LASSO reading 2018

MEMORANDUM

To: BDA students

From: Prof. Roger Bohn

Subject: Readings on LASSO + overfitting

Date: May 15, 2018. Version 1.2

Part I: Readings and Discussion

Data mining methods are used to handle situations that traditional econometrics must run away from. One example is a situation with *more variables than observations*. For example, genetic data on 2000 sites in the genome for each of 1000 patients. In traditional statistical theory, such systems are overdetermined and therefore impossible to solve.

This and similar situations lead to overfitting. In overfitting, we have fit our model using some independent (right hand side) variables that actually have no effect. Of course we can't tell which variables are the irrelevant ones. By testing our model using new data (the validation data set) that was not used during training, we will discover whether serious overfitting occurred. (Compare fits from the training and validation sets. If the validation fit is worse, you have overfit.) But if it did occur, we don't know where.

LASSO (least absolute shrinkage and selection operator) is a very clever idea for choosing only a subset of variables to use. Empirically, it tends to work well to select from among many variables, and thereby to avoid overfitting.

Read: Gareth James et al, ISLR *An Introduction to Statistical Learning with Applications in R*. Chapter 6 introduction, and Chapter 6.2 and 6.4. Look at Lab 2, section 6.6. Chapter 6 also discusses 2 other variable reduction methods: subset selection, and dimension reduction. Skim their descriptions - you should know they are also available.

Study Figure 6.6, page 220.[1] It shows what happens as the λ parameter becomes more and more severed. Initially, the LASSO solution is the same as the regular linear model. As λ rises, it forces down the β coefficients on variables, until some of them are zeroed out. If λ gets huge, all the βs are forced to 0.

High Dimensions: The main benefits of LASSO come when the number of dimensions

[1] Reproduced below.

(degrees of freedom, number of variables) is large relative to the amount of data. So study Section 6.4. Figure 6.23 shows some of the implications.

Ignore: You can ignore the discussions of Bayesian scaling, and of Ridge Regression.

Scaling

There are a few things to watch out for in LASSO. One is that the absolute value of the coefficients becomes important. For example, if distance is measured in km a coefficient might be 50, but if the same distance is measured in meters, the corresponding coefficient would be .05. LASSO would penalize the first case much more than the second; yet they are physically the same relationships.

For better or worse, the algorithm in the ISLR book automatically takes care of scaling the variables (unless you tell it not to). So you can pay little attention to this issue for now. But how the algorithm does the scaling will affect the results. For example it could scale by standard deviations, or by interquartile range of each variable. These will lead to different coefficient estimates. So scaling method becomes something else to tune.

LASSO is an example of *regularization*, which is the technique of adding a constraint to the usual objective function in optimization. So while OLS minimizes the sum of squared errors, LASSO minimizes:

```
(sum of squared errors) + an error function that measure model complexity
```

This strategy comes up all the time in optimization problems (such as nonlinear programming), but to non-specialists it seems bizarre. It works nicely in many contexts so you will want to use it.

LASSO versus Random Forests

Both methods help in situations with lots of variables. Both provide estimates of which variables are the most important (in engineering/business terms, not in statistical terms). Both have lots of tuning parameters you can play with to improve the results.

There are some differences in how you prepare the data sets for the two algorithms. That is, you generally should not take one data set and blindly throw it into both models. Instead, you will want to pre-process the data differently. LASSO assumes a linear model, so if you want any nonlinearity you must insert them yourself by creating nonlinear transformations of variables. RF can handle missing data, while LASSO (and any linear model) cannot except by work-arounds.

Diagrams

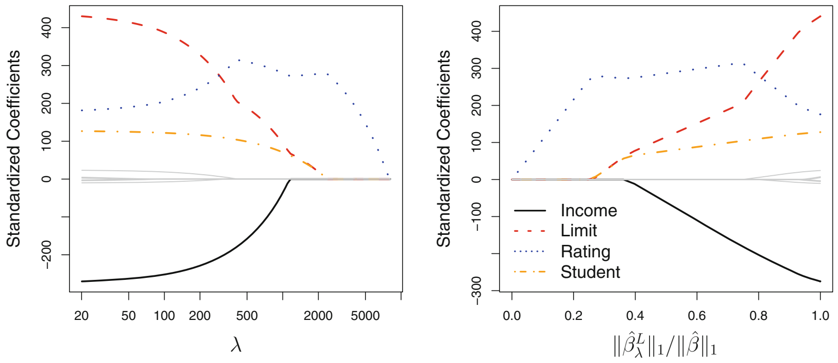

FIGURE 6.6. *The standardized lasso coefficients on the* Credit *data set are shown as a function of λ and $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.*

