STATISTICS 452/652: Statistical Learning and Prediction

September 15, 2020

Lecture 6: Variable Selection: LASSO

(Reading: ISLR 6.2)

1 Goals of lecture

- We have seen that variable selection is a good thing
- It turns out that estimating parameters by LS following variable selection can create biased estimates!
- It can be good to reduce their magnitudes (shrink them back toward 0)
- Oh, by the way, this also reduces variance of coefficient estimates and the predicted values as well.

2 Problems with Least Squares

- Least squares estimates are almost universally used in regression
  - They are easy to compute in simple models and not very hard in complex ones.
  - The theory is easy to work out for linear regression and related methods
  - Inference is relatively easy
  - They are unbiased when the model is correct
    - Across all possible samples of size n, the average parameter estimate equals the true one
  - They have the smallest variance possible among all *unbiased, "linear"* estimates
  - They transfer these properties to predicted values
- But there are many places where LS is not really so good

- These great properties only hold when $f(X) = g(\mathbb{X})$. But how often does that happen?
  - They do NOT hold on the estimates if variable selection is done first
  - They do not hold if the model is mis-specified

Example: Least Squares before and after variable selection (L6 - Selection Bias and Shrinkage.R) This example is another simulation. We generate data from a structure where $Y = \beta_0 + \beta_1 X_1 + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. However, we assume that we didn't know this, and instead measure $p = 10$ variables, so there are 9 variables, $X_2, \ldots, X_{10}$ that are completely useless and just add confusion to the analysis. We simulate $n = 50$ observations from this structure. We fit the full linear regression model using LS, and then run the hybrid stepwise algorithm with BIC as implemented in `step()`. We then repeat this process 100 times to allow us to observe the properties of the analysis methods. See the program for full details.

Parameter Estimates I set the true parameter estimates to be $\beta_1 = 1$ and $\beta_0 = 0$, and of course $\beta_2 = \beta_3 = \ldots, \beta_{10} = 0$. With these settings, I can control the strength of the regression relationship by varying σ to make data more variable or less variable around the slope of 1. When I use $\sigma = 3$ I get a regression relationship that is not very strong (population $R^2 = 0.1$). See Figure 1 for an example of the data sets we are using with these settings.

I then used both least squares and the hybrid stepwise on each of the 100 samples of 50 observations, got out the parameter estimates for each method for all 11 parameters $(\beta_0, \beta_1, \ldots, \beta_{10})$. The summary of 100 estimates for $\hat{\beta}_1$ and $\hat{\beta}_2$ are shown in Figure 2. (The results for $\hat{\beta}_3$ through $\hat{\beta}_{10}$ are essentially the same as for $\hat{\beta}_2$.)

Notice that

- LS gives unbiased estimates of both parameters that have some variability.
- Stepwise estimate of β_1 has an odd bimodal distribution (it is not really displayed well by the boxplot)
  - In this case, it correctly includes X_1 in the model 68 times, and fails to include it 32 times
  - When X_1 is included, the average $\hat{\beta}_1$ is 1.34, overestimating $\beta_1 = 1$ by quite a bit
  - Combined with the 32 times we got $\hat{\beta}_1 = 0$, the overall average is below 1.
- Stepwise usually correctly estimates $\hat{\beta}_2 = 0$ (94 times).
  - The overall average is pretty close to 0, but with much less variability than for LS.

Figure 1: Example data set from simulation. True relationship with X_1 (labelled as "X" in plot) is in red. Points from one sample of $n = 50$ are in blue.

