Statistical Learning and Prediction

Marjolein Fokkema

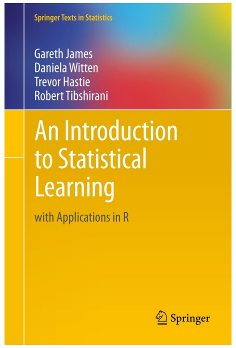
Methodology and Statistics Unit
Leiden University

Room 3B20
m.fokkema@fsw.leidenuniv.nl

This Course

- New methodology for data analysis.
- Focus on *prediction*, instead of *explanation*.
- Latter is/was often the focus in the field of Statistics (accurately modeling distributions).
- Former is/was often the focus of machine learning / computer science (creating fast algorithms that predict well).
- Statistics and Machine Learning: Statistical Learning.

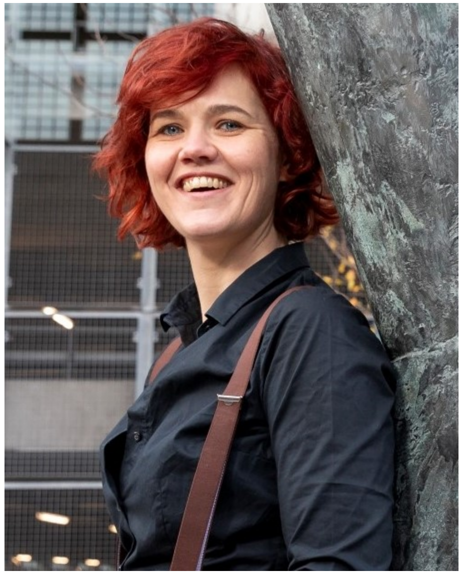
Course ingredients

- Book: 

- Online lectures

- Preparations before each class:
  - Watch lectures
  - Read book chapter(s)
  - Make exercises (not graded)
  - Class-specific preparations: See Brightspace > "General information" > "Preparations lectures"

Two professors



Dr. Tom Wilderjans Dr. Marjolein Fokkema

Topics (two sessions per topic)

1. **Introduction** (M.F.)
2. **Sampling; Logistic regression** (T.W.)
3. **Classification; Subset selection and regularization** (T.W.)
4. **Unsupervised learning** (T.W.)
5. **Splines; Support vector machines** (M.F.)
6. **Support vector machines; Decision trees** (M.F.)
7. **Decision tree ensembles** (M.F.)

Timetable

Sess. 1	Sess. 2	Topic	Instructor
14-11	17-11	Introduction	M.F.
21-11	24-11	Resampling	T.W.
28-11	01-12	Lasso	T.W.
05-12	08-12	Unsupervised	T.W.
12-12	15-12	Non-linearity	M.F.
19-12	22-12	Trees	M.F.
05-01	09-01	Ensembles	M.F.
	16/01	Q&A	both
	26/01	Presentations	both

Evaluation: Assignments

Final grade based on (each weighted 1/3):

Assignment	**Distributed**	**Due**
1. Written	08-12	22-12 (17h)
2. Written	22-12	16-01 (17h)
3. Presentation	22-12	26-01 (13h)

Assignment 1: Individual, written, structured assignment. Covers topics 1-4.

Assignment 2: Individual, written, structured assignment. Covers (mostly) topics 5-7.

Assignment 3: Group assignment (group of 2 or 3 students). Less structured, analysis of data set(s) of students' own choice.

Week 1

For today, I assume you watched the following online lectures:

- Introduction
  - Supervised and Unsupervised Learning (12:12)
- Statistical Learning
  - Statistical Learning and Regression (11:41)
  - Curse of Dimensionality and Parametric Models (11:40)
  - Assessing Model Accuracy and Bias-Variance Trade-off (10:04)
  - Classification Problems and K-Nearest Neighbors (15:37)

This week's / topic's aims

Becoming acquainted with:

- Explanation versus prediction
- Method of k-nearest neighbours
- Bias-variance trade-off
- Benefits of shrinkage (bias)
- Curse of dimensionality

Statistical Learning

- Statistical learning refers to a vast set of tools for understanding data.

  - Supervised: $Y \leftarrow f(X_1, \ldots, X_p)$; predict Y on the basis of X

  - Unsupervised: $X_1, \ldots, X_p$; finding structure (underlying dimensions/groups)

Statistical Learning

Supervised learning models: $\hat{Y} = f(X_1, \ldots, X_p) = \mathbb{E}(Y|X)$

Can be used for:

- Explanation: understanding how the X's are related to Y; possibly causally.

