Saarland University

Summary

Elements of Machine Learning

Winter 2020/2021

Lecturer:
Prof. Dr. Isabel Valera
Prof. Dr. Jilles Vreeken

Documentation:
Christian Schmidt

Contents

Introduction

1.1 Advertising

- $X_{1,...,p}$ are input variables (aka predictors, features, independent variables)
- Y is the output variable (aka response, dependent variable)

In general, we assume a relationship between X and Y of the form

$$Y = f(X) + \epsilon = f(X_1, X_2, ..., X_p) + \epsilon$$

where ϵ is a random additive error term with zero mean.

1.2 Why estimate f?

1.2.1 Prediction

Often inputs X are available, output Y is not, but is **desired**.

- estimating the output then effects a **prediction**

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

We often treat $\hat{f}$ as a black box whose form is not of interest.

- e.g., input is blood profile of a patient, and output is the patient's risk of a severe reaction to a drug.
- the accuracy of $\hat{Y}$ depends on the **reducible error** and the **irreducible error**
- for fixed X and f we have

$$\underbrace{E[Y - \hat{Y}]^2}_{\substack{\text{Expectation over all} \\ \text{possible training sets}}} = E[f(X) + \epsilon - \hat{f}(X)]^2 = \underbrace{E[f(X) - \hat{f}(X)]^2}_{\text{reducible error}} + \underbrace{Var(\epsilon)}_{\text{irreducible error}}$$

The goal of prediction is to minimize the reducible error. The irreducible error cannot be avoided.

1.2.2 Inference

Often we want to go beyond treating $\hat{f}$ as a black box. Rather, we want to **understand the relation** between input and output

- which predictors strongly associate with the response? (Often only few)
- what is the relationship between the response and each predictor? Is it positive or negative? (Sometimes this depends on other predictors)
- is the relationship between the predictors linear or more complicated?

An **example** for inference is the advertising data with questions as:

- which media contribute to sales? which generate the biggest boost?
- how much increase in sales is associated with a given increase in TV ads?

Sometimes both prediction and inference are of interest. There is, however, a **tradeoff** between the two. Linear models, for example, allow easily interpretable predictions but may not be very accurate.

1.3 How to estimate f?

We have a set of n **observations** with inputs and outputs (training data), $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, and are looking for a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y).
In the following we dinstinguish between **parametric** and **nonparametric** methods:

1.3.1 Parametric Methods

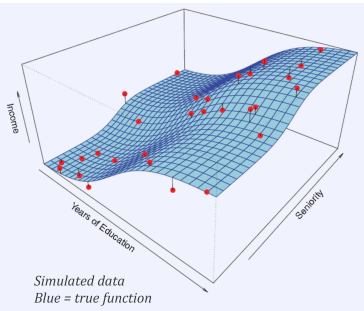
- have a given functional form, usually simple such as a linear model

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

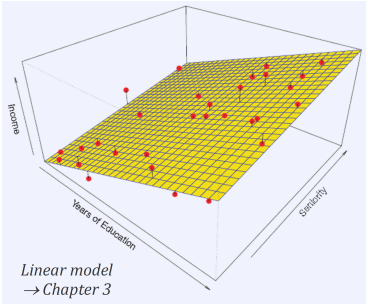
- estimating $\hat{f}$ means choosing the model parameters.
- **problem:** the model may not match the true form of f.

1.3.2 Nonparametric Methods

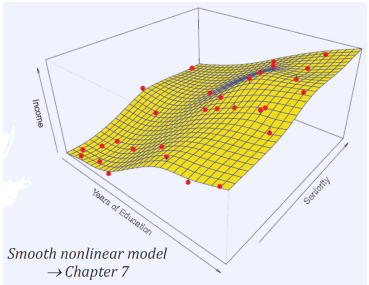
- here we aim at finding the form of f
- choosing the form gives us much more freedom
- we have to choose many parameters; this requires many observations
- otherwise, we risk modelling the noise in the training set: **overfitting**



(a)



(b)



(c)

1.4 Accuracy vs. Interpretability

Why would we ever prefer a more restricted model over a more flexible one?

A flexible model entails a large number of parameters

1. Estimating all parameters is computationally expensive.
2. Complicated models are hard to interpret, so especially when inference is the goal, simple models are preferred.
3. If we have only few observations, we do not have enough information to accurately estimate many parameters. In such cases flexible models incur a high risk of overtraining.

