Problem Set #2

Kevin McAlister

2023-01-19

This is the second problem set for QTM 347. It contains a few proofs related to linear regression and a programming problem that is intended to demonstrate a concept related to prediction error. You may complete this problem set in groups of up to 3. Be sure to indicate any collaborators at the top of your solutions document. All students must turn in a copy of the solutions.

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.

Your final submission on Canvas should include, at a minimum, 2 files: 1) a .Rmd/.ipynb file and 2) either a rendered HTML file or a PDF.

This assignment is due by January 27th, 2023 at 11:59 PM EST.

Problem 1 (35 pts.)

Linear regression is a fundamental tool for statistics and machine learning. At its core, linear regression is a simple task: given a set of P predictors, $\{\mathbf{x}_i\}_{i=1}^N = \boldsymbol{X}$, with each $\mathbf{x}_i$ a $P+1$-vector of predictors with a 1 as the first element (to account for an intercept) and outcomes, $\{y_i\}_{i=1}^N = \boldsymbol{y}$, find the $P+1$-vector $\hat{\boldsymbol{\beta}}$ that minimizes the residual sum of squares:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}^*}{\text{argmin}} \left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)' (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*) \right]$$

This can also be expressed as a summation over errors:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}^*}{\text{argmin}} \left[\sum_{i=1}^N \left(y_i - \boldsymbol{\beta}^{*\prime} \boldsymbol{x}_i \right)^2 \right]$$

Part 1 (10 pts.)

Working with matrix derivatives, show that the $\hat{\boldsymbol{\beta}}$ that minimizes the sum of squared errors is:

$$\underset{\boldsymbol{\beta}^*}{\text{argmin}} \left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)' (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*) \right] = \left(\boldsymbol{X}'\boldsymbol{X} \right)^{-1} \boldsymbol{X}'\boldsymbol{y}$$

and that the Jacobian (the vector of first derivatives with respect to the elements of $\boldsymbol{\beta}^*$) is:

$$2\mathbf{X}'\mathbf{X}\boldsymbol{\beta}^* - 2\mathbf{X}'\mathbf{y}$$

Part 1 Solution

Part 2 (10 pts.)

Part of what makes OLS such a popular algorithm is that it is conceptually simple **and** it has really nice computational properties. One of these properties is that the solution you found above is a **unique minimum** to the sum of squared errors objective function. Argue that the solution above is a unique minimum to the squared error objective function. You can assume that there is nothing weird in the features or the outcome - our feature matrix is of full column rank, $N \geq 2$, and $N >> P$.

Hints:

1) By definition, $\hat{\boldsymbol{\beta}}$ can take any value in $\mathbb{R}^P$. Therefore, there are no exterior boundaries.
2) One way to show that the solution is a unique minimum is to argue that the objective function is strictly convex in $\boldsymbol{\beta}^*$. If you take this approach, I recommend arguing that the matrix of second derivatives is a specific kind of definite.
3) A different method has you do a multivariate version of the second derivative test. This is largely the same as the above approach, but you need only show that a specific point in $\mathbb{R}^P$ is of a specific sign.

Part 2 Solution

Part 3 (15 pts.)

Frequently, we seek to perform **inference** on the values of $\boldsymbol{\beta}$ - we want to determine if the noise associated with the OLS estimator is small enough to say that each β_j is statistically different from zero. First, we want to show that the OLS estimator is **unbiased** for the true value of $\boldsymbol{\beta}$ so that we can claim that our inference is meaningful. Then, we need to derive the **standard error** of the estimator to perform inference. As shown in class, both of these properties are heavily involved in computing the generalizability of a linear regression model.

Show that $\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}$ is unbiased for $\boldsymbol{\beta}$; e.g. $E_{y|x}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$. Then, derive the **variance-covariance matrix** for $\hat{\boldsymbol{\beta}}$; e.g. $\text{Cov}_{y|x}[\hat{\boldsymbol{\beta}}]$. The square root of the diagonal of the variance-covariance matrix then provides the **standard errors**.

Some helpful identities:

1. For multiple linear regression, we assume that $\boldsymbol{X}$ is constant while $E_{y|x}[\boldsymbol{y}] = X\boldsymbol{\beta}$ and $\text{Cov}_{y|x}[\boldsymbol{y}] = \sigma^2\mathcal{I}_N$ where σ^2 is the constant error variance (e.g. a scalar) and $\mathcal{I}_N$ is the $N \times N$ identity matrix.
2. Suppose we want to know $\text{Cov}[\boldsymbol{Ay}]$ where $\boldsymbol{A}$ is a matrix of constants and $\boldsymbol{y}$ is a random vector. Then $\text{Cov}[\boldsymbol{Ay}] = \boldsymbol{A} \times \text{Cov}[\boldsymbol{y}] \times \boldsymbol{A}'$

Part 3 Solution

Problem 2 (65 pts.)

In class, we demonstrated that mean squared prediction error for the linear regression model can be expressed as a function of the **in-sample** mean squared error of the regression fit. Specifically, when the true outcome generating function is of the form $\mathbf{x}'\boldsymbol{\beta}$ and we know that we have all of the predictors, we can compute the limiting expected prediction error as:

$$MSPE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + V[\mathbf{x}'\boldsymbol{\beta}]$$

Furthermore, when we assume that $\mathbf{X}_D$ is fixed, we can further simplify this to:

$$MSPE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + V_{y|x}[\mathbf{x}'\boldsymbol{\beta}] = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \sigma^2\text{tr}(\mathbf{x}_0'(\mathbf{X}_D'\mathbf{X}_D)^{-1}\mathbf{x}_0)$$

Under these restrictive assumptions, the variance term of the linear regression model actually has a simple asymptotic form that gives some information about how the size of the training data and the number of parameters estimated relates to model variance and, in turn, increases the gap between in-sample fit and the expected out of sample fit.

