Model Evaluation

Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Agenda

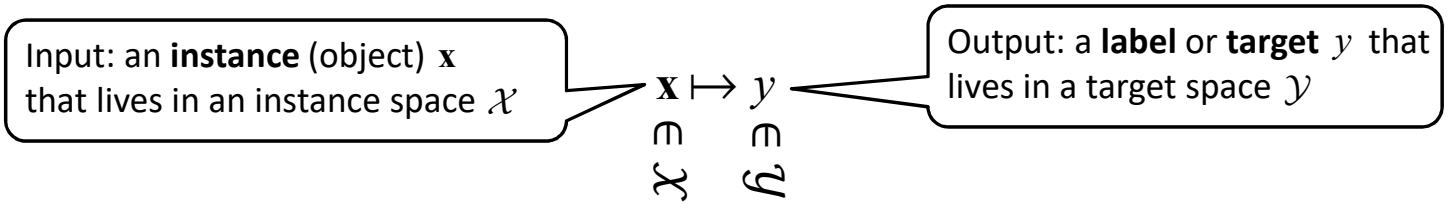
- Error estimators, holdout testing, cross-validation
- Confidence intervals

Agenda

- Error estimators, holdout testing, cross-validation
- Confidence intervals

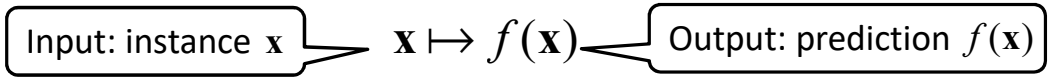
Review: Supervised Learning

- Review: in **supervised learning**, the goal is to make predictions about objects



- To obtain predictions, we are looking for a **model** f that produces a prediction $f(\mathbf{x}) \in \mathcal{Y}$ for an input instance $\mathbf{x}$

$$f : \mathcal{X} \to \mathcal{Y}$$



- Model will be inferred from **training data**: a set of instances with observed targets

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

Review: Assumptions About Data

- For learning to work, we have to assume that there is some reasonably stable relationship between inputs and outputs that can be captured by a model

- Assumption: training example are independently drawn from (constant) joint distribution over inputs and outputs:

$$(\mathbf{x}_n, y_n) \sim p(\mathbf{x}, y)$$

- Because $p(\mathbf{x}, y) = p(\mathbf{x})\, p(y \mid \mathbf{x})$, the assumption can be reformulated as

$$\mathbf{x}_n \sim p(\mathbf{x})$$
$$y_n \sim p(y \mid \mathbf{x}_n)$$

- The instances $\mathbf{x}_n$ are sampled from a probability distribution over instances.
- $p(\mathbf{x})$ describes distribution over population of objects
- For example, certain flowers, digits, or email texts are encountered with a certain probability

- Given an instance $\mathbf{x}_n$, its label is drawn from a distribution $p(y \mid \mathbf{x}_n)$ that represents the relationship between input and output.
- The relationship could be deterministic (probabilities 0 or 1) but this formulation also allows for randomness or noise in data

Review: Error at Application Time

- **The goal of learning is to infer a model that performs well at application time**
  - After training, the model will be deployed in an application domain and has to make predictions for novel instances that have not been part of training data
  - These novel instances are assumed to be drawn from the same distribution as the training data
  - While at application time we only see a new input instance $\mathbf{x}_{new}$, we can image that there is also an unobserved true label y_{new} that indicates what the correct prediction for the instance would have been:

$$(\mathbf{x}_{new}, y_{new}) \sim p(\mathbf{x}, y)$$

New instance at application time

Unobserved true label for the new instance

- A good model will give predictions on new instances that are close to the (unobserved) correct prediction:

$$f_{\boldsymbol{\theta}^*}(\mathbf{x}_{new}) \approx y_{new}$$

Review: Error Measure for Evaluation

- **Question**: once we have collected the training data, implemented the model, and trained it: how well will the model perform on novel data?
- **Model evaluation**: systematically estimate the prediction performance of learned models
- To quantify the performance, we need to define an error measure between predictions $f_{\boldsymbol{\theta}^*}(\mathbf{x}_{new})$ and true labels y_{new}
  - Given by function $\ell_{eval} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
  - Similar in spirit to loss functions used for training, but more freedom in choosing ℓ_{eval} because we do not have to optimize over it (e.g. no need to be differentiable)
  - For classification, most often zero-one-loss

$$\ell_{eval}(y, f_{\boldsymbol{\theta}^*}(\mathbf{x})) = \begin{cases} 0 : y = f_{\boldsymbol{\theta}^*}(\mathbf{x}) \\ 1 : y \neq f_{\boldsymbol{\theta}^*}(\mathbf{x}) \end{cases}$$

  - For regression, often squared or absolute loss:

$$\ell_{eval}(y, f_{\boldsymbol{\theta}^*}(\mathbf{x})) = (y - f_{\boldsymbol{\theta}^*}(\mathbf{x}))^2$$

$$\ell_{eval}(y, f_{\boldsymbol{\theta}^*}(\mathbf{x})) = | y - f_{\boldsymbol{\theta}^*}(\mathbf{x}) |$$

Risk of a Model

- A natural characterization of the predictive performance of a model on novel data is given by the expected error on a randomly drawn instance (new instance here called $(\mathbf{x}, y)$ rather than $(\mathbf{x}_{new}, y_{new})$ to make notation more compact):

$$R(f_{\boldsymbol{\theta}^*}) = \mathbb{E}\Big[\ell_{eval}(y, f_{\boldsymbol{\theta}^*}(\mathbf{x}))\Big]$$
$$= \int\int \ell_{eval}(y, f_{\boldsymbol{\theta}^*}(\mathbf{x}))\, p(\mathbf{x}, y) d\mathbf{x} dy$$

- Here, the expectation is over the process of drawing a novel instance from the joint distribution, $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$
- The quantity $R(f_{\boldsymbol{\theta}^*})$ is called the (true) **risk** of the model
- Of course, the true risk of a learned model $f_{\boldsymbol{\theta}^*}$ cannot be computed, because the joint distribution $p(\mathbf{x}, y)$ is not known
- We can only indirectly observe $p(\mathbf{x}, y)$ through data that we collect
- Therefore, we can only empirically estimate the risk of a model based on data

Excursion: Estimators

- We want to estimate the risk of a model from data
- Formally, an **estimator** is a method that maps observations to an estimate for an unknown quantity

- Example: coin tosses
  - We are tossing a coin N times, and observe N_h heads and N_t tails
  - There is an unknown true probability that the coin will show heads in a random toss, which we call $\theta \in \mathbb{R}$. The coin is not necessarily fair, so in general $\theta \neq 0.5$
  - We can estimate the unknown probability, for example by the estimator

$$\hat{\theta} = \frac{N_h}{N}$$

Maximum likelihood estimator for Bernoulli/Binomial distribution (more later)

