Problem Set #4

Kevin McAlister

February 2nd, 2023

This is the fourth problem set for QTM 347. This homework will cover applied exercises related to predictor selection for linear regression models.

Please use the intro to RMarkdown posted in the Intro module and my .Rmd file as a guide for writing up your answers. You can use any language you want, but I think that a number of the computational problems are easier in R. Please post any questions about the content of this problem set or RMarkdown questions to the corresponding discussion board.

Your final deliverable should be two files: 1) a .Rmd/.ipynb file and 2) either a rendered HTML file or a PDF. Students can complete this assignment in groups of up to 3. Please identify your collaborators at the top of your document. All students should turn in a copy of the solutions, but your solutions can be identical to those of your collaborators.

This assignment is due by February 10th, 2023 at 11:59 PM EST.

Problem 1: A Cool Ridge and LASSO Identity (30 pts)

There is a relationship between the optimal solutions for the regression coefficients under no penalty (the OLS solution), the ridge penalty, and the LASSO penalty. The exact relationship cannot be derived for most cases, but we can gain some knowledge by assuming that the predictors are exactly orthogonal to one another. Since we can always rescale the variance of the features, we can further restrict this to feature sets that have **orthonormal** columns - $\boldsymbol{X'X} = \mathcal{I}_P$.

Recall that OLS, Ridge, and LASSO are all solutions to slightly modified loss functions given a vector of training outcomes, $\mathbf{y}$, and a $N \times P$ matrix of training features, $\mathbf{X}$:

$$\hat{\boldsymbol{\beta}}_O = \underset{\boldsymbol{\beta}^*}{\text{argmin}} \ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*)'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*)$$

$$\hat{\boldsymbol{\beta}}_R = \underset{\boldsymbol{\beta}^*}{\text{argmin}} \ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*)'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*) + \lambda \sum_{i=1}^{N} \beta_j^{*2}$$

$$\hat{\boldsymbol{\beta}}_L = \underset{\boldsymbol{\beta}^*}{\text{argmin}} \ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*)'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*) + \lambda \sum_{i=1}^{N} |\beta_j^*|$$

Assuming the feature matrix, $\boldsymbol{X}$, has orthonormal columns, show that:

$$\hat{\boldsymbol{\beta}}_O = \mathbf{X}'\mathbf{y}$$

$$\hat{\boldsymbol{\beta}}_R = \frac{\hat{\boldsymbol{\beta}}_O}{1+\lambda}$$

$$\hat{\boldsymbol{\beta}}_L = \text{sign}(\hat{\boldsymbol{\beta}}_O) \times \max\left(|\hat{\boldsymbol{\beta}}_O| - \frac{\lambda}{2}, 0\right)$$

where sign$(\cdot)$ returns the sign of the input and max$(\cdot)$ returns the maximum of the two arguments. This definition can also be written as:

$$\hat{\boldsymbol{\beta}}_{L,j} = \begin{cases} 0 & \text{if } \hat{\boldsymbol{\beta}}_{O,j} \leq \frac{\lambda}{2} \\ \hat{\boldsymbol{\beta}}_{O,j} - \frac{\lambda}{2} & \text{if } \hat{\boldsymbol{\beta}}_{O,j} > \frac{\lambda}{2} \end{cases}$$

Notes/Hints:

1. $\sum_{j=1}^{P} \beta_j^2$ can also be expressed as $\beta'\beta$.
2. Suppose we have $\beta' A \beta + \beta'(\lambda \mathcal{I})\beta$. We can condense this to $\beta'(A + \lambda\mathcal{I})\beta$.
3. Expand the OLS part of the loss function first and then substitute $\hat{\beta}_O = \mathbf{X}'\mathbf{y}$.
4. Let $\mathbf{q}$ and $\mathbf{w}$ both be P-vectors. $\mathbf{q}'\mathbf{w} = \sum_{j=1}^{P} q_j w_j$.
5. After your initial expansion of the LASSO loss function and substituting $\hat{\beta}_O = \mathbf{X}'\mathbf{y}$, you'll want to switch from vector notation to summations so you can include the LASSO penalty term inside the summation. Showing the identity holds for any specific element of the coefficient vector is a sufficient proof.
6. **Important**: In the orthogonal case, it is always the case that $\text{sign}(\hat{\boldsymbol{\beta}}_O) = \text{sign}(\hat{\boldsymbol{\beta}}_L)$. You can substitute these two terms freely, but you'll have to reconcile this identity at the end of your proof - this is where the max$(\cdot)$ comes in.
7. These solutions can be found online and in various textbook resources. I think this is a great exercise for thinking through complex minimization problems, so don't spoil yourself unless you really find yourself stuck.

