CS 1305 Supervised Learning Lab

Due 8/2/18 at 4 PM

1 Reading

Before class on Thursday, read *How to Fight Bias with Predictive Policing* by Eric Siegel. In preparation for discussion, consider the following questions:

- What is the central dilemma relating to racial disparity involved with the use of COMPAS described by the article?
- Would the author agree with the idea that a better algorithm could resolve this dilemma?
- What solution does the author pose, and do you think it's sufficient to address this problem? Why or why not?

2 Cell Array

In Matlab, a cell array is a data structure similar to an array, except that it can store any data type in each of its indices. Construct a cell array the same way you'd intend to construct a normal array, but use curly braces ({ }) instead of square braces ([ ]). For example, if I wanted to store the strings 'Matlab', 'is', and 'fun' as a cell array in a variable named cellarr, I'd use the following code:

```
cellarr = {'Matlab', 'is', 'fun'}
```

The official documentation for cell arrays can be found at https://www.mathworks.com/help/matlab/ref/cell.html

3 Files

For this lab, you will be provided the following files. Please refer to the documentation given in the comments heading each Matlab file for more detailed specification.

3.1 getData.m

A function that extracts the data from a selected file and returns it in the form of a matrix that is usable by the gradientDescent() and costFuncVec() functions.
Also can return a conditional subset of the data using a hard coded boolean filter.

- Inputs:
  - variables: a cell array of $k+1$ variable names to use. The first k are inputs indexed by their corresponding coefficients, and the $k+1^{st}$ entry is the output variable
  - filename: name of the file whose data to retrieve (should be an xlsx or csv)
- Outputs:
  - data: a matrix storing data on the variables indicated in the outputs. Records information including the order of the inputs by their coefficients and their observation indices.

3.2 gradientDescent.m

A function that takes in a set of data and runs a gradient descent algorithm on it, returning the final set of coefficients and intercept for a linear predictor function. The algorithm stops on a set number of iterations. The iterations and learning rate parameter (alpha) can be edited in the body of the function.

- Inputs:
  - data: a matrix storing data in the format returned by the getData() function.
- Outputs:
  - beta: a 1-d array containing the intercept and coefficients for a linear predictor function indexed in order.

3.3 costFuncVec.m

A function that calculates and returns the value of the cost function given data and a set of coefficients and intercept for a linear predictor function.

- Inputs:
  - data: a matrix storing data in the format returned by the getData() function.
  - beta: a 1-d array containing the intercept and coefficients for a linear predictor function indexed in order.
- Outputs:
  - cost: the value of the cost function given data and beta.

3.4 htv.csv

A dataset observing 1230 individuals' hourly wages and related variables in the United States.

Originally published by Heckman, Tobias, and Vytlacil (2003), "Simple Estimators for Treatment Parameters in a Latent-Variable Framework," Review of Economics and Statistics 85, 748-755. Provided for use with "Introductory Econometrics: A Modern Approach" by Jeffrey M. Wooldridge.

Variables:

- wage: hourly wage in USD, 1991
- abil: ability measure, non-standardized
- educ: highest grade completed by 1991
- ne: =1 if in northeast, otherwise =0 1991
- nc: =1 if in north-central, otherwise =0 1991
- west: =1 if in west, otherwise =0 1991
- south: =1 if in south, otherwise =0 1991
- exper: years of work experience
- motheduc: highest grade, mother
- fatheduc: highest grade, father
- brkhme14: =1 if broken home, otherwise =0 age 14
- sibs: number of siblings
- urban: =1 if in urban area, otherwise =0 1991
- tuit18: college tuition in USD, age 18

The full documentation can be viewed at http://fmwww.bc.edu/ec-p/data/wooldridge/htv.des

4 Exercise

You may work with one partner to complete this exercise.

Begin by running a gradient descent to compute a linear predictor function for `wage` with inputs `abil, exper,` and `educ` as in the demonstration. Save the 1-dimensional array of parameters you calculate in a variable. Then, run gradient descent again with one of the following changes:

- Remove one variable (other than `educ`)
- Add a new variable
- Impose a condition (other than `urban`$= 1$)

On CMS, submit a typed response of up to one paragraph explaining why you think any changes in your linear predictor's coefficients occur using the concepts of omitted variable bias or biased data. As long as you identify your partner's name and netID in your response, you may submit the same response as your partner.