Statistical Learning and Prediction

Marjolein Fokkema

Methodology and Statistics Unit
Leiden University

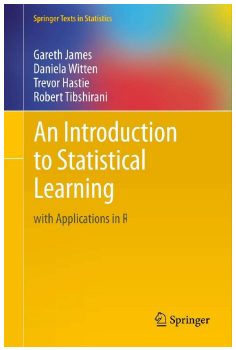
Room 3B20
m.fokkema@fsw.leidenuniv.nl

This Course

- New methodology for data analysis
- Same models, different focus
- Machine Learning / Computer Science
- 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” tab, ”Preparations lectures” tab

Two professors

Dr. Tom Wilderjans

Dr. Marjolein Fokkema

Lectures

1. **Introduction** (Nov 15 & 18; M.F.)

2. **Sampling, logistic regression** (Nov 22 & 25; T.W.)

3. **Classification; subset selection and regularization** (Nov 29 & Dec 2; T.W.)

4. **Unsupervised learning** (Dec 6 & Dec 9; T.W.)

5. **Splines; support vector machines** (Dec 13 & 16; M.F.)

6. **Support vector machines; decision trees** (Dec 20 & 23; M.F.)

7. **Ensembling (of decision trees)** (Jan 10 & 13; M.F.)

8. **Q&A** (Jan 24; M.F. & T.W.)

Evaluation: Assignments

Final grade based on (each with a weight of 1/3):

Assignment	Distributed	Due
1. Written structured assignment	Dec 9	Dec 23 (12:00)
2. Written structured assignment	Dec 23	Jan 17 (17:00)
3. Presentation	Dec 23	Jan 31 (13:00) or Feb 3 (13:00)

1. To be completed individually.

2. To be completed individually.

3. Analysis of a data set of students' own choice; in group of 2 or 3.

Lecture 1a

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)

Lecture (week) aims

Becoming acquainted with:

- Explanation versus prediction
- Method of k-nearest neighbours
- Bias-variance trade-off
- Benefits of shrinkage (bias)
- Overfitting increases near boundaries of sampling space
- Curse of dimensionality

Statistical Learning

- Statistical learning refers to 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: $\hat{Y} = f(X_1, \ldots, X_p)$

  E.g., linear regression: $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p$

  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 estimation of coefficients ($\hat{\beta}_1$ and $\hat{\beta}_2$) is crucial: Accurate estimates = unbiased estimates! I.e.:

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

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 $x_i = [2 \quad 3]$
- 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 $\hat{Y}$. No interest in recovering parameters that generated the data (i.e., explain), but only in obtaining a model that yields as accurate as possible $\hat{Y} = \hat{f}(X)$. That is, minimizeI.e., minimize

