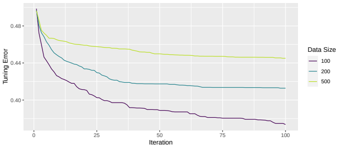
Einführung in das Statistische Lernen

Nested Resampling Motivation



Learning goals

- Understand the problem of overtuning
- Be able to explain the untouched test set principle and how it motivates the idea of nested resampling

MOTIVATION

Selecting the best model from a set of potential candidates (e.g., different classes of learners, different hyperparameter settings, different feature sets, different preprocessing, ...) is an important part of most machine learning problems.

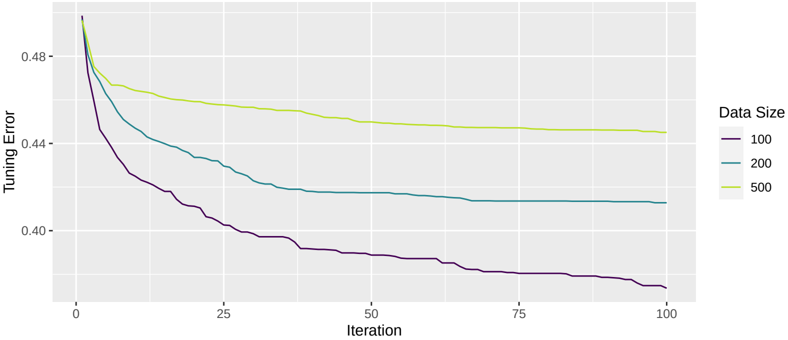
Problem

- We cannot evaluate our finally selected learner on the same resampling splits that we have used to perform model selection for it, e.g., to tune its hyperparameters.
- By repeatedly evaluating the learner on the same test set, or the same CV splits, information about the test set “leaks” into our evaluation.
- Danger of overfitting to the resampling splits / overtuning!
- The final performance estimate will be optimistically biased.
- One could also see this as a problem similar to multiple testing.

INSTRUCTIVE AND PROBLEMATIC EXAMPLE

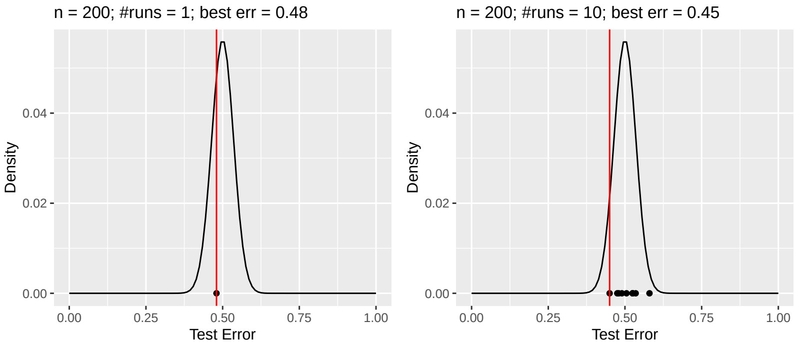
- Assume a binary classification problem with equal class sizes.
- Assume a learner with hyperparameter λ.
- Here, the learner is a (nonsense) feature-independent classifier, where λ has no effect. The learner simply predicts random labels with equal probability.
- Of course, its true generalization error is 50%.
- A cross-validation of the learner (with any fixed λ) will easily show this (given that the partitioned data set for CV is not too small).
- Now let's "tune" it, by trying out 100 different λ values.
- We repeat this experiment 50 times and average results.

INSTRUCTIVE AND PROBLEMATIC EXAMPLE



- Plotted is the best “tuning error” (i.e. the performance of the model with fixed λ as evaluated by the cross-validation) after k tuning iterations.
- We have performed the experiment for different sizes of learning data that were cross-validated.

INSTRUCTIVE AND PROBLEMATIC EXAMPLE



- For 1 experiment, the CV score will be nearly 0.5, as expected
- We basically sample from a (rescaled) binomial distribution when we calculate error rates
- And multiple experiment scores are also nicely arranged around the expected mean 0.5

INSTRUCTIVE AND PROBLEMATIC EXAMPLE

- But in tuning we take the minimum of those! So we don't really estimate the "average performance" anymore, we get an estimate of "best case" performance instead.
- The more we sample, the more "biased" this value becomes.

