Department of Mathematics
ETH Zurich

CAS ETH Machine Learning in Finance and Insurance

BLOCK I. Introduction to Machine Learning. Lecture 1.

Dr. A. Ferrario, ETH Zurich and UZH

Last week we…

1. Introduced the goals and structure of **BLOCK I: Introduction to Machine Learning**
2. Discussed the definitions of machine learning, how to do it in practice and learn it efficiently
3. Reflected on (y)our status-quo with respect to machine learning

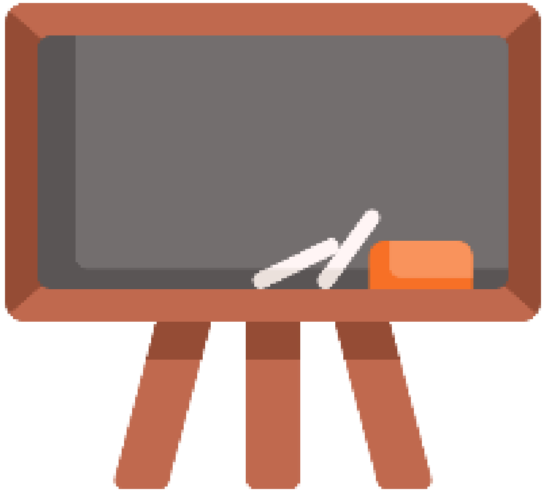
Introducing some Formalism for Machine Learning

- Mathematical background
- Statistical learning

Introducing some Formalism for Machine Learning

Mathematical background

We will use slides to introduce our topics and the blackboard to deep-dive into selected items



To manage “uncertainty”, we need to introduce three concepts

“Uncertainty of events”

I	Probability spaces
II	Random variables
III	Selected random variables and their distributions

“events are said to *occur* or *happen* or *take place*” (Stanford Encyclopedia of Philosophy – “Events” page)

Probability spaces

Probability space	$(\Omega, \mathcal{F}, \mathbb{P})$ Ω is a non-empty set *(sample space)* $\mathcal{F}$ is a σ*-algebra* of subsets of Ω $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a *probability measure*

Probability spaces

The elements of $\mathcal{F}$ are subsets of Ω which are called **events**. In Jacod and Protter's words: "An 'event' is a property which can be observed either to hold or not *after* the experiment is done" (pag. 3, emphasis in original)

Probability space

$(\Omega, \mathcal{F}, \mathbb{P})$

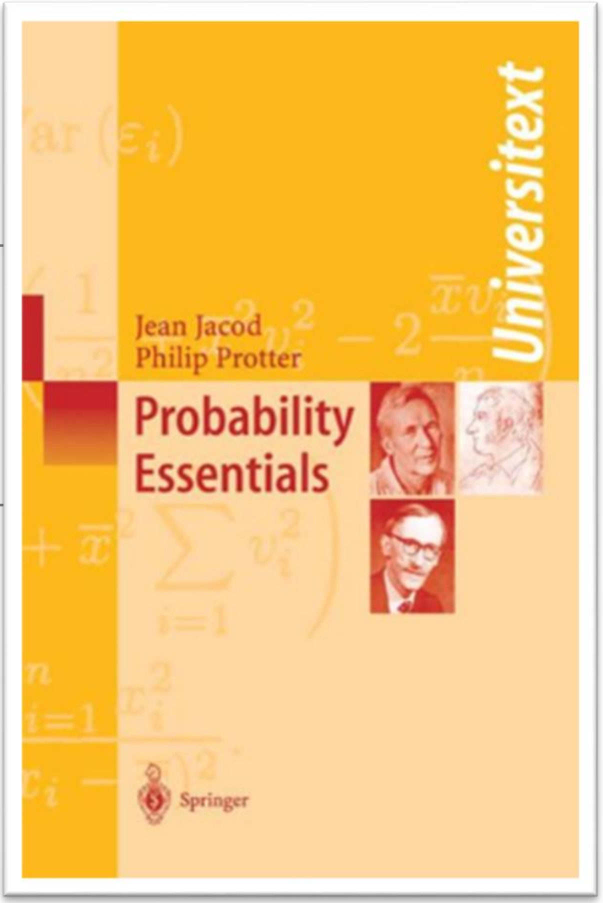
Ω is a non-empty set *(sample space)*

$\mathcal{F}$ is a σ-*algebra* of subsets of Ω

$\mathbb{P} : \mathcal{F} \to [0, 1]$ is a *probability measure*

Interpretation: probability spaces formalize the space of outcomes of "random experiments". Using Jacod and Protter's words:

"Random experiments are experiments whose output cannot be surely predicted in advance. But when one repeats the same experiment a large number of times one can observe some 'regularity' in the average output" (pag. 3)



Jacod, J., & Protter, P. (2004). *Probability essentials*. Springer Science & Business Media.

Probability spaces and random variables

Probability space	$(\Omega, \mathcal{F}, \mathbb{P})$ Ω is a non-empty set *(sample space)* $\mathcal{F}$ is a *σ-algebra* of subsets of Ω $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a *probability measure*
Random variables	$X : \Omega \to \mathcal{X}$ *(input variables/independent variables/features/covariates)*, e.g. $\mathcal{X} = \mathbb{R}^d$ $Y : \Omega \to \mathcal{Y}$ *(output variable/dependent variable/response variable)*, e.g. $\mathcal{Y} \subseteq \mathbb{R}$ (measurable)

Probability spaces and random variables

Probability space

$(\Omega, \mathcal{F}, \mathbb{P})$

Ω is a non-empty set *(sample space)*

$\mathcal{F}$ is a σ-*algebra* of subsets of Ω

$\mathbb{P} : \mathcal{F} \to [0, 1]$ is a *probability measure*

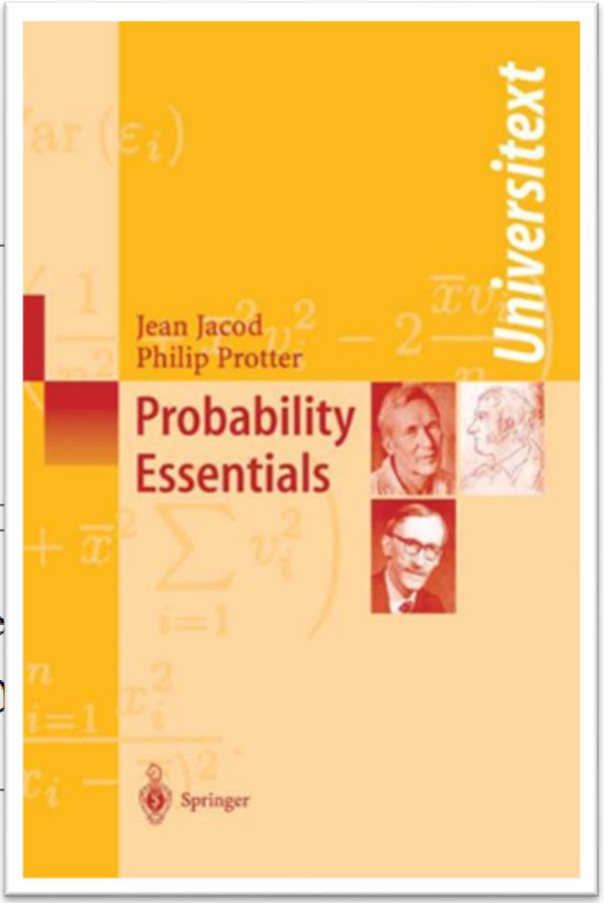
Random variables

$X : \Omega \to \mathcal{X}$ *(input variables/independent variables/features/covariates)*, e

$Y : \Omega \to \mathcal{Y}$ *(output variable/dependent variable/response variable)*, e.g.

Intepretation. Again, Jacod and Protter help us:

“A random variable represents an unknown quantity (hence the name variable) that […] varies with the outcome of a random event. Before the random event, we know which values [the random variable] could possibly assume, but we do not know which one it will take until the random event happens.” (pag. 27)



Jacod, J., & Protter, P. (2004). *Probability essentials*. Springer Science & Business Media.

Probability spaces and random variables

Probability space

$(\Omega, \mathcal{F}, \mathbb{P}$

Ω is a n

$\mathcal{F}$ is a σ

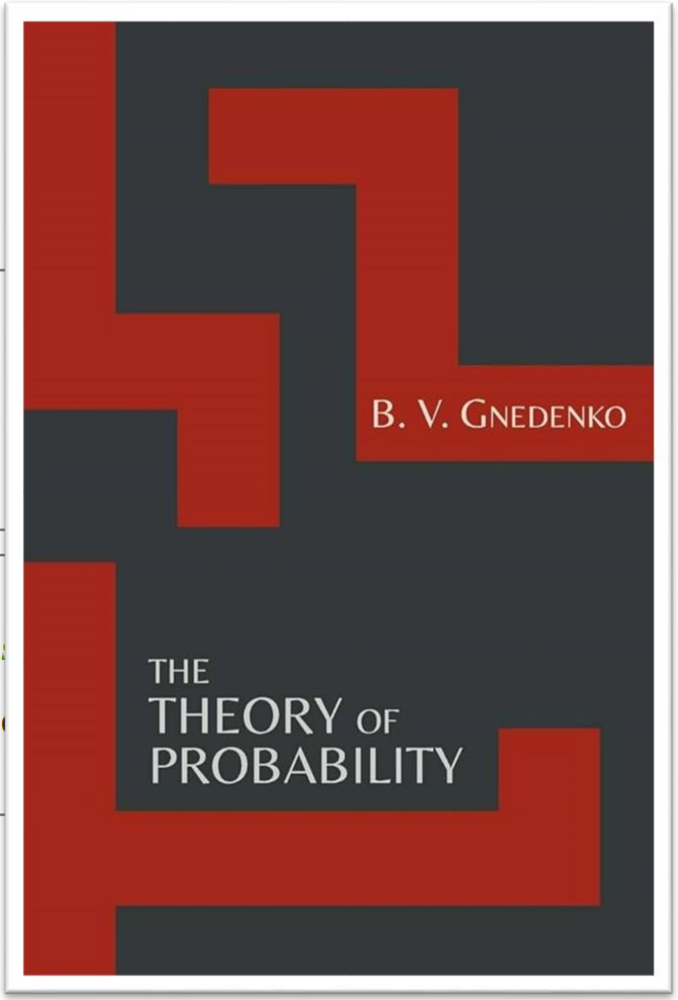
$\mathbb{P} : \mathcal{F}$ –

Random variables

$X : \Omega$ –

$Y : \Omega$ –

Gnedenko's explanation of random variables:
"Each of these [random] variables is capable of taking on various values, under the influence of chance. To specify in advance which value the variable will take on is impossible, since it changes in random fashion from experiment to experiment. […].
Let us summarize the above: a *random variable* is a variable quantity whose values depend on chance and for which there exists a distribution function." (pag. 146, emphasis in original)



B.V. Gnedenko, (1963). The Theory of Probability. Chelsey Publishing Company.

Selected random variables, distributions and expected values

We briefly introduce a few r.v.'s that we will need later

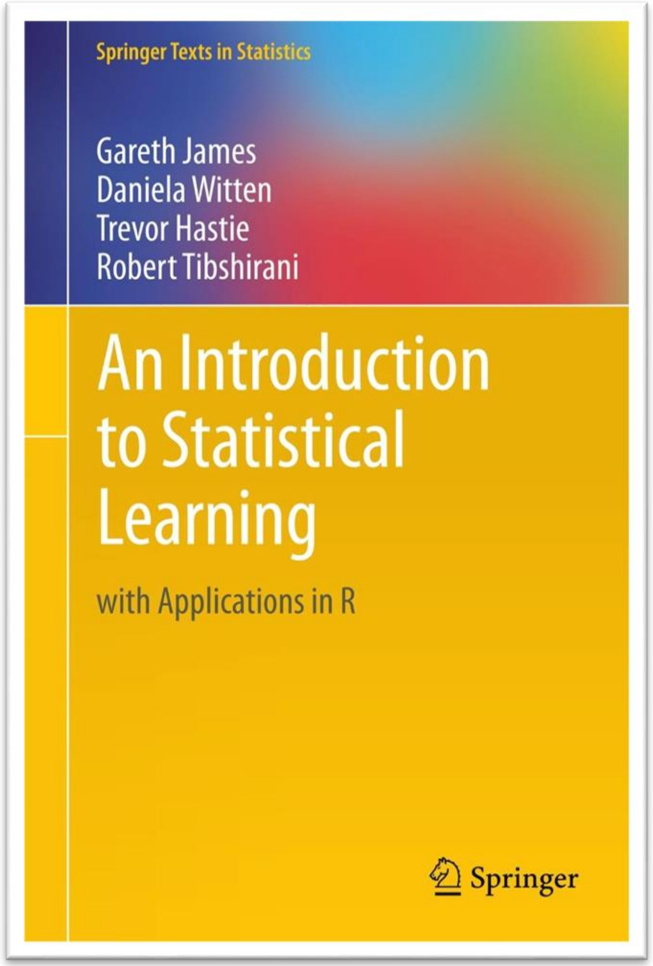
Bernoulli (trial)	A Bernoulli trial is a random experiment with two outcomes (1/0, true/false, success/failure etc.). The probability of success is the same every time the experiment is conducted.
Poisson	The Poisson distribution expresses the probability of a given number of events occurring in a fixed interval of time if these events occur with a known constant mean rate and independently of the time since the last event.
Continuous random variables	Key examples: uniform and normal distributions

Introducing some Formalism for Machine Learning

Statistical learning

Statistical learning
Definition

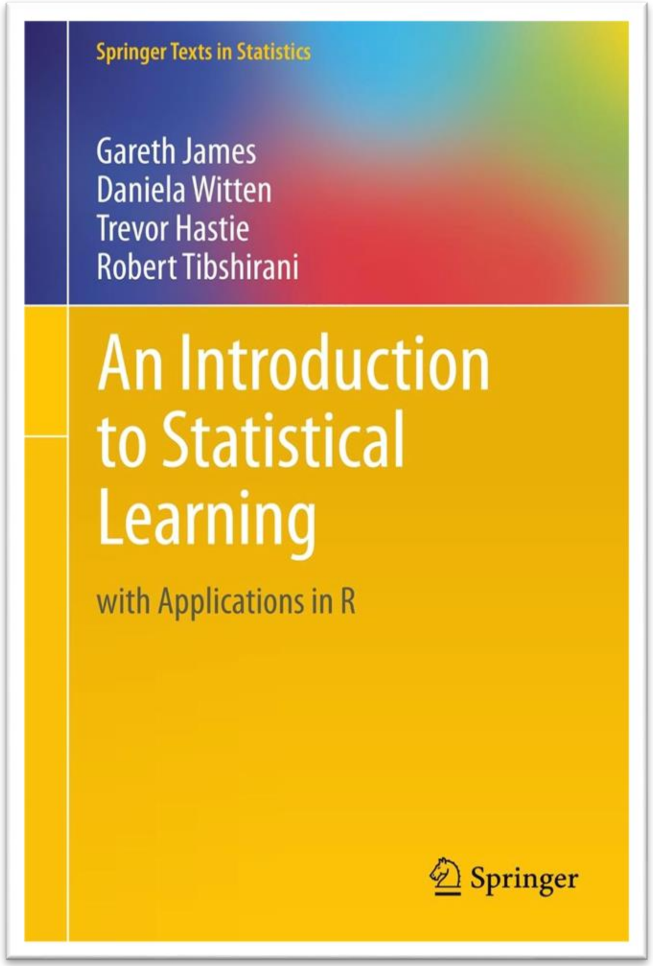
"*Statistical learning* refers to a vast set of tools for *understanding data*." (pag.1, emphasis in original)



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer

Statistical learning
Supervised and unsupervised learning

“These [statistical learning tools] can be classified as supervised or unsupervised. Broadly speaking, supervised statistical learning involves building a statistical model for predicting, or estimating, an *output* based on one or more *inputs.* With unsupervised statistical learning […] we can learn relationships and structure from data.” (pag.1, emphasis in original)



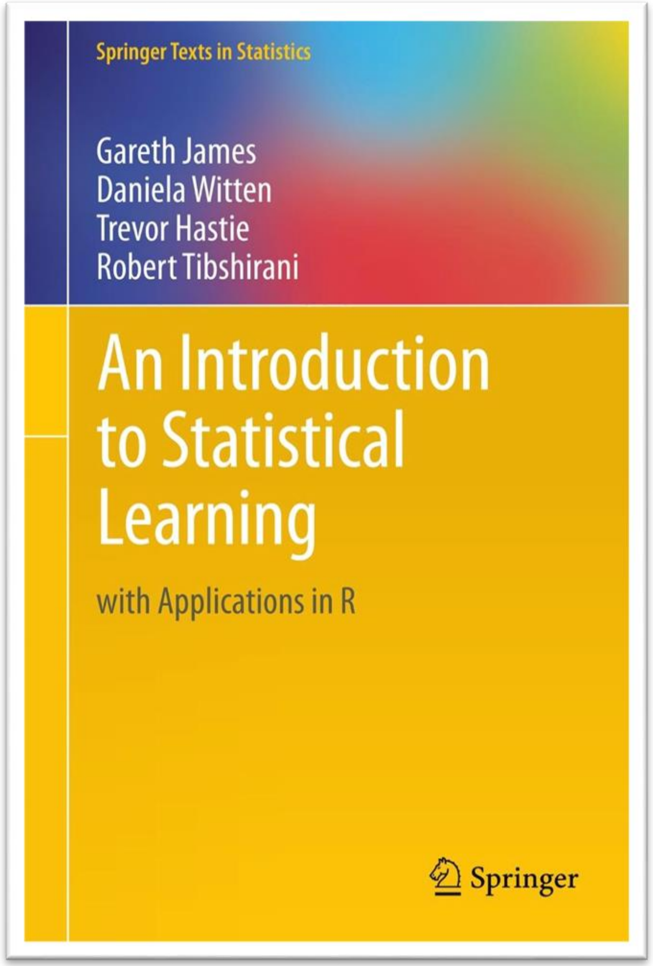
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer.

