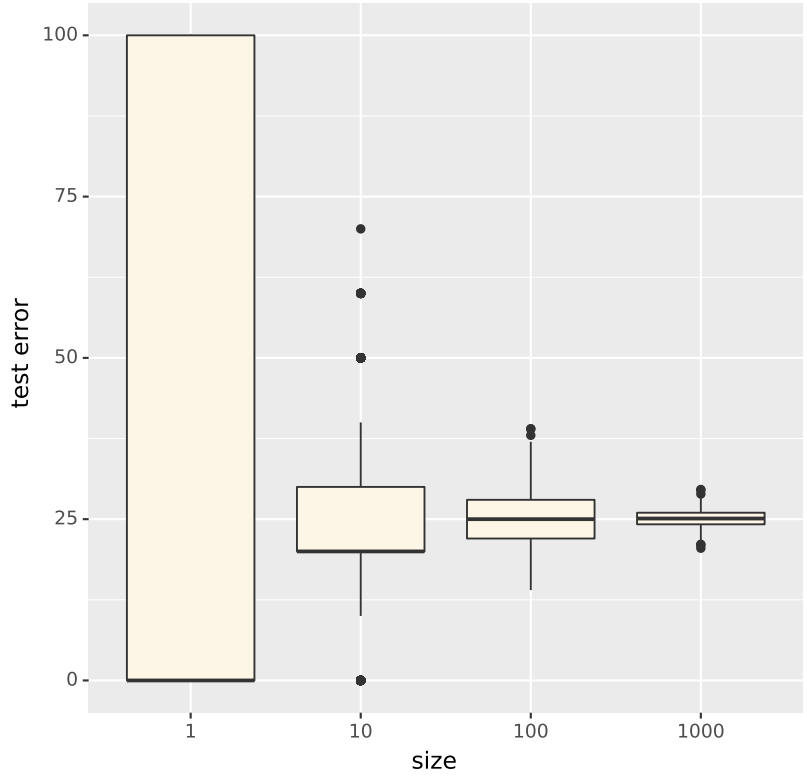
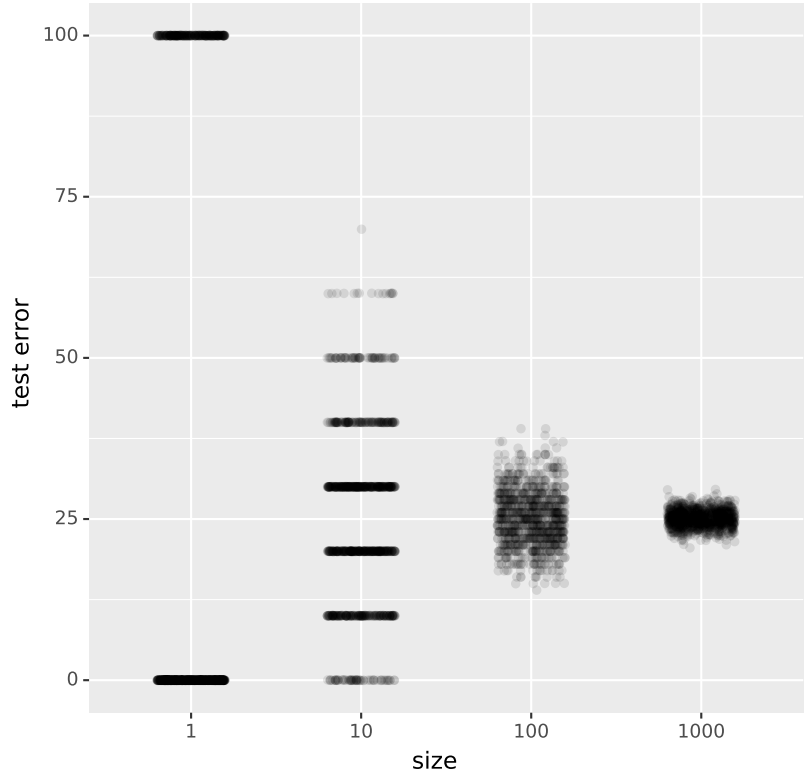
LTAT.02.004 Machine Learning II

Basics of probabilistic modelling

Sven Laur
University of Tartu

Why does empirical risk converge at all?

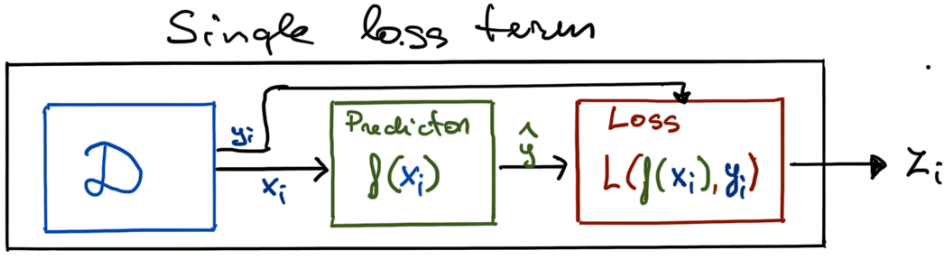




▷ Depends on the test set.

▷ Statistical fluctuations decrease with size.

Empirical risk as mean of random variables



Recall that empirical risk is computed through the following formula

$$R_N(f) = \frac{1}{N} \cdot \sum_{i=1}^{N} L(f(\boldsymbol{x}_i), y_i) = \frac{1}{N} \cdot \sum_{i=1}^{N} z_i$$

where all samples $(\boldsymbol{x}_i, y_i)$ are assumed to be

- ▷ independent from each other,
- ▷ coming from the same distribution.

Law of large numbers

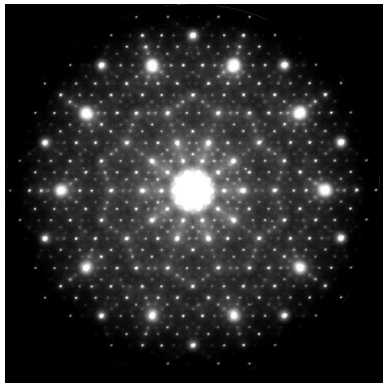
Central limit theorem. Let $z_1, \ldots, z_N$ be independent and identically distributed samples form a *real-valued distribution* with a *finite standard deviation* σ and *mean* μ. Then the random variable

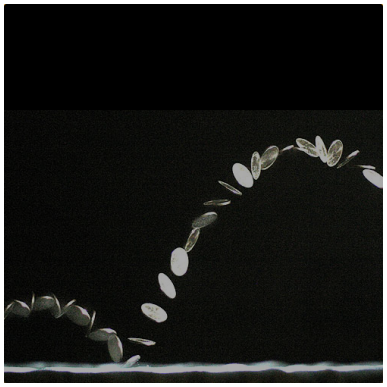
$$S = \sqrt{N}\left(\frac{1}{N} \cdot \sum_{i=1}^{N} z_i - \mu\right)$$

converges *in distribution* to normal distribution $\mathcal{N}(mean = 0, sd = \sigma)$.

- ▷ Under mild assumptions the empirical risk $R_N(f)$ converges to risk $R(f)$.
- ▷ The result is not precise enough to quantify approximation error.

What is probability?





Probability is a measure of uncertainty which can rise in several ways

- ▷ Intrinsic uncertainty in the system
- ▷ Uncertainty caused by inherent instability of the system
- ▷ Uncertainty caused by lack of knowledge or control over the system

Frequentistic interpretation of probability

Probability is an average occurrence rate in long series of experiments.

- ▷ Law of large numbers
- ▷ Probability is a collective property
- ▷ Probabilities can be assigned only to future events

Bayesian interpretation of probability

Probability reflects persons individual beliefs on future or unknown events.

- ▷ Belief updates through the Bayes rule
- ▷ Probability is an inherently subjective property
- ▷ Probabilities can be assigned to past, present and future events

Ultra-frequentistic interpretation of probability

Events with small enough probability do not occur

- ▷ The main tool in classical statistics
- ▷ Errors in judgement does not matter if a gamma ray pulse kills us.
- ▷ One must avoid the lottery paradox in the reasoning

The goal of statistical inference

Frequentist goal

- ▷ The aim of statistics is to design algorithms that work well on average.
- ▷ For that one needs to specify probabilistic model for data sources.
- ▷ Confidence is the fraction of cases the algorithm works as specified.

Bayesian goal

- ▷ The aim of statistics is to design algorithms that allow *rational individuals* to reliably update their beliefs through Bayes formula.
- ▷ Besides the data source model one has to provide model for initial beliefs.
- ▷ Correctness of an algorithm does not make sense.

Frequentistic methods

Illustrative example

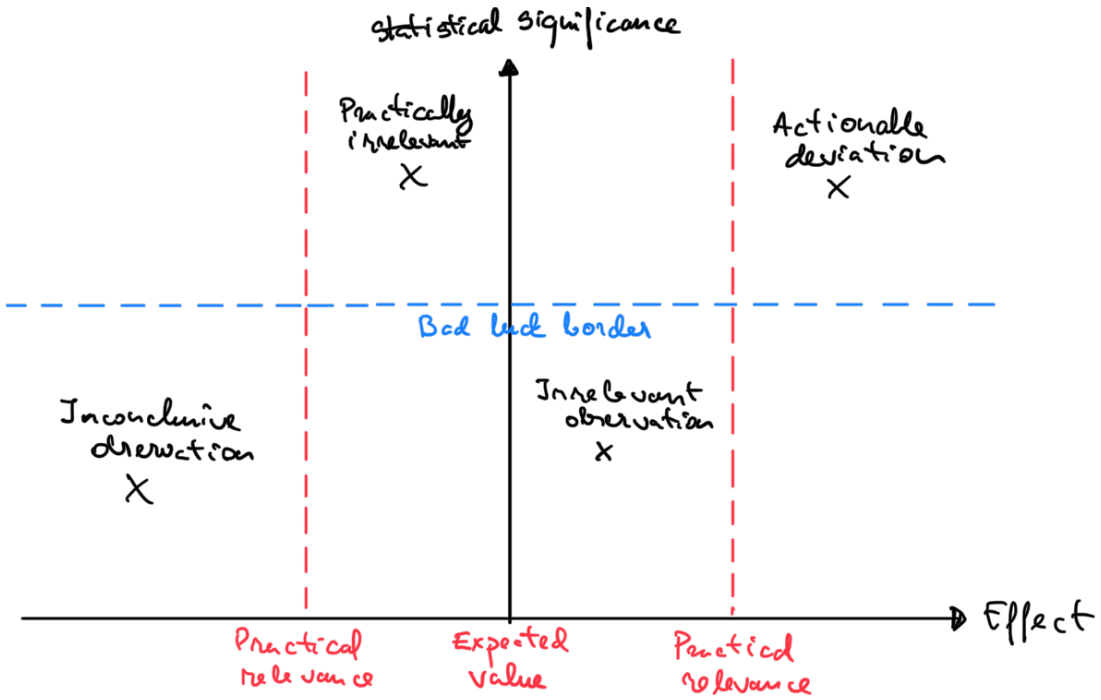
Consider an experiment that yields 2 heads and 8 tails.