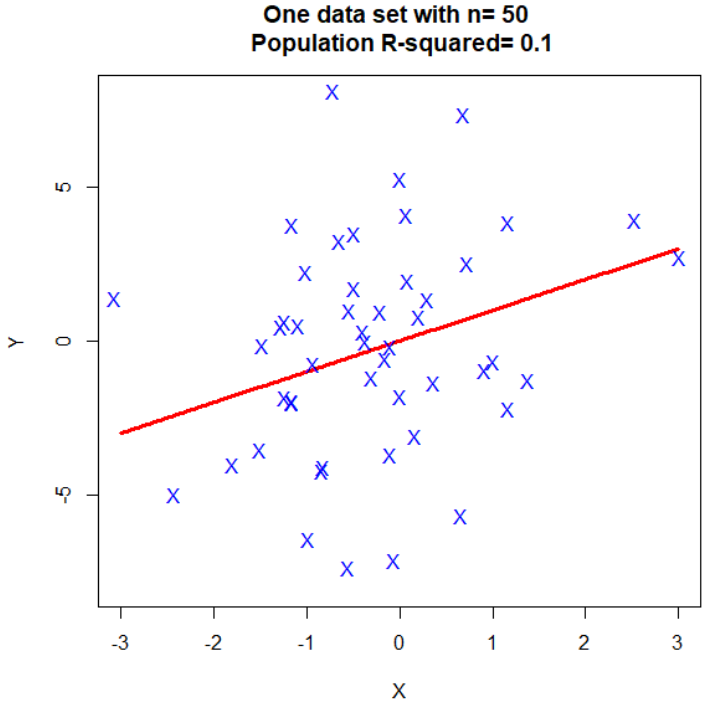


Figure 2: Parameter estimates from Least Squares on the full model and from hybrid stepwise selection, with true parameter values as red lines and sample averages as blue dot. Left panel: estimates $\hat{\beta}_1$ for $\beta_1 = 1$; right panel: estimates for $\hat{\beta}_2$ for $\beta_2 = 0$.

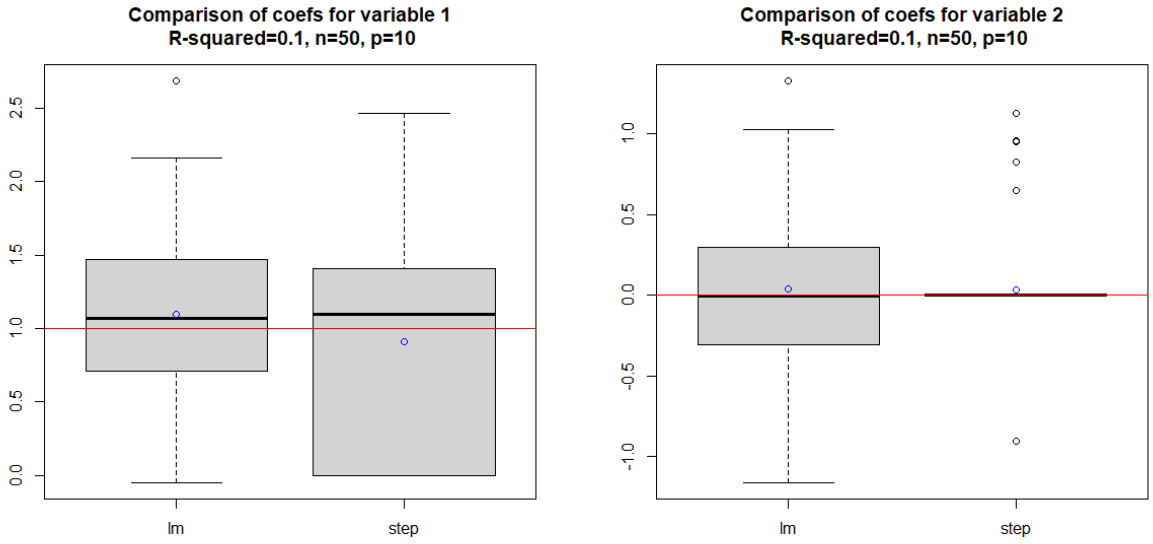
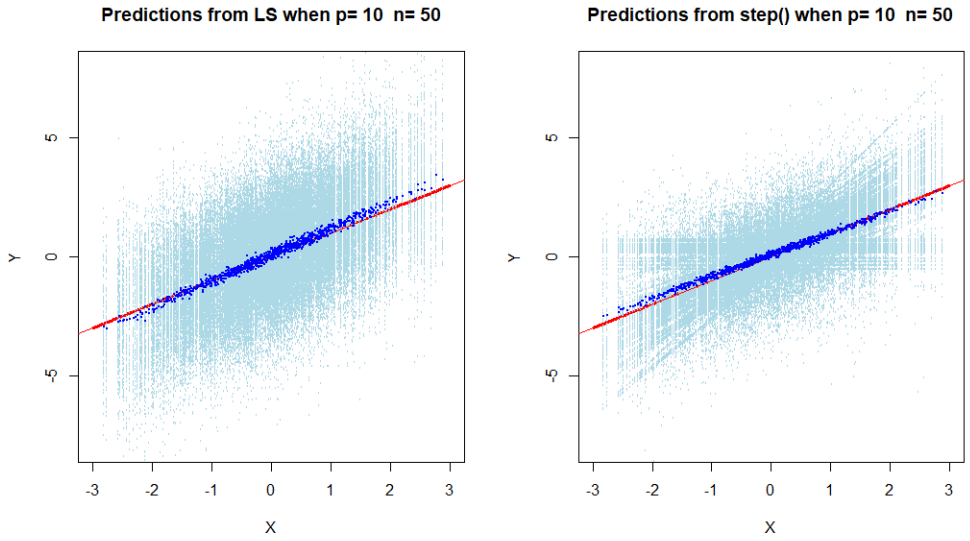


Figure 3: Predicted values from 100 simulated regressions estimates using Least Squares (L) and hybrid stepwise (R). Red line is true model, dark blue are average predictions across 100 data sets.