Statistical learning
Machine learning methods allows us *predicting* in supervised learning problems

Y predicted using… X	
Stock market increase/decrease	Past five days' percentage changes in S&P index
Wage of US males	Socio-demographic information
Price of car insurance policy	Socio-demographic and policy information, driving behaviour
Stress level of an employee	Behavioural data (mouse and keyboard movements) and cardiac activity
Handwritten digit	Images of handwritten digits
Mental health status of an individual	Textual data
…	…

Output, outcome or dependent variable, response

Input, independent variables, features, predictors



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer.

Supervised Learning
Some formalization

Key assumption

Let $(Y, X),\ \text{where } X = (X_1, \ldots, X_d)$.

We assume there exists an **unknown, fixed relationship** between the output r.v. and the input r.v.'s. This relationship holds up to a random error that is independent of the input r.v.'s and has expected value equal to zero.

$$\boxed{Y = f(X) + \epsilon} \quad \epsilon \text{ indep. of } X,\ \mathbb{E}[\epsilon] = 0$$

*

Supervised Learning
Some formalization

Key assumption

Let (Y, X), where $X = (X_1, \ldots, X_d)$.

We assume there exists an **unknown, fixed r**
and the input r.v.'s. This relationship holds up t
of the input r.v.'s and has expected value equa

$$\boxed{Y = f(X) + \epsilon}$$

1

True model: systematic information. In most applications, it is unknown.

2

It is not realistic to assume a (perfectly) deterministic relationship in **all** applications. The error terms encodes approximations in measurements, information not encoded by the input r.v.'s. In other words:

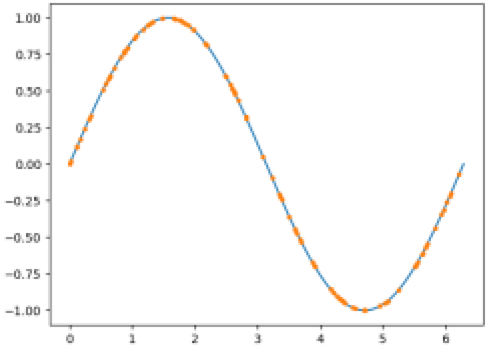
"Generally there will be other unmeasured variables that also contribute to Y, including measurement error. The additive model assumes that we can capture all these departures from a deterministic relationship via the error ϵ" (pag. 28, Hastie et al. The Elements of Statistical Learning book).

"captures measurement errors and other discrepancies" (Hastie - https://www.youtube.com/watch?v=ox0cKk7h4o0)

A couple of examples with simulations – here the fixed relationship is known, as we define it

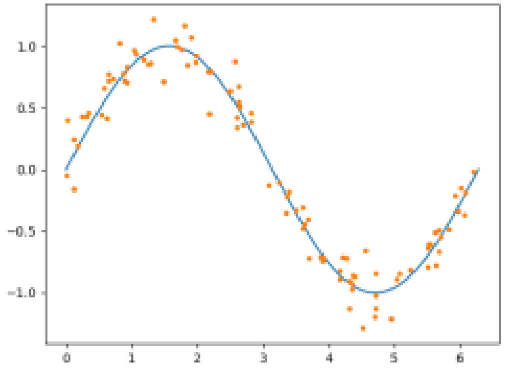
- Deterministic dependence: $Y = f(X)$

  e.g. $X \sim \mathrm{Unif}(0, 2\pi)$, $Y = \sin(X)$
  (X_i, Y_i), $i = 1, \ldots, 100$,
  indep. copies of (X, Y)



- Homoscedastic additive noise: $Y = f(X) + \epsilon$ for ε independent of X with $\mathbb{E}[\varepsilon] = 0$,

  e.g. $X \sim \mathrm{Unif}[0, 2\pi]$, $Y = \sin(X) + \varepsilon$,
  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$,

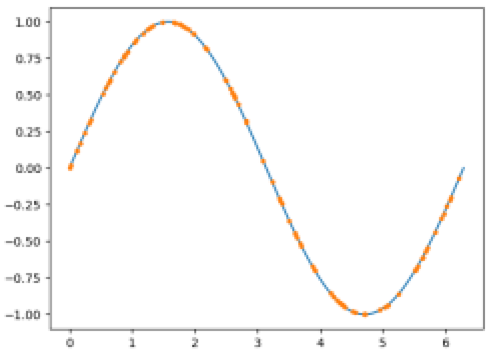


(X_i, Y_i), $i = 1, \ldots, 100$, indep. copies of (X, Y)

A couple of examples with simulations – here the fixed relationship is known, as we define it

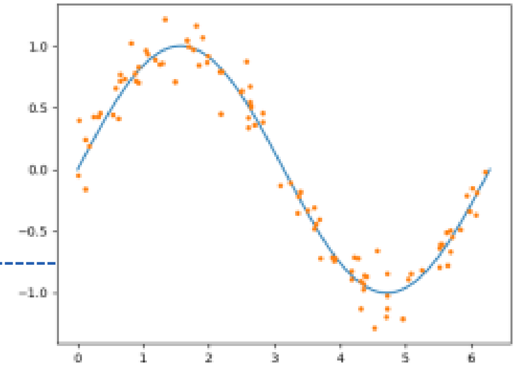
- Deterministic dependence: $Y = f(X)$

  e.g. $X \sim \mathrm{Unif}(0, 2\pi)$, $Y = \sin(X)$
  $(X_i, Y_i), i = 1, ..., 100,$
  indep. copies of (X, Y)



- Homoscedastic additive noise: $Y = f(X) + \epsilon$ for ε independent of X with $\mathbb{E}[\varepsilon] = 0$,

  e.g. $X \sim \mathrm{Unif}[0, 2\pi]$, $Y = \sin(X) + \varepsilon$,
  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$,



(X_i, Y_i),
$i = 1, \ldots, 100$,
indep. copies
of (X, Y)

Remark: Even if, through a blend of semi-omniscience, sheer luck, or guidance from a benevolent oracle, we became aware that the function sin(·) encapsulates the systematic relationship between *X* and *Y*, our ability to precisely predict *Y* would still be hindered by the presence of error.

The problem

Can we find a model that “approximates” the **true model**?

$$? \approx f$$

Supervised Learning
The general strategy

1	$f_{\boldsymbol{\theta}} \in \mathcal{H}, \text{ where } f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$	A family of (appropriate) functions, called "hypothesis class", allowing us to approximate the output r.v. with the input r.v.'s, up to the error term. These functions are the "**machine learning models**".

Supervised Learning
The general strategy

5 $\mathcal{X} = \mathbb{R}^d$

4 $\mathcal{Y} \subseteq \mathbb{R}$

1 $f_{\boldsymbol{\theta}} \in \mathcal{H}, \text{ where } f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$

A family of (appropriate) functions, called “hypothesis class”, allowing us to approximate the output r.v. with the input r.v.’s, up to the error term. These functions are the “**machine learning models**”.

2 Hypothesis class, e.g., all linear regressions*

3 The $\boldsymbol{\theta}$ are parameters – they are our “degrees of freedom” in modeling

1 Linear regressions in *d* variables and *d+1* parameters (also called “weights”)

$$f_{\boldsymbol{\theta}}(x) = \boldsymbol{\theta}^t x = \theta_0 + \theta_1 x_1 + \ldots \theta_d x_d, \ \boldsymbol{\theta}^t = (\theta_0, \ldots, \theta_d) \in \mathbb{R}^{d+1}$$

*Other examples of hypothesis classes: polynomial functions, trees, artificial neural networks etc.

Supervised Learning
The general strategy

1

$$f_{\boldsymbol{\theta}} \in \mathcal{H}, \ \text{where} \ f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$$

A family of (appropriate) functions, called “hypothesis class”, allowing us to approximate the output r.v. with the input r.v.’s, up to the error term. These functions are the “**machine learning models**”.

2

$$L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{+}$$

A loss function, i.e., a quantification of the error we make by approximating the output r.v. with a function of the input r.v.’s

Supervised Learning
The general strategy

1

$$f_{\boldsymbol{\theta}} \in \mathcal{H}, \ \text{where} \ f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$$

A family of (appropriate) functions, called “hypothesis class”, allowing us to approximate the output r.v. with the input r.v.’s, up to the error term. These functions are the “**machine learning models**”.

2

$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$$

A loss function, i.e., a quantification of the error we make by approximating the output r.v. with a function of the input r.v.’s

$$L(z,w) = (z-w)^2$$ squared error

$$L(z,w) = |z-w|$$ absolute error

etc…

(we will use a few different losses in our lectures!)

Supervised Learning
Putting all pieces together: True risk minimization

1 $f_{\boldsymbol{\theta}} \in \mathcal{H}, \ \text{where} \ f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$

2 $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$

True risk minimization

Find $f_{\boldsymbol{\theta}} \in \mathcal{H}$ that minimizes the **true risk functional**

$$\mathbb{E}[L(f_{\boldsymbol{\theta}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\boldsymbol{\theta}}(x), y) dP(x, y)$$

Supervised Learning
Putting all pieces together: True risk minimization

1

$$f_{\boldsymbol{\theta}} \in \mathcal{H}, \text{ where } f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$$

2

$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$$

1

Idea: we want to minimize the expected loss—called “true risk”—we incur when we approximate the output r.v. with a function of the input r.v.’s that is selected from a chosen hypothesis class. To minimize the true risk, we search among all functions in the chosen hypothesis class. This gives us a model $f_{\boldsymbol{\theta}}$ that satisfies $f_{\boldsymbol{\theta}} \approx f$.

True risk minimization

Find $f_{\boldsymbol{\theta}} \in \mathcal{H}$ that minimizes the **true risk functional**

$$\mathbb{E}[L(f_{\boldsymbol{\theta}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\boldsymbol{\theta}}(x), y)dP(x, y)$$

Joint probability distribution (Y, X)

2

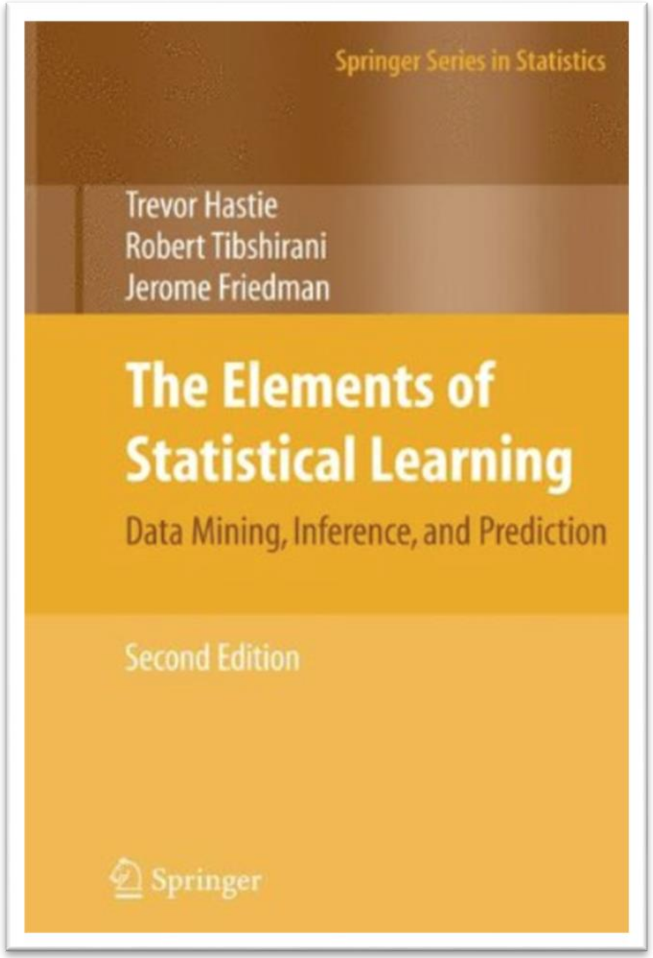
…but this is unknown!

Supervised learning
We change our strategy by introducing empirical data

1 Leverage methods from machine learning (and statistics) **to estimate the true model f using training data** $\{(y_i, x_i)\}_{i=1}^m$ from the input and output r.v.'s.

2 The estimation $\hat{f}$ of f is used to **predict new outputs**:

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



Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: Springer.

Supervised learning
We change our strategy by introducing empirical data

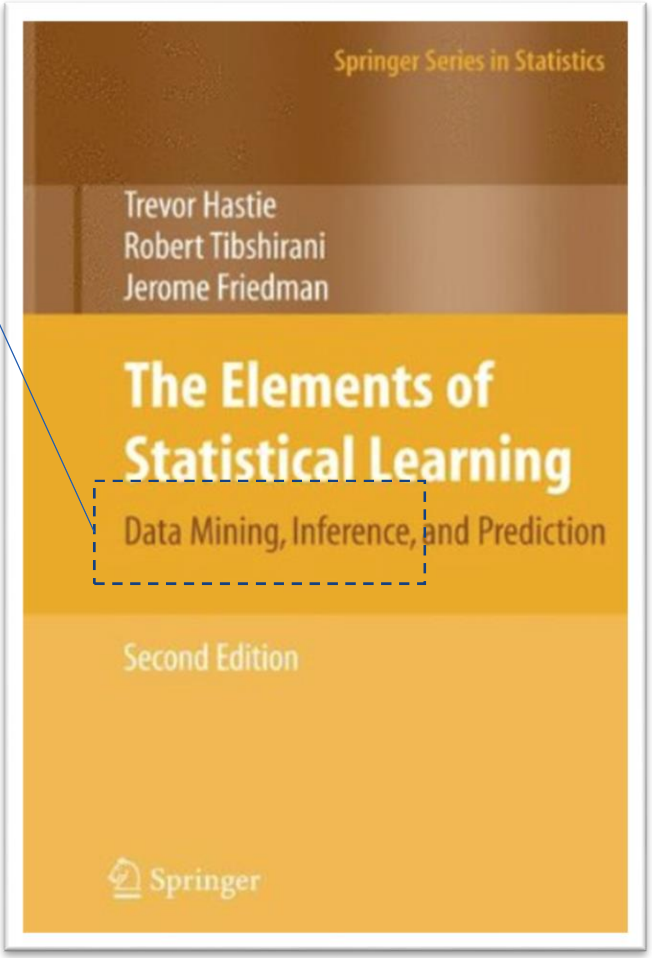
1

What about those?

2

The estimation $\hat{f}$ of f is used to **predict new outputs**:

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



Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: Springer.

Supervised Learning

In applications, we use training data and we try to minimize the empirical risk

Training data

1. $\{(x_i, y_i)\}_{i=1}^{m}$ i.i.d. realizations of (X, Y)

2. $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$

3. $f_{\boldsymbol{\theta}} \in \mathcal{H}$, where $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$

Empirical risk minimization

Find $f_{\boldsymbol{\theta}} \in \mathcal{H}$ that minimizes the **empirical risk functional**

$$E(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} L(f_{\boldsymbol{\theta}}(x_i), y_i)$$

Supervised Learning

In applications, we use training data and we try to minimize the empirical risk

Training data

1	$\{(x_i, y_i)\}_{i=1}^m$ i.i.d. realizations of (X, Y)
2	$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$
3	$f_{\boldsymbol{\theta}} \in \mathcal{H}$, where $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$

1 This is a finite sum that can be computed by a machine, once we collect some training data and we choose a loss function and a hypothesis class.

Empirical risk minimization

Find $f_{\boldsymbol{\theta}} \in \mathcal{H}$ that minimizes the **empirical risk functional**

$$E(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} L(f_{\boldsymbol{\theta}}(x_i), y_i)$$

2 To minimize the empirical risk functional is referred to "**learning**" or "**training**" the machine learning model(s) using the **training data**. There exist algorithms that allow solving this problem and compute the "learned/trained" machine learning model.

Supervised Learning: Summary

Given training data $\{(x_i, y_i)\}_{i=1}^m$, select a family $f_{\boldsymbol{\theta}} \in \mathcal{H}$ of machine learning models and a loss function L.

Then, minimizing the empirical risk functional $E(\boldsymbol{\theta})$, we arrive at the

(learned/trained model) $\hat{f} := f_{\hat{\boldsymbol{\theta}}}$ where $\hat{\boldsymbol{\theta}}$ minimizes $E(\boldsymbol{\theta})$.

1 Customer information from the databases of my company, including an output of interest (buy yes/no, premium, net promoter score…)

2 Family of linear regressions
Family of artificial neural networks
Family of random forests
…

3 Square error
Absolute error
Cross-entropy
Regularization yes/no
…

Supervised Learning: Summa[ry]

Given training data $\{(x_i, y_i)\}_{i=1}^m$, select a family $f_\theta \in \mathcal{H}$ of machine learning models and a loss function L.

Then, minimizing the empirical risk functional $E(\boldsymbol{\theta})$, we arrive at the

4 We need an algorithm to solve the minimization problem…

(learned/trained model) $\hat{f} := f_{\hat{\boldsymbol{\theta}}}$ where $\hat{\boldsymbol{\theta}}$ minimizes $E(\boldsymbol{\theta})$.