- ▷ Frequency of heads is 20%.
- ▷ Can the coin be still fair?

Consider an experiment that yields 1,000,100 heads and 999,900 tails.

- ▷ Frequency of heads is 50.005%.
- ▷ Can the coin be still biased?

Central question in statistical testing



The question is my observation relevant has two aspects

- ▷ Can we explain the difference by sheer luck?
- ▷ Is the difference between expected and observed big enough?

Causation between zero-one events

Assume that condition A causes the event $B = 1$ with probability p, i.e.,

$$\Pr[B = 1|A] = p$$

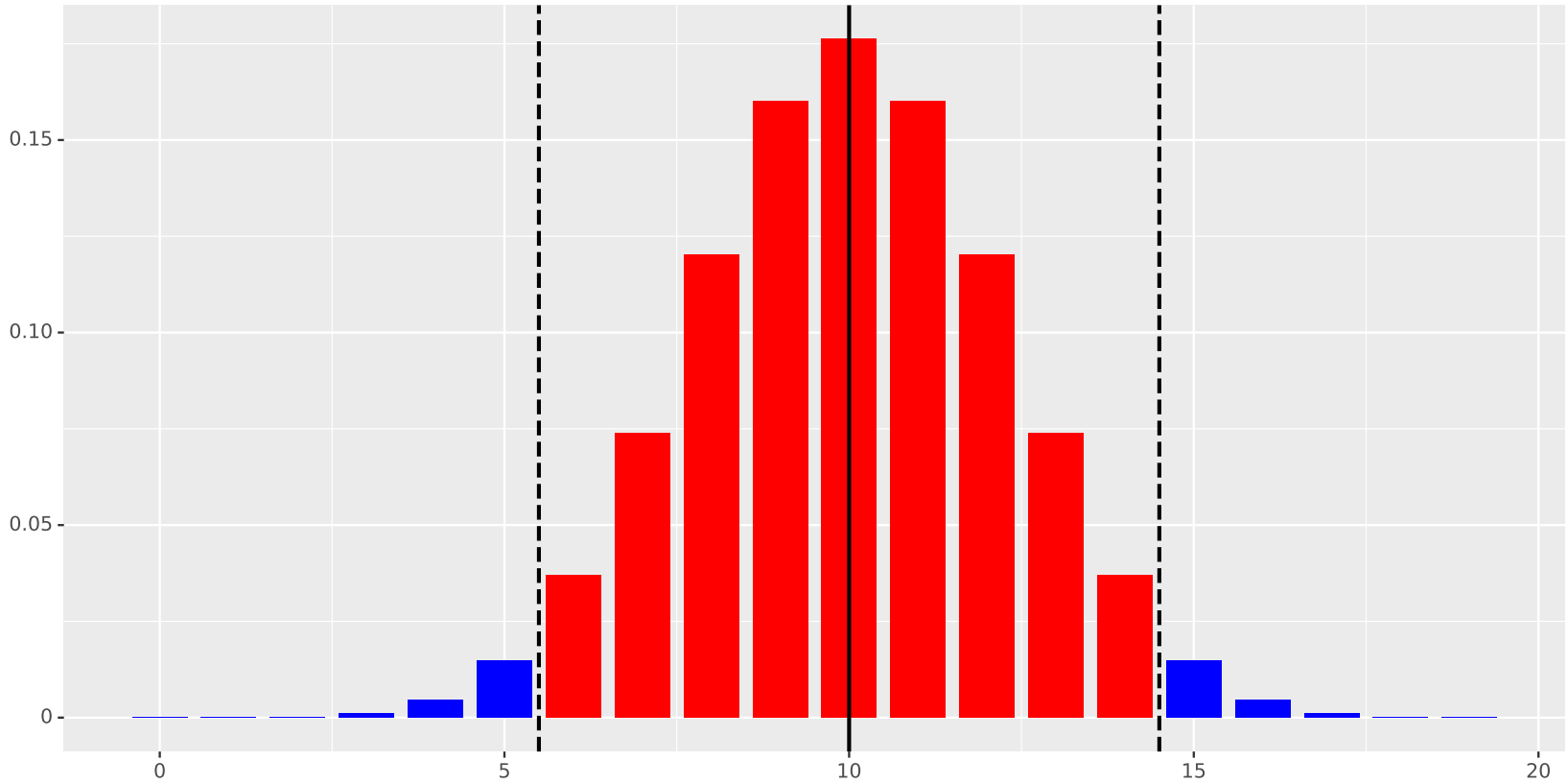
Then the probability is to get k ones in n independent trials is

$$\Pr[B_1 + \cdots + B_n = k|A] = \binom{n}{k} p^k (1-p)^{n-k}$$

The number of ones in known to have a *binomial distribution*

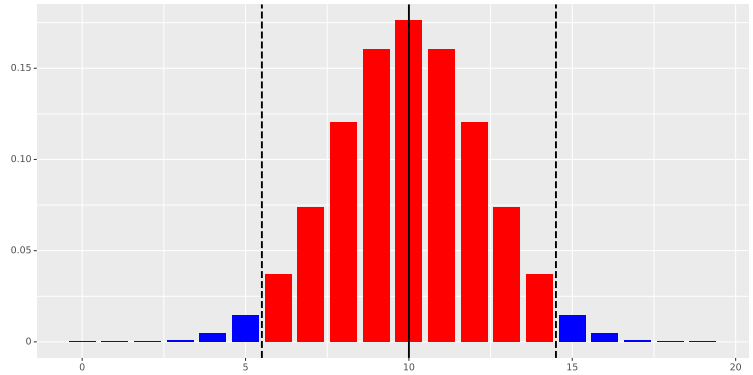
$$B_1 + \cdots + B_n \sim \mathsf{Bin}(n, p)$$

Illustration



The distribution of $B_1 + \ldots + B_n$ depends solely on the number of trials n and the probability p. Some values of $B_1 + \ldots + B_n$ are very unlikely.

Does a classifier beat a random guess?



Consider three algorithms on twenty element test set:

- ◇ Algorithm A gets 9 correct answers;
- ◇ Algorithm B gets 13 correct answers;
- ◇ Algorithm C gets 17 correct answers.

- ▷ Which of them are better than random classifiers?
- ▷ Which of them are classifiers good enough for practical applications?

How to build a statistical test

I. Null hypothesis:

▷ The probability of heads in a coinflip is $\Pr[B_i = 1] = p$.

II. Choose value to compute aka test statistic:

▷ Our test statistic will be $B_1 + \ldots + B_n$.

III. Consequences on the observations:

▷ The observed sum $B_1 + \ldots + B_n \sim \mathrm{Bin}(n = 20, p = 0.5)$.

▷ Limit on the tail probability $\Pr[|B_1 + \ldots + B_n - 10| \geq 6] \leq 5\%$

IV. Test procedure

▷ Reject null hypotesis at *significance level* 5% if $|B_1 + \ldots + B_n - 10| \geq 6$.

Properties of statistical tests

Statistical test is a classification algorithm designed to distinguish a fixed distribution of negative examples specified by a null hypothesis.

Any *fixed* classification *rule* can be converted to a statistical test by finding out the percentage of false positives aka *p-value*:

- ▷ There might exists a closed form solution.
- ▷ We can always estimate p-values using simulations.
- ▷ Observations must be compressed into a single decision value.

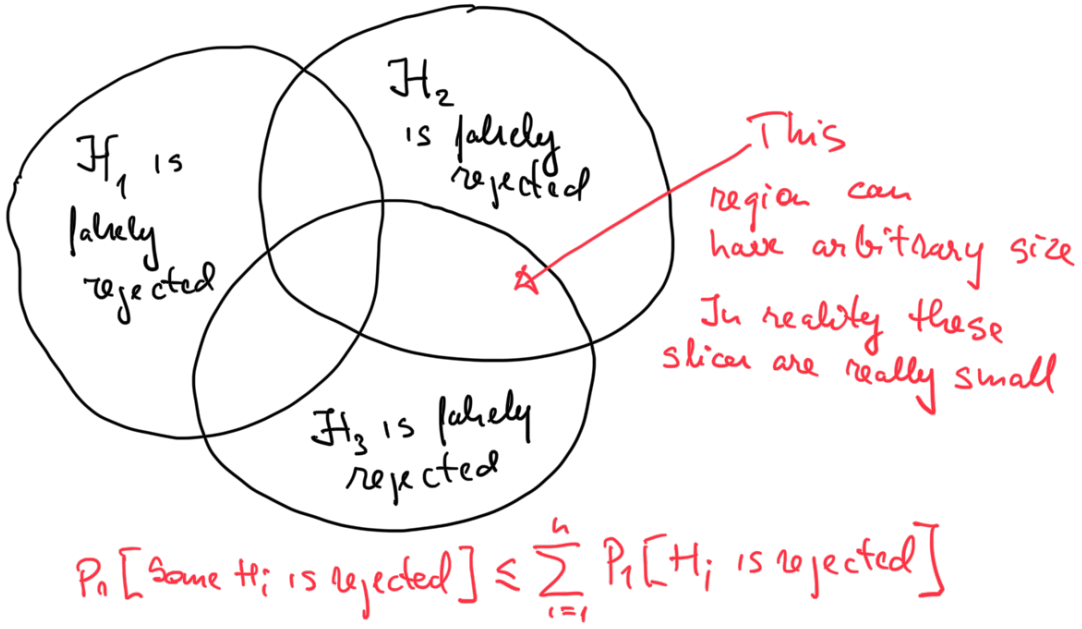
Testing several hypothesis in parallel increases the number of false positives. Several p-value adjustment methods are used to correct the issue:

- ▷ Bonferroni correction is almost optimal
- ▷ FDR correction controls the expected number false positives

