Chapter 6 Notes: Linear Model Selection and Regularization

Statistical Learning with R

Jon Geiger

February 21, 2022

While the standard linear model, $Y = \beta_0 + \beta_1 X + \cdots + \beta_p X_p + \epsilon$, is very useful for problems of inference and even prediction, but there are improvements which can be made.

Alternative models can offer better *prediction accuracy* and *model interpretability*:

- If $n \gg p$, a linear model will have very low variance (deviation in the model's behavior from one set to the next). However, if p is comparable to n, or worse, $p > n$, the variance can be very high or even *infinite*
- It's easiest to interpret a model with only relevant variables. Since least squares coefficients are almost never *exactly* zero, we need to compensate by selecting the variables we use very carefully, using **feature selection** or **variable selection**.

This chapter looks at three alternatives to using least squares to fit a model:

1. **Subset Selection**: identifying a subset of the p predictors which we believe to be related to the response, then fitting least squares on that set.
2. **Shrinkage**: fitting a model involving all p predictors, then shrinking the estimated coefficients relative to the least squares estimates. This is also known as **regularization**. Because this shrinks coefficients, it can estimate some to be almost exactly zero, so this method also performs a form of automatic variable selection.
3. **Dimension Reduction**: involves projecting the p predictors into an M-dimensional subspace, where we have M different linear combinations of the variables. We then use these projections to fit a regression model.

6.1: Subset Selection

The **Best Subset Selection** method fits a linear model for each possible combination of predictors, so 2^p total models will be fit. This is done by the following algorithm:

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model just predicts the sample mean for each observation ($\beta_0 = \bar{y}$)
2. For $k = 1, 2, \ldots, p$:
   a. Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   b. Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

We cannot use RSS or R^2 to select from the $p+1$ models because these values improve monotonically with more predictors. So we must use some other method for choosing the "best" model. In other words, increasing the number of predictors will always decrease the training set error, but we care about the test set error.

For something such as logistic regression, instead of ordering models by RSS in Step 2 of the algorithm, we would use *deviance*, which is negative two times the maximized log-likelihood.

Generally, best subset selection is very computationally intensive, since if $p = 20$, we would need to fit over a million models.

Stepwise Selection is a much more computationally efficient process which only fits a fraction of the models which Best Subset Selection uses.

The *Forward Stepwise Selection* algorithm is as follows:

1. Let $\mathcal{M}_0$ denote the null model, containing no predictors.
2. For all $k = 0, \ldots, p-1$:
    a. Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
    b. Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p (AIC), BIC, or adjusted R^2.

This process essentially starts out with a model with no predictors, then adds the best predictor on top of that. Then another predictor is placed on top of that, and so on. Then, out of all the $p+1$ models with $0, \ldots, p$ predictors, the best model is chosen.

The *Backward Stepwise Selection* algorithm is as follows:

1. Let $\mathcal{M}_p$ denote the full model, containing all p predictors.
2. For all $k = p, p-1, \ldots, 1$:
    a. Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
    b. Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using C_p (AIC), BIC, or adjusted R^2.

This is the same as forward stepwise selection, but it begins with a full model and works its way backwards.

Generally, the training error is a poor estimate of the test error. We can either *adjust* the training error to account for bias, or we can directly estimate the test error using cross-validation.

Estimating the test error without a cross-validation method involves calculating several estimates. These are: C_p, *Akaike information criterion* (AIC), *Bayesian information criterion* (BIC), and *adjusted* R^2.

For a least squares model containing d predictors with n observations, we have:

- $C_p = \dfrac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$ \ The C_p adds a penalty of $2d\hat{\sigma}^2$ to the training RSS to adjust for the underestimation of the test error. It turns out that if $\hat{\sigma}^2$ is an unbiased estimator of σ^2. We select the model with the lowest C_p value.

- AIC $= \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2)$ \ The AIC is defined for a large class of models fit by maximum likelihood with Gaussian errors. In this case, the AIC is proportional to C_p, where constants are irrelevant due to minimization.
- BIC $= \frac{1}{n}(\text{RSS} + \log(n)d\hat{\sigma}^2)$ \ The BIC is derived from a Bayesian point of view, but gives a larger penalty to models with larger amounts of variables, so results in smaller model selections.
- Adjusted $R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$ \ Where normal R^2 is defined as $1 - \text{RSS}/\text{TSS}$, the adjusted R^2 takes into account the ratio of the number of predictors to the number of observations. In other words, adjusted R^2 will decrease with noise predictors, or predictors that add to d while only causing a small decrease in the RSS. The best model should be that which maximizes the Adjusted R^2.

Alternatively, the test error can be directly estimated using cross-validation. This used to be computationally prohibitive, but with modern computing power this is becoming a more attractive option.

