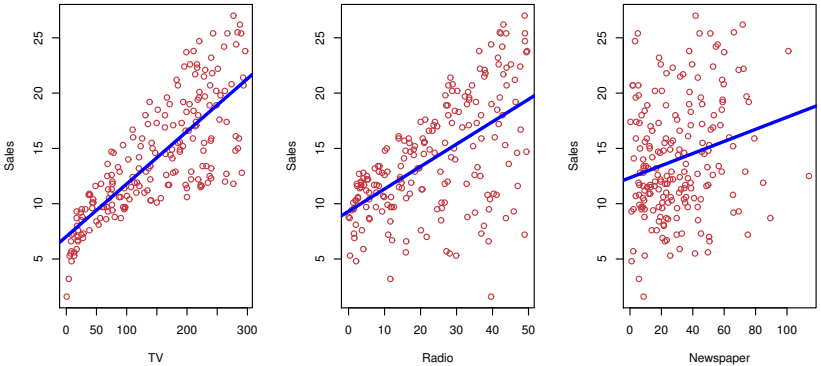
STAT318/462 — Data Mining

Dr Gábor Erdélyi

University of Canterbury, Christchurch, New Zealand

Course developed by Dr B. Robertson. Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.

What is Statistical Learning?



- Shown are `Sales` vs. `TV`, `Radio` and `Newspaper`. The blue line is a linear regression fit.

- **Question:** Can we predict `Sales` using these three predictors?

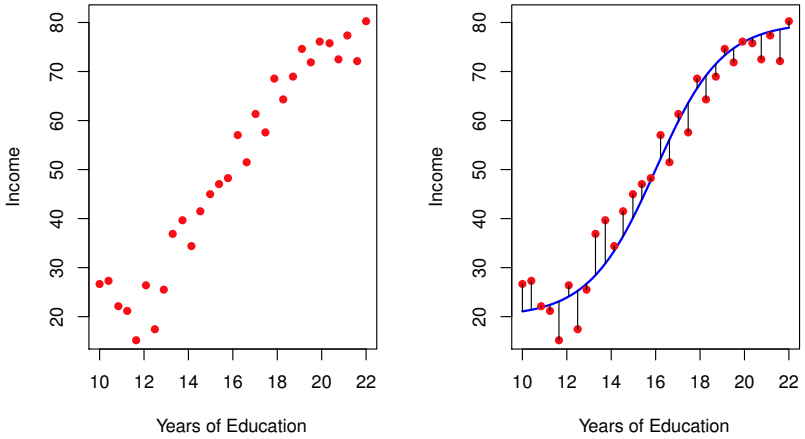
Getting Started

- Suppose we have a **quantitative** response Y (regression problem) and p different predictors,
$$X = (X_1, X_2, \ldots, X_p).$$

- We assume there is a relationship between Y and X of the form
$$Y = f(X) + \epsilon,$$
where ϵ is a random error term which is independent of X and has mean $E(\epsilon) = 0$.

- Essentially, we are interested in approaches for estimating f (statistical learning).

Simulated Example

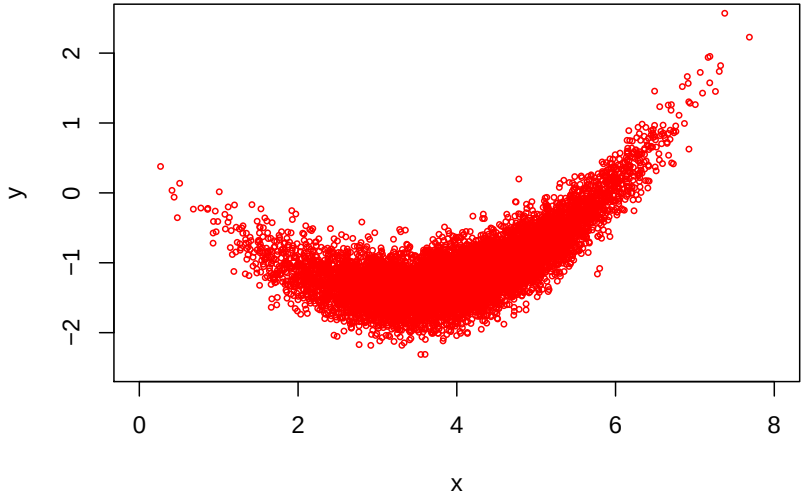


- Shown are `Income` vs. `Years of Education`. The blue line is the true underlying relationship, $f(X)$.

