**Classification:**

In this chapter, we study approaches for predicting qualitative responses, a process that is known as classification .

On the other hand, often the methods used for classification first predict the probability of each of

the categories of a qualitative variable, as the basis for making the classification. In this sense they also behave like regression methods.

Widely used Classifiers, we study in this chapter:

1. ***Logistic Regression***
2. ***KNN***
3. ***Linear discriminant analysis***

Some more computer intensive methods in upcoming chapters:

1. General additive models – Chapter 7
2. Trees – Chapter 8
3. Random Forests - Chapter 8
4. Boosting - Chapter 8
5. Support Vector Machine – Chapter 9

Why Not Linear Regression?

1. Unfortunately, in general there is no natural way to convert a qualitative response variable with more than two levels into a quantitative response that is ready for linear regression.

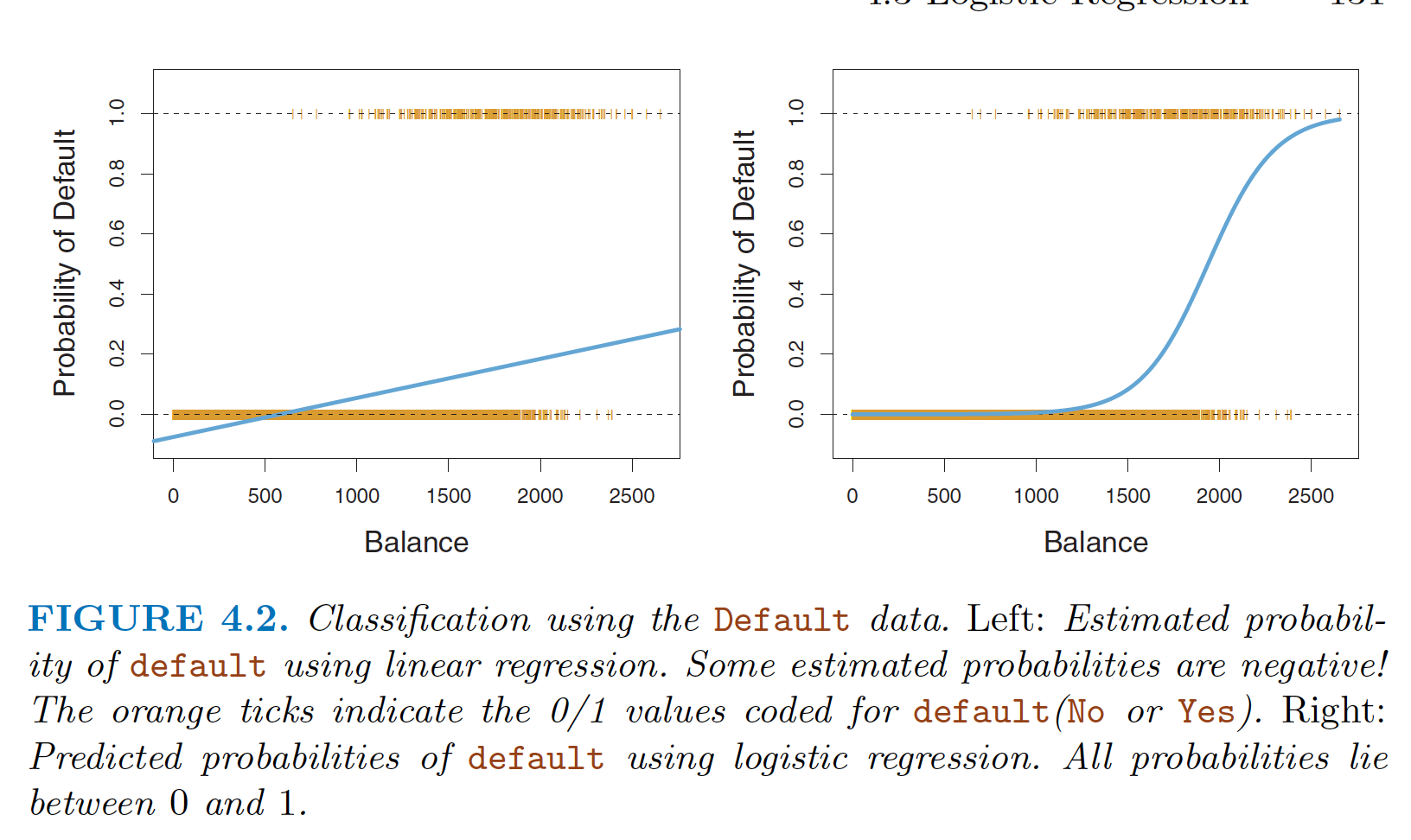
2. For binary variable we can encode Y=0 if Class1 and Y=1 if Class2, We could then fit a linear regression to this binary response, and predict Class2 if ˆY > 0. 5 and Class1 otherwise. In the binary case it is not hard to show that even if we flip the above coding, linear regression will produce the same final predictions.

3. If the response variable’s values did take on a natural ordering, such as mild , moderate , and severe , and we felt the gap between mild and moderate was similar to the gap between moderate and severe, then a 1, 2, 3 coding would be reasonable.

Logistic regression can be seen as a special case of the [generalized linear model](https://en.wikipedia.org/wiki/Generalized_linear_model) and thus analogous to [linear regression](https://en.wikipedia.org/wiki/Linear_regression). The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression. In particular the key differences of these two models can be seen in the following two features of logistic regression. First, the conditional distribution {\displaystyle y\mid x} is a [Bernoulli distribution](https://en.wikipedia.org/wiki/Bernoulli_distribution) rather than a [Gaussian distribution](https://en.wikipedia.org/wiki/Gaussian_distribution), because the dependent variable is binary. Second, the predicted values are probabilities and are therefore restricted to (0,1) through the [logistic distribution function](https://en.wikipedia.org/wiki/Logistic_function) because logistic regression predicts the **probability** of particular outcomes.

Logistic Regression:

Consider again the Default data set, where the response default falls into one of two categories, Yes or No . Rather than modeling this response Y directly, logistic regression models the probability that Y belongs to a particular category.



For the Default data, logistic regression models the probability of default. For example, the probability of default given balance can be written as

Pr(default = Yes|balance )

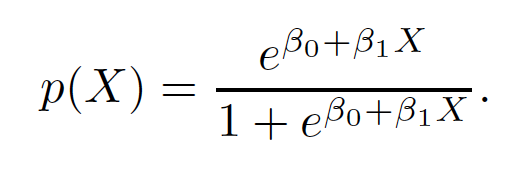
The values of Pr(default = Yes|balance ), which we abbreviate p (balance ), will range between 0 and 1. Then for any given value of balance , a prediction can be made for default . For example, one might predict default = Yes for any individual for whom p (balance ) > 0. 5. Alternatively, if a company wishes to be conservative in predicting individuals who are at risk for default, then they may choose to use a lower threshold, such as p (balance ) > 0. 1.

Lets say we use linear regression here to predict p( X) = Pr( Y = 1| X) with 0/1 coding for the response

p( X) = β0 + β1 X

Any time a straight line is fit to a binary response that is coded as 0 or 1, in principle we can always predict p (X ) < 0 for some values of X and p (X ) > 1 for others (unless the range of X is limited). Which is not sensible at all because output should always in the range [0,1]

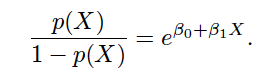
To avoid this problem, we must model p (X ) using a function that gives outputs between 0 and 1 for all values of X . Many functions meet this description. In logistic regression, we use the logistic function



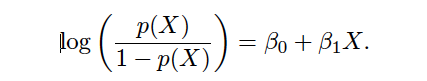
To fit the model, we use a method called maximum likelihood. The logistic function will always produce an S-shaped curve of this form, and so regardless of the value of X , we

will obtain a sensible prediction. We also see that the logistic model is better able to capture the range of probabilities than is the linear regression model in the left-hand plot.

The average fitted probability in both cases is 0.0333 (averaged over the training data), which is the same as the overall proportion of defaulters in the data set.



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 1 in 5 people with an odds of 1/ 4 will default, since p (X ) = 0. 2 implies an odds of 0.2/(1−0.2) = 1/ 4. Likewise on average nine out of every ten people with an odds of 9 will default, since p (X ) = 0. 9 implies an odds of 0.9/(1−0.9) = 9. Odds are traditionally used instead of probabilities in horse-racing, since they relate more naturally to the correct betting strategy. By taking the logarithm of both sides of above eq., we arrive at



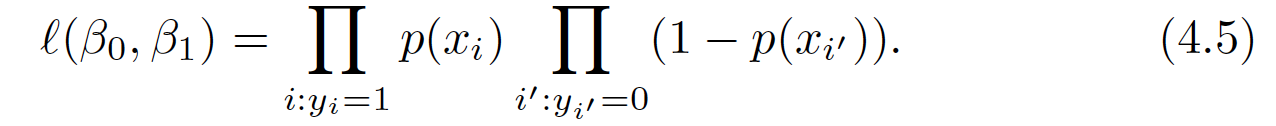
The left-hand side is called the log-odds or logit . We see that the logistic regression model has a logit that is linear in X.

Estimating the Regression Coefficients: coefficients β0 and β1

In Chapter 3, we used the least squares approach to estimate the unknown linear regression coefficients. Although we could use (non-linear) least squares to fit the model (4.4), the more general method of maximum likelihood is preferred, since it has better statistical properties.

