



MONASH University

ETC3250

Business Analytics

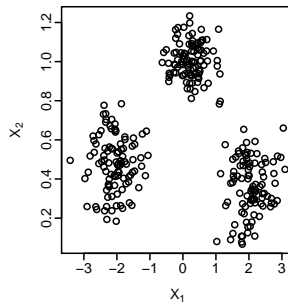
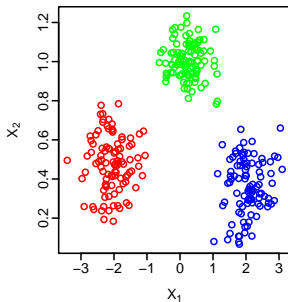
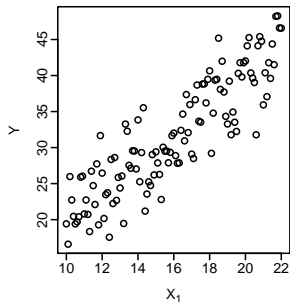
Week 2.
Statistical learning

27 July 2017

Learning from data

- **Better understand** or **make predictions** about a certain phenomenon under study
- **Construct a model** of that phenomenon by finding relations between several variables
- If phenomenon is complex or depends on a large number of variables, an **analytical solution** might not be available
- However, we can **collect data** and learn a model that **approximates** the true underlying phenomenon

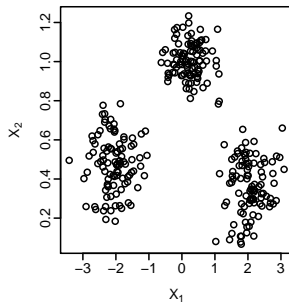
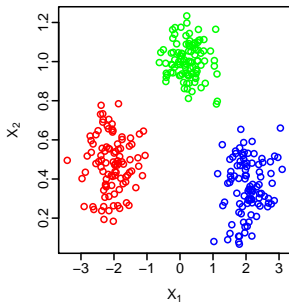
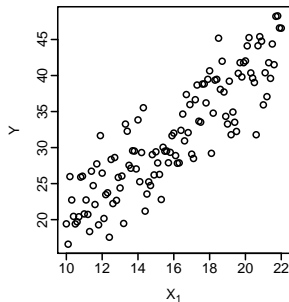
Learning from a dataset



$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \text{ with } x_i = (x_{i1}, \dots, x_{ip})^T$$

Statistical learning provides a framework for constructing models from \mathcal{D} .

Learning from a dataset



$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \text{ with } x_i = (x_{i1}, \dots, x_{ip})^T$$

Statistical learning provides a framework for constructing models from \mathcal{D} .

Different learning problems

- Supervised learning
 - Regression (or prediction)
 - Classification

→ y_i **available for all** x_i
- Unsupervised learning

→ y_i **unavailable for all** x_i
- Semi-supervised learning

→ y_i **available only for few** x_i
- Other types of learning: reinforcement learning, online learning, active learning, etc.

Identification of the best learning problem is important in practice

Supervised learning

$$\mathcal{D} = \{(y_i, \mathbf{x}_i)\}_{i=1}^N,$$

where

$$(y_i, \mathbf{x}_i) \sim P(Y, \mathbf{X}) = P(\mathbf{X}) \underbrace{P(Y|\mathbf{X})}.$$

- Y : response (output)
- $\mathbf{X} = (X_1, \dots, X_p)$: set of p predictors (input)

We seek a function $g(\mathbf{X})$ for predicting Y given values of the input \mathbf{X} . This function is computed using \mathcal{D} .

Supervised learning

We often assume that our data arose from a statistical model

$$Y = f(\mathbf{X}) + \varepsilon,$$

where f is the true unknown function, ε is the random error term with $E[\varepsilon] = 0$ and is independent of \mathbf{X} .

- The additive error model is a useful approximation to the truth
- $f(x) = E[Y|\mathbf{X} = \mathbf{x}]$
- Not a deterministic relationship: $Y = f(\mathbf{X})$

