DATA 572: Supervised Learning

2023W2

Shan Du

Quadratic Discriminant Analysis

- Quadratic discriminant analysis (QDA) assumes that the observations from each class are drawn from a Gaussian distribution, and each class has its own covariance matrix.
- That is, it assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \Sigma_k)$, where Σ_k is a covariance matrix for the kth class.
- Under this assumption, the Bayes classifier assigns an observation $X = x$ to the class for which

Quadratic Discriminant Analysis

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}log|\Sigma_k| + \log\pi_k$$

$$= -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}log|\Sigma_k| + \log\pi_k$$

is largest.

- Unlike in LDA, the quantity x appears as a quadratic function.

Quadratic Discriminant Analysis

- 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.

Quadratic Discriminant Analysis

- 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.
- By instead assuming that the K classes share a common covariance matrix, the LDA model becomes linear in x, which means there are $\boxed{Kp}$ linear coefficients to estimate. Consequently, LDA is a much less flexible classifier than QDA, and so has substantially lower variance.

Quadratic Discriminant Analysis

more flexible model then its more complex and usually has higher variance

- 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.

Quadratic Discriminant Analysis

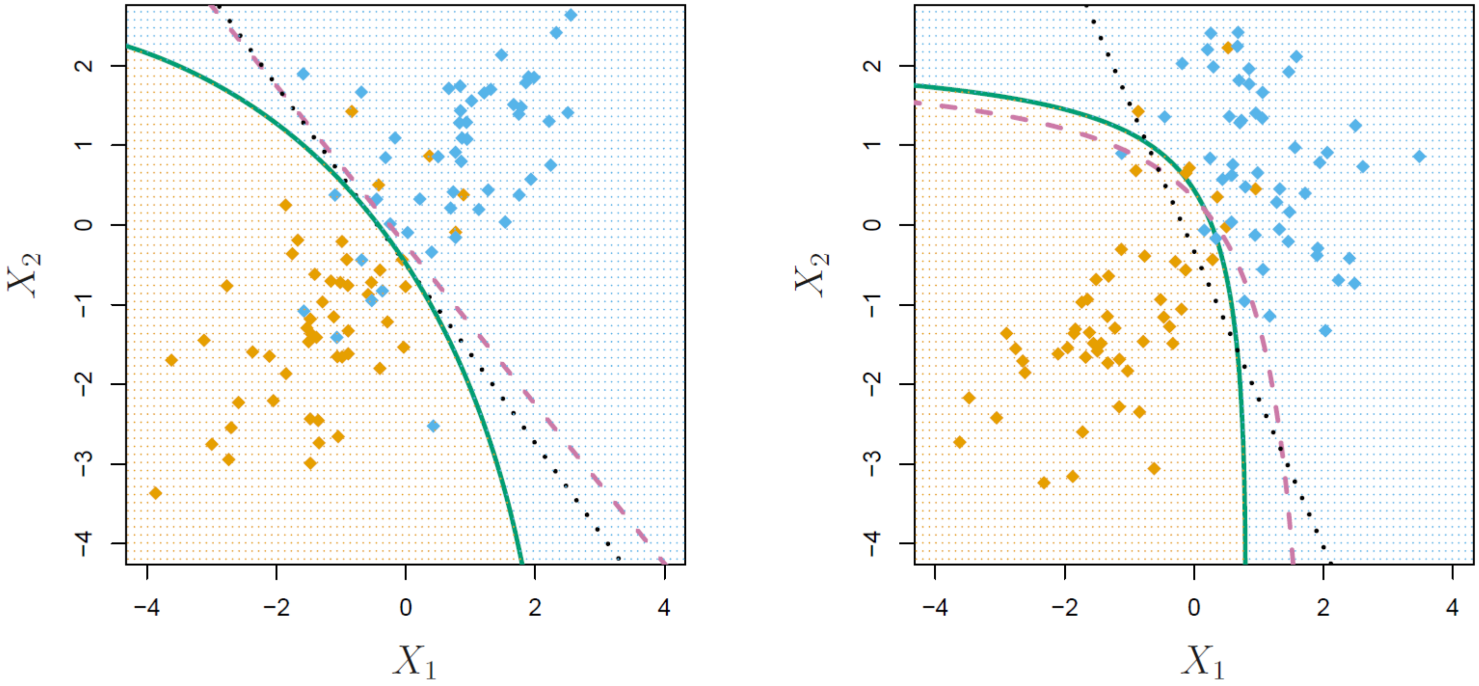


FIGURE 4.9. Left: *The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with* $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$. *The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.* Right: *Details are as given in the left-hand panel, except that* $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$. *Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.*

Naive Bayes

- We use Bayes’ theorem to motivate the popular naive Bayes classifier.
- The naive Bayes classifier takes a different tack for estimating $f_1(x), \dots, f_K(x)$. Instead of assuming that these functions belong to a particular family of distributions (e.g., multivariate normal), we instead make a single assumption:

$$\boxed{\begin{array}{c} Within\ the\ kth\ class, \\ the\ p\ predictors\ are\ independent. \end{array}}$$

Naive Bayes

- Stated mathematically, this assumption means that for $k = 1, \ldots, K$

$$\underline{f_k(x)} = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)$$

where f_{kj} is the density function of the jth predictor among observations in the kth class.

