

Chapter 2

2.1 What is Statistical Learning?

- X denotes the input variable (aka: predictor or independent variable)
- Y denotes the output variable (aka: response or dependent variable)
- The relationship between X and Y can be written as:
$$Y = f(X) + \epsilon$$
- f is a fixed, but unknown function of X and ϵ is the error term.
- ϵ is independent of X and has a mean of 0.
The function f may take more than one input variable. (e.g. income (y) as function of education (X_1) and seniority (X_2)).

Statistics is all about ways to estimate f

2.1.1 Why Estimate f ?

1. Prediction: this is when we know the values of X , but can't easily determine Y .

- $\hat{y} = \hat{f}(x)$
- \hat{Y} is the resulting predicting for Y (aka the predicted response)
- \hat{f} is the estimate for f .
 - It is a *black box* - where we don't really care about \hat{f} as long as it gives accurate predictions for Y .

Generally, \hat{f} will not be a perfect estimate f , as a result the inaccuracy will introduce some error.

Types of Error:

- *reducible error*: can be reduced to improve the accuracy of \hat{f} by using better statistical methods.
- *irreducible error*: since Y is a function of ϵ , not all of the error can be reduced. Therefore there will always be ϵ

Reasons that ϵ is not zero

- ϵ is not zero because it might include variables that are useful in predicting Y .
 - Since we don't measure these unincluded variables they can't be predicted using f .
- ϵ may also contain unmeasurable variation, which also can't be predicted using f

So if we have estimate \hat{f} and predictors X we get the prediction: $\hat{Y} = \hat{f}(X)$.

If we assume that \hat{f} and X are fixed:

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) - \epsilon - \hat{f}(X)]^2 \\ &= [f(X) - \hat{f}(X)]^2 + \text{var}(\epsilon) \end{aligned} \quad (1)$$

- $E = (Y - \hat{Y})^2$ is the expected value of the squared difference between the predicted value and actual value of Y .
- $\text{Var}(\epsilon)$ is the variance associated with the error term ϵ

The irreducible error gives an upper bound on the accuracy of our prediction for Y & will almost always be unknown in practice.

2. Inference

- *Inference* is when we want to know how Y is affected by change in the predictors, $X_1 \dots X_p$, but aren't necessarily interested in making predictions for Y .
- the goal is to understand the relationship between X and Y . How Y changes as a function of $X_1 \dots X_p$
- \hat{f} can't be treated as a *black box* because we have to know its exact form.
- linear models are useful for inference.

Inference is useful for:

- determining which predictors are associated with the outcome.
- determining the relationship between the outcome and each predictor.
- determining whether the relationship between Y and each predictor can be summarized using a linear equation or whether the relationship between the two is more complicated.

2.1.2 How do we Estimate f ?

- n is the number of data points or observations we have.
- *training data* is a subset of the data we have that we use to train (or teach) the method how to estimate f .
- We apply a statistical learning method to the training data in order to estimate f .
- x_{ij} is the value of the j th predictor for observation i .
 $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, p$
 y_i is the response variable for i th observation.
- The training data would be $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $x = (x_{i1}, x_{i2}, \dots, x_{ip})^T$.

The goal is to find \hat{f} such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

There are two statistical learning methods we can use:

1. *Parametric*: to estimate f we only need to estimate one set of parameters.
 - the problem is that it will usually not match the true unknown form of f
 - if the model is too far off from the true f (or the f using all the observations), the estimate will be poor
 - to solve poor fit, we can use more flexible models. But more flexible models requires estimating more parameters.
 - more complex models can lead to *overfitting*: which means they follow the errors too closely.
 - these involve a two-step model-based approach.

Step 1 We make an assumption of f 's form.

For example, if f is linear:

- $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$
- If f is linear, you only need to estimate the coefficients $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_p X_p$

Step 2 We use the training data to *fit* or *train* the model.

