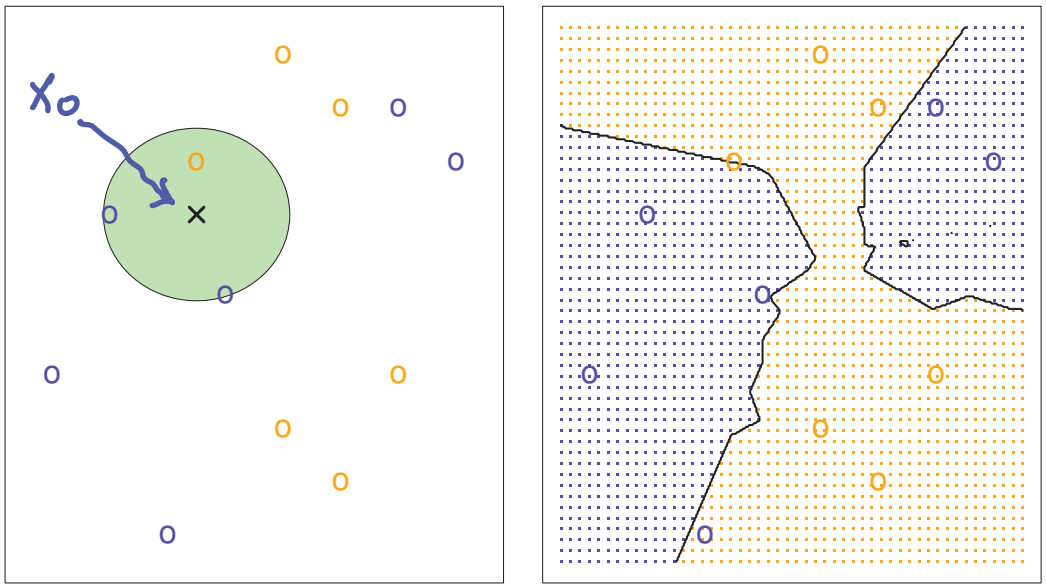
K-Nearest Neighbors

- 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.
- Therefore, the Bayes classifier serves as an unattainable gold standard against which to compare other methods.
- 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.
- 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. That is

$$\hat{Pr}(Y=j \mid X=x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

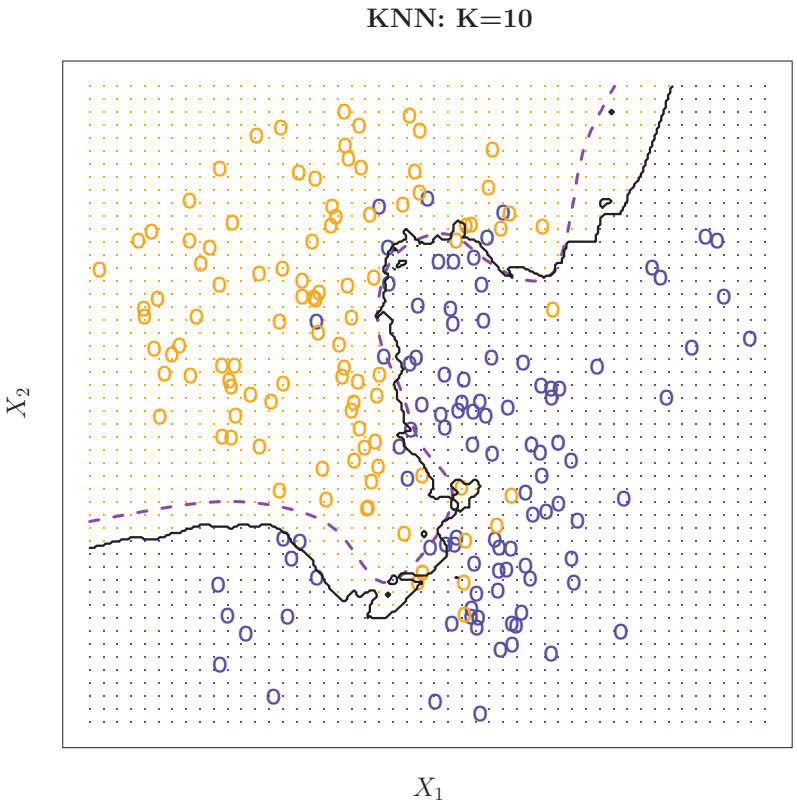
- Finally, KNN classifies the test observation x_0 to the class with the largest estimated probability.



