Data Science Summer School Leoben 2022

Peter Auer, Martin Antenreiter,
Paul O'Leary, Elmar Rückert,
Lorenz Romaner

Intro ML

- Data driven approaches (aka ML)
- Types of ML
- Correlation vs. Causality
- Business cases in ML
- Evaluating results of ML
- Model selection

- Regression methods

What is Machine Learning?

- Goal: Calculating a **prediction**
  - **useful**, actionable

- **Data** driven
  - use (historic) data to **calculate** the prediction

Type of predictions

- Data type
  - Discrete: classification
  - Continuous: regression

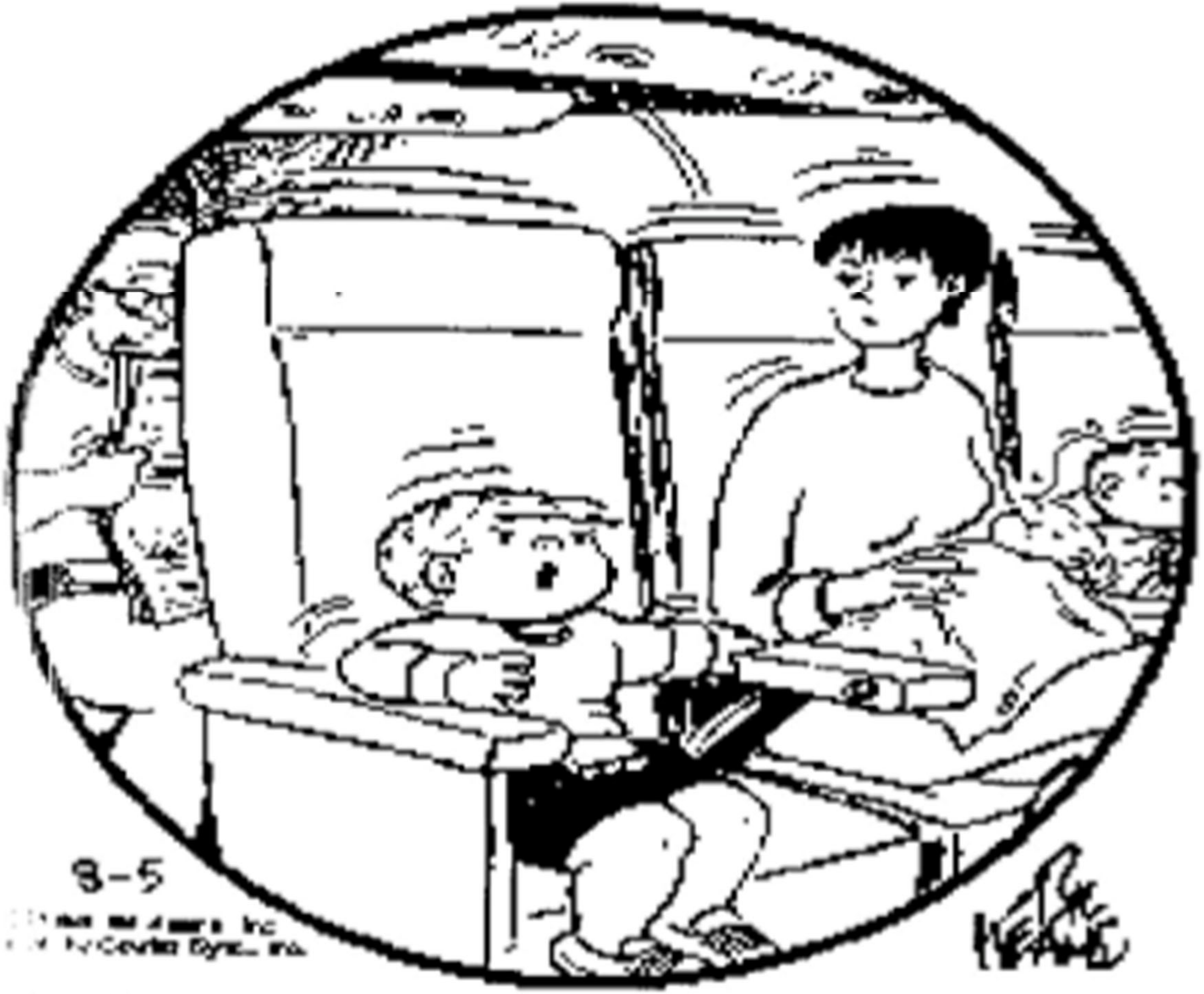
- Interpretation:
  - Label, action
  - Outcome, model parameter
  - Model, significant correlations