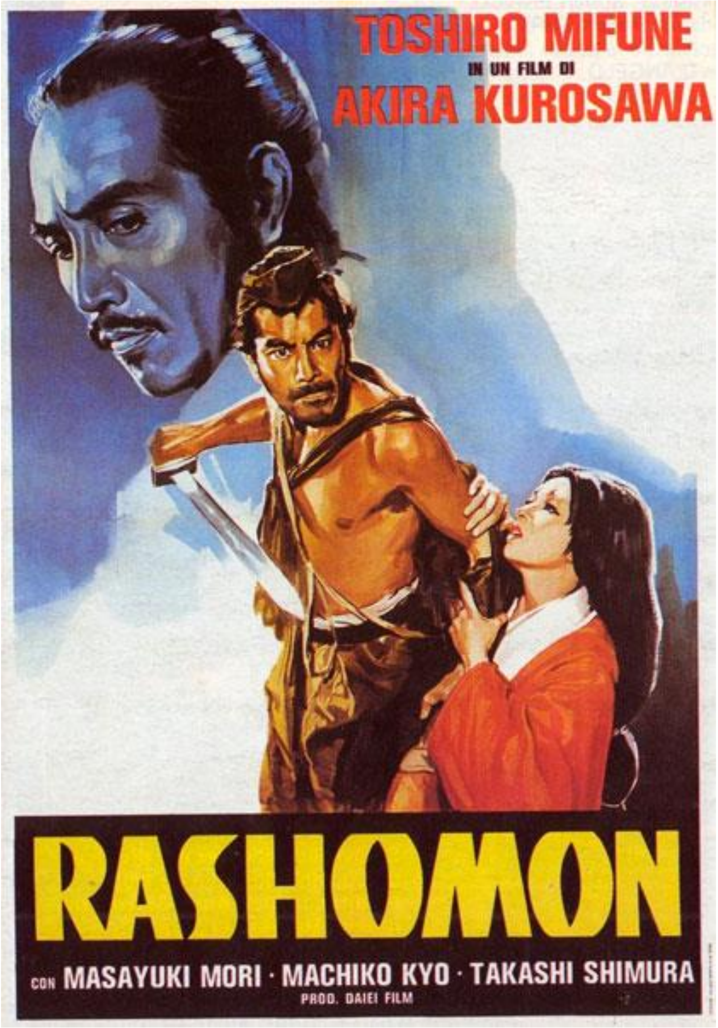
5 "Best" linear regression model
"Best" artificial neural network
"Best" random forest
…

6 The parameters of the "best" model are functions of the training data!

Remark: the “best” model does not need to be unique

This can be a problem for model interpretability, as we will discuss in Block II

“The Rashomon effect”



Rashomon (1950). By A. Kurosawa.

We have an estimated model. And now?

Two key questions

Estimated model

$$\hat{f} := f_{\hat{\boldsymbol{\theta}}} \text{ , with } \hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(x_\bullet, y_\bullet)$$

(training data)

Performance assessment

I. How good is our model on unseen/test data?

Model selection

II. How does model complexity affect model performance on training and unseen/test data?

I. How good is our model on unseen/test data?
Until now, we just considered training data

The empirical risk/mean square error on training data

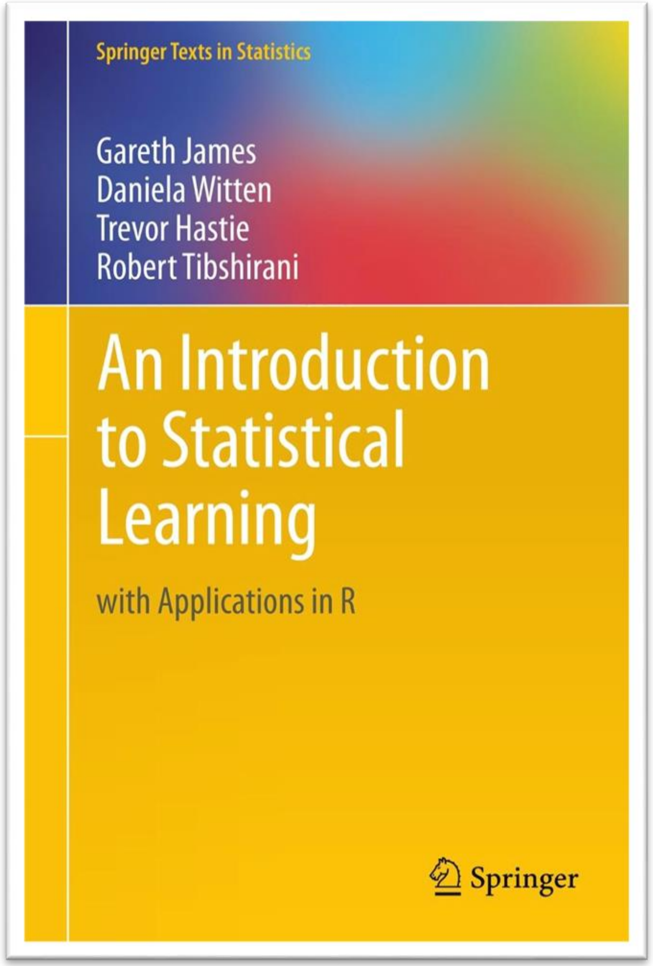
$$\frac{1}{m}\sum_{i=1}^{m} L(f_{\hat{\boldsymbol{\theta}}}(x_i), y_i)$$

By definition, $f_{\hat{\boldsymbol{\theta}}}$ minimizes the empirical risk/mean square error computed on training data $\{(x_i, y_i)\}_{i=1}^{m}$.

Is this enough to do machine learning? Not really…

I. How good is our model on unseen/test data?
Training data are important, but unseen/test data are key

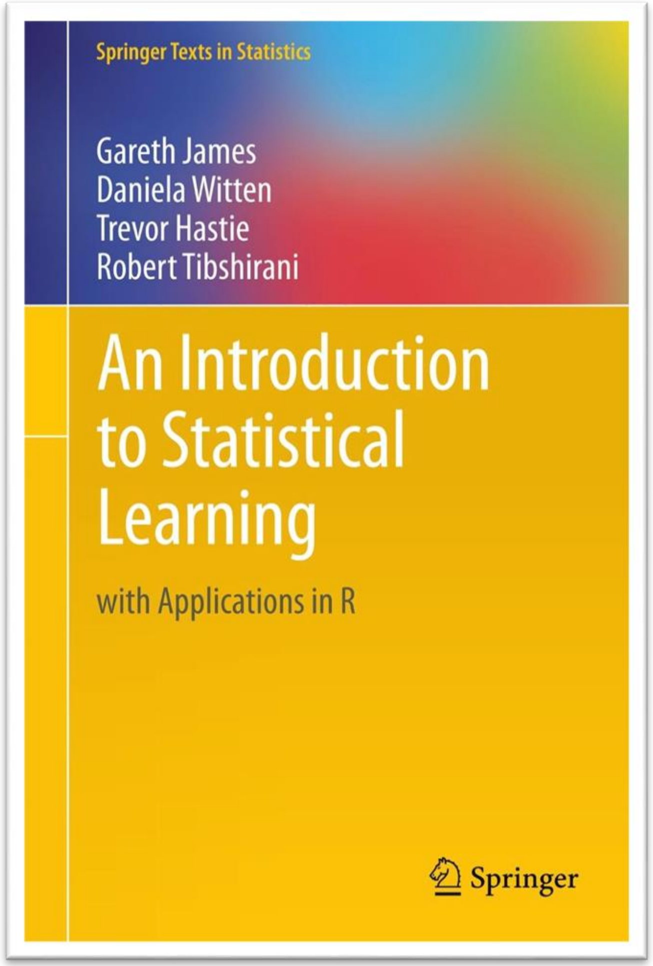
"[…] in general, we do not really care how well the method works on the training data. Rather, *we are interested in the accuracy of the predictions that we obtain when we apply our method* [our trained/learned machine learning model] *to previously unseen test data*" (pag.30, emphasis in original)



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer

I. How good is our model on unseen/test data?
Training data are important, but unseen/test data are key

"Suppose that we are interested in developing an algorithm to predict a stock's price based on previous stock returns. We can train the method using stock returns from the past 6 months. But we don't really care how well our method predicts last week's stock price. We instead care about how well it will predict tomorrow's price or next month's price." (pag.30)



James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: Springer

I. How good is our model on unseen/test data?

An idea: minimizing the empirical risk/mean squared error on unseen/test data

Idea

We would like to train a model $f_{\hat{\boldsymbol{\theta}}}$ that reaches a small empirical risk/mean squared error on unseen/test data:

$$\frac{1}{q}\sum_{j=1}^{q} L(f_{\hat{\boldsymbol{\theta}}}(x_j), y_j) \qquad \{(x_j, y_j)\}_{j=1}^{q} \; i.i.d.$$

(unseen/test data)

I. How good is our model on unseen/test data?

An idea: minimizing the empirical risk/mean squared error on unseen/test data

Idea

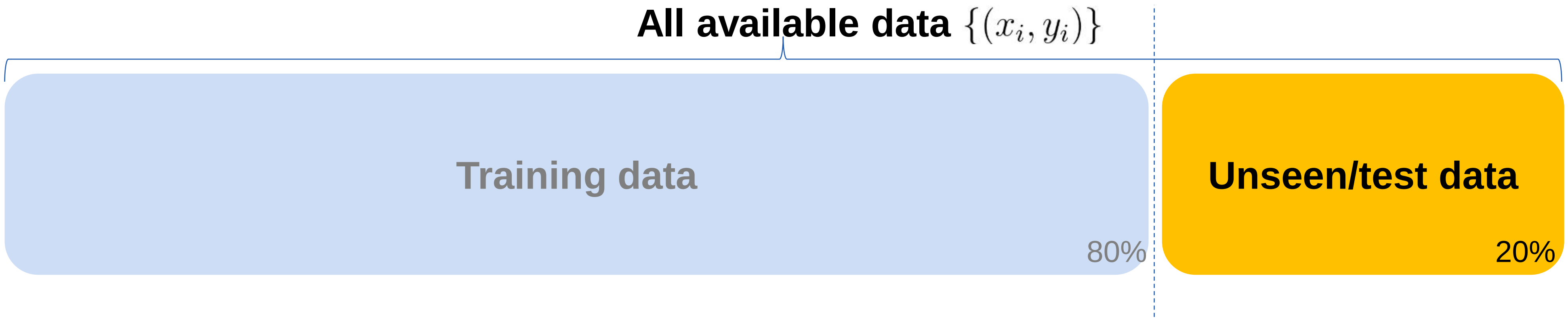
We would like to train a model $f_{\hat{\boldsymbol{\theta}}}$ that reaches a small empirical risk/mean squared error on unseen/test data:

$$\frac{1}{q}\sum_{j=1}^{q} L(f_{\hat{\boldsymbol{\theta}}}(x_j), y_j) \qquad \{(x_j, y_j)\}_{j=1}^{q} \ i.i.d.$$

(unseen/test data)

…unfortunately, we do not have access to the $y_\bullet$ at the time of prediction. We do not know how well the model outputs $f_{\hat{\boldsymbol{\theta}}}(x_\bullet)$ approximate the "true" outputs $y_\bullet$.

I. How good is our model on unseen/test data?
A simple solution

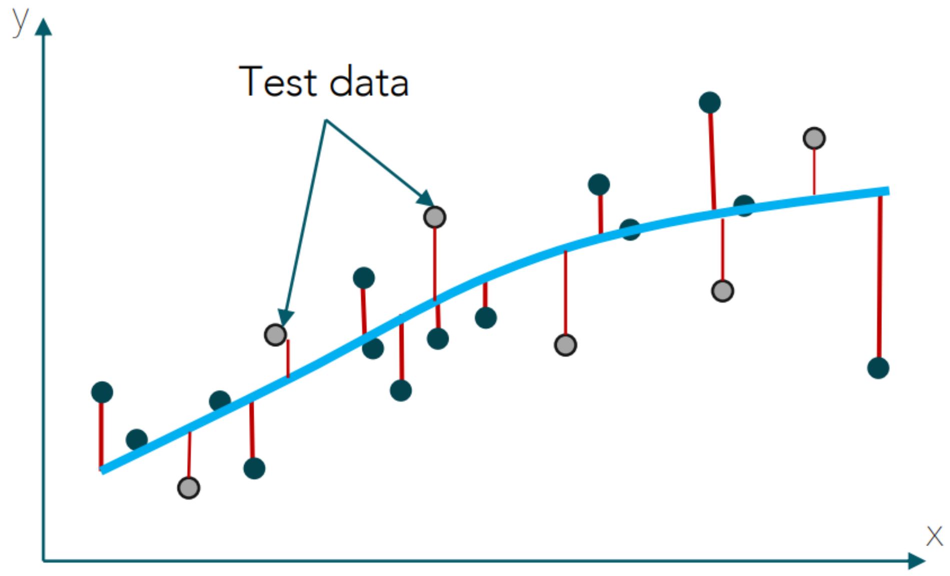


- In its simplest version, our strategy is to *randomly* split available data into (1) training, and (2) unseen/test data. One common split ratio is 80-20%.
- We use only the training data to search for the model $f_{\hat{\theta}}$ that minimizes the empirical risk (on these training data)
- During training, we do not use the unseen/test data

I. How good is our model on unseen/test data?
A simple solution

Training data 80%

Unseen/test data 20%



Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

- **During training** minimize the empirical risk (on training data):

$$\frac{1}{m}\sum_{i=1}^{m} L(f_{\boldsymbol{\theta}}(x_i), y_i)$$

- **After training** compute the empirical risk (on unseen/test data) of the trained model:

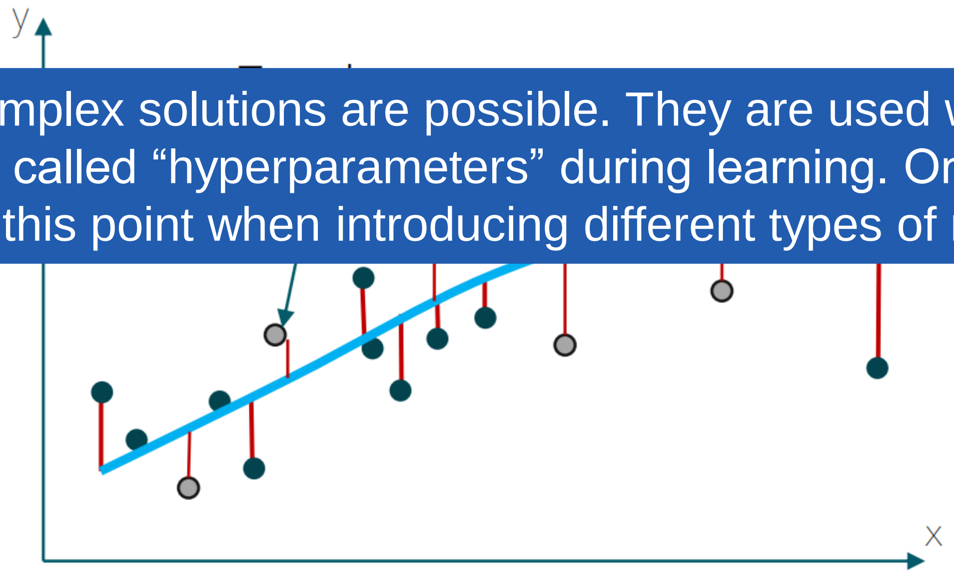
$$\frac{1}{q}\sum_{j=1}^{q} L(f_{\hat{\boldsymbol{\theta}}}(x_j), y_j)$$

I. How good is our model on unseen/test data?
A simple solution

Training data 80%

Unseen/test data 20%

More complex solutions are possible. They are used when we need to estimate additional degrees of freedom, called "hyperparameters" during learning. One famous example is "cross-validation". We will return to this point when introducing different types of machine learning models.



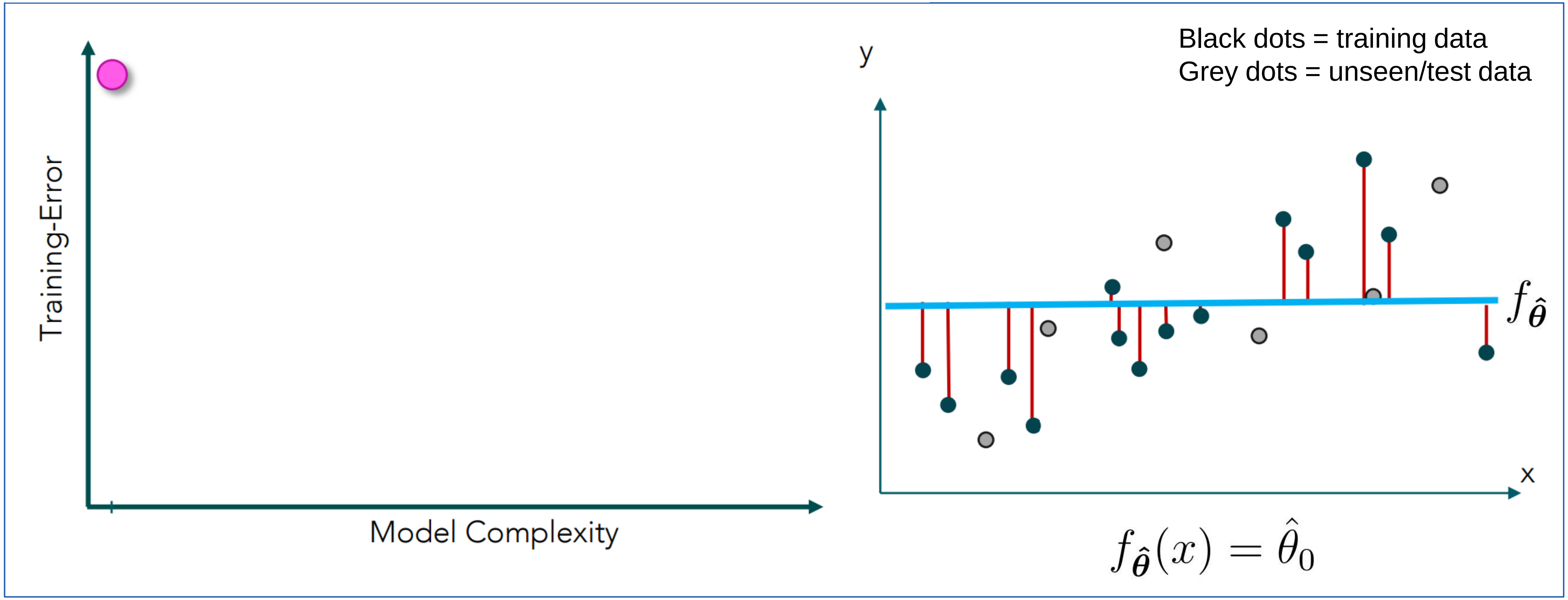
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

