Chapter 2 notes

Karan Mann

Statistical Learning

What is Statistical Learning?

Input variables are denoted by **X**, with subscripts to differentiate them. The input variables are also sometimes referred to as the predictors, independent variables, features or just variables. The output variable is often called the response or dependent variable, and is denoted by **Y**. We observe a quantitative response **Y** to p predictors X_1 **to** X_p. There is an assumption that there is some relationship between **Y** and **X**. We can write this relationship as:

$$Y = f(X) + \epsilon$$

f is a fixed but unknown function of X, and ϵ is an error term that is independent of X and has mean zero. f represents the systematic information X provides about Y. The essence of statistical learning is finding ways to approximate f.

Why Estimate f?

The two main reasons for estimating f are prediction and inference.

Prediction: Frequently we have available inputs **X**, but outputs **Y** are not as readily available. In these situations, we can predict Y using $\hat{Y} = \hat{f}(X)$, since ϵ has mean zero. $\hat{f}$ is typically treated as a black box as we are not concerned with implementation, but rather that it provides accurate predictions for $\hat{Y}$. The accuracy of $\hat{Y}$ for **Y** depend on the reducible error and the irreducible error. The reducible error is an outcome of using $\hat{f}$ to estimate f. Our estimation is not going to be perfect, hence there is clearly some error present. We can minimize this reducible error however, by using the best available statistical learning technique. However, Y is also a function of ϵ and $\hat{Y}$ isn't. Epsilon, by definition, is the error term that can not be predicted using **X**. Variability associated with ϵ will also influence the accuracy of our models. The irreducible error term is not 0 because it may contain unmeasured variables that predict Y, or it may represent immeasurable variation. We can show:

$$E(Y - \hat{Y})^2 = (f(x) - \hat{f}(x))^2 + Var(\epsilon)$$

where $E(Y - \hat{Y})^2$ is the expected value of the squared difference between the predicted value and the actual value and $Var(\epsilon)$ represents the variance in the error term ϵ. We aim to adopt techniques that minimize the reducible error. The irreducible error will always provide an upper bound for the accuracy of our prediction for Y.

Inference: Often, we are interested in understanding the relationship between **Y** and $X_1, ..., X_p$. We still wish to estimate f, but now we wish to understand the construction of $\hat{f}$. We attempt to understand what predictors are significantly associated with the response, the relationship between the response and each predictor and whether a linear model would accurately predict the response. Our ultimate goal, whether it is prediction or inference or a combination of both, guides us towards the model we wish to use.

How Do We Estimate f?

There are many linear and non-linear methods to approximating f. We always assume we have observed a set of n different data points. These observations are called the training data, because we use these observations to train or teach our method to estimate f. x_{ij} represents the value of the j^{th} predictor for the i^{th} observation. y_i represents the response variable for the i^{th} observation. Our training data then consists of $(x_i, y_i$ for i = 1 to n, where x_i is the column vector that contains the p input measurements for the i^{th} observation. We want to find a function $\hat{f}$ such that Y $\approx$ $\hat{f}$(X) for any observation (X,Y). Most statistical learning methods can be categorized as either parametric and non-parametric.

Parametric methods: These methods involve a 2-step model based approach. The first step is to make an assumption about the structure of f, like f is linear in X. This assumption has greatly reduced the difficulty of estimating f. We no longer need to construct a p-dimensional function and can simply estimate the p + 1 coefficients of f. The second step involves we need a procedure to use the training data to fit or train the model. In the linear model, we need to estimate the coefficients $\beta_0..\beta p$. The least squares method is one of many ways to fit a model. The problem for the parametric approach is our assumed form for f can be a bad estimation of the true f. Here, it is likely we get inaccurate predictions. We can address this problem by choosing models that are more flexible giving us a larger number of functional forms to choose from. However, fitting a more flexible model means we must estimate a greater number of parameters. This can lead to a phenomenon called over-fitting which means we follow the error or noise too closely.