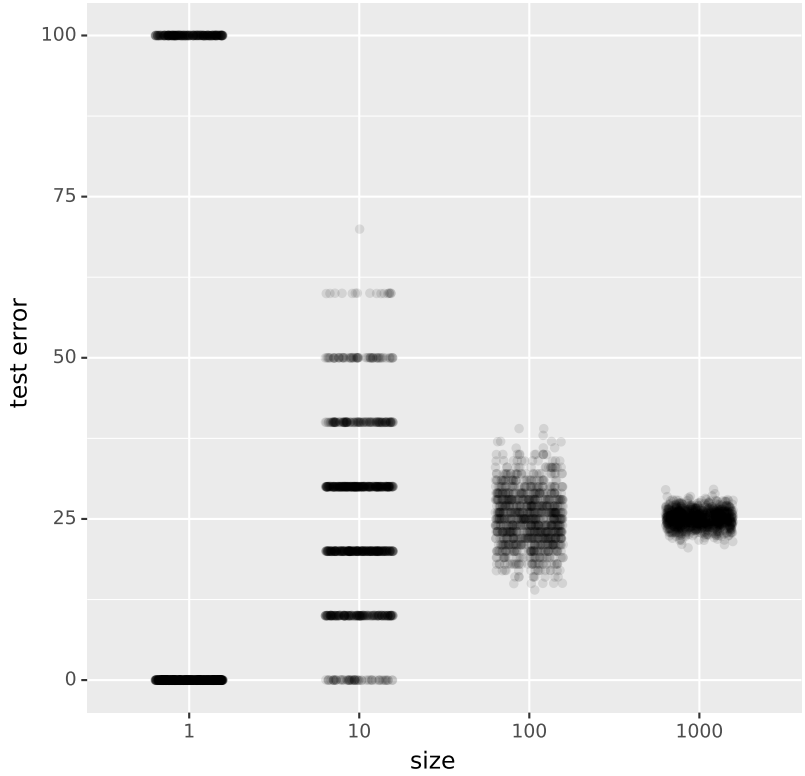
Bonferroni correction for tests

Assume that data is generated so that null hypotheses $\mathcal{H}_1, \ldots, \mathcal{H}_n$ hold.

- ▷ Then we can still reject some the tests due to bad luck.
- ▷ We can use really naive enough bound visualised below.



How far is the true risk?



- How wide error bars cover true risk for *all* observations?
- How wide error bars cover true risk for *most* observations?

How to build confidence intervals

I. Construct a family of statistical tests:

- ▷ Define a statistical test T_p for all possible parameter values p.
- ▷ All tests should share the same test statistic.

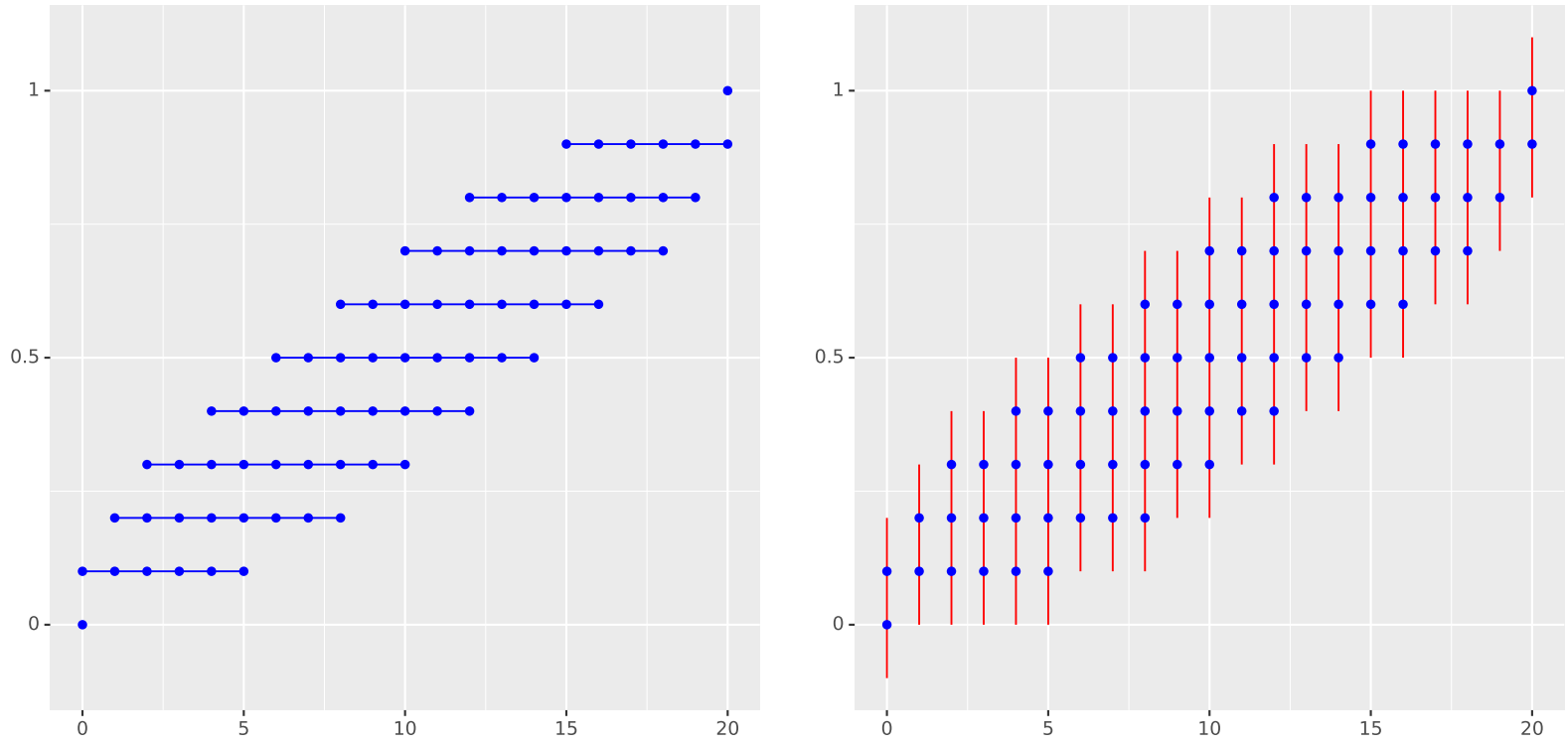
II. Perform multiple hypotesis testing for all parameter values:

- ▷ Accept all parameters values for which p-value is greater than $1 - \alpha$.
- ▷ Output a minimal interval that covers all accepted parameter values.

Rationale

- ▷ The true parameter value is rejected on α-fraction of possible observations.
- ▷ For the remaining cases the true value is inside the predicted interval.

Illustration



- ▷ Acceptance ranges for different parameter values on the left.
- ▷ Extended parameter ranges covering all accepted parameters on the right.
- ▷ These ranges are the desired confidence intervals.

Interpretation of confidence intervals

Definition. Confidence interval for a parameter p is an outcome of an approximation algorithm. The algorithm must output an interval $[\hat{p}-\varepsilon, \hat{p}+\varepsilon]$ such that the true estimate is in the range on α-fraction of cases.

Paradoxical inapplicability

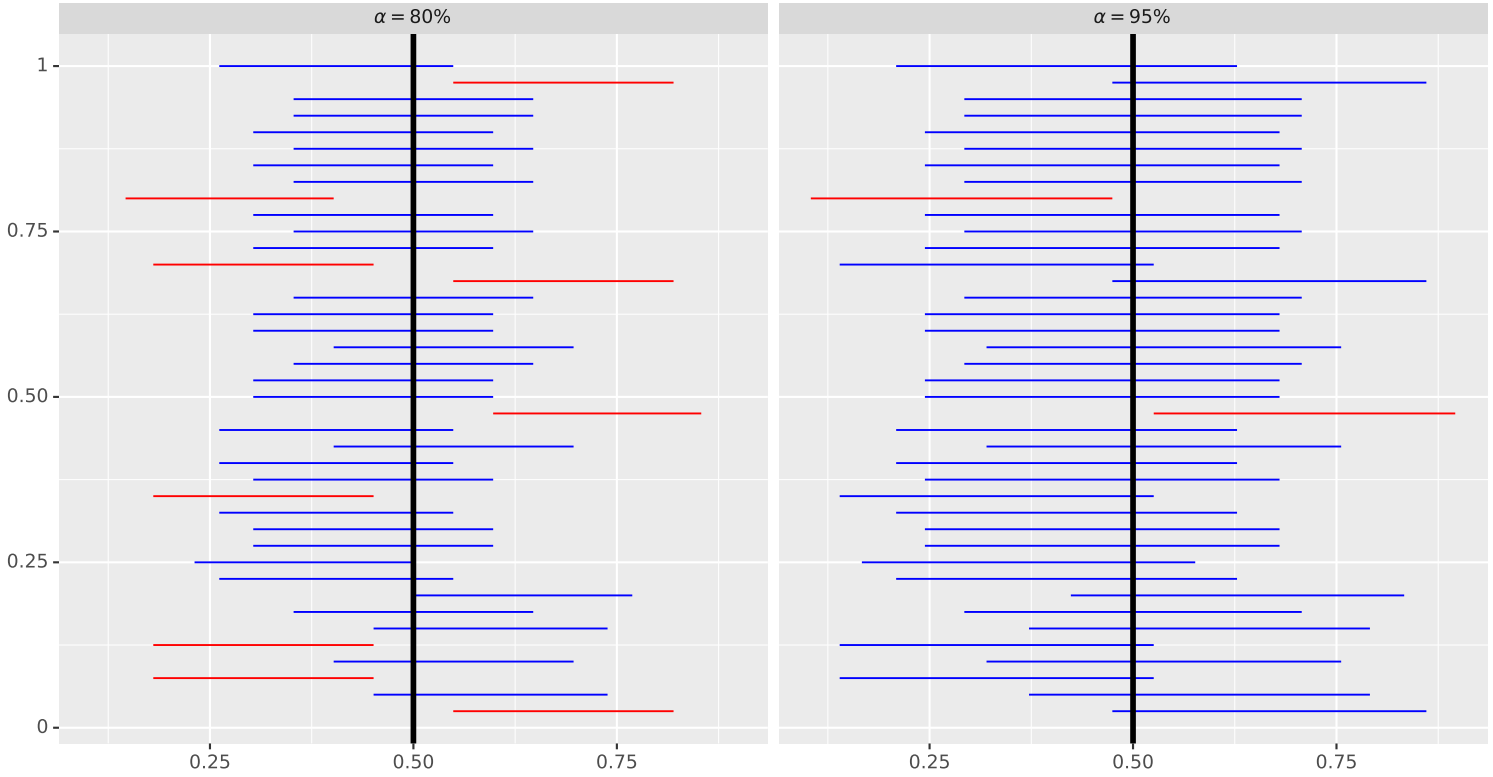
The definition does not state that the probability $p \in [\hat{p}-\varepsilon, \hat{p}+\varepsilon]$ is α!