- Prediction: if we have new observations with known values of X's, what is the expected (predicted) value of Y and how accurate are these predictions?

Explanatory Regression

- Suppose we have data and obtain estimates:

$$\hat{y}_i = 2 + 0.5x_{i1} + 1.5x_{i2}$$

- Estimated coefficients indicate magnitude of the effects
- Standard errors indicate variability of estimated effects
- Statistical tests used to see whether explanatory variables really affect the response
- Adequate parameter estimation is crucial: Accurate estimates = unbiased parameter estimates! That is:

$$\mathbb{E}[\hat{\beta} - \beta] = 0$$

Predictive Regression

- Suppose we have data and obtain estimates:

$$\hat{y}_i = 2 + 0.5x_{i1} + 1.5x_{i2}$$

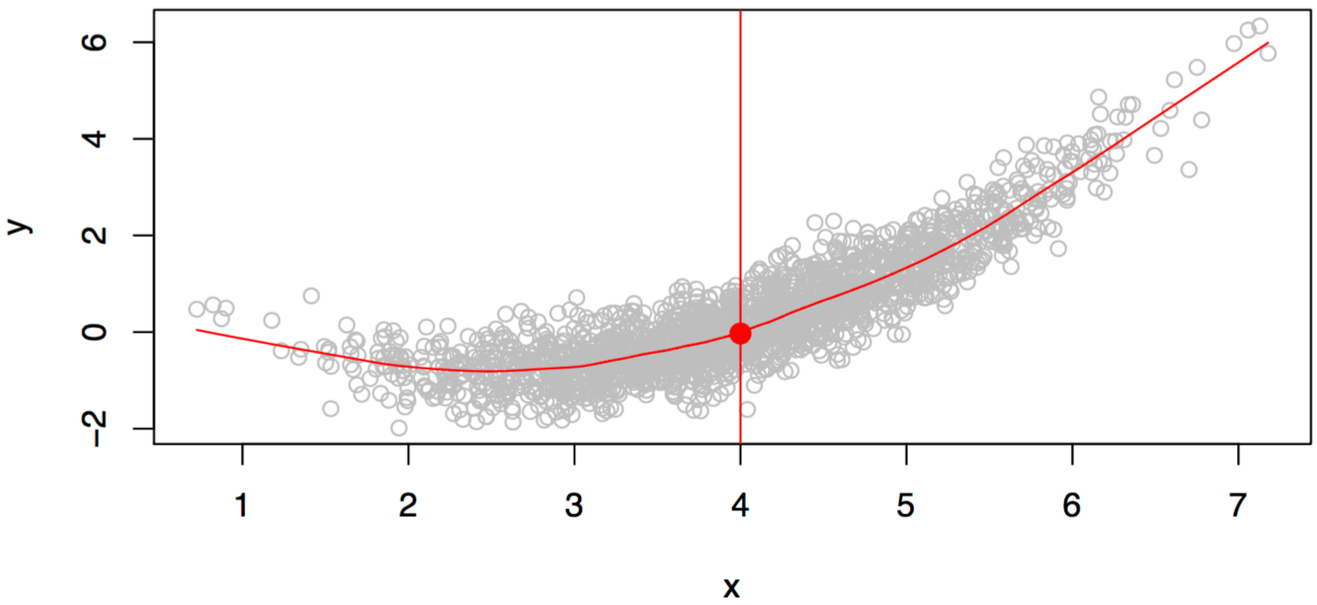
- Suppose we have a new observation $x_i = \begin{bmatrix} 2 & 3 \end{bmatrix}$
- With these values we can predict Y: $\hat{y}_i = 2 + 0.5 \times 2 + 1.5 \times 3 = 7.5$
- Prediction focuses on accuracy of $\hat{Y}$. Do not care for recovering parameters that generated the data. Crucial to obtain a model that yields as accurate as possible $\hat{Y} = \hat{f}(X)$. That is, minimize:

$$\mathbb{E}\left[(\hat{Y} - Y)^2\right]$$

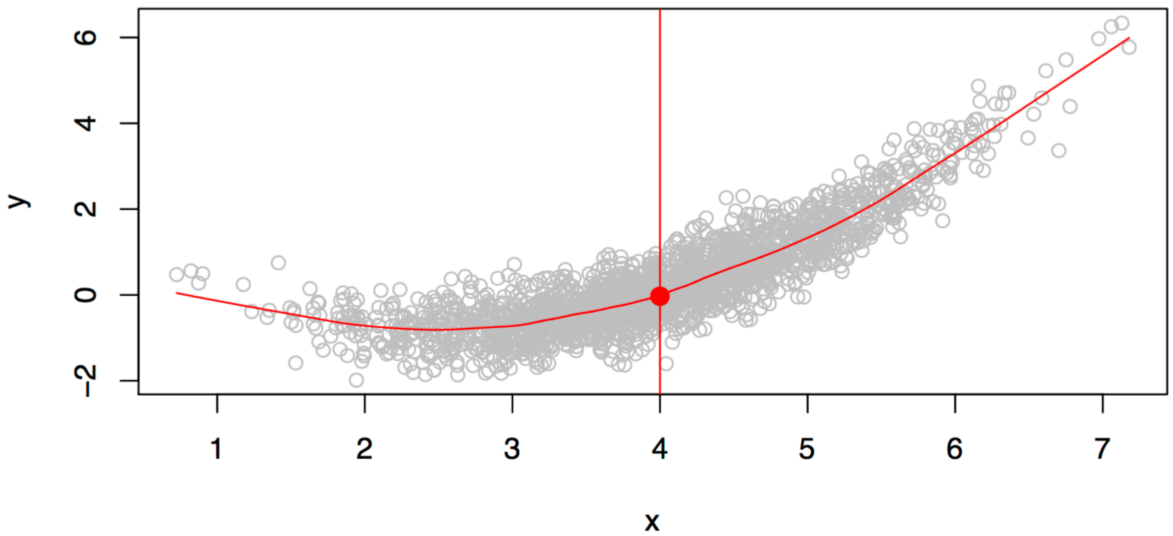
Population

Consider that we have a population P^*, within which the conditional means of the response variable ($Y \in \mathbb{R}$) are given by some function of the predictors ($X \in \mathbb{R}^p$), that is:

$$Y = f(X) + \epsilon.$$



Population



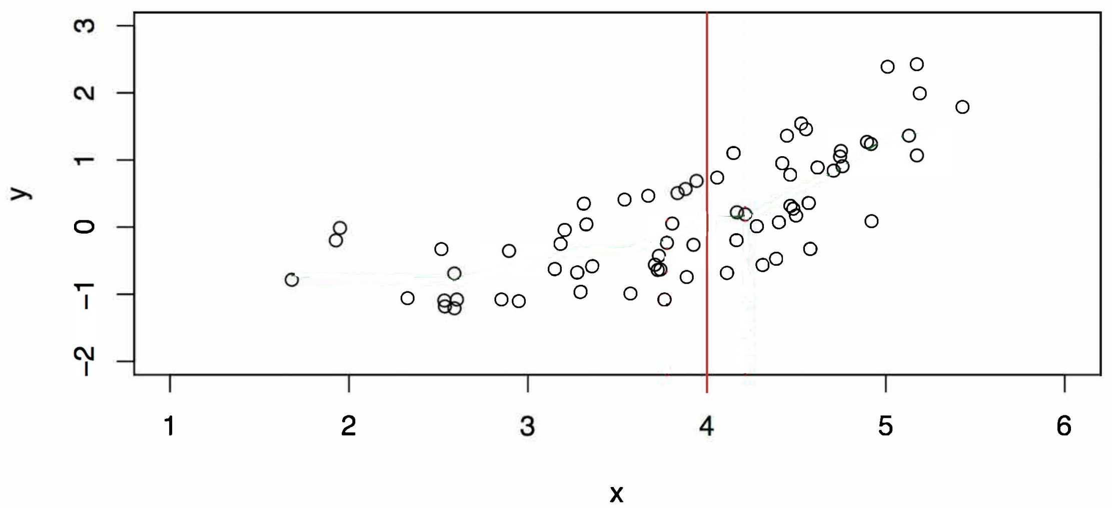
The population regression line gives the conditional means at each point x: $\mathbb{E}(Y = x)$.

If we repeatedly observe all possible values of X, we could construct a perfect $\hat{f}(X)$.

Sample

In practice, we only have sample data comprising n observations: $D = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, and use it to <u>train</u> a model $\hat{f}$:

$$y_i = \hat{f}(x_i) + \epsilon_i$$



We cannot estimate a true conditional mean at all points ($X = x$).

What can we do to obtain a (not perfect but good) $\hat{f}(X)$?

Bias-Variance decomposition

If we train a model on a sample D to obtain an $\hat{f}(x)$, we want to minimize the *expected prediction error*: the error we would make on a new / future / yet unseen observation $(\mathbf{x}_0, y_0)$:

$$\mathrm{EPE}(x_0) = \mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] =$$

$$\sigma^2 + \left[\mathrm{Bias}(\hat{f}(x_0))\right]^2 + \mathrm{Var}\left(\hat{f}(x_0)\right)$$

Bias-variance decomposition: Closer look

- We draw training sample D of size n from a probability distribution P^*.