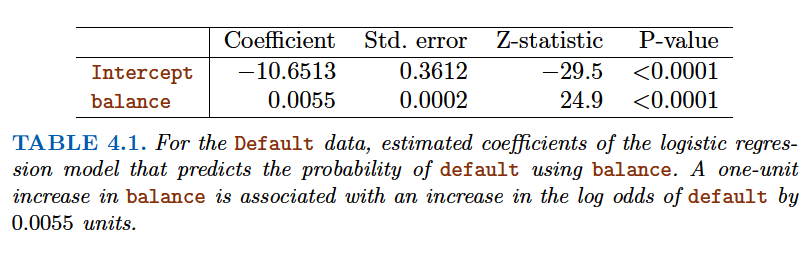
***The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows: we seek estimates for β0 and β1 such that the predicted probability ˆp (xi ) of default for each individual, using (4.2), corresponds as closely as possible to the individual’s observed default status.***

Here, we try to find ˆ β0 and ˆ β1 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 :



The estimates ˆ β0 and ˆβ1 are chosen to maximize this likelihood function.

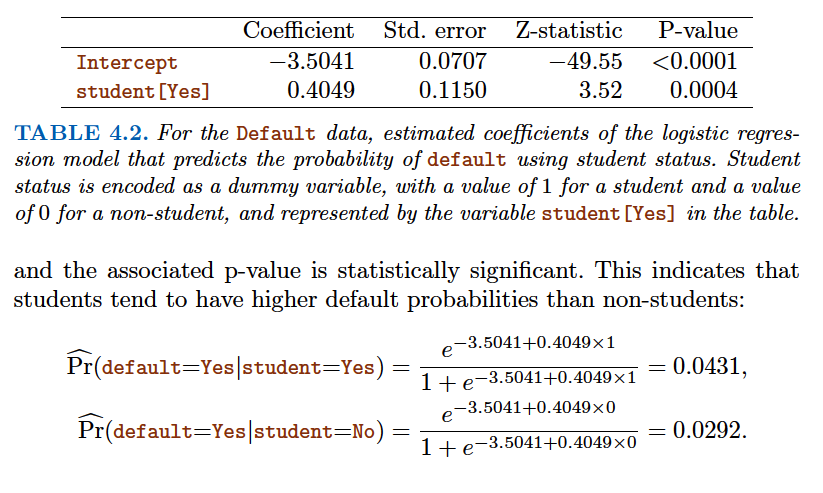
* Maximum likelihood is a very general approach that is used to fit many of the non-linear models that we examine throughout this book.
* In the linear regression setting, the least squares approach is in fact a special case of maximum likelihood.



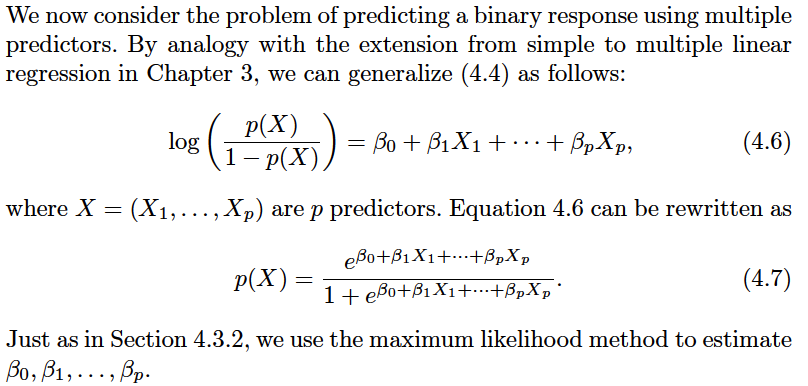
Many aspects of the logistic regression output shown in Table 4.1 are similar to the linear regression output of Chapter 3. For example, we can measure the accuracy of the coefficient estimates by computing their standard errors. The z -statistic in Table 4.1 plays the same role as the t –statistic in the linear regression output. For instance, the z -statistic associated with β1 is equal to ˆ β1/SE ( ˆ β1 ), and so a large (absolute) value of the z -statistic indicates evidence against the null hypothesis H0 : β1 = 0. This null hypothesis implies that p (X ) in other words, that the probability of default does not depend on balance . Since the p-value associated with balance in Table 4.1 is tiny, we can reject H0 . In other words, we conclude that there is indeed an association between

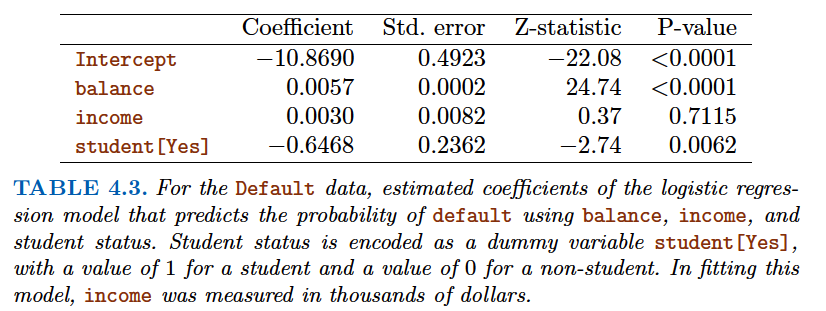
balance and probability of default . The estimated intercept in Table 4.1 is typically not of interest; its main purpose is to adjust the average fitted probabilities to the proportion of ones in the data.

Logistic Regression with Qualitative Variable:

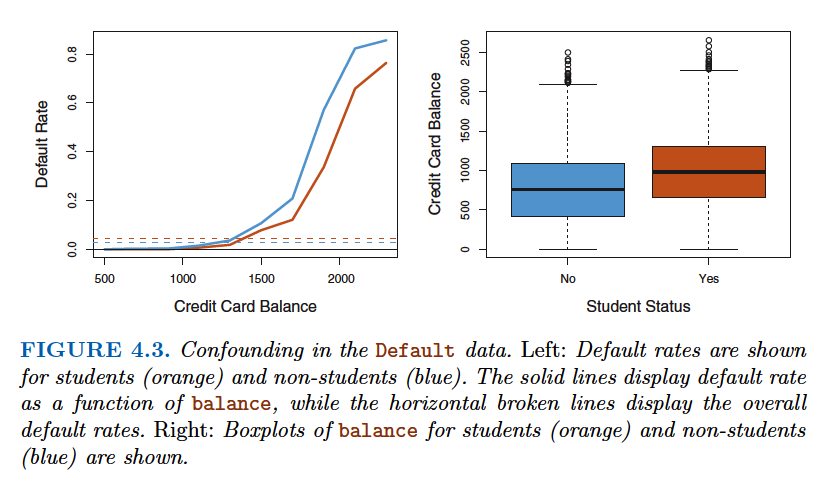


Multiple Logistic Regressions:





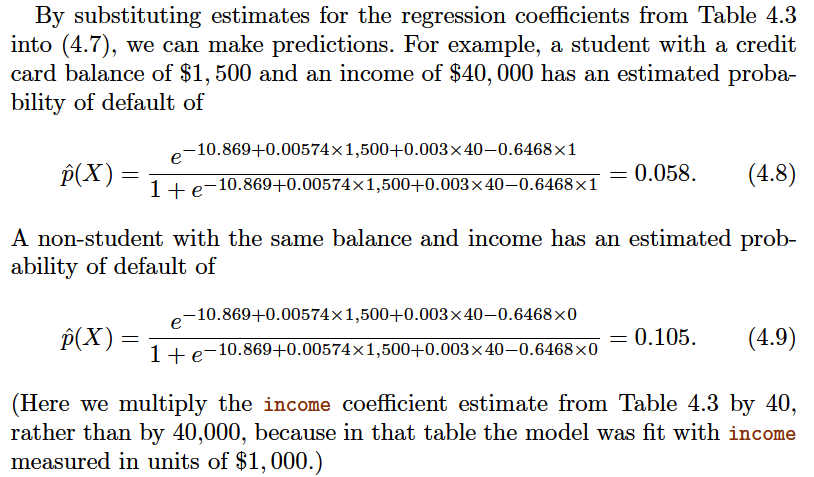
**Confounding Example:**

****

Following simple example illustrates the dangers and subtleties associated with performing regressions involving only a single predictor when other predictors may also be relevant. As in the linear regression setting, the results obtained using one predictor may be quite different from those obtained using multiple predictors, especially when there is correlation among the predictors. In general, the phenomenon seen in Figure 4.3 is known as confounding.

However, the coefficient for the dummy variable (student status) is negative, indicating that students are less likely to default than non-students. In contrast, the coefficient for the dummy variable is positive in Table 4.2. How is it possible for student status to be associated with an increase in probability of default in Table 4.2 and a decrease in probability of default in Table 4.3? The left-hand panel of Figure 4.3 provides a graphical illustration of this apparent paradox. The orange and blue solid lines show the average default rates for students and non-students, respectively, as a function of credit card balance. The negative coefficient for student in the multiple logistic regression indicates that for a fixed value of balance and income , a student is less likely to default than a non-student. Indeed, we observe from the left-hand panel of Figure 4.3 that the student default.

rate is at or below that of the non-student default rate for every value of balance . But the horizontal broken lines near the base of the plot, which show the default rates for students and non-students averaged over all values of balance and income , suggest the opposite effect: the overall student default rate is higher than the non-student default rate. Consequently, there is a positive coefficient for student in the single variable logistic regression output shown in Table 4.2.