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.

$\hat{Pr}(Y = \text{blue} \mid X = x_0) = \frac{2}{3}$, $\hat{Pr}(Y = \text{orange} \mid X = x_0) = \frac{1}{3}$

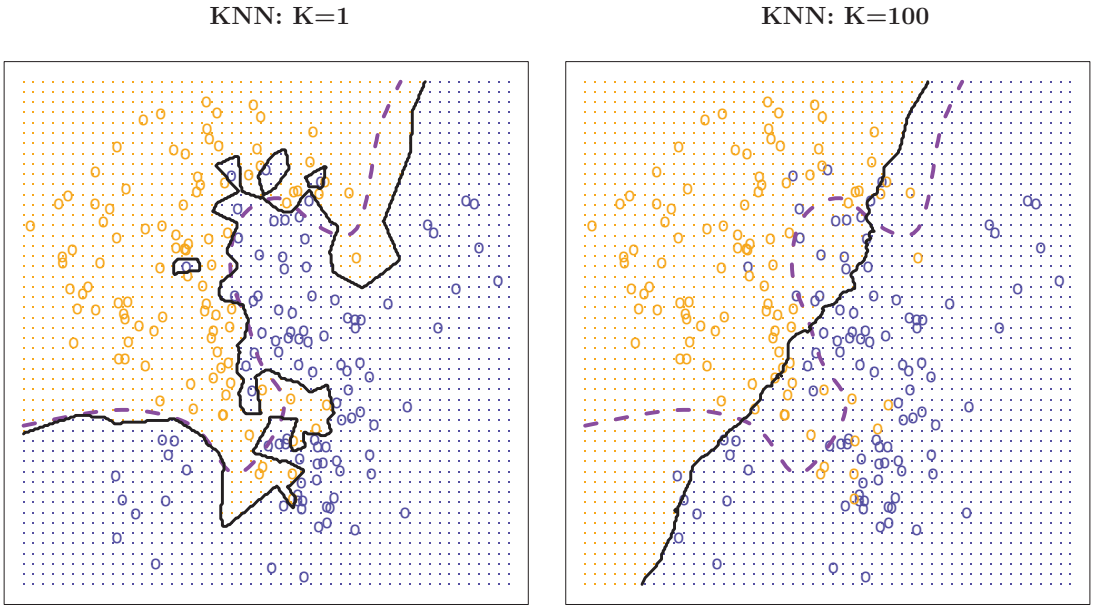
- Despite the fact that it is a very simple approach, KNN can often produce classifiers that are surprisingly close to the optimal Bayes classifier.



The black curve indicates the KNN decision boundary on the larger simulated data set, using $K = 10$. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.