- **During training** minimize the empirical risk (on training data):
- **After training** compute the empirical risk (on unseen/test data) of the trained model:

$$\frac{1}{q}\sum_{j=1}^{q} L(f_{\hat{\boldsymbol{\theta}}}(x_j), y_j)$$

II. How does model complexity affect model performance on training and unseen/test data?

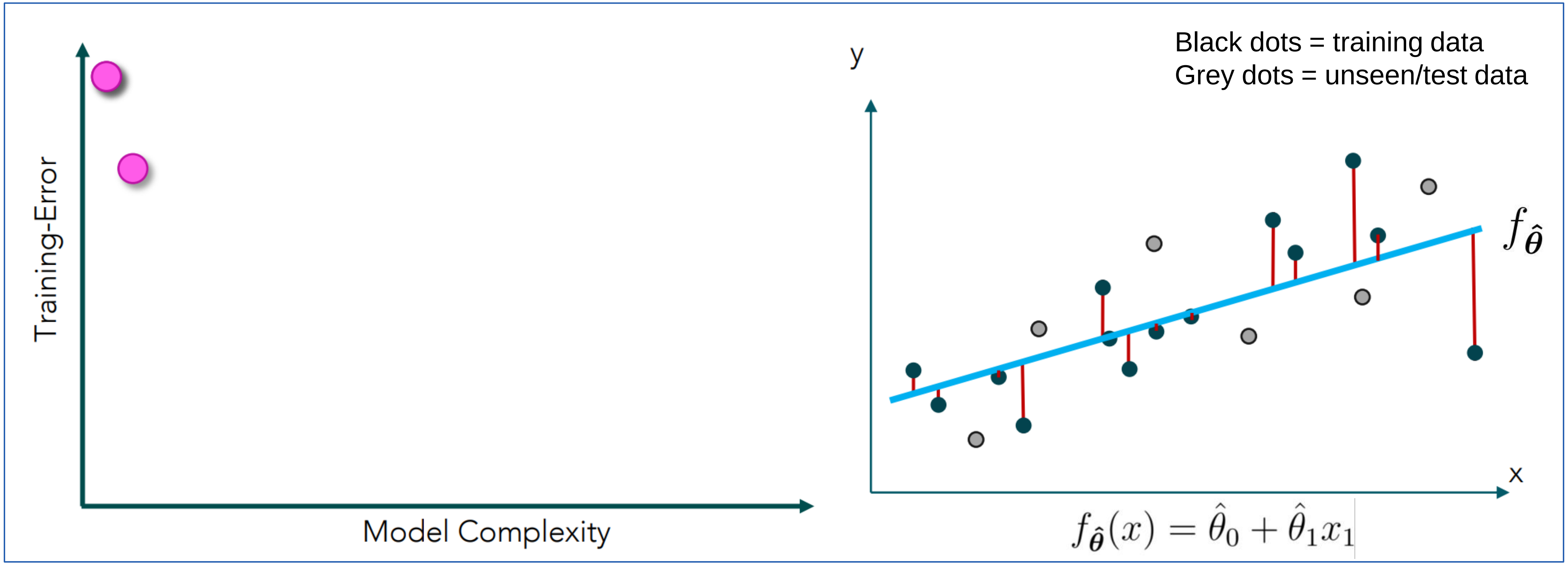
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A constant model reaches a quite high training error

II. How does model complexity affect model performance on training and unseen/test data?

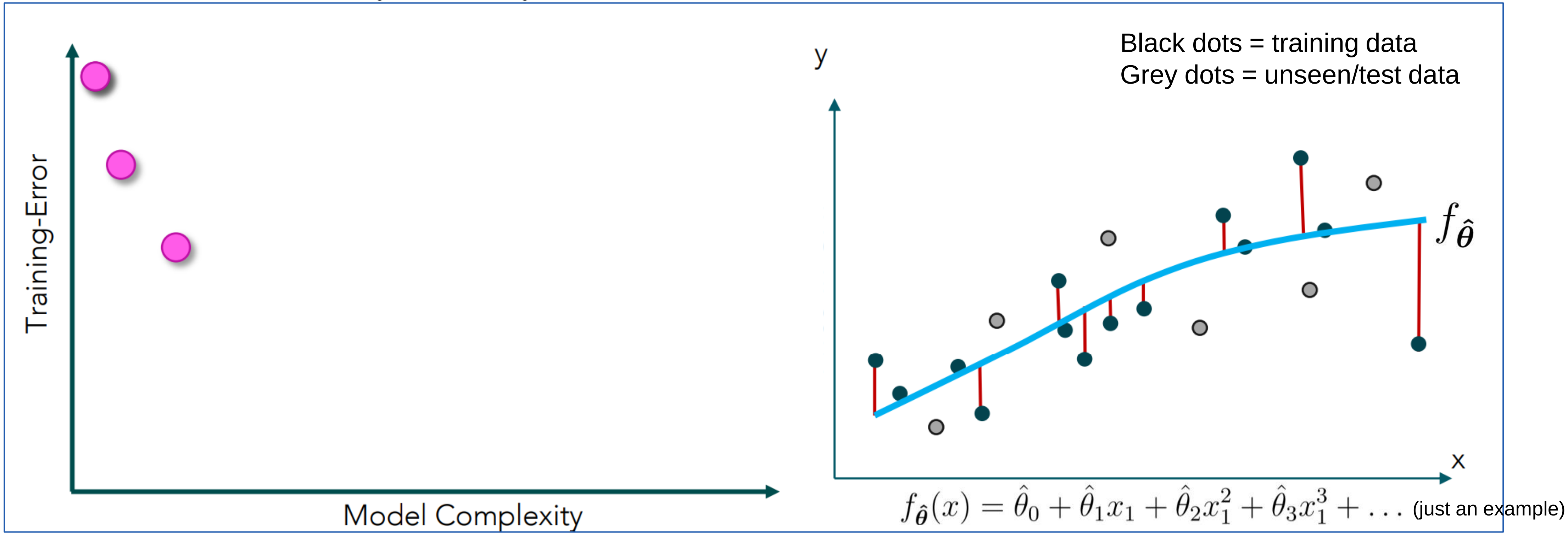
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A linear model starts approximating training data

II. How does model complexity affect model performance on training and unseen/test data?

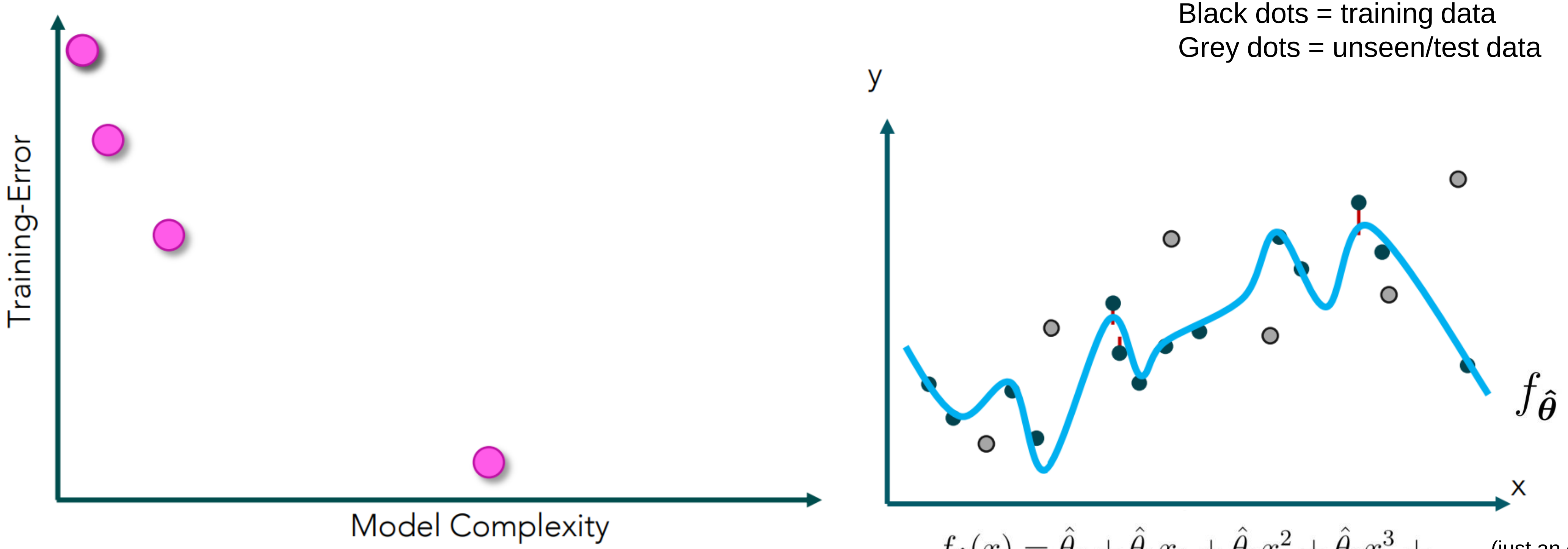
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A moderately nonlinear model improves this approximation (on training data)

II. How does model complexity affect model performance on training and unseen/test data?

Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

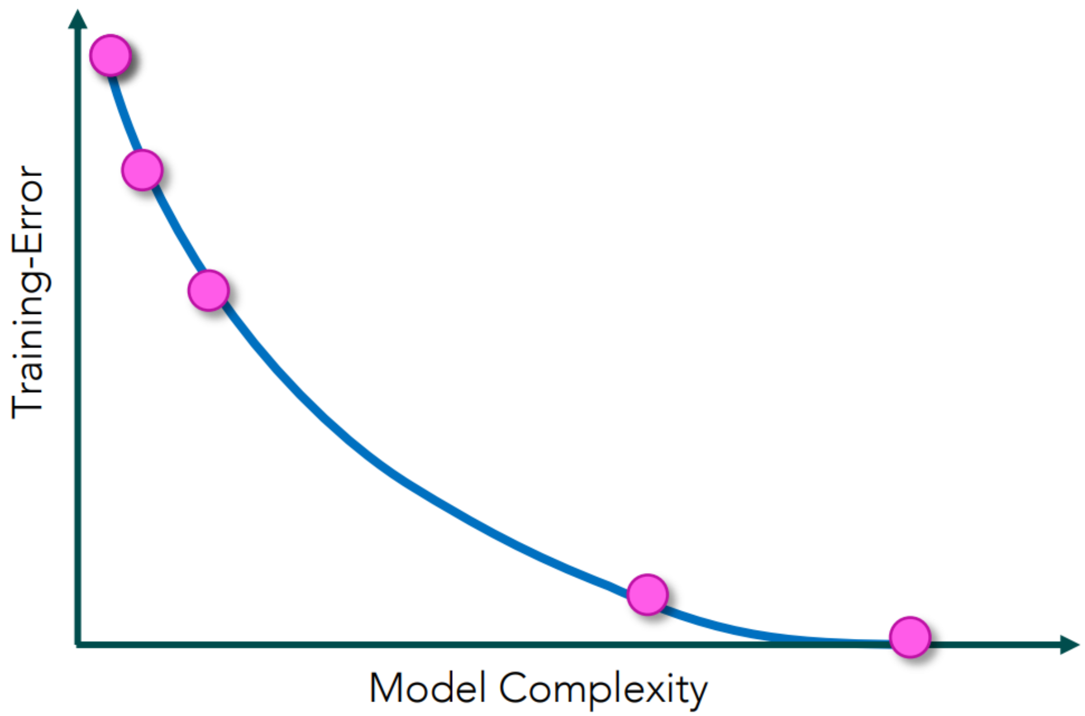


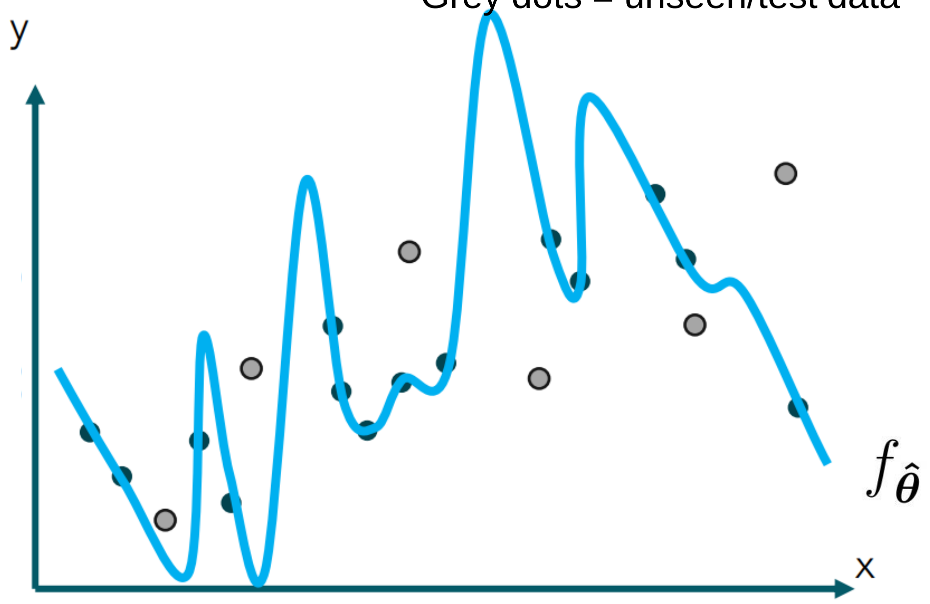
$$f_{\hat{\theta}}(x) = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_1^2 + \hat{\theta}_3 x_1^3 + \ldots$$ (just an example)

Increasing model complexity further improves this approximation (on training data)

II. How does model complexity affect model performance on training and unseen/test data?

Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause





$$f_{\hat{\theta}}(x) = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_1^2 + \hat{\theta}_3 x_1^3 + \dots$$ (just an example)

A highly complex model perfectly fits training data

We need to introduce the generalization error

The generalization error is a measure of the loss we incur when we approximate the output r.v. with the trained model on all possible data

Generalization error	$\mathbb{E}[L(f_{\hat{\boldsymbol{\theta}}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\hat{\boldsymbol{\theta}}}(x), y) dP(x, y)$

We need to introduce the generalization error

The generalization error is a measure of the loss we incur when we approximate the output r.v. with the trained model on all possible data

Generalization error

$$\mathbb{E}[L(f_{\hat{\theta}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\hat{\theta}}(x), y)\,dP(x, y)$$

1 Hopefully, it is small

2 Learned/trained model

3 All data

4 But this is unknown…

We need to introduce the generalization error

The generalization error is a measure of the loss we incur when we approximate the output r.v. with the trained model on all possible data

Generalization error	$$\mathbb{E}[L(f_{\hat{\boldsymbol{\theta}}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\hat{\boldsymbol{\theta}}}(x), y)dP(x, y)$$
Decomposition of the generalization error	Given $Y = f(X) + \epsilon$, where ϵ indep. of X, $\mathbb{E}[\epsilon] = 0$, $\mathbb{E}[\epsilon^2] = \sigma^2$ and the square loss, then: $$\mathbb{E}[L(f_{\hat{\boldsymbol{\theta}}}(X), Y)] = \mathbb{E}[(f_{\hat{\boldsymbol{\theta}}}(X) - f(X))^2] + \sigma^2$$

We need to introduce the generalization error

The generalization error is a measure of the loss we incur when we approximate the output r.v. with the trained model on all possible data

Generalization error

$$\mathbb{E}[L(f_{\hat{\boldsymbol{\theta}}}(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f_{\hat{\boldsymbol{\theta}}}(x), y)dP(x, y)$$

Decomposition of the generalization error

Given $Y = f(X) + \epsilon$, where ϵ indep. of X, $\mathbb{E}[\epsilon],= 0$
$\mathbb{E}[\epsilon^2] = \sigma^2$

and the square loss, then:

$$\mathbb{E}[L(f_{\hat{\boldsymbol{\theta}}}(X), Y)] = \underbrace{\mathbb{E}[(f_{\hat{\boldsymbol{\theta}}}(X) - f(X))^2]}_{1} + \sigma^2$$

1 Average error the estimated model does w.r.t. the “true model”

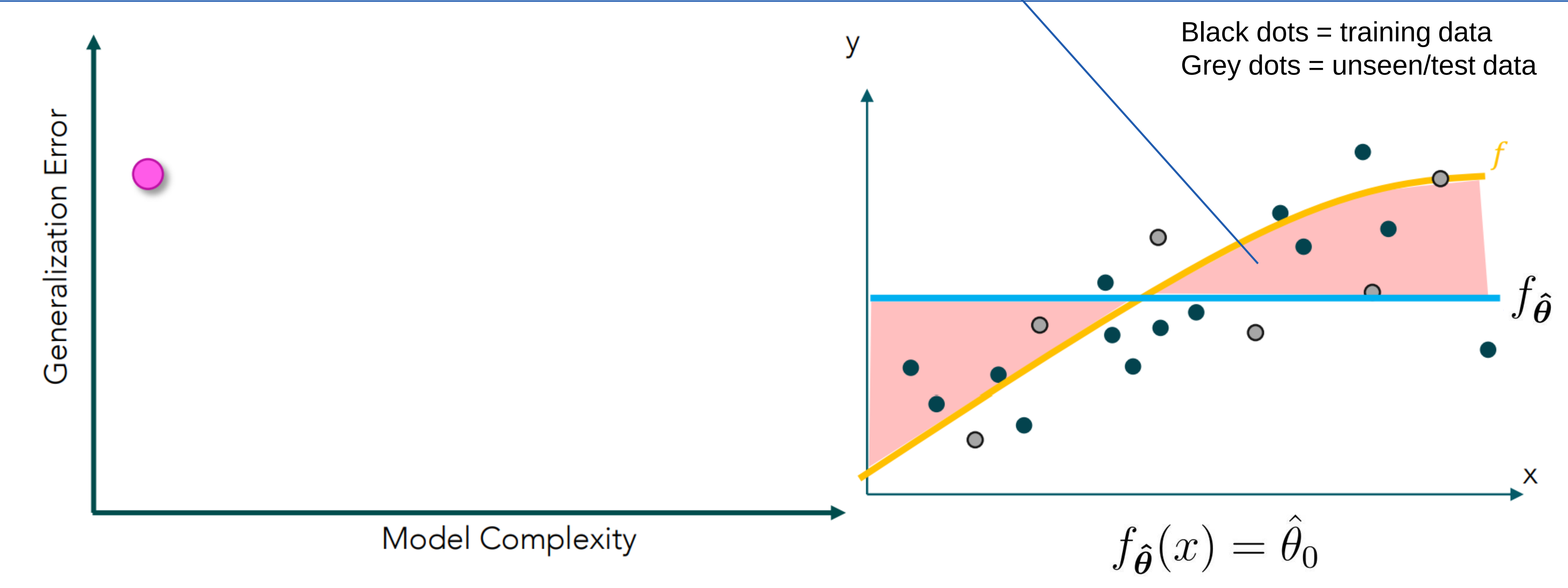
2 We approximate it with the empirical risk on unseen/test data