Problem 1 Solution

Problem 2: Applying Ridge and LASSO

Ridge and LASSO are great methods for parsing through data sets with lots of predictors to find:

1. An interpretable set of *important* predictors - which predictors are **signal** and which ones are just **noise**
2. The set of parameters that minimize expected prediction error (with all the caveats that we discussed in the previous lectures)

Where these methods really shine for purpose 1 (and purpose 2, by construction) is when the ratio of predictors to observations approaches 1. To see this and work through an example using pre-built software, let's try to build a model that predicts IMDB ratings for episodes of the Office (the U.S. Version). `office_train.csv` includes IMDB ratings (`imdb_rating`) for 115 episodes of the office and a number of predictors for each episode:

1. The season of the episode (1 - 9, which should be treated as an unordered categorical variable!)
2. The number of times main characters speak in the episode (`andy` through `jan`)
3. The director of the episode (`ken_kwapis` through `justin_spitzer`) already "dummied out" so that a 1 means the person directed the episode and a 0 means they did not

Let's use this data to build a predictive model for IMDB ratings and check our predictive accuracy on the heldout test set (`office_test.csv`).

For this problem, you can restrict your search to the set of standard linear models (e.g. no interactions, no basis expansions, etc.). If you would like to try to include more terms to improve the model, you are more than welcome to try!

An important step: Get rid of any features in both data sets that are observation specific, like episode number and episode name!

Part 1 (10 pts)

Start by limiting yourself to the standard OLS model.

Find the regression coefficients that minimize the training error under squared error loss and use this model to compute a LOOCV estimate of the expected prediction error using all features.

Which predictors appear to be important? Which ones don't? This can be difficult to tell from the OLS estimates!

Hint:

You can compute LOOCV for OLS with a formula!

$$LOOCV = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1 - H_{i,i}}\right)^2$$

where $H_{i,i}$ is the i'th diagonal element of the hat matrix. In R, you can get the diagonal of the hat matrix for a fitted linear regression model using `hatvalues(ols_mod)`. In Python, you can get the diagonal of the hat matrix for a linear regression fit with `statsmodels` by first computing the influence matrix and then getting the diagonal elements of the hat matrix `.get_influence().hat_matrix_diag`.

Part 1 Solution

Part 2 (10 pts)

Select a subset of "important" predictors and find the set of coefficients that minimizes MSE under squared error loss. Compute the LOOCV estimate of the expected prediction error for your smaller model. How does this LOOCV estimate compare to the LOOCV for the full model?

Note: You don't need to try to perform any subset selection in this step. Just use heuristic methods to determine which coefficients appear to be important for the model!

Part 2 Solution

Part 3 (20 pts.)

Now, consider ridge regression. Using a pre-built implementation of ridge regression, train the model using a large number of possible values for λ.

Using 10-fold cross validation, find a reasonable value of λ that should minimize the expected prediction error. You can choose the actual minimum or a slightly less complex model. Defend this choice.

Create a plot that demonstrates the regression coefficients for the ridge regression with your optimal choice of λ. Which predictors are important? Which ones are not? I recommend using a sideways bar plot - you can see an example construction here.

Notes:

1. To make your training set work with `glmnet`, you need to divide the training set into a $N \times P$ matrix of predictors and a N-vector of outcome values. The predictor matrix **must be a matrix (not a data frame)** to work with `glmnet`. You can always convert a data frame to a matrix using `as.matrix()` in R.
2. `season` is coded as a character string in the original data set. You need to create a series of dummy variables for this character string in order to have the appropriate feature matrix. Instructions for one-hot encoding the feature matrix can be found in the Regularized Regression explainer on Canvas.
3. To plot out the coefficients at your chosen value of λ, you need to extract the optimal coefficients. You can do this following the code examples in the explainer.
4. If you have a better plot idea than a sideways bar plot, go for it!

Part 3 Solution

Part 4 (20 pts.)

Finally, consider linear regression with the LASSO penalty. Using a pre-built implementation, train the model using a large number of possible values for λ.

Using 10-fold cross validation, find a reasonable value of λ that should minimize the expected prediction error. You can choose the actual minimum or a slightly less complex model (smaller λ is less complex). Defend this choice.

Create a plot that demonstrates the regression coefficients for the LASSO regression with your optimal choice of λ. Which predictors are important? Which ones are not?

Part 4 Solution

Part 5 (10 pts)

Which of OLS with all predictors, OLS with your chosen subset of predictors, Ridge, or LASSO has the smallest cross validation estimate of expected prediction error? Do you have any intuition as to why this result occurs?

Using the optimal models from each step, compute an estimate of the expected prediction error using the heldout test data. Does the same relationship hold?

Part 5 Solution