- ▷ The statement $p \in [\hat{p} - \varepsilon, \hat{p} + \varepsilon]$ is either true or false.
- ▷ There is no probability left. We just *do not know* the answer!

Ultra-frequentistic resolution

- ▷ If $1 - \alpha$ is small enough say 5% then the algorithm is always correct.

Illustrative example



By increasing the length of the interval we increase the fraction of runs for which the true value of p lies in the interval.

Problems with confidence intervals

Inability to capture background knowledge

- ▷ What if I know that $p \in [0.1, 0.2]$ and observe $B_1 = \ldots = B_N = 1$?
- ▷ Then the estimate $[\hat{p} - \varepsilon, \hat{p} + \varepsilon]$ is clearly wrong although on average this confidence interval is reasonable.

Multiple hypothesis testing

- ▷ Using several confidence intervals in parallel increases the fraction of cases where some true estimate is out of the predicted range.
- ▷ We can use p-value adjustment methods are used to correct the issue.

Prediction intervals

Even if we know the true relation $y = f(\boldsymbol{x})$ we cannot predict the observation $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$, as the noise term ε_i is not known ahead.

▷ We cannot give upper and lower bounds for y_i which always hold.

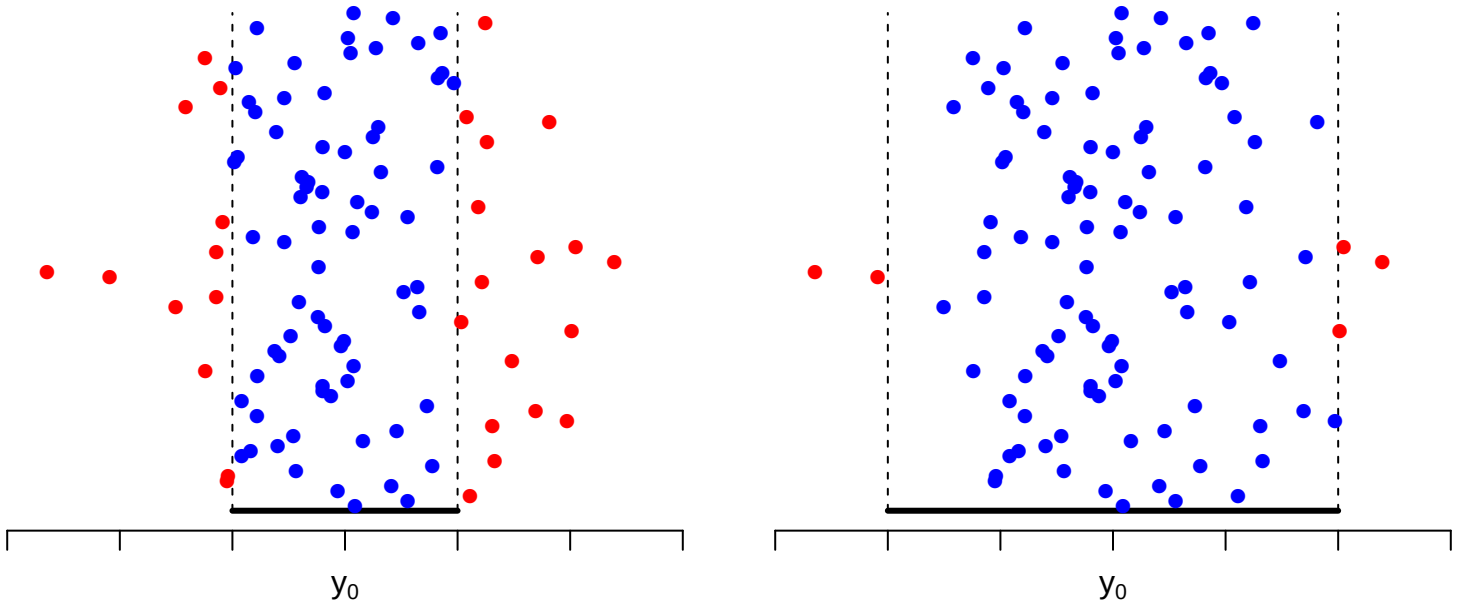
Instead, we can specify a prediction interval $[y_* - \varepsilon, y_* + \varepsilon]$ so that with probability 95% the resulting measurement y_i is in the range.

▷ Usually, the analysis is similar to confidence interval derivation.

Interpretation of prediction intervals is different from confidence intervals.

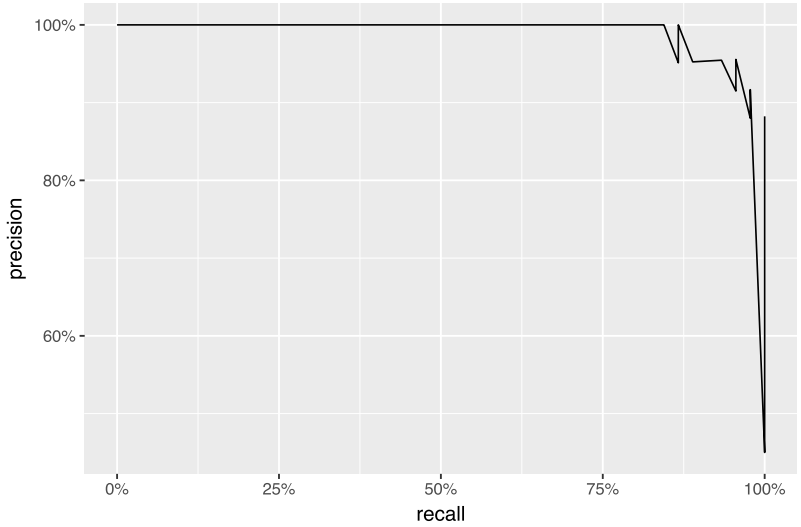
▷ The probability estimate holds for the particular interval.

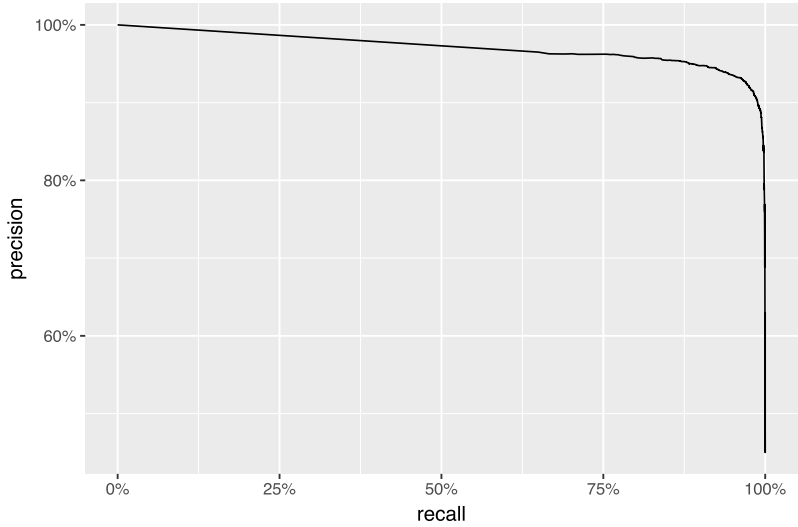
Illustrative example



By increasing the length of the prediction interval we increase the fraction of future measurements which fall into interval.

Fluctuations in performance profiles





- ▷ Precision-recall graph is not smooth if the test set is small.
- ▷ How much the true graph can be different from observed?
- ▷ How many of samples are needed to get a decent resolution?

Confidence envelopes

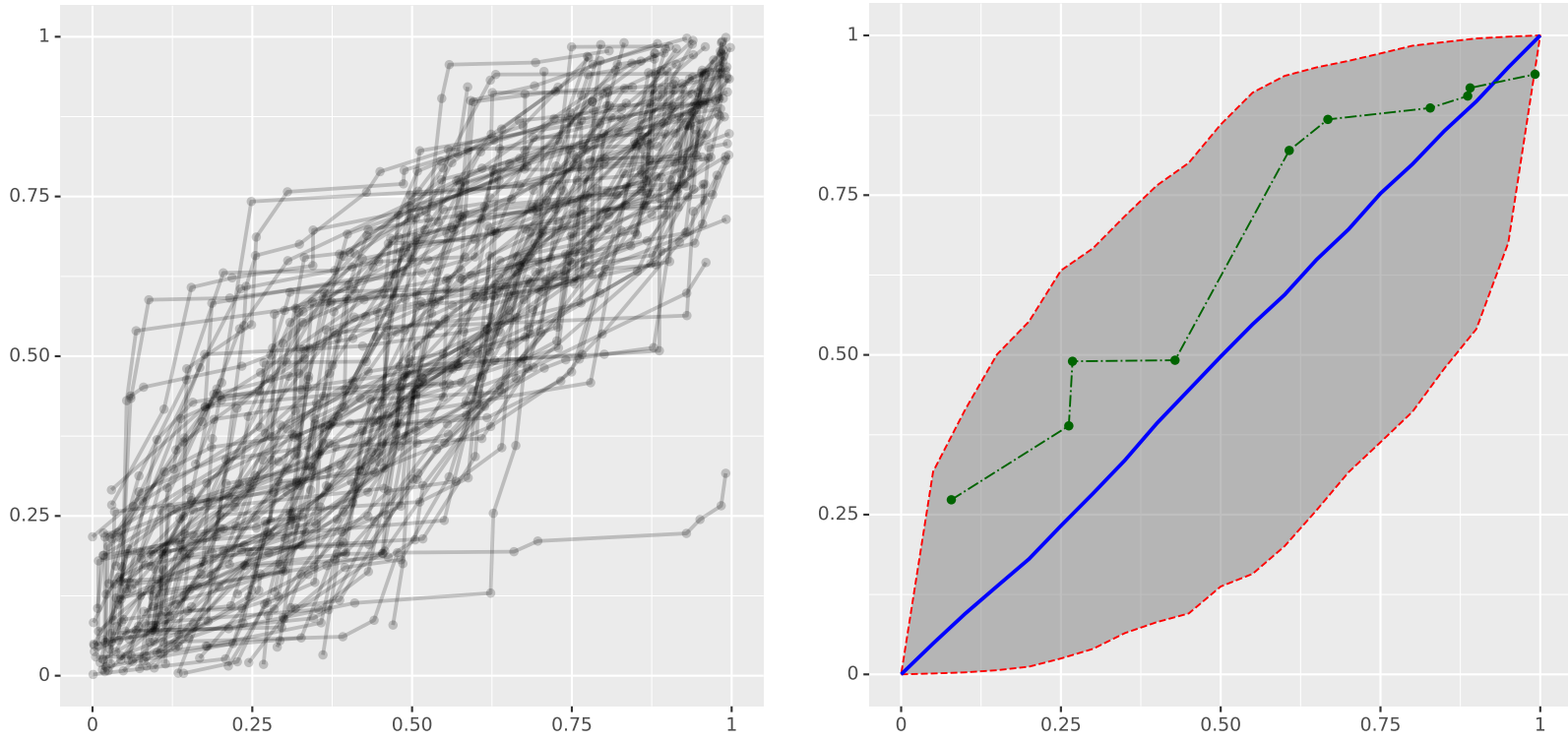
Confidence intervals is a good way to visualise uncertainty of a particular parameter. However, we are sometimes interested in the uncertainty many parameters or in the uncertainty of a function:

- ▷ How a predictor $f : [0, 1] \to \mathbb{R}$ depends on the training set
- ▷ How a ROC curve $\mathrm{ROC} : [0, 1] \to [0, 1]$ depends on the test set
- ▷ How should a quantile-quantile plot be distributed.