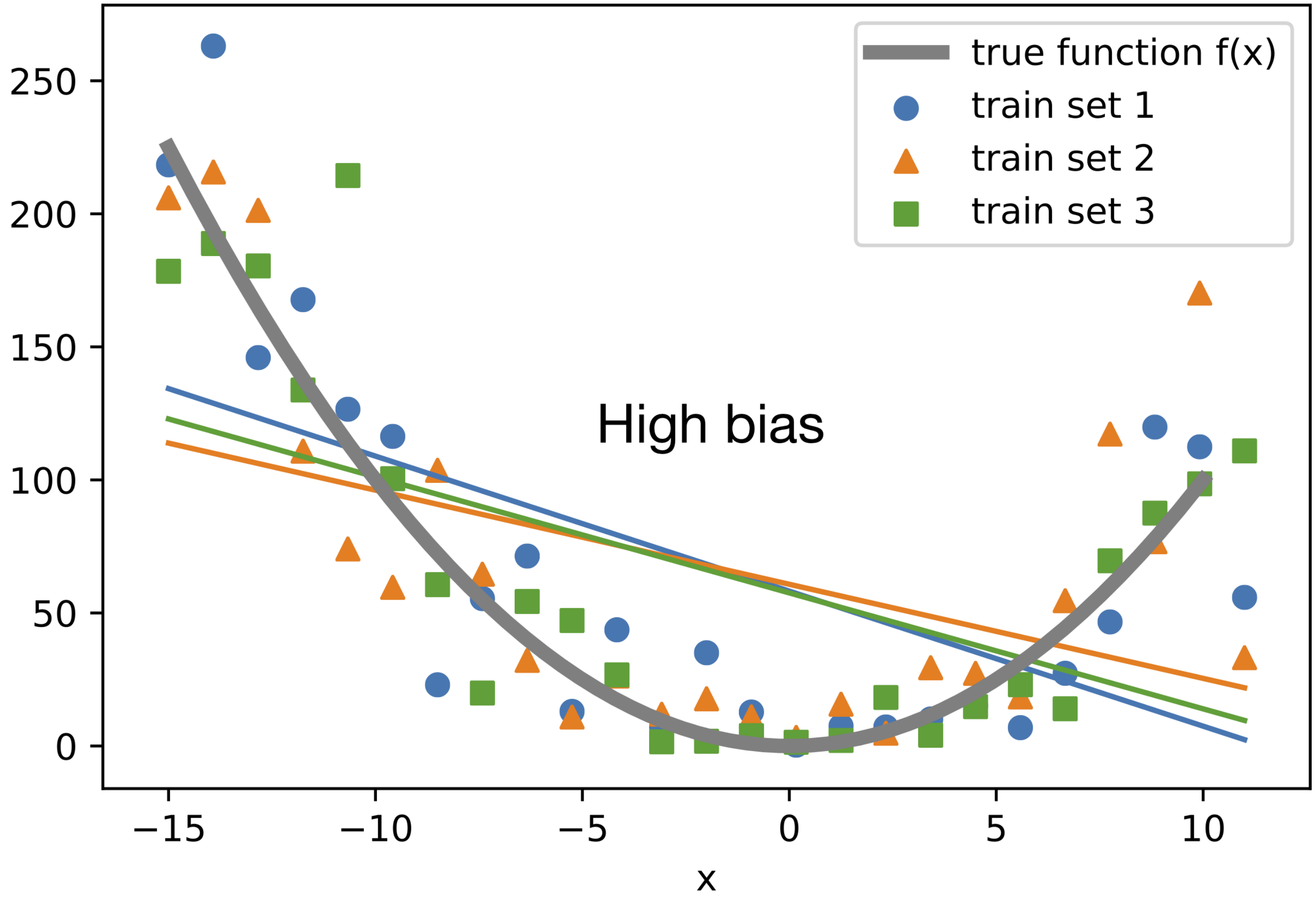
  - Let f be the *true* conditional mean function $\mathbb{E}(Y|X)$ (unknown, depends only on P^*)
  - We apply a fitting method to D to obtain $\hat{f}(X)$.
  - Let $\bar{f}(X) = \mathbb{E}_D[\hat{f}(X)]$

- Aim is to minimize *expected prediction error* over observations in- and excluded in D:

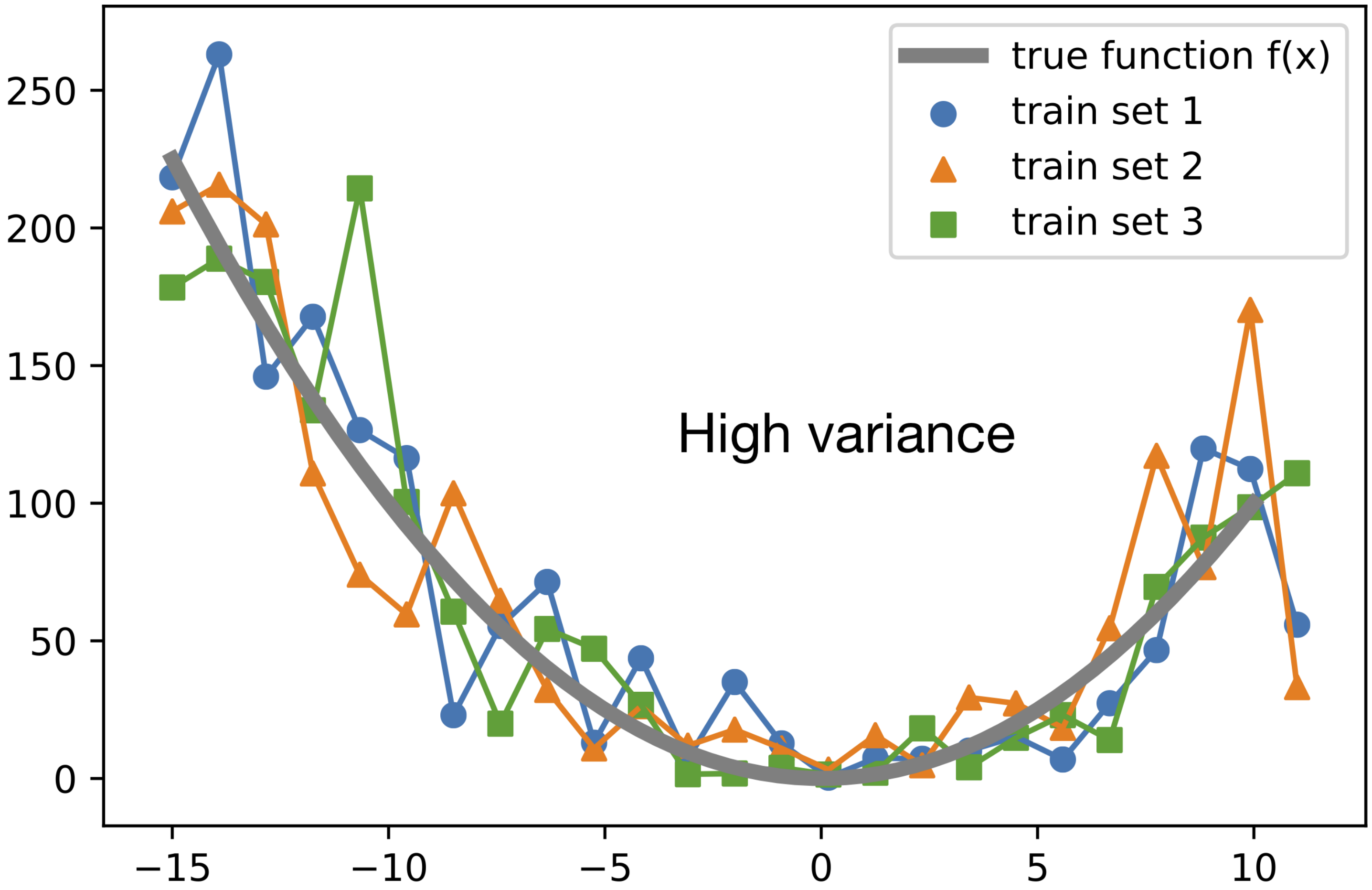
$$
\begin{aligned}
\mathrm{EPE}(\hat{f}) = & \; \mathbb{E}_X[Var(Y|X)] + \\
& \; \mathbb{E}_X[(f(X) - \bar{f}(X))^2] + \\
& \; \mathbb{E}_D \mathbb{E}_{X,Y}[(\hat{f}(X) - \bar{f}(X))^2]
\end{aligned}
$$

- First term is *irreducible*, second and third terms are *reducible* (squared bias and variance).

Bias-variance decomposition



Bias-variance decomposition



Predictive Regression: Bias-variance trade-off

$$EPE(x_0) = \sigma^2 + \left[\text{Bias}(\hat{f}(x_0))\right]^2 + \text{Var}\left(\hat{f}(x_0)\right)$$

- Can we directly compute these quantities when we fit a model to a sample of data?