6.4 Considerations in High Dimensions 241

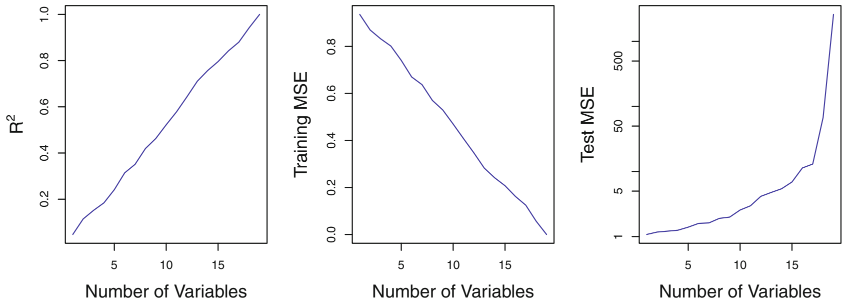

FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The test set MSE increases as more features are included.*

LASSO exercise

MEMORANDUM
To: BDA students
From: Prof. Roger Bohn

Subject: LASSO homework for week 7, May 16

Date: May 15, 2018. Draft version 1.4

Part 2: Exercise

Do section 6.6.2, Lab 2 of the ISLR supplemental textbook. Only do the Lasso, not Ridge regression. It is in Chapter 6, which you should download from Springerlink if you have not already done so. This lab analyzes the Hitters data, which is available from the course Google Drive, https://drive.google.com/open?id=1K54L2Y-AFPEZoUGofYFeVOPlOJ6CLvtU

You can follow the procedure in the lab in Section 6.6.2, page 255. You will have to skim earlier sections of Chapter 6, to see how to set up the LASSO calculations. Use 10-fold cross-validation to choose the best value of LAMBDA, as discussed in Section 6.2.3. Use cross-validation rather than setting aside a validation set, which is new.

Note how they calculate the error associated with the solution:

```
mean((lasso.pred-y.test)^2)
```

We have also discussed other useful error measures such as Mean Absolute Percentage Error (see main textbook chapter 5). Also calculate (using R) the MAPE and the Root Mean Squared Error. For example,

RMSE = MSE^.5

Plot the results using plot(). Plot() is the "simple" plotting system for R. Primarily you should use the ggplot2 system but plot is fine for quick results like this one.

When you write your memo (due Friday), be sure to discuss the managerial significance of the results, including the values of the coefficients. Address your memo to the agent of a player who is about to begin salary negotiation.

A. Run a LASSO analysis of the hitters data.

Use 10-fold cross-validation to tune your model. In this case, the only tuning parameter is λ. Discuss the results in a memo, addressed to the agent of a player who is about to negotiate his salary. (Agent = representative for negotiation and other purposes.) Use a different random seed, so that your results are slightly different than the textbook's.

B. Special Instructions: Do some R deconstruction and debugging of the code in the textbook.

- Attach an added page to your memo which contains your R code, in ready-to-run form. Someone else should be able to copy and paste the code into RStudio, and run it on their system.
- Briefly explain this R code:
    - lasso.coef=predict (out ,type =" coefficients" ,s=bestlam)[1:20 ,]
    - lasso.coef[lasso.coef !=0]
- The following segment of R from the ISLR book, pasted into your R console (bottom left window), may give an error message:
    - model.matrix (Salary~.,Hitters)[,-1] #gives an error. Why??
    - Debugging hint: try *typing* the same line instead of pasting it. Also look at the error message very carefully.
    - This is bizarre. Can you explain it?

Notes:

The Hitters file is of type .RDA, which you probably have not seen it before. When you read in the data you will have to use an appropriate read function. Once again, have your cheat sheets or other references handy.

You will have to do a modest amount of data cleaning. Read Section 6.5.1 for cleaning instructions. You can always do additional cleaning if you think it is useful.

Some of the details of glmnet() are discussed in section 6.6.1

Use a different random number seed than in the book. That way you will get slightly different results. If you eat to double-check, you can also do the calculations with the same seed.

You will use, for the first time, the function *model.matrix*, which converts variables into the proper form for quantitative modeling. For example, it converts categorical variables into dummy variables, and it cam also create quadratic and interaction terms. (Rattle took care of categorical variables automatically, but some R functions cannot handle them directly. It depends on how much work the author of each package wanted to do themselves.) Find documentation for model.matrix, and study the way it is used in Section 6.6.

"The model.matrix() function is particularly useful for creating x; not only does it

> produce a matrix corresponding to the 19 predictors but it also automatically transforms any qualitative variables into dummy variables. The latter property is important because glmnet() can only take numerical, quantitative inputs.”

Functions introduced

Write short definitions of each, and where to look up more explanation. Add them to your personal “cheat sheet.” You may need to add other functions.

model.matrix

is.na()

na.omit()

sum()

glmnet()

cv.glmnet()

predict()

mean()

.