Introduction to Numerical Methods

Lecture 1

Kevin McAlister

University of Michigan

January 13, 2017

Intro to Class and Class Requirements

What is this class?

This is POLSCI 514.

- Continuation of previous semester's 514.
- Cover more complex and abstract concepts related to computing.

Coding

All coding should be done in R.

- For one week (later in the semester), we will be discussing data parsing using Linux one-liners. I'll let you know what to do for that week.

R code should follow Google R Style Guide.

- All lines after a curly bracket should begin after curly brace and be one indentation forward.

Coding

```
my.ols <- function(x,y){
  #Add Intercept to Manifest Matrix
  xx.w.int <- cbind(rep(1,dim(x)[1]),x)
  #Generate OLS estimates of beta
  ols.est <- solve(t(xx.w.int)%*%xx.w.int)%*%t(xx.w.int)%*%y
  #Calculate Residuals
  pred.vals <- xx.w.int%*%ols.est
  resids <- y - pred.vals
  rss <- sum(resids^2)
  #Find Variance of Residuals
  sig2 <- rss/(dim(xx.w.int)[1] - dim(xx.w.int)[2])
  #Standard Errors
  ses <- sqrt(diag(sig2*(solve(t(xx.w.int)%*%xx.w.int))))
  #Return Table
  out.tab <- cbind(ols.est,ses)
  return(out.tab)
}
```

Assignments

This course will have practice coding assignments for each topic.

- After learning the basics, the best way to move from good to great is directed problem solving.

Assignments are graded complete/incomplete

- As long as you make a concerted effort to complete the assignment, you will get a complete.

Assignments will be due two weeks after they are posted.

Assignments

All assignments should have two files submitted.

- .txt with your code
- .pdf with discussion, plots, tables, etc.

The .pdf should be created using LaTeX

- If you need help with LaTeX, let me know. If enough people want a week devoted to LaTeX, we'll do it in class.

Why LaTeX?

- Math is so much easier.
- It looks so damn nice.

Project

- For the last two-ish weeks of class, I want each of you to come up with a project that utilizes some of the skills that you have developed in R.
  - The project can utilize numerical or simulation methods to answer a stats question
  - It can be a data collection and parsing project
  - Computational game theory is okay, as well.
  - Really, almost anything that uses your computer is acceptable.
- My one request is that each of you come by my office hours at least once to discuss your final project.
- Each person should create a short slide deck about the project. These slides should be done using Beamer.
- At the end of the semester, you should write up what you have done. No more than 5 pages.

Computational Complexity

Algorithms and Complexity

- R utilizes a set of base algorithms written in C and FORTRAN to compute many of the statistical models that are used.
- Even though these implementations seem relatively simple (think `lm()`), they are the culmination of 50+ years of work.
- Today's are algorithms make analysis of simple (and even somewhat complex!) problems seem trivial.
- However, much of the cutting edge research in the social sciences is beginning to leverage more complex data and data analysis.
  - This is not easily handled by simple methods.
  - Text data, Hierarchical Models, Online Big Data Classification, Neural Networks, etc.

Algorithms and Complexity

- Time is of the essence with more complex algorithms.
- Consider a naive implementation of OLS:

$$\hat{\beta} = \min_{\hat{\beta}} (Y - X\hat{\beta})^2$$

- Why don't we do this instead of solving directly through the normal equations?
- Time!

Algorithms and Complexity

- How do we measure time for algorithms?
- We could write two algorithms and benchmark their performance against one another.
- This could be arbitrarily complex.
- Think in terms of processing steps.
- Elementary operations count as one "flop".
  - Addition, Subtraction, Division, Multiplication
- How many flops does it take to complete an algorithm?

OLS Complexity

- Calculating $\hat{\beta}$ using the Normal Equations:
  1. $X'X \rightarrow P^2 \times N$ operations
  2. $(.)^{-1} \rightarrow P^3$ operations
  3. $(.)^{-1}X' \rightarrow P^2 \times N$ operations
  4. $(.)^{-1}(.)Y \rightarrow P \times N$ operations
- So, the time to calculate $\hat{\beta}$ is $P^2N + P^3 + P^2N + PN$?
- Not exactly.

Algorithms and Complexity

- There's other stuff going on under the function.
  - Reads, writes, storage, processing, etc.
- So, for relatively small $N * P$, none of this matters. It's just going to be quick.
- As N or P gets large, we can start to think about upper bounds.
- What happens as we drive N or P to infinity?

OLS Revisited

- Recall that it takes $P^2N + P^3 + P^2N + PN$ flops to calculate $\hat{\beta}$.
- Let's think about the order of this function as $N \to \infty$ and P is fixed s.t. $P \leq N$.
    1. P^2N depends fully on N as N gets large.
    2. P^3 drops out
    3. PN depends fully on N.
- Thus, this algorithm has a time complexity of $O(N)$.
- Linear!
- What about if we fix N and increase P?

OLS Revisited

- Trick question!
- Recall that $P \leq N$.
- It doesn't make sense to fix N.
- Think about as both are varying.
  1. P^2N dominates PN as $P^2N > PN$.
  2. P^2N weakly dominates P^3 as $N \geq P$.
- Thus, we say OLS is of time complexity $O(P^2N)$.
- Or $O(N^3)$ as an upper bound.

Why do we care?

- For some work, none of this matters.
- What about this?
  - I want to find all tweets on a given day that may make a claim of election improprieties. I want to utilize Latent Dirichlet Allocation to classify these tweets. I could also begin by finding all tweets that have the word "fraud", then use LDA to classify. What is the time comparison between these two approaches?
- Accuracy vs. Practical Time

Why do we care?

- Efficiency is money!
  - I run a large scale experimental platform at a large company. This company has 14 marketplaces across the world. They rely on my processing of experimental results to make changes to their individual marketplace websites in a data driven way. I can analyze everyone's data utilizing a series of t-tests in under 24 hours. However, I could also use OLS to introduce meaningful covariates. This would take 72 hours per market day.
- Even if an algorithm is more accurate, the delay in processing could result in massive losses for the company.
- Amazon's Black Search Bar

Scalability

- Common theme: How easily does my algorithm handle large amounts of data?
- This is called scalability.
- What makes OLS such an attractive option for larger data sets?
- Bottlenecks are portions of the algorithm which require the most computational time.
- OLS with small P is restricted in matrix multiplication.
- This is a good thing!

Parallelization

- Matrix multiplication is embarrassingly parallel.
- To get element $[2,2]$ of $X'X$, we only need $X[2,.]$ and $X[.,2]$.
- Modern computing can handle big data problems by utilizing multiple threads.
- This is the big idea of the first half of this class!

In Summary

- Time matters.
- Efficiency matters.
- There's a language to computing.
- We can compare algorithms without ever needing to run them.
- Problems that can be done in parallel are desirable.