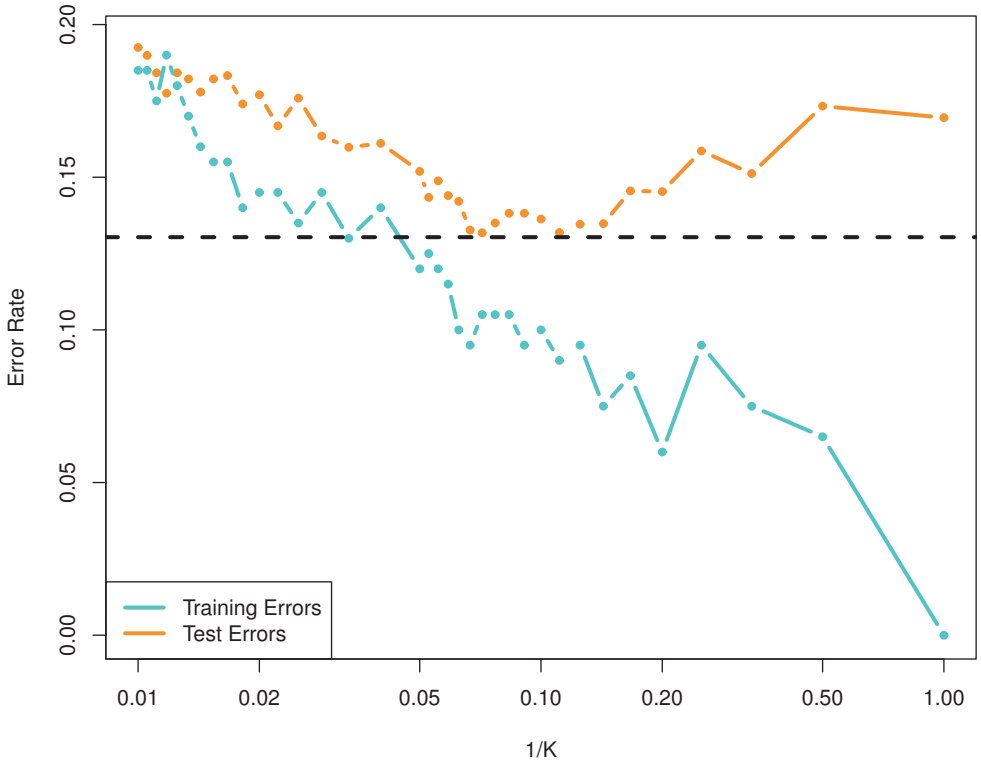
- The choice of K has a drastic effect on the KNN classifier obtained.
- When $K = 1$, the decision boundary is overly flexible and finds patterns in the data that don't correspond to the Bayes decision boundary.

- As K grows, the method becomes less flexible and produces a decision boundary that is close to linear.



A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the simulated data. 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.

- Just as in the regression setting, there is not a strong relationship between the training error rate and the test error rate.
- In general, as we use more flexible classification methods, the training error rate will decline but the test error rate may not.



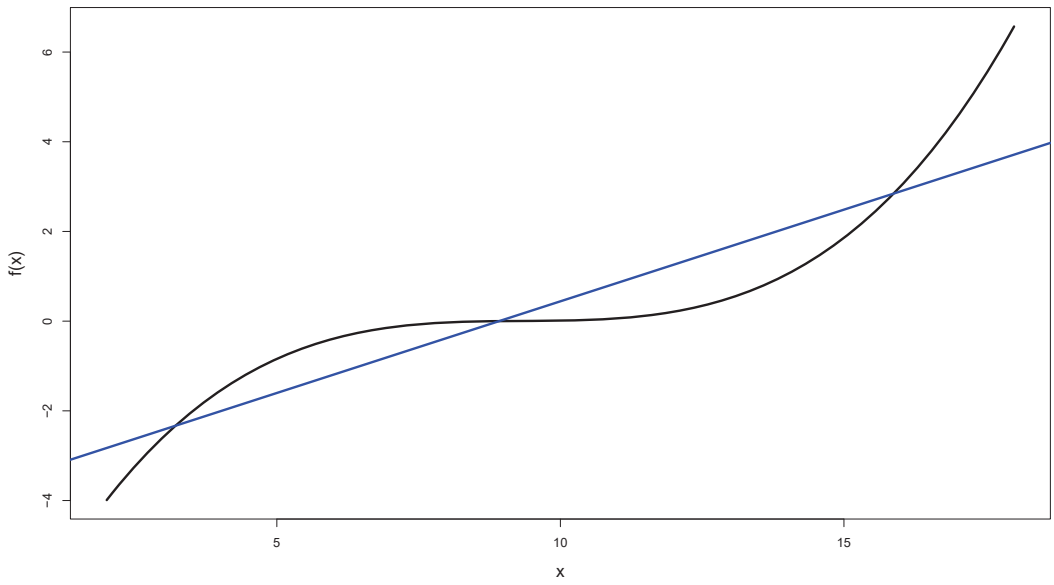
The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the simulated data, as the level of flexibility (assessed using $1/K$ on the log scale) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes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

