Statistical Learning Survey

Abraham Neuwirth

May 29, 2017

Contents

1 Introduction

Christopher Bishop (Bishop 2006) in his seminal book "Pattern Recognition and Machine Learning", introduces the field of statistical learning with a classical story:

> [T]he extensive astronomical observations of Tycho Brahe in the 16th century allowed Johannes Kepler to discover the empirical laws of planetary motion, which in turn provided a springboard for the development of classical mechanics.

It's the archetype of statistical learning success stories. Lots of data, brilliant minds, and a model to illuminate and explain it all.

Statistical learning is the study of using observed data to predict how similar data will behave under similar circumstances. The predictions usually take on one of two forms. The first, called regression, is when these statistical methods are used to predict a numerical value. The second is classification which is a catch-all term for predictions that are of a categorical nature, not numeric. Classification is used both to classify data points into a set of predefined groups as well as to cluster data points into disparate groups discovered by using the statistical learning methods on observed data.

In general, statistical learning methods can be divided into two groups: *supervised* and *unsepervised.* Supervised learning generally refers to constructing (or "training") a model with both observed input and output so that when given new input, the model can predict the output. Unsupervised learning, on the other hand, refers to models that are not given any outputs, but only inputs. The role of the model is to discover the classes that divide the data points, or the relationship between various data points, using various statistical techniques.

While the predictions themselves usually need to be computed by a machine (hence the term "machine learning" that is often used to describe the field), the theory behind it relies on the combination of several mathematical fields such as statistics, probability, linear algbera, complexity theory, and more. Many (if not most) of the methods used were discovered long before machines existed, and exploring their mathematical underpinnings leads to the discovery of new and improved statistical learning tools.

2 Regression

The oldest and most commonly used form of statisical learning methods is regression. Regression is the foundation for many supervised learning methods. Supervised learning are classes of problems were we're given data points y_i and x_i, and the task is usually to find a function $f(x) = y$. The role of the function is that $\hat{y}$ needs to be as close to y_i as possible.

Stated mathematically, if we have for example $y = \beta_0 + \beta_1 x$ where β_0 and β_1 are unknown, we can approximate it with known values, such that $y = \beta_0 + \beta_1 x + \epsilon$. Of course, in practice ϵ is unknown, but regression focuses on methods to reduce ϵ as much as possible.

2.1 Linear Regression

One of the most simplest and common statistical learning methods is linear regression. It is also one of the oldest, having been discovered (Kopf 2015) independently by Adrien-Marie Legendre in 1805 (Legendre 1805), and then again by Carl Friedrich Gauss in 1809 (Gauss, n.d.).

In the most simplest terms, the objective of a regression is to find the best line to approximate a given data set consisting of many points (x_i, y_i), where x_i is the independent variable (also caled predictor, input, or feature) and y_i is the dependent variable (also called outcome or prediction).

The objective function is to find estimated values for x, such that the distance between the actual y and $\hat{y}$ is minimized. More formally:

Definition 2.1. A residual r_i is the difference between the actual value of the dependent variable and the the value predicted by the model such that $r_i = y_i - f(x_i, \beta)$

The least squares method optimizes the sum of the squares of the residuals:

$$S = \sum_{i=1}^{n} r_i^2 \tag{2.1}$$

As an example, we'll use data from Henderson and Velleman (1981) to illustrate how regression finds a line through the data by minimizing an objective function. By visually inspecting the data (figure 2.1), we can see that it slopes downward at an approximate rate of -0.05.

A linear regression reveals a slope of -0.04122 with a y-intercept of 29.59985. Adding the line to the data we can see how the regression draws a straight line through the points (figure 2.2).

The form of regression published by Legendre and Gauss was what is now called Least Squares regression. For one varialbe, what we're trying to find is

$$\hat{y} = \beta_0 + \beta_1 x \tag{2.2}$$

where β_0 is the y-intercept of the line, β_1 the slope, x the independent variable (also commonly called the predictor), and $\hat{y}$ the estimated y. To find that, we need to find the line that minimizes the sum of squares of the residuals (formula (2.1)). With calculus it can be derived that

