FERAY ECE TOPCU

BDA 552

Pre-Class Homework 1

All answers from the book "Introduction to Statistical Learning with Applications in R".

1. **List and explain potential problems in linear regression models.**
When we fit a linear regression model to a particular data set, many problems may occur. Most common among these are the following:

    1. *Non-linearity of the response-predictor relationships.*

       If the relationship between predictors and response variable is not linear, linear regression cannot fit the solution of problem.
       Solution: residual plot. The residuals exhibit a clear U-shape, which provides a strong indication of non-linearity in the data.

    2. *Correlation of error terms.*

       An important assumption of the linear regression model is that the error terms, _1, _2, . . . , _n, are uncorrelated. If in fact there is correlation among the error terms, then the estimated standard errors will tend to underestimate the true standard errors. As a result, confidence and prediction intervals will be narrower than they should be.
       Solution: residual plot. If the errors are uncorrelated, then there should be no discernible pattern.

    3. *Non-constant variance of error terms.*
       Another important assumption of the linear regression model is that the error terms have a constant variance, Var(_*i*) = σ2.
       Solution: One can identify non-constant variances in the errors, or heteroscedasticity, from the presence of a funnel shape in heteroscedatheresidual plot.

    4. *Outliers.*
       outliers are observations for which the response yi is unusual given the predictor xi.

    5. *High-leverage points.*
       observations with high leverage have an unusual value for xi.

    6. *Collinearity.*
       Collinearity refers to the situation in which two or more predictor variables are closely related to one another.
       Solution: examine correlation matrix. For multicollinearity; *variance inflation factor*.

(Book ref: *3.3.3 Potential Problems*, pg 92)

2. Compare Logistic Regression, Linear Discriminant Analysis and Quadratic Discriminant Analysis. What are the advantages and disadvantages?

1.Logistic Regression: if classes are well separated, logistic regression does not fit very well on dataset.

2. Linear Discriminant Analysis: if classes are well separated, logistic regression does fit very well on dataset. The assumption of LDA is all predictors are normally distributed (fit Gaussian), So if the predictors are not normally distributed, LDA may not work very well. Additionally, LDA assumes same assumption for all predictors on covariance matrix.

3. Quadratic Discriminant Analysis: the difference between LDA and QDa is that QDA assumes different assumption for each predictor on covariance matrix.

When the true decision boundaries are linear, then the LDA and logistic regression approaches will tend to perform well. When the boundaries are moderately non-linear, QDA may give better results. Finally, for much more complicated decision boundaries, a non-parametric approach such as KNN can be superior. But the level of smoothness for a non-parametric approach must be chosen carefully.

(Book ref: 4. Classification, pg 154)

3. What are the differences between forward stepwise selection and the backward stepwise selection?

Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
In particular, at each step the variable that gives the greatest *additional* improvement to the fit is added to the model.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (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 having smallest RSS or highest 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.

Backward stepwise selection begins with the full least squares model containing all predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (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 having smallest RSS or highest 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.

Backward selection requires that the number of samples *n* is larger than the number of variables *p*. In contrast, forward stepwise can be used even when n < p, and so is the only viable subset method when p is very large.

(Book ref: 6.1 Subset Selection, pg 208)

4. Why does Ridge regression improve over the least squares?

Ridge regression's advantage over least squares is rooted in the *bias-variance trade-off*. **As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias.**

In general, in situations where the relationship between the response and the predictors is close to linear, the least squares estimates will have low bias but may have high variance. This means that a small change in the training data can cause a large change in the least squares coefficient estimates. In particular, when the number of variables *p* is almost as large as the number of observations *n*, as in the example in Figure 6.5, the least squares estimates will be extremely variable. And if $p > n$, then the least squares estimates do not even have a unique solution, whereas ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance. Hence, ridge regression works best in situations where the least squares estimates have high variance. Additionally, Ridge regression also has substantial computational advantages over best subset selection, which requires searching through 2^p models.

(Book ref: 6.2 Shrinkage Methods, pg 214-217)

5. What are the differences between the Lasso and Ridge equation?

Lasso and Ridge regressions are closely related to each other and they are called shrinkage methods. We use Lasso and Ridge regression when we have a huge number of variables in the dataset and when the variables are highly correlated.We can fit a model containing all p predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero. The two best-known techniques for shrinking the regression coefficients towards zero are *ridge regression* and the *lasso*.

