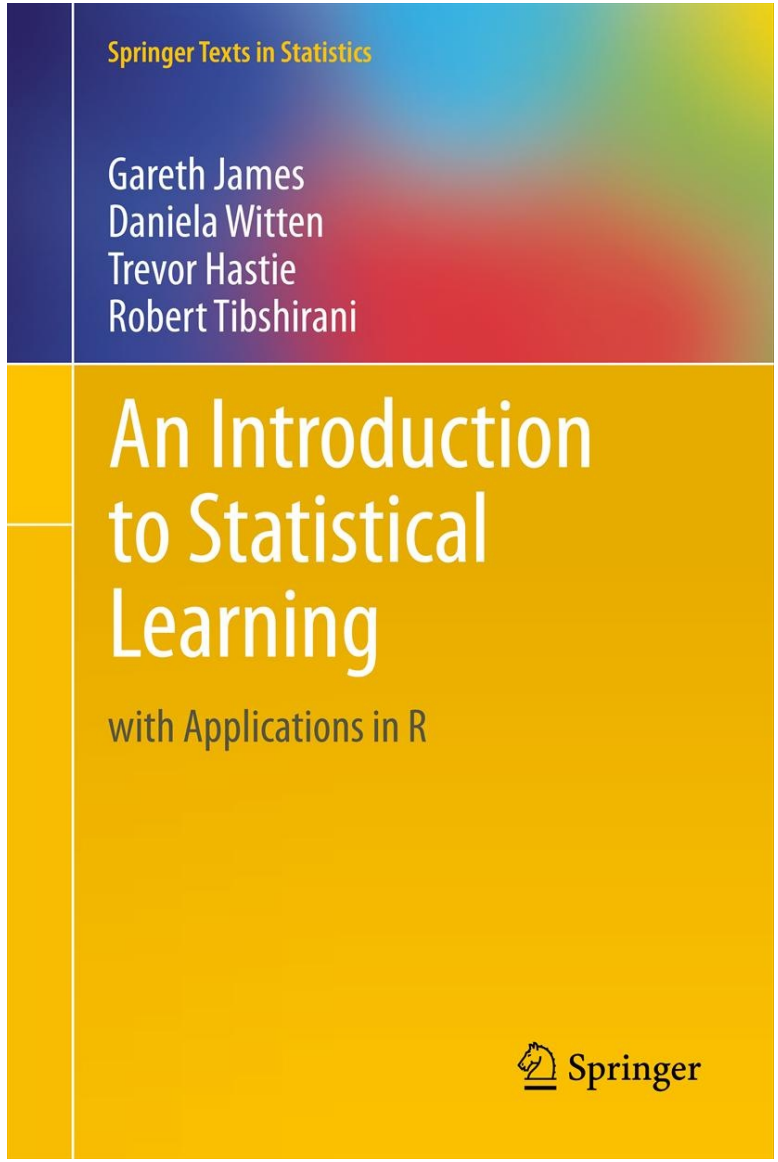
Statistical and Machine Learning Methods for Classification Using R

Pankaj Choudhary
University of Texas at Dallas, USA
pankaj@utdallas.edu

MESIO UPC-UB XIII Summer School
9:00 AM - 12:00 PM, June 17-21, 2019

Course materials available at:
`www.utdallas.edu/~pankaj/classification_course/`



A free PDF of the book and other useful materials available at:
`http://www-bcf.usc.edu/~gareth/ISL/`

Tentative Course Plan

- Part I: Introduction (Chapters 1, 4, and 5)
- Part II: Logistic Regression and Discriminant Analysis (Chapter 4)
- Part III: Tree-Based Methods (Chapter 8)
- Part IV: Support Vector Machines (Chapter 9)

Evaluation Policy

- 5 homework assignments
- Homework 1-4 (15 points each): Due the next day
- Homework 5 (40 points): Due on Friday, June 28
- Total points: 100

Part I:
Introduction

Examples

Ex: A credit card service would like to determine whether or not an online transaction being performed is fraudulent based on the cardholder's purchase history, IP address, etc.