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.3}$$

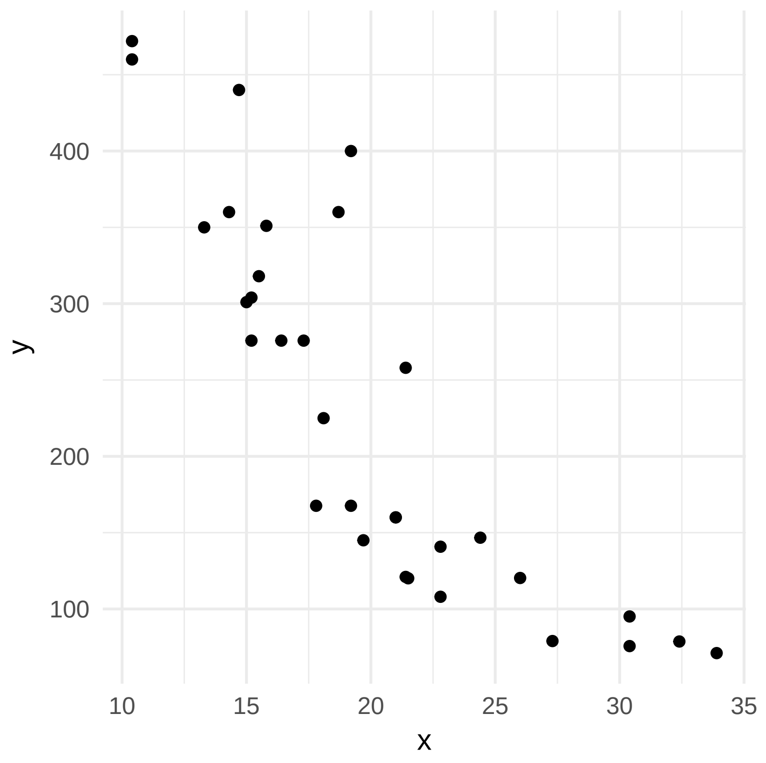

Figure 2.1: the data

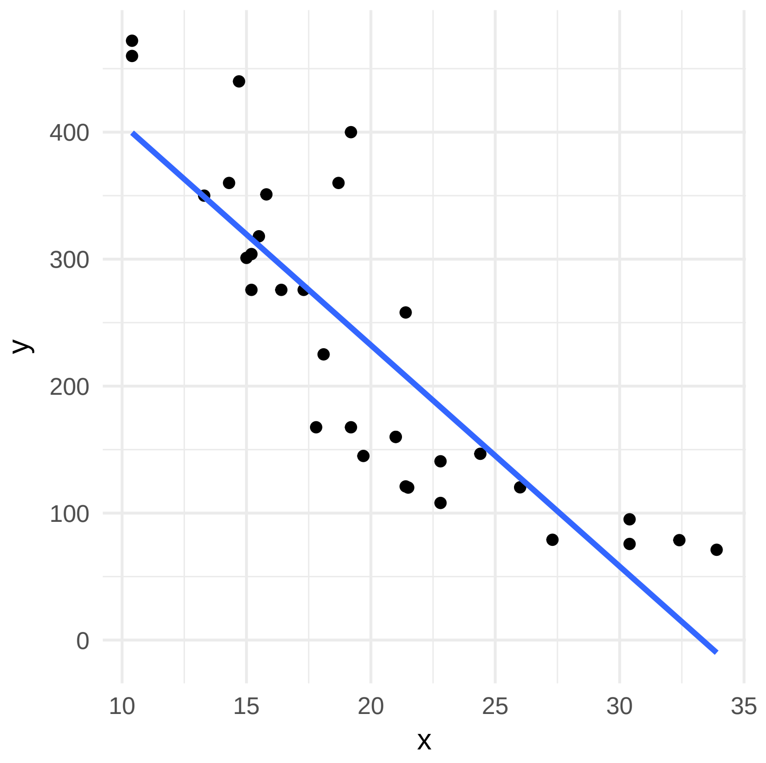

Figure 2.2: regression

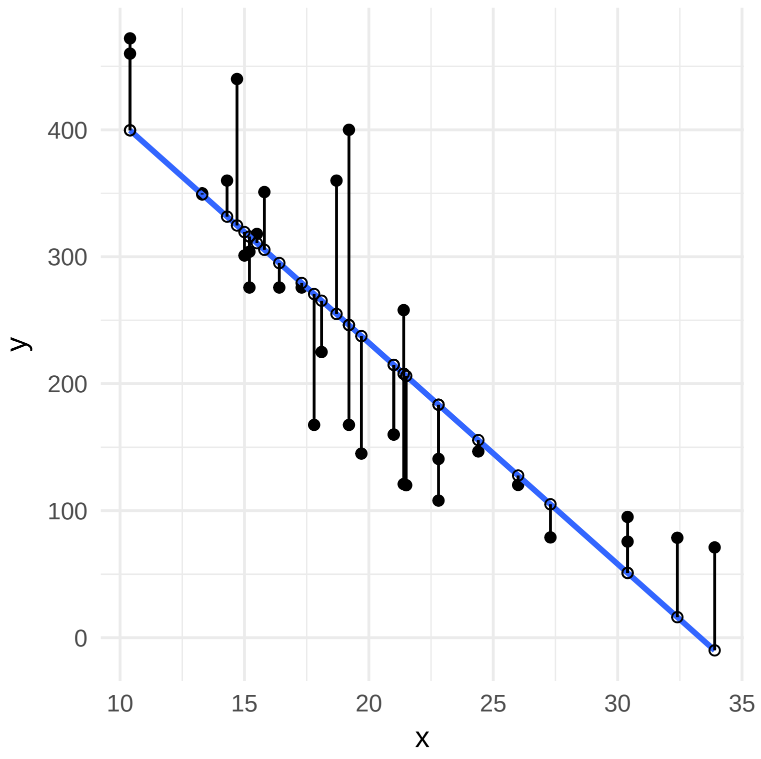

Figure 2.3: residuals

and

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \tag{2.4}$$

where $\bar{x}$ is simply the mean of all observed predictors and $\bar{y}$ are simply the mean value of all observed outcomes.

2.2 Multiple Linear Regression

Simple linear regression can be used to predict the outcome based on a single predictor. But what if we had multiple independet variables which we would like to use to predict the outcome? As an example, using again data from Henderson and Velleman (1981), we can use the independent variables x_1 and x_2 to construct a linear plane in 3 dimensions to predict $\hat{y}$ as depicted in figure 2.4.

In this case we need to find not the line but the plane that minimizes the squared error. With higher dimensional data, our goal is to find a hyperplane that accomplishes the same thing. In such a case we need to use multiple linear regression.