Models can be selected using the *one-standard-error rule.* This rule says that we calculate the standard error of the test MSE for each model size, then select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve. The rationale for this is that if models seem to be roughly equally good, we might as well choose the simplest model.

6.2: Shrinkage Methods

As an alternative to just choosing a subset of predictors (6.1), we can fit a model which contains all p predictors and *constrains* or *regularizes* the coefficient estimates towards zero. Recall that:

$$\text{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Least squares involves choosing parameters which minimize the RSS. In **Ridge Regression**, the ridge regression coefficient estimates $\hat{\beta}^R$ are the values which maximize:

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

where $\lambda \geq 0$ is a tuning parameter, determined separately. The second term here is called a *shrinkage penalty*, and is smallest when $\beta_1, \ldots, \beta_p$ are all close to zero. The tuning parameter λ, then, controls the impact that the second term has on estimating the regression coefficients. Ridge regression will estimate coefficients $\hat{\beta}^R$ for each value of λ, and selecting a good value of λ will yield a better test MSE.

If $\hat{\beta}$ is the vector of least squares coefficient estimates, then $||\hat{\beta}||$ is the magnitude or *norm* of this vector, and we call $||\hat{\beta}||_2$ is the ℓ_2 *norm*, and is defined as $||\hat{\beta}||_2 = \sqrt{\sum_{j=1}^{p}\hat{\beta}_j^2}$, and the ℓ_2 norm of the ridge regression coefficients are defined similarly. As the tuning parameter increases, the ratio of $||\hat{\beta}_\lambda^R||_2/||\hat{\beta}||_2$ will decrease. In other words, the parameters will generally decrease in magnitude as the tuning parameter is increased.

In standard least squares, coefficients are *scale invariant*, in that if each X_j is multiplied by some constant c, the coefficient will simply be multiplied by $1/c$ to compensate. This is not the case with ridge regression. Because of this, we need to make sure that all the predictors are on the same scale, so we standardize them using the formula:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

where the denominator of the fraction is the estimated standard deviation of the jth predictor. This means that all the standardized predictors will have a standard deviation of one.

In terms of bias-variance trade-off, ridge regression decreases variance at the cost of increasing bias. This is because from set to set, the coefficients will generally be smaller, so the variability in the coefficients will also be smaller.

The **Lasso** is another alternative to simple linear regression, which overcomes the interpretive difficulty of not being able to fully eliminate some variables from the model. With the RSS as the same quantity as with ridge regression, the lasso coefficients, $\hat{\beta}^L_\lambda$ minimize the quantity

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

The lasso and ridge regression are similar, but the lasso uses the ℓ_1 norm rather than the ℓ_2 norm, where the ℓ_1 norm of a coefficient vector β is given by $||\beta||_1 = \sum |\beta_j|$. With the tuning parameter λ sufficiently large, some coefficients will be exactly equal to zero. This creates a *sparse model*, which is a model which includes only a subset of the variables.

Because the lasso also serves as a model selection technique, it implicitly assumes that a number of the features have coefficients exactly equal to zero. This assumption is not always a good one, but in the case where there are certain features with low correlation, it can be useful to eliminate them completely. Neither regression technique always dominates the other.

Generally, ridge regression performs better with many predictors, and the lasso performs better with a smaller number of predictors.

Selecting the Tuning Parameter λ can be a tricky problem. Cross-validation allows us to choose a grid of λ values and compute the CV error for each value of λ as described in Chapter 5. Then, we'll select the value of λ for which the CV error is minimized, then re-fit the model using all available observations and the selected value of the tuning parameter.

This is highly important for determining which variables are *noise* and which are *signal.* The lasso does an excellent job at picking out variables which are actually significant without picking up variables which just add noise. See Figure 6.13 in the textbook (p. 251).

6.3: Dimension Reduction Methods

Let $Z_1, Z_2, \ldots, Z_M$ represent $M < p$ linear combinations of our original p predictors. That is,

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

for some constraints $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$, for $m = 1, \ldots, M$. In other words, the Z_m are the new, transformed predictors that we can use to fit our model with. We can then fit the linear regression model

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n$$

using least squares. The regression coefficients for reduced dimensionality are given by $\theta_0, \theta_1, \ldots, \theta_M$. Rather than fitting a model with p predictors in X space, we fit a model with M predictors in Z space. We can notice that:

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

So we can treat this as a special case of linear regression. When $p > n$, selecting a value of $M \ll p$ can significantly reduce variance of the final model.

Choosing $Z_1, Z_2, \ldots, Z_M$, or choosing ϕ_{jm}'s, can be done in different ways. We'll look at *principal components* and *partial least squares.*

Principal Components Analysis (PCA) is useful for transforming a large number of features into a small number.

The **First Principal Component** direction of the data is that along which the observations *vary the most.* In other words, if the points were projected onto the line, and the variance along that line were measured, it would be maximized. This is also the line which is *as close as possible* to the data.