To control variance, we might regularize the coefficients. *Ridge regression* is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity.

While least squares try to minize RSS, Ridge regression is try to minize RSS + shrinkage penalty:

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

Where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately. The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates. When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as $\lambda \rightarrow \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero.

Ridge regression does have one obvious disadvantage. Unlike best subset, forward stepwise, and backward stepwise selection, which will generally select models that involve just a subset of the variables, **ridge regression will include all *p* predictors in the final model.**
The penalty λβ2j in (6.5) will shrink all of the coefficients towards zero, but it will not set any of them exactly to zero (unless λ = ∞**). Increasing the value of *λ* will tend to reduce the magnitudes of the coefficients, but will not result in exclusion of any of the variables**. This may not be a problem for prediction accuracy, but it can create a challenge in model interpretation in settings in which the number of variables **p is quite large.**

The *lasso* is a relatively recent alternative to ridge regression that over comes this disadvantage. The lasso coefficients, $\hat{} \beta L \lambda$, minimize the quantity:

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

While we compare the shrinkage penalty term; we can see that Lasso uses |Bj| instead of square of Bj**. (Lasso uses L1 regularization while Ridge uses L2 Regularization.)**
So, the **lasso performs *variable selection*** **by equalize some coefficients to zero.** As a result, models generated from the lasso are generally much easier to interpret than those produced by ridge regression.

In Summary;
- Both of Lasso and Ridge; while *λ* increases, the variance decreases and the bias increases.
- **Lasso uses L1 regularization while Ridge uses L2 Regularization.**
- Lasso performs variable selection.
- In general, one might expect the lasso to perform better in a setting where a relatively small number of predictors have substantial coefficients, and the remaining predictors have coefficients that are very small or that equal zero. Ridge regression will perform better when the response is a function of many predictors, all with coefficients of roughly equal size.
- As with ridge regression, when the least squares estimates have excessively high variance, the lasso solution can yield a reduction in variance at the expense of a small increase in bias, and consequently can generate more accurate predictions.

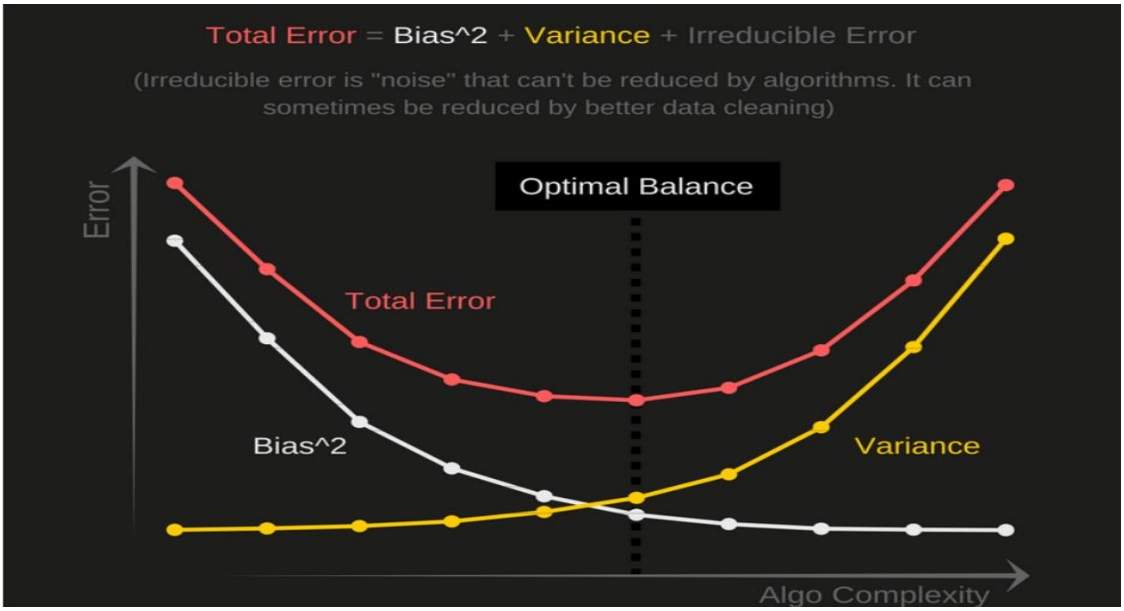
(Book ref: 6.2 Shrinkage Methods, pg 214,223)