$$\mathbb{E}(\hat{Y} - Y)^2$$

Regression

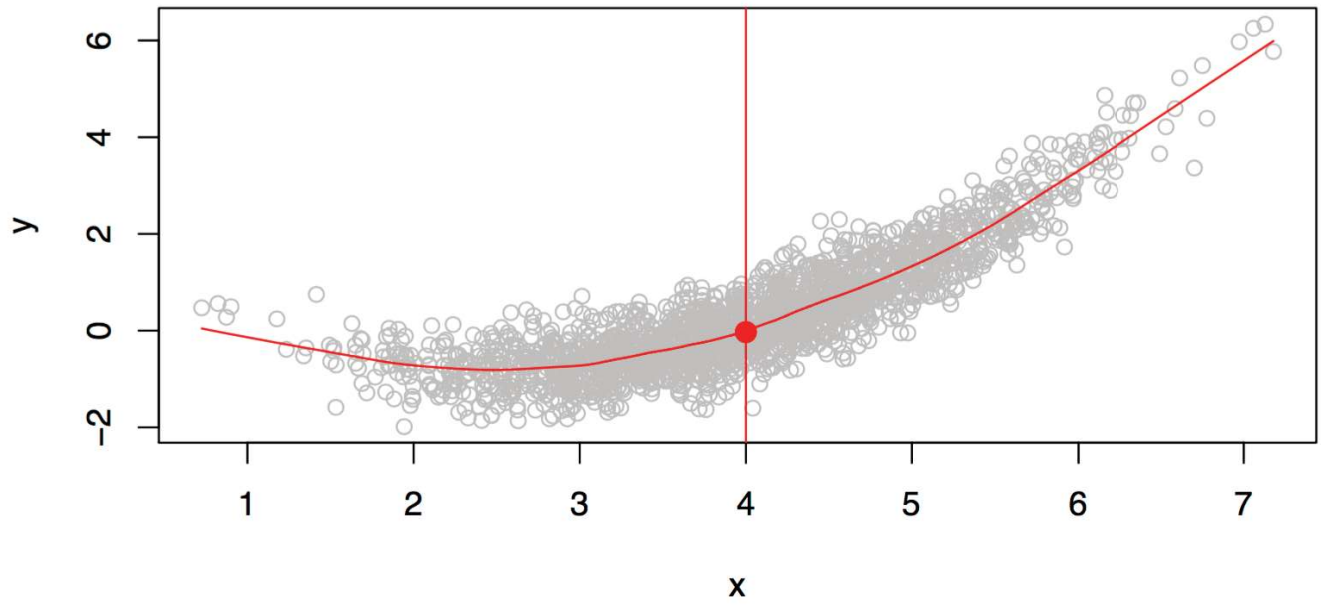
More general, consider we have a population and within this population the conditional means of the response variable ($Y \in \mathbb{R}$) are given by some function of the predictor variables ($X \in \mathbb{R}^p$), that is

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

Generally, we can only collect data from a sample of n persons. These data are denoted $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ and used to train a model $\hat{f}$:

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

Population

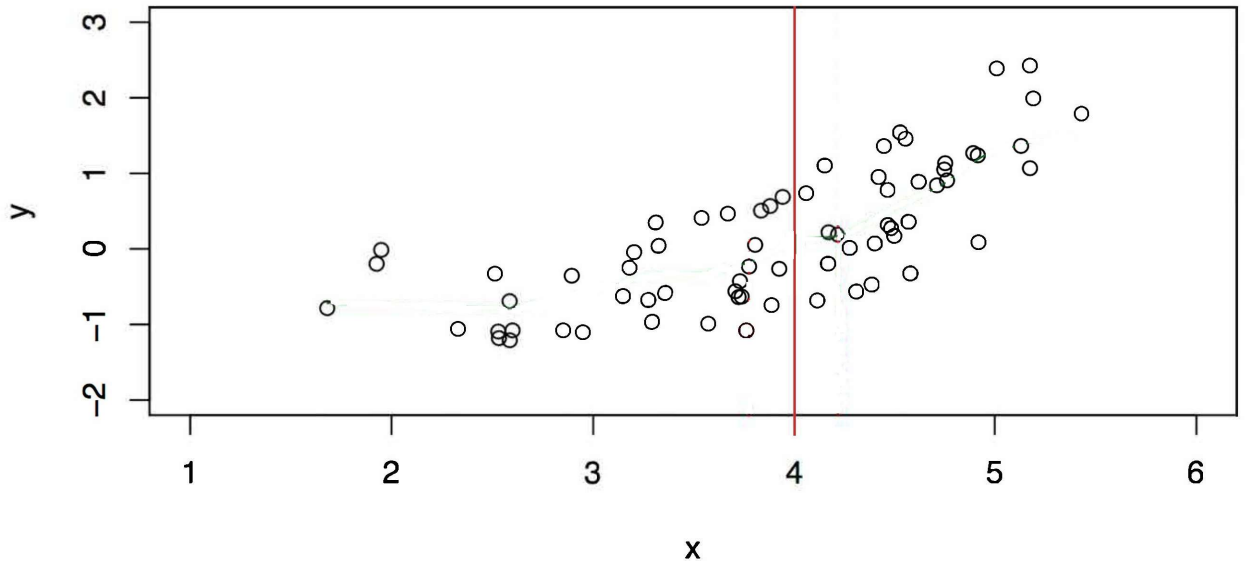


The regression line in the population combines the conditional means at each point x

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

Sample data

In practice, we only have sample data, e.g.:



Due to sparsity we cannot estimate a true conditional mean at all points ($X = x$).