Linear Regression

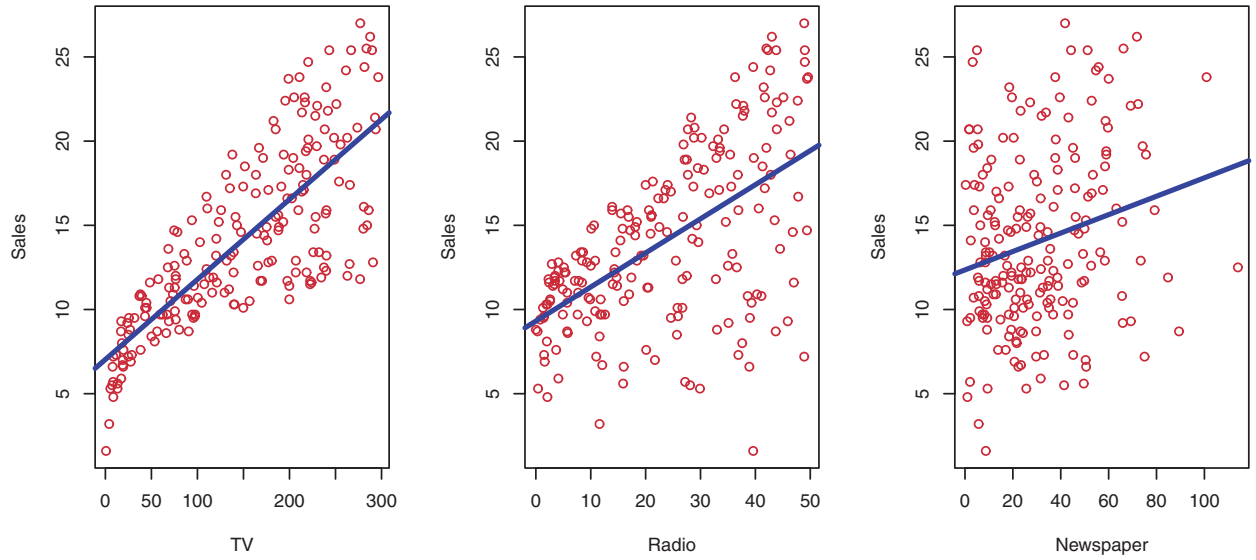
- Linear regression is a very simple approach for supervised learning and a useful tool for predicting a quantitative response.

- It assumes that the dependence of Y on $X_1, ..., X_p$ is linear.

- Although true regression functions are never linear, linear regression models are simple and often provide an adequate and interpretable description of how the inputs affect the output.



Consider the advertising data set:



The plot displays sales, in thousands of units, as a function of TV, radio, and newspaper budgets, in thousands of dollars, for 200 different markets. Each blue line represents a simple model (simple least squares fit of sales to that variable) that can be used to predict sales using TV, radio, and newspaper, respectively.

Questions we might seek to address:

1. Is there a relationship between advertising budget and sales?
2. How strong is the relationship between advertising budget and sales?
3. Which media contribute to sales?
4. How accurately can we estimate the effect of each medium on sales?
5. How accurately can we predict future sales?
6. Is the relationship linear?
7. Is there synergy among the advertising media?

Simple Linear Regression

- We assume a model

$$Y = \beta_0 + \beta_1 X + \epsilon,$$

random err. term independent of X, $E\epsilon = 0$

where β_0 and β_1 are two unknown constants (coefficients, or parameters) that represent the intercept and slope.