For the linear model we want to estimate: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$

One method of fitting the model is (*ordinary*) *least squares*

2. *Non-parametric*:

- do not make explicit assumptions about the functional form of f .
- these methods try to get as close to the data points without being too rough.
- since they don't assume particular form of f , they can fit a wider range of shapes for f .
- will fit the data better since it does not assume the form of f .
- requires substantially more observations in order to get an accurate estimate for f than parametric approaches.

2.1.3 The Trade-off Between Prediction Accuracy and Model Interpretability

Some methods are less flexible because they can produce only a small range of shapes to estimate f (e.g. Linear regression can only create linear functions.)

- less flexible models are better for inference because they are more interpretable.
- it's easier to understand the relationship between Y and X_1, X_2, \dots, X_p

More flexible models include *thin plate splines* can generate a wider range of possible shapes to estimate f .

2.1.4 Supervised vs. Unsupervised Learning

Supervised Learning for each observation of the predictor measurements $x_i, i = 1, \dots, n$ there is an associated response to the measurement y_i .

- Includes: logistic regression, GAM, Boosting, and support vector machines

Unsupervised Learning is more complicated because for every observation $i = 1, \dots, n$ there's a vector of measurements x_i , but no associated response y_i .

- it is called unsupervised because we have no response variable y to supervise our analysis.
 - Includes: cluster analysis (do the observations fall into relatively distinct groups?)

Semi-supervised Learning is when you have observations for the predictors, but the corresponding measurements for the responses are less available.

2.1.5 Regression vs. Classification Problems

Quantitative Variables are number variables.

- problems involving Quantitative data are called *regression problems*.

Qualitative Variables (aka categorical) take on one of a number of *classes* or categories.

- problems involving Qualitative data are called *classification problems*.

The difference between the two is not that clear cut and some methods (*K-nearest-neighbors* and *boosting*) can be used for both quantitative and qualitative data.

Most statistical methods are based on whether the *response* is qualitative or quantitative.

2.2 Assessing Model Accuracy

There is no *best* method in statistical learning. The data you have determines what kind of method you can (and should!) use.

2.2.1 Measuring the Quality of Fit

Measuring the quality of fit is a quantification of the extent to which the predicted response value for a given observation is close to the true response value of the observation.

For regressions, the most common measure is the *mean squared error (MSE)*.

- $MSE = \frac{1}{n} \sum_{i=1}^2 (y_i - \hat{f}(x_1))^2$
 - $\hat{f}(x_i)$ is the prediction \hat{f} gives for the i th observation.
- a small test MSE is the goal
 - A small MSE means that the predicted responses are close to the true responses.
- We're interested in how well our method works with previously unseen data (*i.e.*, not the training data.)
- If we have training observations, which we use to fit obtain an estimate of \hat{f}
 - $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- We then compute $\hat{f}(x_1), \hat{f}(x_2), \dots, \hat{f}(x_n)$
 - If these are approximately equal to y_1, y_2, \dots, y_n then the MSE is small.
- What we want to know is if $\hat{f}(x_0) \approx y_0$ where x_0 and y_0 is a previously unseen test observation *not used to train the statistical learning method*. Essentially, a new data point.
 - This is the *test MSE* which is what we're interested in and the lower the better.
- For a large number of observations we can compute:
 - $Ave(y_0 - \hat{f}(x_0))^2$
 - The average squared prediction error for the test observations: (x_0, y_0)
 - The smaller the number the better
- DO NOT rely on a method that results in a small *training MSE* even though the *test* and *training MSE* are closely related because most methods try to make a small training MSE which does not always result in a small *test MSE*.
 - *Overfitting* Is a method that gives as small *training MSE* but a large *test MSE*.
 - It means that the method may be picking up patterns that are just random chance
- The *training MSE* will generally be smaller than the *test MSE*.
- *Cross-Validation* is a method for estimating the *test MSE* using the training data.