Logistic Regression for >2 Response Classes

The two-class logistic regression models discussed in the previous sections have multiple-class extensions, but in practice they tend not to be used all that often. One of the reasons is that the method we discuss in the next section, discriminant analysis , is popular for multiple-class classification. So we do not go into the details of multiple-class logistic regression here, but simply note thatsuch an approach is possible, and that software for it is available in R.

**Linear Discriminant Analysis:**

Logistic regression involves directly modeling Pr(Y = k|X = x ) using the logistic function. Alternative approach, we model the distribution of the predictors X separately in each of the response classes (i.e. given Y ), and then use Bayes’ theorem to flip these around into estimates for Pr(Y = k|X = x ). When these distributions are assumed to be normal, it turns out that the model is very similar in form to logistic regression.

Why do we need another method, when we have logistic regression?

There are several reasons:

• When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.

• If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.

• As mentioned in Section 4.3.5, linear discriminant analysis is popular when we have more than two response classes.

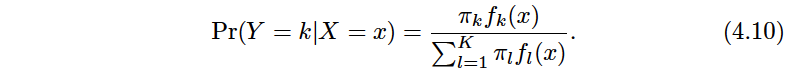
Using Bayes’ Theorem for Classification:

Suppose that we wish to classify an observation into one of K classes, where K ≥ 2. In other words, the qualitative response variable Y can take on K possible distinct and unordered values.

*Prior*: Let πk represent the overall or prior probability that a randomly chosen observation comes from the k th class; this is the probability that a given observation is associated with the k th category of the response variable Y .

*Density Function* : Let fk (X ) ≡ Pr(X = x|Y = k ) denote the density function of X for an observation that comes from the k th class. In other words, fk (x ) is relatively large if there is a high probability that an observation in the k th class has X ≈ x , and fk (x ) is small if it is very unlikely that an observation in the k th class has X ≈ x .

*Then Bayes’ theorem states that*

**

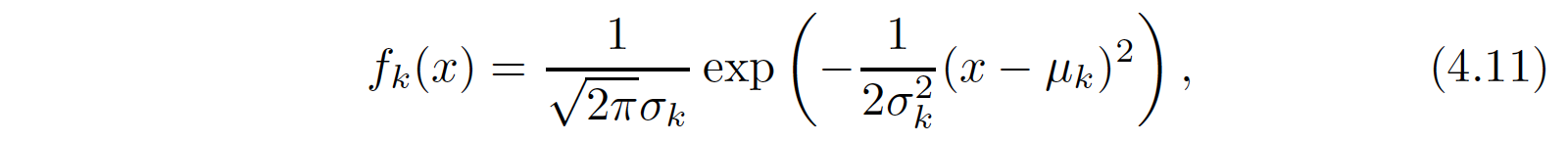
* ***posterior probability pk (x )****:* In accordance with our earlier notation, we will use the abbreviation pk (X ) = Pr(Y = k|X ). We refer to pk (x ) as the posterior probability that an observation X = x belongs to the k th class. That is, it is the probability that the observation belongs to the k th class, given the predictor value for that observation.
* This suggests that instead of directly computing pk (X ) as in Section 4.3.1, we can simply plug in estimates of πk and fk (X ) into(4.10).
* In general, estimating πk is easy if we have a random sample of Y s from the population: we simply compute the fraction of the training observations that belong to the k th class.
* However, estimating fk (X ) tends to be more challenging, unless we assume some simple forms for these densities.

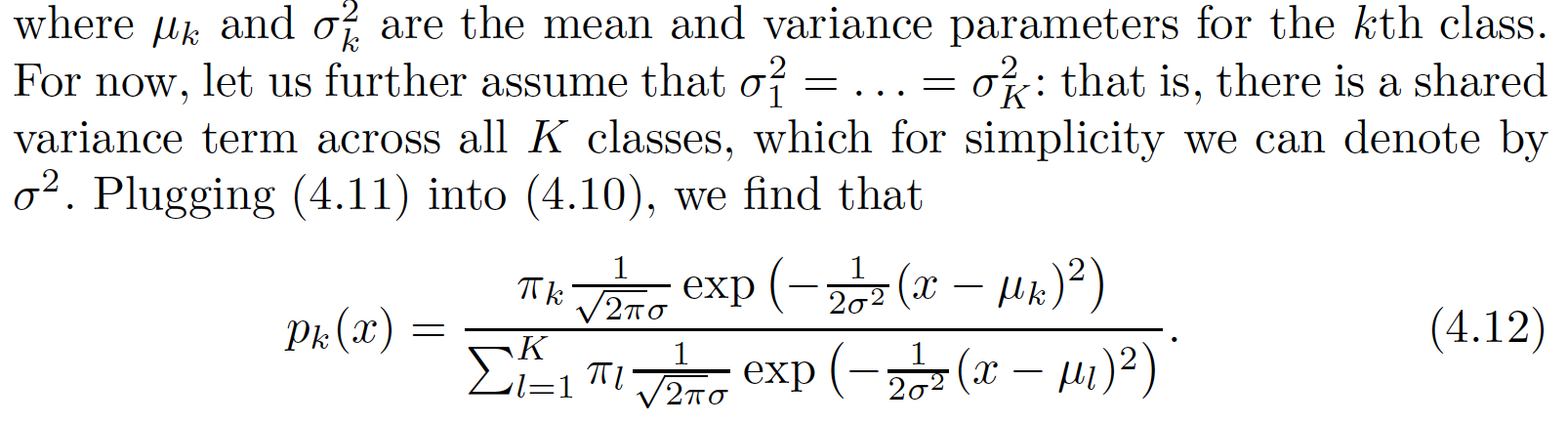
The Bayes classifier, which classifies an observation to the class for which pk (X ) is largest, has the lowest possible error rate out of all classifiers. (This is of course only true if the terms in (4.10) are all correctly specified.) ***Therefore, if we can find a way to estimate fk (X ), then we can develop a classifier that approximates the Bayes classifier.*** Such an approach is the topic of the following sections.

Linear Discriminant Analysis for p = 1

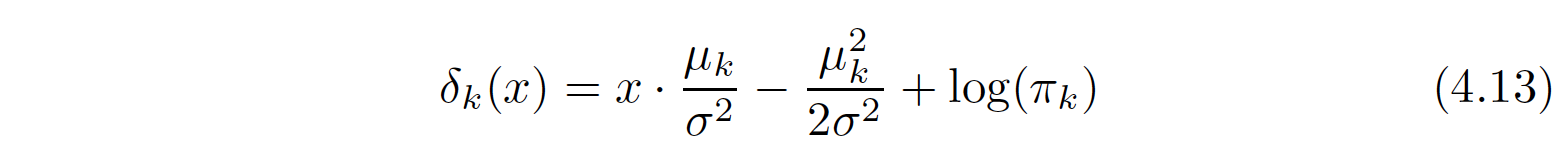
How to estimate fk(x)?

Suppose we assume that fk (x ) is normal or Gaussian . In the dimensional setting, the normal density takes the form