1.5 Supervised vs. Unsupervised Learning

1.5.1 Supervised Learning

- **Data:** inputs and outputs (x_i, y_i) for observations $i = 1, ..., n$ following some unknown functional pattern with noise, e.g. $Y = f(X) + \epsilon$
- **Goal:** find function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for every conceivably seen input X
  (setting is like that of a student who learns from a teacher (supervisor) giving examples)

1.5.2 Semi-supervised Learning

- **Data:** inputs x_i for observations $i = 1, ..., n$, only some outputs y_i
- **Goal:** same as for supervised learning, but also leverages unlabeled data

1.5.3 Unsupervised Learning

- **Data:** inputs x_i for observations $i = 1, ..., n$, no outputs
- **Goal:** elucidate relationships between the variables or the observations
  (often equated with cluster analysis, but many more aspects exist)

1.6 Assessing model accuracy

In regression problems we use the **mean squared error** to assass the quality of fit, here over the training data

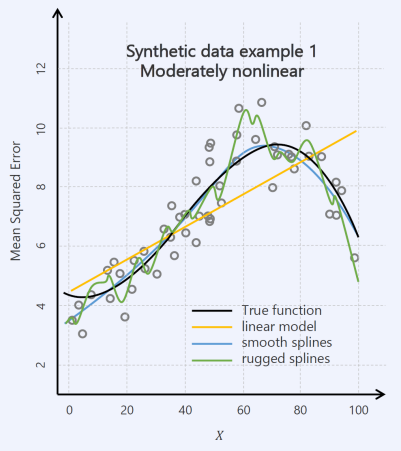
$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$$

We call this the **training error**.

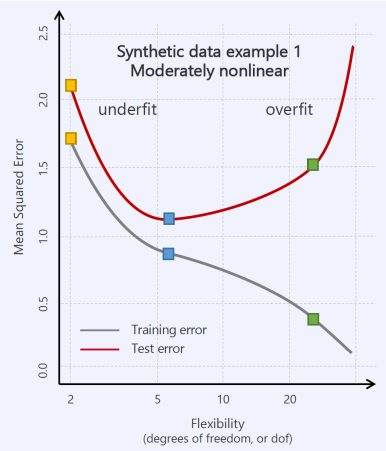
We are more interested in the error over unseen data (x_0, y_0)

$$Ave(\hat{f}(x_0) - y_0)^2$$

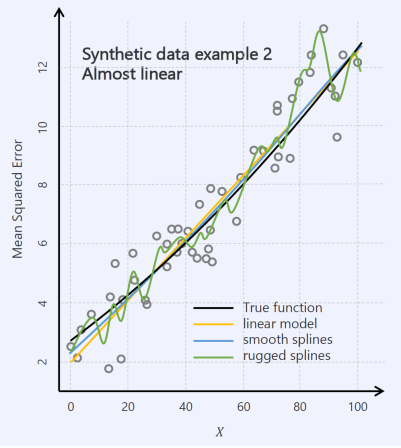
We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE).
If the functional dependence between input and output is not known, the test error is hard to estimate.


Synthetic data example 1
Moderately nonlinear
Mean Squared Error
X
True function
linear model
smooth splines
rugged splines

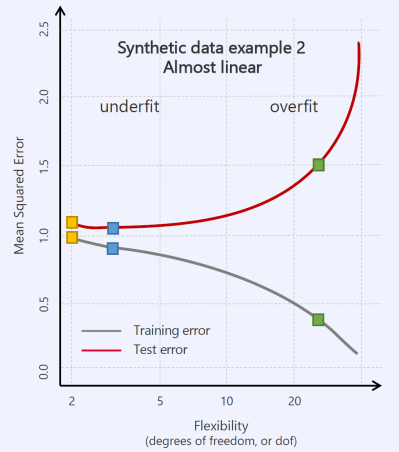
(a)


Synthetic data example 1
Moderately nonlinear
underfit
overfit
Mean Squared Error
Training error
Test error
Flexibility
(degrees of freedom, or dof)

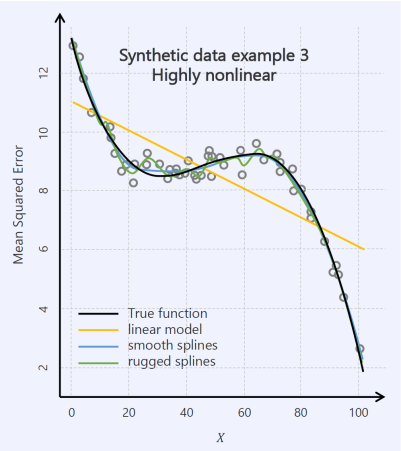
(b)


Synthetic data example 2
Almost linear
Mean Squared Error
X
True function
linear model
smooth splines
rugged splines