Note that with simple linear regression, all the values used to build the model were scalars, eaither means or

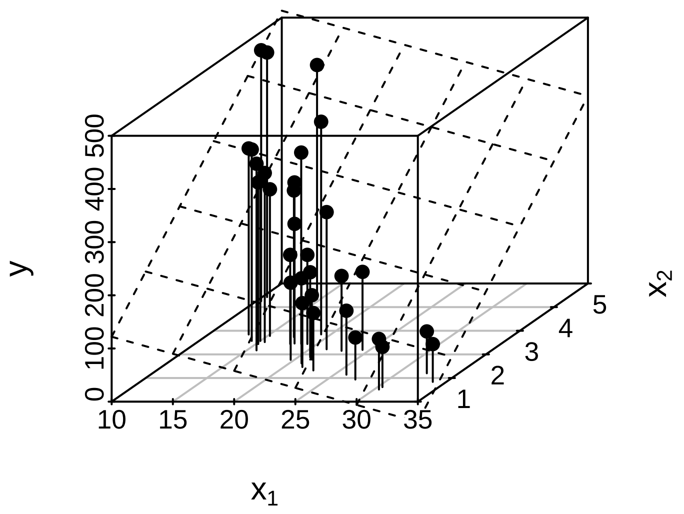

Figure 2.4: two independent variables

sums of the observed values. However, with multiple linear regression such a representation would not suffice. Instead, we represent our data as a matrix, with each column in the matrix representing all observed values for an independent variable, and each row representing a complete singular observation.

As an example, consider a simple linear regression with just one predictor:

$$\hat{y}_i = \beta_0 + \beta_1 x_i \quad \text{for } i = 1, 2, \ldots, n \tag{2.5}$$

In practice, this results in n equations, as follows:

$$\begin{aligned} \hat{y}_1 &= \beta_0 + \beta_1 x_1 \\ \hat{y}_2 &= \beta_0 + \beta_1 x_2 \\ &\vdots \\ \hat{y}_n &= \beta_0 + \beta_1 x_n \end{aligned} \tag{2.6}$$

It is immediatly obvious that a more convenient representation of this data is in matrix form, where we can use matrix multiplication to get $\hat{y}_i$ for $i = 1, 2, \ldots, n$:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \tag{2.7}$$

With the matrix representation, we can just write:

$$\hat{Y} = X\beta \tag{2.8}$$

where $\hat{Y}$ is the outcome matrix, X the predictor matrix, and β the matrix of betas. This can now be generalized easily to as many predictos as we want. If our predictive model looks as follows:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{2.9}$$

where each x_i represents the column vector of one of the k independent variables, we can represent that as a matrix multiplication:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,k} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} = \hat{Y} = X\beta \tag{2.10}$$

Finding the line (or plane, or hyperplane) that minimizes the least squared error is now a systems of linear equations, where we need to find a vector of appropriate βs. It can be shown that formula (2.11) finds the estimates for coefficients for all the predictors that minimizes the squared errors.

$$(X'X)^{-1}X'Y = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} \tag{2.11}$$

2.2.1 Categorical features

So far we've only discussed using numerical predictors in the regression model. But what if we have categorical data which we would like to include in the model? For example, in a model that predicts income based on various socioeconomic data, we might want to include gender or race as a predictor. In order to use categorical features with regression, the values need to be turned into numerical features first. To accomplish that, a dummy vector needs to be appended to the feature matrix. If the categorical feature is binary, one vector suffices. The dummy variable, D_1 will take on the value of 0 or 1. If the feature is not binary, multiple dummy variables will be needed. In general, for a feature with k options, $k-1$ dummy variables will be needed.

The dummy variable can be added to the model in two ways. The first one is if the dummy vector is added to the predictor matrix Y as is as another predictor. In that case, the only affect the dummy variable will have will be to change the intercept based on the binary value of the dummy variably. The second way it can be added to the model is to have it interact with another, numerical, predictor. In that case, the result of the dummy variable will be to shift the slope of the numerical predictor it is multiplied by.

2.3 Generalized Regression

Linear regression makes two important restriction on the nature of the model it produces. The first one is that each predictor is independent from all the other predictors. The second restriction is that the model needs to be linear.

We've already shown (section 2.2.1) that the second restriction can be lifted, by having one predictor interact with another predictor. In other words, instead of the traditional linear model

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \tag{2.12}$$

we can instead multiply individual predictors by each other to get the interaction effect two (or more) predictors might have on each other:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta 3 X_1 X_2 \tag{2.13}$$

The second restriction of linearity can be relaxed by raising the predictors to a power. For example, if we look again at figure 2.1 we might notice that a quadratic formula might fit the data better. Previously, our model was linear in the coefficients and the predictors:

$$\hat{y} = \beta_0 + \beta_1 x \tag{2.14}$$

But if we raise the predictor to the power of two we might get a better fit as illustrated in figure 2.5. For the quadratic fit, the formula looks as follows:

$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 \tag{2.15}$$

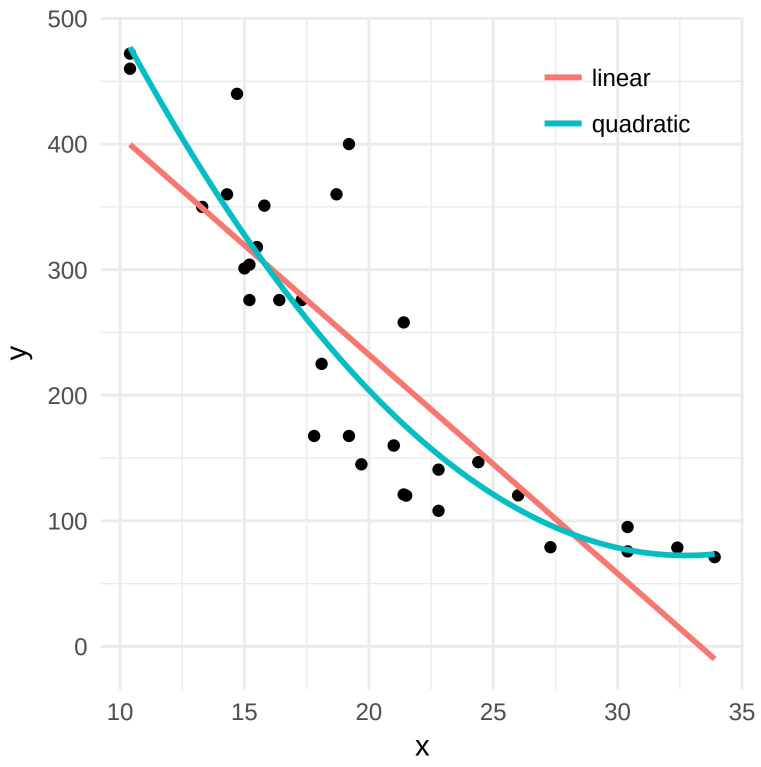

Figure 2.5: linear and non-linear regression

Indeed, if we compare the sum of squared errors of the linear fit (figure 2.3) with the sum of squared errors produced by the quadratic fit (figure 2.6), we see that the quadratic fit minimizes the error more than the linear fit does.

In reality however, the regression is still linear. The reason it produces a non-linear fit is because we transformed one of the predictors by raising it to the power of 2. Other commonly used transformations of predictors are the log and the exponential functions among many other possible transformation functions.

2.4 Loss/Objective Functions

So far we have looked at regression and the measure of its efficiency in terms of squared error. That is, our goal in regression is to find the line that minimizes the sum of the squared differences between the actual y and the predicted $\hat{y}$ (formula ((2.1)).

More formally, we can define it as a loss function.

Definition 2.2. Given a prediction $\hat{y}$ and an actual value y, a loss function $\ell(y, \hat{y})$ measures the divergence of the two.

The goal of a good statistical learning method is to minimize a loss function in an efficient matter (in terms of complexity theory). The Square loss function that we have been using so far, commonly called ℓ_2 loss,

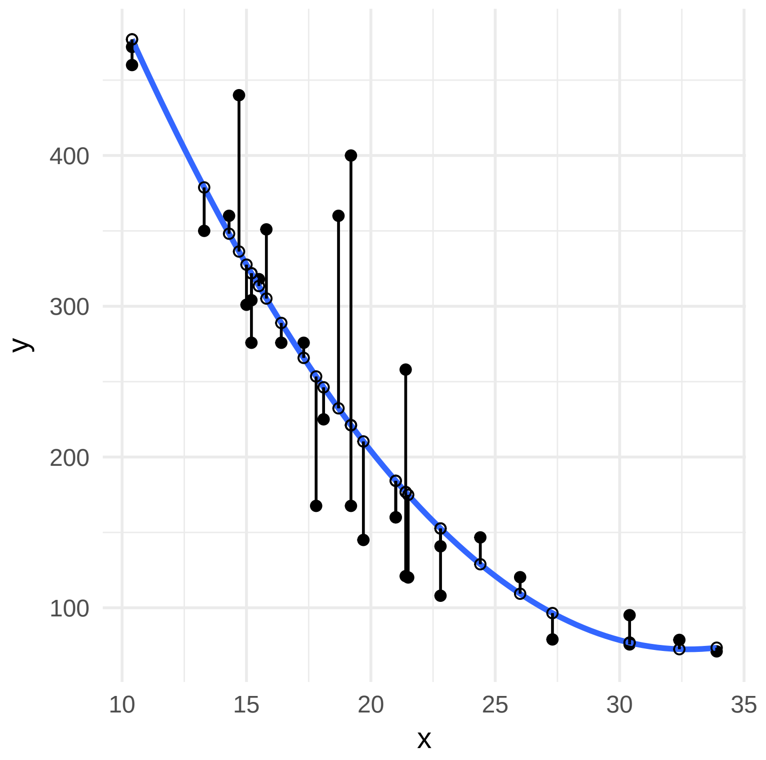

Figure 2.6: Residuals of the quadratic fit

while most commonly used, suffers from several downsides. Most notably, since it relies on squared error, it is not robust. Like the mathematical mean, a few extreme outliers can strongly influence the result in unwanted ways. The reason it is most commonly used is because it's easily differentiably, thus allowing for stable solutios. In other words, since the regression parameters are continous functions of the data, a small change in the data will always produce only a small change in the regression line.

2.4.1 ℓ_2

While ℓ_1 is the most commonly used loss function, it has its downsides, most notably it is not robust (i.e., it is sensitive to outliers). What would be an alternative loss function?

In regression the most simplest loss function would be the residual itself:

$$r = y - \hat{y} \tag{2.16}$$

The loss function which we would then want to minimize would be what is commonly called ℓ_2 regularlization or the Laplace loss function, and it takes the following form

$$S = \sum_{i=1}^{n} |r_i| \tag{2.17}$$

The sum of least absolute errors has the benefit of being robust: like the mathematical median, it is not easily influenced by outliers in the data. However, it is not as easily differentiable as ℓ_1, and it is not stable: Small changes in the data can move the regression solution to a configuration where minimizing ℓ_2 can produces more than one solution or has a vastly different slope.

2.4.2 Huber Loss function

The Huber loss function seeks to combine the benefits of both ℓ_1 and ℓ_2 loss functions. Presented by Huber (1964), the loss function is defined as follows:

$$S = \sum_{i=1}^{n} \begin{cases} r^2 & \text{for } |r| \leq \delta \\ |r| & \text{otherwise} \end{cases} \tag{2.18}$$

where δ is defined as a small non-zero value. Using the Huber loss function, we get a quadratic for small residuals so that the function is differentiable and stable, but for larger residuals we get a linear function so that the loss function is robust and not sensitive to outliers.