- Which prediction method would have lowest *squared bias* on *any* data problem?

- Which statistical method which would have lowest *variance* on *any* data problem?

- How is the variance quantified in OLS regression?

Predictive Regression: Bias-variance trade-off

- Traditional statistical textbooks focus on obtaining unbiased estimates (e.g., OLS, ML):

$$\mathbb{E}[\hat{\beta}] = \beta$$

- (Modern) statistical learning accepts biased parameter estimates as long as the variance decreases more than the squared bias increases.

- ″From a Bayesian perspective, the principle of unbiasedness is reasonable in the limit of large samples, but otherwise it is potentially misleading″ (Gelman et al., 1995)

Exercise 1: Shrinkage

- Generate $n_{train} = 50$ observations X from a uniform distribution (range -3 to 3): `x <- runif(50, -3, 3)`

- Generate response $Y = .1X + \epsilon$, with $\epsilon \sim N(0, 1)$: `eps <- rnorm(50)`

- Generate $n_{test} = 1,000$ test observations from the same distribution.

- Compute $\hat{\beta}_{OLS}$ using the training observations. Use function `lm`; specify `y~0+x` as the model formula to exclude the intercept.

  a) What do we already know about the values of irreducible error, bias and variance of the OLS estimated linear model $\hat{f}$?

  Now generate predictions for the OLS and shrunken versions of the OLS coefficient:

- Generate a vector of shrinkage values $s \in \{0, 0.1, \ldots, 0.9, 1.0\}$.

- Generate predictions for the test set $\hat{y}_i = x_i \cdot s \cdot \hat{\beta}$ (do not use function `predict`, but extract the $\hat{\beta}_{OLS}$ using function `coef`, then manually perform computation to generate predicted values for the test observations, for each value of s).

- For each value of s, compute the mean of the squared prediction errors on the test observations:

$$MSE_{\text{test}} = \frac{1}{n_{test}} \sum (y_i - \hat{y}_i)^2$$

  b) Plot the test MSE values (y-) against shrinkage values (x-axis).

  c) Describe the effecft of shrinkage, and when it is most effective. What is the effect of shrinkage on the irreducible error, bias and variance?

- Repeat the above procedure (except plotting) 100 times (e.g., use a `for` loop)

  d) Create a boxplot with the test MSE values on the y axis and shrinkage values on the x-axis.

  e) Describe the effect of shrinkage, and when it is most effective.

f) What do you think would happen to the shape of the curve if the training sample size were doubled?

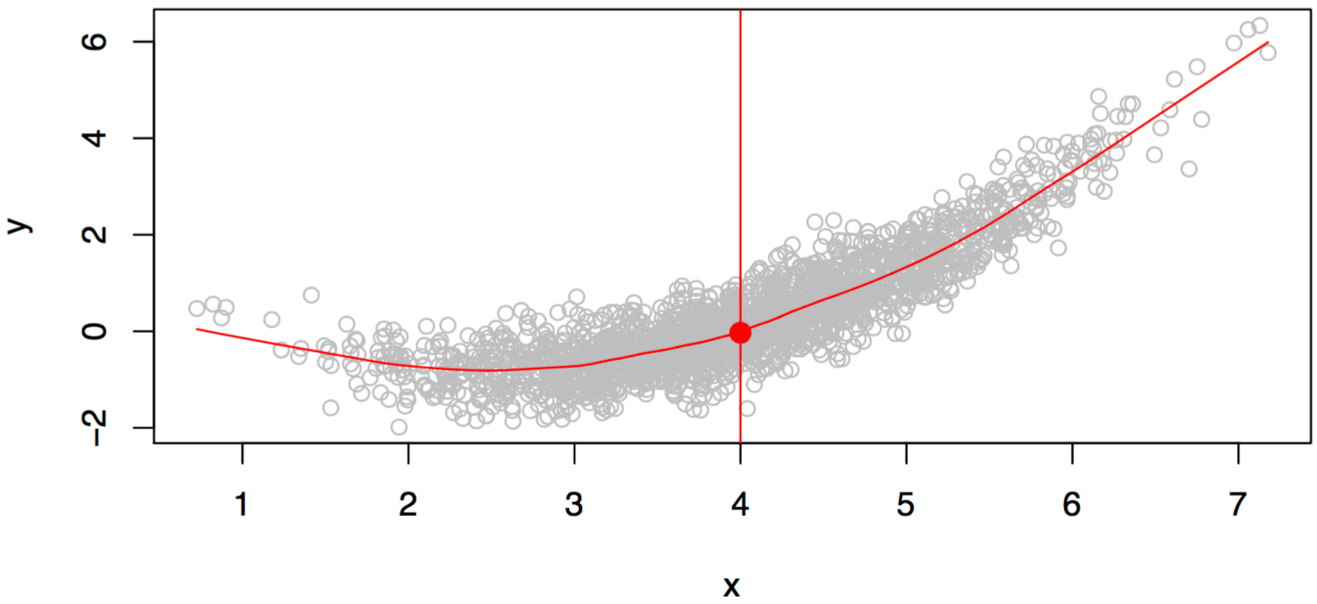
g) What do you think would happen to the shape of the curve if the effect of X were twice as strong (i.e., $Y = .2X + \epsilon$)?