What Is $f(X)$ Good For?

- We can use a good f to make predictions of Y at unseen test cases $X = \mathrm{x}$.
- We can determine which predictors are important in explaining Y, and those that are not.
- Potentially understand how each predictor affects Y.

Is There an Ideal $f(X)$?



- There can be many Y values at $X = \text{x}$.

Is There an Ideal $f(X)$?

- The ideal $f(X)$ at $X = \mathrm{x}$ is

$$f(\mathrm{x}) = E(Y|X = \mathrm{x})$$

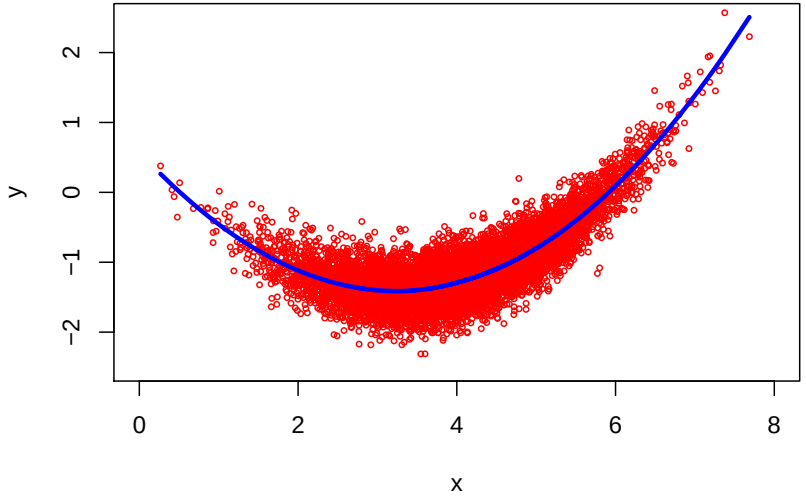
and is called the **regression function**.

- This function is ideal with respect to mean-squared prediction error ($E(Y - f(X))^2$). To be specific, it minimizes

$$E\left[(Y - g(X))^2|X = \mathrm{x}\right]$$

over all functions g at all points $X = \mathrm{x}$.

The Ideal $f(X)$



- The regression function, $f(\mathrm{x}) = E(Y|X = \mathrm{x})$.

The Ideal $f(X)$

- Even if we knew $f(\mathrm{x})$, we would still make errors in prediction because there is typically a distribution of Y values at each $X = \mathrm{x}$.

- This error is called **irreducible error**, given by

$$\epsilon = Y - f(\mathrm{x}).$$

- Suppose we have an estimate $\hat{f}(\mathrm{x})$ for $f(\mathrm{x})$, then we can decompose the average mean-squared error as follows:

$$E\left[(Y - \hat{f}(X))^2 | X = \mathrm{x}\right] = \underbrace{[f(\mathrm{x}) - \hat{f}(\mathrm{x})]^2}_{\text{(reducible)}} + \underbrace{V(\epsilon)}_{\text{(irreducible)}}$$

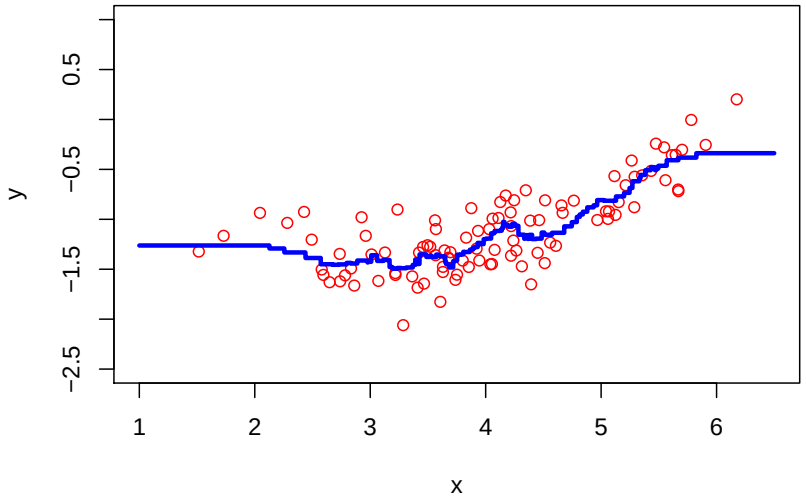
How Do We Estimate $f(x)$?