- We generally denote an estimator for an unknown quantity θ by $\hat{\theta}$

Estimating Risk From Data

- To estimate the risk of a model from data, we can use a set of instances drawn from the joint distribution $p(\mathbf{x}, y)$:

$$\mathcal{T} = \{(\overline{\mathbf{x}}_1, \overline{y}_1), ..., (\overline{\mathbf{x}}_{\overline{N}}, \overline{y}_{\overline{N}})\} \qquad (\overline{\mathbf{x}}_n, \overline{y}_n) \sim p(\mathbf{x}, y)$$

- As the risk is an expectation over the distribution $p(\mathbf{x}, y)$,

$$R(f_{\boldsymbol{\theta}^*}) = \int\int \ell_{eval}(y, f_{\boldsymbol{\theta}}(\mathbf{x}))\, p(\mathbf{x}, y) d\mathbf{x} dy$$

  the risk can be estimated by the average

$$\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) = \frac{1}{\overline{N}} \sum_{n=1}^{\overline{N}} \ell_{eval}(\overline{y}_n, f_{\boldsymbol{\theta}}(\overline{\mathbf{x}}_n))$$

- Important question: where does $\mathcal{T}$ come from?
  - Reuse training data, $\mathcal{T} = \mathcal{D}$? Bad idea!
  - Reserve part of the overall data for testing (holdout testing, see below)
  - Cross-validation (see below)

Expectation of Risk Estimator

- The risk estimator is

$$\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) = \frac{1}{\bar{N}} \sum_{n=1}^{\bar{N}} \ell_{eval}(\bar{y}_n, f_{\boldsymbol{\theta}}(\bar{\mathbf{x}}_n))$$

- The quantity $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ can be considered a random variable:
  - The instances $(\bar{\mathbf{x}}_n, \bar{y}_n) \in \mathcal{T}$ have been drawn from the joint distribution,

$$(\bar{\mathbf{x}}_n, \bar{y}_n) \sim p(\mathbf{x}, y)$$

  - The quantity $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ depends on the $(\bar{\mathbf{x}}_n, \bar{y}_n)$, and is therefore the results of a probabilistic process

- The estimator therefore has an expectation

$$\mathbb{E}\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})\right]$$

which is defined by integrating over all possible draws of instances in the set $\mathcal{T}$

Bias of Risk Estimator

- The **bias** of the risk estimator is defined as

$$\mathbb{E}\left[\hat{R}_{\mathcal{I}}(f_{\boldsymbol{\theta}^*})\right] - R(f_{\boldsymbol{\theta}^*})$$

- The estimator is called **optimistic** if the bias is negative, that is, if

$$\mathbb{E}\left[\hat{R}_{\mathcal{I}}(f_{\boldsymbol{\theta}^*})\right] < R(f_{\boldsymbol{\theta}^*})$$

- The estimator is called **pessimistic** if the bias is positive, that is, if

$$\mathbb{E}\left[\hat{R}_{\mathcal{I}}(f_{\boldsymbol{\theta}^*})\right] > R(f_{\boldsymbol{\theta}^*})$$

- The estimator is called **unbiased** if the bias is zero, that is, if

$$\mathbb{E}\left[\hat{R}_{\mathcal{I}}(f_{\boldsymbol{\theta}^*})\right] = R(f_{\boldsymbol{\theta}^*})$$

Variance of Risk Estimator

- Review: the variance of a random variable X is given by

$$Var[X] = \mathbb{E}\left[\left(X - \mathbb{E}[X]\right)^2\right]$$

- Equivalently, the variance can be defined as

$$Var[X] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2$$

Proof:

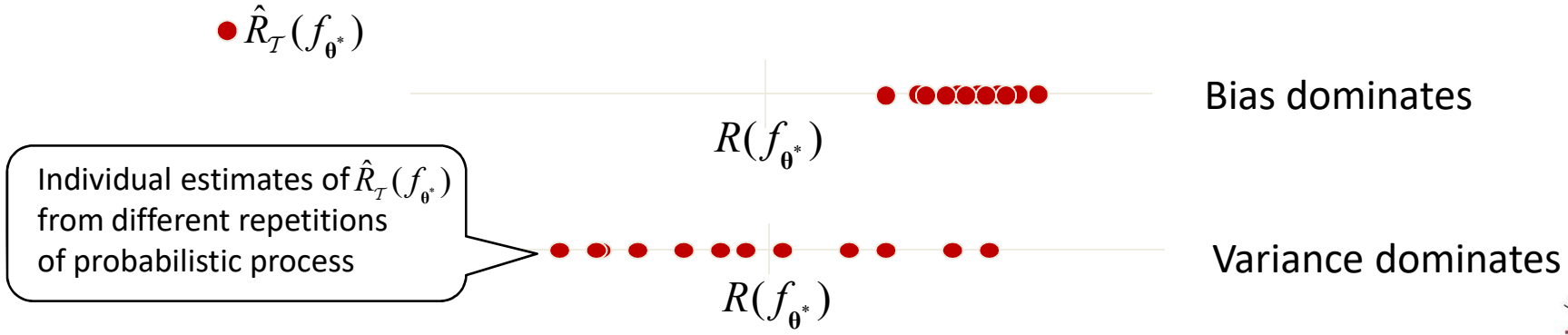
$$\begin{aligned}
Var[X] &= \mathbb{E}\left[\left(X - \mathbb{E}[X]\right)^2\right] && \text{definition of variance} \\
&= \mathbb{E}\left[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2\right] && \text{multiplying out} \\
&= \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[X]^2 && \text{linearity of expectation} \\
&= \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X]^2 + \mathbb{E}[X]^2 \\
&= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2
\end{aligned}$$

Variance of Risk Estimator

- The risk estimator $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ also has a variance

$$Var\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})\right] = \mathbb{E}\left[\left(\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) - \mathbb{E}\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})\right]\right)^2\right]$$
$$= \mathbb{E}\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})^2\right] - \mathbb{E}\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})\right]^2$$

- The larger the number $\overline{N}$ of instances in the data set used for estimation, the smaller the variance of the estimator
- **Variance versus bias**:
  - High variance: large random deviations when estimating the empirical risk
  - High bias: large systematic error when estimating the empirical risk



Expected Error of Estimator: Bias-Variance Decomposition

- We would like the risk estimate $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ to be close to the true risk $R(f_{\boldsymbol{\theta}^*})$

- The expected (quadratic) deviation of an estimate $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ from the true risk $R(f_{\boldsymbol{\theta}^*})$ is given by

$$\mathbb{E}\left[(\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) - R(f_{\boldsymbol{\theta}^*}))^2\right]$$

Over many iterations of the probabilistic process, what is the average quadratic deviation of the estimate from the true risk?