The **Second Principal Component** direction must be orthogonal to that of the first principal component. Similarly, this is such that the variance of the projected data on this component will be maximized.

Principal Components Regression involves taking the first M principal components and using those components as predictors in a linear regression model fit using least squares.

- We assume that *the directions in which $X_1, \ldots, X_p$ show the most variation are the directions that are associated with Y.* This turns out to be a pretty good approximation in most cases.
- We mitigate overfitting by being able to fit a model on fewer predictors since most or all of the information is contained within $Z_1, \ldots, Z_M$.
- For analyzing model performance, plots of variance, squared bias, and test MSE versus number of components are useful.
- When $M = p$, PCR just results in regular multiple linear regression.
- PCR is *not* guaranteed to outperform ridge regression or lasso, but can perform well if most of the data is described by a low number of principal components.
- PCR is *not* a feature selection method, and is very closely related to ridge regression, and (allegedly) ridge regression can even be thought of "as a continuous version of PCR." (consult section 3.5 of ESL for more detail)
- PCR is an *unsupervised* technique because the response Y is *not used* to determine the directions of the principal components.

Partial Least Squares is a *supervised* alternative to PCR which *does* use the response to form new predictors. Similar to PCR, we compute the PLS directions in order to perform regression.

- After standardizing the predictors, PLS computes the first direction Z_1 by setting each ϕ_{j1} equal to the coefficient from the simple linear regression of Y onto each X_j. That is, simple linear regression is performed for each predictor individually for Y.
- In computing $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.
- "To identify the second PLS direction we first *adjust* each of the variables for Z_1, by regressing each variable on Z_1 and taking *residuals*. These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction. We the compute Z_2 using this *orthogonalized* data in exactly the same fashion as Z_1 was computed based on the original data."
- Once $Z_1, \ldots, Z_M$ are computed, we use least squares to fit a linear model to predict Y the same way as in PCR. As with PCR, the value of M is chosen by cross-validation. We standardize the predictors and response before performing PLS.
- PLS can reduce bias, but it has the potential to increase variance, so typically, PCR is used in the place of PLS.

6.4: Considerations in High Dimensions

With advances in technology over the past 20 years, it is commonplace to collect an extremely high amount of feature measurements (p very large), whereas the number of observations n is limited due to cost, availability, or other considerations. Two examples are as follows:

1. Rather than predicting blood pressure based on age, sex, and BMI, one might also collect measurements for half a million individual DNA mutations common to the population (SNPs). In this case, $n \approx 200$ and $p \approx 500,000$.
2. A marketing analyst might look at the words which appear in an individual user's search history. For a given user, each of the p search terms is scored present (0) or absent (1), creating a huge feature vector. If information is gathered for 1000 people, then $n = 1000$ and p is incredibly large, equal to the number of words in the digital lexicon being used.

If a data set has $p > n$, it is classified as *high-dimensional*.

If data is high-dimensional, least squares regression should not be performed. With $p \geq n$, least squares will always form a perfect fit to the training data, with zero variance. This is overfitting and we need to consider other methods for fitting data such as shrinkage methods and dimension reduction methods.

In high dimensions, as the number of variables increases to $p \approx n$, the value of R^2 tends to 1, the training MSE tends to zero, and the test MSE increases at an incredible rate due to it being incredibly biased.

When performing regression with high-dimensional data, it is useful to fit *less flexible* least squares models, such as using forward stepwise selection, ridge regression, lasso, and PCR.

As an example, when performing lasso with $p = 20$ features, a relatively low value of λ can be chosen to minimize the test MSE. However, when we have many more features not contributing any useful information, a relatively high value of λ should be chosen to scale down or eliminate the noise variables.

There are three main important points about regression in high dimensional data:

1. Regularization or shrinkage plays a key role in high-dimensional problems
2. Appropriate tuning parameter selection is crucial for good predictive performance
3. The test error tends to increase as the dimensionality of the problem increases, unless the additional features are truly associated with the response. This is known as the *curse of dimensionality*.

In general, *adding additional signal features that are truly associated with the response will improve the fitted model*. However, adding noise will lead to deterioration in the fitted model and an increased test set error.

Multicollinearity is a *huge* problem in high-dimensional data, since any feature can basically be written as a linear combination of other features.

In the case of having half-a-million SNPs, we might conclude from forward stepwise selection that 17 of those SNPs lead to a good predictive model on the training data. This just means that those 17 form one good model out of many possible models.

Reporting errors is also difficult with high-dimensional data, as we've seen that it's very easy to get a useless model with zero residuals, and thus SSE, p-values, and R^2 values should not be used. Because of this, it's very important to use a cross-validation set and calculate independent test MSE. Test MSE or R^2 is a valid measure of model fit, but training MSE is not.