Naive Bayes

- Once we have made the naive Bayes assumption, we can obtain an expression for the posterior probability

$$\Pr(Y = k|X = x)$$

$$\boxed{= \frac{\pi_k \times f_{k1}\left(x_1\right) \times f_{k2}\left(x_2\right) \times \cdots \times f_{kp}\left(x_p\right)}{\sum_{l=1}^{K} \pi_l \times f_{l1}\left(x_1\right) \times f_{l2}\left(x_2\right) \times \cdots \times f_{lp}\left(x_p\right)}}$$

for $k = 1, \ldots, K$

Naive Bayes

- To estimate the one-dimensional density function f_{kj} using training data $x_{1j}, \dots, x_{nj}$ we have a few options:
  - If X_j is quantitative, we assume that within each class, the jth predictor is drawn from a (univariate) normal distribution and the predictors are independent.
  - If X_j is quantitative, another option is to use a non-parametric estimate for f_{kj}. A very simple way to do this is by making a histogram for the observations of the *j*th predictor within each class. Then we can estimate $f_{kj}(x_j)$ as the fraction of the training observations in the *k*th class that belong to the same histogram bin as x_j.

Naive Bayes

– If X_j is qualitative, then we can simply count the proportion of training observations for the *j*th predictor corresponding to each class. For instance, suppose that $x_j \in \{1, 2, 3\}$, and we have 100 observations in the *k*th class. Suppose that the *j*th predictor takes on values of 1, 2, and 3 in 32, 55, and 13 of those observations, respectively. Then we can estimate f_{kj} as

$$\hat{f}_{kj}(x_j) = \begin{cases} 0.32 & if\ x_j = 1 \\ 0.55 & if\ x_j = 2 \\ 0.13 & if\ x_j = 3 \end{cases}$$