- **Response**: fraudulent or not — **binary classification**
- **Predictors**: purchase history, IP address, etc.

Ex: A patient arrives at an emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions, namely, stroke, drug overdose, or seizure.

- **Response**: Medical condition — one of three possibilities
- **Predictors**: symptomps

Notation

- Y: a *response* (or output or dependent) variable — *qualitative* (or categorical). Values of Y are class labels, e.g., 1, 2, ..., *with no ordering*
- $X_1, \ldots, X_p$: *predictors* (or variables or covariates or features or explanatory or independent variables) — some may be quantitative and others categorical
- X: $(X_1, \ldots, X_p)$
- p: number of predictors
- n: number of subjects (or observations)
- i: subject index $(i = 1, \ldots, n)$
- j: variable index $(j = 1, \ldots, p)$
- Y_i: value of Y for subject i
- X_{ij}: value of X_j for subject i, $X_i = (X_{i1}, \ldots, X_{ip})$

		Predictors					
Subject	Y	X_1	$\dots$	X_j	$\dots$	X_p	X
1	Y_1	X_{11}	$\dots$	X_{1j}	$\dots$	X_{1p}	X_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	Y_i	X_{i1}	$\dots$	X_{ij}	$\dots$	X_{ip}	X_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	Y_n	X_{n1}	$\dots$	X_{nj}	$\dots$	X_{np}	X_n

- **Data**: $(Y_i, X_i), i = 1, \dots, n$ — **training data**. The observations from different subjects are assumed to be *independent*
- $f(X)$: True but unknown function of X that relates Y to X
- $\hat{f}(X)$: Estimate of f from training data — **a key goal**

K-Nearest Neighbors (KNN) classifier

Given a **prediction point** x_0:

Step 1 Pick a positive integer K

Step 2 Identify the set of K points in the training data that are **closest** to x_0. This set — represented by $\mathcal{N}_0$ — contains the K nearest neighbors of x_0.

Step 3 For each class k, estimate the **conditional probability** of the class as