- We cannot compute $E(Y|X = x)$ directly because we don't know the conditional distribution of Y given X.
- At best, we have a few observations of Y near $X = x$.
- One approach is to relax the definition and let

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x)),$$

where $\mathcal{N}(x)$ is a neighbourhood of x.

Nearest Neighbour Averaging



- A sample of $n = 100$ (averaging using 10 neighbours).

Nearest Neighbour Averaging

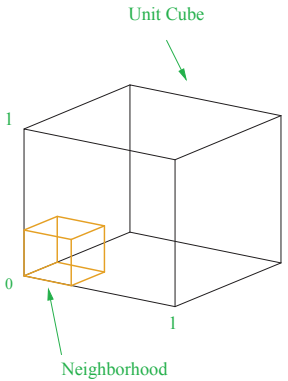
- Problem solved, we have a good estimate for the regression function! **Or do we?**

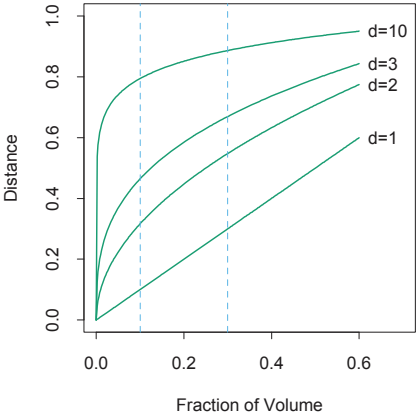
  1. If p (dimension) is small and n (sample size) is large, nearest neighbour averaging works reasonably well!

  2. However, smoother versions exist in this case (e.g. splines).

  3. If p is large, nearest neighbour averaging tends to perform poorly because 'nearest neighbours' tend to be far away. This problem is called the **curse of dimensionality**.

Curse of Dimensionality





- Sub-cube neighbourhood in the unit cube.

Parametric and Structured Models

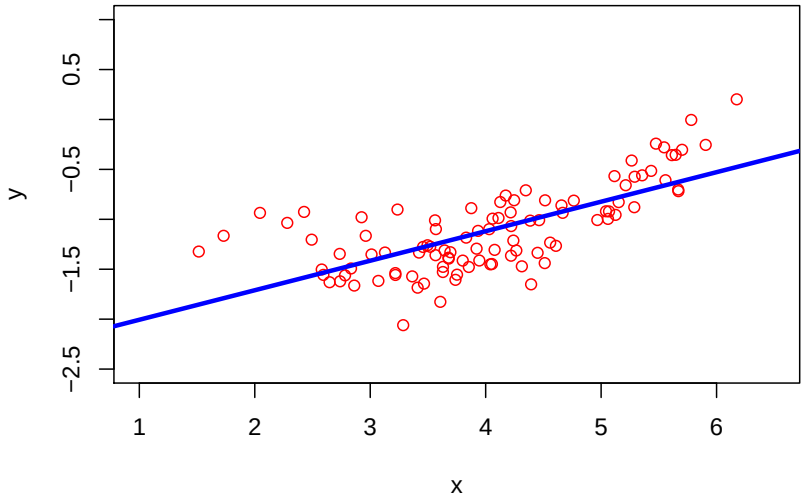
- Make an assumption about the functional form (shape) of f.

- A simple and extremely useful parametric model is the linear model:

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p.$$

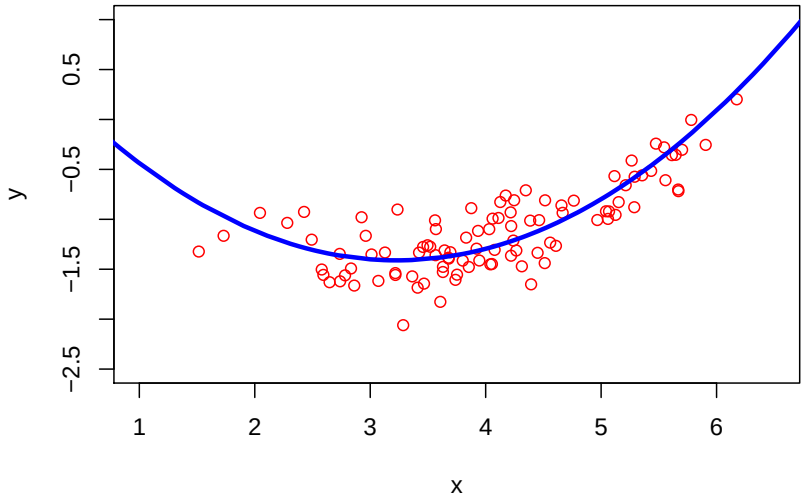
1. $p + 1$ parameters: $\beta_0, \beta_1, \ldots, \beta_p$.
2. Estimate the parameters by fitting the model to training data.
3. *"all models are wrong, but some are useful" (George Box).*