Since this behavior is only defined in expectation, it can be difficult to demonstrate it computationally. However, we can leverage **simulations** to broadly show these properties hold when we create data in such a way that we know all of the properties of the true outcome generating function. This problem will see you build this simulation and make some broad conclusions about how the **optimism gap** changes as a function of N and P.

Part 1 (25 pts.)

Build a function called `pred_gap` that takes three arguments: 1) `n` - the number of training observations to generate, 2) `p` - the number of training features to generate per observation, and 3) `sig2` - the variance of the noise component. Using these inputs, do the following:

1. Generate $N \times P$ random draws from a uniform distribution between -3 and 3. Arrange these draws into a $N \times P$ matrix of training features, $\mathbf{X}$.
2. Generate P more random draws from a uniform distribution between -3 and 3. These will be your true outcome generating regression coefficients, $\boldsymbol{\beta}$. Generate a random draw between -3 and 3 to be your intercept, α, as well.
3. Compute a "denoised" outcome N-vector: $\mathbf{X}\boldsymbol{\beta} + \alpha$
4. Now, we're going to add the noise. Take N random draws from a normal distribution with mean 0 and variance equal to `sig2`. Add these N random draws to the denoised outcome vector and save them as `y_train`.
5. Using the same denoised outcome vector (e.g. $\mathbf{X}$ is fixed), take N more random draws from a normal distribution with mean 0 and variance equal to `sig2`. Add these N random draws to the denoised outcome vector and save them as `y_test`.
6. Using `y_train` as your outcome and $\mathbf{X}$ as your set of features, estimate the set of coefficients that minimize the in-sample mean squared error for a linear regression model (e.g. compute a regression fit using OLS).
7. Using this set of coefficients, generate $\hat{y}_i$ (e.g. predictions) for each observation in the training feature set.
8. Using these predictions, compute the in-sample MSE using `y_train`. Similarly, compute the "out-of-sample" MSE of the fit using `y_test`.
9. Finally, return the size of the scaled optimism gap: $\frac{(\text{Out MSE} - \text{In MSE})}{\sigma^2}$. Note that this final returned value should be a single number.

Hints:

1. There's a really nifty trick in R that will help you greatly here. Suppose you have a data frame, `df`, with a column `y` that is your outcome variable and the rest of the columns are features that should be included on the right-hand side of the regression equation. You can easily denote this in the R formula language as `lm(y ~ ., data = df)`.

Part 1 Solution

Part 2 (10 pts.)

Create a function that wraps around your previous function called `pred_gap_rep` that repeats the above procedure a fixed number of times. This function should have 4 inputs: 1) `n`, 2) `p`, 3) `sig2`, 4) `reps` - the number of times to repeat the procedure outlined in `pred_gap`. This function should:

1. Create a holder for `reps` values of returned estimates of the optimism gap with fixed N and P.
2. Run `pred_gap reps` times and store the size of the scaled optimism gap.
3. Return the average value of the scaled optimism gap at the fixed values of N and P.

Because the properties that we're trying to show only hold in expectation, repeating the procedure a number of times and taking the average reduces the noise that accompanies the value of interest.

Hints:

1. Replicating the function can be done using a for loop in any language. In R, you can also use the function `replicate()` to repeat a particular expression `n` times.

Part 2 Solution

Part 3 (20 pts.)

Using `pred_gap_rep`, we're going to show what happens to the prediction gap as N and P change.

Let's start by examining what happens as N increases. Setting P to 10, run `pred_gap_rep` for 25 replicates setting N equal to 100, 200, 300, 400, 500, 750, 1000, 1500, 2000, 2500, 3000, 4000, and 5000. Plot the returned average gaps as a function of N.

Create two similar plots setting P equal to 5 and 50 and combine the three plots into a single grid plot.

What appears to be the functional relationship between N and the size of the prediction gap for fixed P? Is the relationship linear? Inverse? Quadratic? Log?

Now, do a similar procedure fixing N and varying P. Setting N to 500, run `pred_gap_rep` for 25 replicates setting P equal to values between 2 and 50 by 2. Plot the returned average gaps as a function of P.

Create two similar plots setting N equal to 100 and 2500 and combine the three plots into a single grid plot.

What appears to be the functional relationship between P and the size of the prediction gap for fixed N? Is the relationship linear? Inverse? Quadratic? Log?

Hints:

1. Even though we're averaging over a number of simulation runs, we are still approximating limiting behavior. This can lead to some unintuitive results in cases where we're not really approximating "infinity". In particular, it is totally possible that you'll see that the gap is negative when N is large and P is small. Do not be alarmed! Just know that this can happen and it's not an indication of being incorrect on your part.
2. Depending on how efficiently your base function runs, the replicates may take some time. When possible, avoid for loops and replace them with **matrix multiplication**.
3. Your choice for `sig2` doesn't really matter all that much. The relationships you'll find hold generically regardless of the error variance. However, the relationship is seen most cleanly when σ^2 is neither too big nor too small. You can vary σ^2 in different replicates or not. I recommend setting σ^2 to be a randomly drawn value between 1 and 3.
4. Be wary of really overinterpreting wiggles in functions when the values are really small!

Part 3 Solution

Part 4 (10 pts.)

Provide some intuition for the results you uncovered above. Why does increasing P cause the gap to change in the way it does? Why does increasing N cause the gap to change in the way it does? Think carefully about **why** increasing P changes the model variance.

Similarly, explain why we may not expect these relationships to hold in a more realistic setting where our simplifying assumptions of fixed $\mathbf{X}$ and true linearity are not met.

Part 4 Solution