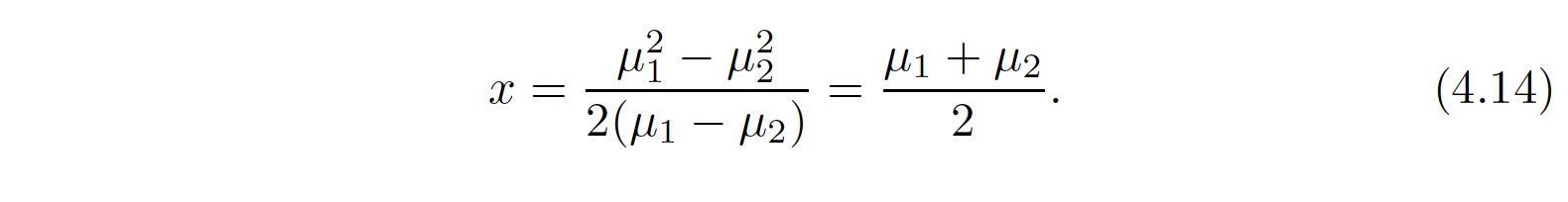




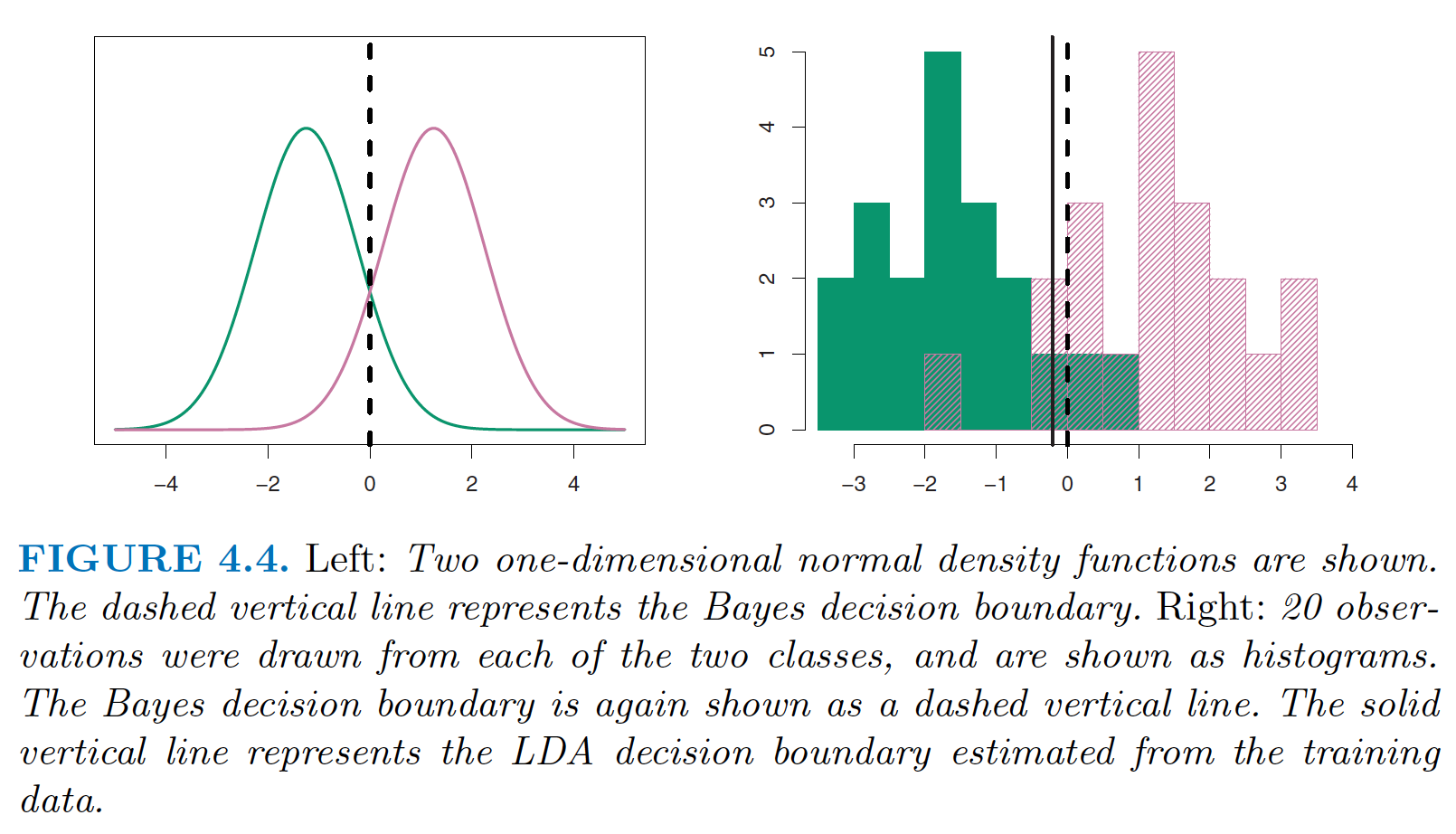
The Bayes classifier involves assigning an observation X = x to the class for which (4.12) is largest. Taking the log of (4.12) and rearranging the terms, it is not hard to show that this is equivalent to assigning the observation to the class for which



is largest. For instance, if K = 2 and π1 = π2 (priors for both the classes are equal), then the Bayes classifier assigns an observation to class 1 if 2x(μ1 – μ2 ) > μ12− μ22, and to class2 otherwise. In this case, the Bayes decision boundary corresponds to thepoint where



An Example: ( In following figure we are comparing Bayes and LDA)



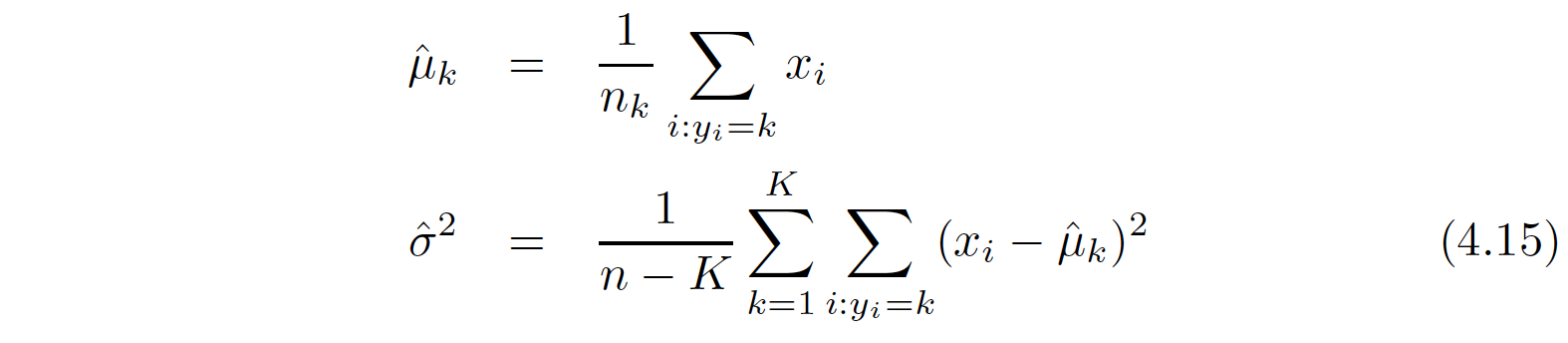
***In a real-life situation, we are not able to calculate the Bayes classifier until we know about distribution of fk(x) and its parameters.***

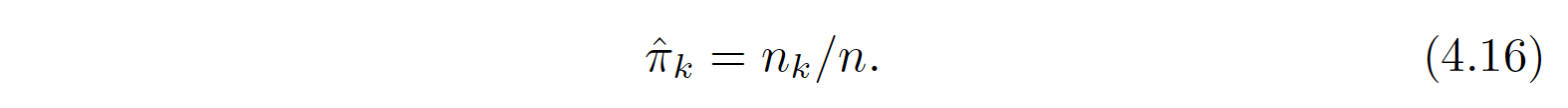
An example is shown in the left-hand panel of Figure 4.4. The two normal density functions that are displayed, f1 (x ) and f2 (x ), represent two distinct classes. The mean and variance parameters for the two density functions are μ1 = − 1. 25, μ2 = 1. 25, and σ12= σ22= 1. The two densities overlap, and so given that X = x , there is some uncertainty about the class to which the observation belongs. If we assume that an observation is equally likely to come from either class—that is, π1 = π2 = 0. 5—then by inspection of (4.14), we see that the Bayes classifier assigns the observation to class 1 if x < 0 and class 2 otherwise. ***Note that in this case, we can compute the Bayes classifier because we know that X is drawn from a Gaussian distribution within each class, and we know all of the parameters involved. In a real-life situation, we are not able to calculate the Bayes classifier.***

In practice, even if we are quite certain of our assumption that X is drawn from a Gaussian distribution within each class, we still have to estimate the parameters μ1, . . . , μK , π1, . . . , πK , and σ2 .

***The linear discriminant analysis (LDA) method approximates the Bayes classifier by plugging estimates for πk , μk , and σ2 into (4.13).*** In particular, the following estimates

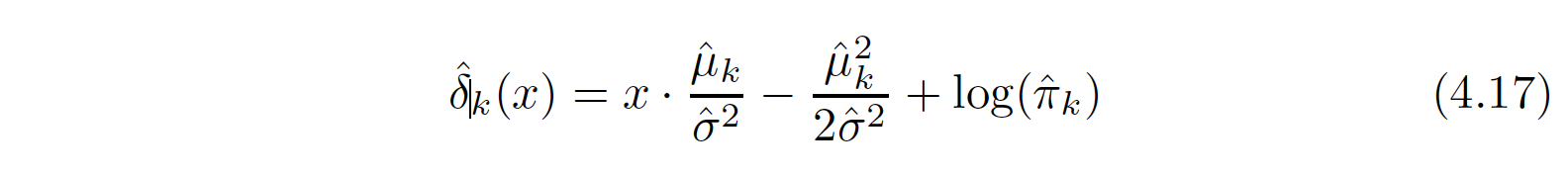
are used:

******

******

where n is the total number of training observations, and nk is the number of training observations in the kth class.

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 ( LDA classifier equation is down, and note that it is linear)



is largest. The word linear in the classifier’s name stems from the fact that the discriminant functions ˆδk (x ) in (4.17) are linear functions of x (as opposed to a more complex function of x ).

The right-hand panel of Figure 4.4 displays a histogram of a random sample of 20 observations from each class.

To implement LDA,

* we began by estimating πk , μk , and σ2 using (4.15) and (4.16).
* We then computed the decision boundary, shown as a black solid line, that results from assigning an observation to the class for which (4.17) is largest.
* All points to the left of this line will be assigned to the green class, while points to the right of this line are assigned to the purple class.

*Comparing Bayes Decision boundary and LDA decision boundary:*

In this case, since n1 = n2 = 20, we have ˆπ1 = ˆπ2 . As a result, the decision boundary corresponds to the midpoint between the sample means for the two classes, (ˆμ1 + ˆμ2 )/ 2. The figure indicates that the LDA decision boundary is slightly to the left of the optimal Bayes decision boundary, which instead equals (μ1 + μ2 )/ 2 = 0. How well does the LDA classifier perform on this data? Since this is simulated data, we can generate a large number of test observations in order to compute the Bayes error rate and the LDA test error rate. These are 10. 6% and 11. 1%, respectively. In other words, the LDA classifier’s error rate is only 0. 5% above the smallest possible error rate! This indicates that LDA is performing pretty well on this data set.