Supervised learning

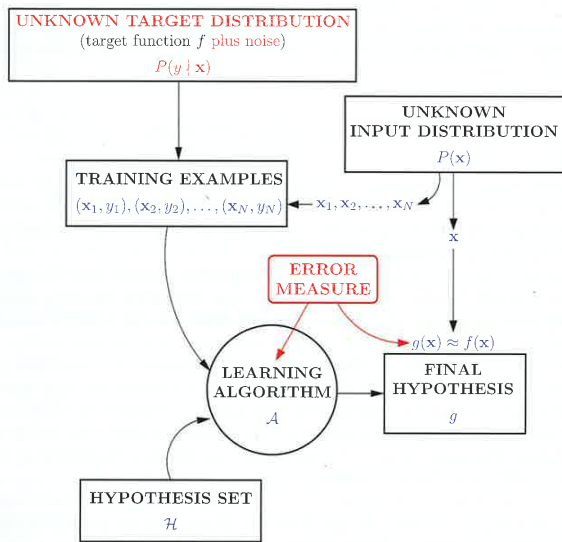
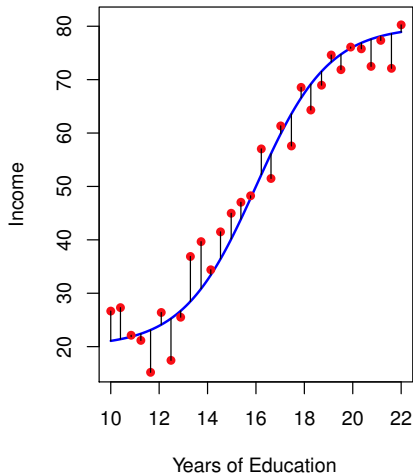
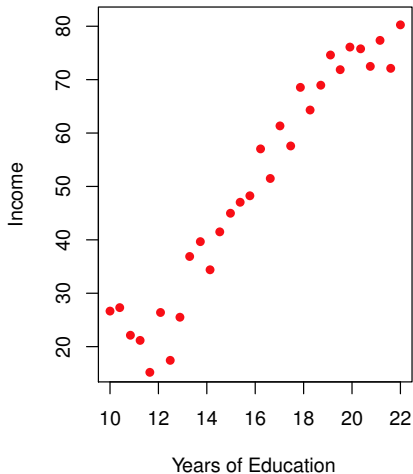
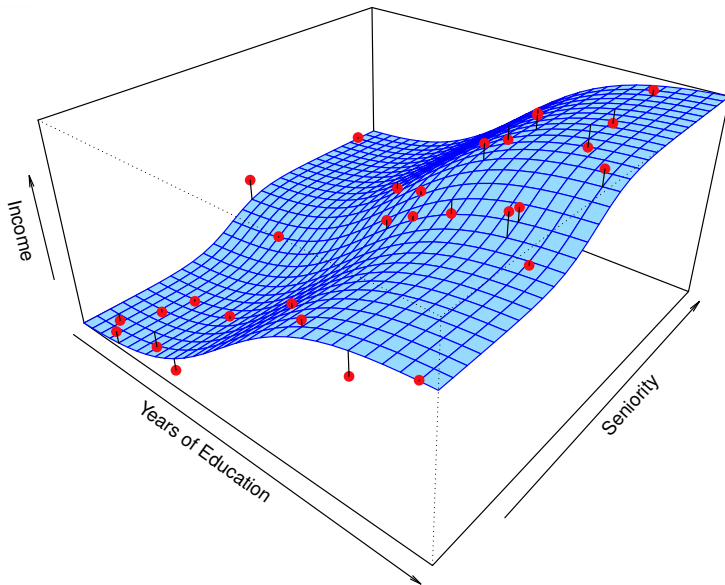


Figure 1.11: The general (supervised) learning problem

Supervised learning - regression



Supervised learning - regression



Why estimate f ?

■ Prediction:

- $\hat{Y} = \hat{f}(X)$

Error decomposition in regression:

$$\begin{aligned} E[(Y - \hat{Y})^2] &= E[(f(X) + \varepsilon - \hat{Y})^2] \\ &= \underbrace{E[(f(X) - \hat{f}(X))^2]}_{\text{Reducible}} + \underbrace{\text{Var}(\varepsilon)}_{\text{Irreducible}} \end{aligned}$$

■ Inference (or explanation):

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?

How do we estimate f ?

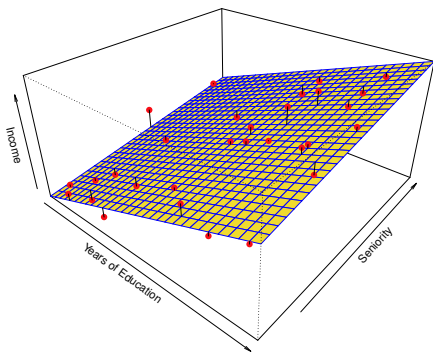
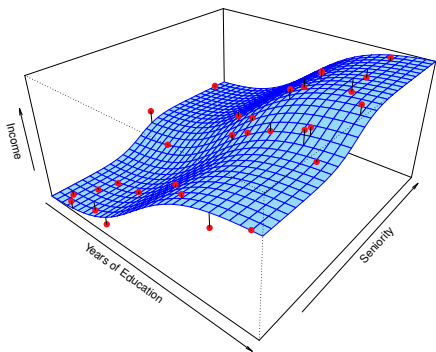
■ Parametric methods

- Assumption about the form of f , e.g. linear:
 $f(X) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$ and $\hat{Y}(x) = \hat{f}(x)$
- 😊 The problem of estimating f reduces to estimating a set of parameters
- 😊 Usually a good starting point for many learning problems
- 😞 Poor performance if linearity assumption is wrong

■ Non-parametric methods

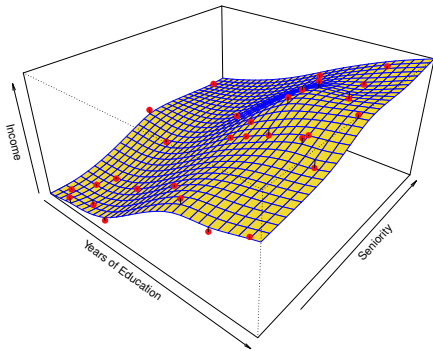
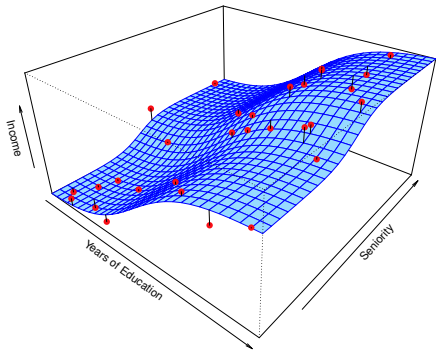
- No *explicit* assumptions about the form of f , e.g. nearest neighbours: $\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$
- 😊 High flexibility: it can potentially fit a wider range of shapes for f
- 😞 A large number of observations is required to estimate f with good accuracy

Regression - estimation of f ?