6. Explain the bias-variance trade-off.

A simple definition, bias is the difference between predictions of model and the true values. Bias is the model's assumption to make the target function easier to learn. So, high bias suggests more assumptions about the form of the target function while low bias suggest less assumptions. Example of high bias ML algo's: Linear Regression and Logistic Regression and example of low bias ML algo's: KNN, SVM, Decision trees. An example of bias error; if we try to apply linear regression on non-linear dataset, the model does not fit on it; Result will be under-fitting. Or, if we have highly correlated two features on linear regression model, they will cause over-fitting because of their high weight of bias. (high bias causes under-fitting!)

When we look at variance, it is the amount of the change on estimation of target function if train data is changed. Ideally, variance should not keenly change when train data is changed. (I mean from same dataset, different train data.) Low variance suggests small changes to estimate of the target with changes to the training data while high variance suggest large changes. So, high variance causes overfitting! Example of low-variance ML algo's: Lienar and Logistic Regression and example of high variance ML algo's: KNN,SVM Decision trees.

The goal of any supervised machine learning algorithm is low bias with low variance for good prediction. While increasing the bias causes decreasing the variance (or vice versa) there is trade-off when they balance due to some methods which are varied according to ML algorithm.



(Extarnal ref: https://elitedatascience.com/bias-variance-tradeoff **)**
(Book ref: *2.2.2 The Bias-Variance Trade-Off* , pg 34)

7. Why does k-fold cross-validation give more accurate estimates of the test error rate than the leave-one-out cross-validation?

LOOCV is a special case of k-fold CV in which k is set to equal n. The advantage of *k*-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV. This has to do with a bias-variance trade-off.

LOOCV has higher variance than does *k*-fold CV with *k* < *n*. Because; when we perform LOOCV, we are in effect averaging the outputs of *n* fitted models, each of which is trained on an almost identical set of observations; therefore, these outputs are highly (positively) correlated with each other. In contrast, when we perform *k*-fold CV with $k < n$, we are averaging the outputs of *k* fitted models that are somewhat **less correlated** with each other, since the **overlap** between the training sets in each model is smaller. Since the mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated, the test error estimate resulting from LOOCV tends to have higher variance than does the test error estimate resulting from *k*-fold CV.

Leave-one-out cross-validation:

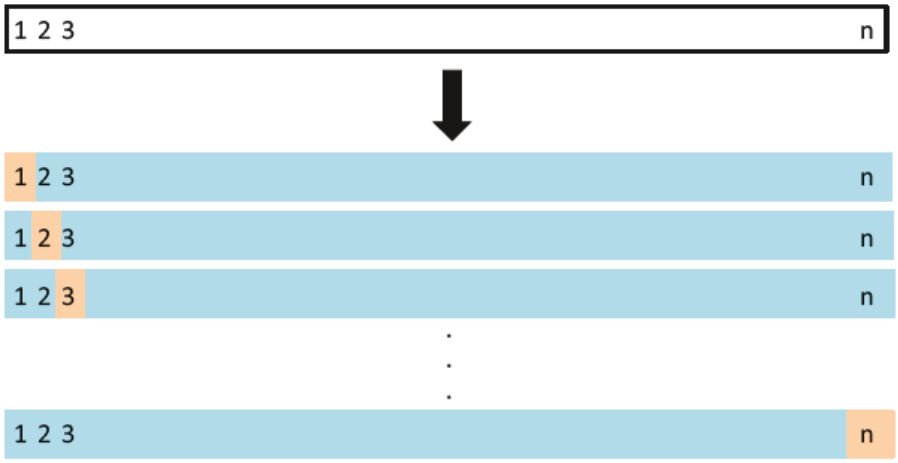


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

K-fold cross-validation:

This approach involves randomly k-fold CV dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds.

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

(Book ref: 5.1 Cross-Validation, pg 176)