32/100 = 0.32

55/100 = 0.55

Naive Bayes

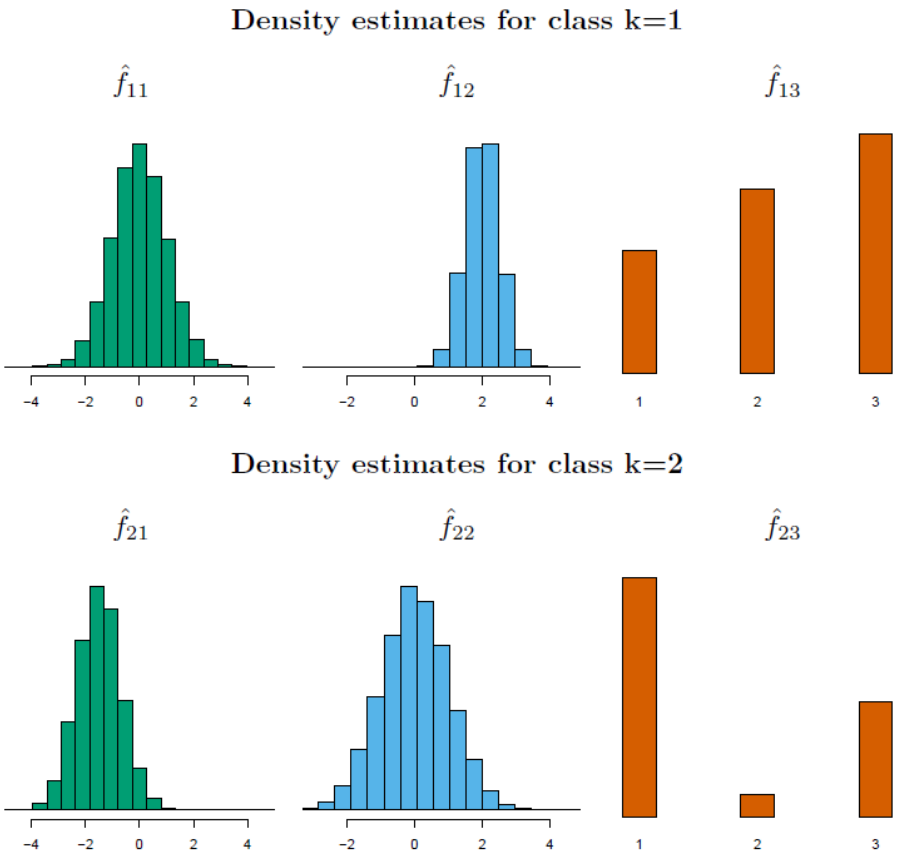


FIGURE 4.10. *In the toy example in Section 4.4.4, we generate data with $p = 3$ predictors and $K = 2$ classes. The first two predictors are quantitative, and the third predictor is qualitative with three levels. In each class, the estimated density for each of the three predictors is displayed. If the prior probabilities for the two classes are equal, then the observation $x^* = (0.4, 1.5, 1)^T$ has a 94.4% posterior probability of belonging to the first class.*

K-Nearest Neighbors (KNN)

- In theory we would always like to predict qualitative responses using the Bayes classifier. But for real data, we do not know the conditional distribution of Y given X, and so computing the Bayes classifier is impossible.
- Many approaches attempt to estimate the conditional distribution of Y given X, and then classify a given observation to the class with highest *estimated probability*. One such method is the *K-nearest neighbors (KNN)* classifier.

K-Nearest Neighbors (KNN)

- Given a positive integer K and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by $\mathcal{N}_0$. It then estimates the conditional probability for class j as the fraction of points in $\mathcal{N}_0$ whose response values equal j:

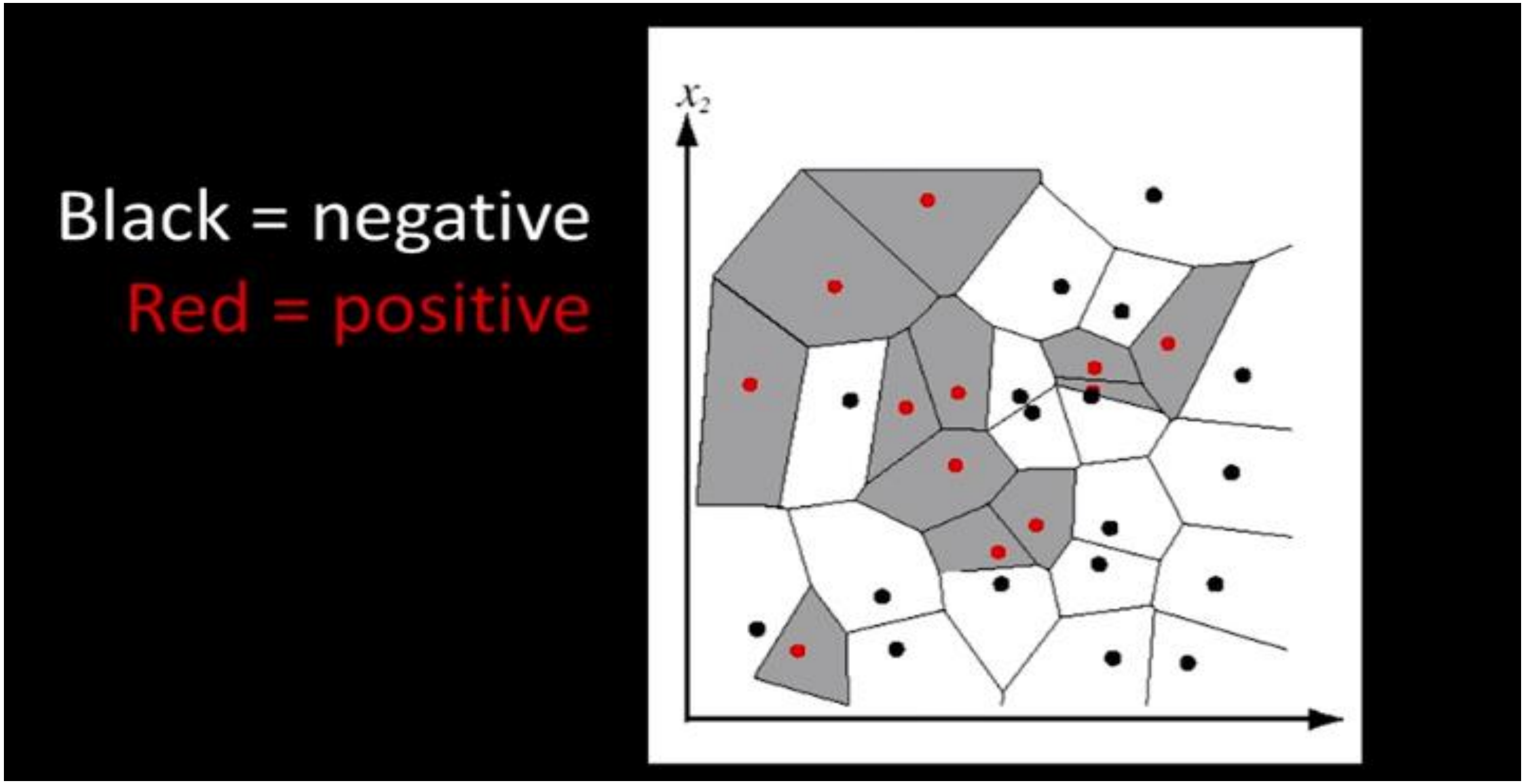
$$\boxed{\Pr(Y = j | X = x_0) = \frac{1}{K} \sum_{i \epsilon \mathcal{N}_0} I(y_i = j)}$$