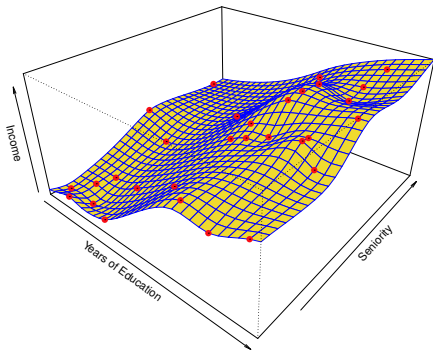
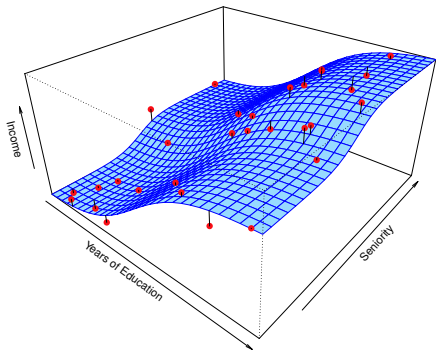


$$\hat{f}(\text{education}, \text{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \text{education} + \hat{\beta}_2 \times \text{seniority}$$

Regression - estimation of f ?



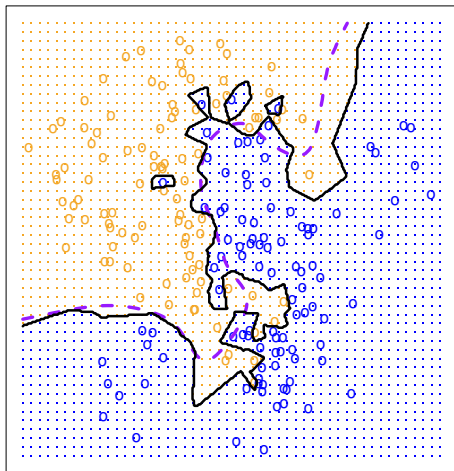
Regression - estimation of f ?



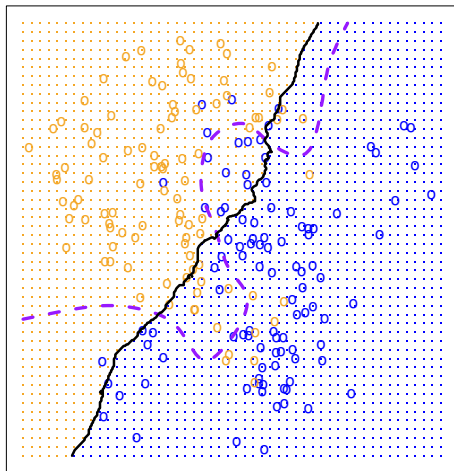
"Why would we ever choose to use a **more restrictive method** instead of a **very flexible approach**?"

Classification - estimation of f ?

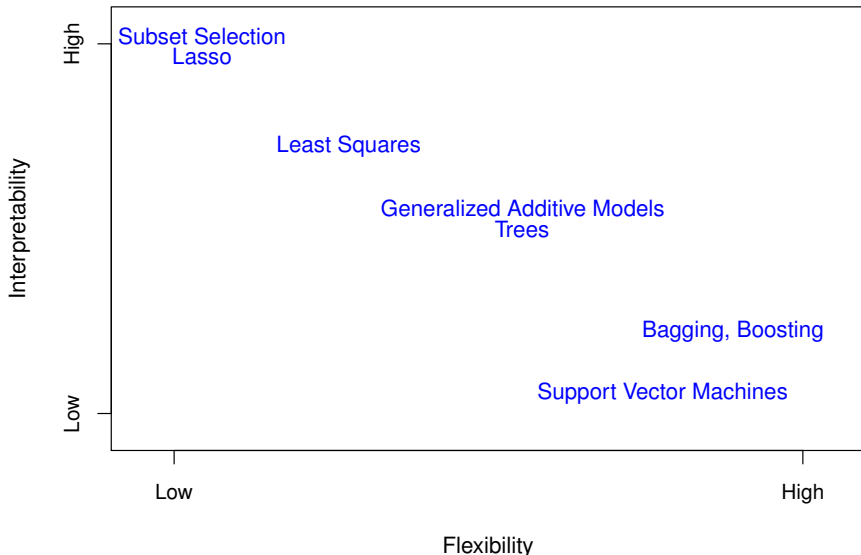
KNN: $K=1$



KNN: $K=100$



Prediction Accuracy vs Model Interpretability



Outline

- 1 Regression problems**
- 2 Classification problems

Assessing model accuracy

Suppose we have a regression model $y = f(x) + \varepsilon$.

Estimate \hat{f} from some **training data**, $Tr = \{x_i, y_i\}_1^n$.

One common measure of accuracy is:

Training Mean Squared Error

$$\text{MSE}_{Tr} = \text{Ave}_{i \in Tr} [y_i - \hat{f}(x_i)]^2 = \frac{1}{n} \sum_{i=1}^n [(y_i - \hat{f}(x_i))]^2$$

Assessing model accuracy

Suppose we have a regression model $y = f(x) + \varepsilon$.

Estimate \hat{f} from some **training data**, $Tr = \{x_i, y_i\}_1^n$.

