Linear Model Selection

Data Science 101 Team

Reading

Material from:

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2021. An Introduction to Statistical Learning, Second Edition.

Referred to as ISLR in the following. Freely available online.

Subset Selection

- We are going to look at some methods for selecting subsets of predictors.
- Suppose you have a dataset and would like to predict outcome Y. However, you have many, many predictors (say there are p of them) to choose from. Which ones to select?

Best Subset Selection

- Idea: Fit a separate least squares regression for each possible combination of the p predictors. This means that we fit all p models containing exactly one predictor, then all models that contain exactly two predictors, and so on.
- Then, look at all models and find the one that is *best*.

Best Subset Selection

Step 1: Let M_0 denote the model with no predictors. It just predicts the sample mean for each observation.

Step 2: On the training set: For each $k = 1, ..., p$:

- Fit all models containing exactly k predictors.
- Pick the best among the models containing exactly k predictors, call it M_k. *Best* here means e.g. having the smallest RSS (Residual Sum of Squares).

Best Subset Selection

- To select among one of the $p + 1$ models (all M_k and M_0) we need to be careful: We know that the RSS decreases on the training set with increasing number of predictors. So if we would simply use the training RSS we would always select the largest one.

Step 3: Select the model that has lowest error on the "test set", use cross-validation ...

Can you see a problem with this approach?

Best Subset Selection

- Computationally infeasible (even with modern computers) if p is great than around 40.
- Aside: There are 2^p models that involve subsets of p predictors. So if $p = 10$, then there are 1,000 models to be considers, if $p = 20$ there are more than one million!
- Huge search space: The larger the search space, the higher the chance of finding models that look good on the training data (but might not have predictive power on future data).

Forward Stepwise Selection

- Idea: Begin with a model containing no predictors, and then add predictors to the model one-at-a-time, until all predictors are in the model.
- In each step, add the variable that gives the greatest *additional* improvement in the fit.
- Not guaranteed to find the **best** model.

Forward Stepwise Selection

Step 1: Again, let M_0 be the model with no predictors.

Step 2: For $k = 0, ..., p - 1$:

(a) Look at all $p - k$ models which increase the number of predictors in M_k with one additional predictor.

(b) Choose the *best* among those $p - k$ models, call it M_{k+1} (again, *best* means e.g. having smallest RSS).

Step 3: Select a single best model from using e.g. the test data, cross validation . . .

Forward Stepwise Selection

- Example: Suppose you have a dataset with $p = 3$ predictors.
- Suppose the best possible one-variable model contains X_1. The best possible two-variable model however contains X_2 and X_3!
- Then forward stepwise selection will fail to select the best possible two-variable model, because M_1 contains X_1, and so M_2 must contain X_1 (and an additional variable)!

Backward Stepwise Selection

- Compared to forward stepwise, it starts with the full least squares model containing p predictors.
- It then removes the *least useful* predictor, one-at-a-time.
- As forward stepwise, not guaranteed to find the **best** model.

Backward Stepwise Selection

Step 1. Let M_p be the model containing all p predictors.

Step 2. For $k = p, p - 1, .., 1$:

(a) Consider all k models containing all except one of the predictors in M_k (i.e. having $k - 1$ predictors).

(b) Then choose the *best* (e.g. smallest RSS) among the k models, call it M_{k-1}.

Step 3: Select the single best model among $M_0, ..., M_p$ using the test set, cross validation, . . .

The concept of “best”

- In order to select the best model with respect to test error, we need to estimate this test error. There are two common approaches:

1. We can directly estimate the test error (using e.g. a validation set approach, or cross-validation).
2. We can indirectly estimate the test error by making adjustments to the training error to account for the bias due to overfitting.

The concept of “best”

- We have seen before that the training MSE is an underestimate of the test MSE.
- When we fit a model to the training data using least squares, we specifically estimate the regression coefficients in a way such that the training RSS (not the test RSS) is as small as possible.
- In particular, if we add more variables to the model, the training error decreases, but the test error may not.