Linear Model



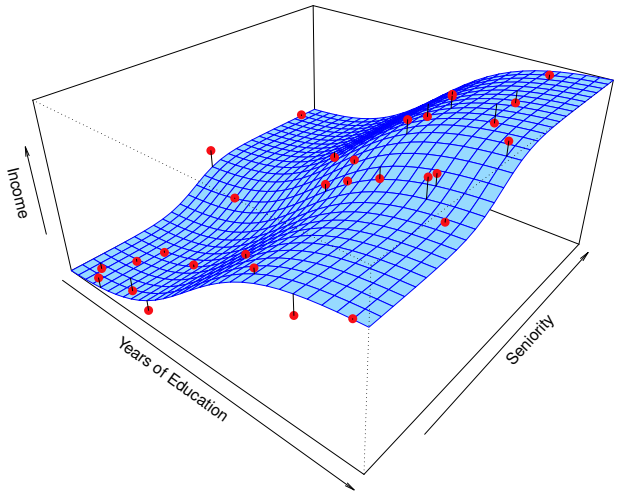
- $\hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X.$

Quadratic Model



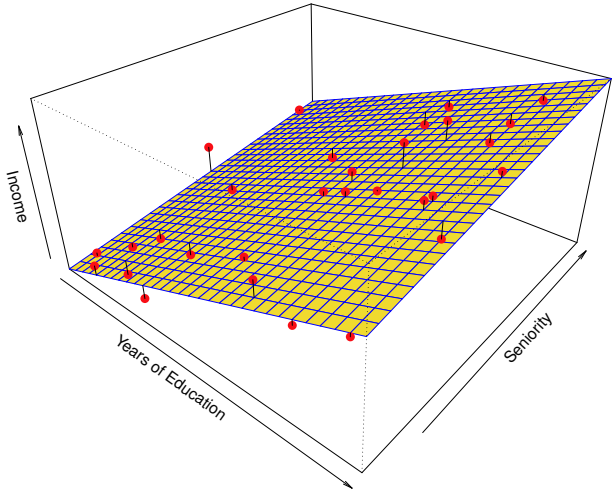
- $\hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$.

Simulated Model



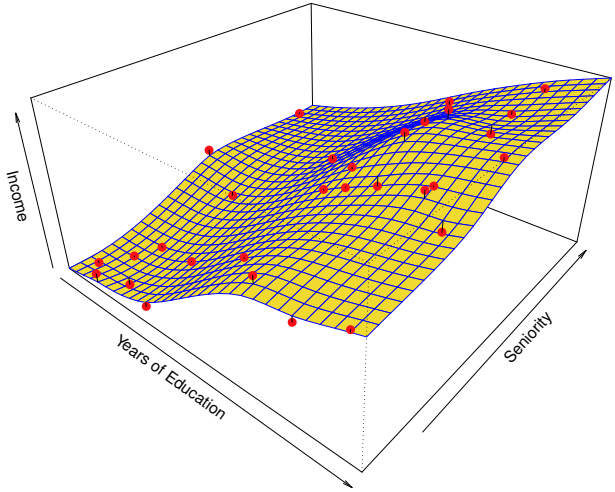
- $\texttt{Income} = f(\texttt{Education}, \texttt{Seniority}) + \epsilon$.