One common measure of accuracy is:

Training Mean Squared Error

$$\text{MSE}_{Tr} = \text{Ave}_{i \in Tr} [y_i - \hat{f}(x_i)]^2 = \frac{1}{n} \sum_{i=1}^n [(y_i - \hat{f}(x_i))]^2$$

Measure **real accuracy** using **test data** $Te = \{x_j, y_j\}_1^m$

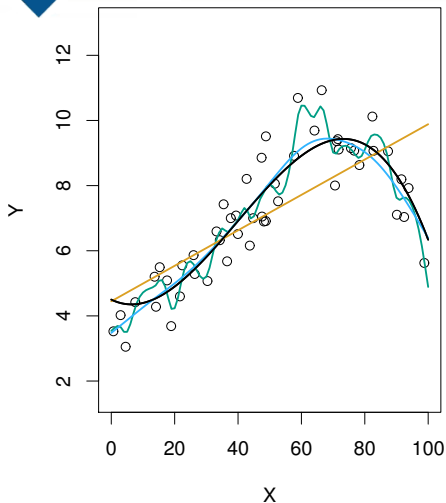
Test Mean Squared Error

$$\text{MSE}_{Te} = \text{Ave}_{j \in Te} [y_j - \hat{f}(x_j)]^2 = \frac{1}{m} \sum_{j=1}^m [(y_j - \hat{f}(x_j))]^2$$

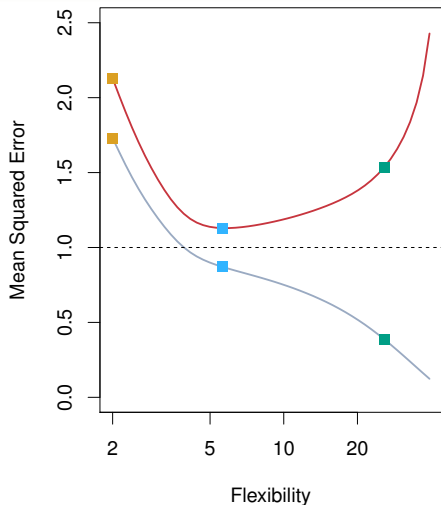
Training vs Test MSEs

- In general, the more **flexible** a method is, the lower its **training MSE** will be. i.e. it will “fit” the training data very well.
- However, the **test MSE** may be higher for a more **flexible** method than for a **simple** approach like linear regression.
- Flexibility also makes interpretation more difficult. There is a trade-off between **flexibility** and **model interpretability**.

Example: splines

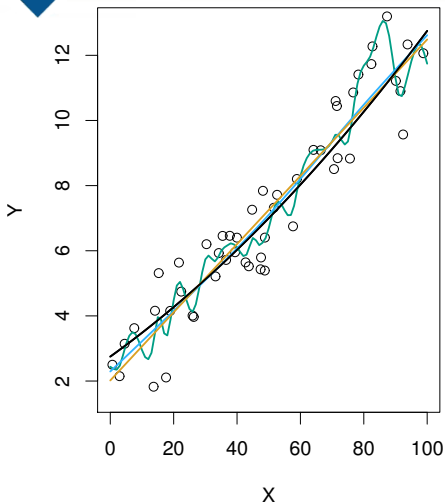


Black: true curve
Orange: linear regression
Blue/green: Smoothing splines

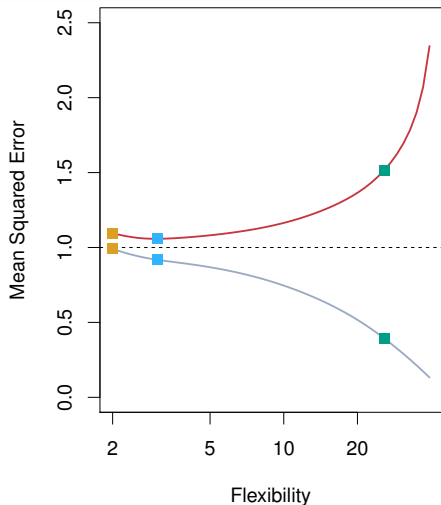


Grey: Training MSE
Red: Test MSE
Dashed: Minimum test MSE

Example: splines

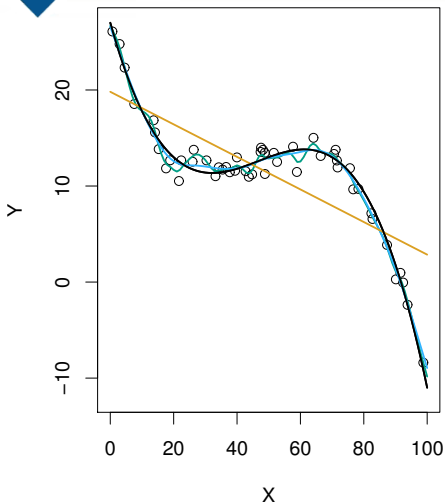


Black: true curve
Orange: linear regression
Blue/green: Smoothing splines

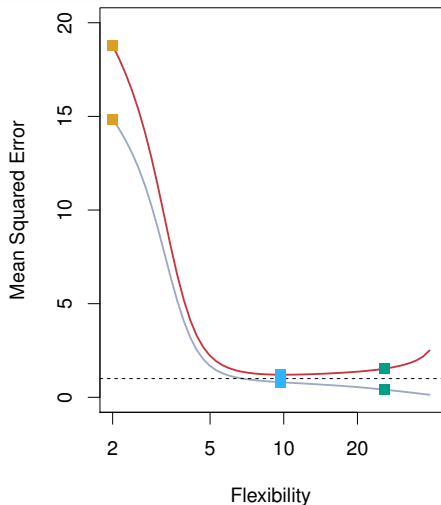


Grey: Training MSE
Red: Test MSE
Dashed: Minimum test MSE

Example: splines



Black: true curve
Orange: linear regression
Blue/green: Smoothing splines



Grey: Training MSE
Red: Test MSE
Dashed: Minimum test MSE

Bias-variance tradeoff

There are two competing forces that govern the choice of learning method: **bias** and **variance**.

