LS 123 Data, Prediction, and Law

Meeting 12

**Pretrial Release Decisions and Prediction, Model Selection**

1. **Kleinberg et al on the pretrial release decision**
   1. this was something with real world implications
      1. cash bail is a problem, since it is patently unfair; it favors the well off in obtaining pretrial release
      2. pretrial detention can mean that you are a risk to society, or that you are risk of not appearing
      3. but it can also serve to coerce (especially indigent) defendants into a plea bargain that is disadvantageous
   2. here is a good example of causation versus prediction too—the paper is interested in prediction, and the relative quality of predictions
      1. since we know pretty well what causes bail decisions, or at least we think we know, since they are designed to make sure the accused appears and/or is not a risk to the community
      2. here we want to know if this is a good place for algorithmic decision making
         1. reasons why it might be? compressed time frame for consideration, judge has to deal with lots of “noise” (especially since much of the signal is in documentary form)
         2. we already know that the system is flawed
         3. maybe judges are not so good at it; we already know that they employ (sometimes by law) instruments to make postconviction decisions
   3. note the economics frame for the problem: we can make a tradeoff here between detaining people in jail and crime reduction
   4. note also that the authors take multiple approaches to the prediction problem
      1. logistic regression estimation, which is still better than judges’ predictions
      2. machine learning algorithm: gradient boosted tree
   5. what did they do?
      1. train the ML model on NYC data
      2. problem of labeled data—you can only find failure to appear or rearrest for people who had initial been released, not those held in jail
         1. they used their risk estimation function to impute outcomes to defendants who were not released, using a separate set of cases to do the imputation
         2. otherwise they could only look at the effect on criminal rearrest for jailing additional (higher risk) defendants
      3. did standard testing and training for machine learning algorithm
         1. divided the data into training and test sets
         2. divided the training set into five parts (crossvalidation) to tune their loss function (the penalty for misprediction)—they don’t want their algorithm to overfit for the data in the sample
         3. save some data for later (post-review test of results)
      4. they then test their algorithm against the State Court Processing statistics compiled by DOJ (and available thru the interuniversity consortium for political and social research) and find similar results
   6. so judges are not great predictors of FTA or of rearrest, and could benefit from the information produced by an algorithm that includes the predictors that are already available
   7. implications: this could help solve the problem of reducing or eliminating cash bail for most criminal defendants; it also would provide some guidance for judicial discretion in pretrial release
   8. problems? knowing exactly what the algorithm is doing, which might invite legal challenges to decisions on release when the defendant does something violent, for example, or might invite challenges when someone is held over for trial and then claims they made a plea without adequate counsel because they were being held, perhaps
2. **Lab 13 Model Selection (do this at about 1:30)**
   1. Choosing a model with least error on test set (conceived more broadly, on new data)
   2. Measuring how well a model fits the hypothetical underlying data that your training data represents—how well it will fit out of sample data generated by the same process (so the test set, or perhaps new data that comes in)
   3. Can do this with Akaike Information Criterion or Bayes Information Criterion, which estimate how much information is lost by the model relative to the process that generated the underlying data (so, measures we want to minimize)
   4. Alternative way of estimating how well the model will do with out of sample data is to just fit it on one piece of the data and test it on a chunk you have held out, and do this repeatedly until you can come up with an average of how much error there is in the estimate—we will do this with cross validation (using all but one chunk of data to fit a model and then testing the error in estimating the held-out chunk: see Aniket’s illustration in diagram on 12\_Model\_Selection)
   5. Leave One Out: computationally expensive; purpose is to get a less biased estimate of error for out-of-sample data—this might make sense when you have a smaller training set, for instance
   6. You should ask what the scoring method is when you do the cross-validation score—it is different for classifiers than it is for continuous values (which is what we are estimating here for ridge, lasso, and OLS); here it is the coefficient of determination, aka R^2, rather than some other measure like rmse