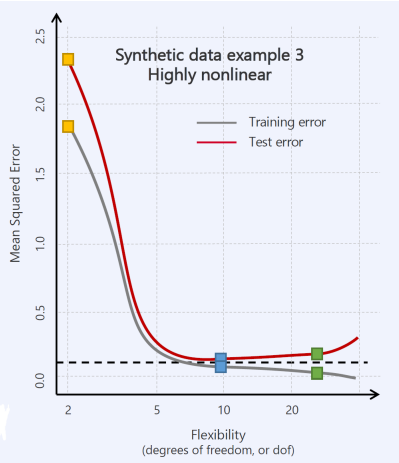
(a)


Synthetic data example 2
Almost linear
underfit
overfit
Mean Squared Error
Training error
Test error
Flexibility
(degrees of freedom, or dof)

(b)


Synthetic data example 3
Highly nonlinear
Mean Squared Error
X
True function
linear model
smooth splines
rugged splines

(a)


Synthetic data example 3
Highly nonlinear
Mean Squared Error
Training error
Test error
Flexibility
(degrees of freedom, or dof)

(b)

1.7 Bias-Variance

1.7.1 Tradeoff

The shape of the curve for test error is due to a basic tradeoff in the MSE

$$E[y_0 - \hat{f}(x_0)]^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

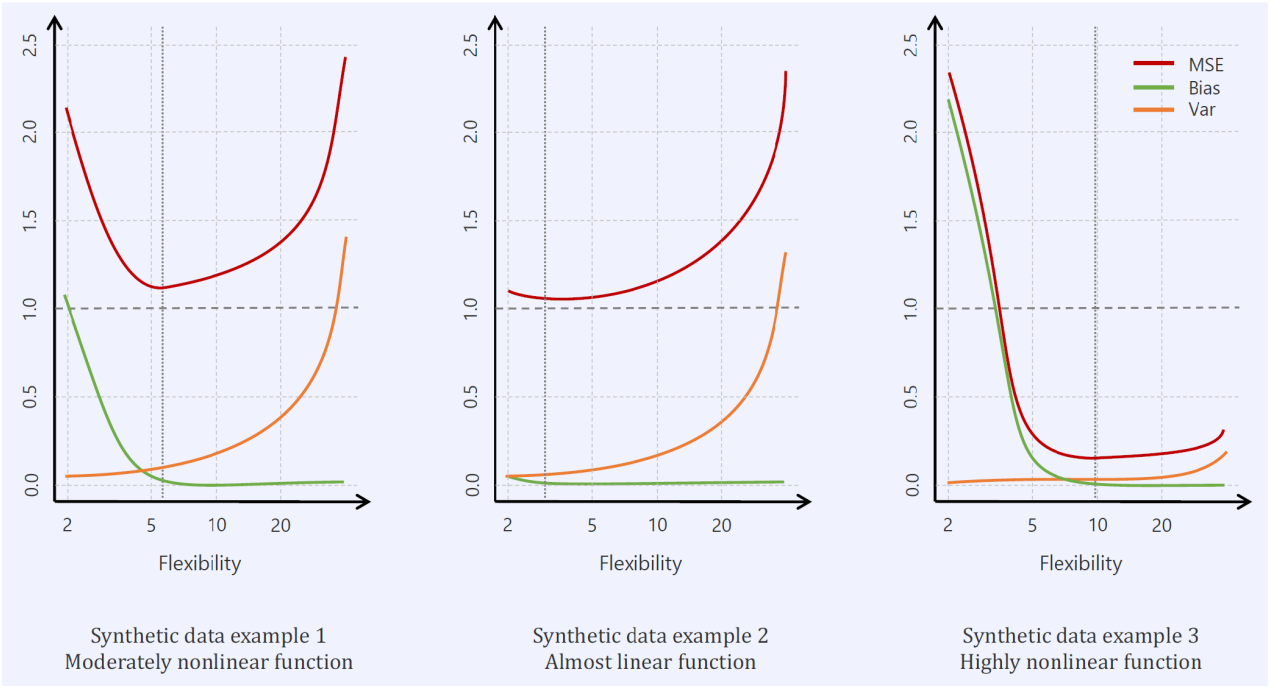
Here **bias** is the systematic deviation of the estimate from the true value

$$Bias(\hat{f}(x_0)) = E[\hat{f}(x_0) - y_0]$$

and **variance** is the variation of the estimate between different training sets

$$Var(\hat{f}(x_0)) = E[\hat{f}(x_0) - E[\hat{f}(x_0)]]^2$$

1.7.2 Bias-Variance Decomposition



Synthetic data example 1
Moderately nonlinear function

Synthetic data example 2
Almost linear function

Synthetic data example 3
Highly nonlinear function

1.8 The Classification Setting

A popular method of measuring classification error (**loss function**) is the **misclassification error**.