Bias

is the error that is introduced by modeling a complicated problem by a simpler problem.

- For example, linear regression assumes a linear relationship when few real relationships are exactly linear.
- In general, the **more flexible** a method is, the **less bias** it will have.

Bias-variance tradeoff

There are two competing forces that govern the choice of learning method: **bias** and **variance**.

Variance

refers to how much your estimate would change if you had different training data.

- In general, the **more flexible** a method is, the **more variance** it has.
- The **size** of the training data has an impact on the variance

The bias-variance tradeoff

MSE decomposition

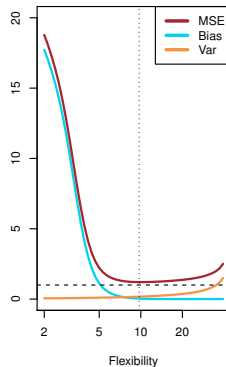
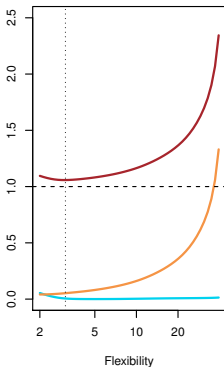
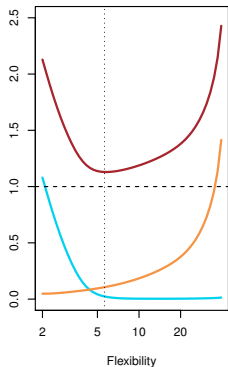
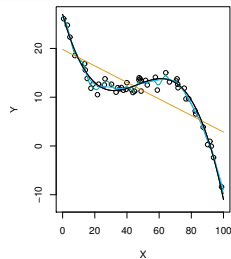
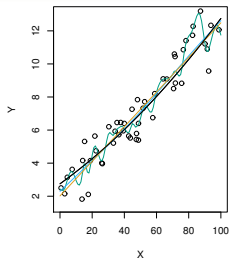
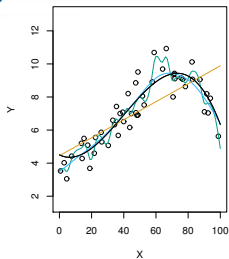
If $Y = f(x) + \varepsilon$ and $f(x) = E[Y | X = x]$, then the expected **test** MSE for a new Y at x_0 will be equal to

$$E[(Y - \hat{f}(x_0))^2] = [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\hat{f}(x_0)) + \text{Var}(\varepsilon)$$

Test MSE = Bias² + Variance + Irreducible variance

- The expectation averages over the variability of Y as well as the variability in the training data.
- As the flexibility of \hat{f} increases, its variance increases and its bias decreases.
- Choosing the flexibility based on average test MSE amounts to a **bias-variance trade-off**.

Bias-variance trade-off



Optimal prediction

MSE decomposition

If $Y = f(x) + \varepsilon$ and $f(x) = E[Y | X = x]$, then the expected **test** MSE for a new Y at x_0 will be equal to

$$E[(Y - \hat{f}(x_0))^2] = [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\hat{f}(x_0)) + \text{Var}(\varepsilon)$$

→ **see proof of MSE decomposition**

The optimal MSE is obtained when

$$\hat{f} = f = E[Y | X = x].$$

Then bias=variance=0 and

$$\text{MSE} = \text{irreducible variance}$$

This is called the “**oracle**” predictor because it is not achievable in practice.

Outline

1 Regression problems

2 Classification problems

Classification problems

Here the response variable Y is **qualitative**.

- e.g., email is one of $\mathcal{C} = (\text{spam}, \text{ham})$
- e.g., voters are one of $\mathcal{C} = (\text{Liberal}, \text{Labor}, \text{Green}, \text{National}, \text{Other})$

Our goals are:

- 1 Build a classifier $C(x)$ that assigns a class label from \mathcal{C} to a future unlabeled observation x .
- 2 Assess the uncertainty in each classification (i.e., the probability of misclassification).
- 3 Understand the roles of the different predictors among $X = (X_1, X_2, \dots, X_p)$.

Classification problems

Here the response variable Y is **qualitative**.

- e.g., email is one of $\mathcal{C} = (\text{spam}, \text{ham})$
- e.g., voters are one of $\mathcal{C} = (\text{Liberal}, \text{Labor}, \text{Green}, \text{National}, \text{Other})$

Our goals are:

- 1 Build a classifier $C(x)$ that assigns a class label from \mathcal{C} to a future unlabeled observation x .
- 2 Assess the uncertainty in each classification (i.e., the probability of misclassification).
- 3 Understand the roles of the different predictors among $X = (X_1, X_2, \dots, X_p)$.

