Chapter 4 Classification:

Often target is categorical / qualitative variable. Classification methods to be used to predict the label. Sometimes these methods can give probability of each class and thus similar to regression that can take values between 0 to 1.
This chapter: Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors

Why not Linear Regression for predicting classes?

Can be explained by taking a target variable with three categories. Assigning values 1,2,3 for 3 categories a,b,c would mean a relationship between them in case of regression. A linreg model would take c as 3 times the magnitude of a or +2 from a. In reality, a label of 2,3,1 for a,b,c is just as fine as 1,2,3. But linreg will interpret it differently and fit a different model, thus giving different prediction. But remember, if target had a natural ordering like mild,severe, more severe and coding was 1,2,3 for these, then linreg might be a good fit sometimes. Most of the times it is not.

If there are two classes, coded as 0 and 1, linear regression can still be used, as finding the output and then predicting as 0 if output < 0.5 , and 1 if output >= 0.5. If the values lie outside [0,1] it is hard to interpret them as probabilities, but still remember, this is possible to use. Curiously it turns out to be same classification as given by LDA (revisited later)

4.3 Logistic Regression

4.3.1 The Logistic Model

How should we model the relationship between $p(X) = \Pr(Y = 1|X)$ and X? (For convenience we are using the generic 0/1 coding for the response). In Section 4.2 we talked of using a linear regression model to represent these probabilities:

$$p(X) = \beta_0 + \beta_1 X. \quad (4.1)$$

But when the right hand side of 4.1 is negative or very high, it is not a probability.

To avoid this problem, we must model $p(X)$ using a function that gives outputs between 0 and 1 for all values of X. Many functions meet this description. In logistic regression, we use the *logistic function*,

logistic function

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \quad (4.2)$$

We use “maximum likelihood” method to fit this model.
The function in 4.1 gives a straight line and that in 4.2 S-shaped. Both give average probability of fitted data same (meaning Expected p(X) in statistics language), but 4.2 is more meaningful.

After a bit of manipulation of (4.2), we find that

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}. \tag{4.3}$$

The quantity $p(X)/[1-p(X)]$ is called the *odds*, and can take on any value between 0 and ∞. Values of the odds close to 0 and ∞ indicate very low and very high probabilities of default, respectively. For example, on average

odds

1 in 5 people would default with an odds of ¼ for p(X) = 0.2

By taking the logarithm of both sides of (4.3), we arrive at

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X. \tag{4.4}$$

The left-hand side is called the *log-odds* or *logit*. We see that the logistic regression model (4.2) has a logit that is linear in X.

log-odds
logit

Increase of one unit in X increases log odds by beta1, or odds by exp^(beta1)
But change in p(X) can not be quantified in terms of beta1, because it depends on current value of X as well.

4.3.2 Estimating the Regression Coefficients

default status. In other words, we try to find $\hat{\beta}_0$ and $\hat{\beta}_1$ such that plugging these estimates into the model for $p(X)$, given in (4.2), yields a number close to one for all individuals who defaulted, and a number close to zero for all individuals who did not. This intuition can be formalized using a mathematical equation called a *likelihood function*:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'})). \tag{4.5}$$

likelihood function

After this a phenomenon of “confounding” is discussed in the book. How the trend reverses from simple logistic regression to multiple regression. For example, a student individually may be riskier when no other information is available (on cc balance, income etc). But he may be less risky with credit card info available. This is because they are correlated.