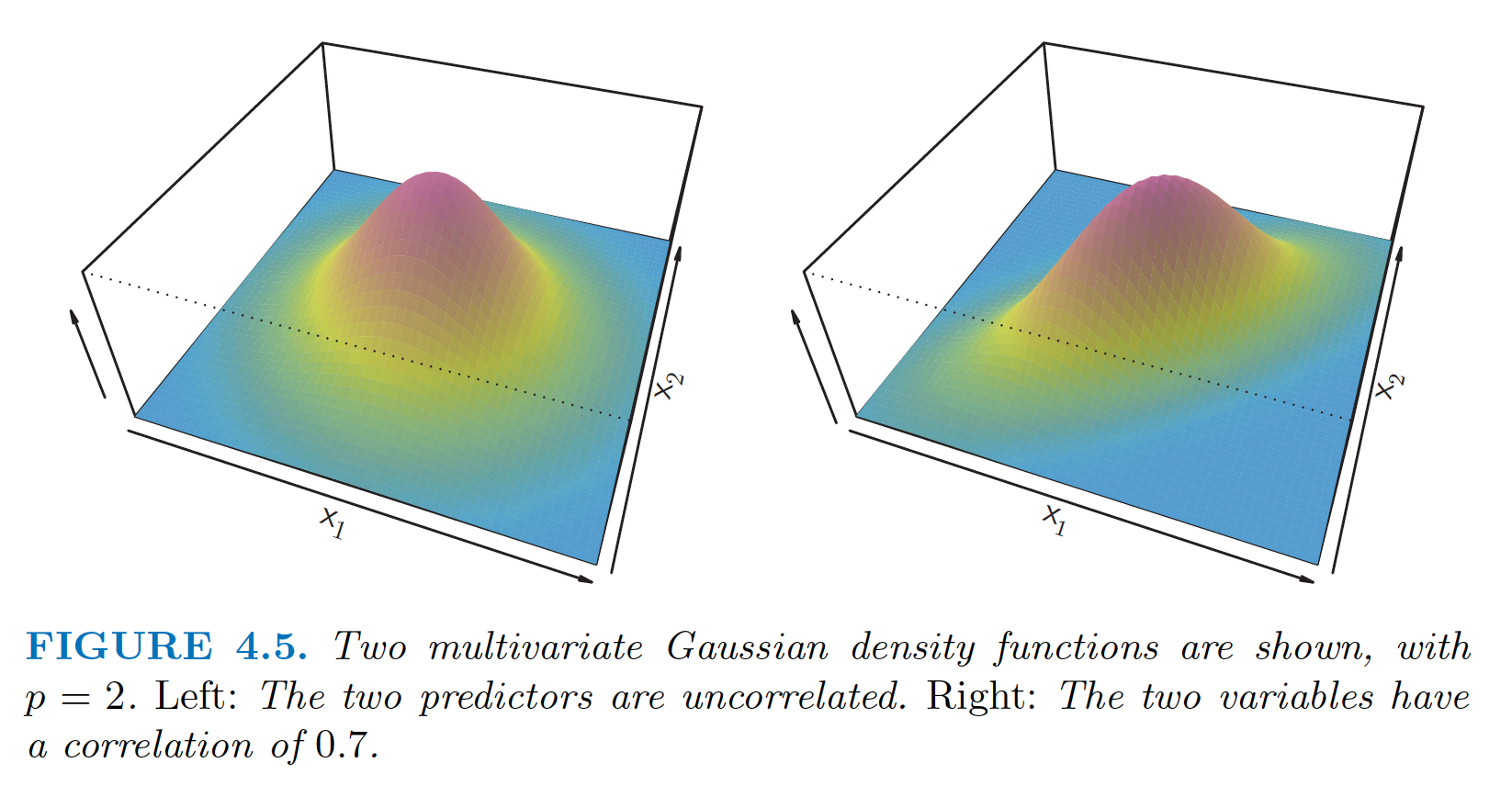
***SUMMARY:***

To reiterate, the LDA classifier results from assuming that the observations within each class come from a normal distribution with a class-specific mean vector and a common variance σ2 , and plugging estimates for these parameters into the Bayes classifier. In Section 4.4.4, we will consider a less stringent set of assumptions, by allowing the observations in the k th class to have a class-specific variance, σk2 .

Linear Discriminant Analysis for p >1

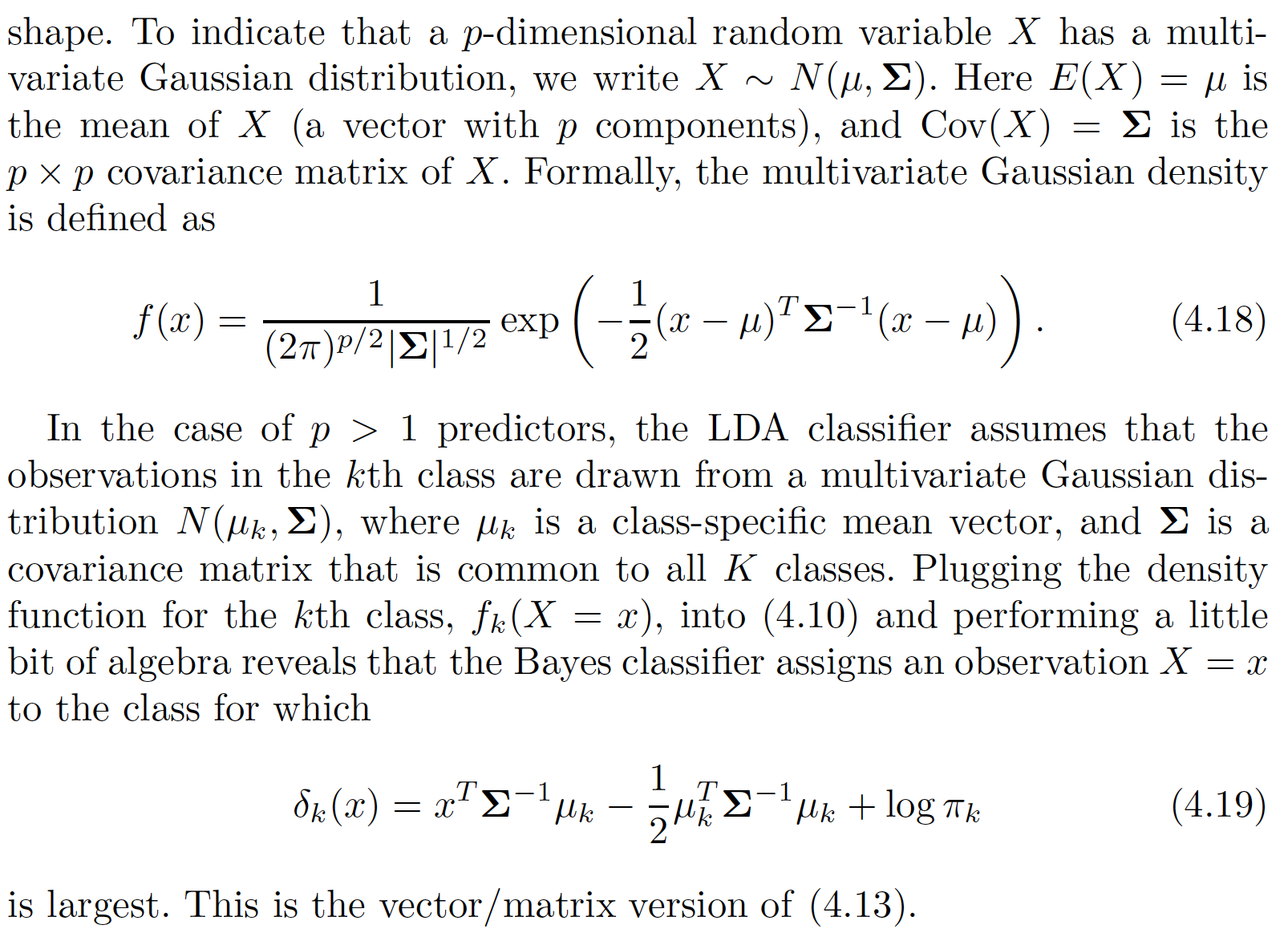
We now extend the LDA classifier to the case of multiple predictors. To do this, we will assume that X = (X1,X2, . . .,Xp ) is drawn from a *multivariate Gaussian (or multivariate normal) distribution*, with a class-specific *mean vector* and a *common covariance matrix*.