K-Nearest Neighbors (KNN)

- Finally, KNN classifies the test observation x_0 to the class with the largest probability.

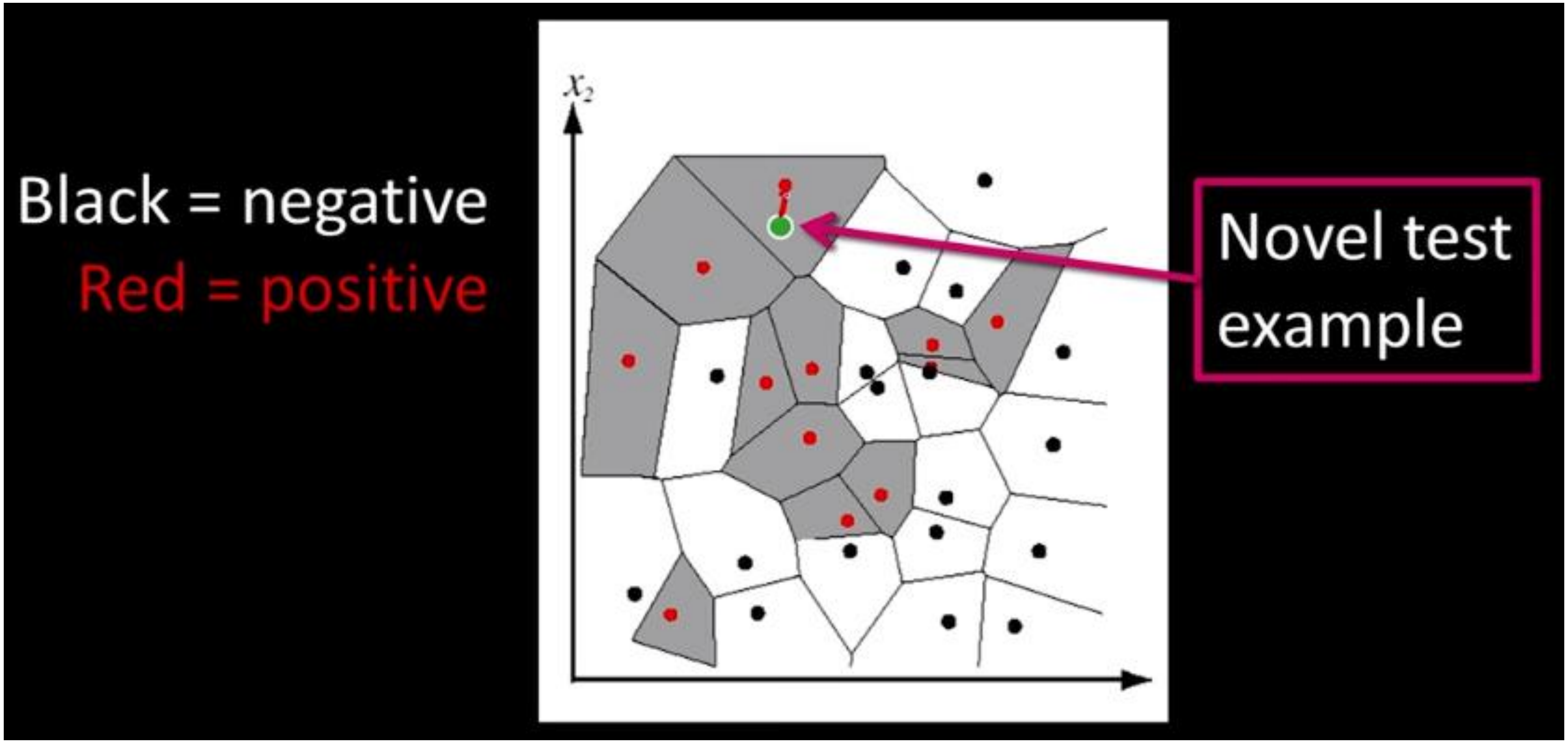
Nearest Neighbor Classification

Choose label of nearest training data point



Nearest Neighbor Classification

Choose label of nearest training data point

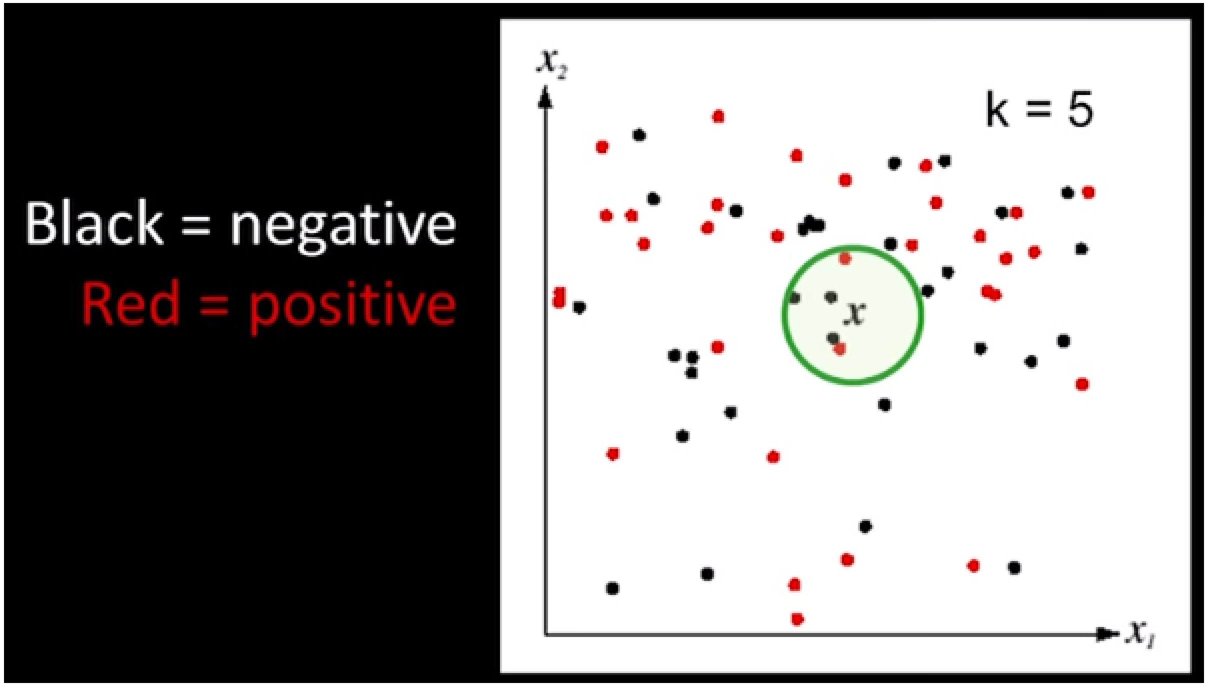


Black = negative
Red = positive

Novel test example

K-Nearest Neighbors Classification

- For a new point, find the k closest points from training data
- Labels of the k points "vote" to classify



If query lands here, the 5NN consist of 3 negatives and 2 positives, so we classify it as negative.

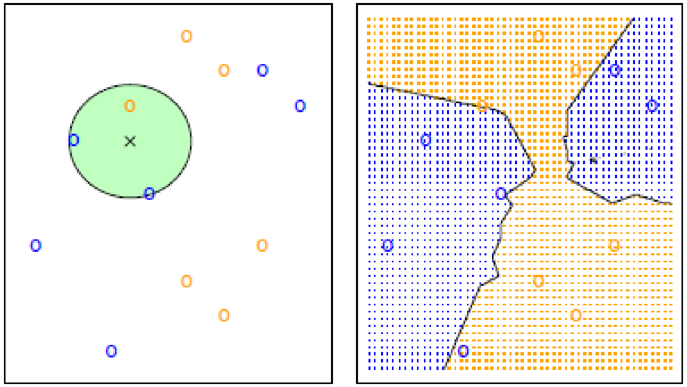


FIGURE 2.14. *The KNN approach, using* $K = 3$*, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