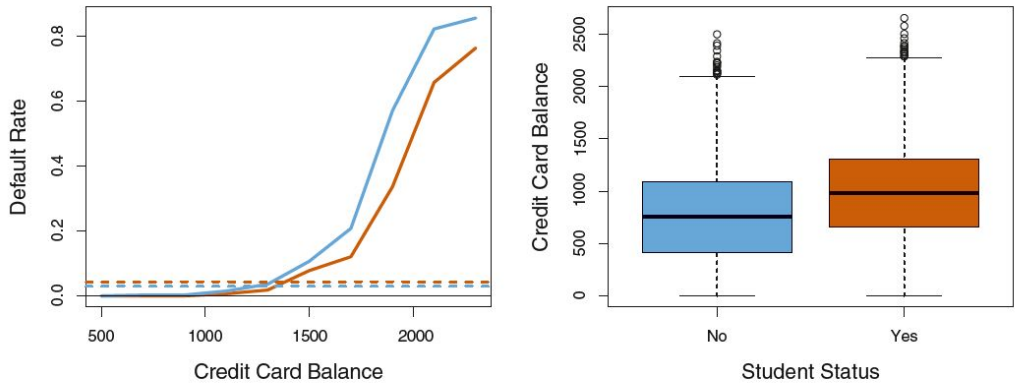

FIGURE 4.3. *Confounding in the* `Default` *data.* Left: *Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of* `balance`*, while the horizontal broken lines display the overall default rates.* Right: *Boxplots of* `balance` *for students (orange) and non-students (blue) are shown.*

Logistic regression for multi class is doing several one vs all and selecting the one with highest probability. (not in the book, but from Prof Andrew's course)

4.4 Linear Discriminant Analysis

In logistic regression, we model conditional response Y given predictors X.
In LDA, we model distributions of each of predictors X against Y and then use Bayes' theorem to flip them around to get P(Y=k|X=x). Why do we need this?

1. When classes are well separated, parameter estimates for LogReg are unstable (but why!) LDA does not suffer from this problem

Why? Because Maximum likelihood estimates do not exist. Sigmoid curve would be like sgn(x) function with a DC at 0.5
https://stats.stackexchange.com/questions/254124/why-does-logistic-regression-become-unstable-when-classes-are-well-separated/254205
log(p/(1-p)) = aX+b, p is 1 when classes are well separated. Hence left hand side is infinity, right hand side should be infinity, coefficients blow up. This is the case even if p is not 1 but close to 1 right? Coefficients become extremely large.

2. If n is small and distribution of X is nearly normal, then LDA is more stable
3. More than 2 classes, LDA is more suitable

Bayes' theorem states

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}. \qquad (4.10)$$

4.4.1: LDA for p = 1

Suppose we assume that $f_k(x)$ is *normal* or *Gaussian*. In the one-dimensional setting, the normal density takes the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right), \tag{4.11}$$

pk(X) is P(Y=k|X)

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}. \tag{4.12}$$

Take log of 4.12, assigning Y to kth class, for which 4.13 is the largest. [the denominator is common for all pk(x) and not to be considered for comparisons]

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \tag{4.13}$$

For 2 classes this translates to

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}. \tag{4.14}$$

In practice, even if X is not drawn from normal distribution, we approximate it with Gaussian.

$$\begin{aligned} \hat{\mu}_k &= \frac{1}{n_k} \sum_{i:y_i=k} x_i \\ \hat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2 \end{aligned} \tag{4.15}$$

n- total number of training observations
nk - training in kth class.

information, LDA estimates π_k using the proportion of the training observations that belong to the kth class. In other words,

$$\hat{\pi}_k = n_k/n. \tag{4.16}$$

The LDA classifier plugs the estimates given in (4.15) and (4.16) into (4.13), and assigns an observation $X = x$ to the class for which

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k) \tag{4.17}$$

is largest. The word *linear* in the classifier's name stems from the fact that the *discriminant functions* $\hat{\delta}_k(x)$ in (4.17) are linear functions of x (as opposed to a more complex function of x).

4.4.3 Linear Discriminant Analysis for $p > 1$

We assume X = (X1,X2,....Xp) are drawn from a multivariate distribution.

$p \times p$ covariance matrix of X. Formally, the multivariate Gaussian density is defined as

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right). \tag{4.18}$$

Choose kth class for which

$$\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log \pi_k \tag{4.19}$$

is largest. This is the vector/matrix version of (4.13).

Accuracy : This is the simplest scoring measure. It calculates the proportion of correctly classified instances.

Accuracy = (TP + TN) / (TP+TN+FP+FN)

Sensitivity (also called *Recall* or *True Positive Rate*): Sensitivity is the proportion of actual positives which are correctly identified as positives by the classifier.

Sensitivity = TP / (TP +FN)

Specificity (also called *True Negative Rate*) : Specificity relates to the classifier's ability to identify negative results. Consider the example of medical test used to identify a certain disease. The specificity of the test is the proportion of patients that do not to have the disease and will successfully test negative for it. In other words:

Specificity: TN / (TN+FP)

Precision: This is a measure of retrieved instances that are relevant. In other words:

Precision: TP/(TP+FP)

Sensitivity and Recall are same for only binary classification.

