WHAT IS LEARNING?

Learning = Hypothesis space + Risk + Optimization

- The **hypothesis space** $\mathcal{H}$ is the search space of the learning algorithm. It is a predefined set of functions (also called models) from which the learning algorithm picks one function/model.

  Example: Space of linear models.

- The **risk** is a metric to evaluate and compare the different models in the hypothesis space.

  Example: Sum of squared errors.

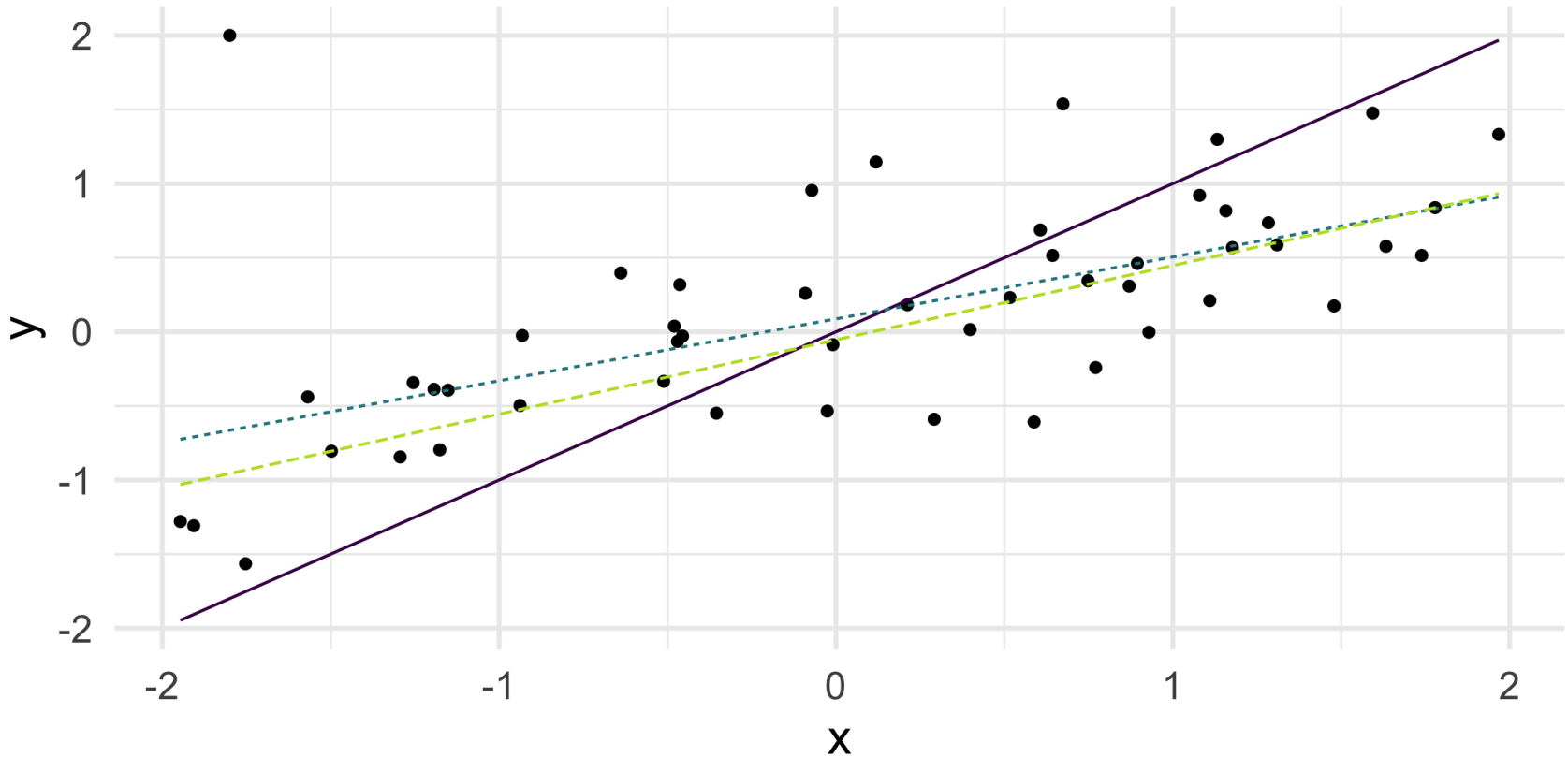
- The **optimizer** is the algorithm used to minimize the risk over the hypothesis space.