A variant of the Huber loss function is the pseudo-Huber loss function which is a smooth approximation of Huber, and it ensures continous derivatives for all degrees:

$$S = \sum_{i=1}^{n} \delta^2 (\sqrt{1 + (r/\delta)^2} - 1) \tag{2.19}$$

This function approximates $r^2/2$ for small values of r but it approaches a straight line with slope δ as values of r approach infinity.

3 Classification

So far we have only examined statistical models that give numerical outcomes. What about predicting categorical or qualitative outcomes, or as they are often called, classes? Such types of problems are called classification problems and there are several statistical methods that deal with such type of problems. As an example, using data once again from Henderson and Velleman (1981), suppose we have two groups in our data (figure 3.1), can we predict the class of each data point using predictors such as x and y in this case? A theoretical classifier might find a division between the two groups such as the one in figure 3.2, where each data point that is above the line is in group a while each data point below it is in group b.

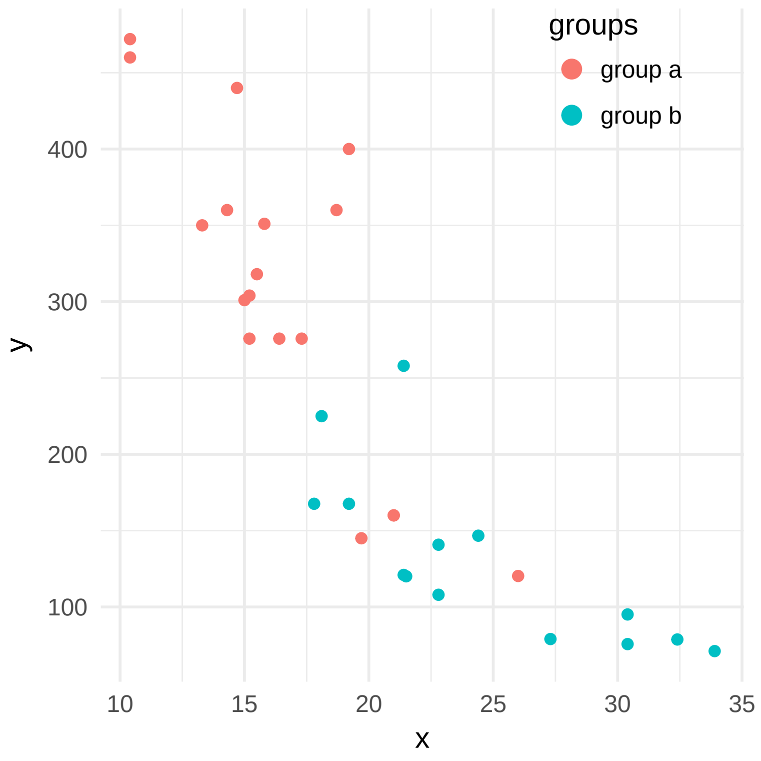

Figure 3.1: The raw data

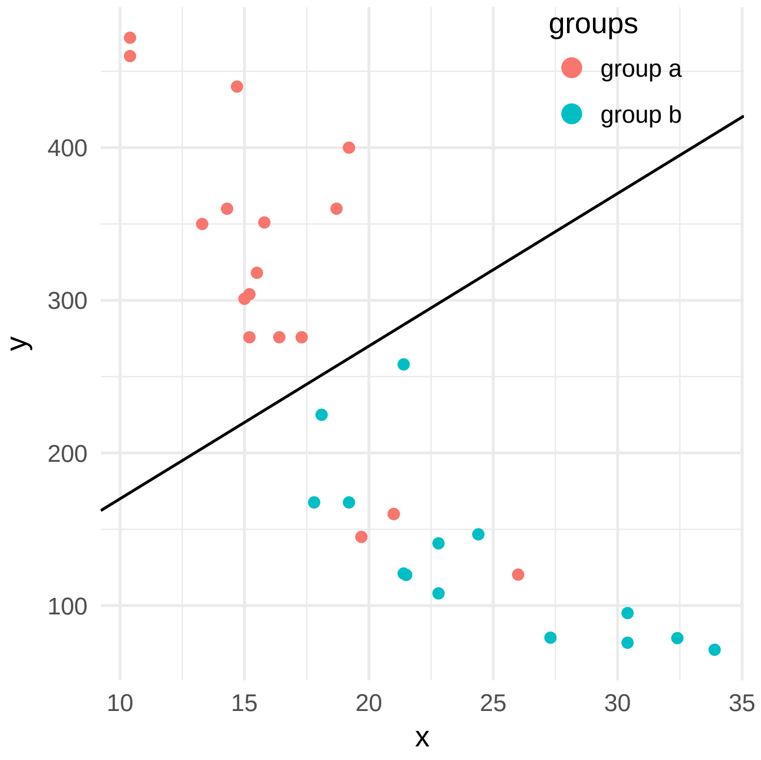

Figure 3.2: A theoretical classification

3.1 Logistic Regression

One of the most popular methods for classification is logistic regression. The biggest obstacle in applying regression to categorical outcomes is in how to encode the different classes in the mathematical formula. Many times there is no inherent ordering to the different classes (for example, genders have no inherent ordering and it makes as much sense to encode male as 1 and female as 0 as the opposite), and even when there is it is not necessarily true that the classes are equally distant from each other. Even in the case of numerical outcomes, such as age, we might not be able to use regression if the outcome does not make sense. For example, if we're trying to predict age, what does it mean if the intercept is a negative value? Or how do we interpert a prediction of an unreasonably old age?

