1 Linear Model Selection and Regularization

Linear models are often simple and easy to interpret at the cost of having high bias if the relationship in the data is not linear. Some considerations about linear models

- If $n >> p$, least square estimates often have less variance. If n is larger than p, then least square estimates can have some variance. While if $n < p$, we are looking at non unique solutions which can cause lot of variation in the test predictions.
- It is often the case that many of the predictors do not have a relationship with the response. Hence, it is a good idea to remove those and make the model more interpretable at the cost of some bias. Least square estimates almost never give zero coefficients.

There are major ways in which the number of variables in the model can be reduced

- Selecting a **subset of variables** that go well with the response. This itself can be done by forward selection, backward elimination etc.
- **Shrinking** some of the **coefficients** to zero. This is a great help in reducing the variance of the predictions.
- **Dimension Reduction** helps in projecting the p predictors onto a M dimensional space where $M < p$. This utilizes linear combinations to create a set of new features.

1.1 Subset Selection

1.1.1 Best Subset Selection

This is a naive approach that essentially tries to find the best model among 2^p models that are trained on all possible subsets of the p variables. As we increase the subset of variables, the training error will monotonically decrease whereas the same cannot be said for the test error. A number of criteria like test MSE, R^2, AIC etc can be used to pick the models.

In case of classification models, similar argument holds and a more general error metric *deviance* can be used. *Deviance* is defined as $-2 * \log \textit{likelihood}$ of the data. The smaller the *deviance*, the better the model fit.

The huge search space presented by this approach easily overfits as the search space presents more opportunities to find better fits. However, this causes a higher variance in the predictions on future data and can possibly also have higher test error.

1.1.2 Forward Stepwise Selection

This is a greedy approach that significantly shrinks the search space being checked (in comparison to the best subset selection approach).
Forward Stepwise Selection Algorithm

1. Let M_0 denote the null model, i.e., the model with no predictors
2. For $k = 0, 1, \ldots, p - 1$
   (a) Consider all $p - k$ models formed by adding a single predictor to the model M_k
   (b) Select the best model M_{k+1} among the $p - k$ models on the basis of the error metric
3. From the models $M_0, M_1, \ldots, M_p$, select the one with the lowest cross validation error on the evaluation choosing the appropriate error metric

This approach effectively has reduced the search space from 2^p to $1 + p(p + 1)/2$. Although, now it is not guaranteed that the model selected will be the best one among 2^p.

1.1.3 Backward Stepwise Selection

This approach is the opposite of forward stepwise selection. We recursively reduce the number of variables in our model.

1. Let M_p denote the complete model, i.e., the model with all p predictors
2. For $k = p, p-1, \ldots, 1$
   (a) Consider all $k-1$ models that keep all but one predictors in the current model M_k
   (b) Among these, select the best model M_{k-1} with the lowest error
3. From the models $M_0, M_1, \ldots, M_p$, select the one with the lowest cross validation error on the evaluation choosing the appropriate error metric

The number of models explored is same as the forward stepwise method.
A hybrid approach is usually selected where we start with the usual forward selection method, but while adding variables, we do not add a variable if it does not give significant improvement. Another approach can be to remove redundant variables using p-test at every step of forward selection.

1.2 Metrics for evaluating Subset Models

In linear models, as we add more variables, the training error usually monotonically decreases. Test error may not behave in the same way. When training a model, the coefficients obtained are specific for minimizing the training error and hence will have less bias in comparison to the test error.
Hence, subset evaluation using training error will usually favour models with more number of variables. To overcome this

- Correct the training error estimate to correctly calculate test error
- Use a validation test or k-fold validation for better estimate of test error

1.2.1 C_p Estimate

For a least square fitted model,

$$C_p = \frac{1}{n}(RSS + 2p\hat{\sigma}^2)$$

p is the number of predictors and $\hat{\sigma}^2$ is the estimate of the error associated with each observation. This is typically evaluated using the model built on all p predictors.
Clearly, as p increases, we are penalizing the model more to compensate for the decrease in the training RSS. When $\hat{\sigma}$ is an unbiased estimate of σ, we can show that this is infact an unbiased estimate of the test MSE.