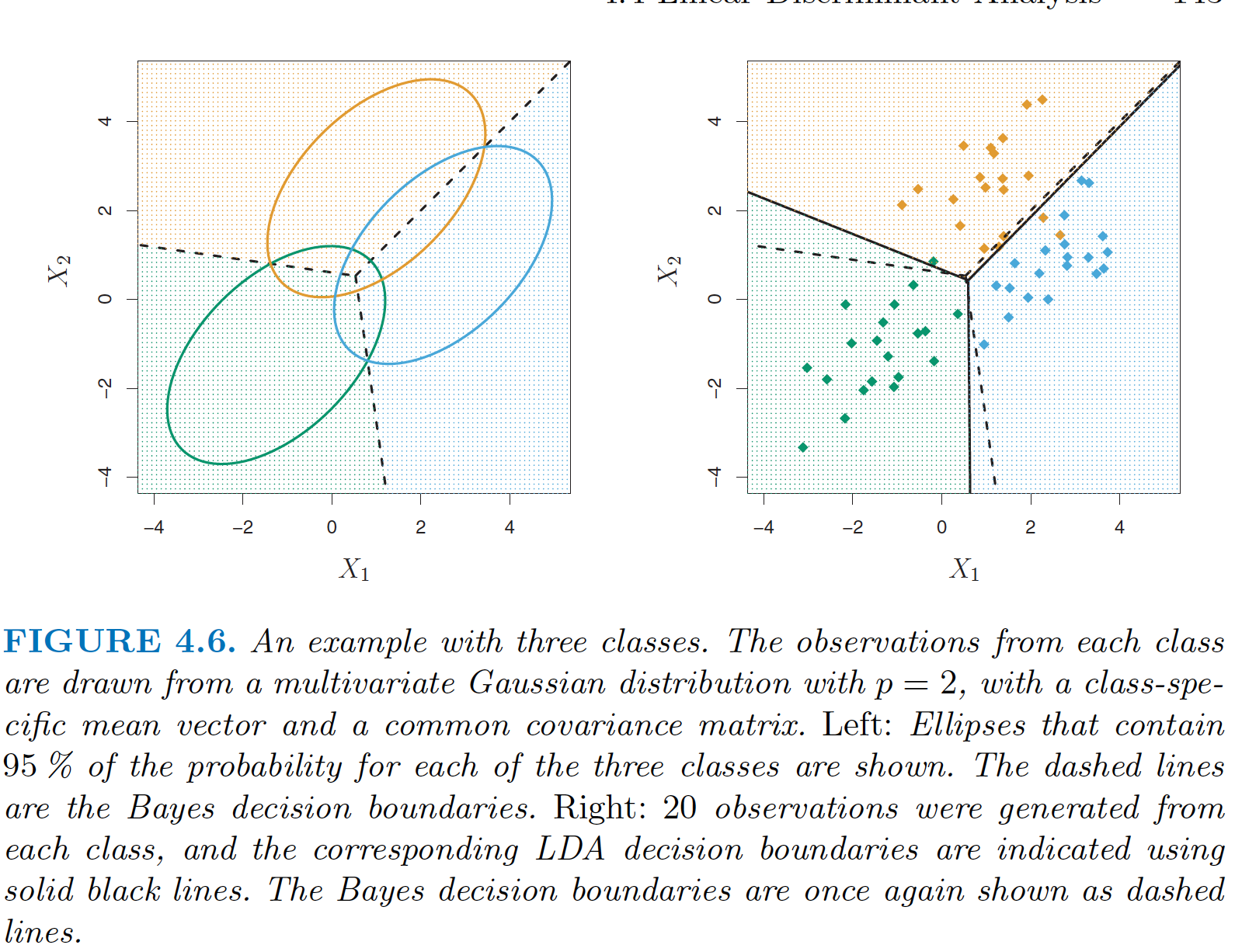
The ***multivariate Gaussian distribution*** assumes that each individual predictor follows a one-dimensional normal distribution, as in (4.11), with some correlation between each pair of predictors. Two examples of multivariate Gaussian distributions with p = 2 are shown in Figure 4.5.



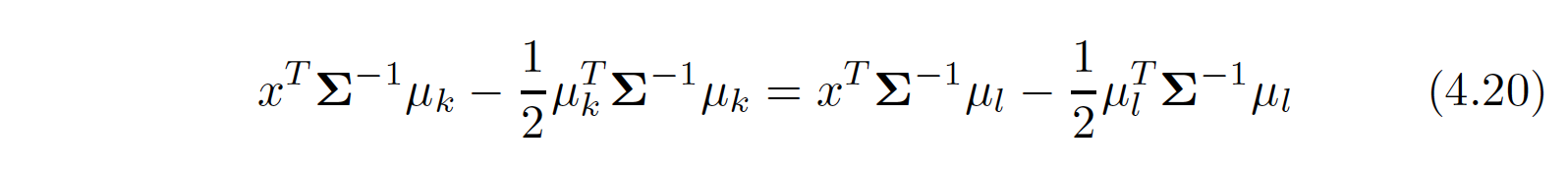
Some Features of Multivariate Guassian Distribution:

* The height of the surface at any particular point represents the probability that both X1 and X2 fall in a small region around that point.
* In either panel, if the surface is cut along the X1 axis or along the X2 axis, the resulting cross-section will have the shape of a one-dimensional normal distribution.
* The left-hand panel of Figure 4.5 illustrates an example in which Var(X1 ) = Var(X2 ) and Cor(X1,X2 ) = 0; this surface has a characteristic ***bell shape***.
* However, the bell shape will be distorted if the predictors are correlated or have unequal variances, as is illustrated in the right-hand panel of Figure 4.5. In this situation, the base of the bell will have an elliptical, rather than circular





The dashed lines are the Bayes decision boundaries. In other words, they represent the set of values x for which δk(x ) = δl(x ); i.e.



for k <> l(L) . (The log πk term from (4.19) has disappeared because each of the three classes has the same number of training observations; i.e. πk is the same for each class.)

These three Bayes decision boundaries divide the predictor space into three regions. The Bayes classifier will classify an observation according to the region in which it is located.

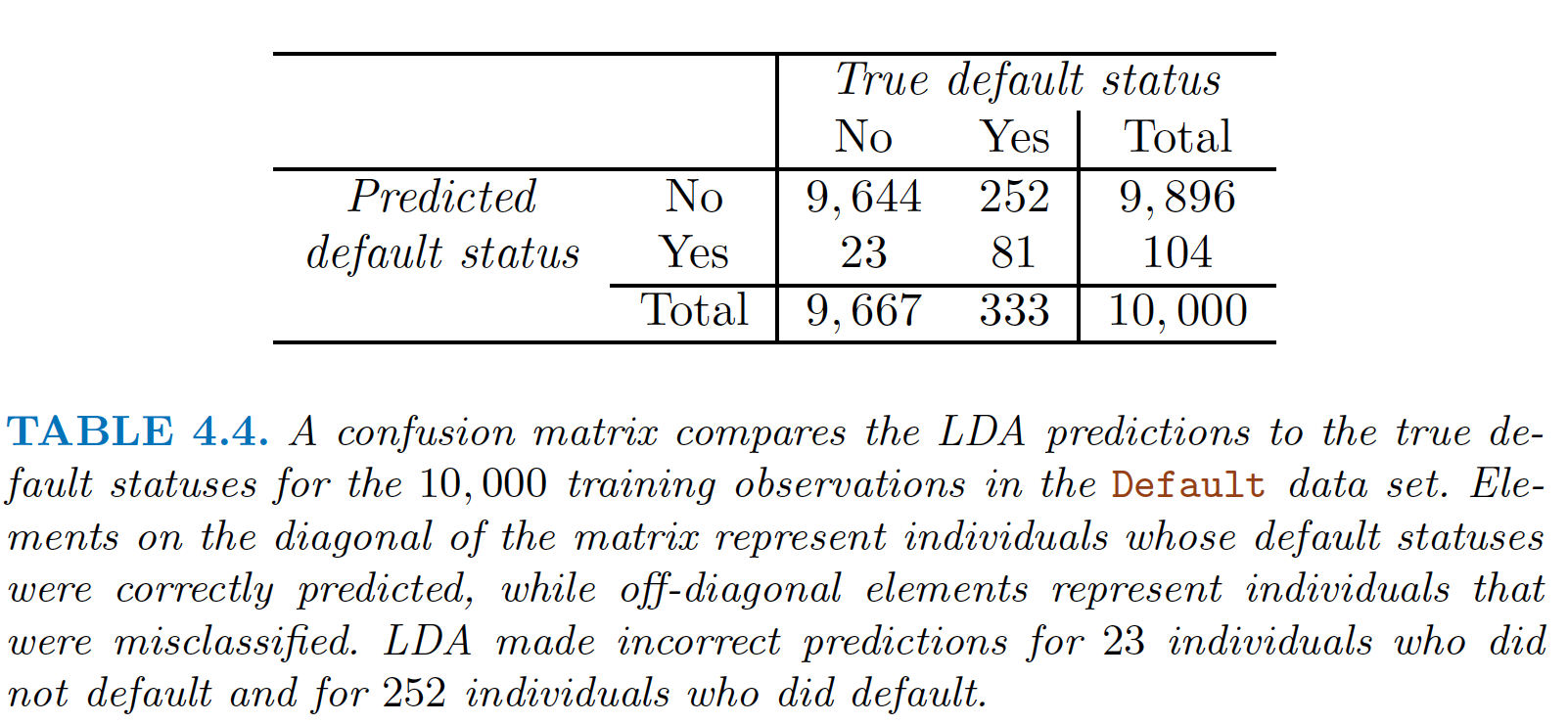
*Once again, we need to estimate the unknown parameters μ1, . . . , μK , π1, . . . , πK , and Σ ; the formulas are similar to those used in the one dimensional case, given in (4.15).*

To assign a new observation X = x , LDA plugs these estimates into (4.19) and classifies to the class for which ˆδk (x ) is largest. Note that in (4.19) δk (x ) is a linear function of x ; that is, the LDA decision rule depends on x only through a linear combination of its elements. Once again, this is the reason for the word linear in LDA.