  Example optimizer: Gradient descent.

Loss Functions

LOSSES: MEASURING ERRORS POINT-WISE

Given the hypothesis space of linear models, which model will be returned by a learning algorithm (under “perfect” optimization)?



Answer: It depends on the metric we use to compare models.

LOSSES: MEASURING ERRORS POINT-WISE

- Let us assume that there is a probability distribution $\mathbb{P}_{xy}$ defined on $\mathcal{X} \times \mathcal{Y}$ induced by the process that generates the observed data $\mathcal{D}$.
- Further, let $(\mathbf{x}, y)$ denote the random variables that follow this distribution.
- We consider a model $f \in \mathcal{H}, f : \mathcal{X} \to \mathbb{R}^g$, and want to quantify the "goodness" of the function.
- Intuitively, a "good" function outputs values $f(\mathbf{x})$ which are close to the targets $y \in \mathcal{Y}$

$$y \approx f(\mathbf{x})$$

for $(\mathbf{x}, y) \sim \mathbb{P}_{xy}$.

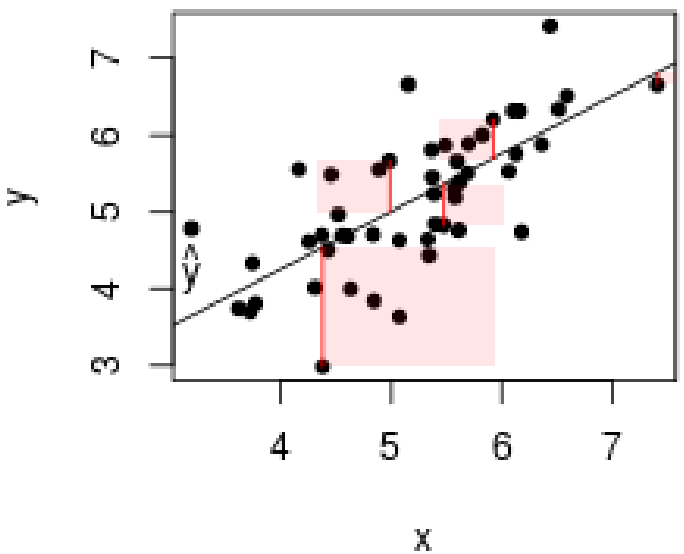
LOSSES: MEASURING ERRORS POINT-WISE

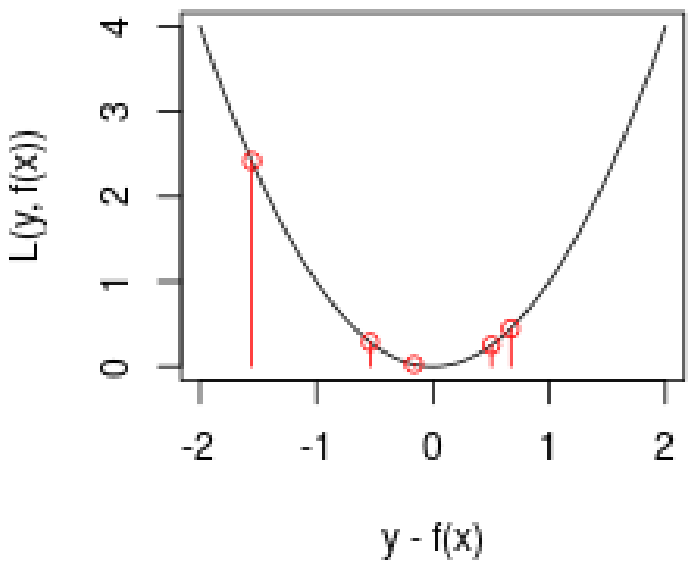
- We quantify the "goodness" of a model $f(\mathbf{x})$ **point-wise** via a **loss** function

$$L : \mathcal{Y} \times \mathbb{R}^g \to \mathbb{R},$$

  which compares the prediction and the real target $L(y, f(\mathbf{x}))$.