Non-parametric methods: These methods do not assume a functional for f. Instead, we attempt to get f to best fit the data points without getting too rough or wiggly. Non-parametric methods allow us to accurately fit a larger range of possible forms for f. The disadvantage of using a non-parametric method is that since we are not reducing the problem to estimating a small number of coefficients, we need a large number of observations to accurately estimate f. We wish for our fitted model to be smooth to decrease the likelihood of over-fitting the model. When we over-fit our model, we have the model replicate our observed data very closely. However, when we attempt to predict for observations where the response is unknown, we yield inaccurate predictions.

The Trade-off Between Prediction Accuracy and Model Interpretability

It is clear now that some methods are flexible and some methods are restrictive. The least-squares method, for example, is restrictive because it restricts possible functional forms of f to only linear models. Thin plate splines, on the other hand, are much more flexible and therefore capable of producing many different structures. We might prefer a more restrictive model when we are also interested in interpretability.

Supervised Versus Unsupervised Learning

In the supervised learning domain, all of the observations have an associated response measurement. The aim is to use the predictor measurements to accurately predict unknown responses, or better understand the relationship between predictors and the response. In the unsupervised learning domain, we have observation predictor measurements, but do not have associated response measurements. It is called unsupervised because we lack a response to guide our learning. Sometimes, we have some m response measurements associated with our observations, but n - m observations do not have a response. In this case, we call it a semi-supervised learning problem.

Regression Versus Classification Problems

Variables can be quantitative or qualitative. Quantitative variables take on numerical values. Qualitative variables take on values from K different categories or classes. When the response is quantitative, we call the problem a regression problem. When the response is qualitative, we call the problem a classification problem. We select models based on whether the response is quantitative or qualitative. However, whether the predictors are quantitative or qualitative is less important.

Assessing Model Accuracy

There is not a single best method that most accurately predicts all datasets. Different methods optimize accuracy for different datasets.

Measuring the Quality of Fit

To evaluate performance of a model, we need to examine how close our predictions are to the true response values. In other words, we wish to quantify how close our predictions are to observed data. In regression, a commonly used measure is mean squared error.

$$MSE = \frac{1}{n}\sum_{n=1}^{n}(y_i - \hat{f}(x_i))^2$$

where $\hat{f}(x_i)$ is the prediction for the i^{th} observation. The MSE will be small if the predicted values are close to the true response values, and will be large if for some of the observations the predicted values differ largely from the true response values. The aforementioned MSE is computed using the training data, so it is called the training MSE. We are more interested in evaluating our model against new, unseen data. Mathematically, we wish to knew whether $\hat{f}(x_0)$ is approximately y_0, where y_0 is an unmeasured response. We want to choose the method that results in the lowest possible test MSE, not the lowest possible training MSE. In practice, if we had a large number of test observations, we could take the average of the squared difference between the actual response for a test observation and the predicted response for a test observation and choose the model that minimizes this function. Sometimes, we have test observations we can use in this procedure. In other cases, we have no test observations. We might seek to use the training MSE as a solution, but there is no guarantee that a low training MSE means a low test MSE. A fundamental problem is many methods specifically choose coefficients to minimize the training MSE. It follows, of course, that the training MSE will be low in these models. Flexible models that fit the observed data very well do not necessarily provide good estimates of the true f. We can formally denote the flexibility of a model as the degrees of freedom. The degrees of freedom quantity tells us about the level of flexibility in a model. The training MSE will decrease monotonically as degrees of freedom increases. When a given method yields a low training MSE but a high test MSE, we are over-fitting the data. Over-fitting happens when our model is trying to closely simulate the patterns of the data, however, some patterns may be the result of random chance and not a prediction of the bigger picture. The test MSE will be high in this case because the supposed patterns our model detected do not exist in the test observations. Over-fitting specifically refers to the case when a less flexible model would have resulted in a smaller test MSE.

The Bias-Variance Trade-off

The expected test MSE, for a given value x_0 can be decomposed into three fundamental quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$, and the variance of the error terms ϵ.