Correlation vs. causality

Chair for Information Technology

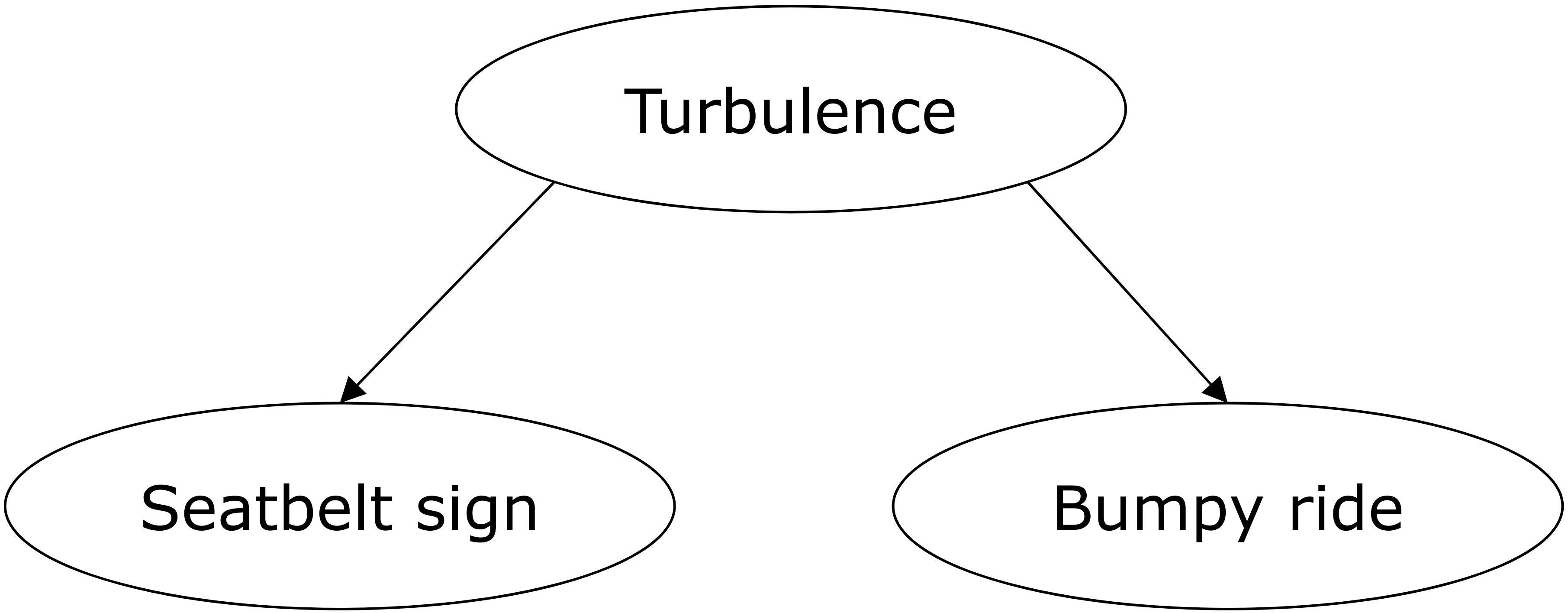
- We can observe only correlations!

- Causality is *plausible* if hidden causes (*confounding factors*) have been accounted for *as much as possible*.



"I wish they didn't turn on that seatbelt sign so much! Every time they do, it gets bumpy."

Confounding factor



Pragmatic approach to correlations

- Correlations are useful, if they allow good predictions.

- **Warning:**
  If the actual cause changes, then predictions based on previous correlations may become useless.

Utility of predictions

- Definition of the utility depends on the goal for making predictions.
- Boils down to an objective function.

- Example 1: Predictive maintenance
- Example 2: Electric Arc Furnace

- More difficult to find an objective function for unsupervised learning.

Simple objective: Minimize loss functions

- Notation:
  - Input $\boldsymbol{x}$
  - Prediction $\hat{y} = h(\boldsymbol{x})$ by hypothesis h
  - Correct prediction y
  - Loss $L(\boldsymbol{x}, y, \hat{y})$.

Simple loss functions

- Classification error:

$$L(\boldsymbol{x}, y, \hat{y}) = \begin{Bmatrix} 1 \text{ if } \hat{y} \neq y \\ 0 \text{ if } \hat{y} = y \end{Bmatrix}$$

- Quadratic error (for regression):

$$L(\boldsymbol{x}, y, \hat{y}) = (\hat{y} - y)^2$$

Example 1: Utility of predictive maintenance

- Gain from saved maintenance
- Cost of unnecessary maintenance
- Cost of failure caused by missed maintenance
- (Cost for collecting data and learning how to predict)