UNTOUCHED TEST SET PRINCIPLE

Countermeasure: simulate what actually happens in model application.

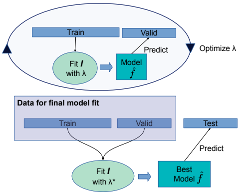
- All parts of the model building (including model selection, preprocessing) should be embedded in the model-finding process **on the training data**.
- The test set should only be touched once, so we have no way of "cheating". The test data set is only used once *after* a model is completely trained, after deciding, for example, on specific hyperparameters.
  Only if we do this are the performance estimates we obtained from the test set **unbiased estimates** of the true performance.

UNTOUCHED TEST SET PRINCIPLE

- For steps that themselves require resampling (e.g., hyperparameter tuning) this results in **nested resampling**, i.e., resampling strategies for both
  - tuning: an inner resampling loop to find what works best based on training data
  - outer evaluation on data not used for tuning to get honest estimates of the expected performance on new data

Einführung in das Statistische Lernen

Training - Validation - Test



Learning goals

- Understand how to fulfill the untouched test set principle by a 3-way split of the data
- Understand how thereby the tuning step can be seen as part of a more complex training procedure

TUNING PROBLEM

Remember:

We need to

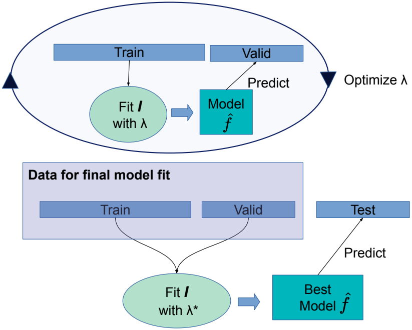
- **select an optimal learner**
- without compromising the **accuracy of the performance estimate** for that learner

– for that we need an **untouched test set**!

TRAIN - VALIDATION - TEST

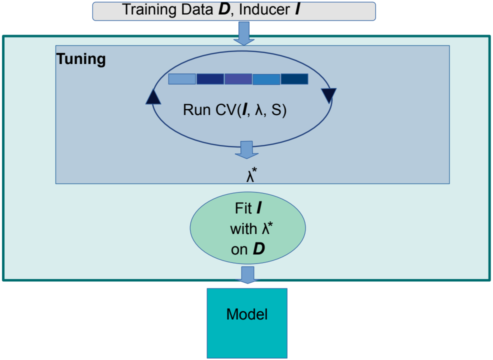
Simplest method to achieve this: a 3-way split

- During tuning, a learner is trained on the **training set**, evaluated on the **validation set**
- After the best model configuration $\lambda^\star$ has been selected, we re-train on the joint (training+validation) set and evaluate the model's performance on the **test set**.



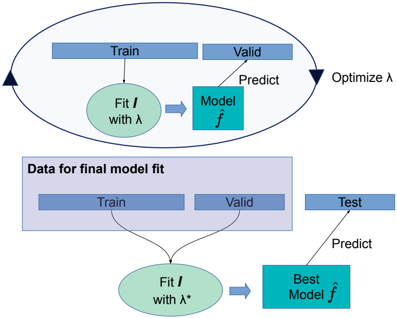
TUNING AS PART OF MODEL BUILDING

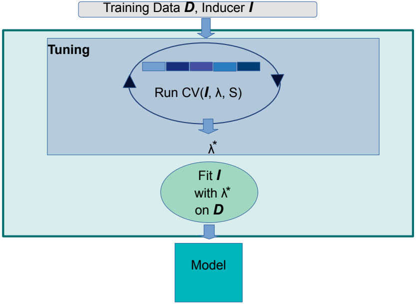
- Effectively, the tuning step is now simply part of a more complex training procedure.
- We could see this as removing the hyperparameters from the inputs of the algorithm and making it “self-tuning”.



TUNING AS PART OF MODEL BUILDING

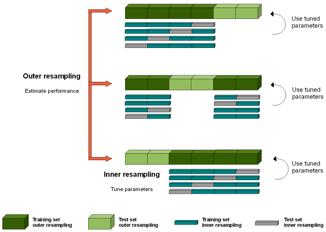
More precisely: the combined training & validation set is actually the training set for the “self-tuning” endowed algorithm.