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0)]^2 + Var(\epsilon)$$

The expected test MSE refers to the value we would get by averaging a large number of possible estimates for f at the observation x_0. The overall expected test MSE can be obtained by averaging out the expected test MSEs at all x_0 in the test set. The decomposition tells us that to minimize the expected test MSE, we want to minimize the variation of $\hat{f}(x_0)$ and the squared bias of $\hat{f}(x_0)$. We recall that ϵ is an irreducible quantity. Variance and squared bias are non-negative quantities so $Var(\epsilon)$ provides a lower bound for the expected test MSE.

Variance refers to the amount $\hat{f}$ would change if we trained it on different training data. Ideally, the estimate for f should not vary much between training sets. A method would high variance would have large changes in $\hat{f}$ for small changes in the training data. Generally, more flexible models have higher variance. This follows from the fact that changing a data point in a highly flexible model (one which follows the training data very closely) would change the model substantially. Bias refers to the error associated with approximating a function that can extremely complex by a simple model. A non-linear true f approximated by a linear $\hat{f}$ will have high bias whereas a linear true f approximate by a linear $\hat{f}$ might have low bias. Typically, more flexible models have lower bias.

General rule: as models get more flexible, variance will increase and bias will decrease. The relative rate of change of these two quantities will determine whether the expected test MSE increases or decreases.

The relationship between bias, variance and test set MSE is called bias-variance trade-off. Finding a model with a low squared bias and low variance is challenging, but is the model one strives for.

The Classification Setting

What do we do when our response y_i is qualitative?

The most common approach for quantifying the accuracy of our model is the training error rate.

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

$\hat{y}_i$ is the predicted class label placed on the i^{th} observation and I is an indicator variable that outputs 1 if $y_i \neq \hat{y}_i$ and 0 otherwise. In other words, 1 is output if the prediction is classified incorrectly, and 0 is output if the prediction is classified correctly. This is for training data, but as in the regression case, we are more interested in the test error rate. A good classifier is one for which test error rate is lowest.

Bayes Classifier Method: The test error rate is minimized by a simple classifier that assigns each observation to its most likely class, given its predictor values. In other words, we wish to assign a test observation x_0 to the class j such that

$$Pr(Y = j|X = x_0)$$

is largest. The above is a conditional probability, held upon the condition that $X = x_0$. This simple classifier is called the Bayes classifier. In a two-class problem, the statement $Pr(Y = 1|X = x_0) > 0.5$ tells us which class to assign $\hat{y}_0$ to, as it specifies that the probability x_0 is associated with class 1 is greater than the probability that x_0 is associated with class 2.

The Bayes decision boundary is the boundary that specify which areas belong to which class. More specifically, the Bayes decision boundary corresponds to the points at which Pr($Y = j \mid X = x_0$) is exactly 0.5. Consequently, the Bayes classifier chooses what class to place an observation in, based on the Bayes decision boundary.

The Bayes classifier provides the lowest possible error rate, called the Bayes error rate. The error rate will be 1 - $max_j Pr(Y = j|X = x_0)$. In words, the error rate will be the largest possible error rate possible minus the probability that x_0 belongs to the class for which its probability is highest. The overall Bayes error rate is given by 1 - E($max_j Pr(Y = j|X = x_0)$) where the expectation averages the probability over all possible values of X.

K-Nearest Neighbours: For real data, we do not know the conditional distribution of Y given X, and so computing the Bayes classifier is impossible. Many approaches like the K-Nearest Neighbours classifier attempts to estimate the conditional distribution of Y given X, then classify observations based on the highest estimated probability.

Given a positive integer K, and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0 represented by N_0. We then estimate the conditional distribution of Y, by calculating the probability the training observations in N_0 belong to the j^{th} class. We find this through the equation:

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

KNN then classifies x_0 to the class with the highest conditional probability. We note that typically small K's correspond to more flexible models, and large K's correspond to less flexibility. In other words, flexibility decreases as K increases. This is true because following a smaller set will follow small variations in the data, instead of following the general pattern of the data.

Once again, there is not a strong relationship between the training error rate and the test error rate. The training error, such as when K=1, is small, but the test error could be large.