$$\hat{P}(Y = k|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = k),$$

i.e., the fraction of points in $\mathcal{N}_0$ whose response equals k

Step 4 Classify x_0 to the class with the largest probability

Binary classification: Two classes, labelled 1 & 2

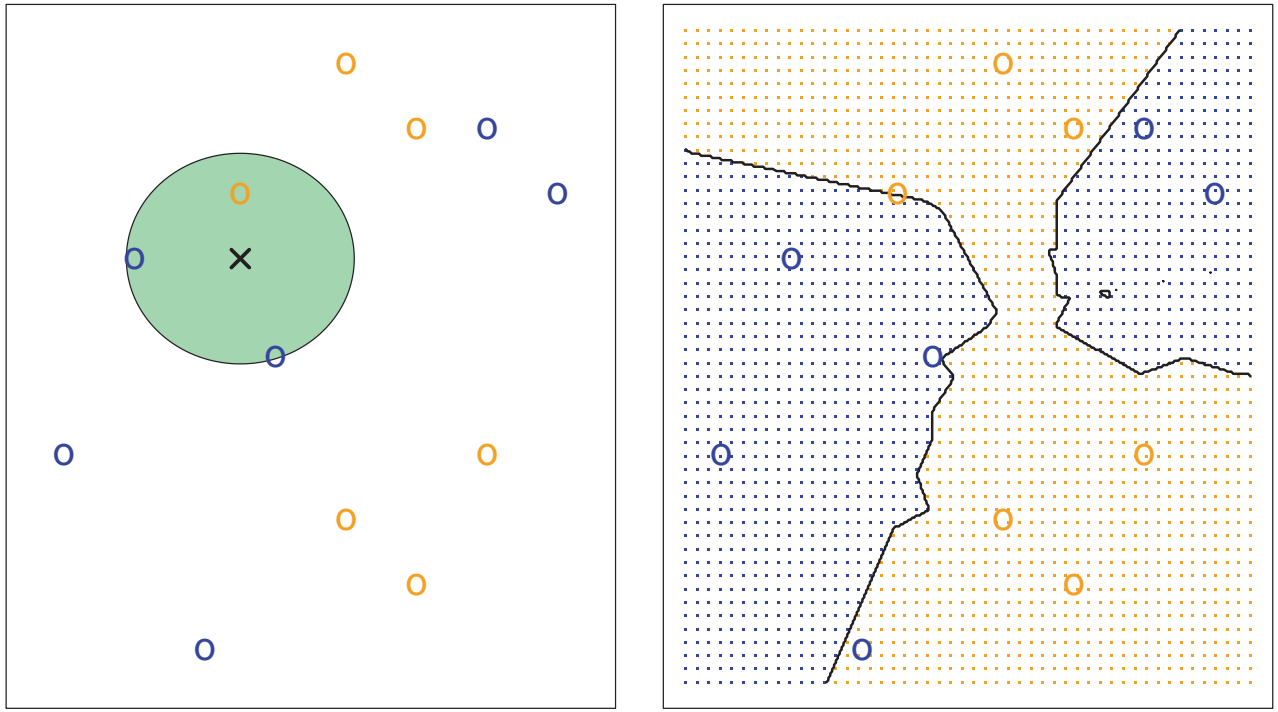


FIGURE 2.14. *The KNN approach, using* $K = 3$*, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, 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.*

Source: ISL

- Use Euclidean distance to measure closeness to a point x
- The classification is decided by the **majority vote**, with ties broken at random. If there are ties for the Kth nearest neighbor, all candidates are included in the vote.
- The number of neighbors K controls flexibility
- More flexible = less restrictive, i.e., better at following the data
- As K increases, flexibility decreases, implying that bias increases and variance decreases
- Assuming no ties, what is the proportion of misclassified observations in training data for $K = 1$? Zero
- **Q:** How to choose the optimal K? (Later)
- **Q:** What is f here?

Learning

Learning: Obtaining an estimate $\hat{f}$ of f from the data

- **Inference**: Understand the form of $f(X)$, i.e., how f changes with X — often but not always the goal is to make predictions for Y
- **Prediction**: Predict Y from X
- **Classification**: Prediction when Y is categorical

Statistical Learning versus Machine Learning

- Machine learning arose as a subfield of Artificial Intelligence.
- Statistical learning arose as a subfield of Statistics.
- *There is much overlap* — both fields focus on supervised and unsupervised problems:
  - Machine learning has a greater emphasis on *large scale* applications and *prediction accuracy*.
  - Statistical learning emphasizes *models* and their interpretability, and *precision* and *uncertainty*.
- But the distinction has become more and more blurred, and there is a great deal of "cross-fertilization".
- Machine learning has the upper hand in *Marketing!*

Source: ISL authors

Prediction vs Inference

Prediction: Determined using $\hat{f}(X)$

- $\hat{f}$ can be treated like a black box, i.e., don't care about the exact form of $\hat{f}$, provided it yields accurate predictions
- interpretation of $\hat{f}$ is not of concern

Inference: Understanding the relationship f between Y and X

- Which predictors are associated with Y?
- What is the form of f — linear or non-linear?
- $\hat{f}$ cannot be a black box, need to know its exact form
- Helpful if $\hat{f}$ has a simple expression as interpretation of $\hat{f}$ is a primary concern

The distinction between prediction and inference is important as different methods may be appropriate for different purposes — often both are of interest

- **Inference**: Linear models or extensions
- **Prediction**: Can use highly non-linear methods

Our main focus: Prediction

Fact: No single method dominates others in all data situations, i.e., different methods may be appropriate for similar type of but different data sets, and often simple methods beat the fancy ones.

Parametric vs Nonparametric Estimation of f

Parametric method: Assumes a functional form for $f(X)$ and estimates the parameters (unknown coefficients) in the model.

Ex: (Linear model) $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$, and the parameters $\beta = (\beta_0, \ldots, \beta_p)$ can be estimated using, e.g., using maximum likelihood.

Advantage: Easy to fit and interpret

Disadvantage: Model assumed for f will at best be an *approximation* to the true unknown f. If the assumption is seriously wrong, $\hat{f}$ will be quite poor

Alternative: Use *flexible* models that may fit many different functional forms for f. But they involve estimating a greater number of parameters, which may lead to *overfitting*.