Parametric Model: Linear Model

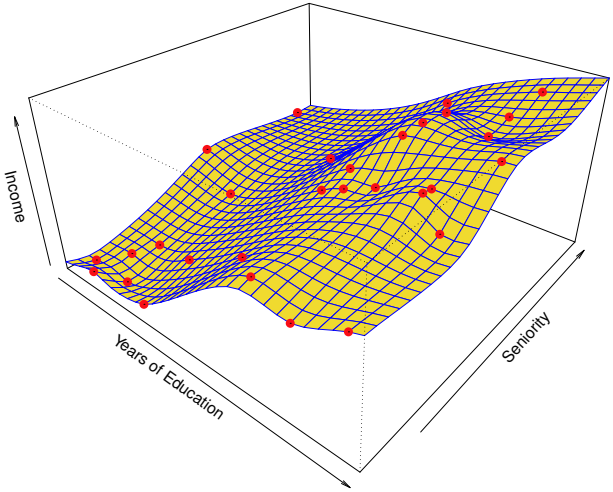


- $\texttt{Income} \approx \hat{\beta}_0 + \hat{\beta}_1 \texttt{Education} + \hat{\beta}_2 \texttt{Seniority}$.

Non-Parametric Model: Thin-Plate Spline



Non-Parametric Model: Thin-Plate Spline



Some Trade-offs

1. Interpretability vs. prediction accuracy.
2. Good-fit vs. over-fit or under-fit.
3. Black-box vs. parsimony.

"Everything should be made as simple as possible, but not simpler." (Einstein)

Assessing Model Accuracy

- There is no *free lunch* in statistical learning meaning that there is no *best* method for all problems. We need a way of determining how well each method performs.

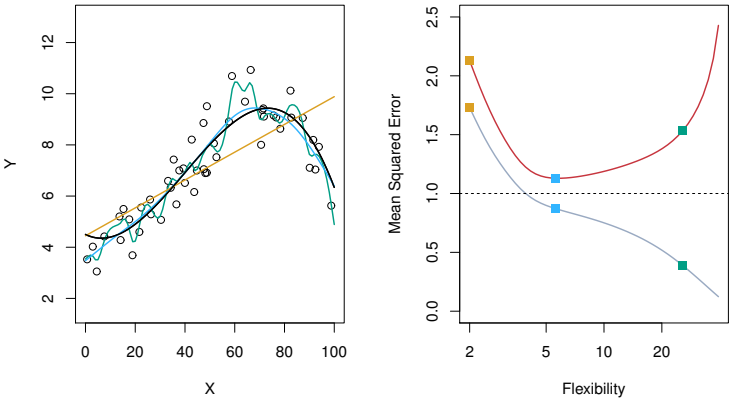
- Suppose we fit a model $\hat{f}(\mathrm{x})$ to some training data $\mathrm{Tr} = \{\mathrm{x}_i, y_i\}_{i=1}^{n}$. From this set we compute the **training mean-squared error**:

$$\mathrm{MSE}_{\mathrm{Tr}} = \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{f}(\mathrm{x}_i)]^2.$$

- Suppose we have **unseen** test data $\mathrm{Te} = \{\mathrm{x}_i, y_i\}_{i=1}^{m}$. From this set we compute the **testing mean-squared error**:

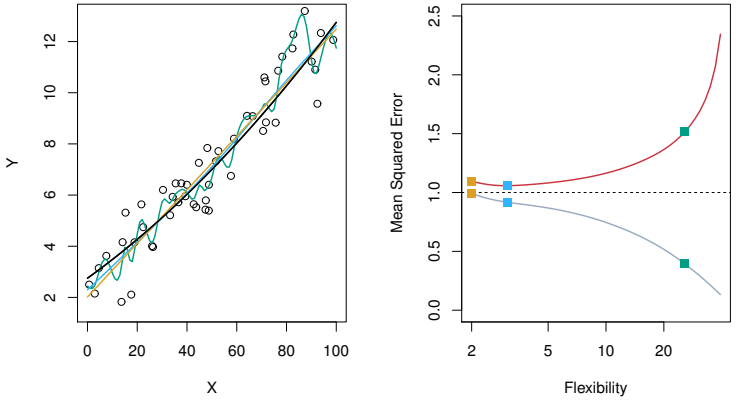
$$\mathrm{MSE}_{\mathrm{Te}} = \frac{1}{m}\sum_{i=1}^{m}[y_i - \hat{f}(\mathrm{x}_i)]^2.$$

Assessing Model Accuracy



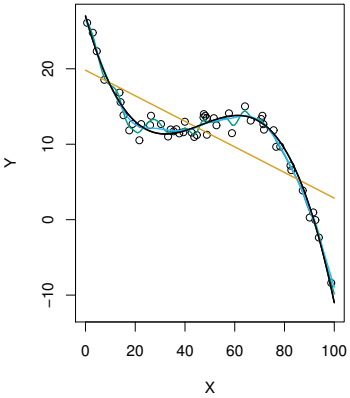
- $f(X)$ is black, MSE_{Tr} is grey, MSE_{Te} is red, and three models are shown in the other colours (yellow is linear, blue and green are splines).

