Stat 610 Final Project Options

Due Thursday, December 12, 11:59pm.

You may work in groups of up to 3 for this assignment.

Please set up a time to come and talk to me: I'll give you more details and tell you about problems you might want to look out for.

Assignment

You can choose from the following options:

– *Hubs and authorities for network analysis*: Supreme Court opinions cite previous opinions, and these citations can be used to define a network. One way of characterizing nodes in a network is by assigning them "hub" and "authority" scores (https://nlp.stanford.edu/IR-book/html/htmledition/hubs-and-authorities-1.html). You will write functions that compute "hub" and "authority" scores for nodes in a network, and apply them to a network of Supreme Court opinions.

This project is based on Fowler and Jeon, "The Authority of Supreme Court Precedent" *Social Networks*, (2008) (pdf available at http://jhfowler.ucsd.edu/authority_of_supreme_court_precedent.pdf). Using the data at http://fowler.ucsd.edu/judicial.htm, recreate Figure 6 or Figure 10 in the paper, or make an analogous figure containing cases you are interested in. The website contains pre-computed hub and authority scores for the opinions. You will of course re-compute these, but you can use them to check your work.

– *Markov models for language*: A naive but sometimes amusing model for language is a Markov model. In this model, language is assumed to be a sequence in which the next word is drawn from a distribution that depends only on the current word. The model can be elaborated slightly to one in which the next word depends on the current word along with the previous m words, for some fixed value of m.

You will write functions that fit such models from text and that generate new text from the fitted model. You can use as input text either the books we used in homework 1 or text of your choice.

– *Approximate Bayesian computation for disease outbreaks*: Epidemiologists have developed models for disease spread, but these models often lead to intractable likelihoods. One strategy for fitting parameters in such models is approximate Bayesian computation (ABC), which is a simulation-based method for fitting models to data (we will discuss ABC in detail later in the course).

You will write functions that perform ABC to fit parameters in a model of the spread of a virus. This will involve:

– Drawing parameters from a prior distribution,

– Drawing data according to a probability model given the parameters,

- Computing a similarity measure between the simulated data and an observed dataset, and finally

- Keeping or discarding the samples based on that similarity.

The goal will be to recreate Figures 3a and 3c in Tony and Stumpf, "Simulation-based model selection for dynamical systems in systems and population biology", *Bioinformatics* (2010) (available at https://academic.oup.com/bioinformatics/article/26/1/104/182571), with the data provided in the supplement to that paper.

– *Model selection and post-selection inference*: In regression modeling, we sometimes want to include only a subset of our variables in the model and would like that decision to be data dependent. Two solutions to this problem are *forward stepwise selection* and *backward stepwise selection*, in which you fit a sequence of models, either adding or subtracting one variable at a time until some stopping criterion is reached. These procedures work for variable selection, but they invalidate the standard methods of inference in linear models.

You will:

- Write code implementing either forward stepwise selection or backward stepwise selection.

- Set up a simulation experiment, run your method on simulated data, obtain p-values or confidence intervals in the selected model, and report on whether the hypothesis tests and confidence intervals were valid.

– *Projection pursuit*: Projection pursuit (https://en.wikipedia.org/wiki/Projection_pursuit) is a method for finding "interesting" representations of a multivariate data set, where interesting can be defined by the user but is usually defined as maximally different from a normal distribution.

You will implement projection pursuit and run your algorithm on real and simulated data.

If you would like, you may also perform a replication or a partial replication of a published paper. The caveat is that the replication must involve a reasonable computing component: if the work would be primarily data cleaning and using packages other people have developed, it is not a good candidate for a project. If you think you have a good idea, email me with the paper, what you would like to do, and a short description about why replication would make a good project. No guarantees that I will agree with you, but you have the option.

Submission parameters

– Github repository containing your code. You are required to have at least 3 different commits.

– Github repository should also containing a file with tests for your code. If we run

```
devtools::test_dir('.')
```

the tests should run and your code should pass the tests.

– A pdf writeup describing what you did and what results you got.