Overfitting:

- Follow the training data too closely
- Find patterns in the training data that are due to random chance but don't exist in reality
- Fits the training data quite well but gives *poor* predictions for **test data** — data not used in training of the method
- Misclassification error rate small in training data but large in test data

Nonparametric method: Does not make any assumptions about the functional form for f

Advantage: Can fit a wide range of possible shapes for f

Disadvantage: Since it does not reduce the problem of estimating f to a small number of parameters, it requires a large n (much larger than is typically needed for a parametric method) for accurate estimation of f.

Q: Why would we ever choose a more restrictive (less flexible) method instead of a very flexible approach?

A: More flexible methods are less interpretable. **Inference**: Use relatively inflexible methods. **Prediction**: May want to use the most flexible methods, but they may lead to overfitting.

Error in Classification

Values of Y: A finite number of class labels, e.g., $1, 2, \ldots$, denoted by k — no ordering

Predicted value: $\hat{Y}$ (a class label) — determined using $\hat{f}(X)$

Error: Take *zero-one error*, i.e.,

$$I(\hat{Y} \neq Y) = \begin{cases} 1, & \text{if } \hat{Y} \neq Y \\ 0, & \text{if } \hat{Y} = Y \end{cases}$$

Expected error rate: $E\{I(\hat{Y} \neq Y)\} = P(\hat{Y} \neq Y)$ — *probability of misclassification*

Bayes classifier: Minimizes the **expected error rate** — the **best one**. The class k predicted by it is the one for which the *conditional probability* $P(Y = k|X)$ is maximum, i.e., it predicts the **most likely** class.

Two-class problem:

- Predict class 1 if $P(Y = 1|X) > 0.5$ and class 2 otherwise
- *Bayes decision boundary*: $\{x : P(Y = 1|X = x) = 0.5\}$

Bayes error rate: Error rate for the best classifier — provides a *lower bound* for probability of misclassification

In practice, $P(Y = k|X)$ is **unknown** — estimate it from training data.

Evaluating a Classifier

Training error rate: $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$ — **misclassification rate**

Disadvantage: We care little about how well the method works on training data. It should work well on **test data** that are not used in training of the method.

Test error rate: Ave$\{I(y_0 \neq \hat{y}_0)\}$ over test observations (x_0, y_0)

Note 1: We almost always expect the training error to be smaller than the test error.

Note 2:: Accuracy of a classifier is best measured using **test error rate**. We would like to use a method that minimizes it.

Training vs Test Error Rates

Fact: As model flexibility increases, the training error *decreases*, whereas the test error rate follows a *U shape*.

Q: Why do we observe a U shape for test error rate?

Fact: Bias and variance of $\hat{f}$ contribute to test error rate via its **mean squared error** (MSE), which equals

$$\text{MSE}(\hat{f}) = E\{(\hat{f} - f)^2\} = \{\text{bias}(\hat{f})\}^2 + \text{var}(\hat{f}).$$

Bias-variance tradeoff: As the model flexibility increases, $\{\text{bias}(\hat{f})\}^2$ decreases and $\text{var}(\hat{f})$ increases. However, they change at different rates, leading to a U shape for MSE, which in turn leads to a U shape for test error.

Overfitting:

- Small training error but large test error
- A less flexible model yields a smaller test error

KNN Revisited

- The number of neighbors K controls flexibility — affects the bias-variance tradeoff
- As K increases, flexibility decreases, implying that bias increases and variance decreases

Q: How to choose the optimal K?
A: Plot test error rate against $1/K$ — generally expect to see a U shape