Classification problem

In place of MSE, we now use:

Error rate

$$\text{Error rate} = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{f}(x_i))$$

where $\hat{f}(x_i)$ is the predicted class label and $I(y_i \neq \hat{f}(x_i))$ is an indicator function.

- That is, the error rate is the fraction of misclassifications.
- The training error rate is misleading (too small).
- We want to minimize the test error rate:
 $E(I(y_0 \neq \hat{f}(x_0)))$

Classification problem

In place of MSE, we now use:

Error rate

$$\text{Error rate} = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{f}(x_i))$$

where $\hat{f}(x_i)$ is the predicted class label and $I(y_i \neq \hat{f}(x_i))$ is an indicator function.

- That is, the error rate is the fraction of misclassifications.
- The training error rate is misleading (too small).
- We want to minimize the test error rate:
 $E(I(y_0 \neq \hat{f}(x_0)))$

Optimal classifier

- A classification rule assigns each value of \mathbf{x} to one of the available classes $\mathcal{C}_1, \dots, \mathcal{C}_K$
- Such a rule will divide the input space into regions \mathcal{R}_k called decision regions, one for each class, such that all points in \mathcal{R}_k are assigned to class \mathcal{C}_k
- In order to find the optimal decision rule, consider first of all the case of two classes. A misclassification occurs when an input vector belonging to class \mathcal{C}_1 is assigned to class \mathcal{C}_2 or vice versa:

$$\begin{aligned}\Pr(\text{misclassification}) &= \Pr(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + \Pr(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} \Pr(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} \Pr(\mathbf{x}, \mathcal{C}_1) d\mathbf{x}\end{aligned}$$

To which class should we assign each point \mathbf{x} ?

Optimal classifier

- To minimize $\Pr(\text{misclassification})$, we should arrange that each \mathbf{x} is assigned to whichever class has the smaller value of the integrand
- For a given value of \mathbf{x} , if $\Pr(\mathbf{x}, \mathcal{C}_1) > \Pr(\mathbf{x}, \mathcal{C}_2)$, we should assign \mathbf{x} to class \mathcal{C}_1
- Since $\Pr(\mathbf{x}, \mathcal{C}_k) = \Pr(\mathcal{C}_k|\mathbf{x})\Pr(\mathbf{x})$, and $\Pr(\mathbf{x})$ is common to both terms, the minimum probability of mistake is obtained if each value of \mathbf{x} is assigned to the class for which the posterior probability $\Pr(\mathcal{C}_k|\mathbf{x})$ is largest
- For the more general case of K classes, it is slightly easier to maximize the probability of being correct, and use the same arguments as above

Optimal classifier

Suppose the K elements in \mathcal{C} are numbered $1, 2, \dots, K$. Let

$$p_k(x) = \Pr(Y = k \mid X = x), \quad k = 1, 2, \dots, K.$$

These are the **conditional class probabilities** at x .
Then the **Bayes** classifier at x is

$$C(x) = j \quad \text{if } p_j(x) = \max\{p_1(x), p_2(x), \dots, p_K(x)\}$$

- This gives the minimum average test error rate.
- It is an “oracle predictor” because we do not usually know $p_k(x)$.

Optimal classifier

Suppose the K elements in \mathcal{C} are numbered $1, 2, \dots, K$. Let

$$p_k(x) = \Pr(Y = k \mid X = x), \quad k = 1, 2, \dots, K.$$

These are the **conditional class probabilities** at x .
Then the **Bayes** classifier at x is

$$C(x) = j \quad \text{if } p_j(x) = \max\{p_1(x), p_2(x), \dots, p_K(x)\}$$

- This gives the minimum average test error rate.
- It is an “oracle predictor” because we do not usually know $p_k(x)$.

Optimal classifier

Suppose the K elements in \mathcal{C} are numbered $1, 2, \dots, K$. Let

$$p_k(x) = \Pr(Y = k \mid X = x), \quad k = 1, 2, \dots, K.$$

These are the **conditional class probabilities** at x .
Then the **Bayes** classifier at x is

$$C(x) = j \quad \text{if } p_j(x) = \max\{p_1(x), p_2(x), \dots, p_K(x)\}$$

- This gives the minimum average test error rate.
- It is an “oracle predictor” because we do not usually know $p_k(x)$.

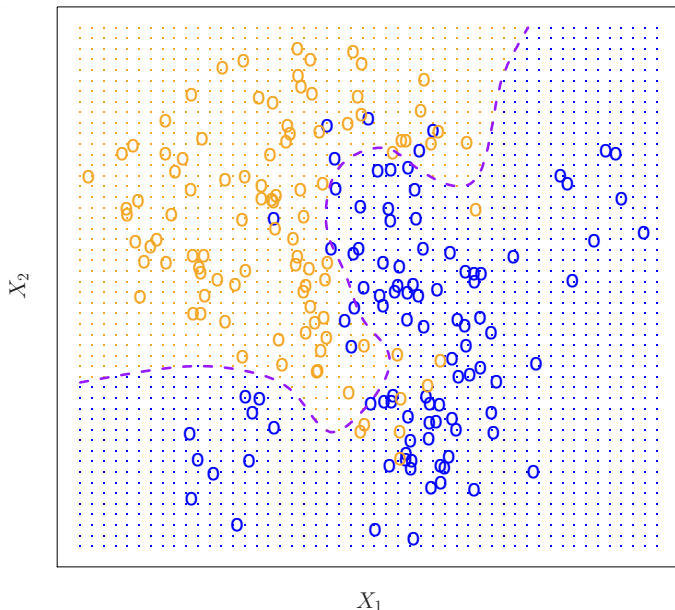
Bayes error rate

Bayes error rate

$$1 - E(\max_j \Pr(Y = j|X))$$

- The “Bayes error rate” is the lowest possible error rate that could be achieved if we knew exactly the “true” probability distribution of the data.
- It is analagous to the “irreducible error” in regression.
- On test data, no classifier can get lower error rates than the Bayes error rate.
- In reality, the Bayes error rate is not known exactly.