LDA on Default dataset in order to predict whether or not an individual will default on the basis of credit card balance and student status. The LDA model fit to the 10,000 training samples results in a training error rate of 2. 75 %. This sounds like a low error rate, but two caveats must be noted.

* Since training error is low, but real quantity of interest are test error rates. We might except this classifier to perform worse on new data (test data). We can specifically adjust the parameters of our model to do well on the training data. The higher the ratio of parameters *p* to number of samples *n*, the more we except this overfitting to play a role.
* Second, since only 3. 33% of the individuals in the training sample defaulted, a simple but useless classifier that always predicts that each individual will not default, regardless of his or her credit card balance and student status, will result in an error rate of 3. 33%. In other words, the trivial null classifier will achieve an error rate that is only a bit higher than the LDA training set error rate.

***Confusion matrix:***

******

Only 23 out of 9, 667 of the individuals who did not default were incorrectly labeled. This looks like a pretty low error rate! However, of the 333 individuals who defaulted, 252 (or 75. 7%) were missed by LDA. So while the overall error rate is low, the error rate among individuals who defaulted is very high. From the perspective of a credit card company that is trying to identify high-risk individuals, an error rate of 252/ 333 = 75. 7% among individuals who default may well be unacceptable.

***sensitivity and specificity:***

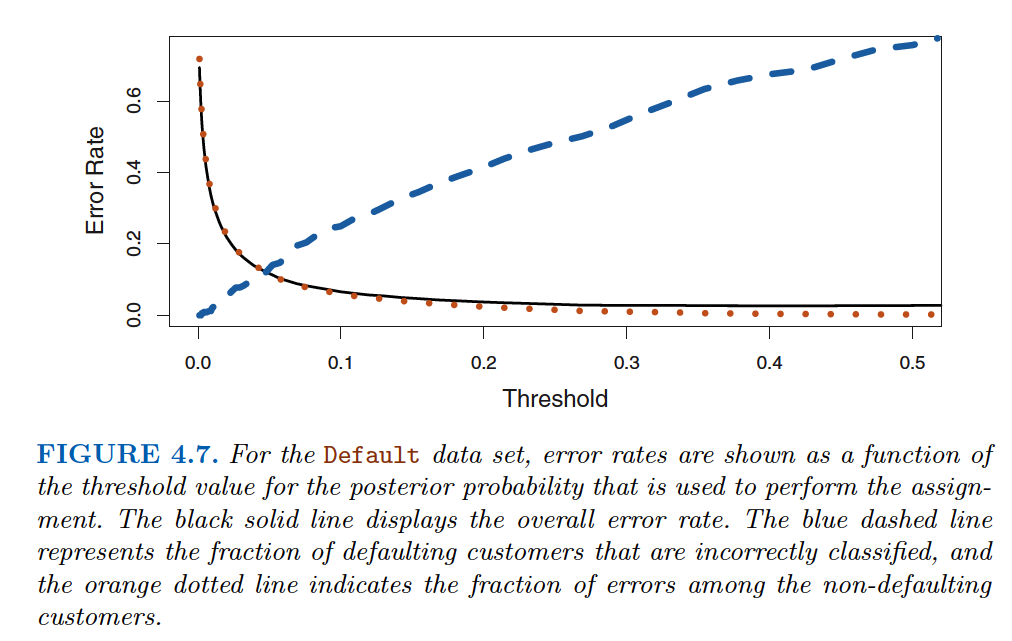
characterize the performance of a classifier or screening test. In this case the ***sensitivity*** is the percentage of true defaulters that are identified, a low 24.3% in this case. The ***specificity*** is the percentage of non-defaulters that are correctly identified, here (1 −23/ 9667)\* 100 = 99. 8%.

*Main objective for this model* : A credit card company might particularly wish to avoid

Incorrectly classifying an individual who will default, whereas incorrectly classifying an individual who will not default, though still to be avoided, is less problematic. We will now see that it is possible to modify LDA in order to develop a classifier that better meets the credit card company’s needs.

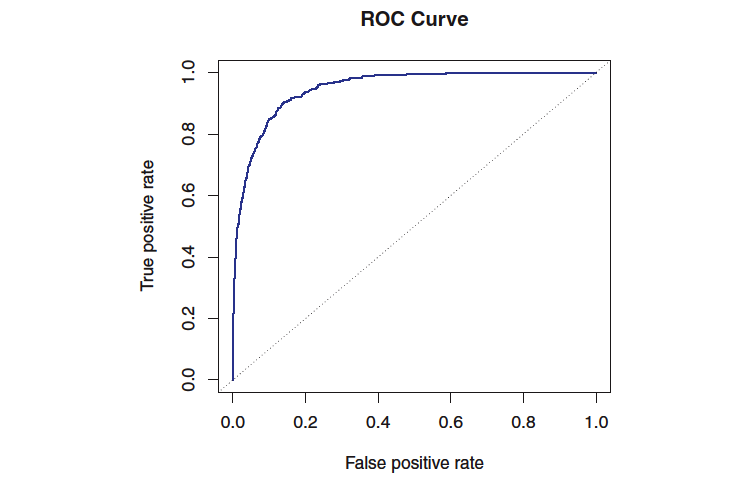
Solution of above problem : For instance, we might label any customer with a posterior probability of default above 20% to the default class instead of 50%.

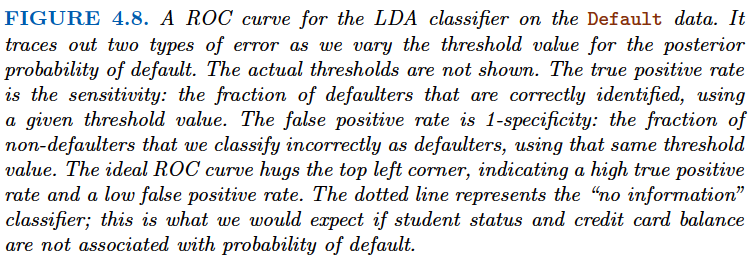
In following graph as we take lower posterior probability threshold (Pr(default = Yes|X = x ) > 0. 2) then error rate in classifying default customer is getting lower but there is increment in error rate of classifying non-default customer and in over all error rate.



***How can we decide which threshold value is best?***

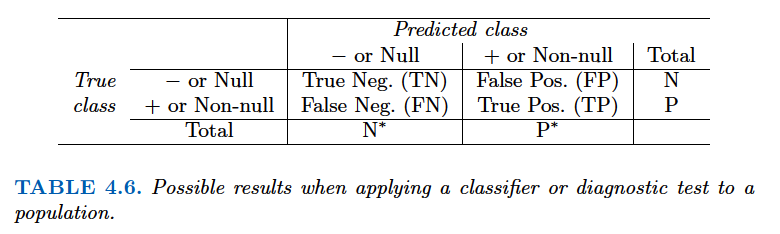
* Such a decision must be based on domain knowledge , such as detailed information about the costs associated with default.
* ***ROC curve: receiver operating characteristics*** from communications theory

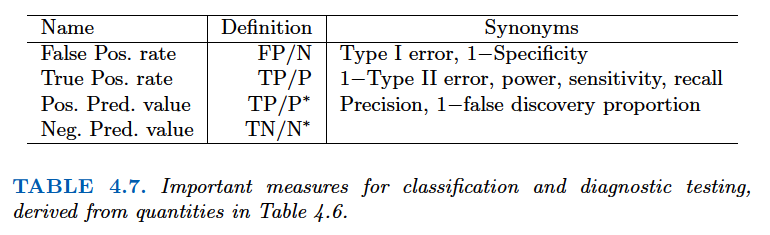




The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the (ROC) curve (AUC). An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier. For this data the AUC is 0. 95, which is close to the maximum of one so would be considered very good. We expect

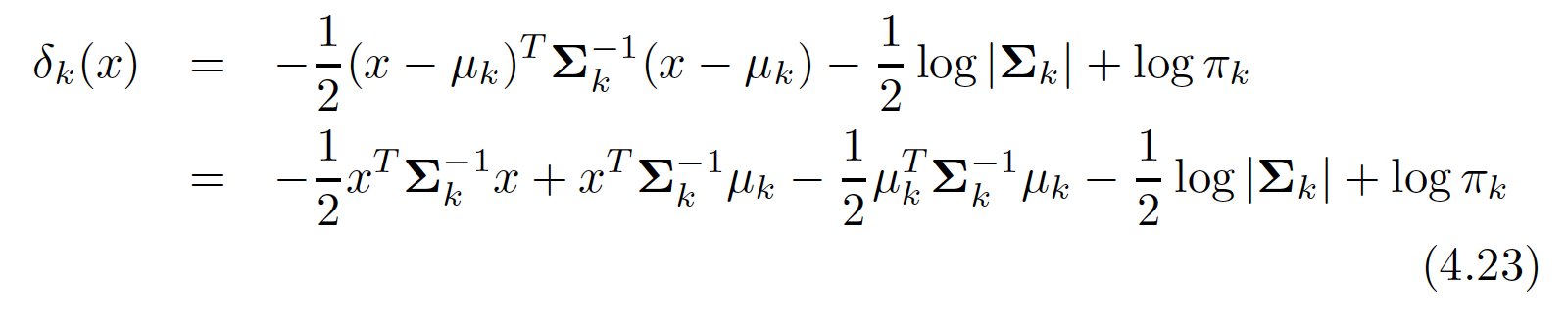
a classifier that performs no better than chance to have an AUC of 0.5 (when evaluated on an independent test set not used in model training).