3 Irreducible component due to error

II. How does model complexity affect model performance on training and unseen/test data?

The bigger the shaded area, the bigger the generalization error

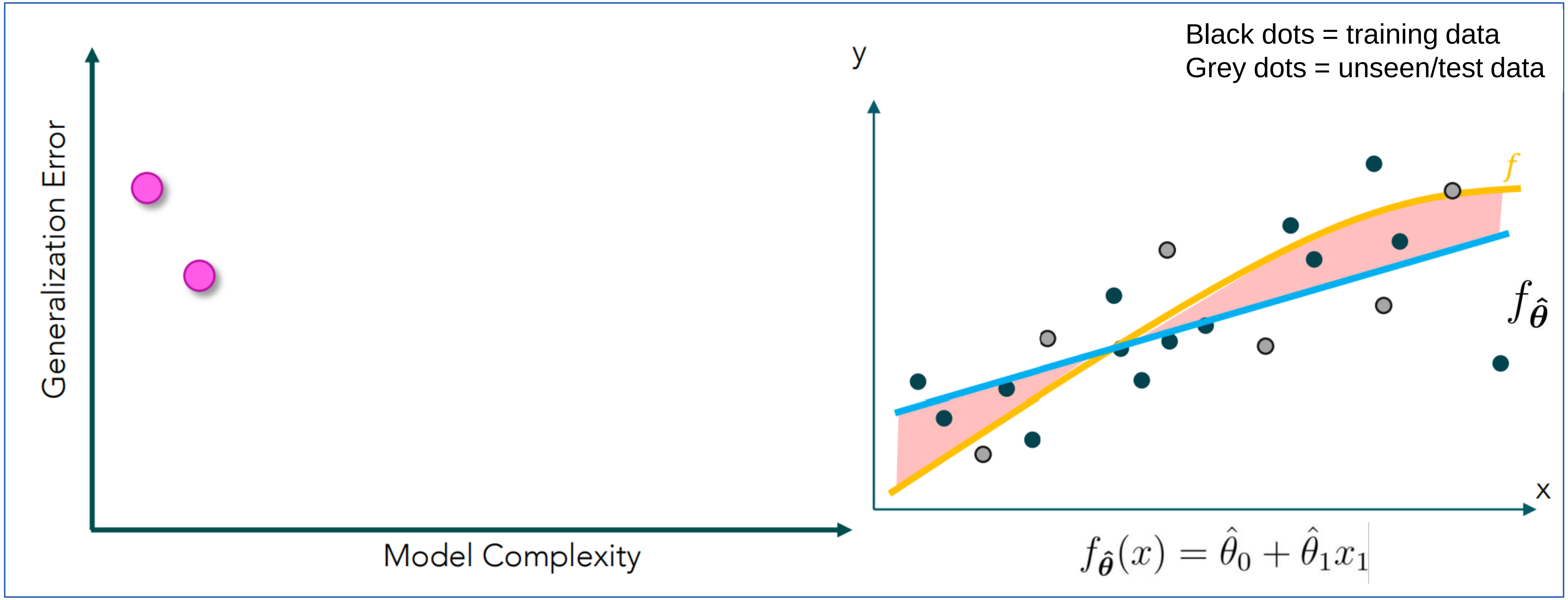
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A constant model reaches a quite high generalization error

II. How does model complexity affect model performance on training and unseen/test data?

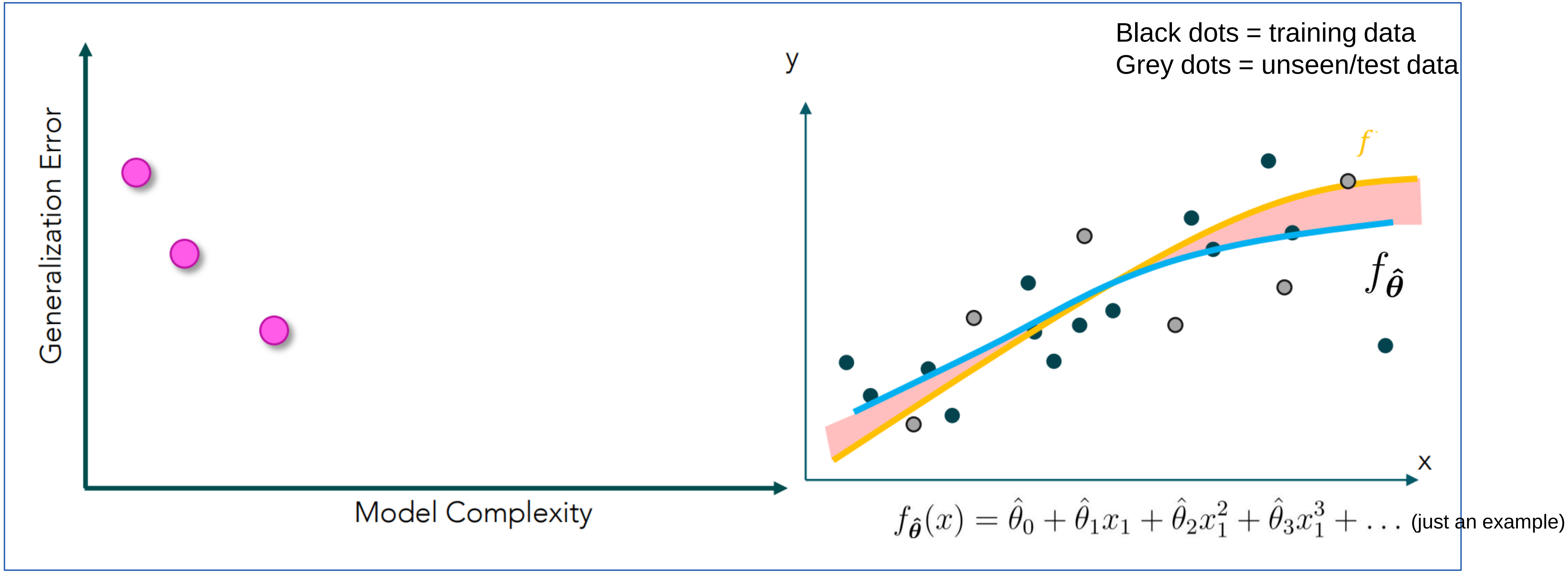
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A linear model reduces the generalization error

II. How does model complexity affect model performance on training and unseen/test data?

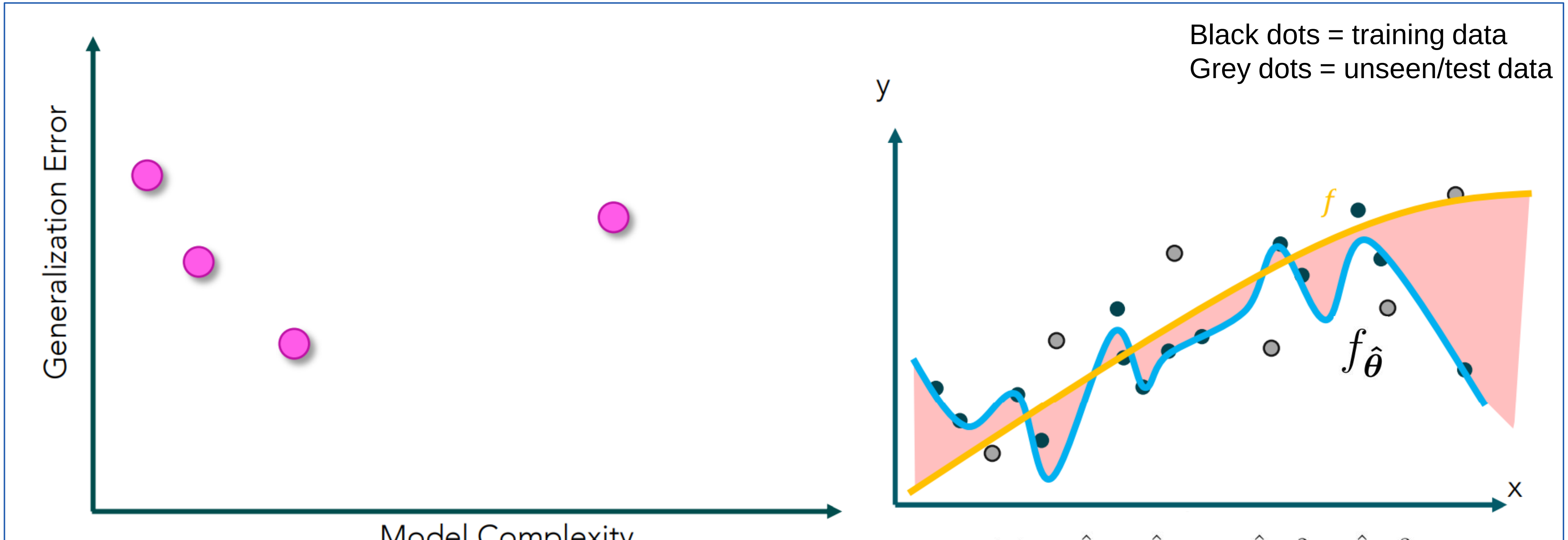
Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



A moderately nonlinear model further reduces the generalization error

II. How does model complexity affect model performance on training and unseen/test data?

Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

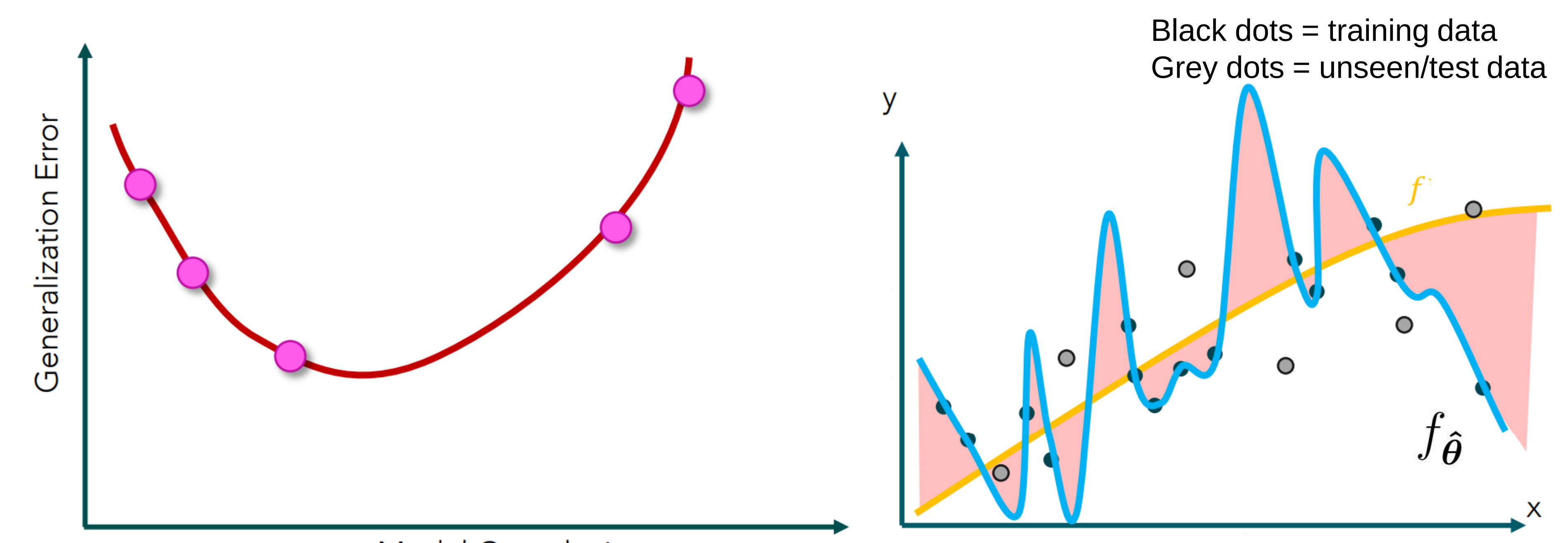


$$f_{\hat{\boldsymbol{\theta}}}(x) = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_1^2 + \hat{\theta}_3 x_1^3 + \ldots$$ (just an example)

A highly nonlinear model reaches a high generalization error

II. How does model complexity affect model performance on training and unseen/test data?

Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

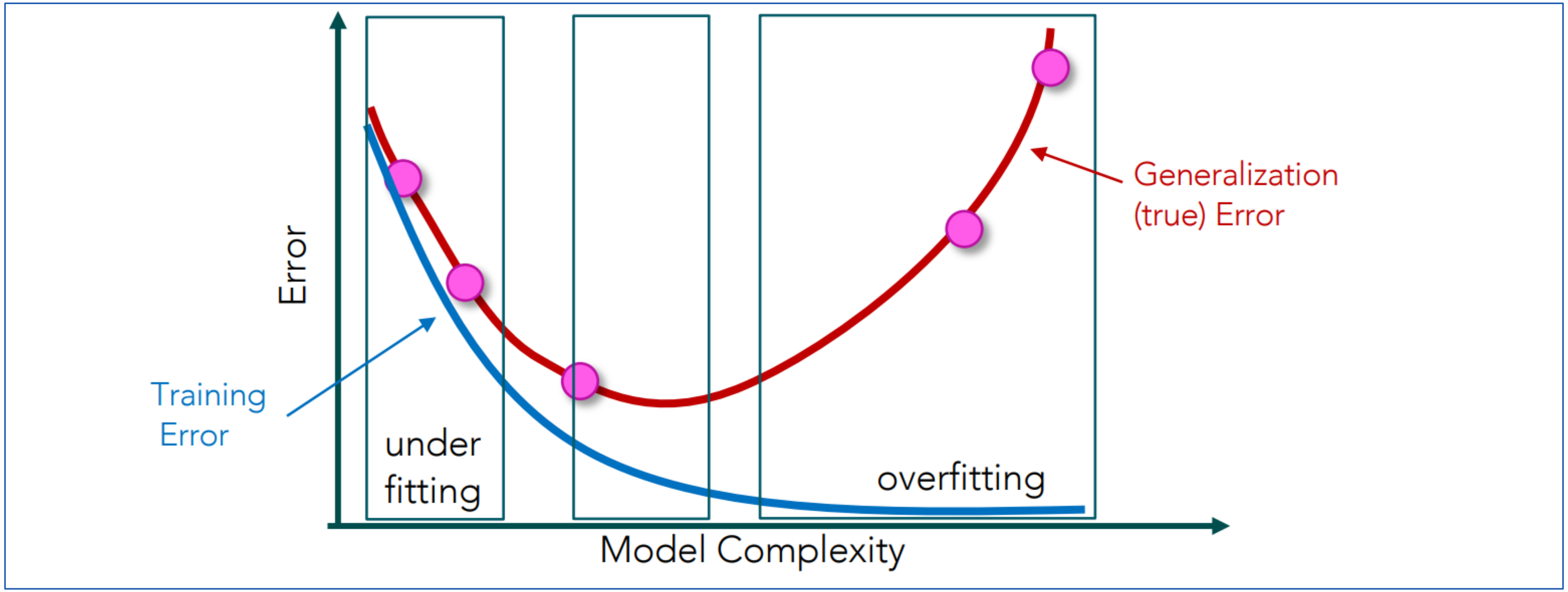


$$f_{\hat{\theta}}(x) = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \hat{\theta}_2 x_1^2 + \hat{\theta}_3 x_1^3 + \ldots$$ (just an example)

The highly nonlinear model perfectly fitting training data has a very large generalization error

Summary: underfitting vs. overfitting machine learning models

Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause



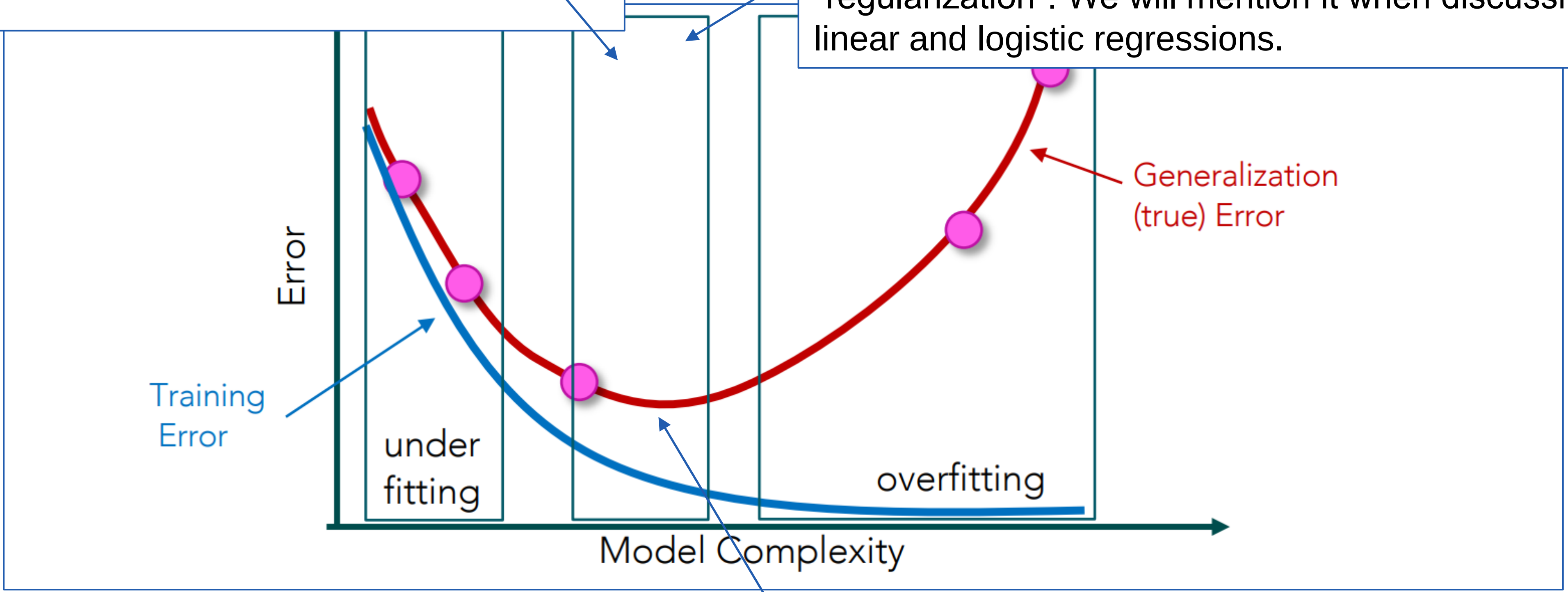
Summary: underfitting vs. overfitting machine learning models

3

However, also regularization needs some fine tuning of what we call “hyper-parameters”. This can be achieved using sampling techniques, such as cross-validation

2

There exists a method to control for the model complexity and avoid under/overfitting: “regularization”. We will mention it when discussing linear and logistic regressions.