Example 2: Power consumption of an electric arc furnace

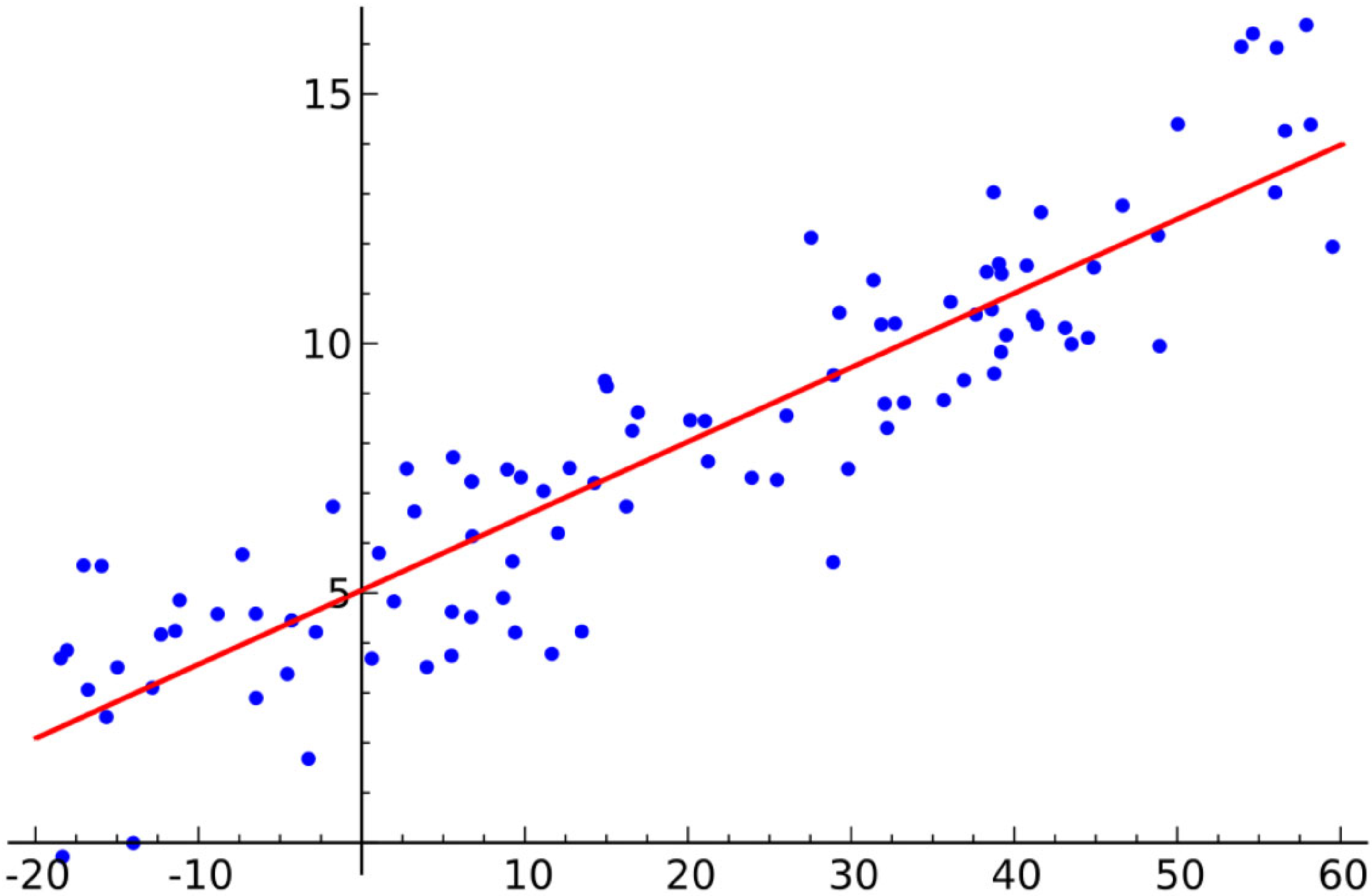
- Site with furnace and other machines
- Logs of power consumption and activities
- Goal:
  - Predict power consumption
  - Predict peeks in the power consumption

Which training data can we use?

- Supervised Learning with historic data:
  - Data were collected in the past.
  - Consist of pairs $(\boldsymbol{x}_i, y_i)$ where $y_i \approx f(\boldsymbol{x}_i)$ is approximately the correct prediction for $\boldsymbol{x}_i$.
- Example linear regression:

$$y_i = \boldsymbol{\theta} \cdot \boldsymbol{x}_i + \varepsilon_i \;,\; i = 1, \ldots, m.$$

Linear Regression



Problem with historic data

- How well do these data represent the current situation?
  - Are the correlations in the historic data still predictive?

- Can be decided only case by case.
- Statistical methods can be used to check.

Other types of training data

- Unsupervised learning:
  Only inputs $\boldsymbol{x}_i$, $i = 1, \dots, m$.
  - Clustering
  - Finding associations

- Reinforcement learning:

$$(s_1, a_1, r_1), (s_2, a_2, r_2), \dots$$

Reward r_t for action a_t in state s_t.

Other types of data acquisition

- Online learning:
  - Data $(\boldsymbol{x}_t, y_t)$ arrive sequentially,
  - y_t needs to be predicted before it is observed.
- Active learning:
  - When obtaining y_i for some $\boldsymbol{x}_i$ is expensive, the learning algorithm might select inputs $\boldsymbol{x}_i$ for which y_i is obtained.

Which learning algorithm?

- Approach: Calculate prediction function (hypothesis) that minimizes (surrogate) loss function on training data.

- Example linear regression:
  Choose $\boldsymbol{\theta}$ such that

$$\sum_{i=1}^{n}(y_i - \boldsymbol{\theta} \cdot \boldsymbol{x}_i)^2 \rightarrow Min$$