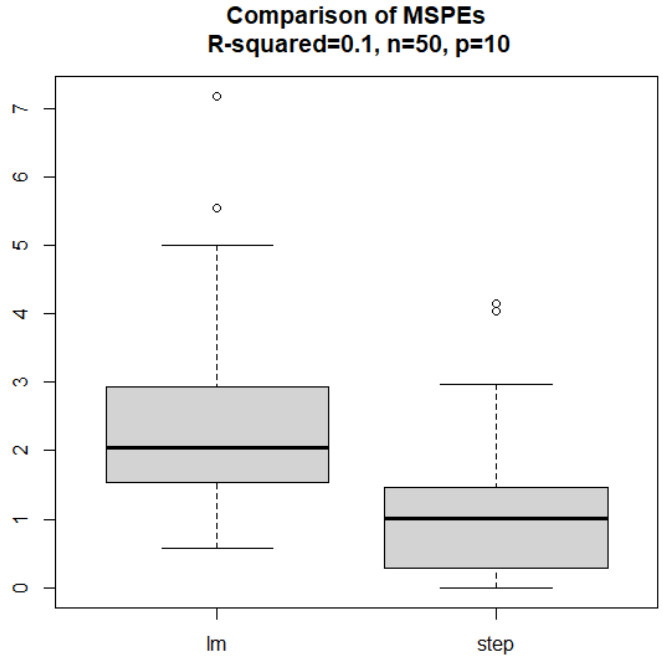
Prediction This hints at what happens when the two methods are compared for their predicted values. The left panel of Figure 3 shows the predicted values $\hat{Y}$ from each of the 100 models, plotted against X_1 (labelled as "X"). Notice that that there a lot of variability in the prediction lines from the full LS model when plotted against the only important variable. Some of that variability is due to different estimates of $\hat{\beta}_1$ across the samples, but a lot of it comes from the useless variability caused by adding $\hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 + \ldots + \hat{\beta}_{10} X_{10}$ to the predictions from $\hat{\beta}_1 X_1$. While the average prediction hovers pretty close to the true structure, the added variability is killer.

Compare this to the predicted valued from stepwise in the right panel of Figure 3 shows the benefit of variable selection. We see several interesting things:

- Sometimes, the stepwise procedure correctly estimates ALL useless parameters to be 0.
  - When this happens, the plot of $\hat{Y}$ vs.~X_1 is just a straight line
  - We can clearly see groups of cases where $\hat{\beta}_1 = 0$ and $\hat{\beta}_1 \neq 0$
- In general, the predicted values have a lot less variability, despite the fact that some of them are clearly biased by estimating β_1 wrong.
- The average estimate is actually fairly close to the truth, again despite the biased examples

We can summarize Figure 3 by computing the MSPE for every estimated regression from both models, although actually we are comparing the predictions to the true means rather than a new test set of Y's. This plot is in Figure 4. It is now obvious that variable selection can have a positive impact on the overall performance of regression models.

Figure 4: MSPEs for full model least squares and .



While this example shows the value of variable selection, it also points to some problems with stepwise.

- Some of the worst predictions from stepwise in Figure 3 are from cases *where it actually found the right model!*
  - When X_1 was included in the model, its parameter was usually overestimated
  - Generally, variables are more likely to be included when δ errors in data help to make them look important
    - When slope magnitude appears to be too close to 0, it gets set to 0
  - LS follows by chasing the same errors, overestimating slope magnitude.
- If the algorithm recognized this problem, it could shrink the parameter estimates whenever variables are selected

3 Ridge Regression, a "shrunken least squares"

How do we shrink parameter estimates?

- "Shrinkage" means setting parameter value somewhere between 0 and LS estimate
  - It's a way to reduce model complexity!

    - Like having a variable be "partway" in the model
    - Reacting less to individual δ errors.