- It can be shown that the expected quadratic deviation can be decomposed into the variance plus the squared bias:

$$\mathbb{E}\left[(\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) - R(f_{\boldsymbol{\theta}^*}))^2\right] = Var\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})\right] + \underbrace{\mathbb{E}\left[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) - R(f_{\boldsymbol{\theta}^*})\right]^2}_{\textbf{bias}}$$

- This is called the **bias-variance decomposition** of the error
- Therefore, would ideally like a risk estimator with low bias and low variance

Risk Estimate on the Training Data

- Back to the question: which data do we use for the set $\mathcal{T} = \{(\overline{\mathbf{x}}_1, \overline{y}_1), ..., (\overline{\mathbf{x}}_{\overline{N}}, \overline{y}_{\overline{N}})\}$?
- First attempt: training data, $\mathcal{T} = \mathcal{D}$
  - Model $f_{\boldsymbol{\theta}^*}$, trained on data $\mathcal{D}$
  - Risk of model $f_{\boldsymbol{\theta}^*}$ estimated by

$$\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}^*}) = \frac{1}{N} \sum_{n=1}^{N} \ell_{eval}(y_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n))$$

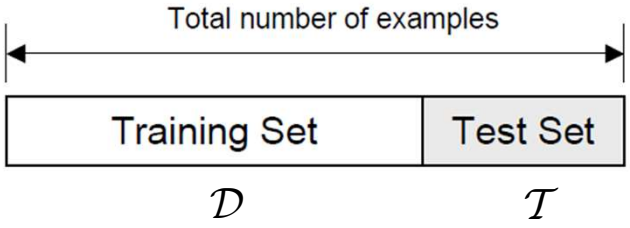
- **Problem: this empirical risk estimate is a strongly optimistic estimator of the risk** $R(f_{\boldsymbol{\theta}^*})$
- Let's look, for a fixed training set $\mathcal{D}$, at all possible models in the model space (for parameterized models: the set $\{f_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \mathbb{R}^D\}$)
  - For some models, it will hold that $\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) < R(f_{\boldsymbol{\theta}})$, for other models, it will hold that $\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) > R(f_{\boldsymbol{\theta}})$
  - The learning algorithm typically picks a model with low $\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}})$: it looks for a model whose predictions match the training labels
  - Likely that it picks one of those models for which $\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) < R(f_{\boldsymbol{\theta}})$
  - Therefore, in expectation, $\hat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) < R(f_{\boldsymbol{\theta}})$

Holdout Testing

- The reason why the risk estimate on the training data is optimistic is the dependency between the model and the training data
- This problem can be solved by using independent test data for evaluating a model
- **Holdout Testing**:
  - Assume we overall have a data set $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_L, y_L)\}$ available
  - Split the data set into two disjunct sets:

For training: $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

For evaluation: $\mathcal{T} = \{(\mathbf{x}_{N+1}, y_{N+1}), \ldots, (\mathbf{x}_L, y_L)\}$



  - Train a model $f_{\boldsymbol{\theta}^{\mathcal{D}}}$ on the data set $\mathcal{D}$
  - Estimate the risk of model $f_{\boldsymbol{\theta}^{\mathcal{D}}}$ on test set $\mathcal{T}$: $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}}) = \frac{1}{L-N} \sum_{n=N+1}^{L} \ell_{eval}(y_n, f_{\boldsymbol{\theta}^{\mathcal{D}}}(\mathbf{x}_n))$
  - Finally, train another model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ on all data $\mathcal{L}$
  - Return the model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ (for deployment) and use $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$ as its risk estimate

Analysis of Holdout Testing

- Holdout testing: at the end, why do we retrain the model on all of the available training data, $\mathcal{L}$?
  - The more data a model is trained on, the better the model (generally)
  - We want the best model possible: no reason not to use all the data for training
- The holdout testing procedure returns the model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ as the final model together with the risk estimate $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$
- Is $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$ an unbiased, optimistic or pessimistic estimate of the performance of the final model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$?
- **The estimator $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$ is pessimistic**:
  - Because the test data $\mathcal{T}$ is independent of the training data $\mathcal{D}$, $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$ is an unbiased estimator of $R(f_{\boldsymbol{\theta}^{\mathcal{D}}})$
  - However, the model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ has been trained on more data and therefore has lower true risk than $f_{\boldsymbol{\theta}^{\mathcal{D}}}$, therefore the estimator is pessimistic for $f_{\boldsymbol{\theta}^{\mathcal{L}}}$
  - But in contrast to estimating performance on training data, the estimate is useful: estimation on training data can be arbitrarily optimistic, while this estimator is only slightly conservative

How Large Should Holdout Set Be?

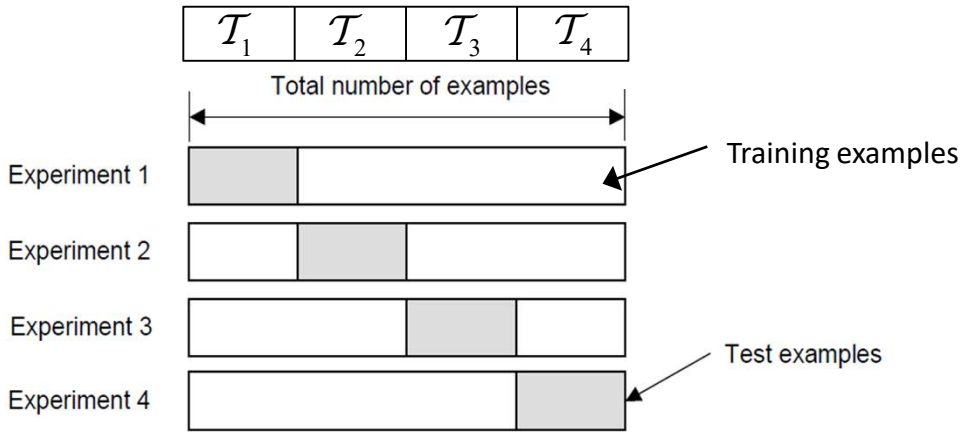
- What are the advantages/disadvantages if the holdout set $\mathcal{T}$ is chosen small/large?
  - In order to reduce the variance of the error estimate $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^{\mathcal{D}}})$, we would like to choose a large holdout set
  - In order to reduce the bias of the estimator, we would like to choose a small holdout set (because then the difference between models $f_{\boldsymbol{\theta}^{\mathcal{D}}}$ and $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ will be smaller)

- Due to this trade-off, hold out testing needs a large initial data set $\mathcal{L}$ to obtain good estimates: need to set aside a reasonable large $\mathcal{T}$ to control variance, but $\mathcal{T}$ still needs to be small relative to $\mathcal{D}$ in order to control bias of estimator

- Alternative if size of original data set $\mathcal{L}$ is limited: **cross validation** (see below)