Source: Introduction to Machine Learning – Prof. F. Yang, Prof. A. Krause

GOAL

1

We would like to train models that avoid underfitting (too “simple”) and overfitting (too “complex”)

Model selection and performance assessment in machine learning: Key takeaways

I	**Training vs. Generalization Error**: Training error measures model performance on the training dataset, while generalization error assesses its performance on unseen data. The goal is to minimize both, ensuring the model learns from the data without overfitting or underfitting.
II	**Complexity vs. Training vs. Test Errors:** As model complexity increases, training error typically decreases, but test error first decreases then increases due to overfitting. The challenge is finding the right balance where test error is minimized, indicating optimal model complexity.
III	**Train vs. Test Split:** Dividing the dataset into a training set and a test set is essential for evaluating model performance. The training set is used to train the model, while the test set assesses its generalization to new data, helping to mitigate overfitting.
IV	**Beyond Train vs. Test Split:** Other methods improve the estimation of model performance based on train vs. test split. We will discuss them (train vs. validation vs. test) in the forthcoming lectures.

Self-Study

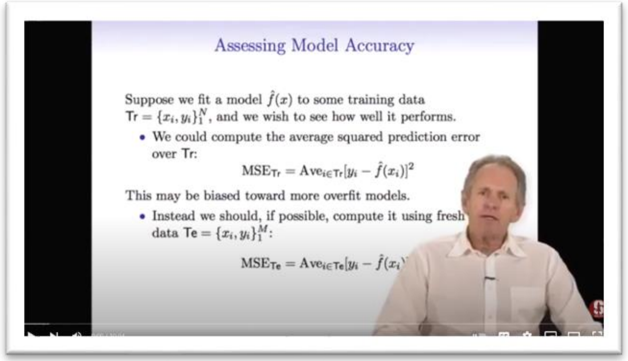
Self-study: Closing with supervised learning

Topics for self-study

From Moodle download the book:

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer.

- **Section 2.1** What is Statistical Learning? (Especially 2.2.1-2.1.3 and 2.2.1-2.2.2)



Extra: watch the short video (with T. Hastie!):
https://www.youtube.com/watch?v=pvcEQfcO3pk&t=8s

Machine Learning Methods (Part 1)

- Linear regression

Linear regression

Linear regression: ideas

I	Simplest model to predict numerical outputs (distances, temperatures, prices, time durations…). It assumes a linear relationship between a scalar output/outcome and the dependent variables, up to random contributions that we call “noise”
II	Although the linear assumption is often an oversimplification, it is rather useful to learn the basics of machine learning modelling
III	Linear models are also used in statistics. Here, we do machine learning: our main goal is to compute numerical predictions on unseen test data as accurately as possible

Linear regression
Notation

Model	$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d + \varepsilon$ for an $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ independent of $(X_1, \ldots, X_d)$	We model a numerical output r.v. with a linear comb. of *d+1* r.v.

Linear regression
Notation

Model

$$Y = \underbrace{\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d}_{\text{Linear combination of } d+1 \text{ random variables.}} + \varepsilon \text{ for an } \underbrace{\varepsilon \sim \mathcal{N}(0, \sigma^2) \text{ independent of } (X_1, \ldots, X_d)}_{\text{these assumptions allow to determine statistical properties of the estimators}}$$

We model a numerical output r.v. with a linear comb. of *d+1* r.v.

Linear combination of *d+1* random variables.

$\beta_\bullet$ are *d+1* coefficients

these assumptions allow to determine statistical properties of the estimators

output random variable

error term (also a random variable)

Linear regression
Notation

Model	$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d + \varepsilon$ for an $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ independent of $(X_1, \ldots, X_d)$	We model a numerical output r.v. with a linear comb. of *d+1* r.v.
Empirical setting	Let $(x_{i1}, \ldots, x_{id}, y_i, \varepsilon_i)$, $i = 1, \ldots, m$, be iid realizations of $(X_1, \ldots, X_d, Y, \varepsilon)$ Matrix notation: $y = A\beta + \boldsymbol{\varepsilon}$, where $y = \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \end{pmatrix}, \; A = \underbrace{\begin{pmatrix} 1 & x_{11} & \ldots & x_{1d} \\ 1 & x_{21} & \ldots & x_{2d} \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_{m1} & \ldots & x_{md} \end{pmatrix}}_{\text{design matrix}}, \; \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \vdots \\ \beta_d \end{pmatrix}, \; \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \vdots \\ \varepsilon_m \end{pmatrix}$ typically, $m \geq d + 1$ (more observations than parameters)	We can use the matrix notation to represent the (empirical) linear model in a compact form

Linear regression
Notation

Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d + \varepsilon \text{ for an } \varepsilon \sim \mathcal{N}(0, \sigma^2) \text{ independent of } (X_1, \ldots, X_d)$$

We model a numerical output r.v. with a linear comb. of *d+1* r.v.

Empirical setting

Let $(x_{i1}, \ldots, x_{id}, y_i, \varepsilon_i)$, $i = 1, \ldots, m$, be iid realizations of $(X_1, \ldots, X_d, Y, \varepsilon)$

Matrix notation: $y = A\beta + \boldsymbol{\varepsilon}$, where

$$y = \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \end{pmatrix}, \quad A = \underbrace{\begin{pmatrix} 1 & x_{11} & \ldots & x_{1d} \\ 1 & x_{21} & \ldots & x_{2d} \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_{m1} & \ldots & x_{md} \end{pmatrix}}_{\text{design matrix}}, \quad \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \vdots \\ \beta_d \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \vdots \\ \varepsilon_m \end{pmatrix}$$

typically, $m \geq d + 1$ (more observations than parameters)

We can use the matrix notation to represent the (empirical) linear model in a compact form

Data! Realizations of the output r.v.

The design matrix depends on data! Realizations of the *d+1* r.v.

These coefficients are unknown…we need to estimate them

Again, these are data

Linear regression
Learning the model with Ordinary Least Squares (OLS)

OLS

Empirical loss minimization with the square loss function = ordinary least squares:

$$\min_{b \in \mathbb{R}^{d+1}} \|A\,b - y\|_2^2 = \min_{b \in \mathbb{R}^{d+1}} \sum_{i=1}^{m} \left(\sum_{j=0}^{d} x_{ij} b_j - y_i \right)^2 \quad \text{where } x_{i0} = 1, \quad i = 1, \ldots, m$$

Lemma

Since $\|A\,b - y\|_2^2$ is convex in $b \in \mathbb{R}^{d+1}$, b is a minimizer $\Leftrightarrow$ $\nabla_b \|A\,b - y\|_2^2 = 0$

$$\Leftrightarrow \quad 0 = \frac{\partial}{\partial b_k} \sum_{i=1}^{m} \left(\sum_{j=0}^{d} x_{ij} b_j - y_i \right)^2 = \sum_{i=1}^{m} 2 \left(\sum_{j=0}^{d} x_{ij} b_j - y_i \right) x_{ik} \quad \text{for all } k = 0, 1, \ldots, d$$

$$\Leftrightarrow \quad A^T A\,b = A^T y \quad \text{(normal equation)}$$

If the columns of A are linearly independent, $A^T A$ is regular, and the unique solution of $A^T A\,b = A^T y$ is

$$\hat{\beta} = (A^T A)^{-1} A^T y$$

depends on the data
$x_{ij}, y_i, \quad i = 1, \ldots, m,$
$j = 0, \ldots, d$

(Python computes these parameters for us)

We model a numerical output r.v. with a linear comb. of *d+1* r.v.

*the columns of the design matrix must be linearly independent. Otherwise, the problem admits multiple solutions

Linear regression

Against overfitting: lasso, ridge and elastic net

$$\min_{b \in \mathbb{R}^{d+1}} \|A\, b - y\|_2^2 + \lambda\, r(b)$$

where

$$
\begin{cases}
r(b) = 0 \\
r(b) = \|b\|_1, \ \ \|b\|_1 = \sum_{k=1}^{d+1} |b_k| \\
r(b) = \|b\|_2^2, \ \ \|b\|_2^2 = \sum_{k=1}^{d+1} b_k^2 \\
r(b) = \rho \|b\|^1 + \frac{1-\rho}{2} \|b\|_2^2
\end{cases}
$$

`sklearn.linear_model`.LinearRegression

LASSO (Least absolute shrinkage and selection operator)

Ridge

Elastic Net

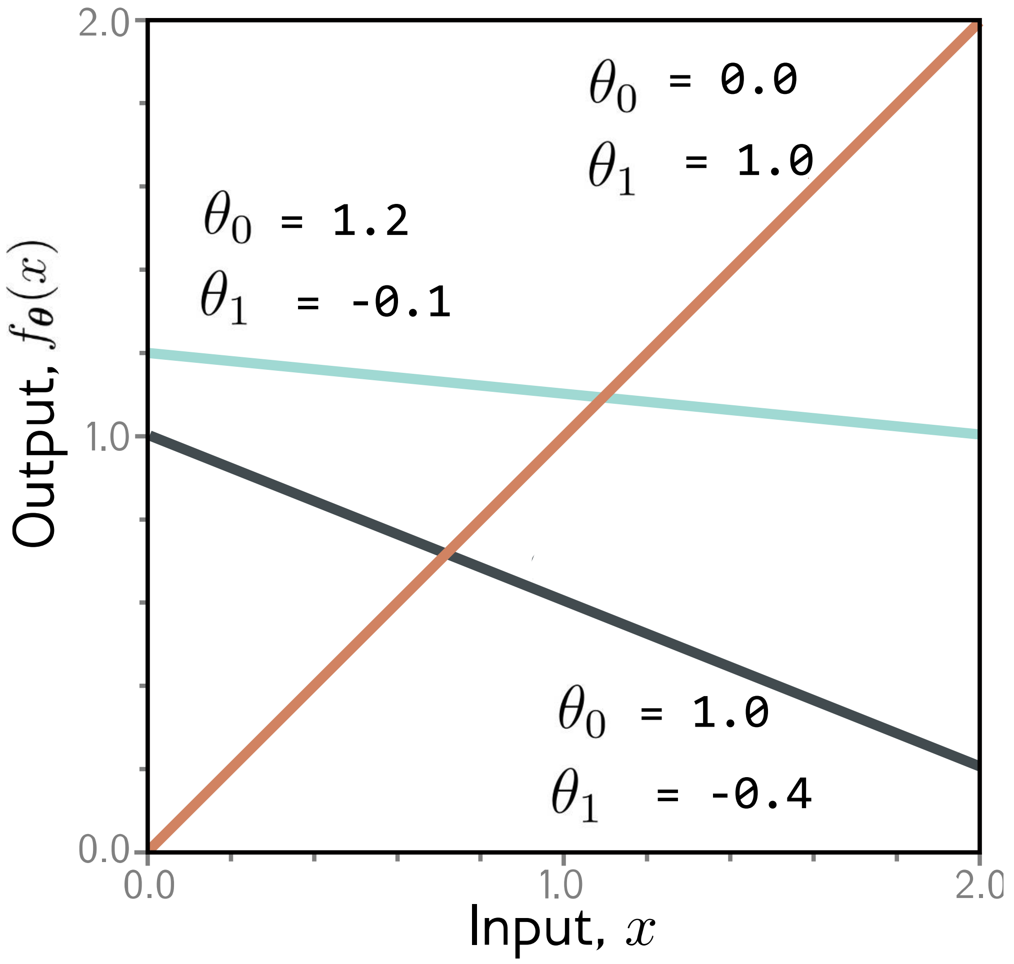
Let us train a machine learning model

Toy example: linear regression in one variable

01/11

$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x$$

intercept slope



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model

Toy example: linear regression in one variable

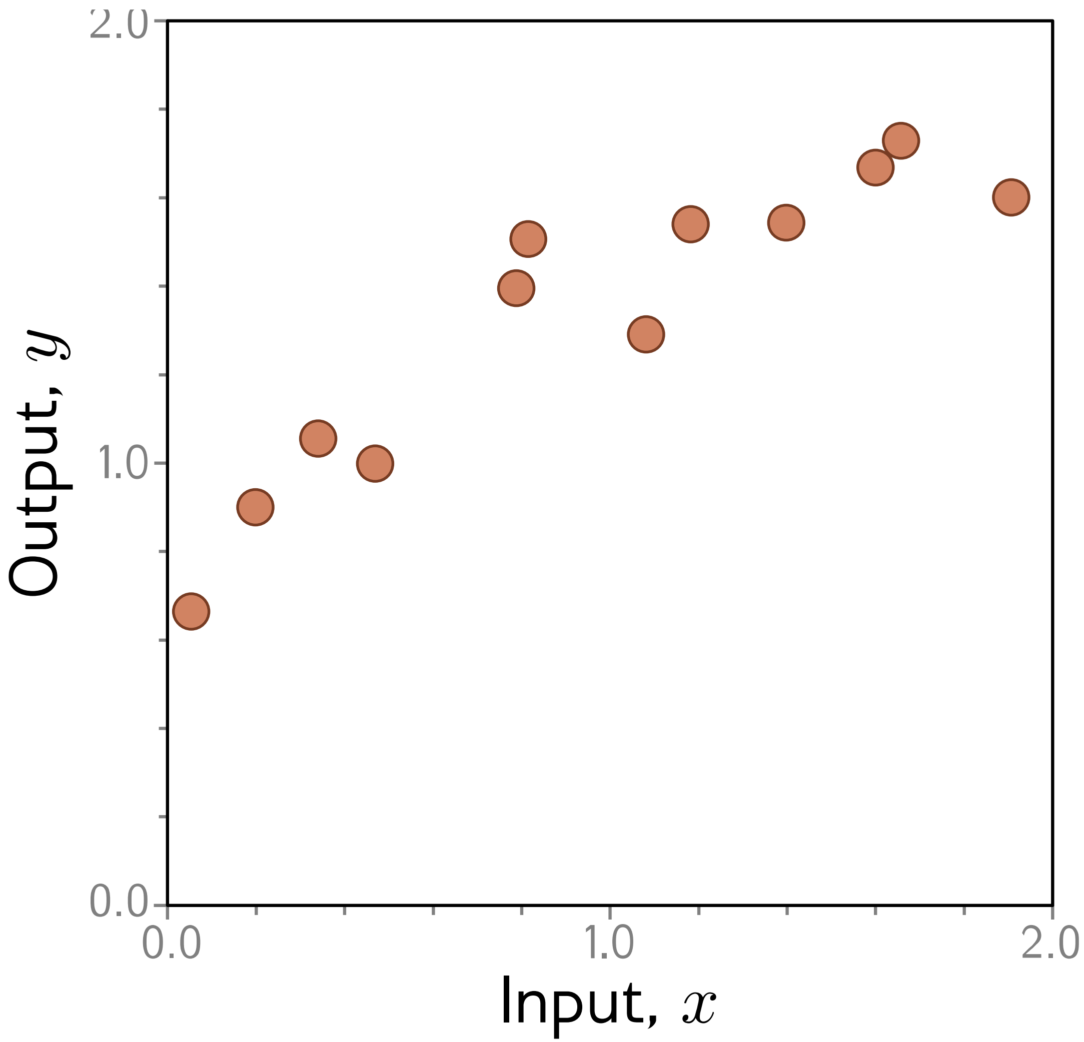
Let us simulate N samples $\{x_i, y_i\}$ (N=12)