Non-linear Regression

Often we fit a linear regression, assuming that the conditional means in the population lie on a straight line.

This assumption is most likely false!

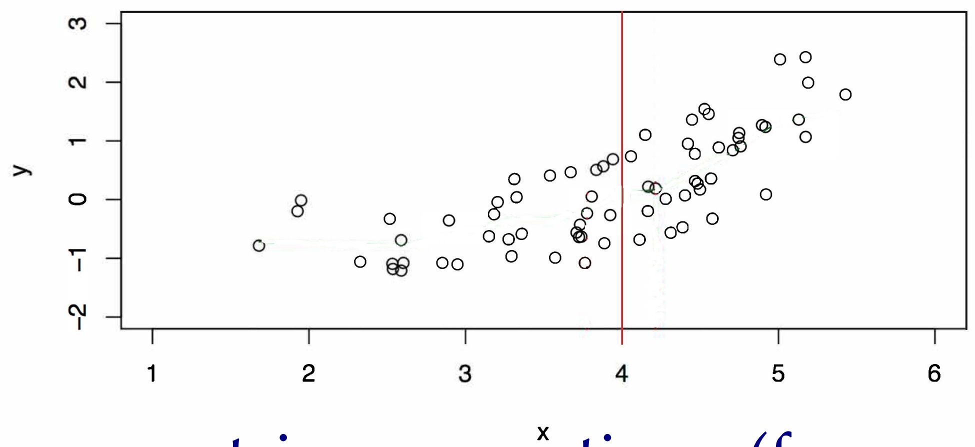
Population: Non-linear Regression



The regression line in the population (i.e., the *true* association between X and Y) combines the conditional means at each point x

Sample data: Linear Regression

We obtain sample data:



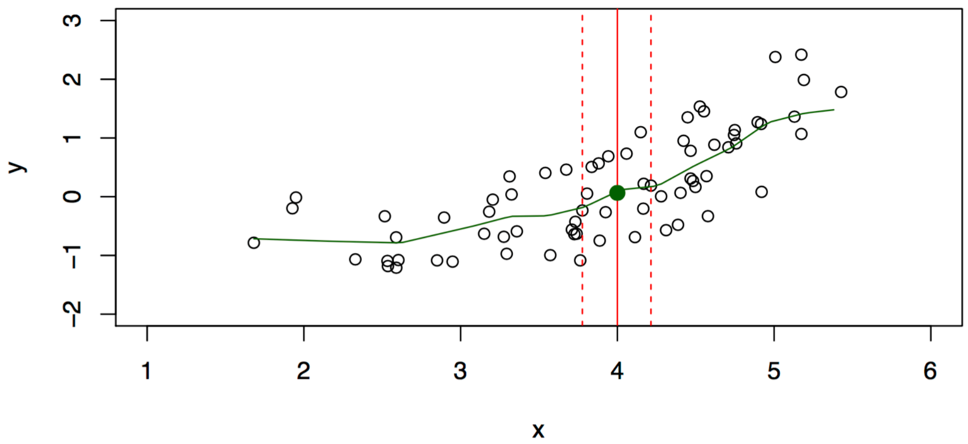
We can make parametric assumptions (for example: linear) and obtain an $\hat{f}(X)$.

- What kind of bias does this introduce? How does it affect the variance? The irreducible error?

- What if we fit a kth order polynomial? What happens to the bias if k increases? To the variance? To the irreducible error?

Sample data: Non-linear Regression

Using sample data, we obtain $\hat{f}(X)$ using kNN:



- What happens to the size of the neighbourhood if we increase k?
- What happens to the bias?
- What happens to the variance?
- What happens to the irreducible error?

Multiple Predictor Variables

- With multiple predictors the observations are further spread out through the space.
- Nearest neighbours might not be near at every point.
- Then flexible models become very wild.
- This is known as the *curse of dimensionality*.
- More structure in f is needed.
- What increases if p increases? Bias, variance or irreducible error?
- How should we then keep EPE (reasonably) low?