Confidence bands are generalisations of confidence intervals

- ▷ Pointwise confidence band is a collection of confidence intervals
- ▷ Simultaneous confidence band must enclose α-fraction of functions.
- ▷ Simultaneous confidence bands are much wider than pointwise bands.

Illustrative example



- ▷ Distribution of qq-lines visualised through a sample on the left.
- ▷ A simulation based pointwise 95% confidence envelope on the right.
- ▷ The significance level that qq-line is inside the envelope is ca 50%.

Permutation tests

Baseline problem:

- ▷ Achievable accuracy depends on the data distribution.
- ▷ Artefacts in the dataset may bias performance measures.

Label permutation. A random permutation π on outputs y_i destroys correlations between input-output pairs $(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})$ but preserves marginal distribution of inputs and outputs.

Permutation test. Estimate how probable is to achieve equal or higher accuracy than was observed on the real data.

- ▷ If this probability is small then there must be signal in the data.
- ▷ The test completely neglect the effect size, i.e., how much results differ.
- ▷ Statistical significance does not imply utility!

Crossvalidation

Empirical risk and law of large numbers

Under mild assumptions the empirical risk $R_N(f)$ converges to risk $R(f)$ and we can actually use normal distribution to estimate probabilities:

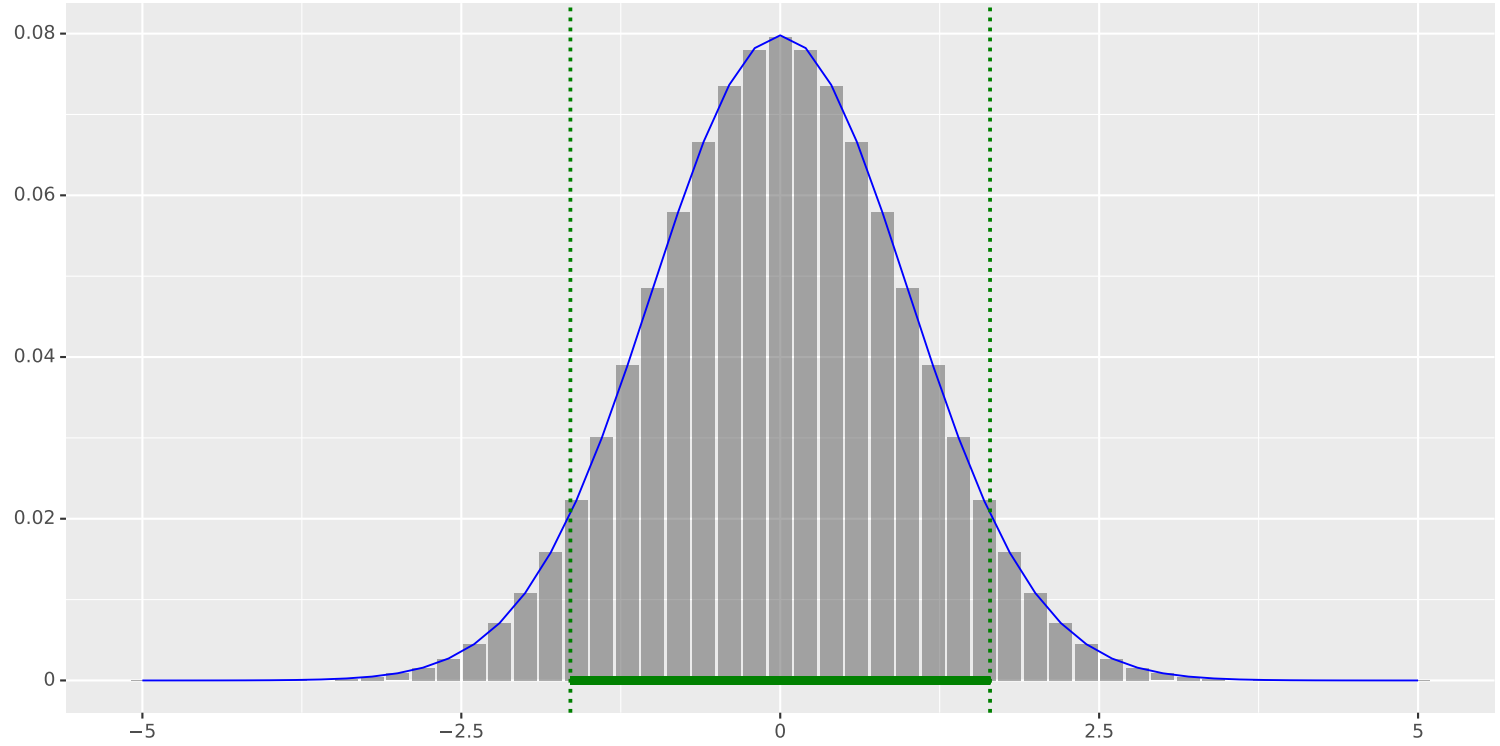
$$\Pr\left[|R_N(f) - R(f)| \geq \varepsilon\right] \lesssim 2 \cdot \int_{-\infty}^{\varepsilon} \frac{\sqrt{N}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{Nt^2}{2\sigma^2}\right) dt$$

for a finite value σ where σ^2 is the variance of loss $\mathbf{D}(R(f))$.

What do we need to apply this result

- ▷ Test set elements must be independent and from the same distribution.
- ▷ CLT assumes that *risk μ is finite* and *standard deviation σ is finite*.
- ▷ Test set must be large enough that approximation is good enough.
- ▷ We *need to approximate* σ so that we can estimate the integral.

Visual representation



Convergence implies that the centre area of is well approximated

- ▷ 95% confidence intervals are roughly the same for both distributions

Moment matching

We know that the empirical risk $R_N(f)$ converges to normal distribution

- ▷ Normal distribution is fixed by a mean μ and variance σ^2
- ▷ We can estimate mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of a loss term $L(f(\boldsymbol{x}), y)$
- ▷ Then the estimates of mean and variance of the empirical risk are

$$\mathbf{E}(R_N(f)) \approx \hat{\mu}$$

$$\mathbf{D}(R_N(f)) \approx \frac{\hat{\sigma}^2}{N}$$

- ▷ This allows us to approximate $R_N(f)$ with normal distribution

Why do we need a test set at all

Machine learning algorithm

- ▷ Count number of zeroes n_0 and number of ones n_1 in training sample.
- ▷ If $n_0 > n_1$ output $f_0(x) \equiv 0$, otherwise output $f_1(x) \equiv 1$.

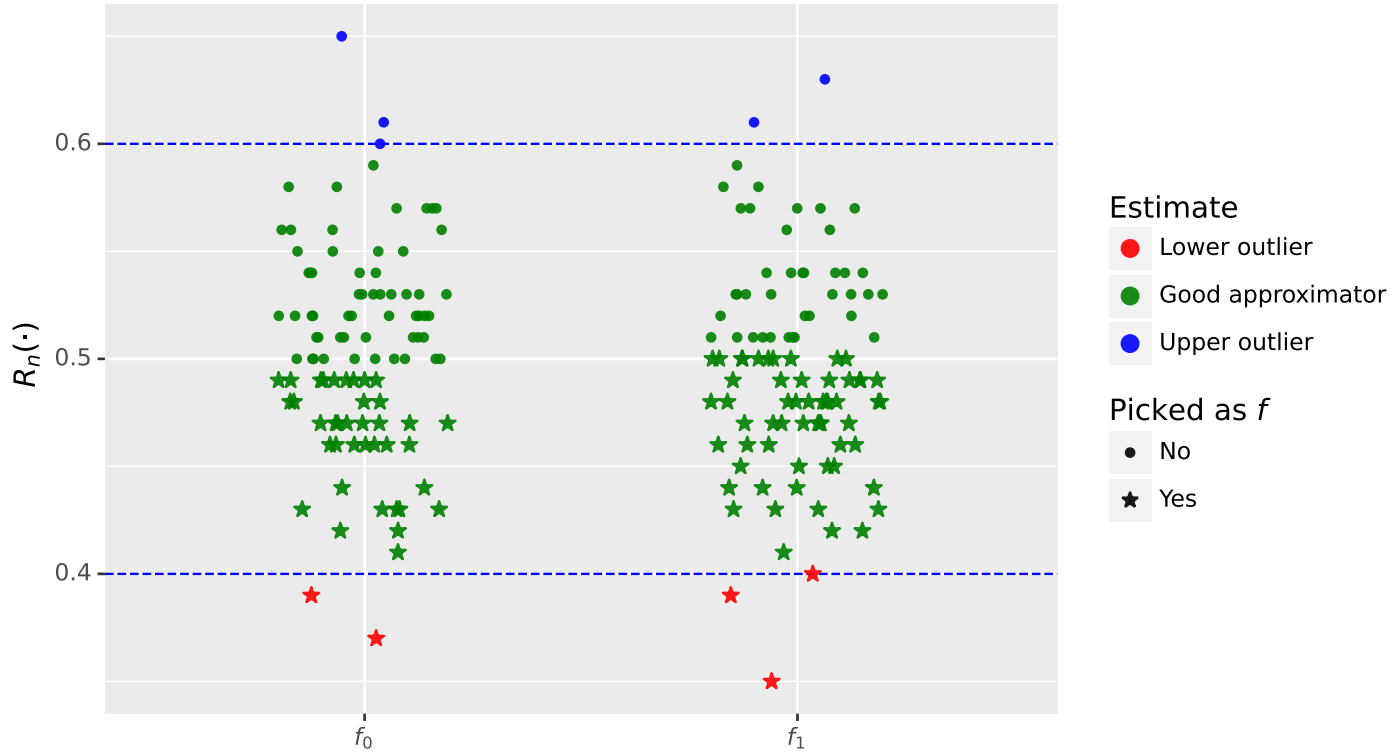
Data source

- ▷ Choose the input x randomly form the range $[0, 1]$
- ▷ Choose the label y randomly from the set $\{0, 1\}$.