- Main approach to achieving it is by penalizing the LS criterion
  - Kind of like how information criteria "penalize" sMSE for taking on too much model complexity

3.1 Ridge Regression Coefficients

- The usual least squares criterion finds parameter estimates that minimize

$$Q_{LS} = \sum_{i=1}^{n}(y_i - [\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}])^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

- Ridge Regression minimizes

$$Q_{R,\lambda} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = Q_{LS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

  - $\lambda\sum_{j=1}^{p}\beta_j^2$ is a shrinkage penalty that controls how far between 0 and the LS estimates
  - $\lambda \geq 0$ is a tuning parameter that needs to be chosen by the analyst somehow (more on that later)
  - $\sum_{j=1}^{p}\beta_j^2$ is a measure of the aggregated sizes of the parameters, called the "L_2 norm".
  - The parameter estimates that minimize this are denoted by $\hat{\beta}_j^{R,\lambda}$, $j = 1, \ldots, p$

- Minimizing $Q_{R,\lambda}$ is different from minimizing Q_{LS}
  - Imagine β_j, $j = 1, \ldots, p$ all starting at 0
    - Then $Q_{LS} = Q_{R,\lambda} = \sum_{i=1}^{n}(y_i - \beta_0)^2$, since $\sum_{j=1}^{p}\beta_j^2 = 0$
    - Call this Q_{LS}^0 and $Q_{R,\lambda}^0$
  - Now slowly move all parameters from 0 toward their LS estimates $\hat{\beta}_j$, $j = 1, \ldots, p$
    - Q_{LS} slowly decreases from Q_{LS}^0 toward its minimized value, say Q_{LS}^{min}
    - The aggregate size $\sum_{j=1}^{p}\beta_j^2$ slowly increases from 0 toward the full size of the LS estimates, $\sum_{j=1}^{p}\hat{\beta}_j^2$
  - Finally consider the effect of λ on $Q_{R,\lambda}$
    - If $\lambda = 0$ then
      - $Q_{R,0} = Q_{LS}$ since the second term is always 0
      - The optimal solution is the LS estimates, $\hat{\beta}_j^{R,0} = \hat{\beta}_j$, $j = 1, \ldots, p$
      - Optimal size is $\sum_{j=1}^{p}\hat{\beta}_j^2$

Figure 5: Example Ridge coefficient paths from example in ISLR. Left: Optimal estimate as λ increases from 0 toward ∞. Right: Optimal estimate as the aggregate length changes as a fraction of the LS optimal, from 0 to 1.

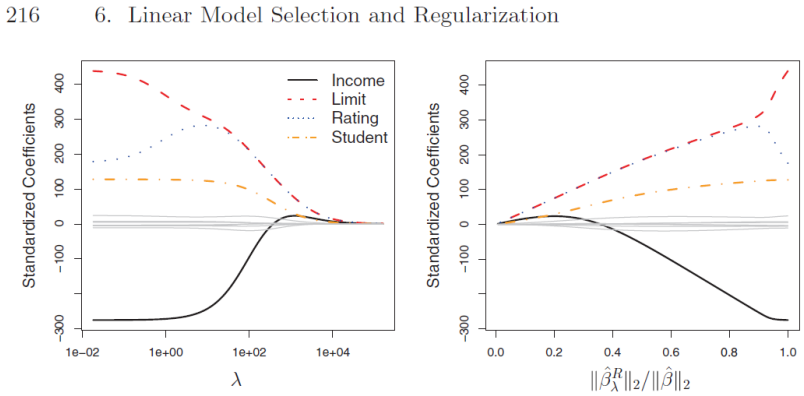


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

  - If $\lambda = \infty$ then
    - Increasing the aggregate size $\sum_{j=1}^p \beta_j^2$ by *any* amount blows $Q_{R,\infty}$ to ∞
    - So the optimal solution is $\hat{\beta}_j^{R,\infty} = 0$, $j = 1, \ldots, p$
    - Optimal size is 0
  - For any $0 < \lambda < \infty$, then
    - $Q_{R,\lambda} > Q_{LS}$
    - It reaches its minimum value at a smaller aggregate size than $\sum_{j=1}^p \hat{\beta}_j^2$
  - Thus, the parameter estimates are shrunk compared to the LS estimates
    - Larger λ results in smaller optimal estimates that have more shrinkage

- In reality the values of $\hat{\beta}_j^{R,\lambda}$, $j = 1, \ldots, p$ that are optimal for a given λ are not obtained by simply multiplying the LS estimates by a number between 0 and 1
  - Individual estimates might be larger or smaller than LS estimates by varying amounts
  - Aggregate size is smaller
- For each λ there is a different set of optimal parameter estimates
  - A COEFFICIENT PATH is a plot that shows how these optimal parameter estimates change λ changes
  - Equivalently, as their aggregate size is changed from 0 ($\lambda = \infty$) to the full LS estimates ($\lambda = 0$)
  - Figure 5 shows these plots for the example used in ISLR

3.2 Choosing λ

It's all about the bias-variance tradeoff!