Therefore, instead of predicting the value itself, logistic regression predicts probabilities of classes. In the case of a binary class where the question is simply whether a data point falls in a class or not, we can model the regression to output a value between 0 and 1, where anything below .5 will be interpreted to mean that the data point does not fall in the class and everything above it will be interpreted as belonging to the class, or vice versa. Obviosuly, given that we're interpreting the output values as probabilities, the closer they are to 0 or 1, the more certain we are with the classification.

3.1.1 Simple Logistic Regression

To transform our categorical results, we use the logistic function, also known as the sigmoid curve because of its S shape (Verhulst 1820):

$$\begin{aligned} f(x) &= \frac{1}{1+x^{-x}} \\ &= \frac{e^x}{1+e^x} \end{aligned} \tag{3.1}$$

How do we use this for modeling the probabilites of a binary outcome? Let's first consider the simple case of one predictor. Replacing x in the logistic function with our simple linear regression equation $\beta_0 + \beta_1 x$ we get:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \tag{3.2}$$

While this may not look like it's linear, a simple algebraic manipulation can lead us to an equation that is linear in X:

$$
\begin{aligned}
p(X) &= \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \\
\frac{p(X)}{1-p(X)} &= e^{\beta_0+\beta_1 X} \\
log\left(\frac{p(X)}{1-p(X)}\right) &= \beta_0+\beta_1 X
\end{aligned}
\tag{3.3}
$$

The left hand side of the final form in formula (3.3) is often referred to as the log-odds since it is the log of the probability odds $(p(x)/1-p(x))$.

Now that we've transformed the regression equation to give output values between 0 and 1, we need to define a new loss function to optimize for. While we could use the least squares loss to fit the model, that would be non-linear. A more general loss function used for logistic regression, is called the maximum likelihood, and is defined as follows:

$$
\ell(\beta_0,\beta_1) = \prod_{i:y_1=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'})) \tag{3.4}
$$

We choose β_0 and β_1 so that they maximize the likelihood function.

3.1.2 Multiple Logistic Regression

So far we have only considered a logistic regression with one predictor. What about multiple predictors? Just like with linear regression, we can easily generalize our logistic regression formula ((3.3)) to multiple predictors, as follows:

$$
log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0+\beta_1 X_1+\cdots+\beta_k X_k \tag{3.5}
$$

where $X = X_1, \dots X_k$ are k predictors. Like in the previous section, we use the maximum likelihood to estimate $\beta_0, \beta_1, \dots, \beta_k$.

Let's return to our previous example (figure 3.1). Using logistic regression, we can correctly predict the class of all but 6 data points using .5 as the cutoff between the two classes (figure 3.3).

While an outcome of .5 might be the most natural default cutoff between the two classes, it is not always the right one. For example, looking at the probabilites of our example above (3.4), we can clearly see that a lower value (perhaps .25) might be a better cutoff, giving us less error. On the other hand, such an approach might lead to overfitting the model to our training data.