Cross-Validation

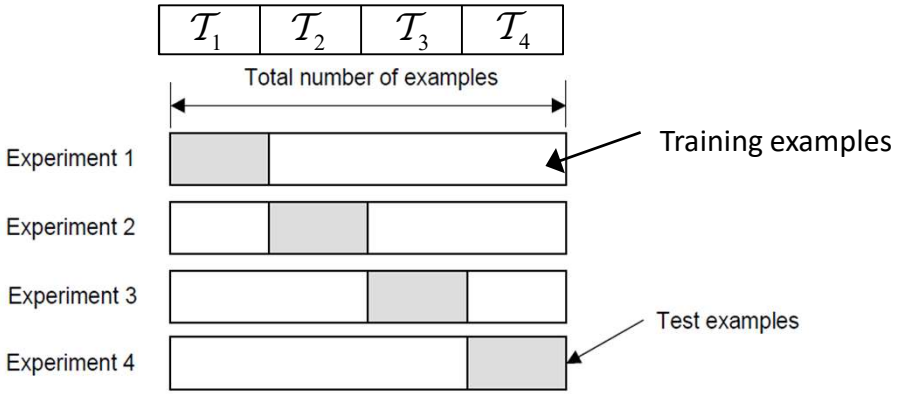
- ***K*-fold cross-validation**: repeatedly split data into training and evaluation parts
  - Assume we overall have a data set $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_L, y_L)\}$ available
  - Split the data set into *K* equally sized disjunct sets $\mathcal{T}_1, \ldots, \mathcal{T}_K$
  - For each $k \in \{1, \ldots, K\}$:
    - train a model $f_{\boldsymbol{\theta}_k}$ on $\mathcal{D}_k := \mathcal{L} \setminus \mathcal{T}_k$
    - evaluate the model on $\mathcal{T}_k$, resulting in risk estimate $\hat{R}_{\mathcal{T}_k}(f_{\boldsymbol{\theta}^k})$
  - Train a final model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ on all data $\mathcal{L}$
  - Return model $f_{\boldsymbol{\theta}^{\mathcal{L}}}$ together with risk estimate $\hat{R}_{avg} = \frac{1}{K} \sum_{k=1}^{K} \hat{R}_{\mathcal{T}_k}(f_{\boldsymbol{\theta}^k})$

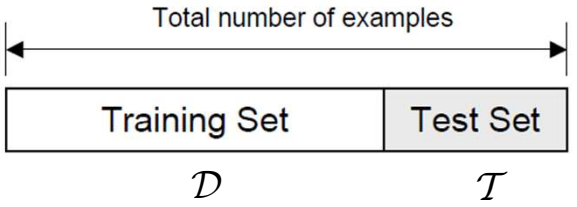


Analysis of Cross-Validation

- Is the estimator $\hat{R}_{avg}$ unbiased, optimistic or pessimistic for the model $f_{\theta^{\mathcal{L}}}$?
- The estimator is pessimistic, for the same reasons as for holdout testing:
  - Estimators $\hat{R}_{\mathcal{T}_k}(f_{\theta^k})$ are unbiased for models f_{θ_k}
  - But model $f_{\theta^{\mathcal{L}}}$ has been trained on (slightly) more data, so it will have lower risk

- An advantage of cross-validation is that the variance of the estimator $\hat{R}_{avg}$ is lower, because it averages over all test sets $\mathcal{T}_k$
- At the same time, bias is not too strong if *K* is large enough: models f_{θ_k} are trained on (K-1)/K of all the data

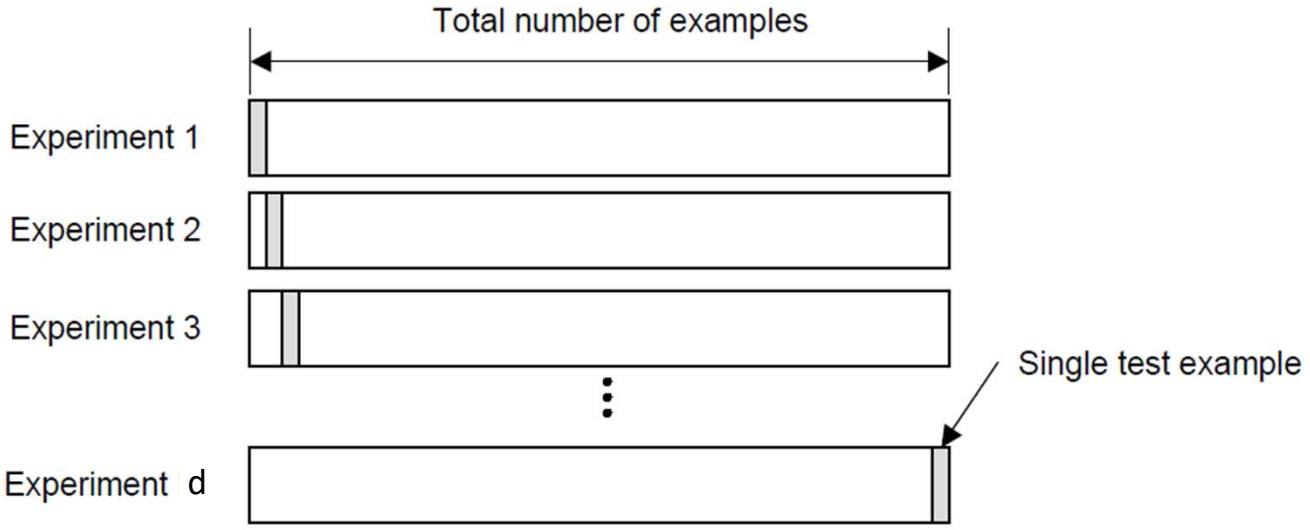




Analysis of Cross-Validation

- How should the parameter *K* be chosen?
  - to minimize bias, *K* should be relatively large
  - however, large *K* are computationally challenging: need to train many models
  - practical choices often *K=5* or *K=10*

- Can also choose *K=L*, this is called leave-one-out cross validation:



Applications of Model Evaluation

- Application of model evaluation: when do we use methods such as holdout testing or cross-validation?
  1. Often important from an application perspective: before deploying a model, need to know how well the model will work
  2. Sometimes different models are trained (e.g. different learning algorithms), and model evaluation is used to select one of the models which works best
  3. A special case of Point 2 is hyperparameter tuning: there is a hyperparameter such as a regularization weight that needs to be set. We train and evaluate models with different hyperparameters

Hyperparameters of Models

- **Hyperparameters**: Most learning algorithms have parameters that need to be set before learning can start, for example the learning rate in gradient descent or the regularization weight for models using regularization