True risk value

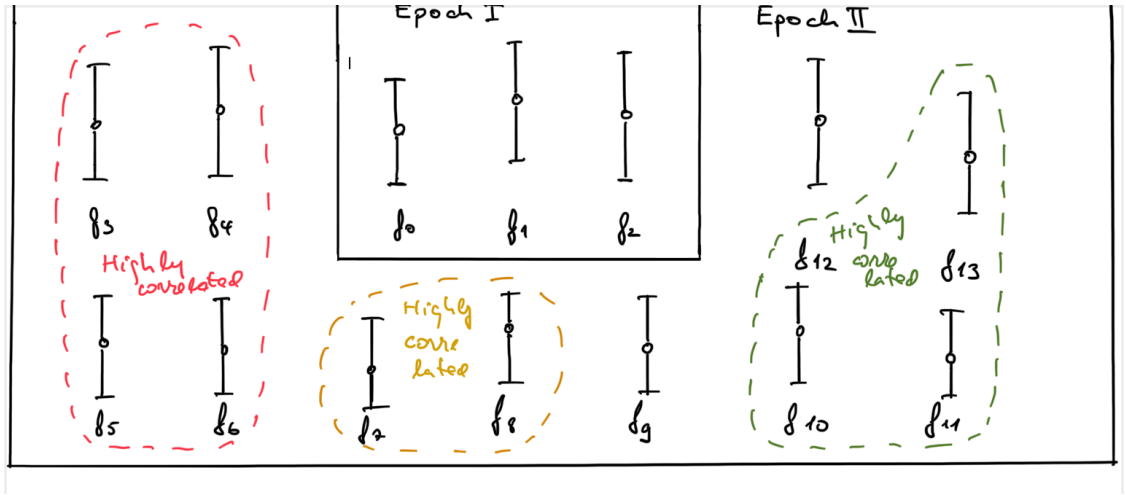
- ▷ Clearly the risk of both rules $R(f_0) = R(f_1) = 0.5$.
- ▷ The risk of of our learning algorithm $R(f)$ is also 0.5.

What happens during the training phase



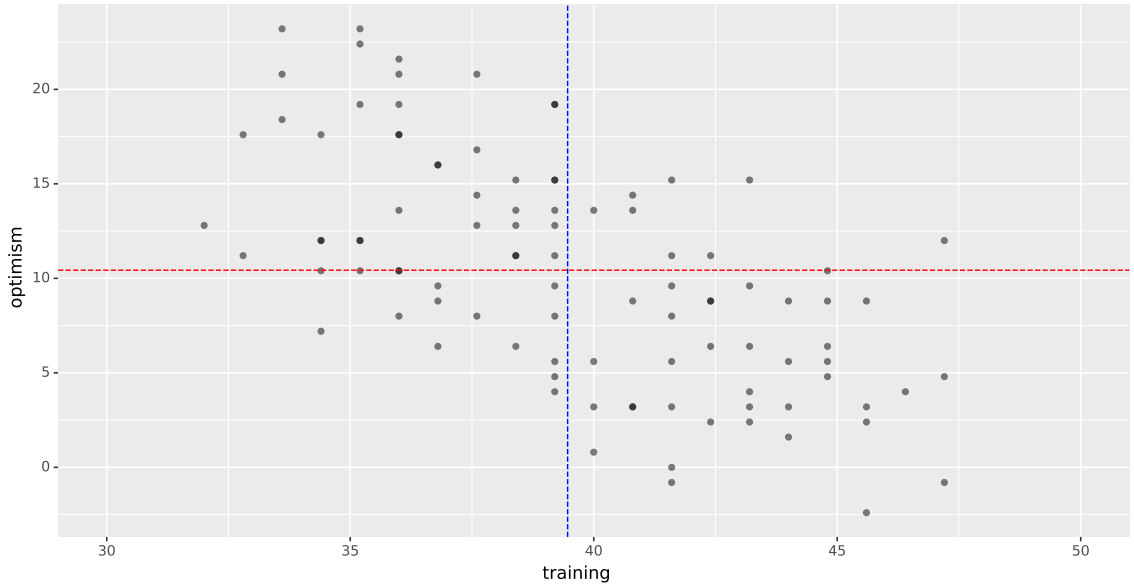
- ▷ We always choose the function f_i that underestimates the true risk!
- ▷ The probability that we go below the range effectively doubles.

What happens in real ML algorithms



- ▷ Not all function are achievable at once.
- ▷ More epochs open up more confidence intervals to compare.
- ▷ Not all empirical risk measurements are independent.
- ▷ Empirical risk values of similar prediction functions are correlated.

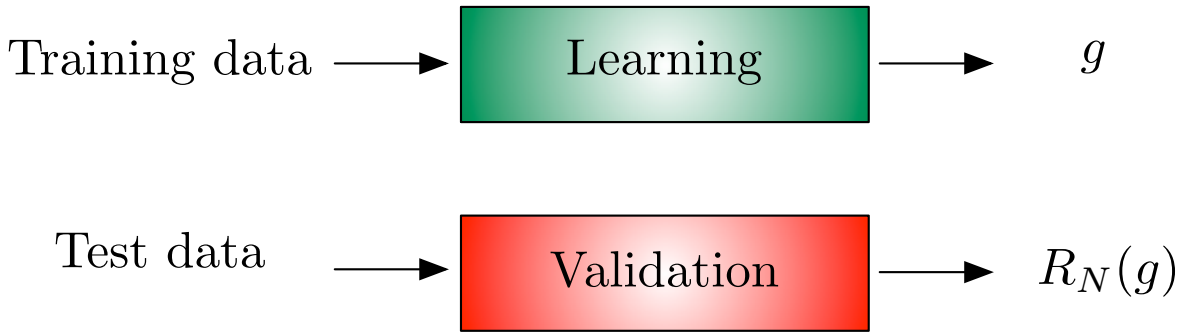
Generalisation gap aka optimism



By knowing the *optimism* $\Delta = R(f) - R_N(f_i)$ we can correct $R_N(f)$.

- ▷ Optimism is usually anti-correlated with empirical risk $R_N(f)$.
- ▷ Commonly mean value of Δ is used for the correction.
- ▷ Simple shifting does not resolve the systematical bias.

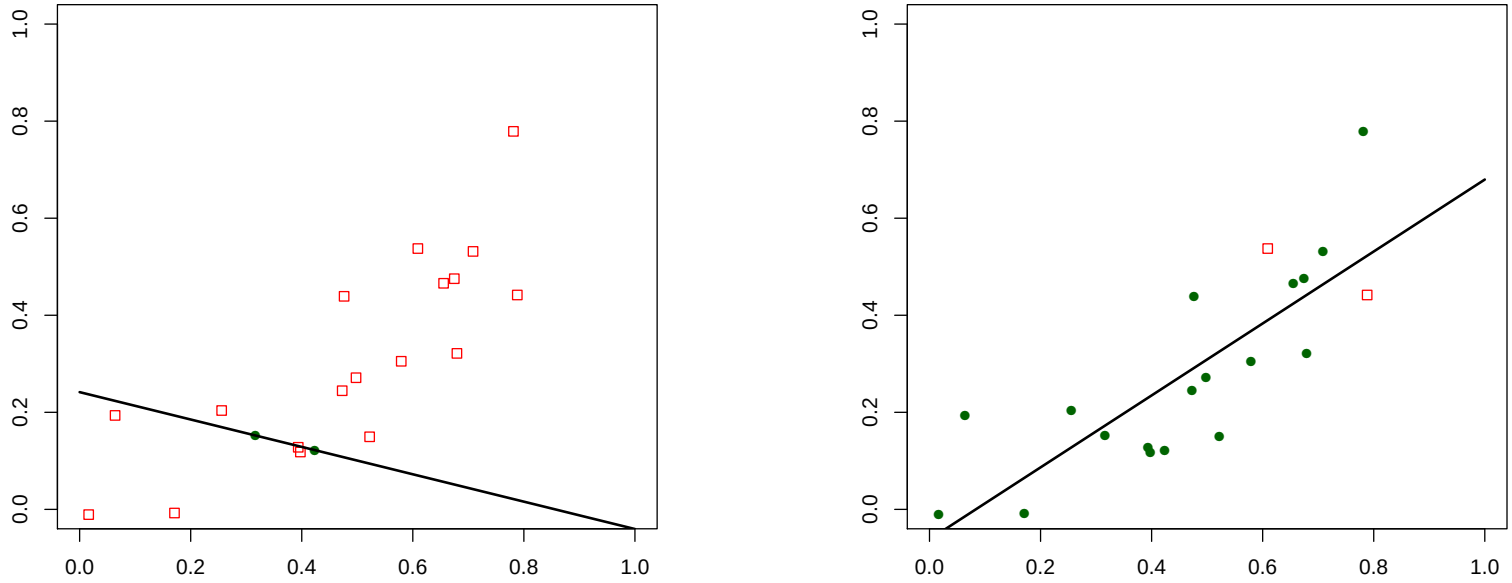
Why does the holdout testing work



By randomly splitting the data into training and test data we assure

- ▷ The training and test sets are independent under IID assumption.
- ▷ On a training set we compare many models and choose few winners.
- ▷ These functions are independent from the test set data.
- ▷ As there number of functions is small the law of large numbers holds.