On the training set is

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

where for a predicate p,

- $I(p) = 1$ if $p = true$ and
- $I(p) = 0$ otherwise.

The test error is

$$AVE(I(y_0 \neq \hat{y}_0))$$

1.8.1 Bayes Classifier

The test error is minimized by the following very simple calssifier

$$arg \underset{j=1,...,k}{max} Pr[Y = j \mid X = x_0]$$

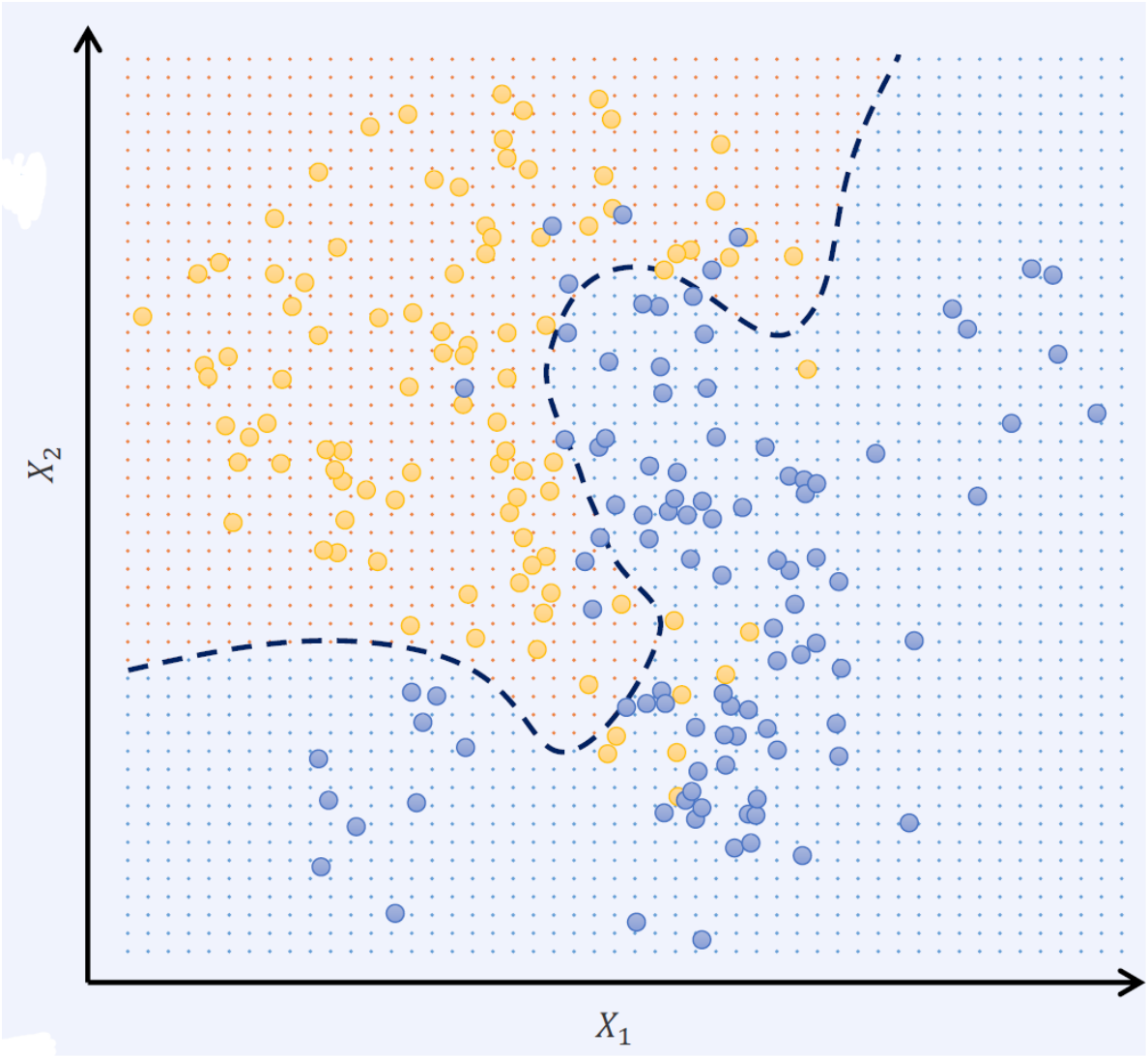
for a classification problem with k classes $1, ..., k$.