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

Predictive Regression

Suppose we have training data of size n, to which we apply an algorithm to obtain an $\hat{f}(x)$.

We want to minimize the prediction error we would make on a new / future / yet unseen observation $(\mathbf{x}_0, y_0)$:

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

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

Bias-variance trade-off: Elaboration

- We have a probability distribution P^*, we draw a sample $\mathcal{T}$ of size n
- Let f_B be Bayes optimal f (gives the true conditional mean; unknown, depends on P^*)
- Let $\bar{f}(X) = \mathbb{E}_{\mathcal{T}}[\hat{f}(X)]$
- Aim is to minimize *expected prediction error:*

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

Predictive Regression

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

- Note: similar, but not identical, to what OLS minimizes:

$$\hat{\beta} = \arg\min_{\beta}\left(\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2\right) = \arg\min_{\beta}\left(\sum_{i=1}^{n}(y_i - x_i^T\beta)^2\right)$$

- What are the difference(s)?

Predictive Regression

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

- Can or do we ever compute or estimate the quantities in this formula?
- Can you think of a statistical method which would yield least squared bias on any data problem?
- Can you think of a statistical method which would yield least variance on any data problem?
- How is the variance represented in OLS regression?

Predictive Regression

- 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; use function `runif`)
- Generate response $Y = .1X + \epsilon$, with $\epsilon \sim N(0,1)$ (use function `rnorm`)
- Generate $n_{test} = 1,000$ test observations from the same distributions.
- Compute $\hat{\beta}_{OLS}$ using the training observations (use function `lm`; specify ~ 0 in the model formula to exclude the intercept)

Exercise 1: Shrinkage (continued)

- Multiply $\hat{\beta}_{OLS}$ with shrinkage values $c \in \{0, 0.1, \ldots, 0.9, 1.0\}$; generate predictions for the test set $\hat{y}_i = x_i c\hat{\beta}$ (i.e., generate predictions for the test set for each value of c).

- Compute the average squared prediction error or test MSE $= \frac{1}{n_{test}} \sum (y_i - \hat{y}_i)^2$ for each value of c.

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

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

Exercise 1: Shrinkage (continued)