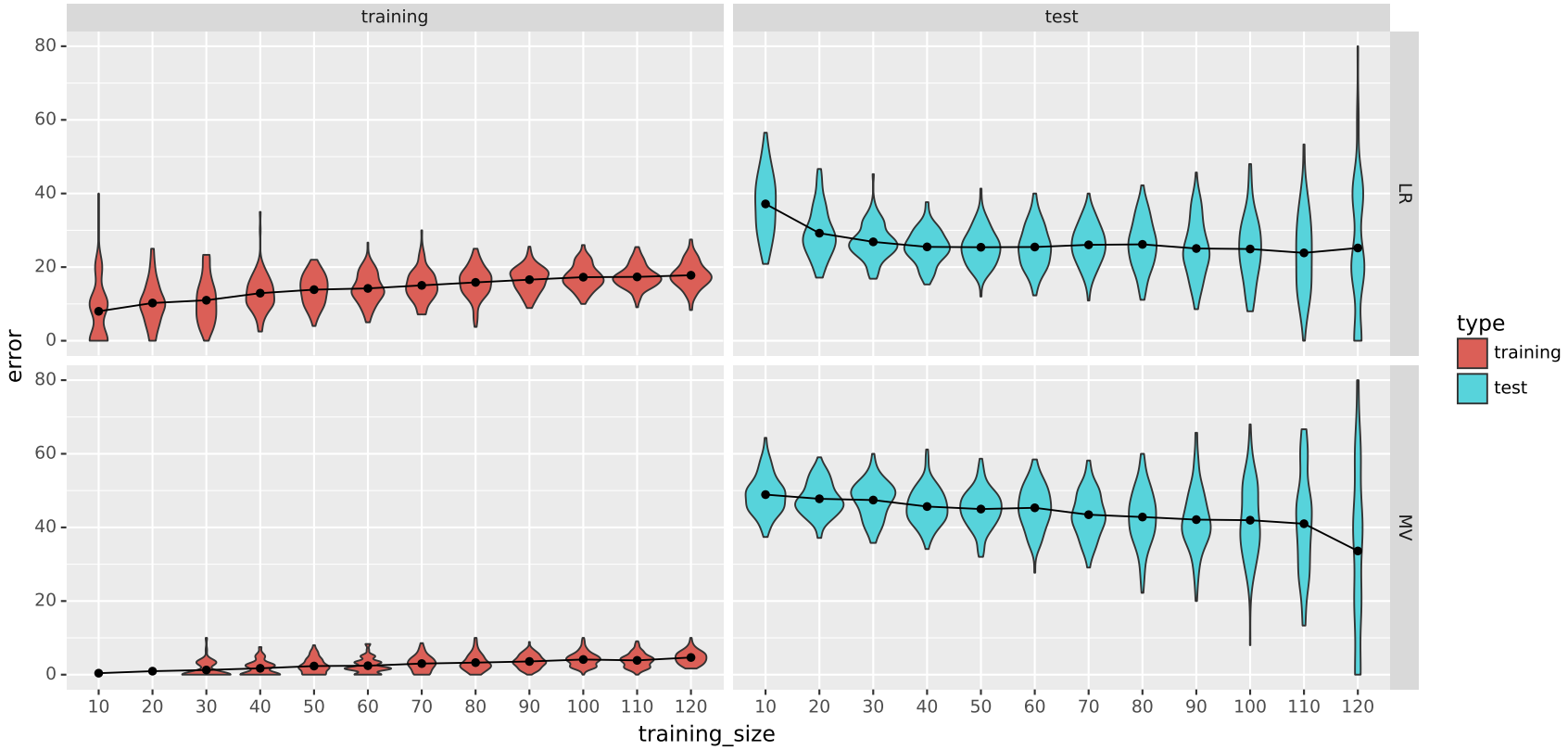
Why is holdout testing problematic



If the number of available data points is small we have to choose:

- ▷ a small training set and bad model but good estimate on risk
- ▷ a big training set and good model but bad estimate on risk

Why is holdout testing problematic



Typical tradeoffs between learning-bias and variance of the validation error.

Crossvalidation as an engineering trick

To reduce holdout error, we can do several holdout experiments. Since we do not have enough data, we redo splitting and training on the same data.

This idea yields a generic crossvalidation scheme

1. Generate several splits of test and training data
2. For each split train the model and compute holdout error
3. Tabulate results

	Split 1	Split 2	$\dots$	Split k
Training error	S_1	S_2	$\dots$	S_k
Test error	E_1	E_2	$\dots$	E_k
Optimism Δ	$E_1 - S_1$	$E_2 - S_2$	$\dots$	$E_k - S_k$

4. Compute averages $E = \frac{1}{k}(E_1 + \cdots + E_k)$ and $\Delta = \frac{1}{k}(\Delta_1 + \cdots + \Delta_k)$
5. Visualise results and compute confidence intervals for estimates if needed.

What does crossvalidation measure?

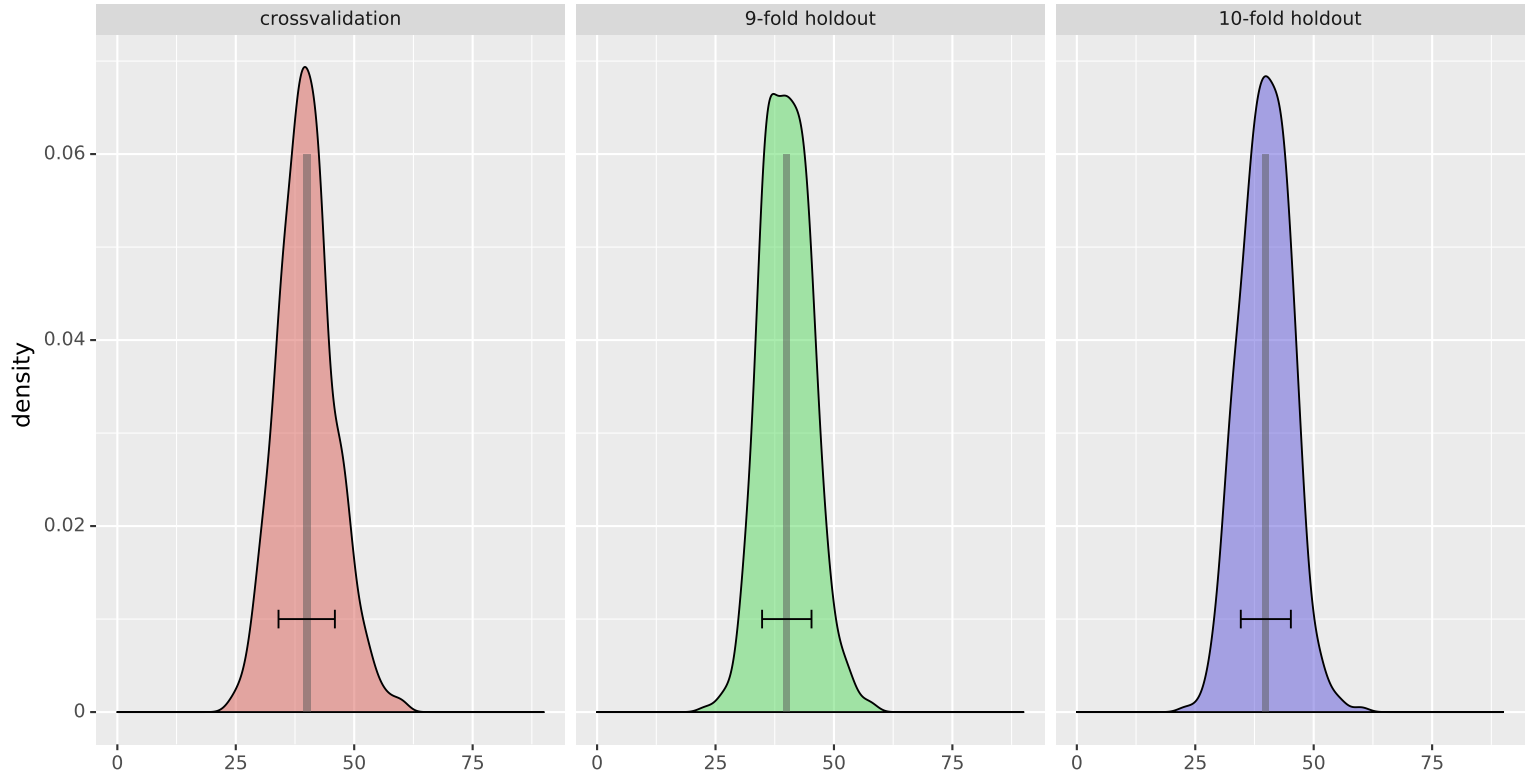
For each fold we have a separate predictor f_i and test error E_i:

- ▷ Average E characterises average behaviour of $f_1, \ldots, f_k$.
- ▷ Algorithm can use only (1-1/k) fraction of the available data.
- ▷ If there is not enough data for training E overestimates the error.

To estimate the performance of a classifier f trained on the entire data:

- ▷ We must estimate the difference between test and training error $\Delta(f)$.
- ▷ For normal ML algorithm optimism decreases by increasing the size n.
- ▷ Crossvalidation estimates Δ at the point $(1 - 1/k) \cdot n \lesssim n$.
- ▷ Hence we can go from training error to test error estimate.
- ▷ Training and test set fluctuations influence the outcome.

Crossvalidation vs holdout estimates



- ▷ Crossvalidation error is slightly larger as the training set is smaller.
- ▷ Crossvalidation error is slightly more fluctuating due to correlations.
- ▷ Quite often these effects are quite small in practice.

Theoretical explanation

Theorem. Crossvalidation error $E = \frac{1}{k}(E_1 + \cdots + E_k)$ is an unbiased estimate for the average test error that is taken over all models that are trained on $(1 - 1/k) \cdot n$ samples.