- We need to choose a value of λ
- λ indirectly controls the complexity of the model
  - Larger λ shrinks parameter estimates more
    - They chase errors less
    - Model is less flexible, less complex
  - Small λ does opposite
- Optimal λ balances between
  - increasing bias from shrinking parameters
  - decreasing variance by chasing errors less
- Can be chosen by CV
  - Luckily, n-fold (leave-one-out) CV error is approximated by a relatively simple formula
  - Called "generalized cross-validation", GCV.
  - So can compute GCV for a range of λ values and pick the one whose estimates minimize GCV

3.3 How/Why does ridge work?

- Works best when problem is susceptible to overfitting by LS (have potentially high variance)
  - Multicollinearity
  - Large p
  - Small n
  - High variability in δ errors
- In these problems, LSEs can be highly influenced by particular δ errors in sample
  - Shrunken estimates move less toward errors
- But: Ridge merely shrinks parameter estimates, doesn't select variables
  - Along the coefficient path, parameter estimates shrink *toward* 0, but don't ever get there exactly
- Would be nice to have something that selects variables AND shrinks coefficient estimates.

4 The LASSO

- One of the biggest developments in regression modeling in recent time.
- In 1996, Rob Tibshirani (one of ISLR's authors) updated ridge regression so that it could not only shrink the parameter estimates, but could also serve as a variable selection method.
- The "Least Absolute Selection and Shrinkage Operator" (LASSO) uses a criterion very much like the ridge criterion $Q_{R,\lambda}$,
  - Instead of an L_2-norm to aggregate the size of the parameter estimates in the penalty, it uses an "L_1-norm":

$$Q_{LASSO,\lambda} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

  - λ plays same role as before
    - Larger λ shrinks parameter estimates more
      - More bias, less variance
    - Small λ does opposite
- The seemingly minor alteration between $Q_{LASSO,\lambda}$ and $Q_{R,\lambda}$ makes all the difference.
  - The geometry behind this penalty in p-dimensional β-space is such that optimal solutions to tend to fall on the axes where some of the $\beta_j = 0$. (See ISLR Figure 6.7, P622).
  - Optimal parameter estimates for given λ are $\hat{\beta}_j^{L,\lambda}$, $j = 1, \ldots, p$, and often some are set to 0
  - Coefficient path keeps estimates at *exactly* 0 until the variable can explain enough squared error to overcome the penalty of λ on its magnitude.
  - The end result is that as λ moves from ∞ toward 0, only one parameter estimate is non-zero for a while.
    - Then one more becomes non-zero and these two both change until another one can become non-zero, and so on.
    - This is depicted for the same example from ISLR Figure (6) .
  - Parameter estimates that are not set to 0 usually have their coefficients shrunk.
    - As with Ridge regression, there is no guarantee that all parameter estimates are smaller than their OLS estimates at all points on the coefficient path
- λ is usually chosen by CV, because GCV formula no longer works
  - As usual, there can be lots of variability due to random partitioning into folds
  - *When the sample is large enough,* it can be advantageous to use the "1SE rule" for selecting λ

Figure 6: Example LASSO coefficient paths from example in ISLR. Left: Optimal estimate as λ increases from 0 toward ∞. Right: Optimal estimate as the aggregate length changes as a fraction of the LS optimal, from 0 to 1