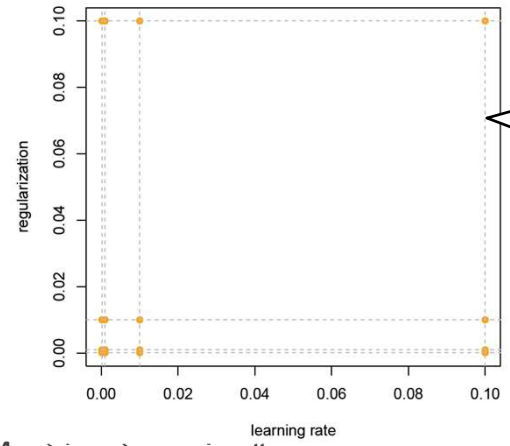
- The hyperparameters $\boldsymbol{\lambda} \in \mathbb{R}^A$ parameterize the corresponding learning algorithm $\mathcal{A}_{\boldsymbol{\lambda}}$
- To obtain a final model, we need to
  1. Identify good hyperparameters $\boldsymbol{\lambda}^*$
  2. Train the final model using the learning algorithm $\mathcal{A}_{\boldsymbol{\lambda}^*}$ with hyperparameters $\boldsymbol{\lambda}^*$

- Good hyperparameters should also be picked based on data: essentially, train and evaluate models with different sets of hyperparameters using holdout testing or cross-validation

Hyperparameter Tuning

- **Hyperparameter tuning**: Given a learning algorithm $\mathcal{A}_{\boldsymbol{\lambda}}$ parameterized with hyperparameters $\boldsymbol{\lambda} \in \mathbb{R}^A$, and overall data set $\mathcal{L} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_L, y_L)\}$
  - Choose a set of candidate hyperparameter values $\{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_T\} \in \mathbb{R}^A$
  - For $t \in \{1, ..., T\}$: obtain error estimate $\hat{R}_t$ for model trained with hyperparameters $\boldsymbol{\lambda}_t$ through holdout testing or cross-validation on $\mathcal{L}$ (using the same split into training and test data for each run)
  - Choose best hyperparameters $\boldsymbol{\lambda}_{t^*}$ where $t^* = \arg\min_t \hat{R}_t$

- How to choose the set of candidate values for hyperparameters?
- One frequently used approach is grid search:

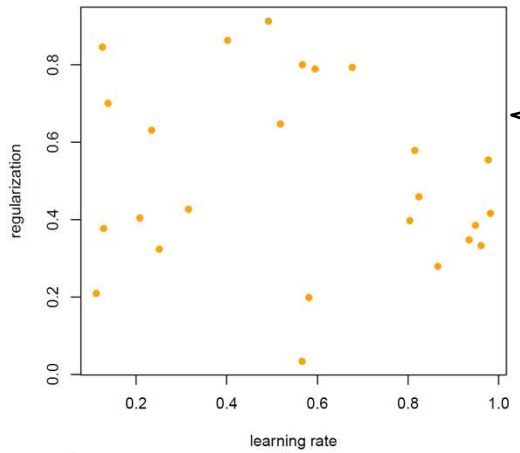


Grid search:
- Choose a set of values for each hyperparameter in the vector of hyperparameters $\boldsymbol{\lambda} \in \mathbb{R}^A$
- Candidates $\{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_T\}$ are all combinations of candidate values for the individual hyperparameters
- Note that the number of candidates T is the product of the number of values for the individual hyperparameters, so generally exponential in the number of hyperparameters A

Hyperparameter Tuning

- **Hyperparameter tuning**: Given a learning algorithm $\mathcal{A}_{\boldsymbol{\lambda}}$ parameterized with hyperparameters $\boldsymbol{\lambda} \in \mathbb{R}^A$, and overall data set $\mathcal{L} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_L, y_L)\}$
  - Choose a set of candidate hyperparameter values $\{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_T\} \in \mathbb{R}^A$
  - For $t \in \{1, ..., T\}$: obtain error estimate $\hat{R}_t$ for model trained with hyperparameters $\boldsymbol{\lambda}_t$ through holdout testing or cross-validation on $\mathcal{L}$ (using the same split into training and test data for each run)
  - Choose best hyperparameters $\boldsymbol{\lambda}_{t^*}$ where $t^* = \arg\min_t \hat{R}_t$

- How to choose the set of candidate values for hyperparameters?
- Another frequently used approach is random search:



Random search:
- Choose a range of values for each hyperparameter in the vector of hyperparameters $\boldsymbol{\lambda} \in \mathbb{R}^A$
- Candidates $\boldsymbol{\lambda}_t$ are randomly constructed by drawing each individual hyperparameter uniformly from its range
- Number of candidates T can be arbitrarily chosen
- Usually slightly better results than grid search

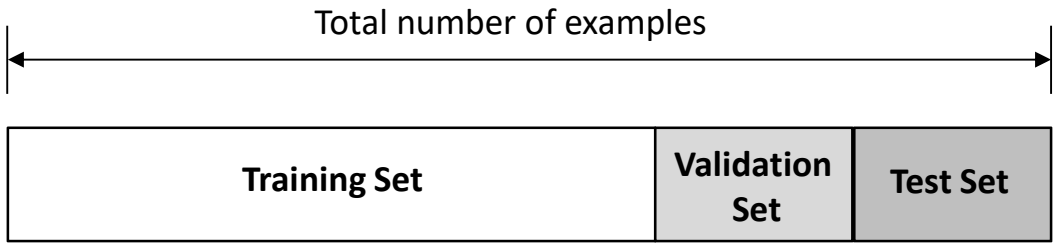
Hyperparameter Tuning and Final Error Estimates

- Hyperparameter tuning will yield an (approximately) optimal vector of hyperparameters $\boldsymbol{\lambda}_{t^*} \in \mathbb{R}$

- We then train a final model on the entire data $\mathcal{L}$ with the best hyperparameters
- During hyperparameter tuning, we have obtained an error estimate $\hat{R}_{t^*}$ for the model with best hyperparameters $\boldsymbol{\lambda}_{t^*}$
- However, we should not use $\hat{R}_{t^*}$ to estimate the error rate of our final model on novel instances at application time:
  - We have tried out many different hyperparameter settings $\boldsymbol{\lambda}_t \in \{\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_T\}$ and then selected the best one (based on the test data)
  - This is essentially like a training procedure: we have „trained“ the hyperparameter $\boldsymbol{\lambda}$ on the test data
  - Therefore, error estimate will be optimistic

Training, Validation, Test

- If we want to both tune hyperparameters and obtain a realistic estimate of the error on novel instances at application time, need to further split the data:
  - Split overall data $\mathcal{L}$ into a training set $\mathcal{D}$, a validation set $\mathcal{V}$, and a test set $\mathcal{T}$
  - Train candidate models (for different hyperparameters) on the training set, select the best hyperparameter combination on the validation set
  - Train model with the best hyperparameters on $\mathcal{D} \cup \mathcal{V}$
  - Evaluate this model on $\mathcal{T}$ to obtain error estimate $\hat{R}$
  - Train a final model on all data $\mathcal{L}$
  - Deploy this final model together with error estimate $\hat{R}$



- The same idea can also be used within cross-validation: in each cross-validation iteration, use one subset for testing, one for validation, and the rest for training

