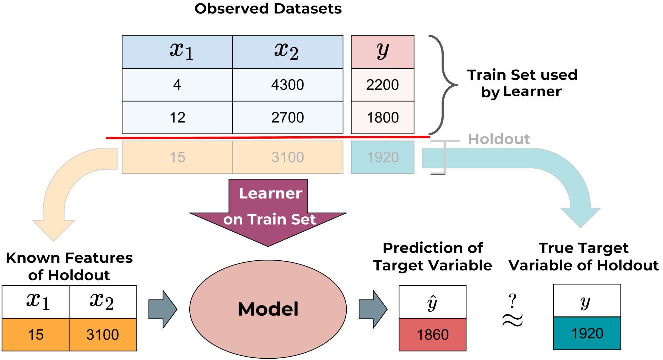
Introduction to Machine Learning

Introduction: Losses & Risk Minimization

compstat-lmu.github.io/lecture_i2ml

HOW TO EVALUATE MODELS

Compare predictions from a model with observed target values:



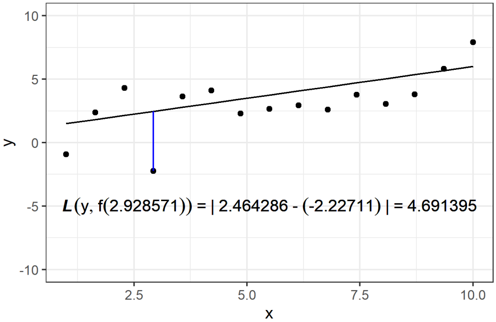
MOTIVATION

- Assume we trained a model to predict flat rent based on some features (size, location, age, ...).
- The real rent of a flat is EUR 1600, our model predicts EUR 1300.
- How do we measure the performance of our model?
- Need to define a suitable criterion, e.g.:
  - Absolute error $|1600 - 1300| = 300$
  - Squared error: $(1600 - 1300)^2 = 90000$
    (puts more emphasis on predictions that are far off the mark)
- The choice of this metric has a major influence on the final model, because it determines what constitutes a *good* model: it will determine the ranking of the different models $f \in \mathcal{H}$.
- the metric we use is called the **loss function**.

LOSS

The **loss function** $L\left(y, f(\mathbf{x})\right)$ quantifies the "quality" of the prediction $f(\mathbf{x})$ of a single observation $\mathbf{x}$.

How "close" $f(\mathbf{x})$ is to y can be quantified e. g. by $L\left(y, f(\mathbf{x})\right) = |f(\mathbf{x}) - y|$.

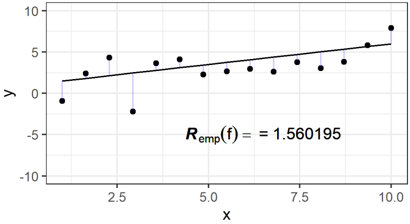


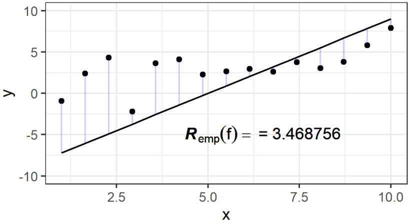
RISK

The **risk function** quantifies the "quality" of the whole model.

The ability of a model f to reproduce the association between $\mathbf{x}$ and y that is present in the data $\mathcal{D}$ can be measured by the **average loss**, also called **"empirical risk"**:

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



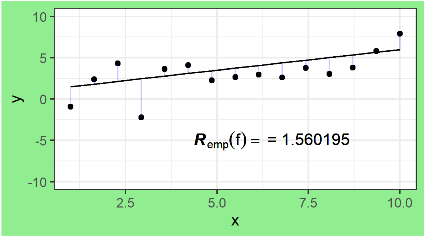


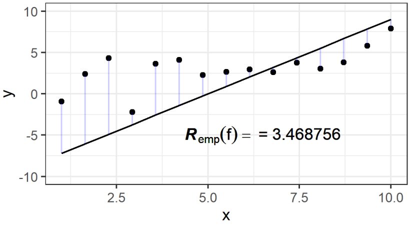
RISK MINIMIZATION

The best model is the model with the smallest risk.

The process of finding the best model is called **empirical risk minimization** (ERM).

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





RISK MINIMIZATION

Since the model f is usually defined by **parameters** $\boldsymbol{\theta}$ in a parameter space Θ, this becomes:

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

Most learners in ML try to solve the above *optimization problem*, which implies a tight connection between ML and optimization.

FURTHER REMARKS

- For regression tasks, the loss often only depends on the residual $L(y, f(\mathbf{x})) = L(y - f(\mathbf{x})) = L(\epsilon)$.
- The choice of loss implies which kinds of errors are important or not – requires *domain knowledge*!
- For learners that correspond to probabilistic models, the loss determines / is equivalent to distributional assumptions.
- Since learning can be re-phrased as minimizing the loss, the choice of loss strongly affects the computational difficulty of learning:
  - How smooth is $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ in $\boldsymbol{\theta}$?
  - Is $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ differentiable so that we can use gradient-based methods?
  - Does $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ have multiple local minima or saddlepoints over Θ?