This classifier can be computed on **synthetic data** for which the **probability distribution is known**, but not for real data, as we do not know the probability distribution.

1.9 Example Binary Classification

100 observations in each of two groups, synthetic data with noise.

- **Bayes decision boundary:** points with $Pr[Y = 1 \mid X = x_0] = 0.5$ is dashed.
- **Bayes error rate:** the irreducible error $1 - E[\underset{j=1,2}{max} Pr[Y = j \mid X]]$ in genaral, and 0.1304 for this example.

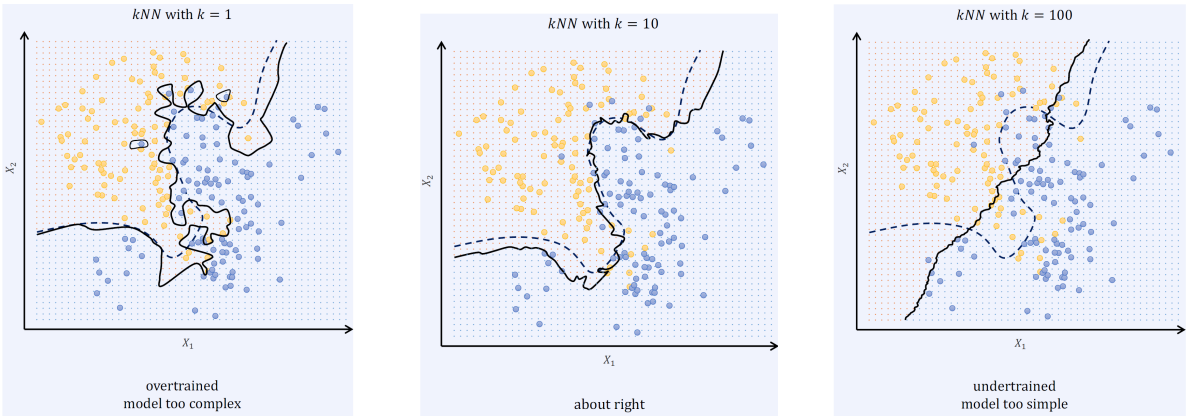


1.9.1 Nearest Neighbors

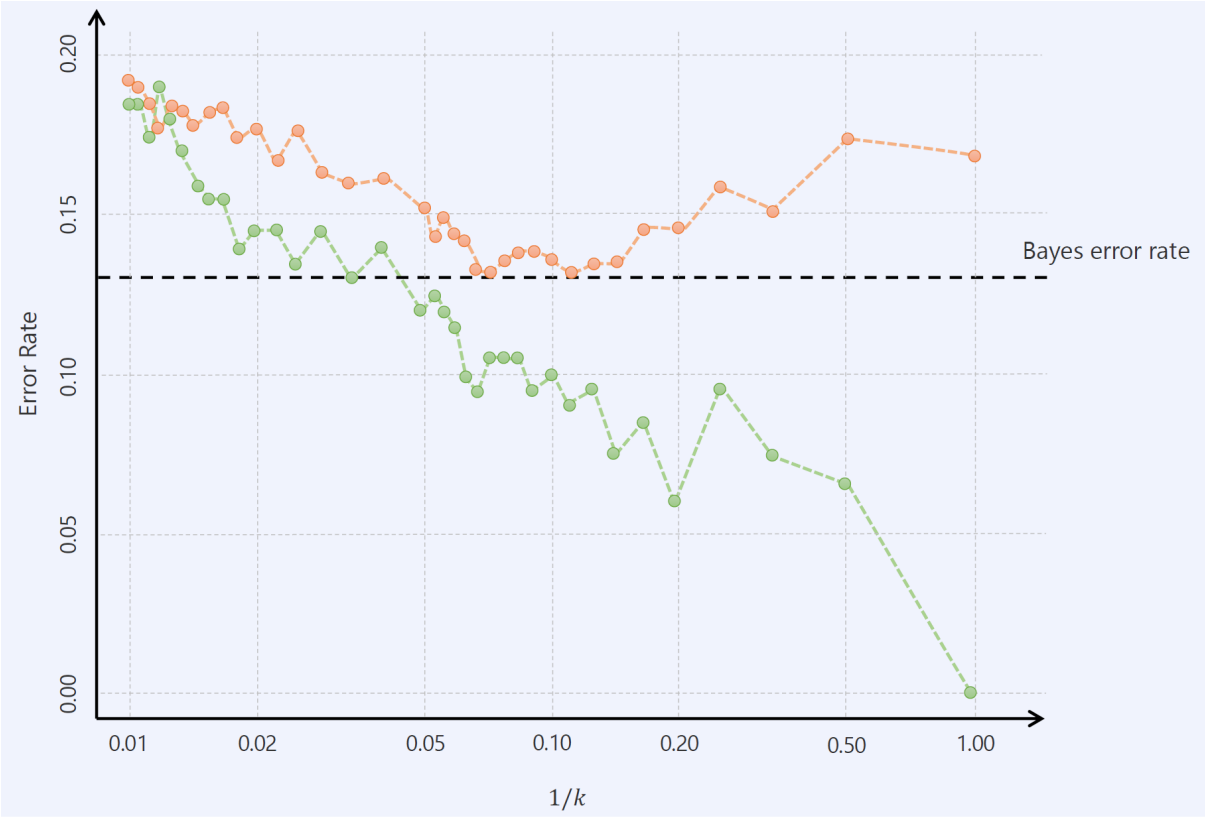
k-nearest neighbors (kNN)
Classifies each pointto the majority class among its k nearest neighbors