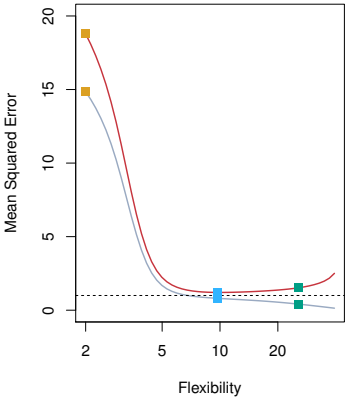
Assessing Model Accuracy



- $f(X)$ is black, MSE_{Tr} is grey, MSE_{Te} is red, and three models are shown in the other colours (yellow is linear, blue and green are splines).

Assessing Model Accuracy





- $f(X)$ is black, MSE_{Tr} is grey, MSE_{Te} is red, and three models are shown in the other colours (yellow is linear, blue and green are splines).

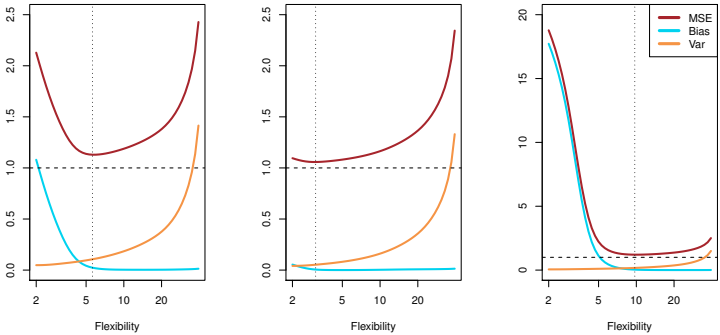
Bias-Variance Trade-off

- Suppose we fit a model to some training data and let (x_0, y_0) be a test observation.

- If the true model is
$$Y = f(X) + \epsilon,$$
with regression function $f(x) = E(Y|X = x)$, then the expected test MSE at $X = x_0$ is
$$E[y_0 - \hat{f}(x_0)]^2 = V(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + V(\epsilon).$$

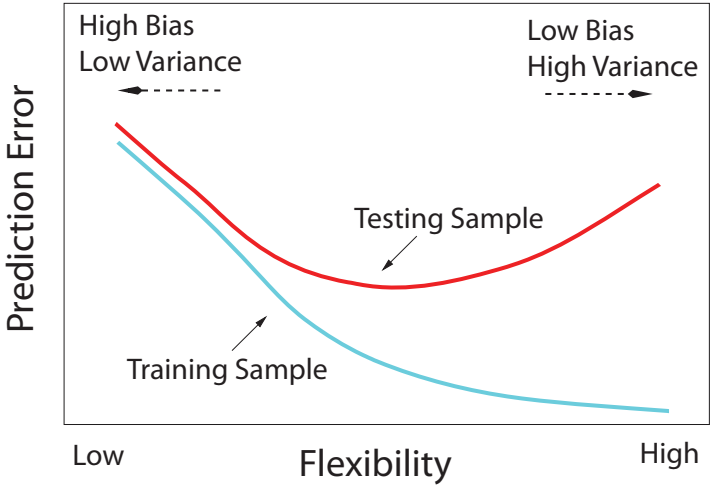
- Typically, as the flexibility of $\hat{f}$ increases, its bias decreases and its variance increases — a **bias-variance trade-off**.

Bias-Variance Trade-off



- The bias-variance decomposition for slides 23,24 and 25. The vertical dashed line indicates the flexibility level corresponding to the smallest test mean-squared error.

Bias-Variance Trade-off



Classification Problems

- Suppose we have a **qualitative** response Y (classification problem) that takes values from a finite (unordered) set

$$\mathcal{C} = \{y_1, y_2, \ldots, y_k\},$$

and p different predictors,

$$X = (X_1, X_2, \ldots, X_p).$$

- Our goal is to build a **classifier** $f(X)$ that assigns a **class label** from $\mathcal{C}$ to an unclassified X.

- We need to quantify uncertainty in each classification and understand how different predictors affect classifications.