Q: Parametric or nonparametric?
A: Nonparametric

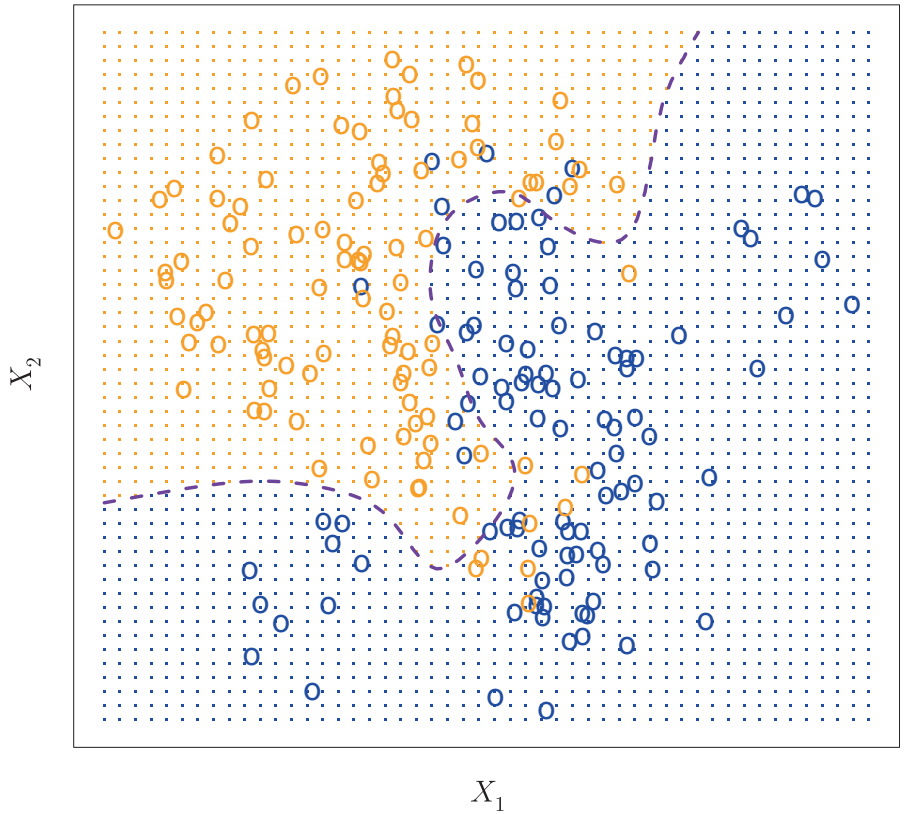


FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*

Source: ISL

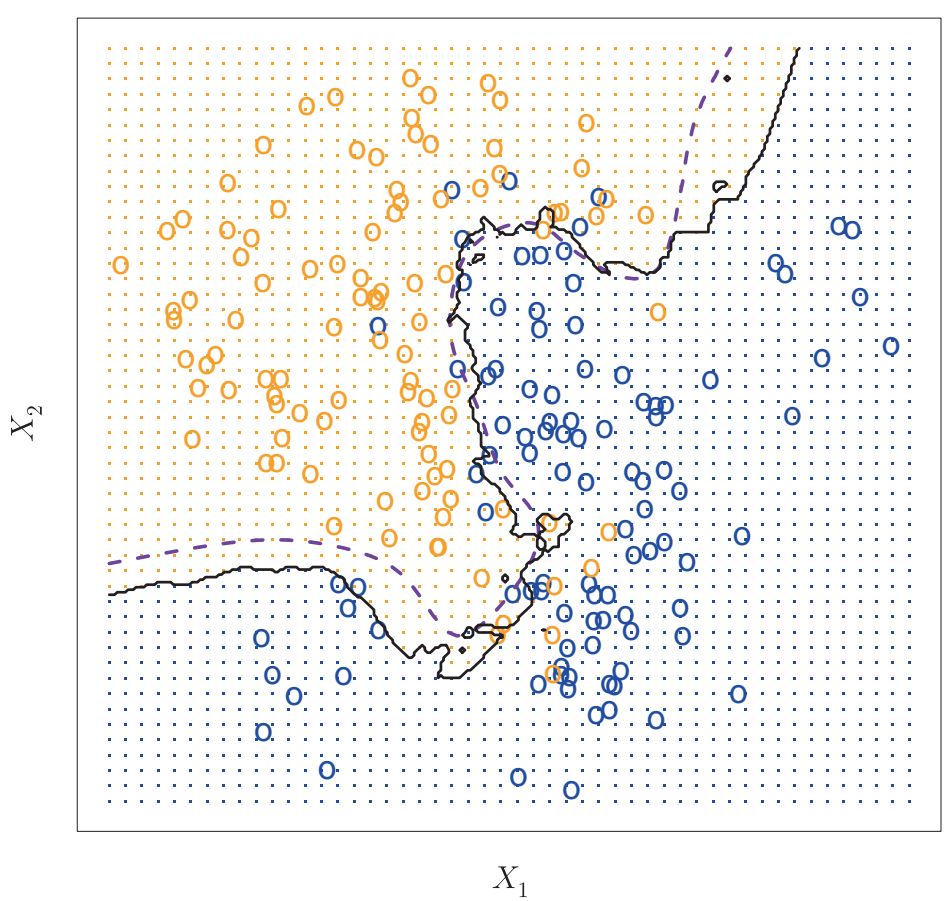


FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using $K = 10$. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

Source: ISL

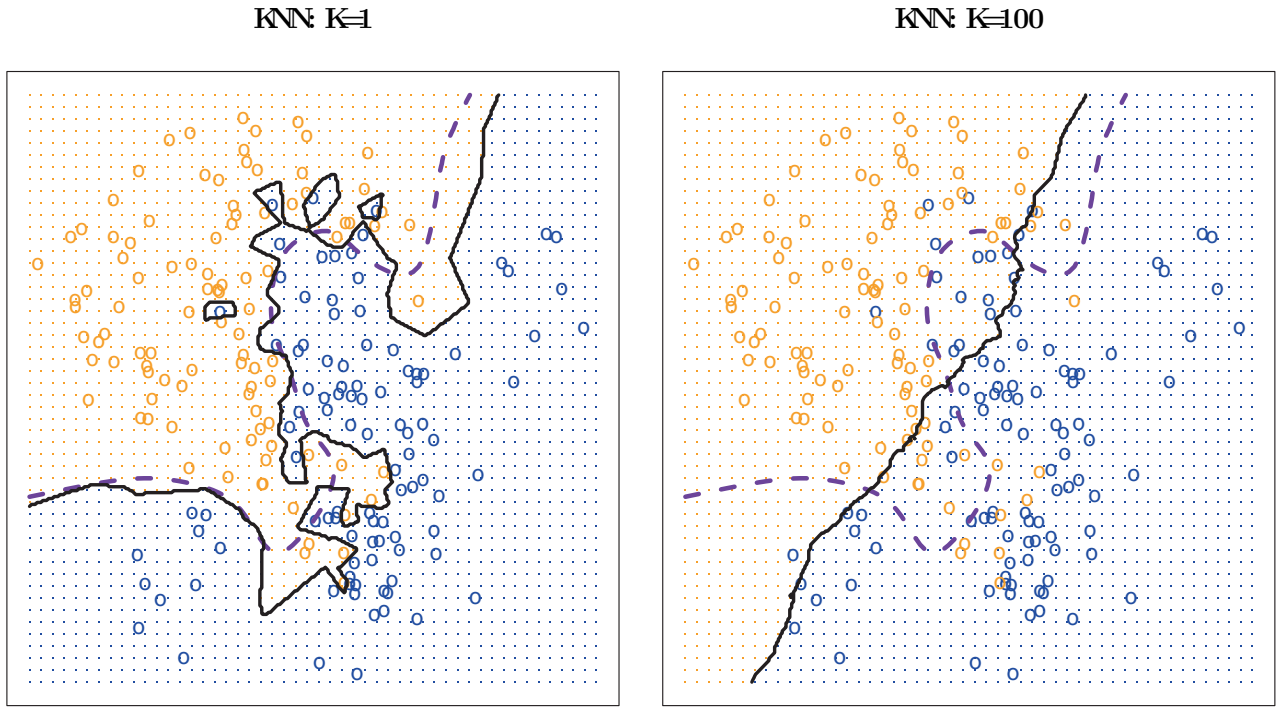


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple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