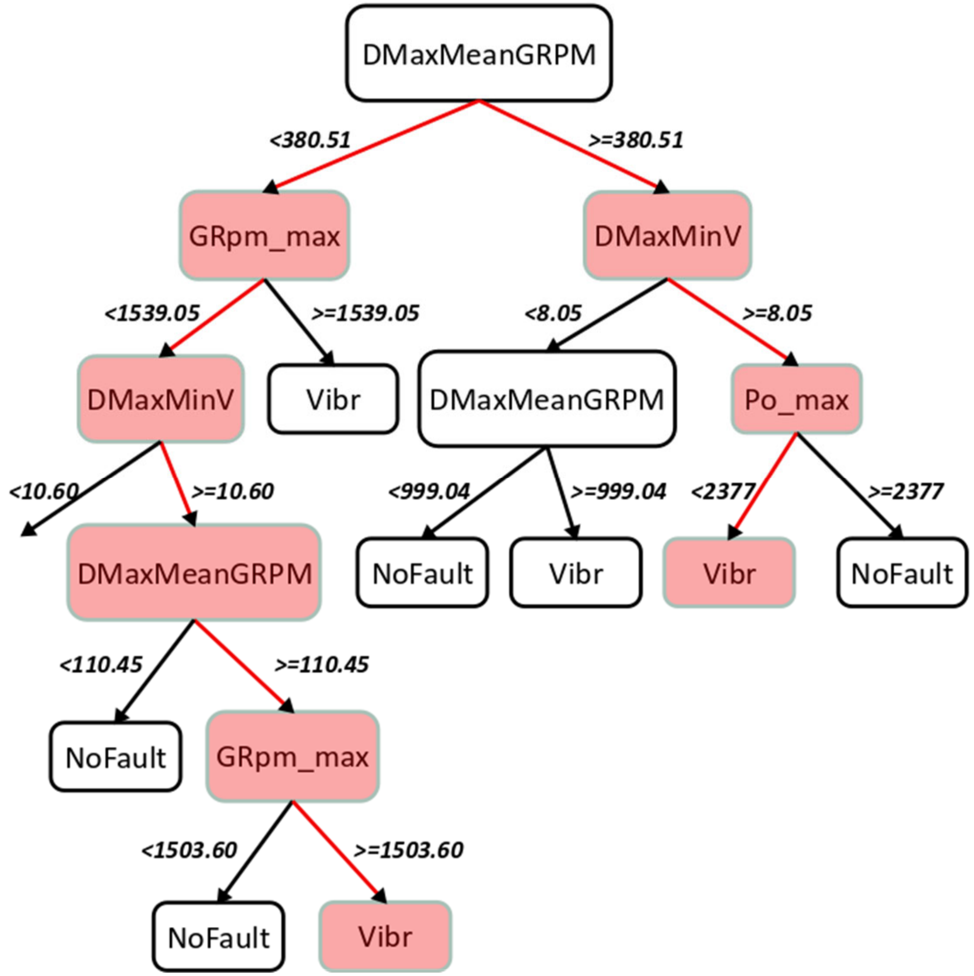
- Prediction: $\hat{y} = h(\boldsymbol{x}|\boldsymbol{\theta}) = \boldsymbol{\theta} \cdot \boldsymbol{x}$

Other classes of prediction functions $h(x|\theta)$

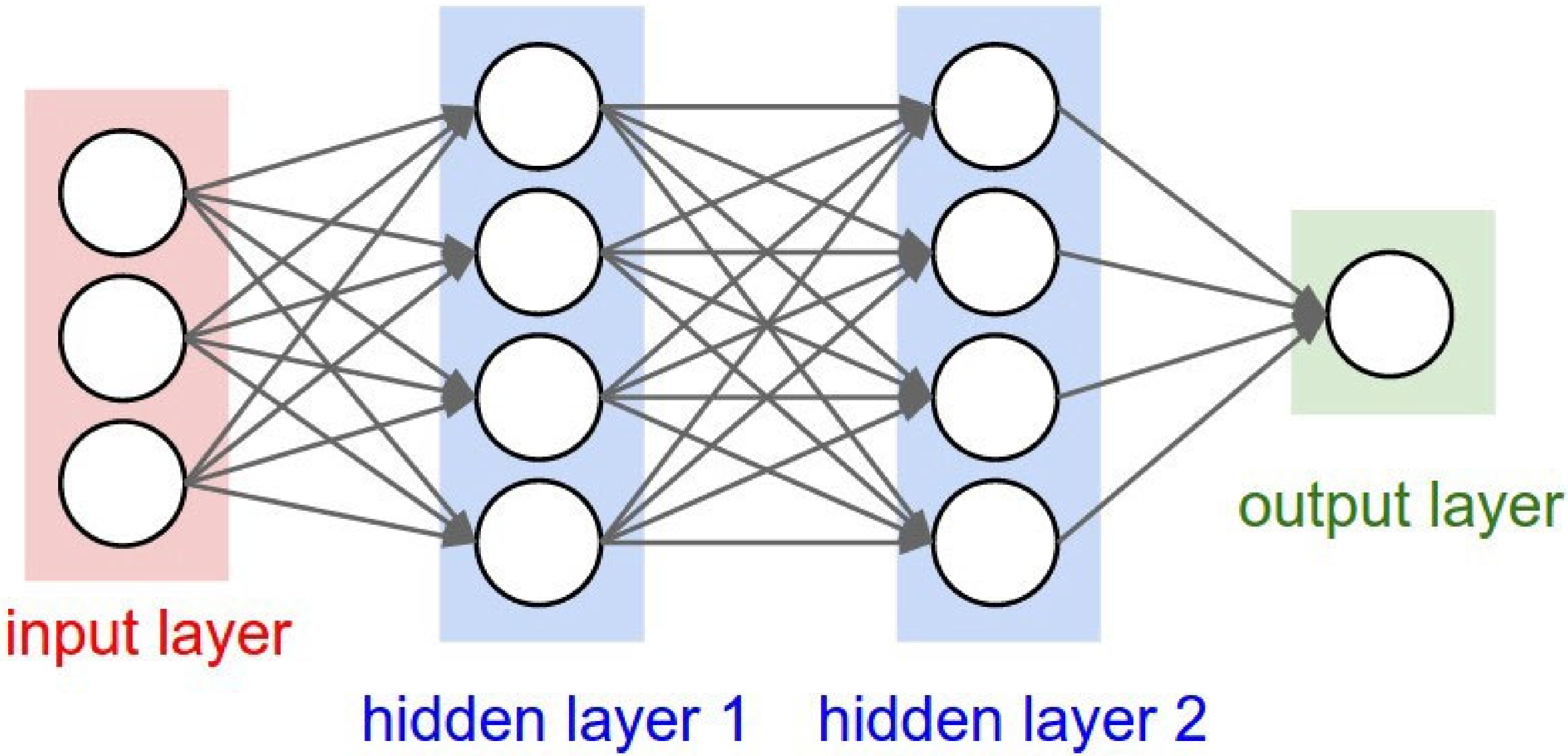
- Decision trees
- Neural networks
- Nearest neighbor classifiers
- Support Vector Machines
- ...