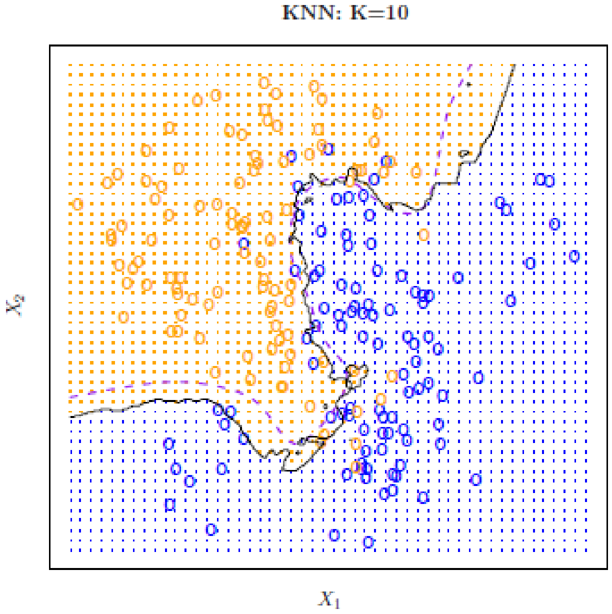


FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using* $K = 10$*. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

K-Nearest Neighbors (KNN)

- The choice of K has a drastic effect on the KNN classifier obtained.

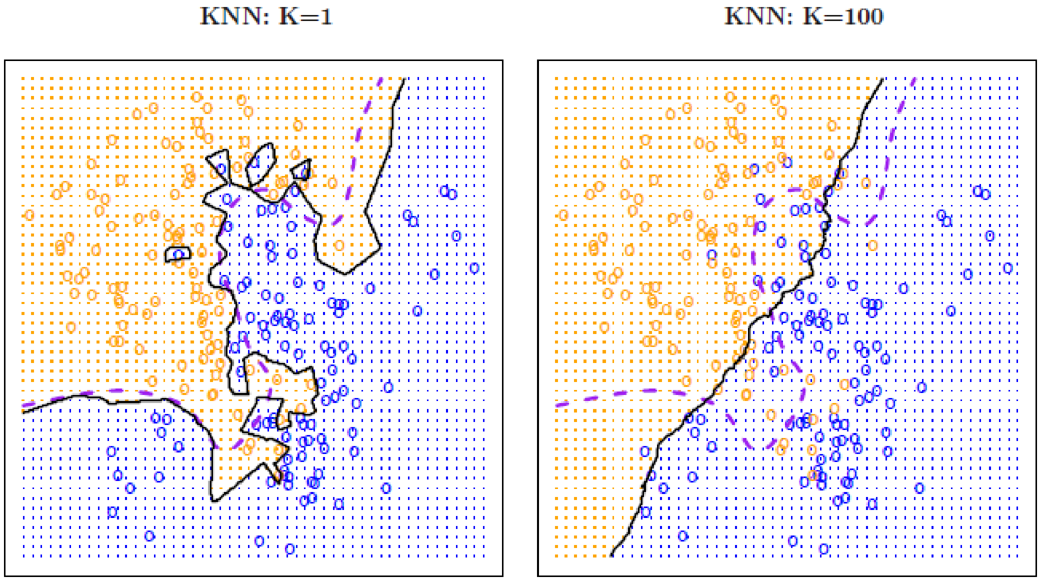


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

K-Nearest Neighbors (KNN)

- KNN takes a completely different approach from the classifiers seen earlier in this lecture.
- KNN is a completely non-parametric approach: no assumptions are made about the shape of the decision boundary. We make the following observations about KNN:
  - Because KNN is completely non-parametric, we can expect this approach to dominate LDA and logistic regression when the decision boundary is highly non-linear, provided that n is very large and p is small.

K-Nearest Neighbors (KNN)

- In order to provide accurate classification, KNN requires a lot of observations relative to the number of predictors—that is, n much larger than p. This has to do with the fact that KNN is non-parametric, and thus tends to reduce the bias while incurring a lot of variance.
- In settings where the decision boundary is non-linear but n is only modest, or p is not very small, then QDA may be preferred to KNN. This is because QDA can provide a non-linear decision boundary while taking advantage of a parametric form, which means that it requires a smaller sample size for accurate classification, relative to KNN.
- Unlike logistic regression, KNN does not tell us which predictors are important.

K-Nearest Neighbors (KNN)

- As a non-parametric learning algorithm, k-nearest neighbors is not restricted to a fixed number of parameters. We usually think of the k-nearest neighbors algorithm as not having any parameters, but rather implementing a simple function of the training data.
- In fact, there is not even really a training stage or learning process. Instead, at test time, when we want to produce an output y for a new test input x, we find the k-nearest neighbors to x in the training data X. We then return the average of the corresponding y values in the training set.

An Empirical Comparison of Classification Methods

- We now compare the *empirical* (practical) performance of logistic regression, LDA, QDA, naive Bayes, and KNN.
- We generated data from six different scenarios, each of which involves a binary (two-class) classification problem.
- In three of the scenarios, the Bayes decision boundary is linear, and in the remaining scenarios it is non-linear.