Example: $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ (point-wise squared errors)





LOSSES, RESIDUALS AND PSEUDO-RESIDUALS

- Regression losses usually only depend on the **residuals**

$$r \quad := \quad y - f(\mathbf{x})$$

- A loss is called **distance-based** if
  - it can be written in terms of the residual

$$L(y, f(\mathbf{x})) = \psi(r) \text{ for some } \psi : \mathbb{R} \to \mathbb{R}$$

  - $\psi(r) = 0 \Leftrightarrow r = 0$.
- A loss is **translation-invariant**, if $L(y + a, f(\mathbf{x}) + a) = L(y, f(\mathbf{x}))$.

LOSSES, RESIDUALS AND PSEUDO-RESIDUALS

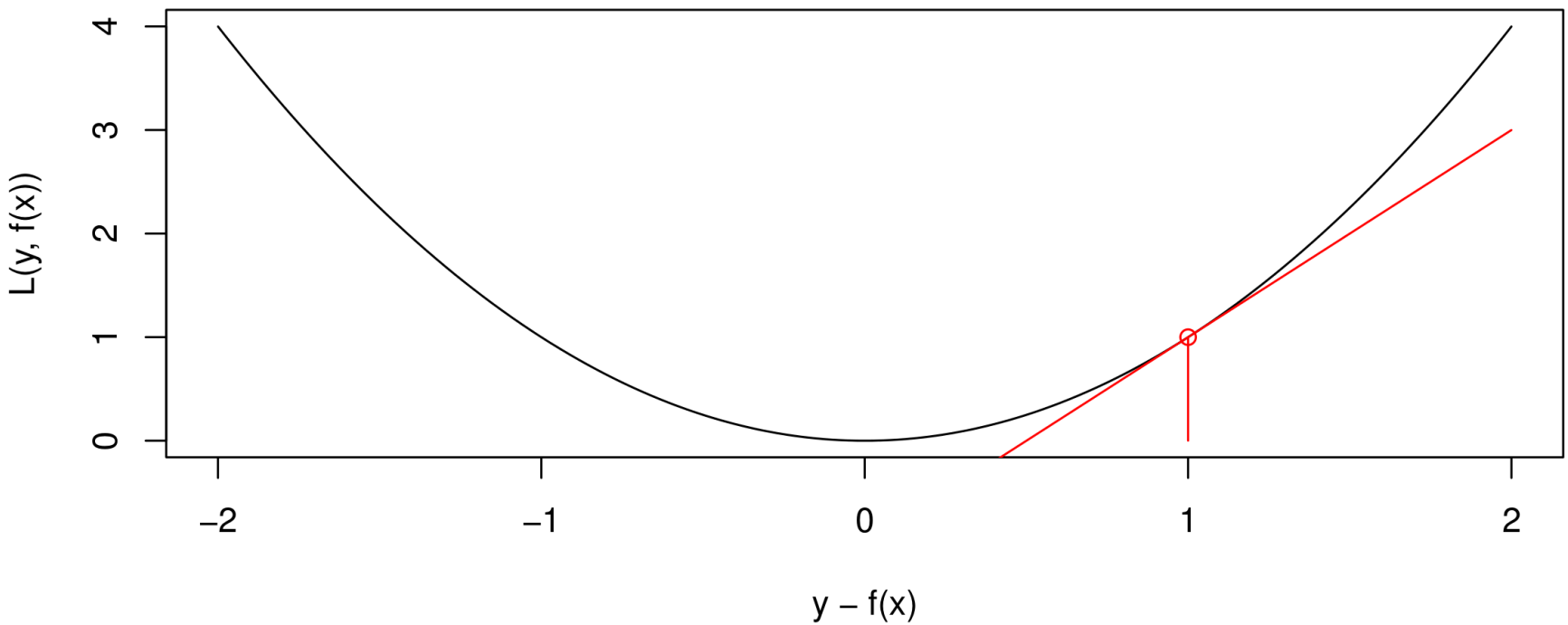
- We further define **pseudo-residuals** as the negative first derivatives of loss functions w.r.t. $f(\mathbf{x})$

$$\tilde{r} \quad := \quad -\frac{\partial L\left(y, f(\mathbf{x})\right)}{\partial f(\mathbf{x})}.$$

- We will gain more intuition about the principle of pseudo-residuals in a later chapter.

LOSS PLOTS

We call the plot that shows the point-wise error, i.e. the loss $L(y, f(\mathbf{x}))$ vs. the **residuals** $r := y - f(\mathbf{x})$ (for regression), **loss plot**. The pseudo-residual corresponds to the slope of the tangent in $(y - f(\mathbf{x}), L(y, f(\mathbf{x})))$.