- No free lunch!

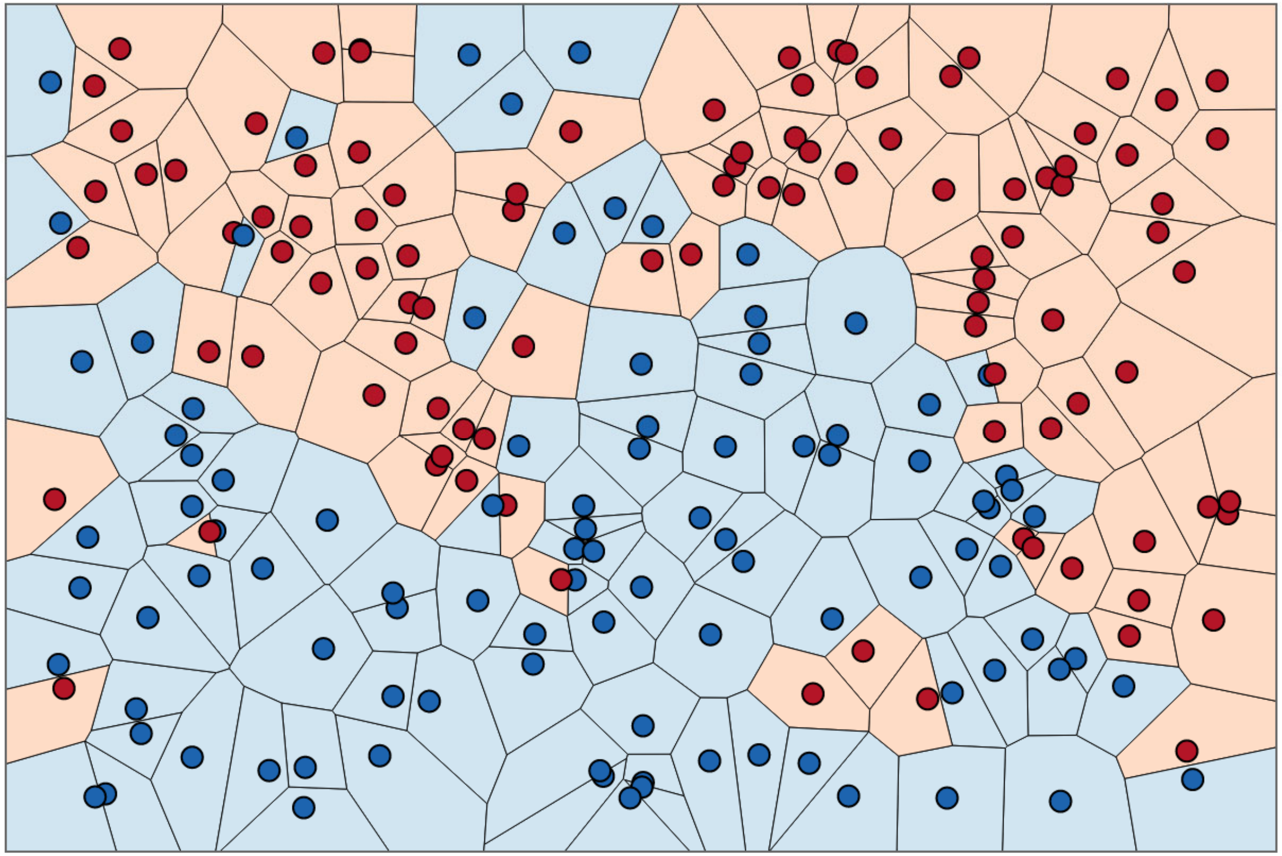
Decision tree



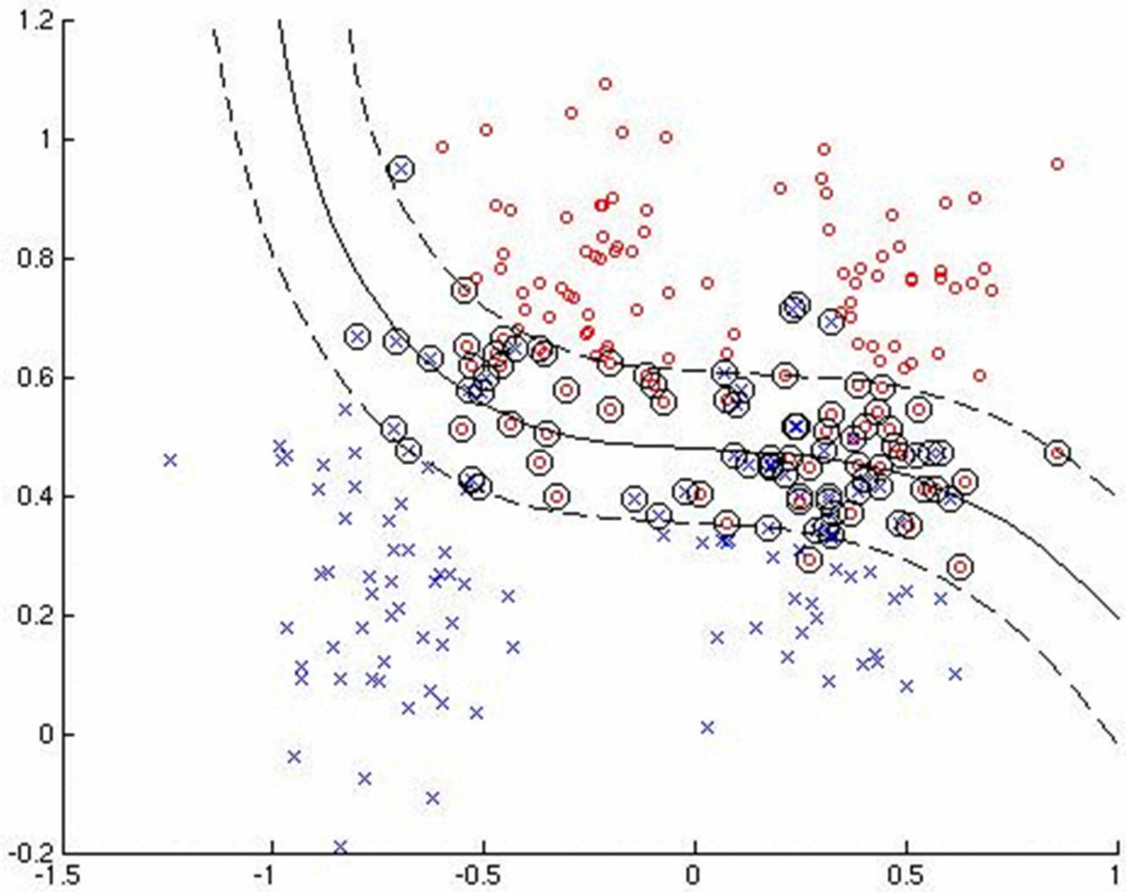
Neural network



Nearest neighbor classifier



Support Vector Machine



Chair for Information Technology

The learning algorithm

- The hypotheses class (class of prediction functions) $h(\boldsymbol{x}|\boldsymbol{\theta})$ is usually chosen a-priori.

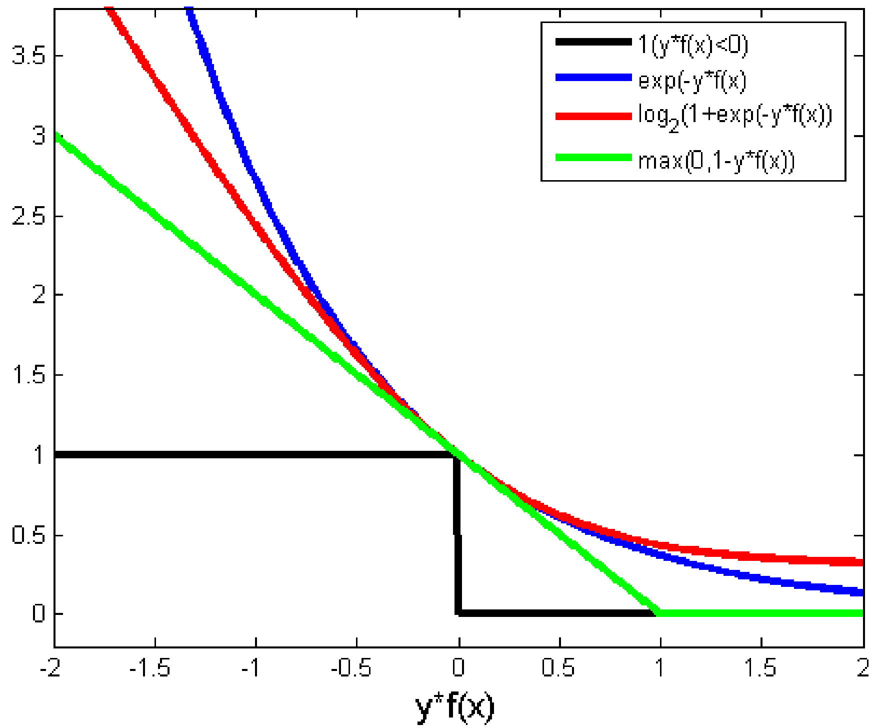
- The learning algorithm optimizes the parameters $\boldsymbol{\theta}$ to minimize a loss function on the training data:

$$\sum_{i=1}^{n} L(\boldsymbol{x}_i, y_i, h(\boldsymbol{x}_i|\boldsymbol{\theta})) \rightarrow Min$$

Surrogate loss functions

- Some loss functions are hard to optimize, e.g. the classification error, $y_i \neq h(x_i|\boldsymbol{\theta})$.

- Use a surrogate loss function that is easier to optimize.