$$arg \max_{j=1,\ldots,k} \frac{1}{k} \sum_{\mathbb{N}_0} I(y_i = j)$$

where $\mathbb{N}_0$ is the set of the k data points nearest to x_0.



(a) **(b)** **(c)**



Linear Regression

2.1 Simple Linear Regression

We assume the following relationship among the data

$$Y \approx \beta_0 + \beta_1 X$$

We regress Y on (onto) X

$$sales \approx \beta_0 + \beta_1 \times TV$$

This is a simple **unvariate model**. β_0, intercept, and β_1 slope, both are **coefficients** or **parameters**. The estimated value of Y on input $X = x$ is denoted by

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

Given training data set of n observations

$$(x_1, y_1), (x_3, y_2), ..., (x_n, y_n)$$

Goal estimate the unknown coefficients β_0 and β_1 such that

$$y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$$

for all $i = 1, ..., n$ and for future values of x.

2.2 Estimating the Coefficients

We We measure deviation of the estimate to the true value by a **loss function**. We mostly use the **least-squares** function for this purpose.
Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, then $e_i = y_i - \hat{y}_i$ is the **residual sum of squares (RSS)**

$$\begin{aligned} RSS &:= e_1^2 + e_2^2 + ... + e_n^2 \\ &= (y_1 - \hat{\beta}_0 + \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 + \hat{\beta}_1 x_2)^2 + ... + (y_n - \hat{\beta}_0 + \hat{\beta}_1 x_n)^2 \end{aligned}$$

This is a quadratic function in β_0 and β_1. Setting its derivative to zero yields the least-square coefficient estimates.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})^2} \qquad \text{and} \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

with

$$\hat{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad \text{and} \qquad \hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