220 6. Linear Model Selection and Regularization

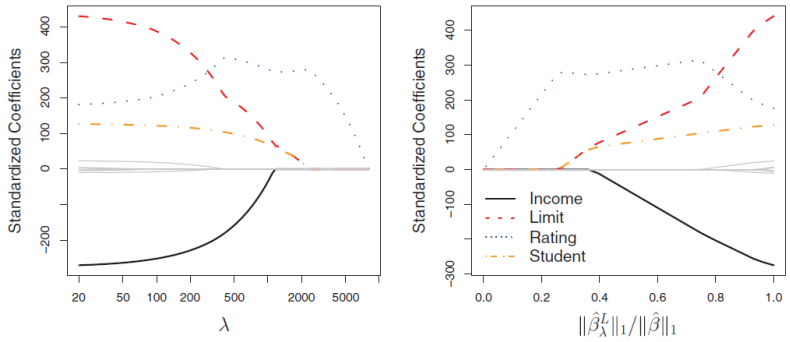


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

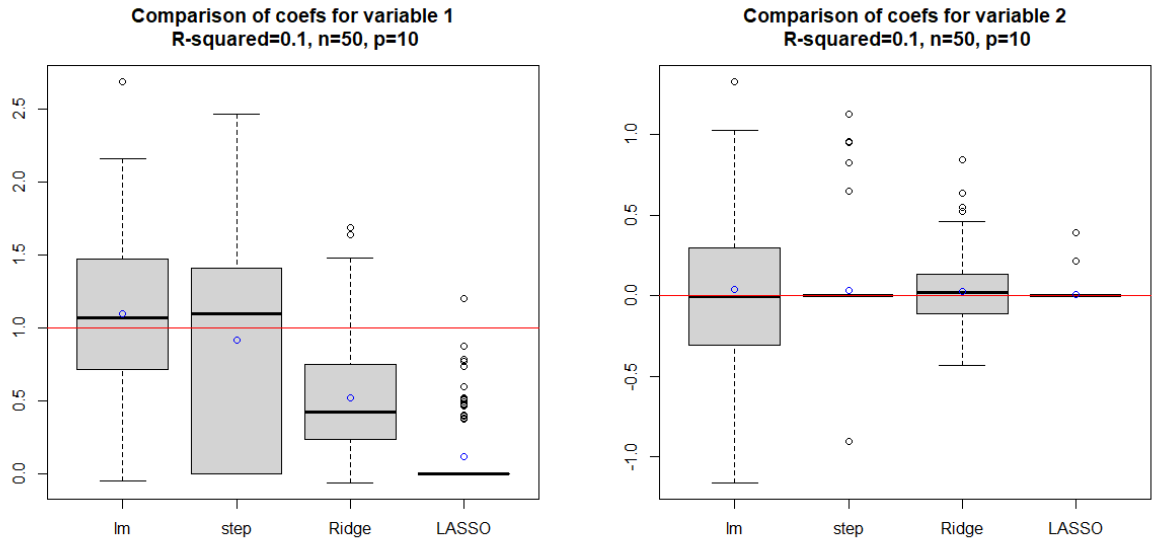
  - Rather than choosing λ that minimizes CV, use the one that produces the least complex model whose CV is within 1 SE of the minimum.
  - See example below.

- LASSO has shown pretty good performance in a variety of prediction problems, and is often recommended for general use
  - Both as a variable selection tool and as a prediction tool
  - Prediction quality is partly because parameter estimates are optimized with λ to produce optimal CV error
    - If LSEs would predict better, would choose λ near 0.

Example: Least Squares before and after variable selection (L6 - Selection Bias and Shrinkage.R) We return to the simulation where we generate data from a structure where $Y = \beta_0 + \beta_1 X_1 + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$, but there are 9 additional useless variables. We now apply ridge regression using GCV to estimate the optimal λ, and LASSO using CV with the 1SE rule for its λ. We add the results from these two methods to the plots comparing the parameter estimates and predicted values.

Parameter Estimates The summary of 100 estimates for β_1 and β_2 are shown in Figure 7. Both ridge and LS do no variable selection. The shrinkage in ridge compared to LS there is clearly less variance, but also more bias. Just looking at this comparison, it's hard to gauge which one works better in the sense of coming closest to the red lines more often. Comparing the two variable-selection methods, LASSO and stepwise, we see that LASSO not only sets parameter estimates to 0 more often (79/100 data sets for β_1, 98/100 for β_2), it also shrinks the estimates when they are not zero. I should point out that this example is one in which the very weak relationship is hard to model, so LASSO decides that

Figure 7: Parameter estimates from LS, Stepwise, Ridge, and LASSO, with true parameter values as red lines and sample averages as blue dot. Left panel: estimates $\hat{\beta}_1$ for $\beta_1 = 1$; right panel: estimates for $\hat{\beta}_2$ for $\beta_2 = 0$.



it's actually better not to bother trying. (The results for β_3 through β_{10} are essentially the same as for β_2.)