Theoretical Risk Minimization

(THEORETICAL) RISK MINIMIZATION

- The (theoretical) **risk** associated with a certain hypothesis $f(\mathbf{x})$ measured by a loss function $L(y, f(\mathbf{x}))$ is the **expected loss**

$$\mathcal{R}(f) := \mathbb{E}_{xy}[L(y, f(\mathbf{x}))] = \int L(y, f(\mathbf{x})) \, \mathrm{d}\mathbb{P}_{xy}.$$

- Our goal is to find a hypothesis $f(\mathbf{x}) \in \mathcal{H}$ that **minimizes** the risk.

(THEORETICAL) RISK MINIMIZATION: LIMITATION

Problem: Minimizing $\mathcal{R}(f)$ over f is generally not feasible or practical:

- $\mathbb{P}_{xy}$ is unknown (if it were known, we could use it directly to construct optimal predictions).
- We could estimate $\mathbb{P}_{xy}$ in non-parametric fashion from the data $\mathcal{D}$ (i.i.d. drawn from $\mathbb{P}_{xy}$), e.g. by kernel density estimation, but this really does not scale to higher dimensions (see "curse of dimensionality").
- We can efficiently estimate $\mathbb{P}_{xy}$, if we place rigorous assumptions on its distributional form, and methods like discriminant analysis work exactly this way. **Machine learning** usually studies more flexible models.

Empirical Risk Minimization

EMPIRICAL RISK MINIMIZATION

Let

$$\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)} \right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)} \right) \right),$$

with observations $\left(\mathbf{x}^{(i)}, y^{(i)} \right) \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{xy}$.

An alternative (without directly assuming anything about $\mathbb{P}_{xy}$) is to approximate $\mathcal{R}(f)$ based on $\mathcal{D}$ by means of the **empirical risk**

$$\mathcal{R}_{\text{emp}}(f) \quad := \quad \sum_{i=1}^{n} L \left(y^{(i)}, f \left(\mathbf{x}^{(i)} \right) \right)$$

Learning then amounts to **empirical risk minimization**

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \mathcal{R}_{\text{emp}}(f).$$

EMPIRICAL RISK MINIMIZATION

Notes:

- The risk is often denoted as empirical mean over $L(y, f(\mathbf{x}))$

$$\bar{\mathcal{R}}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right).$$

  The factor $\frac{1}{n}$ does not make a difference in optimization, so we will consider $\mathcal{R}_{\text{emp}}(f)$ most of the time.

- If f is parameterized by $\boldsymbol{\theta} \in \Theta$, this becomes:

$$\begin{aligned} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) &= \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) \\ \hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) \end{aligned}$$

MACHINE LEARNING = OPTIMIZATION?

Learning (often) means solving the above **optimization problem**.
There is a very tight connection between ML and optimization, but still, there are substantial differences:

- In machine learning, we want to find a model that is optimal w.r.t. the theoretical risk $\mathcal{R}(f)$.
- In general, we cannot compute the theoretical risk, because the data generating process $\mathbb{P}_{xy}$ is not known.
- Instead, we use observed data $\mathcal{D}$ to formulate the empirical risk $\mathcal{R}_{\text{emp}}(f)$.
- However, $\mathcal{R}_{\text{emp}}(f)$ is a good approximation for $\mathcal{R}(f)$ only if $\mathcal{D}$ is an unbiased, independent and large enough sample from $\mathbb{P}_{xy}$.
- So in machine learning, we optimize an approximated version of the problem we are actually interested in.

THE ROLE OF LOSS FUNCTIONS

Why should we care about how to choose the loss function $L(y, f(\mathbf{x}))$?

- **Statistical** properties of f: Choice of loss implies statistical properties of f like robustness and an implicit error distribution.
- **Computational** / **Optimization** complexity of the optimization problem: The complexity of the optimization problem

$$\arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$$

  is influenced by the choice of the loss function, i.e.

  - Smoothness of the objective

    Some optimization methods require smoothness (e.g. gradient methods).
  - Uni- or multimodality of the problem

    If $L(y, f(\mathbf{x}))$ is convex in its second argument, and $f(\mathbf{x} \mid \boldsymbol{\theta})$ is linear in $\boldsymbol{\theta}$, then $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ is convex; every local minimum of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ is a global one. If L is not convex, $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ might have multiple local minima (bad!).