We choose to minimize the empirical risk:

$$E(\boldsymbol{\theta}) = \frac{1}{12}\sum_{i=1}^{12} L(f_{\boldsymbol{\theta}}(x_i), y_i) =$$

$$= \frac{1}{12}\sum_{i=1}^{12} (\underbrace{\theta_0 + \theta_1 x_i}_{\text{approximation of the true value } y_i} - y_i)^2$$

true value



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model

Toy example: linear regression in one variable

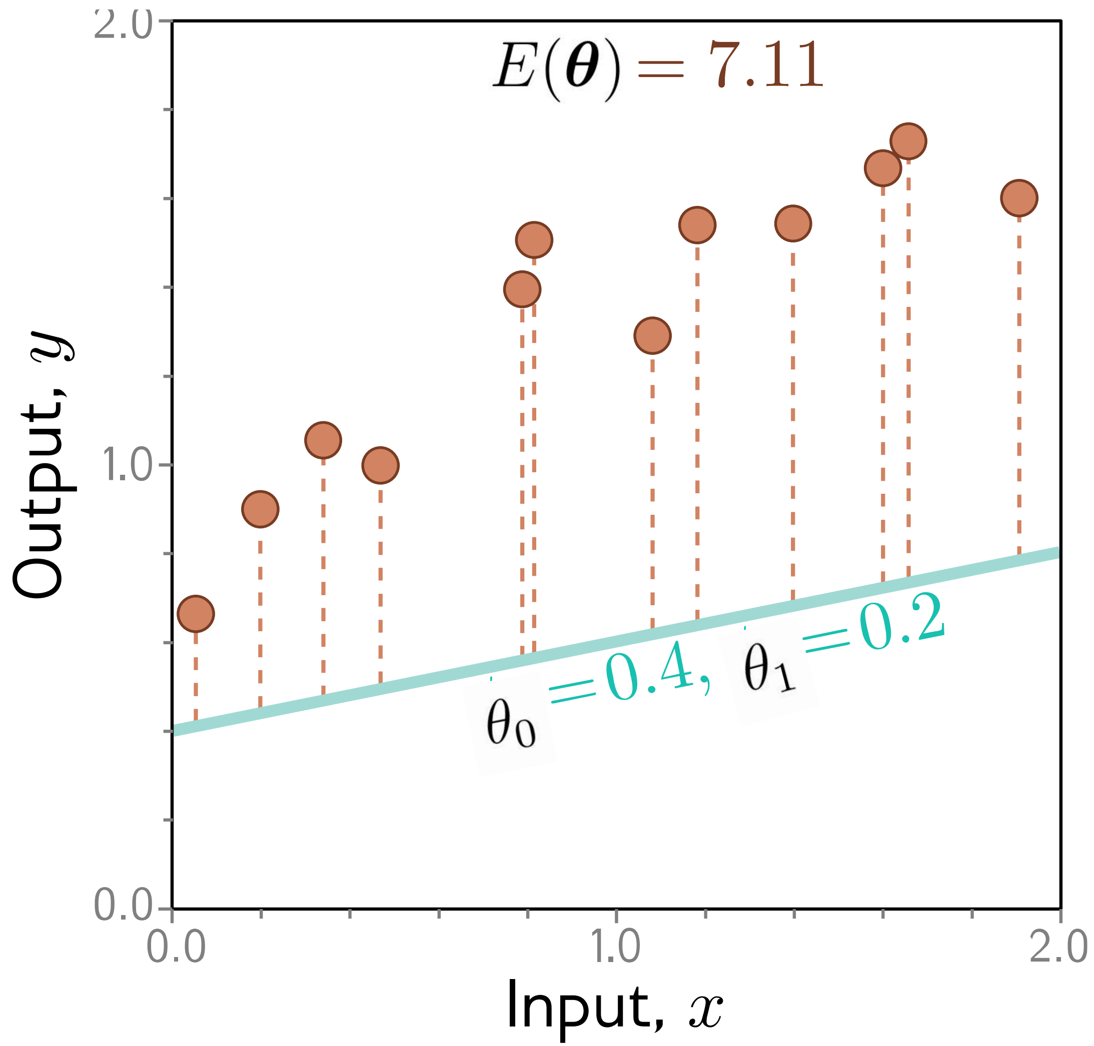
Let us simulate N samples $\{x_i, y_i\}$ (N=12)

We choose to minimize the empirical risk:

$$E(\boldsymbol{\theta}) = \frac{1}{12}\sum_{i=1}^{12} L(f_{\boldsymbol{\theta}}(x_i), y_i) =$$

$$= \frac{1}{12}\sum_{i=1}^{12}(\underbrace{\theta_0 + \theta_1 x_i}_{\text{approximation of the true value } y_i} - y_i)^2$$

true value



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model
Toy example: linear regression in one variable

Let us simulate N samples $\{x_i, y_i\}$ (N=12)

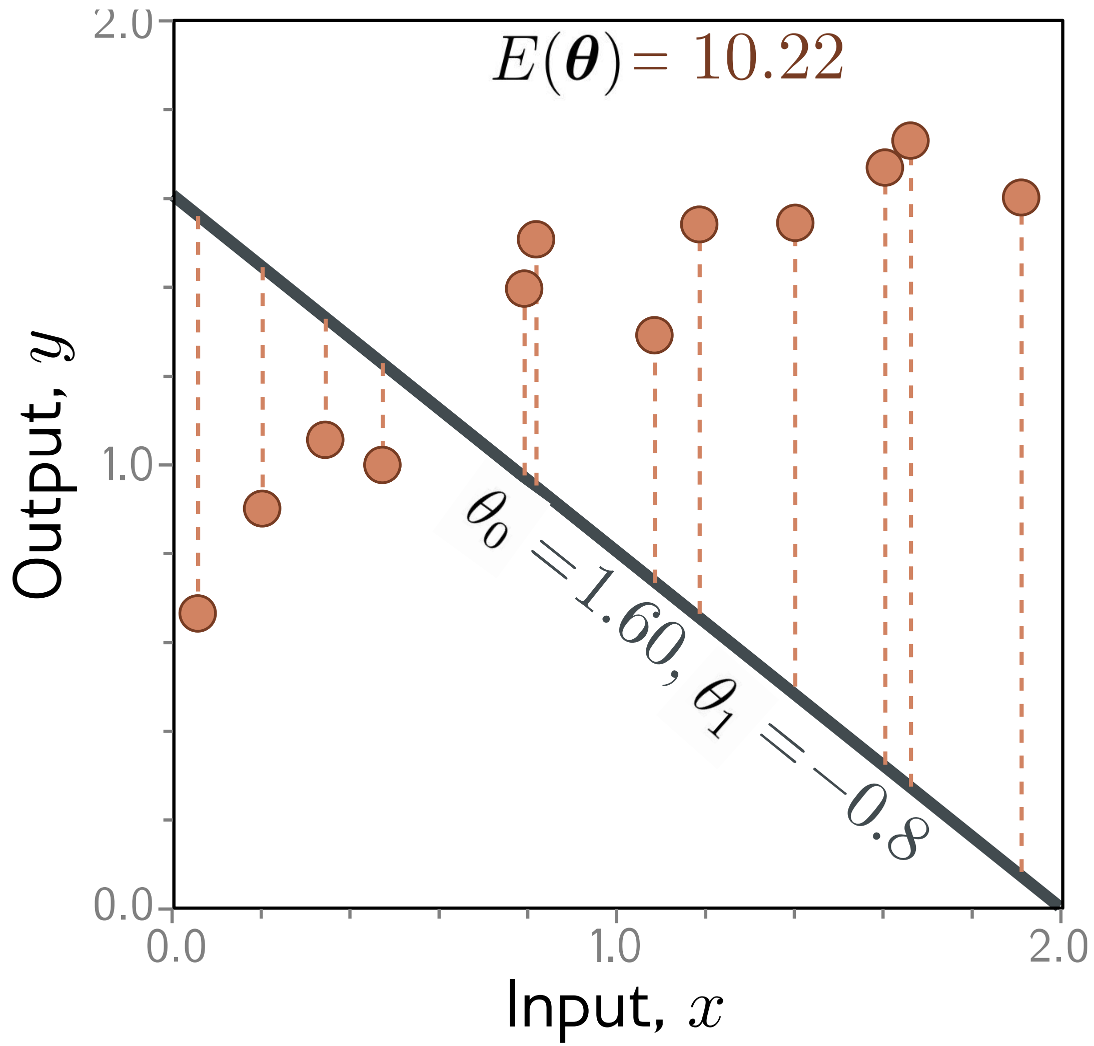
We choose to minimize the empirical risk:

$$E(\boldsymbol{\theta}) = \frac{1}{12}\sum_{i=1}^{12} L(f_{\boldsymbol{\theta}}(x_i), y_i) =$$

$$= \frac{1}{12}\sum_{i=1}^{12}(\underbrace{\theta_0 + \theta_1 x_i}_{\text{approximation of the true value } y_i} - y_i)^2$$

true value

approximation
of the true value y_i



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model
Toy example: linear regression in one variable

05/11

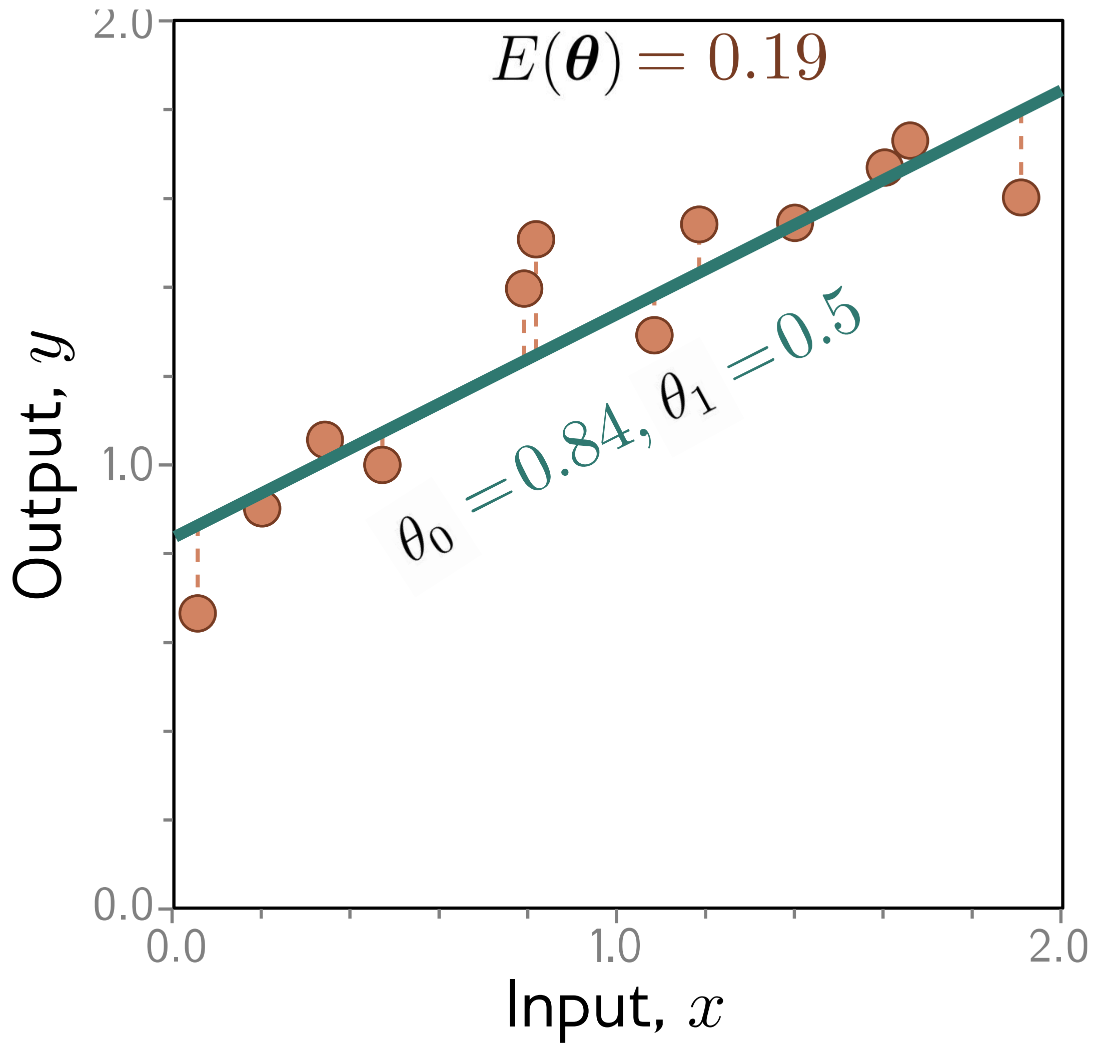
Let us simulate N samples $\{x_i, y_i\}$ (N=12)

We choose to minimize the empirical risk:

$$E(\boldsymbol{\theta}) = \frac{1}{12}\sum_{i=1}^{12} L(f_{\boldsymbol{\theta}}(x_i), y_i) =$$

$$= \frac{1}{12}\sum_{i=1}^{12}(\underbrace{\theta_0 + \theta_1 x_i}_{\text{approximation of the true value } y_i} - y_i)^2$$

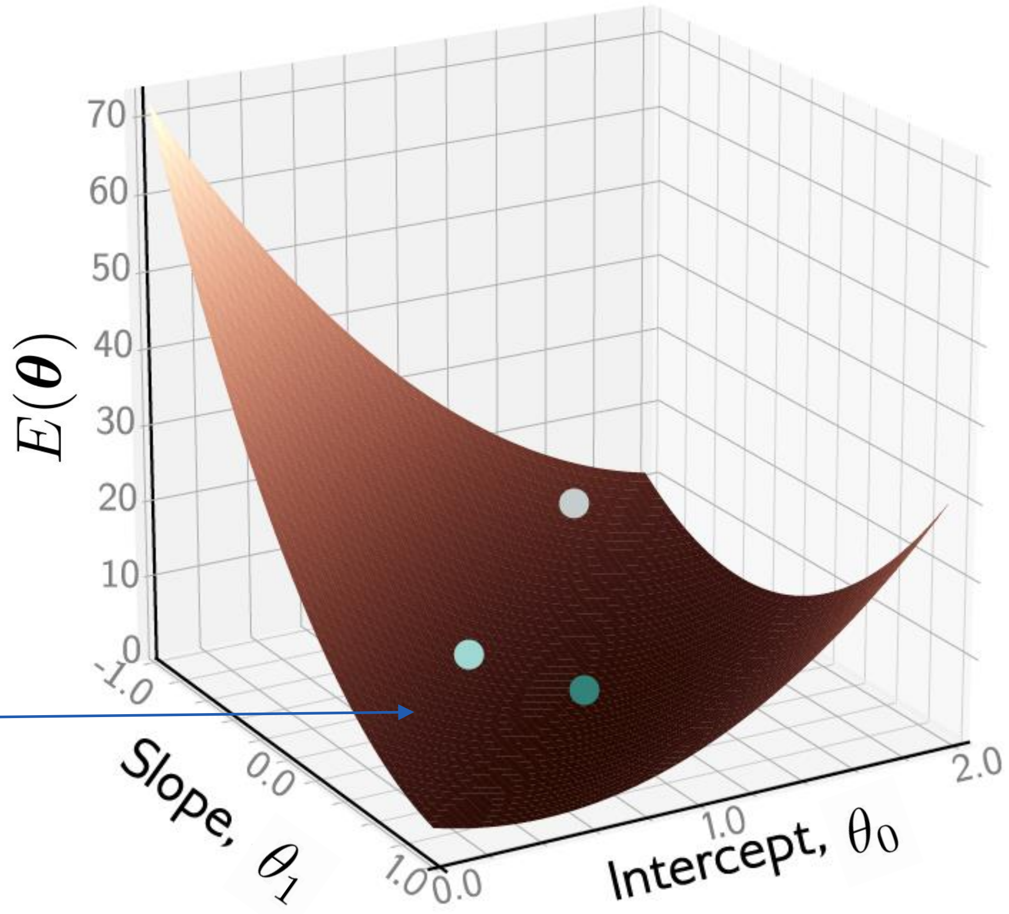
true value



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model

Toy example: linear regression in one variable



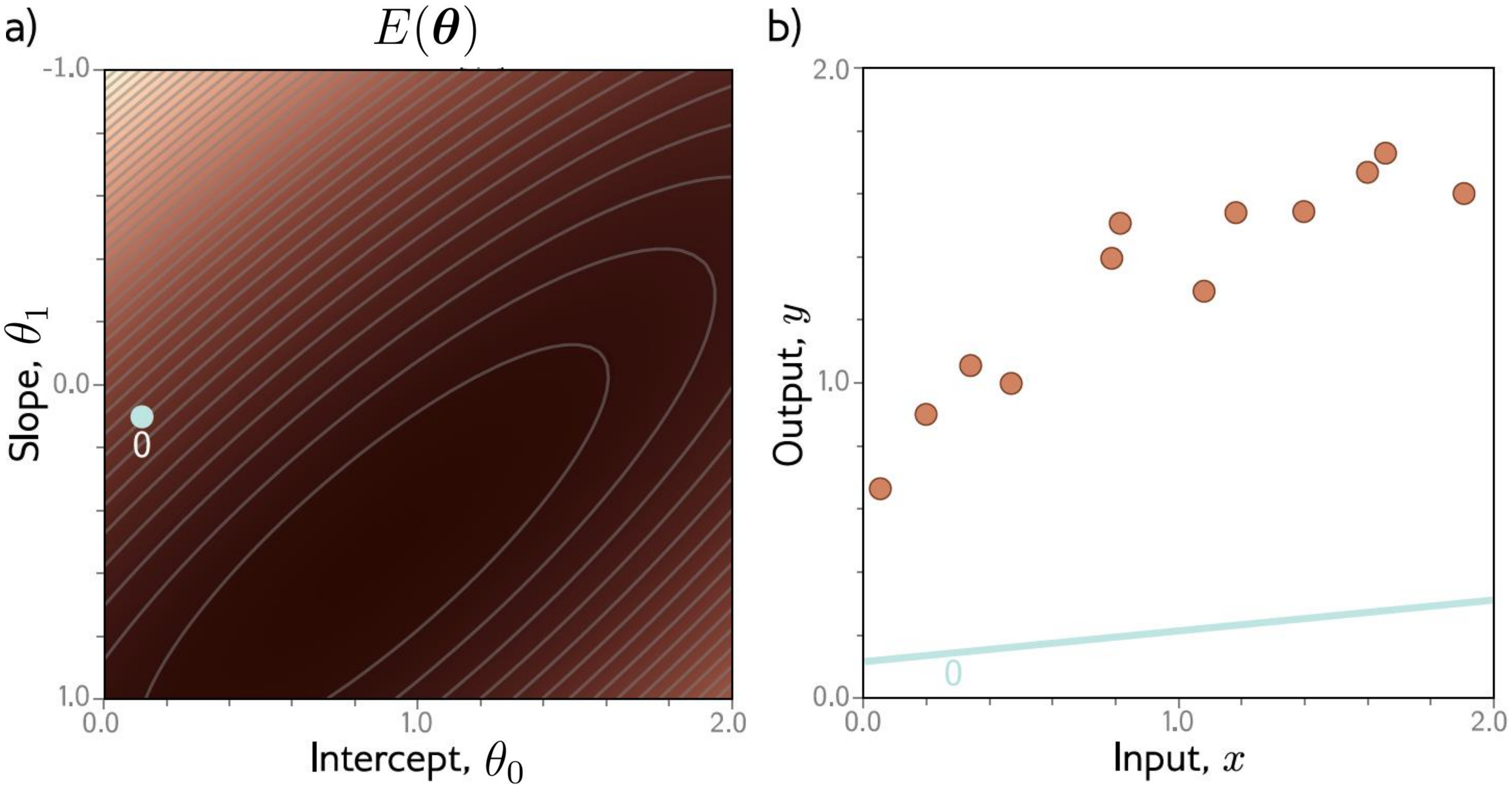
The darker the colour, the smaller the value of $E(\boldsymbol{\theta})$