Predictions LASSO's extreme insistence on not modeling the slope for X_1 seems crazy, but there is a method to the madness. The left panel of Figure 8 shows the predicted values from each of the 100 models for ridge, and the right panel shows them for LASSO. Compared to Figure (3), the predicted values for ridge have way less variability than those from LS, and lie generally much closer to the true structure. But LASSO has found the winning strategy for these data by deciding that it is too hard to estimate the slope with good precision, and hence ignoring it. The predictions look strange and unsatisfying, but from a pure optimality perspective, they are not as far from the truth as it seems.

The MSPEs across a large test set for each of the 100 models from each method are shown in Figure 9. Comparing ridge to the LS results, it is obvious that shrinkage can have a positive impact on the overall performance of regression models. LASSO has the most compact (consistent) performance from one data set to the next, although the average MSPEs for stepwise, ridge, and LASSO are pretty close.

Example: Ridge and LASSO on the Prostate Data (L6 - Ridge LASSO Prostate.R) The program for this example shows how to use the `lm.ridge()` function from the `MASS` package for ridge regression. Unfortunately, there is no `predict()` function for `lm.ridge()`, so one has to compute predicted values manually using estimated coefficients and the X data. LASSO is done with `glmnet()` and `cv.glmnet()` from the `glmnet` package. The latter function computes 5-fold CV error for a sequence of about 68 different values of λ. It

Figure 8: Predicted values from 100 simulated regressions estimates using Least Squares (L) and hybrid stepwise (R). Red line is true model, dark blue are average predictions across 100 data sets.

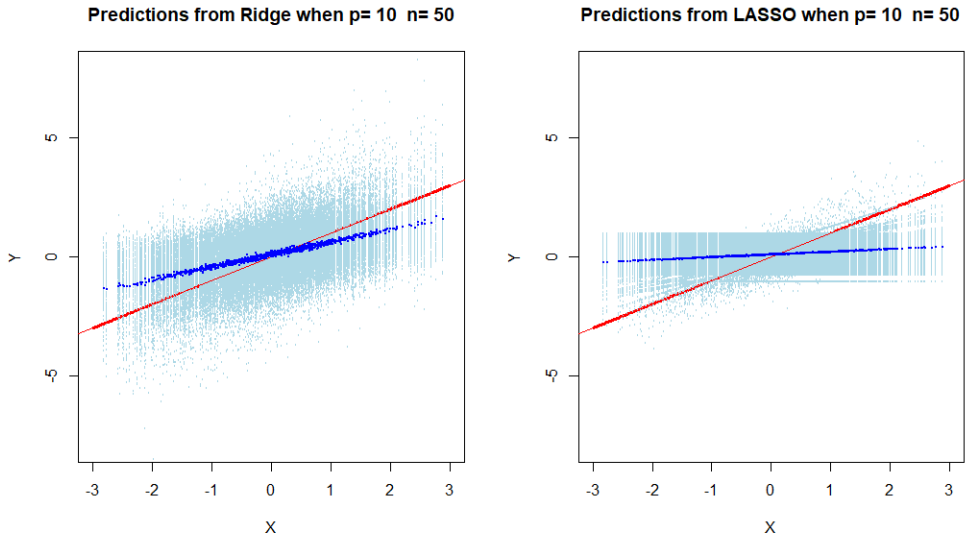
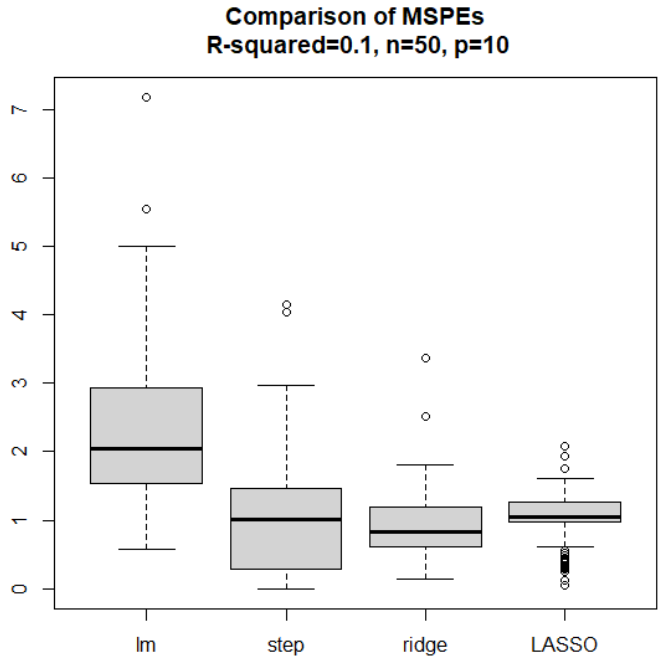


Figure 9: MSPEs for full model least squares and .



allows you to select the λ that has the minimum CV error or use the 1SE rule (the default in `predict()`).