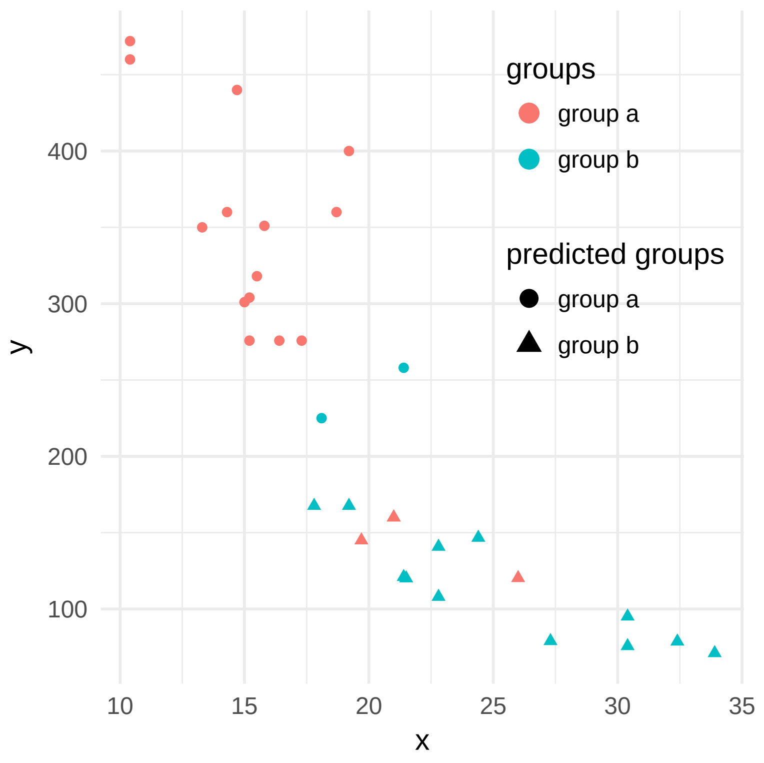

Figure 3.3: The true data classes vs. the predicted data classes

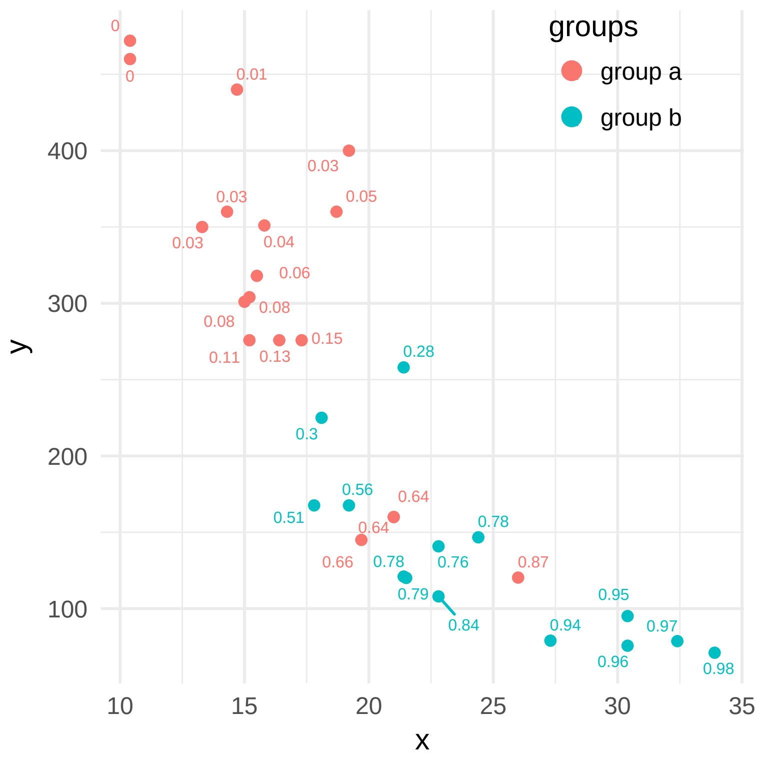

Figure 3.4: The probabilites of each point given by logistic regression

3.1.3 Categorical Predictors

Like with linear regression (section 2.2.1), categorical predictors can be used with logistic regression by using dummy variables to represent the categorical classes. The only difference is that the dummy variables won't change the predicted outcome itself, but the probability of the predicted outcome. The same is true when we interact the dummy variables with other (numerical) predictors. The effect of such interaction is changing the probability associated with the interacted predictor.

3.1.4 Multinomial Classes

So far we have only discussed logistic regression in a situation where our outcome variable can only take two values or classes. But what about multiclass problems? multinomial logistic regression is not often used to predict more than a binary classification because much better methods are available. Nevertheless, generalizations to the logistic regression exist which allow using it for predicting more than two classes.

The most simple generalization is to employ a technique similar to how we encode multinomial classes as predictors (see section 2.2.1). For k classes, we can run $k-1$ independent binary logistic regressions, where one outcome is used as a pivot against which all other classes are regressed. For example, if we choose outcome k as our pivot, our set of logistic regression will take on the following form:

$$\begin{gathered} log\left(\frac{p(\text{class 1})}{p(\text{class k})}\right) = B_1 X_i \\ log\left(\frac{p(\text{class 2})}{p(\text{class k})}\right) = B_2 X_i \\ \dots \\ log\left(\frac{p(\text{class k-1})}{p(\text{class k})}\right) = B_{k-1} X_i \end{gathered} \tag{3.6}$$

We can then find the probabilites for class k, using the fact that the sum of all probabilites need to sum to 1:

$$p(k) = \frac{1}{1 + \sum_{k=1}^{k-1} e^{\beta_k X_i}} \tag{3.7}$$

and for all other classes $1, 2, ..., k-1$ we can use the followin formula:

$$
\begin{aligned}
p(1) &= \frac{e^{\beta_1 X_i}}{1 + \sum_{k=1}^{k-1} e^{\beta_k X_i}} \\
p(2) &= \frac{e^{\beta_2 X_i}}{1 + \sum_{k=1}^{k-1} e^{\beta_k X_i}} \\
&\dots \\
p(k-1) &= \frac{e^{\beta_{k-1} X_i}}{1 + \sum_{k=1}^{k-1} e^{\beta_k X_i}}
\end{aligned}
\tag{3.8}
$$

3.2 Naive Bayes

Naive Bayes is a simple classification method that applies Bayes' theorem to the problem of classification. Recall that Bayes' theorem states as follows:

Theorem 3.1 (Bayes' theorem)**.**

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

With some basic algebra it can be shown that $P(A \cap B) = P(A|B)P(B)$. Since we know that $P(A \cap B) = P(B \cap A)$, we can restate Bayes' theorem as in formula (3.9).

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)} \tag{3.9}$$

If we know the prior probability of specific features given a classification, we can then estimate the probability of an observation's classification given the existence of those specific features.

As an example, suppose we're trying to find the probability that an incoming email is spam given that it contains the word "cash." If we know the prior probability of the word "cash" appearing in emails we have classified as spam, we can know calculate the probability of a new email being spam given that it contains the word "cash." In mathematical notation $P(\text{spam}|\text{cash}) = \frac{P(\text{cash}|\text{spam})P(\text{spam})}{P(\text{cash})}$.

The Naive Bayes classifier (formula (3.10)) takes advantage of our ability to compute prior probabilities, to estimate posterior probabilities for classification purposes. It's naive because it makes a naive assumption about the data it is fed, namely, that all of the features in the data are equally important and that they are all independent of each other. In our spam example, we will consider each word as equally indicative of spam or not spam and as independent of other words. While this assumptions are clearly not true in most cases, Naive Bayes still gives surprisingly well results.

$$P(C_L|F_1, F_2, ..., F_n) = \frac{1}{Z} p(C_L) \prod_{i=1}^{n} p(F_i|C_L) \tag{3.10}$$

3.3 Linear Discriminant Analysis

While logistic regression is great for predicting binary classes, it's harder to use and harder to interpert for multinomial classes. Logistic regression also suffers from other downsides among them the fact that if the classes are separated, the estimates for the predictor coefficients can be unstable. Linear discriminant analysis can be used to overcome these issues.

Suppose we have observation that we want to classify into one of K classes.

The basis for LDA is Bayes' theorem.

3.4 K-Nearest Neighbors

K-nearest neighbors (k-NN in short) is a non-parametric classification method. k-NN classifiers use measures of distance between data points to assign them a class. By calculating the distance of a point from its k closest neighbors, we can estimate the class of the point.