An Empirical Comparison of Classification Methods

- For each scenario, we produced 100 random training data sets. On each of these training sets, we fit each method to the data and computed the resulting test error rate on a large test set.

An Empirical Comparison of Classification Methods

- *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. normal, different mean, and uncorrelated
- *Scenario 2*: Details are as in *Scenario 1*, except that within each class, the two predictors had a correlation of −0.5. normal, different mean, correlated
- *Scenario 3*: As in *Scenario 1*, there is substantial negative correlation between the predictors within each class. However, this time we generated X_1 and X_2 from the $t - distribution$, with 50 observations per class. The t-distribution has a similar shape to the normal distribution, but it has a tendency to yield more extreme points—that is, more points that are far from the mean. normal with extreme points, different mean, correlated

An Empirical Comparison of Classification Methods

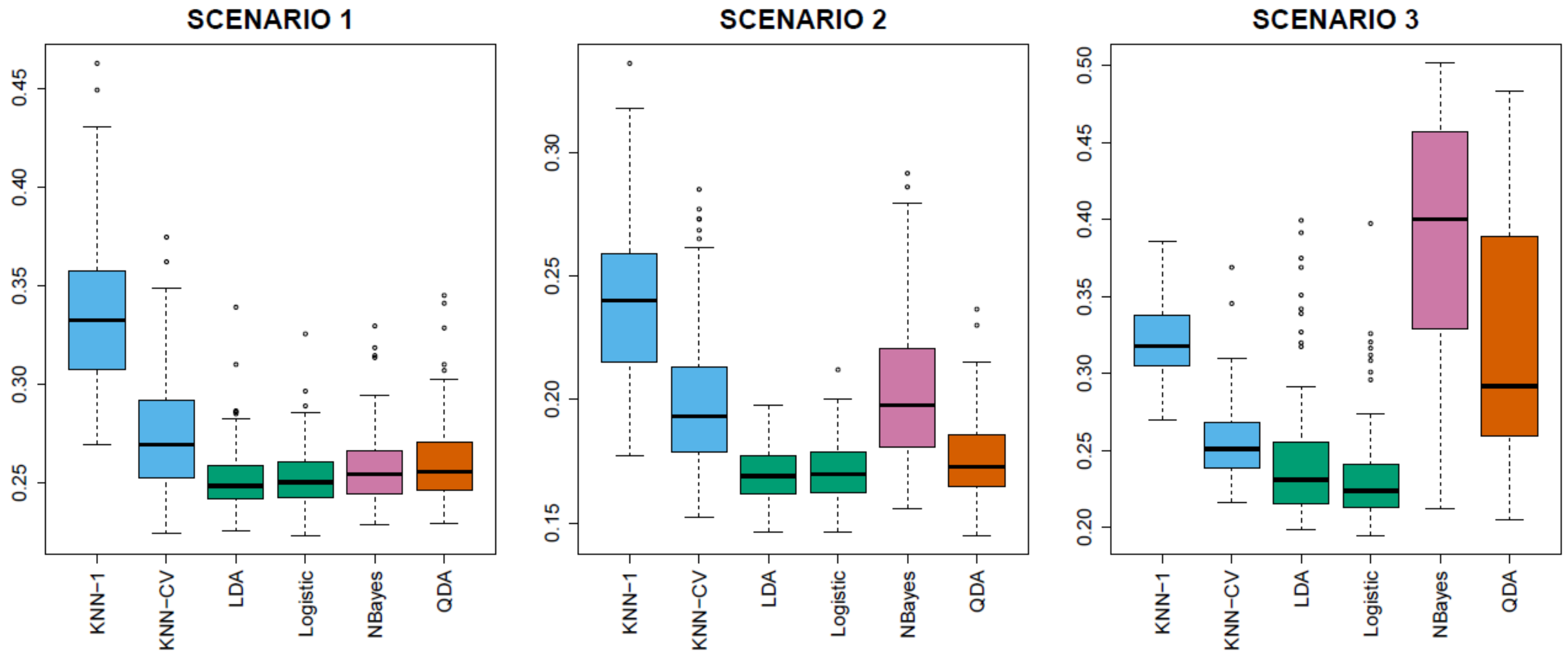


FIGURE 4.11. *Boxplots of the test error rates for each of the linear scenarios described in the main text.*

LDA and logistic good because linear boundary, Naive Bayes good because uncorrelated

LDA and Logistic still good because linear boundary - Naive Bayes worse because there is correlation between predictors

LDA worse because not normal, Naive Bayes bad because of correlation between predictors, and QDA is bad because bad b/c linear boundary

An Empirical Comparison of Classification Methods

- *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.
- *Scenario 5*: The data were generated from a normal distribution with uncorrelated predictors. Then the responses were sampled from the logistic function applied to a complicated non-linear function of the predictors.