I once again split the data in half and run the methods on each half to predict the other. Details are in the program, and you can run the program to see the output. One remarkable thing: LASSO-1SE actually selects the same two variable model for both halves! They have different parameter estimates, but both halves use variables `lcavol` and `lcp`. Also, when I compute the MSPE using LS, ridge, LASSO-min and LASSO-1SE, I get the following:

Method	Fit to Set 1	Fit to Set 2
LS	0.63	0.51
Ridge	0.66	0.47
LASSO-Min	0.41	0.37
LASSO-1SE	0.58	0.50

In this particular data set, it seems that the min-CV estimate of λ is rather better than the 1SE. This happens often when the sample is small and/or the number of parameters is large relative to the sample size. Here, our two sets have 45 and 52 observations, which is not very large, particularly for a problem with 8 variables.

---

5 What to take away from this

- Variability is a serious issue when it comes to doing prediction
- Often, using biased regression parameter estimates ("shrinkage") can yield predictions that have smaller average error than using unbiased estimates
- Variable selection does the same thing in a different way, reducing variance by eliminating needless parameters from a model
- Combining these two things, which LASSO does, can be a winning strategy for prediction in linear models.
- BUT: NONE OF THIS HELPS US TO OVERCOME BIAS IF THE TRUE STRUCTURE IS NOT LINEAR!
  - Stay tuned for more methods to deal with this!

6 Exercises

Concepts

The examples from this lecture showed results from *one* possible situation where the various methods of estimating regression parameters could be compared. Bias and variance of the different methods may change according to certain settings, like n, p, and σ^2. In this exercise we explore how these settings affect performance by rerunning a homework version of **L6 - Selection Bias and Shrinkage.R**, changing the settings each time. For each question below, **report the MSPE plot, say which method(s) seems to be best in that setting**, and notice (but do not report) how bias and variance interact to create these different results.

1. $n = 25, p = 10, \sigma = 3$
2. $n = 500, p = 10, \sigma = 3$
3. $n = 50, p = 2, \sigma = 3$
4. $n = 50, p = 30, \sigma = 3$
5. $n = 50, p = 10, \sigma = 0.5$
6. $n = 500, p = 100, \sigma = 3$
7. $n = 500, p = 2, \sigma = 0.5$

Application

Refer to the Air Quality data described previously, and the analyses we have done with `Ozone` as the response variable, and the five explanatory variables (including the two engineered features).

1. Use ridge regression on the data:
   (a) Using GCV, identify the optimal λ from the sequence 0 to 100 by 0.05. **Report the optimal value for λ.**
   (b) **Report the parameter estimates for the optimal model, and compare them to those from least squares. Are the ridge estimates all smaller.**
   (c) Compute $\sum_{i=1}^{p}(\hat{\beta}_j^R)^2$ and the same calculation using the least squares estimates. **Report both numbers as well as their ratio for ridge/LS to see how much shrinkage has taken place.**
2. Use LASSO on the data
   (a) Using default CV in `cv.glmnet`, identify the optimal λ, λ_{min}, and the "1SE" λ, λ_{1SE}. **Report both numbers.**

(b) **For both selections of λ, report the parameter estimates for the optimal model. Comment on how the two sets of estimates differ from each other.**

(c) Compare the variables selected by both versions of LASSO to those from the hybrid stepwise from Lecture 5. **What differences are there?**

3. Use 10-fold CV to estimate the MSPE for ridge, LASSO-min, and LASSO-1SE. That is,

   (a) Set the seed to 2928893 before running the `sample.int()` function.

   (b) Create 10 folds

   (c) Run the three analyses on each training set

      i. Find the best versions of each for that training set

      ii. Use those best versions to compute the prediction error on the validation set

   (d) **Report the separate MSPEs from each fold, $MSPE_v$, $v = 1, \ldots, 10$ and the MSPE for the full data.**

   (e) **Make a boxplot of the 10 CV error estimates showing the boxes for least squares, hybrid stepwise, ridge, and LASSO. Comment on any apparent differences in how the methods seem to perform.**

   (f) **Repeat this using relative MSPE.**