Bayes optimal classifier



k-Nearest Neighbours

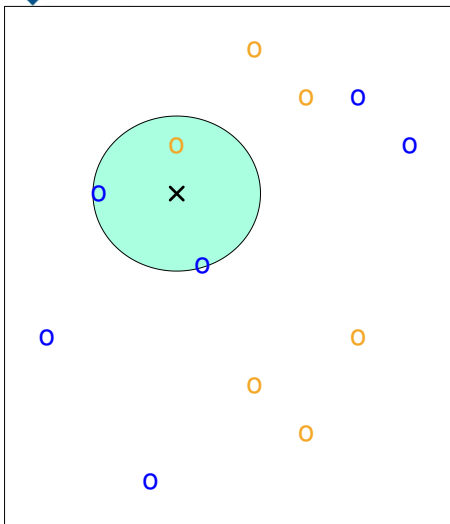
One of the simplest classifiers. Given a test observation x_0 :

- Find the K nearest points to x_0 in the training data: \mathcal{N}_0 .
- Estimate conditional probabilities

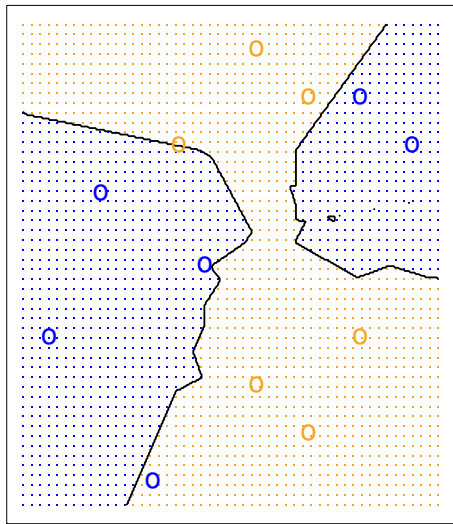
$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$

- Apply Bayes rule and classify x_0 to class with largest probability.