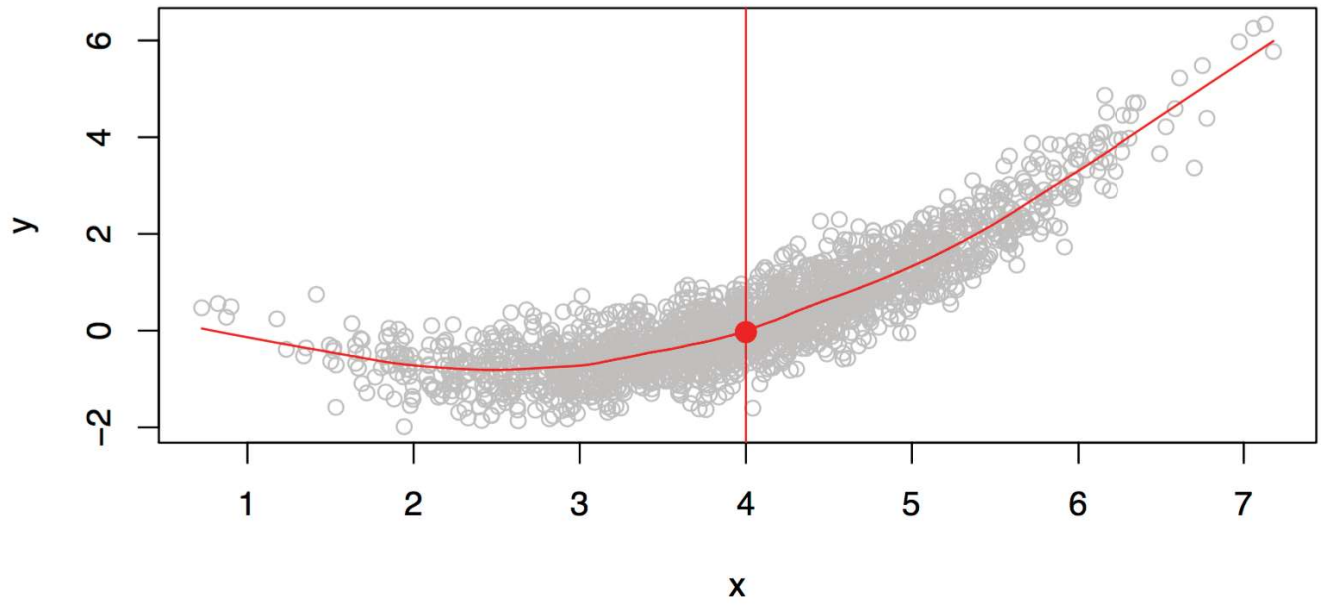
- Repeat the above procedure (except plotting) 100 times (e.g., use a `for` loop)
- Create 2 boxplots, one with the test MSE values on the y axis, one with $c\hat{\beta}_{OLS}$ values for each value as c (x-axis).
- Describe the effect of shrinkage, and when it is most effective.
- Repeat above with doubled sample size (i.e., $N = 100$)
- Repeat above for twice as large effect size (i.e., $Y = .2X + \epsilon$)
- Describe how optimal amount of shrinkage is affected by sample and effect size.

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!

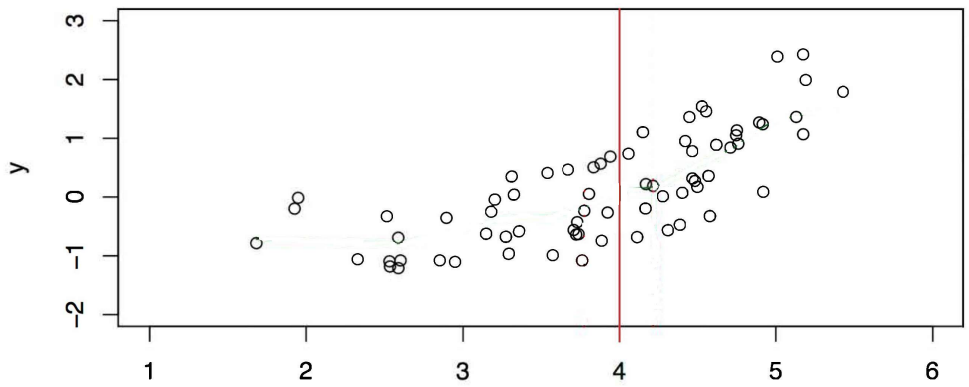
Population: Non-linear Regression



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: (Non-)linear Regression

We obtain sample data:

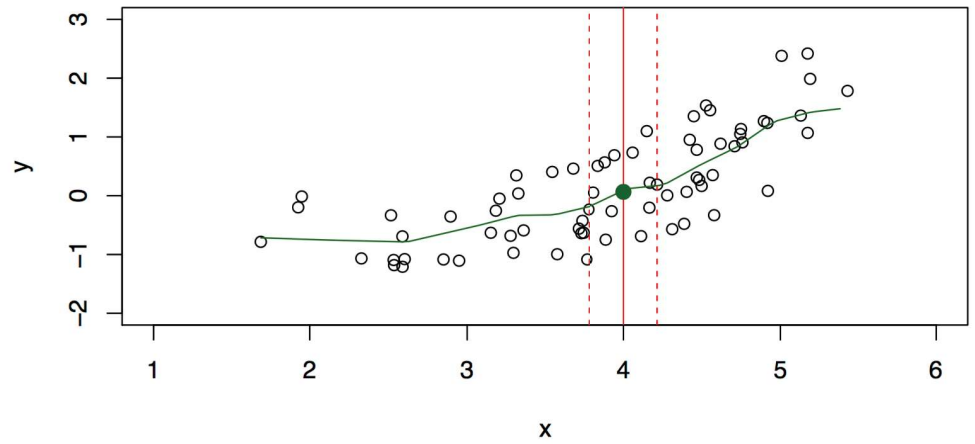


We can make parametric assumptions (for example: linear) and obtain an $\hat{f}(X)$.

- What can we say about the bias this introduces? Variance? 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?