Source: ISL

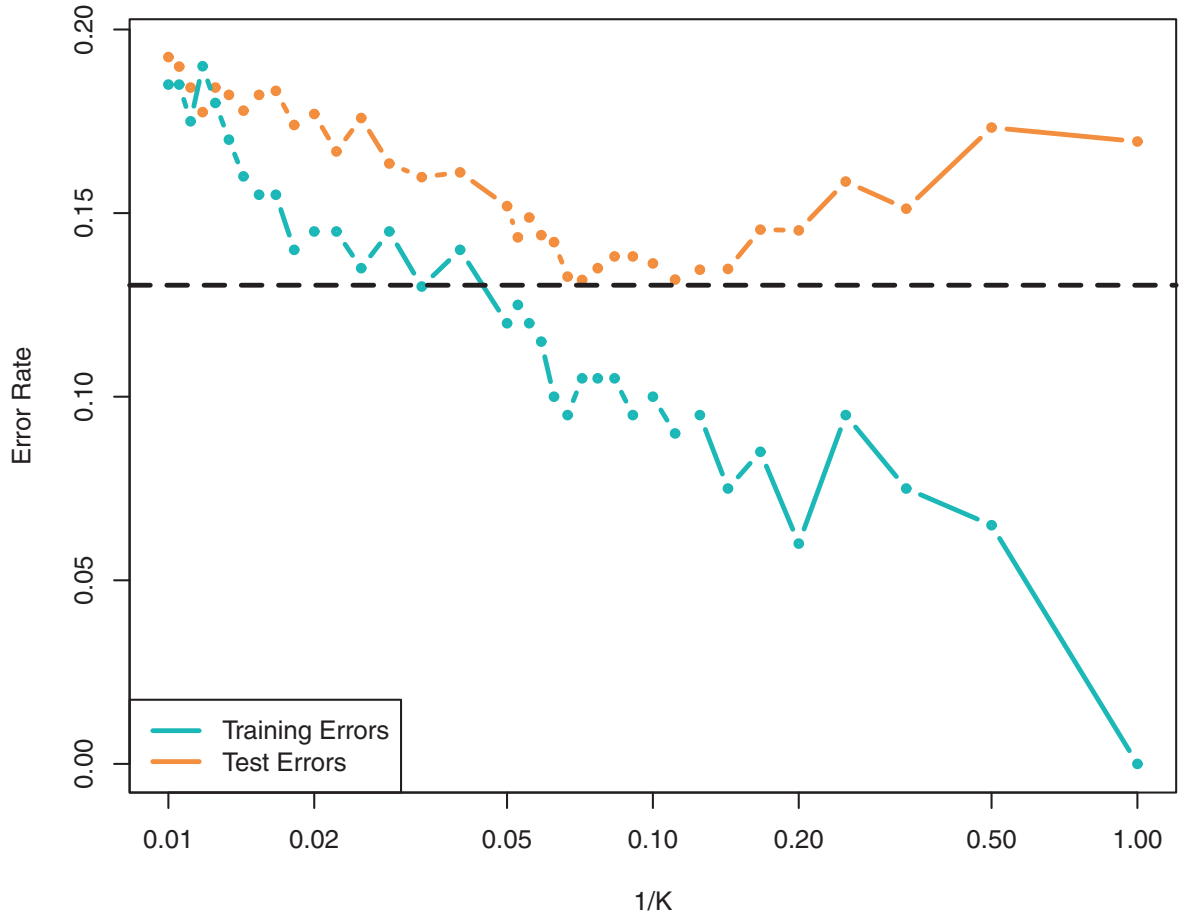


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

Source: ISL

Takeaways

- Prediction vs inference
- Parametric vs nonparametric method
- Overfitting
- Training error vs test error
- Due to the bias-variance tradeoff, as model flexibility increases, the training error rate decreases but the test error follows a U shape
- The U shape helps determine optimal amount of flexibility