Agenda

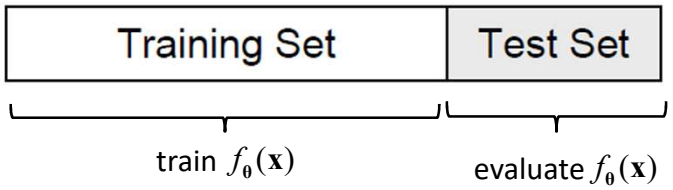
- Error estimators, holdout testing, cross-validation
- Confidence intervals

Estimating Remaining Uncertainty

- By methods such as holdout testing and cross-validation, we can get an estimate of the risk of a model on novel data at application time
- Because these estimates are based on finite samples of data, they will never be perfectly accurate
- We now turn towards the problem of characterizing the uncertainty in the risk estimates: how confident are we that the risk estimates are correct?
- For the remainder of the lecture, we assume that the problem is binary classification, and that the error measure under study is zero-one classification error:

$$\ell_{eval}(y, f_\theta(\mathbf{x})) = \begin{cases} 0 : y = f_\theta(\mathbf{x}) \\ 1 : \text{ otherwise} \end{cases}$$

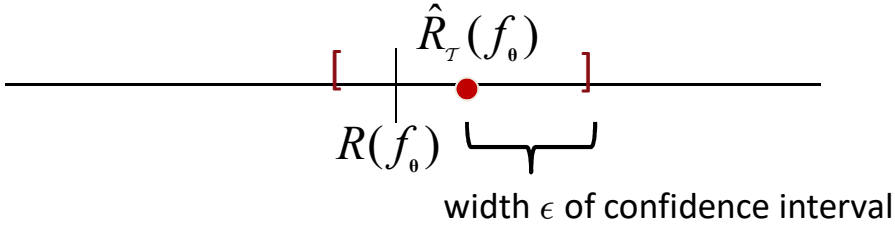
- We also assume that the model is evaluated on independent test data, such that the error estimate itself is unbiased:



Idea: Confidence Interval

- **Idea**: characterize the uncertainty of a risk estimate by means of a **confidence interval**
  - Risk estimate $\hat{R}_{\mathcal{T}}(f_\theta)$ is based on sample $\mathcal{T}$, and therefore a random variable
  - Specify interval around the risk estimate such that the true risk lies within the interval „most of the time“
  - This quantifies the uncertainty of the risk estimate

Visualization of confidence interval:



- Route to confidence interval: analyze the distribution of the random variable $\hat{R}_{\mathcal{T}}(f_\theta)$

Central Limit Theorem

- **Central Limit Theorem**: Let $z_1, \ldots, z_N$ be independent draws from a distribution $p(z)$ with $\mathbb{E}[z] = \mu$ and $Var[z] = \sigma^2$. Then it holds that, for $N \to \infty$,

$$\sqrt{N}\left(\frac{1}{N}\sum_{n=1}^{N} z_n - \mu\right) \to \mathcal{N}(0, \sigma^2)$$

Intuition: The average of the z_n converges to the mean μ, thus the average minus μ converges to zero

Intuition: with increasing N, the variance also goes to zero, but multiplying with $\sqrt{N}$ cancels this effect, thus the variance approaches the variance σ^2 of the original variable

- Convergence is „in distribution“, which means that the cumulative distribution functions converge pointwise (not important for our considerations)
- Central limit theorem tells us (approximately, for large *N)*:

$$\sqrt{N}\left(\frac{1}{N}\sum_{n=1}^{N} z_n - \mu\right) \sim \mathcal{N}(0, \sigma^2)$$

divide by $\sqrt{N}$, add μ

$$\Rightarrow \frac{1}{N}\sum_{n=1}^{N} z_n \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right)$$

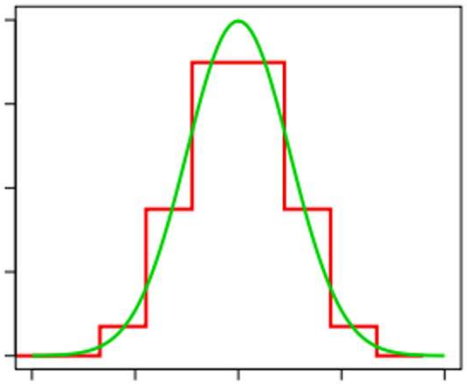
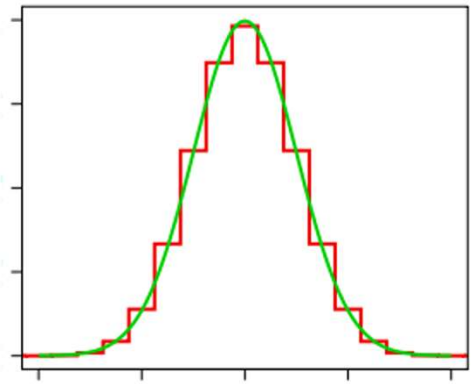
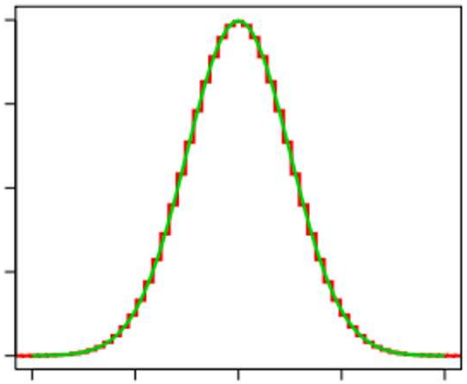
Scaling normally distributed random variable by $1/\sqrt{N}$ scales variance by $1/N$
Adding μ changes mean to μ

Example Central Limit Theorem

- **Example central limit theorem**: **average of Bernoulli variables**
- Let $z_1, \ldots, z_N$ be independent draws from a Bernoulli distribution, that is

$$z_n \sim \text{Bern}(z_n \mid \theta)$$

- Average $\frac{1}{N}\sum_{n=1}^{N} z_n$ follows (rescaled) Binomial distribution
- Binomial distribution approaches Normal distribution

Central Limit Theorem: Error Estimator

- We now apply the central limit theorem to the error estimator
- Notation: let the holdout set be $\mathcal{T} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ for notational convenience (but of course a sample that is independent from training set)
- Error estimator is

$$\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) = \frac{1}{N} \sum_{n=1}^{N} \ell_{eval}(y_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n))$$

- Because we assumed

$$\ell_{eval}(y, f_{\boldsymbol{\theta}}(\mathbf{x})) = \begin{cases} 0 : y = f_{\boldsymbol{\theta}}(\mathbf{x}) \\ 1 : \text{ otherwise} \end{cases}$$

  the quantity $\ell_{eval}(y, f_{\boldsymbol{\theta}}(\mathbf{x}))$ is a Bernoulli-distributed variable
- Because the error estimate is unbiased, the expectation of $\ell_{eval}(y, f_{\boldsymbol{\theta}}(\mathbf{x}))$ is the true risk:

$$\mathbb{E}[\ell_{eval}(y_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n))] = R(f_{\boldsymbol{\theta}})$$