Exercise 2: Curse of Dimensionality

- Generate a dataset with $p = 10,000$ predictor variables and sample size $n = 100$; $X \sim N(\mathbf{0}, \mathbf{I})$. $\mathbf{I}$ is an $p \times p$ identity matrix; thus, all predictors follow a standard normal distribution and are uncorrelated, so you could generate the data as follows: `matrix(rnorm(10000*100), nrow = 100)`.

- Compute Euclidian distances between all points in the dataset, once for each of $p \in \{1, 2, 10, 100, 1000, 10000\}$ dimensions. So, for $p = 1$ you compute distances only in the first dimension (i.e., 1st column of the data matrix), for $p = 2$ you compute distances in the first two dimensions (i.e., first 2 columns of the data matrix), etc. item Hint: use functions `dist` and `hist`.

- Create a histogram of the Euclidian distances between observations for each $p \in \{1, 2, 10, 100, 1000, 10000\}$. It is helpful to keep the limits of the x-axis the same in each histogram (specify argument `xlim`), including 0 and the maximum distance.

- Based on the histograms, do you think the nearest neighbours are near in 1-dimensional space? In 2-dimensional space? In 10-, 100-, 1000-, 10000-dimensional space?

Evaluating predictive accuracy

For continuous outcomes, we can use the following error measures:

$$MSE_{\text{test}} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2$$

$$MAE_{\text{test}} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |y_i - \hat{y}_i|$$

Classification

Response variable Y may be a categorical variable with categories $\mathcal{C} = 1, \ldots, k, \ldots, K$.

Again, we want to predict response Y based on predictors X:

- Can directly construct a classifier $\hat{f}(X) = C(X)$ that assigns a predicted category from C based on X.

- Can construct a function $\hat{f}(X)$ that provides conditional probabilities: $\hat{p}_k(X) = Pr(Y = k|X = x)$.

  Then Bayes classifier assigns $C(X) = k$ if $\hat{p}_k(x) = \max\{\hat{p}_1(x), \ldots, \hat{p}_K(x)\}$

Evaluating predictive accuracy: Binary outcomes

- Misclassification rate:

$$\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} I(y_i \neq \hat{y}_i)$$

- Squared error loss on predicted probabilities (a.k.a. Brier score) :

$$\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2$$

- Cross-entropy:

$$-\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

Evaluating predictive accuracy: Binary outcomes

- Brier score and cross-entropy quantify accuracy of predicted *probabilities*.

- MCR quantifies accuracy of predicted *class labels* only.

- Statistical modeling (modeling probabilities) and real-world decision making (deciding yes/no) should be separated.

- E.g., if my doctor tells me after running one or more tests that I have a disease, I would want to know how *certain* this diagnosis is, before I decide on further actions.

Exercise 3

The following training data were obtained:

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

We also have two test observations: x_{test1} = [0 0 0] and x_{test2} = [2 2 0].

a) For both test observations, compute the Euclidian distances to each of the training observations.

b) For kNN with k = 1, compute the predicted class and predicted probability of class red, for each test observation.

c) Do the same for kNN with k = 3.

The true labels of the two observations were y_{test1} = Green and y_{test2} = Red.

d) Compute the misclassification rate, Brier score and cross-entropy for the test observations for $k = 1$ and $k = 3$.

e) As a benchmark, compute the misclassification rate, Brier score and cross-entropy for assigning the test observations to the majority class in the training data (red). Does kNN improve over assigning to the majority class?

Exercise 4

Load the Boston Housing data. We are going to predict median house value in neighbourhoods of Boston:

```
library("MASS"); data(Boston)
```

First, visually inspect distributions and associations in the dataset using function plot.

Select a sample of 400 observations as the training set; use the remaining observations as a test set. E.g.:

```
train <- sample(1:nrow(Boston), size = 400)
```

Fit and evaluate models for predicting `medv`:

Exercise 4 (continued)

a) As a benchmark, first compute the variance of the response variable among the test observations.

b) Fit a linear regression model to the training observations using function `lm`.

c) Fit kNN using function `knn.reg` from library **`FNN`**. Use a `for` loop to fit models for $k = 1$ through 10.

d) For each fitted model, generate predictions for the test observations and compute test MSE.

e) Compare the performance of kNN and OLS. What is the optimal value of k?

Homework

Make the following exercises from chapter 2:

- Exercise 2.1
- Exercise 2.3
- Exercise 2.5