Quadratic Discriminant Analysis:

Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes’ theorem in order to perform prediction. However, unlike LDA, ***QDA assumes that each class has its own covariance matrix.*** That is, it assumes that an observation from the kth class is of the form X ∼ N(μk,Σk), where Σk is a covariance matrix for the kth class. Under this assumption, the Bayes classifier assigns an observation X = x to the class for which following is largest.

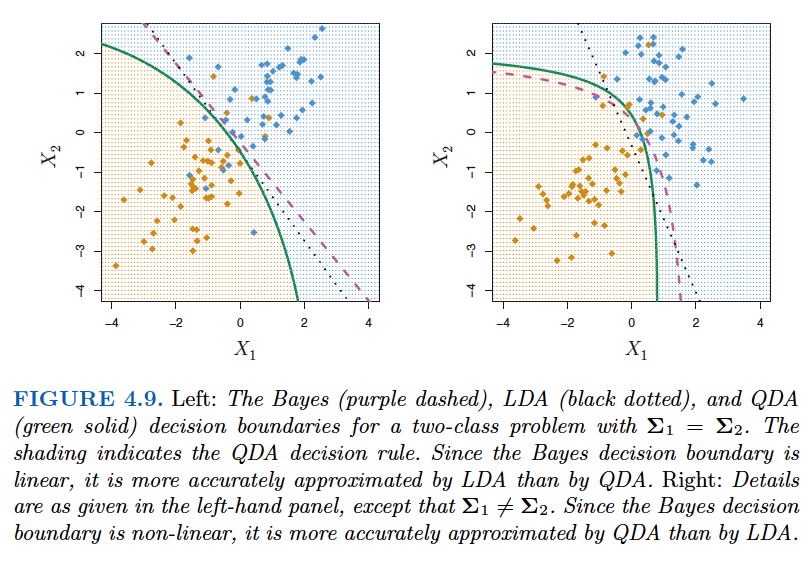


***Why does it matter whether or not we assume that the K classes share a common covariance matrix? In other words, why would one prefer LDA to QDA, or vice-versa?***

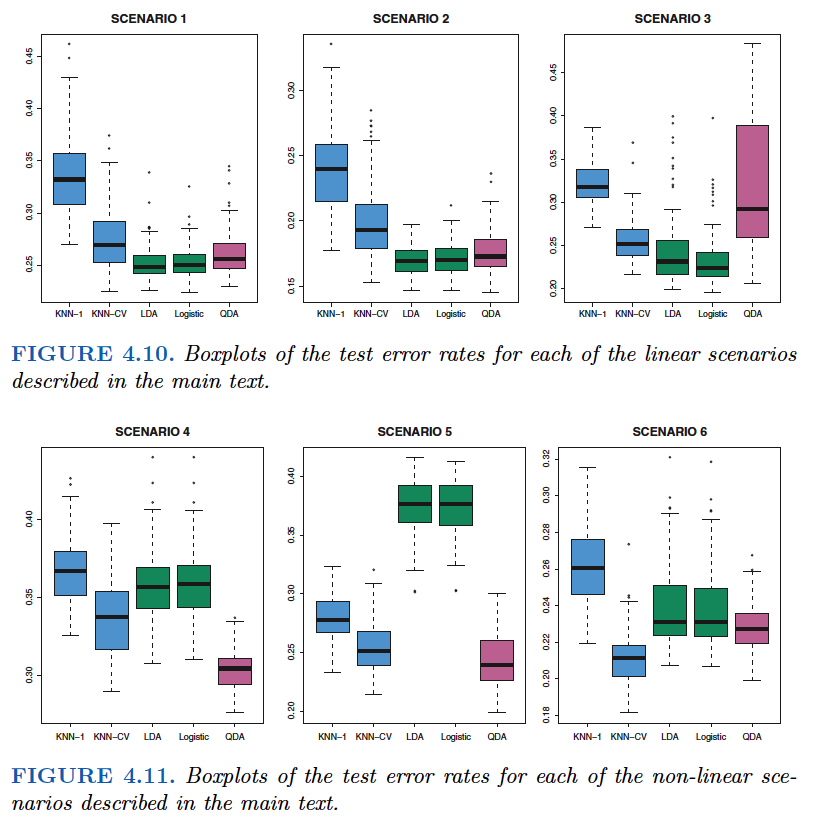
The answer lies in the bias-variance trade-off. When there are p predictors, then estimating a covariance matrix requires estimating p(p +1)/ 2 parameters. QDA estimates a separate covariance matrix for each class, for a total of Kp(p +1)/ 2 parameters.With 50 predictors this is some multiple of 1,275, which is a lot of parameters. By instead assuming that the K classes share a common covariance matrix, the LDA model becomes linear in x , which means there are Kp linear coefficients to estimate. Consequently, LDA is a much less flexible classifier than QDA, and so has substantially lower variance. This can potentially lead to improved prediction performance. But there is a trade-off: if LDA’s assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.

***Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial. In contrast, QDA is recommended if the training set is***

***very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.***

******

As a result, the Bayes decision boundary is linear and is accurately approximated by the LDA decision boundary. The QDA decision boundary is inferior, because it suffers from higher variance without a corresponding decrease in bias. In contrast, the right-hand panel displays a situation in which the orange class has a correlation of 0.7 between the variables and the blue class has a correlation of − 0.7. Now the Bayes decision boundary is quadratic, and so QDA more accurately approximates this boundary than does LDA.



In each of the six scenarios, there were p = 2 predictors. The scenarios

were as follows:

Scenario 1: There were 20 training observations in each of two classes.

The observations within each class were uncorrelated random normal

variables with a different mean in each class. The left-hand panel

of Figure 4.10 shows that LDA performed well in this setting, as

one would expect since this is the model assumed by LDA. KNN

performed poorly because it paid a price in terms of variance that

was not offset by a reduction in bias. QDA also performed worse

than LDA, since it fit a more flexible classifier than necessary. Since

logistic regression assumes a linear decision boundary, its results were

only slightly inferior to those of LDA.

Scenario 2: Details are as in Scenario 1, except that within each

class, the two predictors had a correlation of − 0. 5. The center panel

of Figure 4.10 indicates little change in the relative performances of

the methods as compared to the previous scenario.

Scenario 3: We generated X1 and X2 from the t-distribution , with

t-

50 observations per class. The t -distribution has a similar shape to distribution

the normal distribution, but it has a tendency to yield more extreme

points—that is, more points that are far from the mean. In this setting,

the decision boundary was still linear, and so fit into the logistic

regression framework. The set-up violated the assumptions of LDA,

since the observations were not drawn from a normal distribution.

The right-hand panel of Figure 4.10 shows that logistic regression

outperformed LDA, though both methods were superior to the other

approaches. In particular, the QDA results deteriorated considerably

as a consequence of non-normality.

Scenario 4: The data were generated from a normal distribution,

with a correlation of 0. 5 between the predictors in the first class,

and correlation of − 0. 5 between the predictors in the second class.

This setup corresponded to the QDA assumption, and resulted in

quadratic decision boundaries. The left-hand panel of Figure 4.11

shows that QDA outperformed all of the other approaches.

Scenario 5: Within each class, the observations were generated from

a normal distribution with uncorrelated predictors. However, the responses

were sampled from the logistic function using X2

1 , X2

2 , and

X1 °ø X2 as predictors. Consequently, there is a quadratic decision

boundary. The center panel of Figure 4.11 indicates that QDA once

again performed best, followed closely by KNN-CV. The linear methods

had poor performance.

154 4. Classification

Scenario 6: Details are as in the previous scenario, but the responses

were sampled from a more complicated non-linear function. As a result,

even the quadratic decision boundaries of QDA could not adequately

model the data. The right-hand panel of Figure 4.11 shows

that QDA gave slightly better results than the linear methods, while

the much more flexible KNN-CV method gave the best results. But

KNN with K = 1 gave the worst results out of all methods. This

highlights the fact that even when the data exhibits a complex nonlinear

relationship, a non-parametric method such as KNN can still

give poor results if the level of smoothness is not chosen correctly.

Summary of above :

If the decision boundary is linear and distribution of classes is normal then LDA will outperform other methods but if distribution is not normal then Logistic regression may outperform.

If two classes has correlation pf 0.5 and -0.5 in variables then it would need a quadratic decision boundary. Then QDA would outperform every other method.

Responses were sampled from a more complicated non-linear function. Figure 4.11 shows that QDA gave slightly better results than the linear methods, while the much more flexible KNN-CV method gave the best results. But KNN with K = 1 gave the worst results out of all methods. ***This highlights the fact that even when the data exhibits a complex nonlinear relationship, a non-parametric method such as KNN can still give poor results if the level of smoothness is not chosen correctly.***