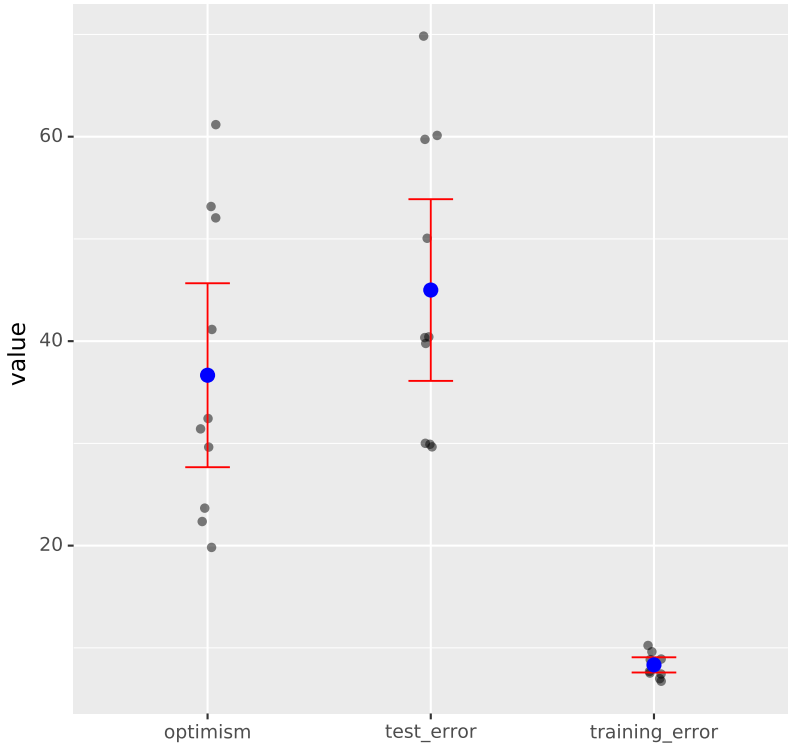
Proof

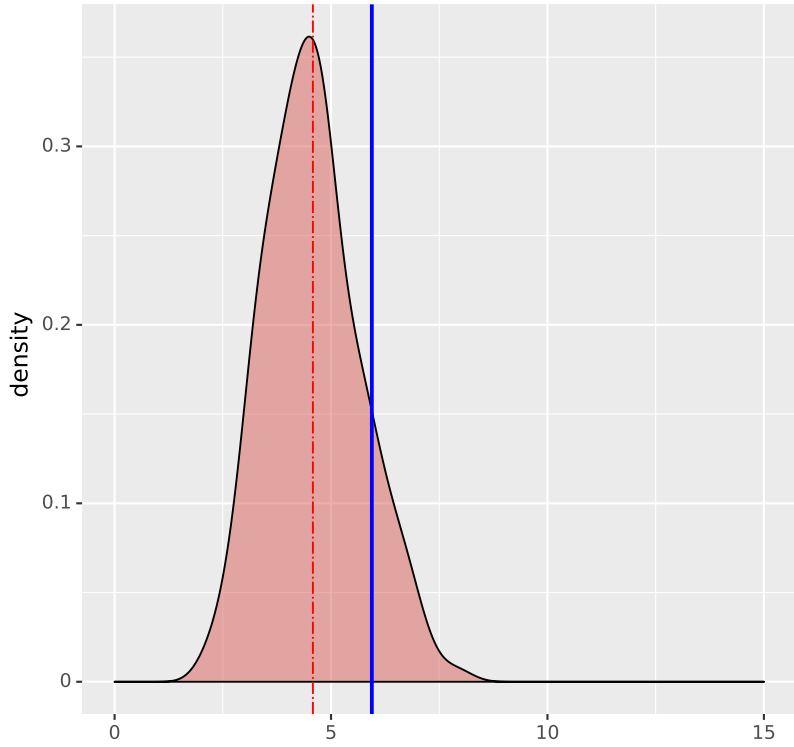
$$\mathbf{E}[E] = \frac{\mathbf{E}[E_1] + \cdots + \mathbf{E}[E_k]}{k} = \mathop{\mathbf{E}}_{tr}\left[\mathop{\mathbf{E}}_{\boldsymbol{x},y}\left[L(f(\boldsymbol{x}), y) | f = \mathcal{A}(train)\right]\right]$$

where

- ▷ the outer expectation is taken over all possible training sets
- ▷ the inner expectation measures the risk of the fitted model

Crossvalidation variance estimate





- ▷ The naive variance estimate for the crossvalidation error is biased.
- ▷ The estimate usually gives smaller confidence intervals as they are.
- ▷ This must be accounted in the estimates of optimism and test error.

Theoretical explanation

Theorem. The variance of crossvalidation error $E = \frac{1}{k}(E_1 + \cdots + E_k)$ is a weighted average consisting of three components

$$\theta = \frac{1}{n} \cdot \sigma^2 + \frac{m-1}{n} \cdot \omega + \frac{n-m}{n} \cdot \gamma$$

where

- m is the number of samples in each fold, i.e., $m \approx n/k$.
- σ^2 is the average variance of true test examples.
- ω is the within-block covariance of test errors sharing the same test set.
- γ is the between-block covariance of test errors cause by the fact that
  - training set have large intersection
  - test fold is inside the training set of another split.

What else can we do with crossvalidation?

Comparing different algorithms

- ▷ We can tune hyperparameters of the algorithm
- ▷ We can estimate which algorithm on average behaves better
- ▷ We can quantify the stability of the performance ranking

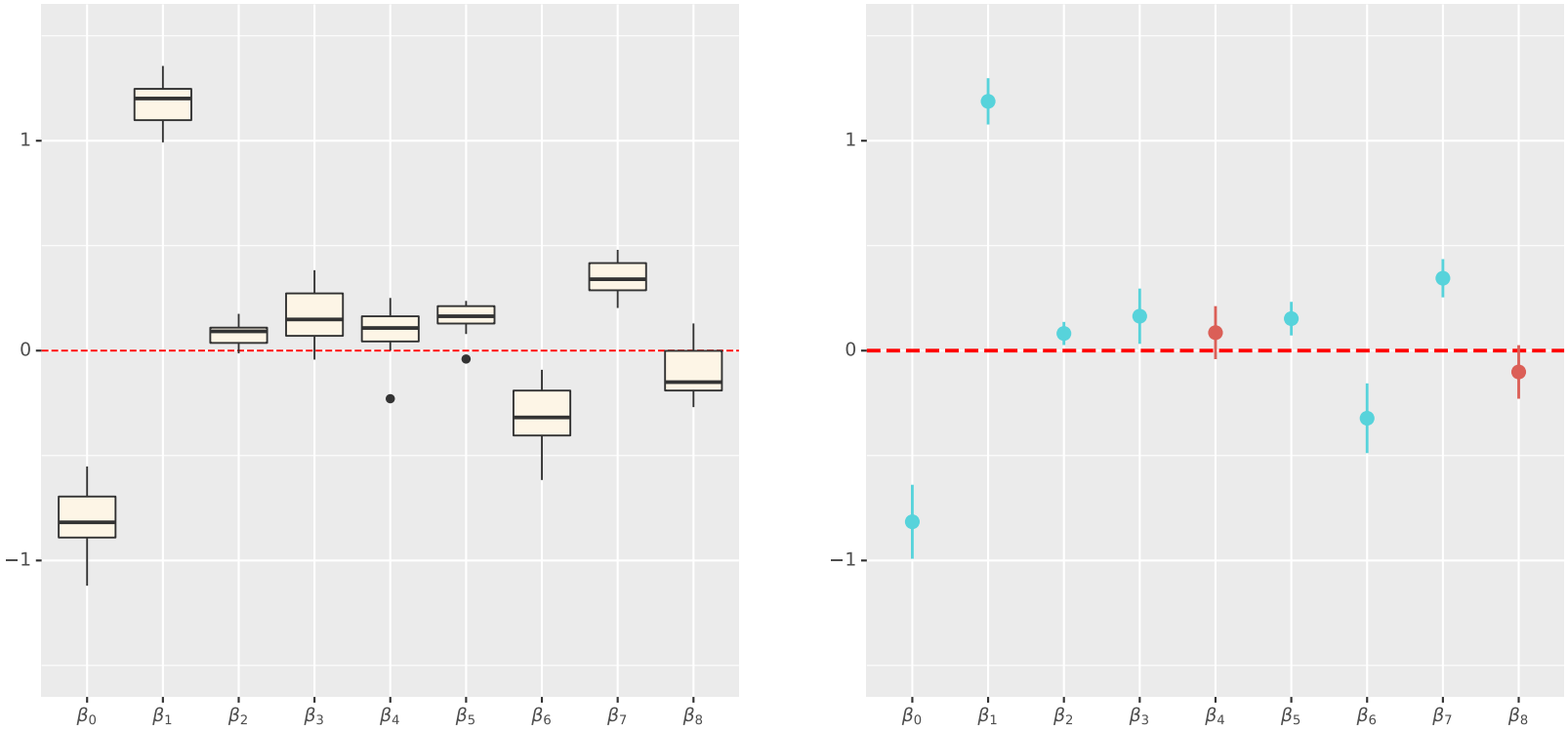
Estimating variance of model parameters

- ▷ Different folds give different parameter instances
- ▷ Parameter confidence intervals can be used for diagnostics
- ▷ Confidence intervals can be used for pruning spurious coefficients

Finding hard instances

- ▷ Different folds give different mismatches $\hat{y}_i \neq y_i$
- ▷ Corresponding problem instances $(\boldsymbol{x}_i, y_i)$ can be studied further

Estimating variance of model parameters



- ▷ The method is applicable for models with compact parametrisation.
- ▷ Each split defines a new model f_i with coefficient $\boldsymbol{\beta}$.
- ▷ The variability of a coefficient β_i shows its certainty and relevance.

Other flavours of cross validation

Exhaustive data splitting

- ▷ Leave-one-out method, leave-p-out method

Partial splitting

- ▷ K-fold cross validation for $K = 5, 10$
- ▷ Monte-Carlo crossvalidation with a fixed split ratio, e.g $1 : 9$. Same split can occur more than once
- ▷ Repeated learning testing with a fixed split ratio, e.g $1 : 9$. Same split can occur only once.

Bootstrapping as an alternative

We could use the entire date set for validation if we could get another dataset for training the model. Bootstrapping is an engineering trick to create a new dataset out of a thin air.

1. Draw N samples from the original dataset with replacement to get a *bootstrap sample* D_B, e.g. the same element can occur more than once.
2. Train the model on the bootstrap sample D_B.
3. Estimate the test error on the original dataset D.
4. Repeate the procedure 20-200 times.
5. Compute necessary statistics and visualise the results if needed.

Standard way how to use bootstrapping

Bootstrapping is mostly used to estimate optimism

- ▷ The model is trained and the training error S_i is computed.
- ▷ The test error E_i is usually computed on the entire dataset.
- ▷ Optimism is computed as $E_i - S_i$.

Note that it does not make sense to compute test error on the entire dataset as we have used some of the data to build a model. Advanced bootstrap methods like **.632 bootstrap** and **.632 bootstrap+** use only the out of training set error and later find a tradeoff between training an test error.

$$E_{\text{boot}} = 0.368 \cdot S_{\text{Train}} + 0.632 \cdot E_{\text{Out-of-training-set}}$$

Other uses of bootstrapping

Estimate the noise-tolerance of the machine learning method

- ▷ Generate a bootstrap sample.
- ▷ Corrupt with an appropriate noise.
- ▷ Train the model and estimate the performance.

Estimate the variance of model coefficients

- ▷ Generate a bootstrap sample.
- ▷ Estimate model parameters.
- ▷ Visualise parameters and compute empirical quantiles.
- ▷ Drop parameter which fluctuate around zero.