LDA is trying to approximate the Bayes classifier, which has the lowest total error rate out of all classifiers. Bayes classifier does not look at which class errors come from, its goal is to maximize accuracy. So, LDA in its plain form may not be of much use to say some problem where identifying true positives when positives are minority class is important. Maximizing recall for positive class is most important. Precision is also important to eliminate too many false positive identification.
We modify LDA for imbalanced classes, by adjusting the threshold for posterior probability, P(Y=k|X=x) from 0.5 to 0.2. This always increases the false positives and reduces accuracy.

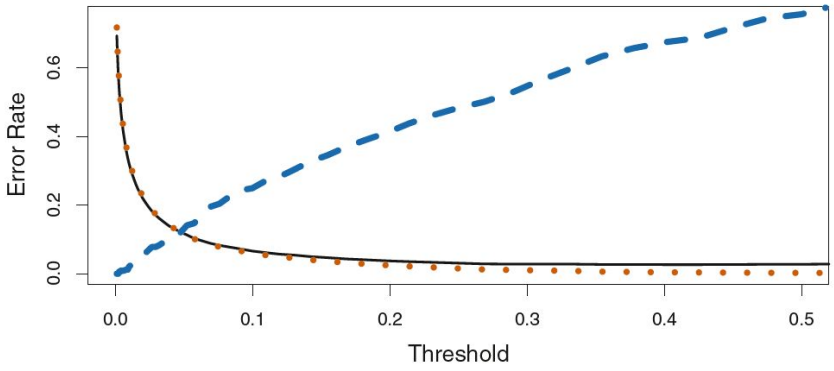

FIGURE 4.7. *For the* `Default` *data set, error rates are shown as a function of the threshold value for the posterior probability that is used to perform the assignment. The black solid line displays the overall error rate. The blue dashed line represents the fraction of defaulting customers that are incorrectly classified, and the orange dotted line indicates the fraction of errors among the non-defaulting customers.*

How should we choose threshold, depends on domain knowledge.
ROC: Receiver Operating Characteristics, comes from communication theory.
It is True Positive Vs False Positive rates for all possible thresholds.
False positive = 1 - specificity. True positive = Sensitivity.

Area Under Curve: One number to compare different classifiers.

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Table 4.7 lists many of the popular performance measures that are used in this context. The denominators for the false positive and true positive rates are the actual population counts in each class. In contrast, the denominators for the positive predictive value and the negative predictive value are the total predicted counts for each class.

A note from SMOTE paper

ations. The Receiver Operating Characteristic (ROC) curve is a standard technique for summarizing classifier performance over a range of tradeoffs between true positive and false positive error rates (Swets, 1988). The Area Under the Curve (AUC) is an accepted traditional performance metric for a ROC curve (Duda, Hart, & Stork, 2001; Bradley, 1997; Lee, 2000). The ROC convex hull can also be used as a robust method of identifying potentially optimal classifiers (Provost & Fawcett, 2001). If a line passes through a point on the convex hull, then there is no other line with the same slope passing through another point with a larger true positive (TP) intercept. Thus, the classifier at that point is optimal under any distribution assumptions in tandem with that slope.

SMOTE seems like a good way to oversample minority class. Not application specific, works in feature space.

4.2 SMOTE

We propose an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement. This approach is inspired by a technique that proved successful in handwritten character recognition (Ha & Bunke, 1997). They created extra training data by performing certain operations on real data. In their case, operations like rotation and skew were natural ways to perturb the training data. We generate synthetic examples in a less application-specific manner, by operating in "feature space" rather than "data space". The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Our implementation currently uses five nearest neighbors. For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.

Multi class classification: Logistic Regression:

One Vs Rest Classifier can miss some points.
The normal Logistic Regression with 3 labels will assign each test X to one of the three.
Confusion matrix, classification reports cant be drawn for One Vs Rest classifiers.
ROC is said to be done with only One Vs Rest for multiclass, in stack overflow.
But I've found that by accessing probabilities of different labels, I get different AUCs

https://www.quora.com/What-is-the-difference-between-a-ROC-curve-and-a-precision-recall-curve-When-should-I-use-each

https://www.quora.com/What-is-the-relationship-between-Accuracy-precision-and-AUC-Area-Under-the-Curve