Assessing Classification Accuracy

- The most common approach to assess accuracy is the **error rate** (zero-one loss function).

- Suppose we fit a classifier $\hat{f}(\mathrm{x})$ to some training data $\mathrm{Tr} = \{\mathrm{x}_i, y_i\}_{i=1}^{n}$. From this set we compute the **training error rate**:

$$\mathrm{ER}_{\mathrm{Tr}} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i),$$

  where $I(.)$ is an indicator variable that equals one if the condition is true (in this case, if $y_i \neq \hat{y}_i$) and zero otherwise.

- Suppose we have **unseen** test data $\mathrm{Te} = \{\mathrm{x}_i, y_i\}_{i=1}^{m}$. From this set we compute the **testing error rate**:

$$\mathrm{ER}_{\mathrm{Te}} = \frac{1}{m}\sum_{i=1}^{m} I(y_i \neq \hat{y}_i).$$

Is There an Ideal Classifier?

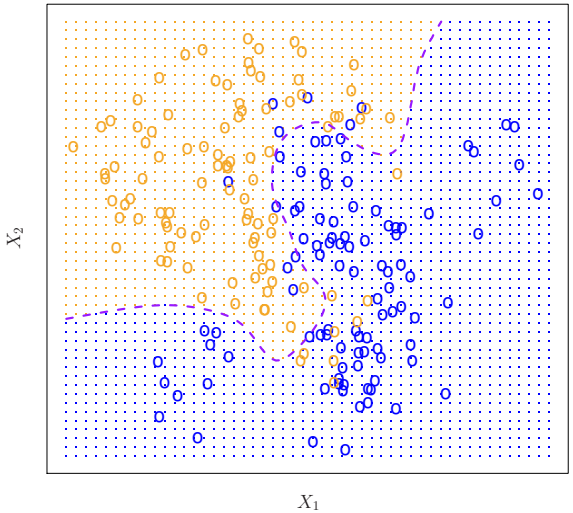
- Just as in the regression setting, there can be multiple classifications at $X = \mathrm{x}$.
- The ideal $f(X)$ at $X = \mathrm{x}$ is

$$f(\mathrm{x}) = y_k \quad \textbf{if} \quad \Pr(Y = y_k | X = \mathrm{x}) = \max_{y_j \in \mathcal{C}} \Pr(Y = y_j | X = \mathrm{x})$$

  and is called **Bayes classifier**. That is, assign each observation to its most likely class given its predictor values.

- Bayes classifier is ideal in the sense that it minimizes the expected prediction error with a zero-one loss function.

Bayes Classifier



- A simulated data set consisting of two groups and the Bayes decision boundary.

Bayes Classifier

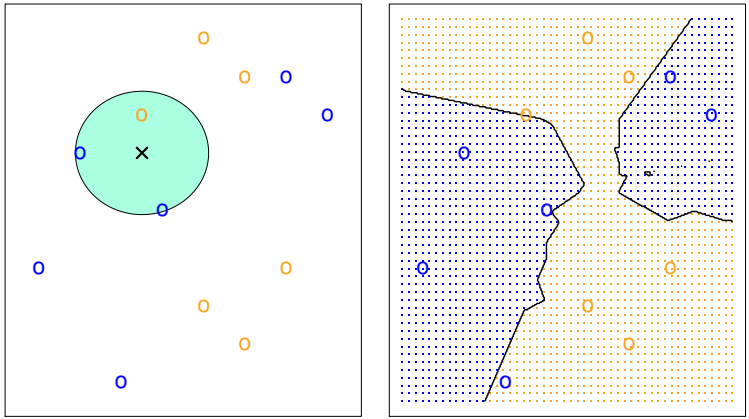
- Unfortunately, for real-world problems, we do not know the conditional distribution of Y given X so it is impossible to compute Bayes classifier.
- We can estimate the conditional distribution using nearest neighbour averaging (as we did for regression), provided we have enough local information.
- The simplest method is the k-**nearest neighbours** (KNN) classifier.

KNN Classifier

- **Goal:** Classify a test observation x_0 to a class from $\mathcal{C} = \{y_1, y_2, \ldots, y_N\}$.