- Because $\ell_{eval}(y_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n))$ is a Bernoulli variable, its variance is given by

$$Var[\ell_{eval}(y_n, f_{\boldsymbol{\theta}}(\mathbf{x}_n))] = R(f_{\boldsymbol{\theta}})(1 - R(f_{\boldsymbol{\theta}}))$$

Central Limit Theorem: Error Estimator

- We can apply the central limit theorem: $\ell_{eval}(y_1, f_\theta(\mathbf{x}_1)),\ldots,\ell_{eval}(y_N, f_\theta(\mathbf{x}_N))$ are *N* independent draws from a Bernoulli distribution
- The error estimator $\hat{R}_T(f_\theta)$ is the average over these draws
- Expectation and variance of an individual draw are given by

$$\mathbb{E}[\ell(y_n, f_\theta(\mathbf{x}_n))] = R(f_\theta) \qquad Var[\ell_{eval}(y_n, f_\theta(\mathbf{x}_n))] = R(f_\theta)(1 - R(f_\theta))$$

- Central limit theorem says:

$$\hat{R}_T(f_\theta) \sim \mathcal{N}(R(f_\theta), \frac{R(f_\theta)(1 - R(f_\theta))}{N}) \qquad \text{(approximately, large enough } N\text{)}$$

- First result for distribution of error estimator $\hat{R}_T(f_\theta)$, but depends on unknown true risk $R(f_\theta)$. It tells us that
  - estimator is unbiased: mean is the true risk $R(f_\theta)$
  - variance of estimator is approximately

$$\sigma^2_{\hat{R}_T} := \frac{R(f_\theta)(1 - R(f_\theta))}{N}$$

Distribution of Error Estimator

- Distribution of error estimator, plugging in $\sigma^2_{\hat{R}_T}$ (approximately, for large *N*):

$$\hat{R}_T(f_\theta) \sim \mathcal{N}(R(f_\theta), \sigma^2_{\hat{R}_T})$$

$$\frac{\hat{R}_T(f_\theta) - R(f_\theta)}{\sigma_{\hat{R}_T}} \sim \mathcal{N}(0,1)$$

Substracting the mean $R(f_\theta)$ and dividing by standard deviation $\sigma^2_{\hat{R}_T}$ yields standard normal distribution

- Expression still contains $R(f_\theta)$ in enumerator, but that is ok (see below)
- A problem is that $\sigma_{\hat{R}_T}$ depends on $R(f_\theta)$ and therefore cannot be computed
- To solve this problem, we replace the variance

$$\sigma^2_{\hat{R}_T} := \frac{R(f_\theta)(1 - R(f_\theta))}{N}$$

with a variance estimate

$$s^2_{\hat{R}_T} := \frac{\hat{R}_T(f_\theta)(1 - \hat{R}_T(f_\theta))}{N}$$

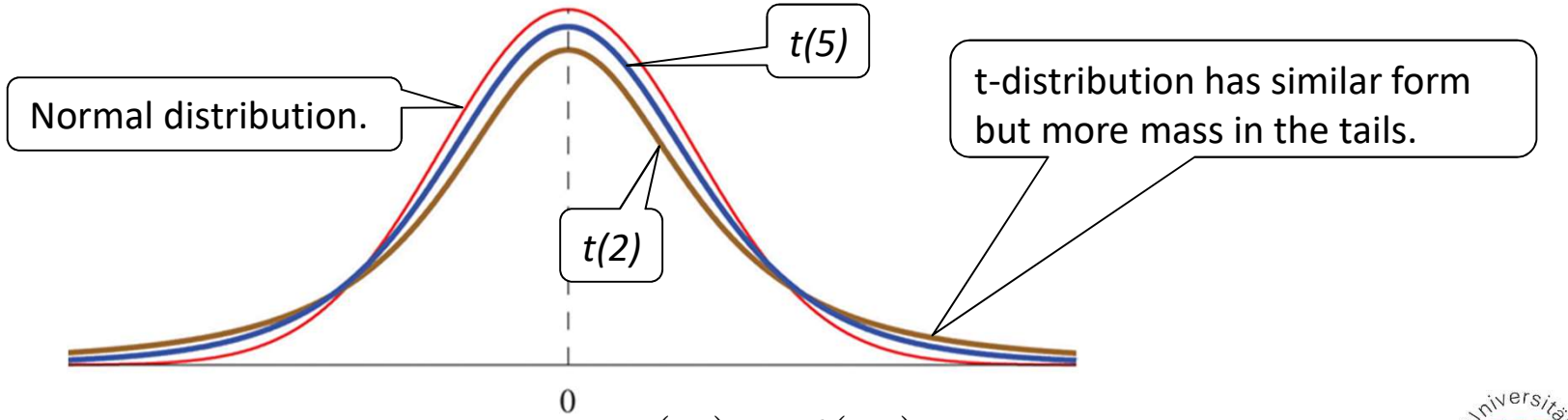
Distribution of Error Estimator

- When variance $\sigma^2_{\hat{R}_T}$ is replaced by variance estimate $s^2_{\hat{R}_T}$, the normal distribution becomes a Student's t-distribution (no proof):

$$\frac{\hat{R}_T(f_\theta) - R(f_\theta)}{s_{\hat{R}_S}} \sim t(N)$$

Student's t-distribution with *N* degrees of freedom

- However, for large *N,* the Student's t-distribution becomes a normal distribution again, so we can keep working with the normal as an approximation:



Convergence: $\lim_{N\to\infty} t(N) = \mathcal{N}(0,1)$

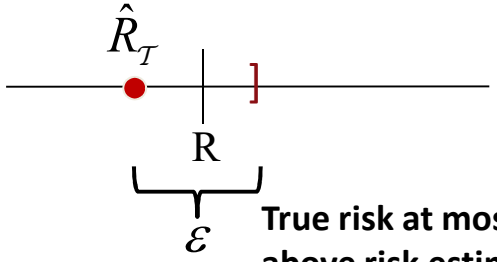
Bound For True Risk

- So what does the empirical risk $\hat{R}_{\mathcal{T}}(f_{\theta})$ tell us about the true risk $R(f_{\theta})$?
  - From empirical risk $\hat{R}_{\mathcal{T}}(f_{\theta})$ compute empirical variance $s^2_{\hat{R}_{\mathcal{T}}}$
  - We can then derive a one-sided upper bound for true risk: probability that true risk is at most ε above estimated risk

$$p(R(f_{\theta}) \leq \hat{R}_{\mathcal{T}}(f_{\theta}) + \varepsilon) = p(R(f_{\theta}) - \hat{R}_{\mathcal{T}}(f_{\theta}) \leq \varepsilon)$$

$$= p\left(\frac{R(f_{\theta}) - \hat{R}_{\mathcal{T}}(f_{\theta})}{s_{\hat{R}_{\mathcal{T}}}} \leq \frac{\varepsilon}{s_{\hat{R}_{\mathcal{T}}}}\right)$$

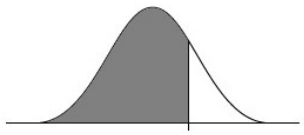
$$\frac{\hat{R}_{\mathcal{T}}(f_{\theta}) - R(f_{\theta})}{s_{\hat{R}_{\mathcal{T}}}} \sim \mathcal{N}(0,1) \approx \Phi\left(\frac{\varepsilon}{s_{\hat{R}_{\mathcal{T}}}}\right)$$



True risk at most ε above risk estimate?