1.2.2 Akaike Information Criterion (AIC)

AIC is defined for a large class of models fit by the maximum likelihood estimate.
For least squares fit in linear models, the errors are assumed to be gaussian and thus, AIC and least squares mean the same thing. For this case

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2p\hat{\sigma}^2)$$

where we have omitted an additive constant for the sake of simplicity.
For least squares models, C_p and AIC are proportional to each other.

1.2.3 Bayesian Information Criterion (BIC)

BIC is derived from a Bayesian point of view, but ends up looking similar to the above defined errors.
For least squares error without constants, the BIC is

$$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + \log(n)p\hat{\sigma}^2)$$

Note that the $\log(n)$ term will put a heavier weight on the error term for large p. Hence, BIC will tend to select models with lower number of variables in comparison to say C_p.

1.2.4 Adjusted R^2

Recall that R^2 is defined as $1 - RSS/TSS$. Adjusted R^2 is

$$Adjusted\ R^2 = 1 - \frac{\frac{RSS}{n-p-1}}{\frac{TSS}{n-1}}$$

This adjusted R^2 might increase or decrease when adding variables due to the terms corresponding to p. The intuition is that, after the correct number of variables have been identified, the decrease in RSS is less in comparison to the decrease in $n-p-1$ which will slightly increase the Adjusted R^2.

1.3 Shrinkage Methods

Instead of using a subset of predictors, we can also use all of the predictors and shrink the coefficients towards zero. This approach significantly reduces the variance in the model estimates. The famous ones here are *Ridge Regression* and *Lasso Regression*.

1.3.1 Ridge Regression

Ridge Regression is very similar to the least square estimate for linear regression except that we add a term corresponding to the squared sum of the regression coefficients in the error.

$$\begin{aligned} error &= RSS + \lambda \sum_{j=1} p\beta_j^2 \\ &= (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \end{aligned}$$

λ is a tuning parameter that needs to be chosen separately. It acts as a weight between the error in the data and how large are the regression coefficients. It is also known as the shrinkage penalty.
Note that we will not include the intercept term in shrinkage because it is simply the mean estimate of the model when all the predictors are zero and may not necessarily zero.

Using least squares estimate,

$$\begin{aligned} error &= (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta \\ \frac{\partial error}{\partial\beta} &= 0 \\ \implies 0 &= -Y^TX - Y^TX + \beta^TX^TX + \beta^TX^TX + \lambda\beta^T + \lambda\beta^T \\ \beta^T(X^TX + \lambda I) &= Y^TX \\ (X^TX + \lambda I)^T\beta &= X^TY \\ (X^TX + \lambda I)\beta &= X^TY \\ &\boxed{\beta = (X^TX + \lambda I)^{-1}X^TY} \end{aligned}$$

$\lambda = 0$ will result in the simple least squares regression while $\lambda \to \inf$ will force the coefficients to go towards zero.

As is clear from the formula, the error term is sensitive to the actual scale of the coefficients which is ultimately dependent on the predictors themselves. In a simple least squares regression, the coefficients will scale up and down depending on how the data is scaled. The same is not true for Ridge Regression.
Hence when using **Ridge Regression, it is always advisable to *standardize* the predictors** before training the model using

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

The success of Ridge Regression is based in the **bias variance tradeoff**. If the data is linear, simple linear regression will have a very low bias but high variance, making it sensitive to the training data. As λ is introduced, it forces the model to have less flexibility by reducing the coefficiets value and subsequently their power on the prediction. This causes a reduction in the variance at expense of slight increase in bias. However, this trend is not monotonic with increasing λ and the appropriate value must be chosen based on the errors observed.