3.4.1 Distance functions

For continuous variables, there are several distance functions. The most commonly used (Hu et al. 2016) is the Euclidean distance function (formula (3.11)), which gives the actual distance between two points in Euclidean space "as the crow flies".

$$\begin{aligned} D(x, x') &= \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \cdots + (x_n - x'_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2} \end{aligned} \tag{3.11}$$

The Euclidean distance function does not come without its downsides, however. One downside is that it's very sensitive to extreme differences in a single feature. So if one attribute (out of potentially many) of a neighboring point is larger, it will throw off the classification accuracy.

Another commonly used distance function is the Manhattan distance function (formula (3.12)), so called because it only uses horizontal and vertical distances between points. This alleviates the above mentioned downside to the Euclidean distance function because extreme differences in one attribute are not squared.

$$\begin{aligned} D(x, x') &= |x_1 - x'_1| + |x_2 - x'_2| + \cdots + |x_n - x'_n| \\ &= \sum_{i=1}^{n} |x_i - x'_i| \end{aligned} \tag{3.12}$$

These distance functions can be generalized to what is known as the Malinowski distance function (formula (3.13). When $q = 2$ the Malinowski distance is exactly equal to the Euclidean distance, and when $q = 1$ it is equal to the Manhattan distance. But you can also set q to other values depending on how much extreme differences in single features need to be penalized.

$$\begin{aligned} D(x, x') &= \sqrt[q]{(|x_1 - x'_1|)^q + (|x_2 - x'_2|)^q + \cdots + (|x_n - x'_n|)^q} \\ &= \sum_{i=1}^{n} \sqrt[q]{(|x_i - x'_i|)^q} \end{aligned} \tag{3.13}$$

For discrete and categorical variables, the Euclidean distance and similar continuous distance functions are not at all informative, especially when there's no inherent ordering. In such cases, the Hamming distance (formula (3.14)) can be used. It should also be noted that the Malinowski distance function (formula (3.13)) is equal to the Hamming distance when $q = 0$.

$$D(x, x') = \sum_{i=1}^{n} \begin{cases} 0 & \text{if } x_i \neq x'_i \\ 1 & \text{if } x_i = x'_i \end{cases} \tag{3.14}$$

3.4.2 The classifier

The simplest k-NN classifier is when $k = 1$, i.e. the 1-nearest neighbor classifier. Using the 1-NN, the class of a point x is simply the class of its closest neighbor in the feature space as calculated by the appropriate distance function.

When $k = n$, the k-NN classifier devolves into a simple majority vote; each new data point will be assigned the class of the majority of already classified data points. To choose an appropriate k various optimization techniques can be employed.

If the data is highly dimensional (with number of dimensions, or features, greater than ~10), dimension reduction can be done to lower the number of dimensions.

4 Clustering

4.1 K-Means

5 Regularlization

Over-fitting is a problem. regularization penalizes biggest predictors. “Regularization can be accomplished by restricting the hypothesis space $\mathcal{H}$”

5.1 Tikhonov regularization (Ridge Regression)

5.2 Lasso Regression

5.3 The Curse of Dimensionality

5.4 Principal Components

Dimensionality reduction

6 Reinforcment Learning

So far, the statistical learning models we have discusses are used to either predict a numerical outcome or a categorical class. But there exists a third type of prediction that can not be modeled as a single predicted outcome, but rather as a series of steps that have to be taken to reach a specific goal.

The classical example of such type of problem is the multi-armed bandit problem, as presented by Gittins (1989): a gambler with a selection of many slot machines to play has to decide which slot machine to choose, how many times to play each machine, and in which order, based on each machine’s individual probability distribution. Put simply, the gambler needs to optimize the reward earned by the pulling of levers of the slot machines.

Gittins (1989) presented a theorem, called the Gittins Index, which gives an optimal policy for the gambler to follow, which will maximize the reward. Ultimately, the gambler can play again and again using the information of wins and loses to decide which machine to play next. But so-called “exploitation” of the machine with the highest winning probabilities has a downside, exploration of the probabilities of other machines falls. The problem that reinforcement learning deals with is to optimize the tradeoff between exploration and exploitation.

7 Deep Learning

References

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Gauss, Carl Friedrich. n.d. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum*.

Gittins, J. C. 1989. *Multi-Armed Bandit Allocation Indices*. Wiley, Chichester, NY.

Henderson, Harold V., and Paul F. Velleman. 1981. "Building Multiple Regression Models Interactively." *Biometrics* 37 (2). [Wiley, International Biometric Society]: 391–411. http://www.jstor.org/stable/2530428.

Hu, Li-Yu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai. 2016. "The Distance Function Effect on K-Nearest Neighbor Classification for Medical Datasets." *SpringerPlus* 5 (1). doi:10.1186/s40064-016-2941-7.

Huber, Peter J. 1964. "Robust Estimation of a Location Parameter." *The Annals of Mathematical Statistics* 35 (1). Institute of Mathematical Statistics: 73–101. http://www.jstor.org/stable/2238020.

Kopf, Dan. 2015. "The Discovery of Statistical Regression." *Priceonomics*. https://priceonomics.com/the-discovery-of-statistical-regression/.

Legendre, Adrien-Marie. 1805. *Nouvelles Méthodes Pour La Détermination Des Orbites Des Comètes*. Firmin Didot, Paris, France.

Verhulst, Pierre-François. 1820. "Recherches Mathématiques Sur La Loi d'accroissement de La Population." *Nouveaux Mémoires de L'Académie Royale Des Sciences et Belles-Lettres de Bruxelles* 18: 1–42.