algorithm
• Monte Carlo methods
• Markov Chain Monte Carlo methods
• Gradient-based optimization methods (conjugate gradient, quasi-Newton)
• Convex Optimization
R: a programming environment for statistical computing.

Based on Bell Labs’ S language by John Chambers

Started by Ihaka and Gentleman at the Univ. of Auckland


A high-level interpreted language with convenient features for loading, manipulating and plotting data

- Free, open source.
- A huge collection of user-contributed packages to perform a wide variety of tasks
- Widely used in academia, and increasingly popular in industry
Install R from http://cran.r-project.org/

Interact with R via the prompt, GUI or scripts.

```r
> print('Hello world')
[1] "Hello world"
```

RStudio provides a more convenient *Integrated Development Environment* (IDE) to interact with R

Layout includes editor, console, workspace/history/plots/packages/files tabs

Convenient user interface with e.g. point-and-click options

You should install RStudio Desktop (available at rstudio.org)
> x <- rgamma(3, 2, 1)  # Generate Gamma(2,1) variables
[1] 0.6768685 1.5953583 0.7012949
> z <- sum(x)
[1] 2.973522
> p <- x / z  # Normalize by sum
[1] 0.2276319 0.5365215 0.2358466
> # A random (Dirichlet distributed) prob. vector
> sum(p) == 1
[1] TRUE

> # Careful
> 1.2 == 3 * .4
[1] FALSE
> all.equal(1.2, 0.3*4)  # Can specify tolerance
[1] TRUE
my_dirichlet <- function(n, shape, rate=1) {
  len <- length(shape);
  x <- matrix(rgamma(n*len, shape, rate),len,n)
  z <- colSums(x)
  p <- t(x) / z    # Column-major ordering
  return(p)
}
Using a script

```r
> source("my_dirichlet.r");
> n <- 2
> shape <- c(2,2,2);
> p <- my_dirichlet(n, shape);

[,1] [,2] [,3]
[1,] 0.2756446 0.20718038 0.517175
[2,] 0.3827195 0.09951152 0.517769

> n <- 1000;
> p <- my_dirichlet(n, shape)
> dt <- data.frame(x=p[,1], y=p[,2], z=p[,3])
> plot <- ggtern(dt, aes(x,y,z)) + geom_point() +
    tern_limits(labels=seq(.2,1,.2),breaks=seq(.2,1,.2))
> plot <- plot + xlab("p1") + ylab("p2") + zlab("p3")
> plot
```
The package `ggtern` is not installed by default.

```r
> install.packages("ggplot2")  # Download package
> library("ggplot2");          # Load package
> install.packages("ggtern")   # Download package
> library("ggtern");          # Load package
```
> n <- 10000;
> shape <- c(2,2,2);
> n <- 10000;
> shape <- c(20, 20, 20);
> n <- 10000;
> shape <- c(20, 2, 2);
R doesn’t have its own dirichlet generator, but we can download a package.

```r
> install.packages("gtools");
> library(gtools);
> rdirichlet(2, c(2, 2, 2))
   [,1]  [,2]  [,3]
[1,] 0.1739971 0.6290579 0.1969450
[2,] 0.1537574 0.5241740 0.3220686
```

Can we verify our code is correct?
IF YOU’RE STUCK:

```r
> help(rgamma)
> example(rgamma)
```

R-manual:
http://cran.r-project.org/doc/manuals/R-intro.pdf

If you’ve used other languages:

www.johndcook.com/R_language_for_programmers.html

The internet.
No class on Thursday, Sept 10

Write up homework using **knitr** and R Markdown

Submit HWs to course email (see lecture 1)

Questions about course/homework:

- First option: post on piazza
- For specific problems, email me or discuss at office hours
G.H. Givens and J.A. Hoeting.  
*Computational Statistics.*  

G. James, D. Witten, T. Hastie, and R. Tibshirani.  
*An Introduction to Statistical Learning: with Applications in R.*  

C. Lauro.  
Computational statistics or statistical computing, is that the question?  

N. Matloff and N.S. Matloff.  
*The Art of R Programming: A Tour of Statistical Software Design.*  

Paul Teetor.  
*R cookbook.*  
O’Reilly, Beijing, 2011.