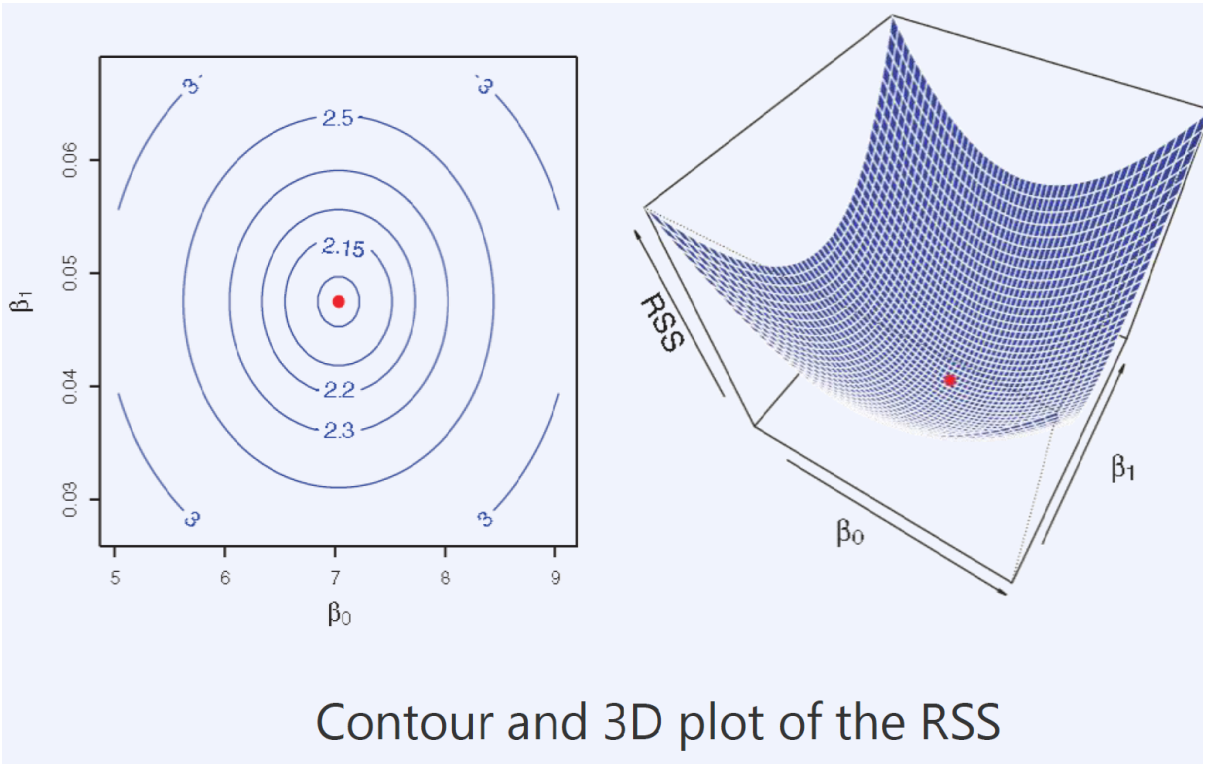


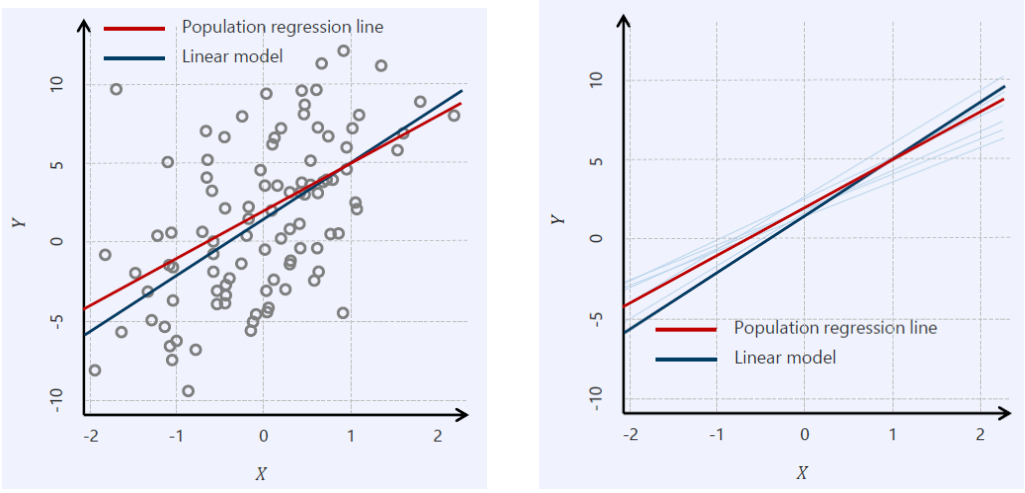
Figure 2.1: Contour and 3D plot of the RSS

2.3 Accuracy of Coefficient Estimates

We assume the true relationship has an additional noise component that is independent from observations

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (*)$$

This is the **population regression line**, the best linear approximation to the true relationship between X and Y, given that the true relationship is (*). The population regression line is usually unobserved. The least-squares fit on the trining data is given by $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. For example on figure (a) we see



(a) **(b)**

Population regression line (red) and least squares fit (blue) on simulated data $= 2 + 3 + \epsilon$ with Gaussian error ϵ with 0 mean.
And on figure (b) we see Ten least squares fits on different randomly chosen training data sets.

2.4 Unbiased Estimates

How do we estimate the mean μ of a random variable Y?

- the sample estimate over a finite set of observations is the average

$$Ave(y_1, y_2, ..., y_n) = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}$$

- on average, we have $\bar{y} = \mu$. $\bar{y}$ is an unbiased estimate for μ

In same fashion, **the least-square fit is an unbiased estimate for the population regression line**!

- furthermore, among all unbiased linear estimators, the least square fit is the one with the smallest variance (**Gauss Markov Theorem**)

2.4.1 Assessing the Accuracy of Estimates

The **standard error** of μ is

$$SE(\mu) = \sqrt{Var(\mu)} = \sqrt{\frac{\sigma^2}{n}}$$

- σ population standard deviation item n sample size

We assume (as we usually do) that all observations are independent.
The **more** observations we have, the **smaller** the standard error.
The standard errors of the least-square coefficients are

$$SE(\hat{\beta_0})^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$
$$SE(\hat{\beta_1})^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where $\sigma^2 = Var[\epsilon]$.
Again we assume (mostly incorrectly) that errors are independent and uncorrelated, with the common variance σ^2.