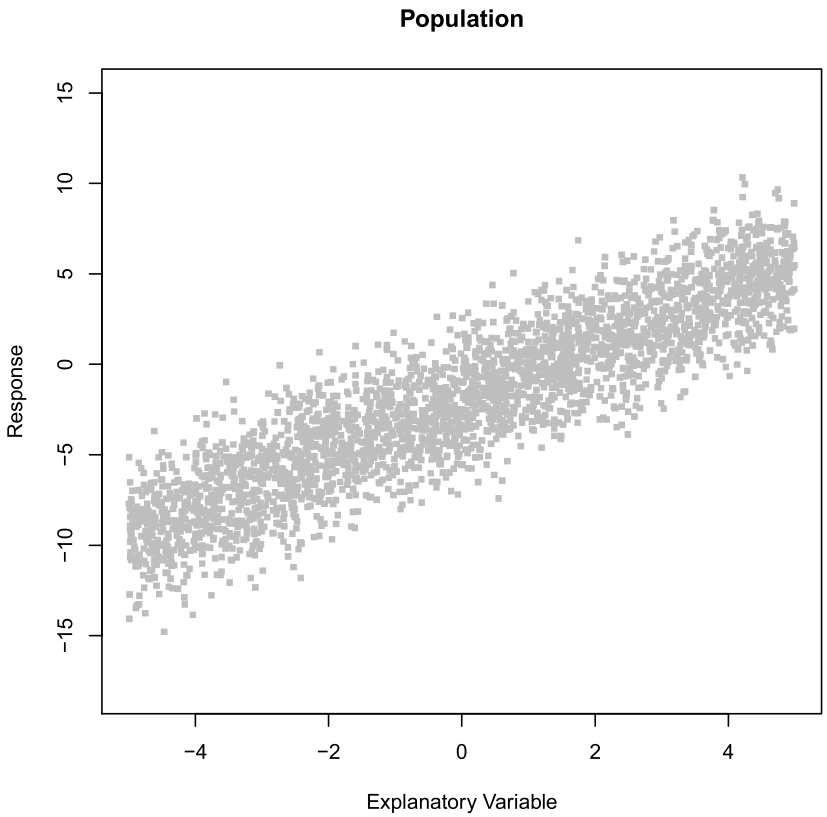
Sample data: Non-linear Regression

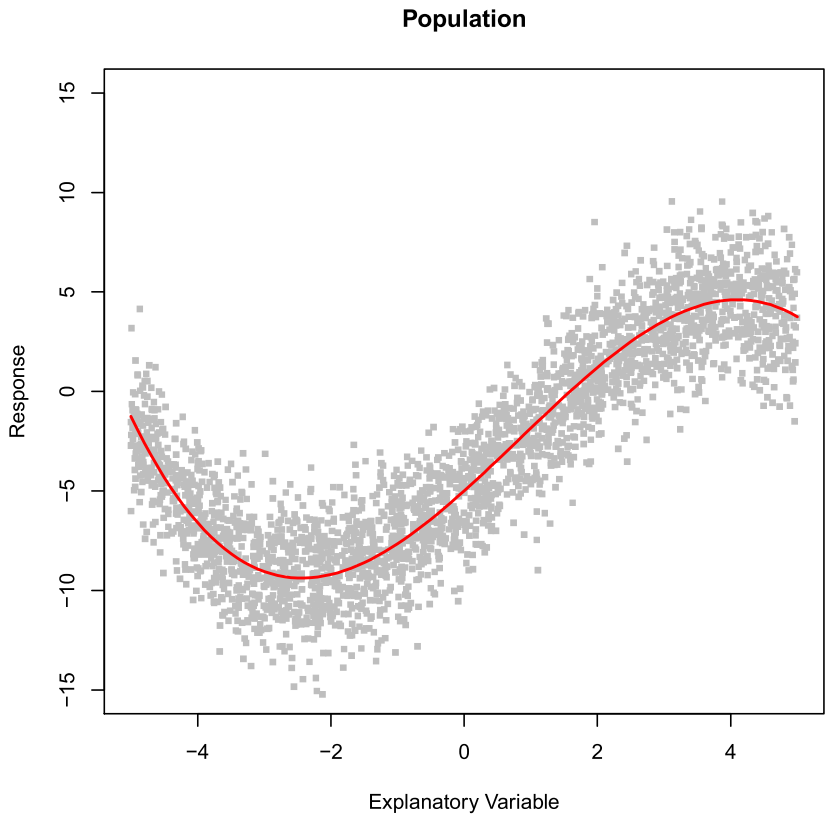
Using sample data, we obtain $\hat{f}(X)$, e.g., using kNN:



- 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?

Populations





What can you say about the bias of a kNN model? And about the bias of an OLS regression model?

-

Excercise 2: Non-Linear Regression

- Generate a training and test set (each of size 50) of data consisting of a single predictor X (uniformly distributed from -5 to 5) and

$$Y = X + 8\sin(X/2) + \epsilon$$

  with $\epsilon \sim N(0, 1)$.

- Fit polynomial regression models to the training data of degree 1 to 15, make predictions on the test set and compute the prediction error for each degree.

- Make a plot with the degree of the polynomial against the prediction error in the test set.

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
- QUESTION: *How can we impose structure?*

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 $\mathcal{C}$ based on X
- Can construct a function $\hat{f}(X)$ that provides conditional probabilities: $\hat{p}_k(X) = Pr(Y = k|X = x)$ (statistically, this is preferred: quantify (un)certainty)
- Bayes classifier assigns $C(X) = k$ if $\hat{p}_k(x) = \max\{\hat{p}_1(x), \ldots, \hat{p}_K(x)\}$

Error measures

- To evaluate classification accuracy, (in ISL book) misclassification rate is often computed:

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

- Alternatively, could compute squared error on predicted probabilities (a.k.a. Brier score):

$$\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{p}(x_i))^2$$

- Q: What are the (dis)advantages of each measure?

Error measures

- For continuous response variables, we can compute MSE:

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

- Alternatively, could compute mean absolute error (MAE):

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

- Q: What are the (dis)advantages of each measure?

Exercise 3: Curse of Dimensionality

- Generate a dataset with $p = 10,000$; $n = 100$; each $X_j \sim N(\mathbf{0}, \mathbf{I})$ (that is, all predictors follow a standard normal distribution and are uncorrelated).

- Create a histogram of the Euclidian distances between all points in the dataset, one for each of $p \in \{1, 2, 10, 100, 1000, 10000\}$ dimensions.

- Hint: use functions `dist` and `hist`; on the x-axis of each histogram, make sure you include 0 and the maximum distance.

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

Exercise 4

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]
- x_{test2} = [2 2 0]

Exercise 4 (continued)

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 test misclassification rate and test MSE on predicted probabilities for $k = 1$ and $k = 3$.

e) Also compute the test misclassification rate for assigning all new observations to the majority class. Does kNN improve over assigning to the majority class?

Exercise 5

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 5 (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