From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

To minimize the empirical risk we can follow two approaches:

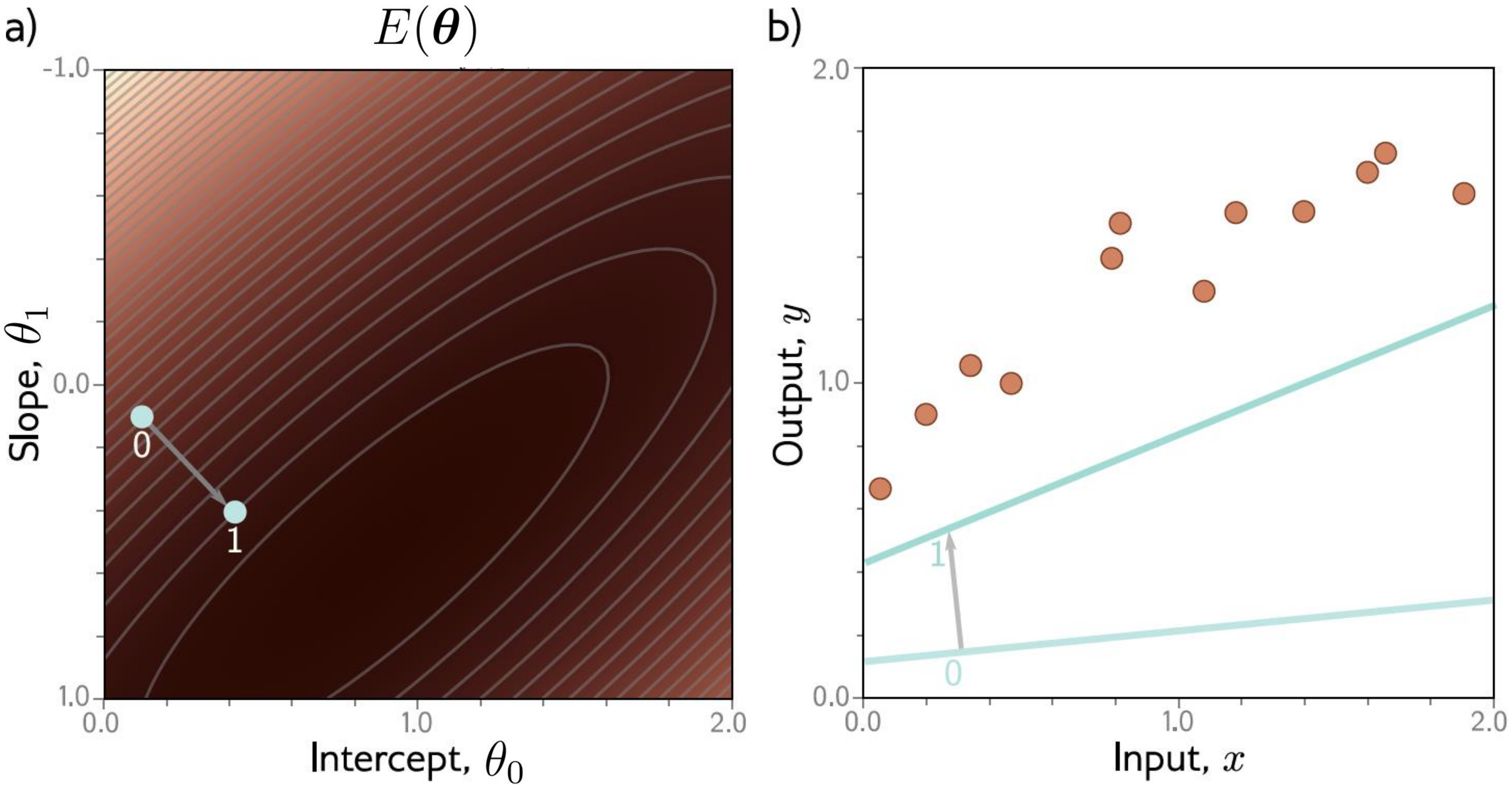
1. **"Brute force"** – there exists a closed solution to the minimization problem
2. **Stepwise approach** that approximates the closed solution in a finite number of steps

Let us train a machine learning model
Toy example: linear regression in one variable



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

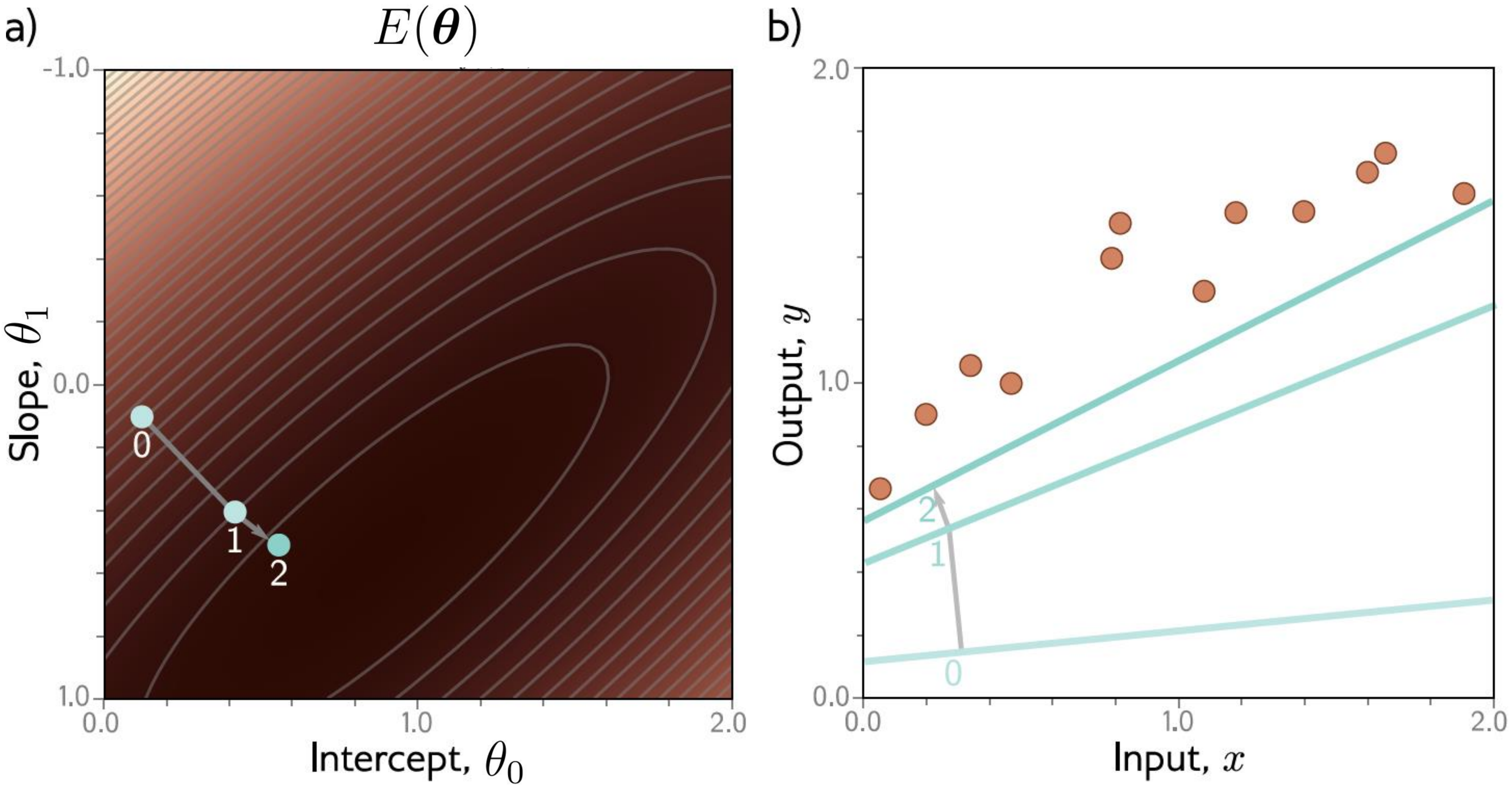
Let us train a machine learning model
Toy example: linear regression in one variable



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model
Toy example: linear regression in one variable

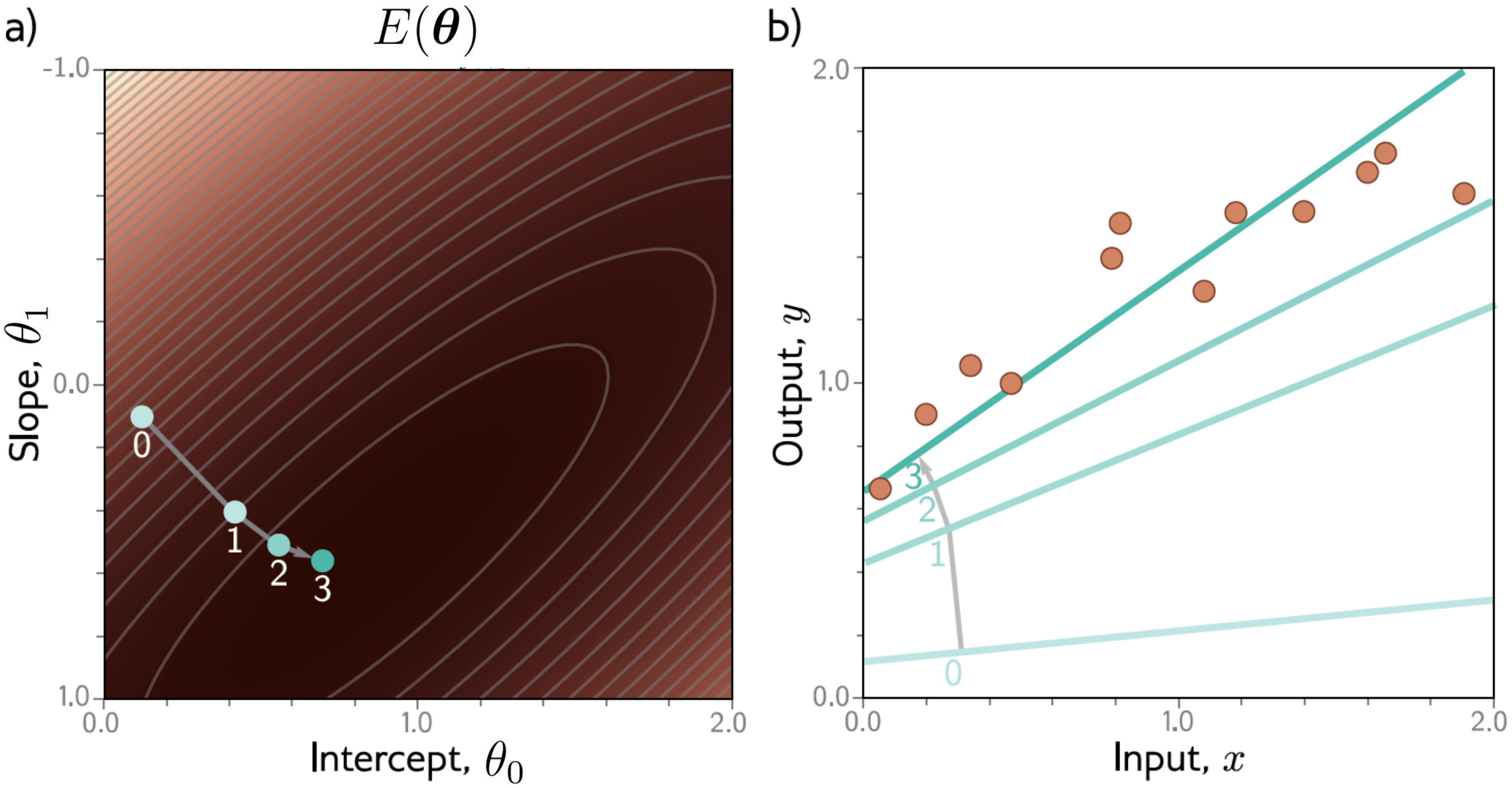
09/11



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

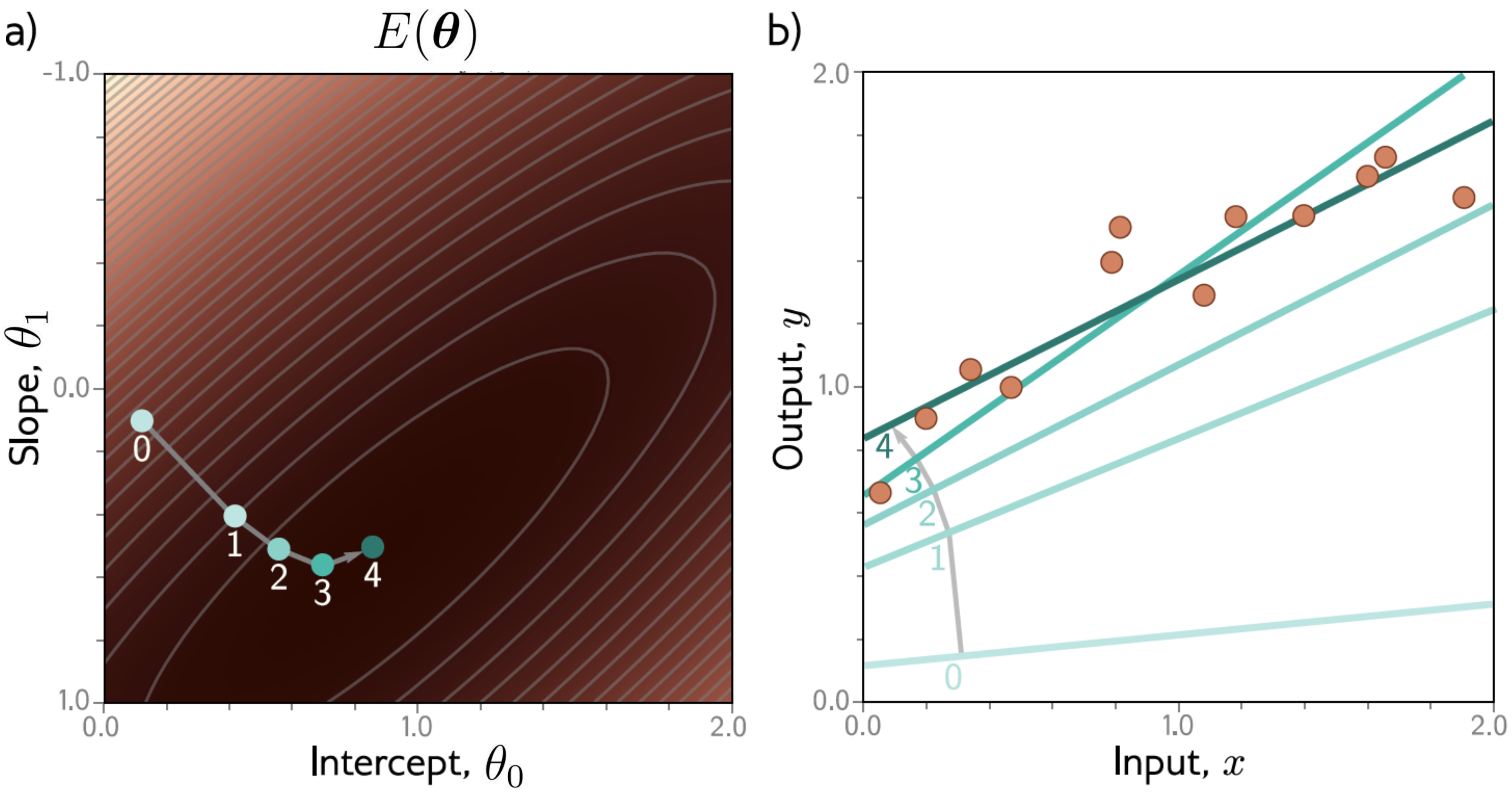
Let us train a machine learning model

Toy example: linear regression in one variable



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Let us train a machine learning model
Toy example: linear regression in one variable



From , S.J.D. Prince (2023), Understanding Deep Learning, MIT Press, Available at "http://udlbook.com"

Go to our Moodle page and download the file under
Notebooks and Colab Instructions/1_Lecture_March_08_2024

Google Colab: `lin_regr.ipynb` (PART 1)

Linear regression: Key takeaways

I	**Fundamentals**: Linear regression is an ideal starting point for understanding supervised learning, loss functions, and the bias-variance tradeoff.
II	**Interpretability**: The model's coefficients provide clear insights into the relationship between features and the target variable, useful for interpretative analysis in various domains.
III	**Regularization Introduction**: Linear regression introduces regularization concepts like Ridge and Lasso, crucial for handling overfitting and feature selection.
IV	**Advanced Models Foundation**: It lays the groundwork for understanding more complex models, demonstrating basic principles applicable to advanced algorithms like neural networks.

Self-Study: Closing with linear regression

Self-Study

From Moodle download the book:

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer.

- **Section 3.1.3** Assessing the Accuracy of the Model
- **Section 3.2.2** Some Important Questions
- **Section 3.3.3** Potential Problems

Feedback! See you on March 15th