Observations

1. $SE(\hat{\beta_1})$ decreases as the x_i are more spread out - the slope is easier to determine.
2. $SE(\hat{\beta_0}) = SE(\hat{\mu})$ if $\bar{x} = 0$ in which case $\hat{\beta_0} = \bar{y}$.
3. σ is not known but we can provide a sample estimate for it, the residual standard error

$$RSE = \sqrt{\frac{RSS}{(n-2)}}$$

2.5 Computing Confidence Intervals

95% **confidence interval**

- interval that will contain the true value with 95% probability
- limits are computed from the sample (training) data

For the linear regression coefficient $\hat{\beta}_1$ it takes approximately the form

$$[\hat{\beta}_1 - 2 \cdot SE(\hat{\beta}_1) \ , \ \hat{\beta}_1 + 2 \cdot SE(\hat{\beta}_1)]$$

analogously for $\hat{\beta}_0$

$$[\hat{\beta}_0 - 2 \cdot SE(\hat{\beta}_0) \ , \ \hat{\beta}_0 + 2 \cdot SE(\hat{\beta}_0)]$$

Why is this the case? (ESL)

- we assume that the error in the output is Gaussian distributed
- the coefficient estimates are then also Gaussian distributed (!)

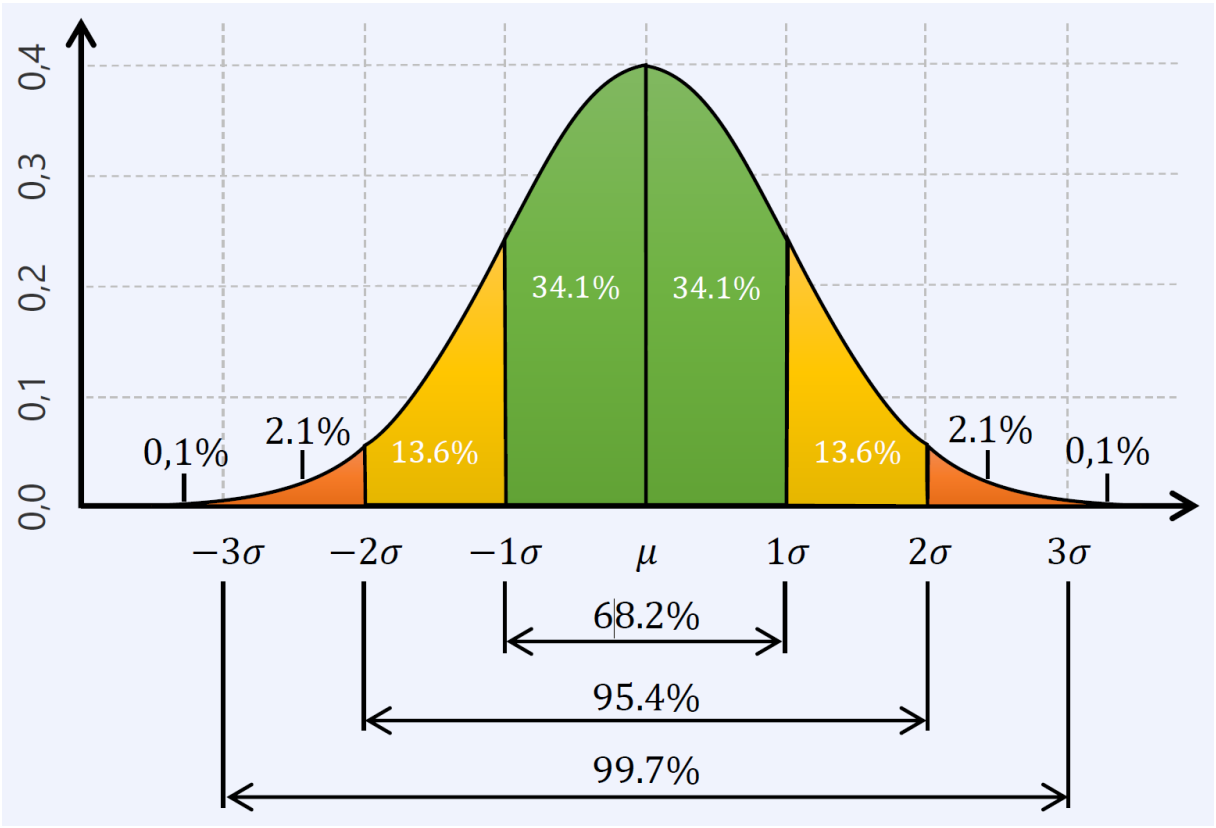


Figure 2.2: Probability mass in a Gaussian