  - Here,

$$\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(x \mid 0,1)dx$$

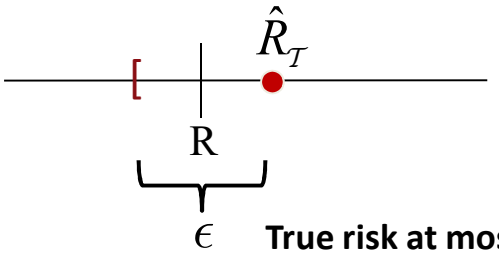


is the cumulative distribution function of the standard normal distribution

Symmetric Bound For True Risk

- On last slide, we derived (probabilistic) upper bound for true risk given estimated risk
- We can also derive lower bound for true risk: Because the distribution of $\hat{R}_T(f_\theta)$ is symmetric around $R(f_\theta)$, the probability that the true risk is at most ϵ below estimated risk is the same as for the upper bound:

$$p(R(f_\theta) \geq \hat{R}_T(f_\theta) - \varepsilon) \approx \Phi\left(\frac{\varepsilon}{s_{\hat{R}_T}}\right)$$



True risk at most ε below risk estimate?

- Two-sided interval: what is the probability that the true risk is at most ϵ away from the estimated risk?

$$p(|R(f_\theta) - \hat{R}_T(f_\theta)| \leq \varepsilon) = 1 - \overbrace{p(R(f_\theta) - \hat{R}_T(f_\theta) > \varepsilon)}^{\text{above interval}} - \overbrace{p(\hat{R}_T(f_\theta) - R(f_\theta) > \varepsilon)}^{\text{below interval}}$$

$$\approx 1 - 2\left(1 - \Phi\left(\frac{\varepsilon}{s_{\hat{R}_T}}\right)\right)$$

Size of Interval

- So far, we have computed probability that a bound holds for a particular interval size ε
- Idea: choose ε in such a way that bounds hold with a certain pre-specified probability $1-\delta$ (e.g. $\delta = 0.05$)
- One-sided $1-\delta$-confidence interval: bound ε such that

$$p(R(f_{\theta}) \leq \hat{R}_{T} + \varepsilon) = 1 - \delta$$

- Two-sided $1-\delta$-confidence interval: bound ε such that

$$p(|R - \hat{R}_{T}(f_{\theta})| \leq \varepsilon) = 1 - \delta$$

- For symmetric distributions (here: normal distribution) it always holds that:
  - ε for one-sided $1-\delta$-interval = ε for two-sided $1-2\delta$ interval.
  - ε for one-sided 95%-interval = ε for two-sided 90% interval.
  - Thus, it suffices to derive ε for one-sided interval.

Size of Interval

- Compute one-sided 1-δ-confidence interval: Determine ε such that bound holds with probability 1-δ

$$p(R(f_\theta) \leq \hat{R}_T(f_\theta) + \varepsilon) = 1 - \delta$$

Approximately, according to result from above

$$\Leftrightarrow \Phi\left(\frac{\varepsilon}{s_{\hat{R}_T}}\right) = 1 - \delta$$

Apply inverse $\Phi^{-1}(z)$ of cumulative distribution function of standard normal distribution to both sides of equation

$$\Leftrightarrow \frac{\varepsilon}{s_{\hat{R}_T}} = \Phi^{-1}(1-\delta)$$

$$\Leftrightarrow \varepsilon = s_{\hat{R}_T} \Phi^{-1}(1-\delta)$$

- If we choose the side of the one-sided confidence interval as $\varepsilon = s_{\hat{R}_T} \Phi^{-1}(1-\delta)$, it will hold with probability 1-δ

- Two-sided confidence interval $[\hat{R}_T(f_\theta) - \varepsilon, \hat{R}_T(f_\theta) + \varepsilon]$ will hold with probability 1-2δ

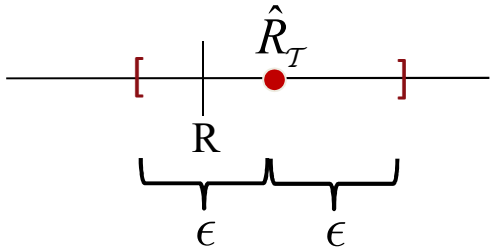
Example Confidence Interval

- **Example for computation of confidence interval**:
  - We have observed an empirical risk of $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}}) = 0.08$ on $N = 100$ test instances
  - Compute the empirical standard deviation:

$$s_{\hat{R}_{\mathcal{T}}} = \sqrt{\frac{0.08 \cdot 0.92}{100}} \approx 0.027$$

  - Choosing $\delta = 0.05$ (one-sided confidence level of $1-\delta = 0.95$, two-sided confidence level will be $1-2\delta = 0.9$)
  - Compute $\varepsilon = s_{\hat{R}_{\mathcal{T}}} \Phi^{-1}(1-\delta) \approx 0.027 \cdot 1.645 \approx 0.045.$

- The confidence interval $[\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}}) - \epsilon, \hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}}) + \epsilon]$ contains the true risk in 90% of the cases.



Interpretation of Confidence Intervals

- Care should be used when interpreting confidence intervals: the random variable is the empirical risk $\hat{R}_T(f_\theta)$ and the resulting interval, not the true risk $R(f_\theta)$
- Correct: „The probability to obtain a confidence interval from an experiment that contains the true risk is 95%“
- Wrong: „We have obtained a confidence interval of size ε from an experiment. The probability that the interval contains the true risk is 95%“
- The latter statement does not work because after completion of the experiment, the interval contains the risk or does not contain the risk. There is no distribution over the true risk

Summary

- In practical applications of machine learning, being able to estimate the error of a trained model is important
  - have to know how many errors to expect after deployment of model
  - want to compare different models, or tune hyperparameters

- The error of a model needs to be estimated based on independent test data
  - holdout testing: single train-test split
  - cross-validation: multiple train-test split with averaging

- Error estimates are based on finite samples of randomly drawn data, and therefore never fully accurate
  - An error estimate can be considered the result of a random process (data sample)
  - Can characterize the uncertainty in the error estimate using confidence intervals