2.2.2 The Bias Variance Trade-Off

The expected *test MSE* (for a given value, x_0) is the sum of three fundamental quantities:

1. The *variance* of $\hat{f}(x_0)$
 2. The squared *bias* of $\hat{f}(x_0)$
 3. The variance of the error terms ϵ .
- this is:
 - $E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$
 - Where $E(y_0 - \hat{f}(x_0))$ is the *expected test MSE*.
 - It is the average test MSE "we'd obtain if we repeatedly estimated \hat{f} using a large number of training sets and tested each at x_0 "

- The test MSE will never be below $Var(\epsilon)$ (the irreducible error)
- Our goal is to have *low variance* and *low bias* through the *bias-variance tradeoff*
 - *Variance*: the amount that \hat{f} would change if we estimated it using a different training data set.
 - Ideally, we would not like \hat{f} to vary much between data sets.
 - a high variance would mean that small changes in the training data results in large changes in \hat{f} .
 - More flexible methods will have higher variance.
 - *Bias*: is the error that is introduced by approximating real life using a simple model.
 - The closer real life is to the model used, the less bias there will be.
 - More flexible methods tend to have less bias.

2.2.3 The Classification Setting

This refers to problems where the outcome of interest is qualitative (y_1, \dots, y_n)

The most common way to quantify the accuracy of the estimate \hat{f} is training the *error rate* - which is the proportion of mistakes that are made if we apply the estimated \hat{f} to the training observations.

Training error rate equation is computed using the data that was used to train the classifier.

- $\frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{y}_i)$
 - \hat{y}_i is the predicted class label for the i th observation using \hat{f}
 - $I(y_i \neq \hat{y}_i)$ is an *indicator variable*
 - $y_i \neq \hat{y}_i$ is coded with a 1
 - $y_i = \hat{y}_i$ is coded with a 0
 - if $I(y_i \neq \hat{y}_i) = 0$ then the i th observation was classified correctly by the method, otherwise it was misclassified.

We are still interested in the *test* error rate which is the error rate returned when we apply the classifier to test observations *not* used in training the data.

- The *test error rate* is:
 - $Ave(I(y_0 \neq \hat{y}_0))$
 - \hat{y}_0 is the predicted class label that results from applying the classifier to the test observation with predictor x_0

We want the test error to be the smallest - which indicates we have a good classifier.

The Bayes Classifier

The test error rate is minimized (on average) by a simple classifier that assigns each observation to the most likely class, given its predictor values.

- we should assign a test observation with predictor vector x_0 to the class j for which the below is the largest
 - $Pr(Y = j|X = x_0)$
 - This is a *conditional probability* equation.
 - It is the probability that $Y = j$, given the observed predictor vector x_0

This is *the Bayes classifier*. In a two-class problem it says that there are only two possible response values.

- it corresponds to predicting the first class if $Pr(Y = 1|X = x_0) > 0.5$, and class two otherwise.
- The *Bayes decision boundary* is where the points have a probability of exactly 50%.
 - The *Bayes classifier* is determined by this boundary.
- The *Bayes classifier* produces the *Bayes error rate* which is the lowest possible test error rate.
 - The overall *Bayes Error Rate*: $1 - E(\max_j Pr(Y = j|X))$
 - where the expectation averages the probability over all possible values of X
 - It is analogous to the irreducible error.

Since we will never know the conditional distribution of Y given X in a real data set, the *Bayes classifier* is an unattainable gold standard against which we compare other models.

K-Nearest Neighbors

Attempts to estimate the conditional distribution of Y given X , then classifies a given observation to the class with the highest *estimated* probability.

- Given a positive integer K and a test observation x_0 , *KNN*:
 1. Identifies the K points in the training data that are closest to x_0 (represented by \mathcal{N}_0)
 2. Estimates the conditional probability for class j as a fraction of points in \mathcal{N}_0 whose response value equals j .
 - $Pr(y = j|x = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$
 3. Applies Bayes rule and classifies the test observation x_0 to the class with the largest possibility.
- The user picks the value for K .
 - A low K will produce a flexible model, but may have low bias but very high variance.
 - The higher the K , the less flexible the model, which will result in low variance but high bias.
- The *training* error and the *test* error do not have a strong relationship.