Binary Classifier: Additional Performance Measures

Probability of misclassification, $P(\hat{Y} \neq Y)$ — the expected error rate that the Bayes classifier minimizes — can be written as

$$
\begin{aligned}
&P(\hat{Y} \neq Y, Y = 1) + P(\hat{Y} \neq Y, Y = 2) \\
&= P(\hat{Y} \neq Y | Y = 1)P(Y = 1) + P(\hat{Y} \neq Y | Y = 2)P(Y = 2) \\
&= P(\hat{Y} = 2 | Y = 1)P(Y = 1) + P(\hat{Y} = 1 | Y = 2)P(Y = 2) \\
&= P(\hat{Y} = 2 | Y = 1)\pi_1 + P(\hat{Y} = 1 | Y = 2)\pi_2
\end{aligned}
$$

- **Two class-specific errors**: Predict 1 as 2 and 2 as 1
- $P(\hat{Y} \neq Y)$ combines probabilities of the class-specific errors and the marginal proportions (**prevalence**) into one overall measure — can call it *total* **probability of misclassification**
- Total error may be low but class-specific errors may be high

Measures of Class-Specific Performance

Oftentimes, we can think of a classifier as a diagnostic test with

- Class 1: $+$ — aka **non-null** class (indicates a change from the normal state, e.g., a disease)
- Class 2: $-$ — aka **null** class (indicates no change from the normal state, e.g., no disease)

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N^*	P^*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

- aka **confusion matrix** — correct predictions on the diagonal; misclassifications on the off-diagonal

Source: ISL

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. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

- **True positive rate**: $P(\hat{Y} = +|Y = +)$ — **sensitivity**
- **True negative rate**: $P(\hat{Y} = -|Y = -)$ — **specificity**
- **False positive rate**: $P(\hat{Y} = +|Y = -)$
- **False positive rate + true negative rate = 1**, i.e., **false positive rate = 1− specificity**
- **False positive rate + false negative rate $\neq$ 1**
- **Class-specific error rates**: **1** − sensitivity and **1** − specificity
- Want both sensitivity and specificity to be high, or equivalently, the two error rates to be small.

Source: ISL

Returning to **probability of misclassification**, we see that:

$$\begin{aligned} P(\hat{Y} \neq Y) &= (1 - \text{sensitivity})\pi_+ + (1 - \text{specificity})\pi_- \\ &= \frac{\text{FN}}{\text{P}} \frac{\text{P}}{\text{N} + \text{P}} + \frac{\text{FP}}{\text{N}} \frac{\text{N}}{\text{N} + \text{P}} = \frac{\text{FN} + \text{FP}}{\text{N} + \text{P}}, \end{aligned}$$

which is as expected. Next, consider a general classifier that predicts class '+' if $p_+(x) \geq p_0$ and class '−' otherwise, where p_0 is a cutoff. Of course, the Bayes classifier uses $p_0 = 0.5$.

Q: What would be the sensitivity and specificity if $p_0 = 0$?
A: $p_0 = 0 \implies$ everyone is classified as '+'
$\implies$ sensitivity $= \text{TP}/\text{P} = 1$ and specificity $= \text{TN}/\text{N} = 0$

Q: What would be the sensitivity and specificity if $p_0 = 1$?
A: $p_0 = 1 \implies$ everyone is classified as '−'
$\implies$ sensitivity $= \text{TP}/\text{P} = 0$ and specificity $= \text{TN}/\text{N} = 1$

Tradeoff Between Sensitivity & Specificity

Fact: In general, as the cutoff $p_0 \uparrow 1$, the sensitivity $\downarrow 0$, and the specificity $\uparrow 1$.

- **tradeoff between sensitivity and specificity** — if one increases, the other decreases
- Effect of p_0 on probability of misclassification is not as clear cut as it would depend on the relative rates of change and also on the prevalence. However, for a Bayes classifier, this probability will be minimum when $p_0 = 0.5$.
- Can plot both sensitivity and specificity (or their one-minus versions) against the cutoff p_0 to see the class-specific performance of a classifier

Takeaways

- Confusion matrix
- Misclassification rate is an overall measure of error that combines class-specific error rates and prevalence in a single measure
- Class-specific performance can be evaluated using sensitivity and specificity
- Tradeoff between sensitivity and specificity
- The cutoff p_0 can be adjusted to achieve desired levels of sensitivity and specificity