The concept of “best”

- Thus, we can’t use the training RSS to select among a set a set of models with different numbers of variables.
- There are several methods out there, such as C_p, AIC, BIC and the Adjusted R^2. The idea of these is to add different types of penalties for the number of predictors.
- If you are interested in learning more about these, check out section 6.1 in ISLR (not necessary).

Shrinkage

- The methods we considered above all used least squares to fit linear models that contain subsets of the predictors.
- However, we have briefly seen alternatives where we can fit a model using all p predictors, and then shrinks coefficient estimates towards zero: Ridge and the Lasso.
- Let's recall what they are doing:

Review: Ridge Regression

- Least squares tries to find $\beta_0, ..., \beta_p$ such that

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})$$

is minimized.

- Ridge is similar to least squares, except that we now want to minimize:

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

Review: Ridge

- If $\lambda = 0$, the penalty has no effect, and we get the least squares estimates.
- If lambda goes to ∞, the ridge regression estimates will approach zero.
- Ridge includes all p predictors in the final model. It will shrink all of the coefficients to zero, but not set them exactly to zero (unless $\lambda = \infty$).
- Including all p predictors might not be a problem for accuracy, but could be a challenge for interpretation.

Review: Lasso

- The lasso minimizes

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

.

- The lasso penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero. Hence, it performs variable selection. Why exactly is beyond the scope of this class, see section 6.2 in ISLR if you're interested.
- When $\lambda = 0$, then the lasso gives the least squares fit. When λ gets very large, the lasso gives the null model with all coefficient estimates equal to zero.

Selecting the tuning parameter

- So how do we select λ, the "tuning parameter" in Ridge and Lasso?
- A good way to do so is using cross-validation.
- What is cross-validation?

Review: Validation / Test Set Approach

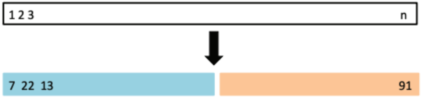

Figure 5.1 in ISLR.

Idea: Randomly split our dataset of n observations into a training set (blue) and a validation set (beige). First fit on the training set, then evaluate performance on validation set.

- Do you see any “disadvantages”?

Validation Set Approach

Potential Disadvantages:

1. Validation estimate of the test error can be variable, depending on precisely which observations are included in the training, and which in the validation set.
2. Only a subset of the observations are used to fit the model. Statistical methods tend to perform worse with fewer observations.

Cross validation, a “refinement” of the validation set approach, addresses these two issues.

Leave-One-Out Cross Validation

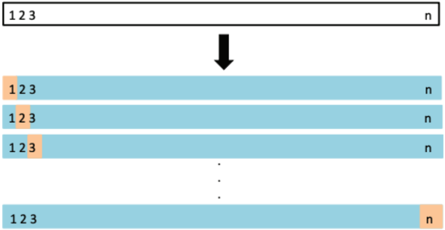

Figure 5.4 in ISLR.

- Repeatedly split into train (blue, all but one observation) and validation (beige). The first training set contains all observations except observation 1, the second all but observation 2, etc.
- Test error is estimated by averaging the n resulting MSEs.

5-Fold CV

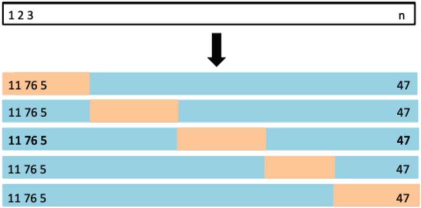

Figure 5.5 in ISLR.

- Randomly split into 5 non-overlapping groups. Each of the fifth act as a validation set (beige) and the remainder as training set (blue).

- Test error is estimated by averaging the five resulting MSEs.

How many folds?

- Typically, one performs cross validation using 5 or 10 folds.