Evaluating the learned hypothesis

- Uses optimal parameter $\boldsymbol{\theta}$ with

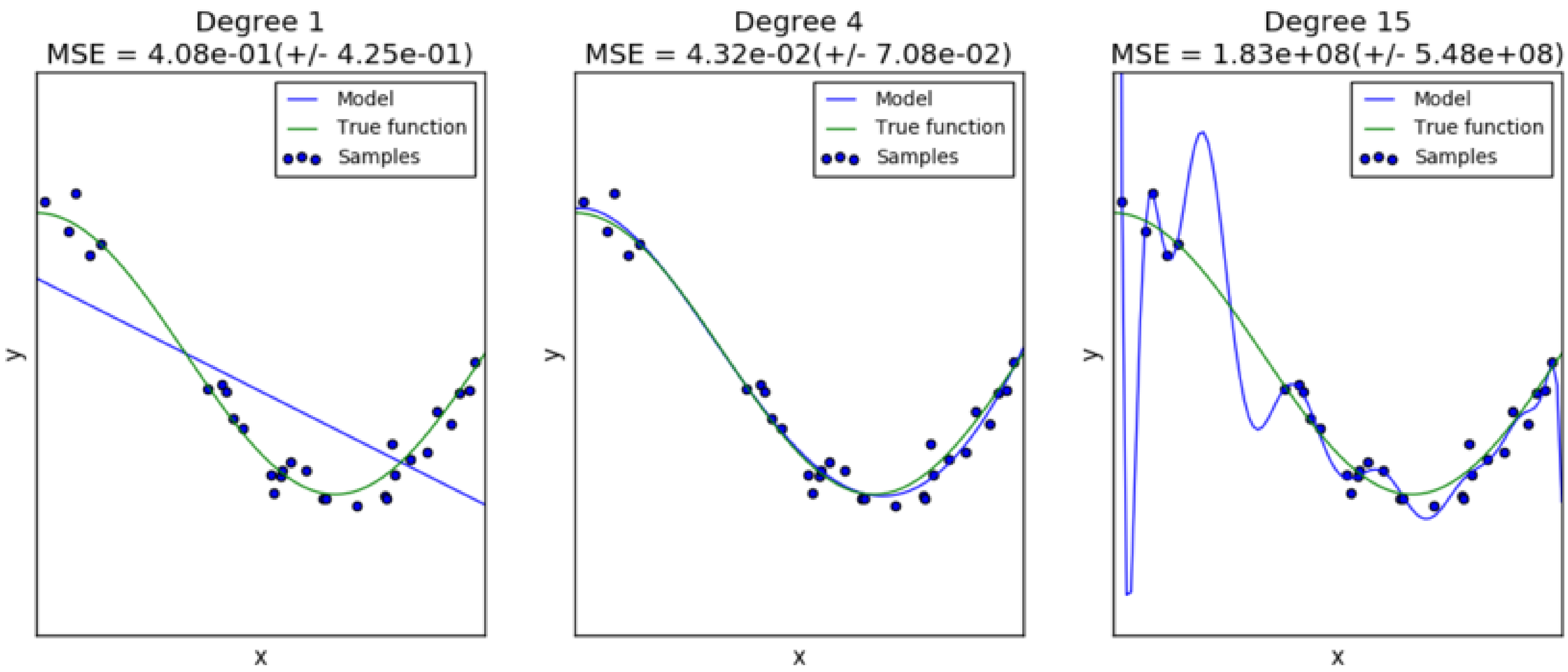
$$\sum_{i=1}^{n} L(\boldsymbol{x}_i, y_i, h(\boldsymbol{x_i}|\boldsymbol{\theta})) \rightarrow Min$$

- Does this hypothesis predict well for new data,

$$L(\boldsymbol{x}, y, h(\boldsymbol{x}|\boldsymbol{\theta})) = ?$$

- **Need to evaluate the hypothesis with test data.**

Underfitting – Overfitting



Underfitting - Overfitting

Chair for Information Technology

- Unterfitting: The class of prediction functions is not rich enough to allow for good predictions.

- Overfitting symptom:
  The loss on new data is “much larger” than on the training data.

Reason for Overfitting

- Parameters $\boldsymbol{\theta}$ are fitted too tightly to the training data
  $$\frac{1}{n}\sum_{i=1}^{n} L(\boldsymbol{x}_i, y_i, h(\boldsymbol{x}_i|\boldsymbol{\theta}))$$
  encoding also errors in the training data.
- This may cause large prediction loss on new data.
- Counter measure: Restrict the optimization of the prediction function.

Model selection

- Explicitly or implicitly choose the class of prediction functions.
- Concrete methods are often tied to the type of prediction function.
- Usually require the selection of hyperparameters.
- Mostly done by using a validation set or cross validation.

Methods for model selction

- *Pruning* for decision trees
- *Choice of architecture* and *Early Stopping* for neural networks
- Regularization:

$$\sum_{i=1}^{n} L\big(\boldsymbol{x}_i, y_i, h(\boldsymbol{x}_i|\boldsymbol{\theta})\big) + C * R(\boldsymbol{\theta})$$

- $R(\boldsymbol{\theta})$ is a regularization function that prefers simple/small parameters.

Chair for Information Technology

ML-Process: (Supervised Learning from historic data)

1. Problem description
   - What needs to be predicted, using which information?
   - What is the loss function?
2. Are there good training data?
3. Split into training, evaluation and test data.
4. Which hypotheses class?
   - Which preprocessing of the data?
   - Which learning algorithm?
   - Which hyperparameters?
5. Learn a hypothesis
6. Evaluate the hypothesis

⮤ **Iterate**

6. Test of the final hypothesis

A-priori knowledge and physical models

- The less an algorithm needs to learn, the easier learning is:
  - Use a-priori knowledge and existing models.

- Often this can be done by preprocessing the data, such that only the missing parts need to be learned.
- Or restrictions can be put on the prediction function.