kNN Classifier

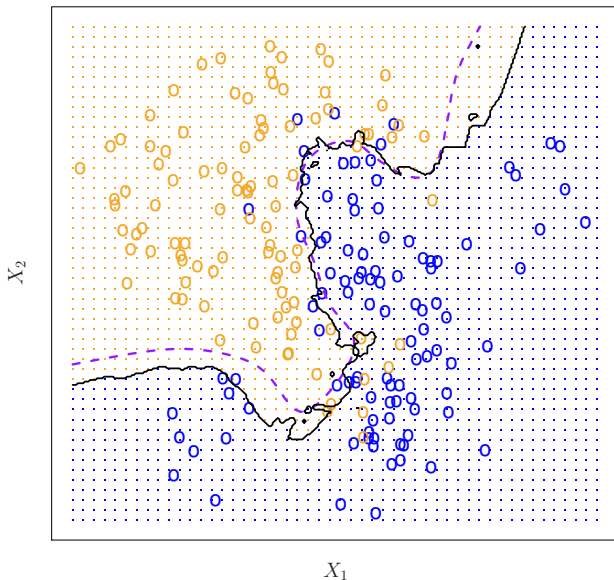


$K = 3.$



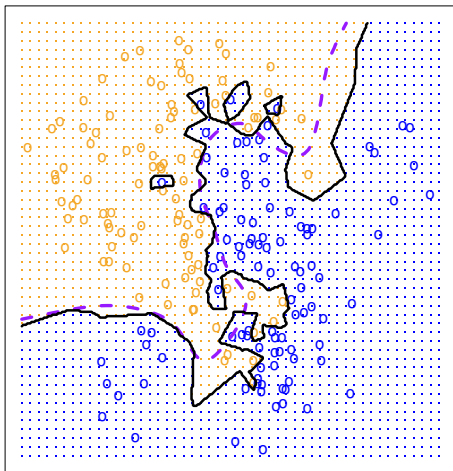
kNN Classifier

KNN: K=10

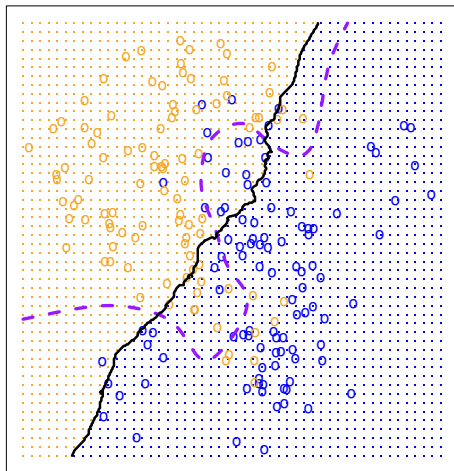


kNN Classifier

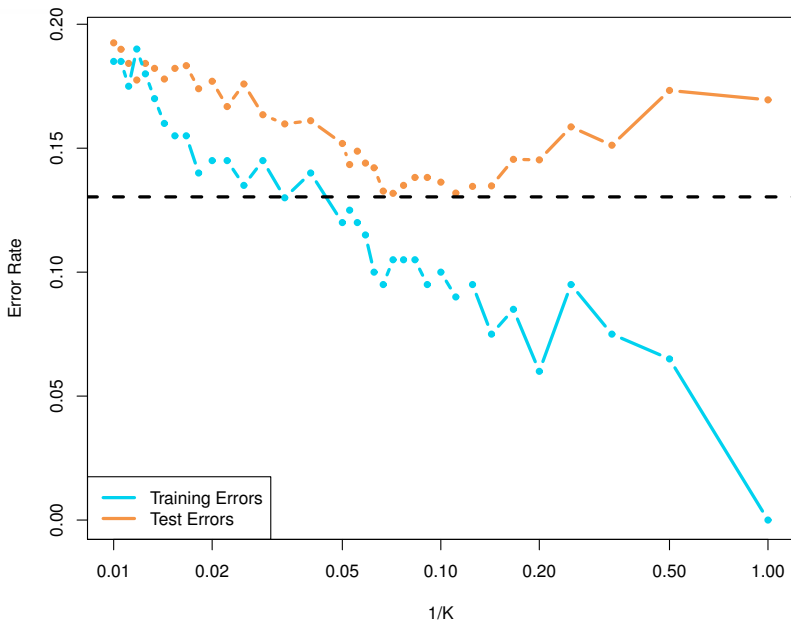
KNN: $K=1$



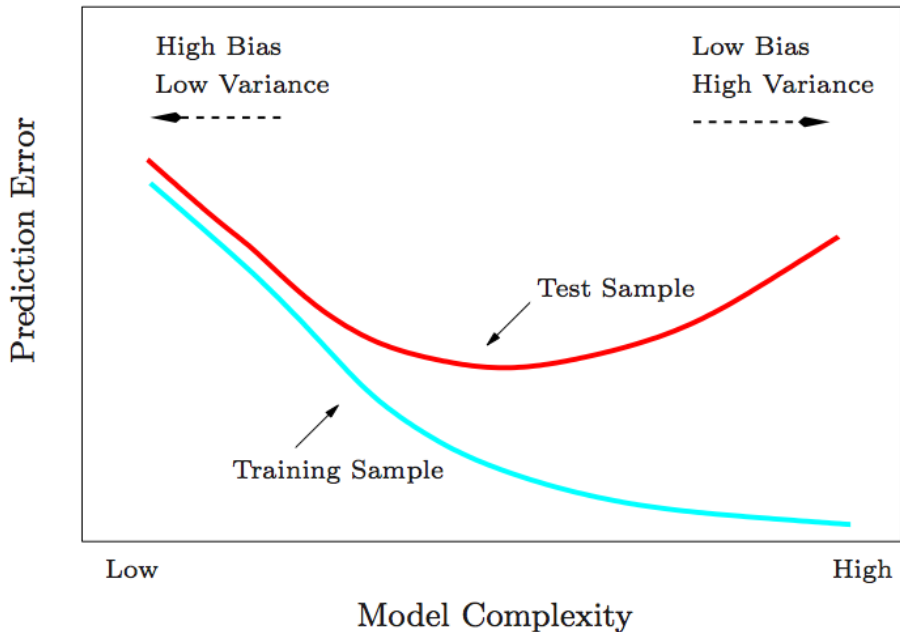
KNN: $K=100$



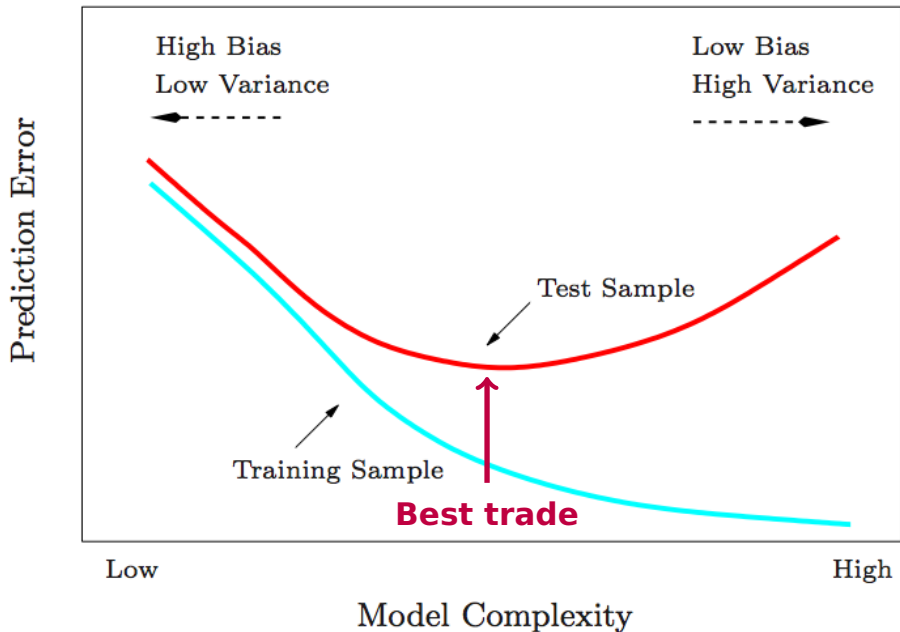
kNN Classifier



A fundamental picture



A fundamental picture



A fundamental picture