- Given some estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients, we predict future sales using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

where $\hat{y}$ indicates a prediction of Y on the basis of $X = x$. The *hat* symbol denotes an estimated value for an unknown parameter or coefficient, or to denote the predicted value of the response.

Estimation of the Parameters by Least Squares

- Let

$$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$$

represent n observation pairs, each of which consists of a measurement of X and a measurement of Y.

- Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, be the prediction for Y based on the ith value of X. Then $e_i = y_i - \hat{y}_i$ represents the ith *residual*.

- We define the *residual sum of squares* (RSS) as

$$\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2,$$

or equivalently as

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

- The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize the RSS.

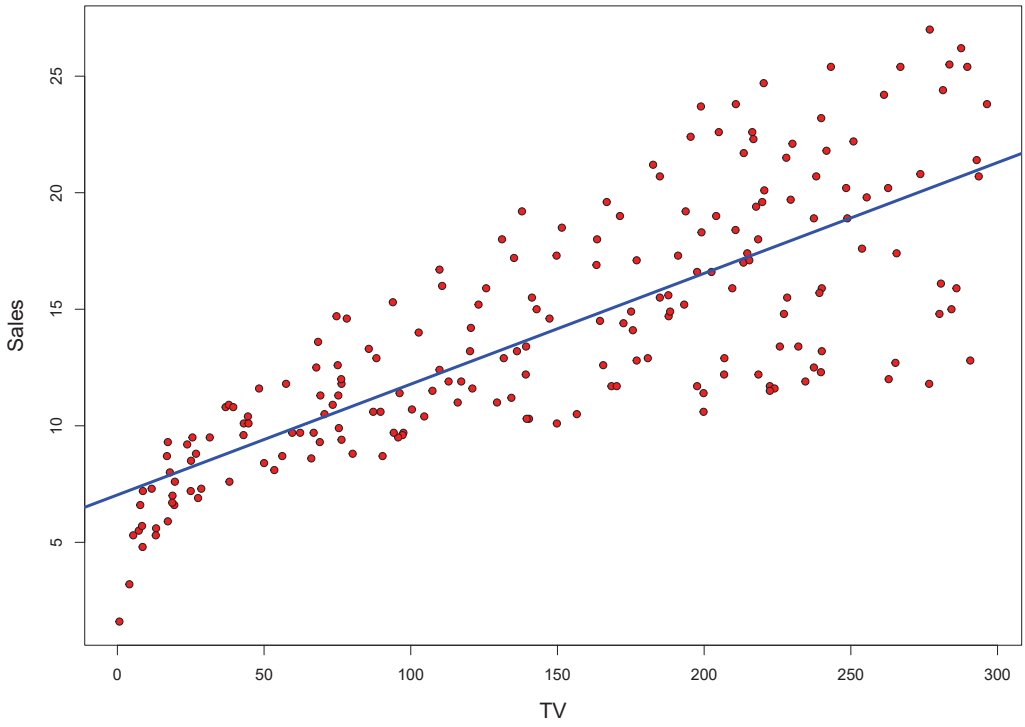
one can show that the minimizers are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ are the sample means.

Example: Advertising data.

```
> Adv <- read.csv("Advertising.csv", header=T)
> attach(Adv)
> lm_fit=lm(sales ~ TV, data=Adv)
> plot(TV,sales, xlab="TV", ylab="Sales", pch = 21, bg="red", cex.lab=1.3)
> abline(lm_fit, col="blue", lwd = 3)
```



Least squares fit for the regression of sales onto TV. In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.

```
> names(lm_fit)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

> lm_fit$coefficients
(Intercept)          TV
 7.03259355  0.04753664

> lm_fit$residuals
          1           2           3           4           5
 4.12922549  1.25202595  1.44977624  4.26560543 -2.72721814
          6           7           8           9          10
-0.24616232  2.03404963  0.45350227 -2.64140866 -5.93041431
         11          12          13          14          15
-1.57476548  0.16128975  1.03603441 -1.96741599  2.26517814
```