An Empirical Comparison of Classification Methods

- *Scenario 6*: The observations were generated from a normal distribution with a different diagonal covariance matrix for each class. However, the sample size was very small: just $n = 6$ in each class.

An Empirical Comparison of Classification Methods

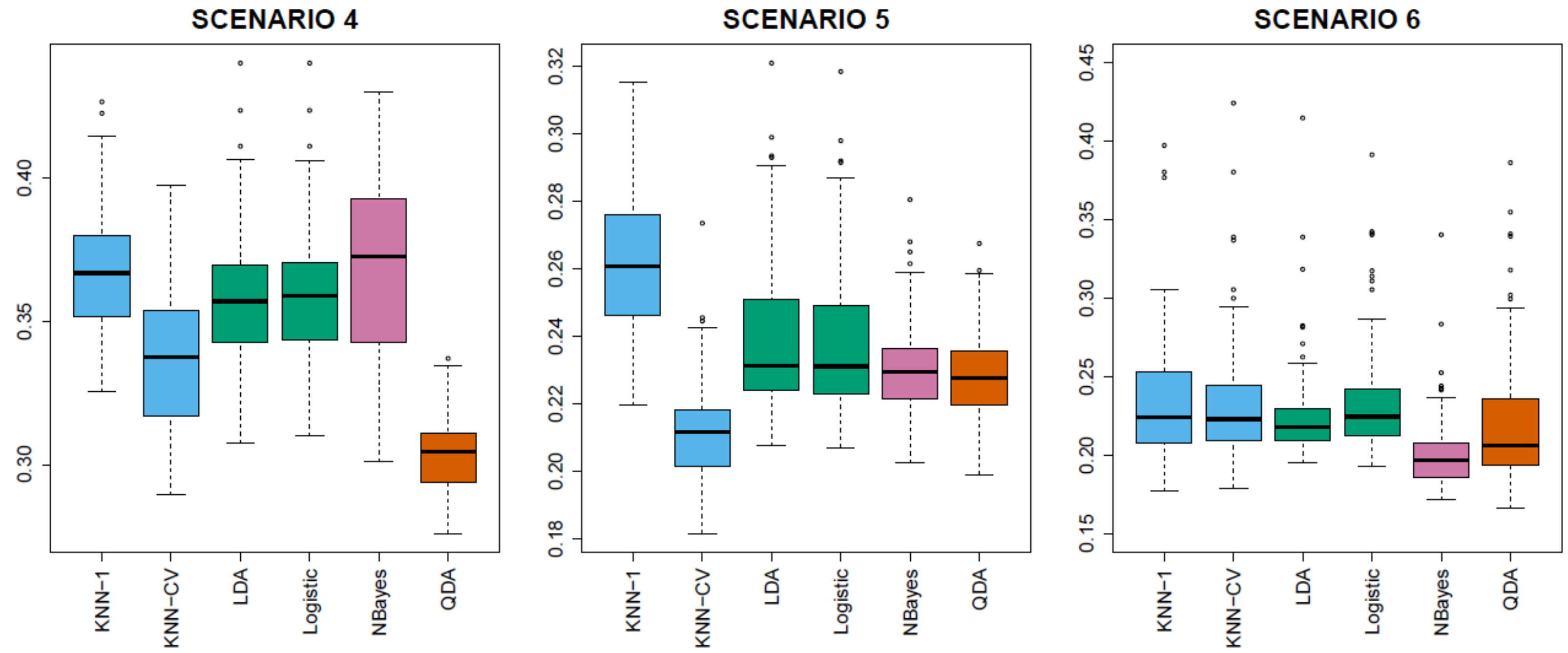


FIGURE 4.12. *Boxplots of the test error rates for each of the non-linear scenarios described in the main text.*

Non-linear boundary so QDA is good, KNN-CV more flexible so better than KNN-1, and LDA/Logistic bad because non-linear boundary, Naive bayes bad because of correlation

Naive bayes better b/c removed correlation b/w predictors, KNN-CV good because non-linear boundary

An Empirical Comparison of Classification Methods

- No one method will dominate the others in every situation.
- When the true decision boundaries are linear, then the LDA and logistic regression approaches will tend to perform well.
- When the boundaries are moderately non-linear, QDA or naive Bayes may give better results.
- Finally, for much more complicated decision boundaries, a non-parametric approach such as KNN can be superior. But the level of smoothness for a non-parametric approach must be chosen carefully.