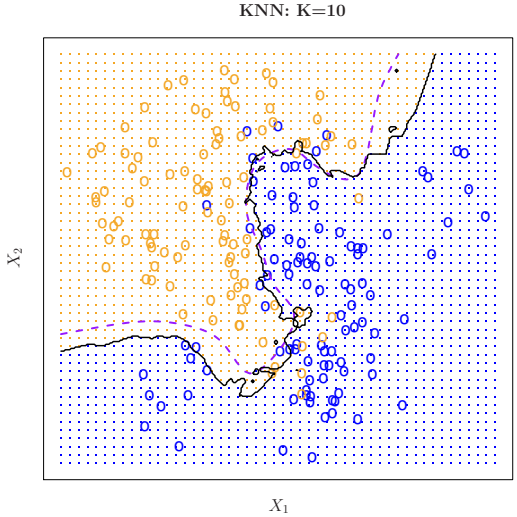
- **Steps:**
  1. Choose a positive integer k.
  2. Find the k nearest training observations to x_0. Call this set of observations $\mathcal{N}_0$.
  3. For $j = 1, 2, \ldots, N$, estimate the probability of being in class y_j using
  $$\widehat{\Pr}(Y = y_j | X = x_0) = P_j = \frac{1}{k} \sum_{i: x_i \in \mathcal{N}_0} I(y_i = y_j).$$
  4. Classify x_0 to class y_i, where $P_i \geq P_j$ for $j = 1, 2, \ldots, N$ (the majority class in $\mathcal{N}_0$).

KNN Classifier



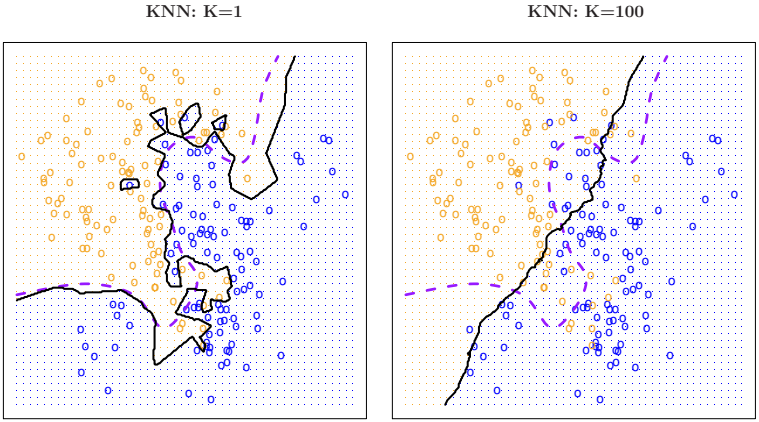
- KNN decision boundary when $k = 3$ is used.

KNN Classifier



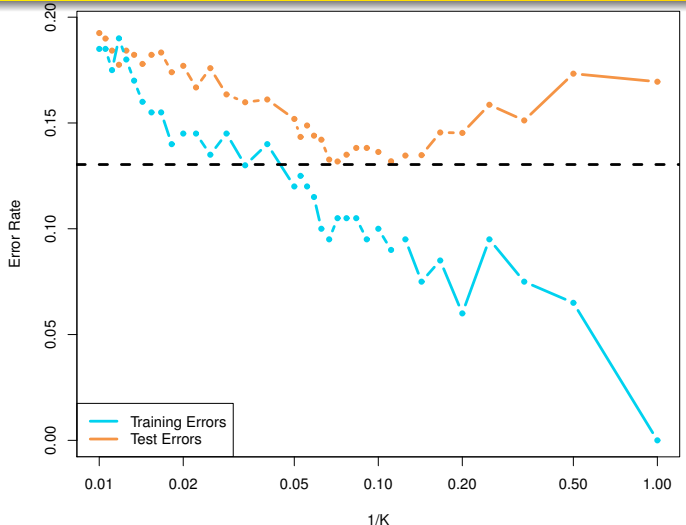
- $k = 10$ gives a very good approximation to the Bayes decision boundary.

KNN Classifier



- Voronoi Tessellation ($k = 1$) over-fit vs. $k = 100$ under-fit.

KNN Classifier



- The KNN training error rate decreases as the number of neighbours decreases and the characteristic U shape is clear in the testing error rate.

Classification Problems

- Just as in the regression setting, localized classification methods start to break down as dimension increases (curse of dimensionality).

- We can make an assumption about the functional form of the decision boundary

  e.g. linearly separable;

  or make an assumption about class distributions

  e.g. normally distributed.

- We will consider a variety of parametric and non-parametric classification methods in this course.