Final test of hypotheses

Assumptions:

- We have trained a prediction function $h : \mathbb{R}^d \to \mathbb{R}$.
- We have n test examples $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, drawn independently from some distribution $P(\boldsymbol{x}, y)$.

Goal:

- Estimate the error $L(\boldsymbol{x}, y, h(\boldsymbol{x}))$ for a new examples $(\boldsymbol{x}, y)$ drawn from $P(\boldsymbol{x}, y)$.
- Either $\mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} L(\boldsymbol{x}, y, h(\boldsymbol{x}))$ or $P\{(\boldsymbol{x}, y) : L(\boldsymbol{x}, y, h(\boldsymbol{x})) > \ell\}$.

Probability of large error p_ℓ

$$p_\ell := P\{(\boldsymbol{x}, y) : L(\boldsymbol{x}, y, h(\boldsymbol{x})) > \ell\}$$

$$S_n = \sum_{i=1}^{n} \mathbb{I}\{L(\boldsymbol{x}_i, y_i, h(\boldsymbol{x}_i)) > \ell\},$$

$$p_\ell \approx \hat{p}_\ell := \frac{1}{n} S_n.$$

We seek an upper confidence bound $\bar{p}_\ell$ on p_ℓ, depending on n, with $P\{p_\ell > \bar{p}_\ell\} < \delta$, confidence parameter δ, e.g. $\delta = 0.01$,

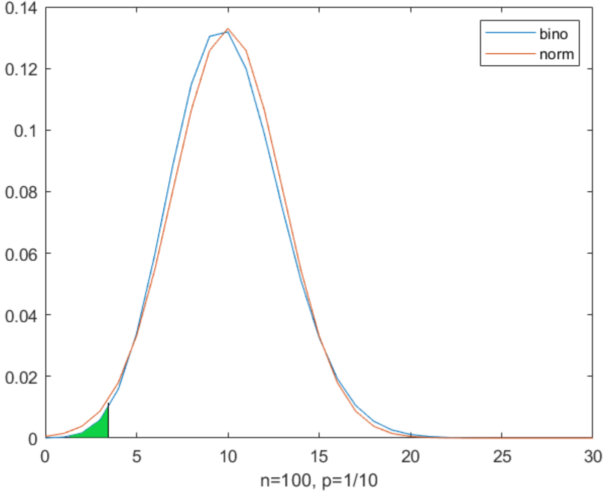
$$\bar{p}_\ell := \hat{p}_\ell + \Delta \ .$$

Confidence bound for S_n

The number of test examples S_n follows a binomial distribution with parameters n and p_ℓ,

$$
\begin{aligned}
&P\left\{S_N = k\right\} = \binom{n}{k} p_\ell^k (1-p_\ell)^{n-k}, \\
&\mathbb{E} S_n = n p_\ell, \\
&\mathbb{V} S_n = n p_\ell (1-p_\ell).
\end{aligned}
$$

Confidence bound for S_n - Figure



Confidence bound for p_ℓ (1)

$$\begin{aligned} P\{p_\ell > \bar{p}_\ell\} &= P\{p_\ell > \hat{p}_\ell + \Delta\} = P\{np_\ell > n\hat{p}_\ell + n\Delta\} \\ &= P\{\mathbb{E}S_n > S_n + n\Delta\} = P\{S_n - \mathbb{E}S_n < -n\Delta\} \\ &= P\left\{\frac{S_n - \mathbb{E}S_n}{\sqrt{np_\ell(1-p_\ell)}} < -\Delta\sqrt{\frac{n}{p_\ell(1-p_\ell)}}\right\} \\ &\approx P\left\{\mathcal{N}_{0,1} < -\Delta\sqrt{\frac{n}{p_\ell(1-p_\ell)}}\right\} \\ &< \delta \end{aligned}$$

if for the $(1-\delta)$-quantile C_δ of the standard normal distribution,

$$C_\delta = \Delta\sqrt{\frac{n}{p_\ell(1-p_\ell)}}.$$

Confidence bound for p_ℓ (2)

$$C_\delta = \Delta\sqrt{\frac{n}{p_\ell(1-p_\ell)}}$$

$$\Delta = C_\delta\sqrt{\frac{p_\ell(1-p_\ell)}{n}}$$

But p_ℓ is unknown, just $p_\ell \leq \hat{p}_\ell + \Delta$. Plugging in and solving for Δ gives

$$\Delta \approx C_\delta\sqrt{\frac{\hat{p}_\ell(1-\hat{p}_\ell)}{n}} + \frac{C_\delta^2}{n}.$$

Upper bound on the mean error $\mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} L(x, y, h(x))$

Impossible without further assumptions:

- Let $y = 0$ for all x but $P(0, B) = 1/B$ and $h(x) = 0$ for all x.
- If $n \ll B$, then it is unlikely that $(0, B)$ is among the test data.
- For the square loss the observed error is 0, but $\mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} L(\boldsymbol{x}, y, h(\boldsymbol{x})) = P(0, B) * (B - 0)^2 = B$.

For **bounded loss**, e.g. $L(\boldsymbol{x}, y, h(\boldsymbol{x})) \in [0, 1]$, we get

$$\mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} L(\boldsymbol{x}, y, h(\boldsymbol{x})) \leq \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{x}_i, y_i, h(\boldsymbol{x}_i)) + \frac{C_\delta}{2\sqrt{n}}$$

with probability δ.