Einführung in das Statistische Lernen

Nested Resampling



Learning goals

- Understand how the 3-way split of the data can be generalized to nested resampling
- Understand the goal of nested resampling
- Be able to explain how resampling allows to estimate the generalization error

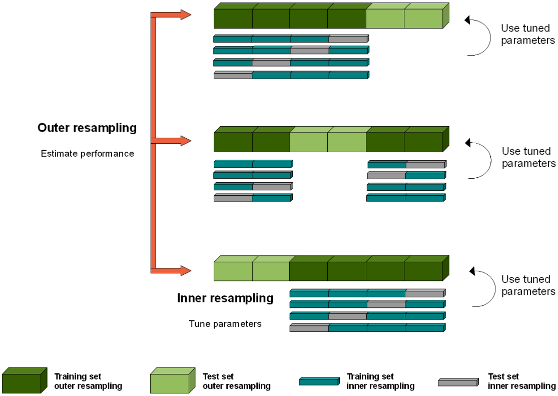
NESTED RESAMPLING

Just like we can generalize hold-out splitting to resampling to get more reliable estimates of the predictive performance, we can generalize the training/validation/test approach to **nested resampling**.

This results in two nested resampling loops, i.e., resampling strategies for both tuning and outer evaluation.

NESTED RESAMPLING

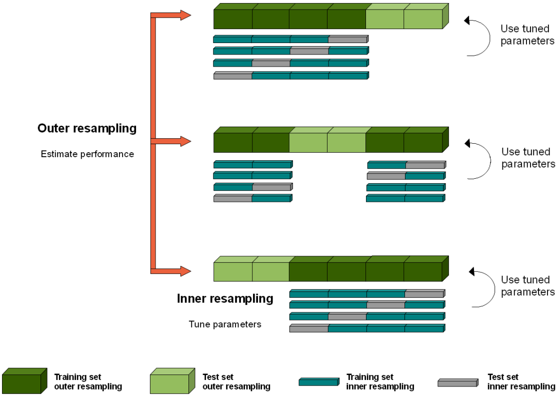
Assume we want to tune over a set of candidate HP configurations $\lambda_i; i = 1, \dots$ with 4-fold CV in the inner resampling and 3-fold CV in the outer loop. The outer loop is visualized as the light green and dark green parts.



NESTED RESAMPLING

In each iteration of the outer loop we:

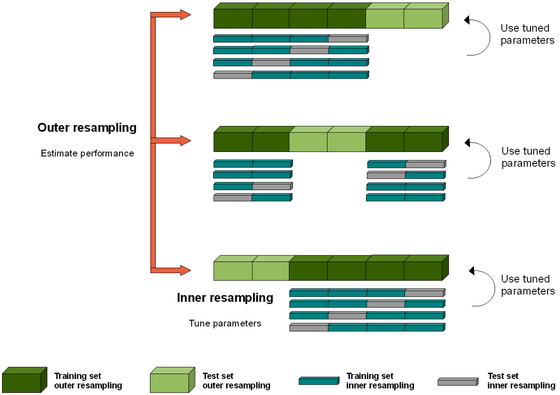
- Split off the light green testing data
- Run the tuner on the dark green part of the data, e.g., evaluate each λ_i through fourfold CV on the dark green part



NESTED RESAMPLING

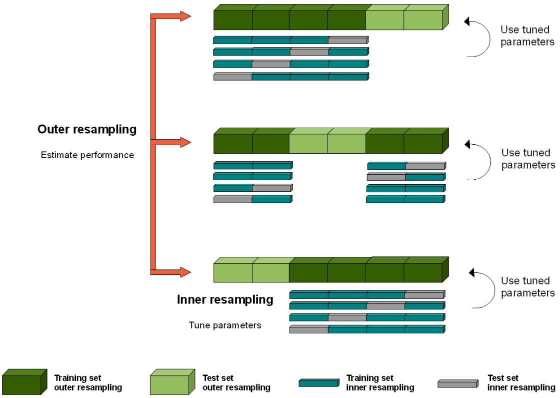
In each iteration of the outer loop we:

- Return the winning λ^* that performed best on the grey inner test sets
- Re-train the model on the full outer dark green train set
- Evaluate it on the outer light green test set



NESTED RESAMPLING

The error estimates on the outer samples (light green) are unbiased because this data was strictly excluded from the model-building process of the model that was tested on.



NESTED RESAMPLING - INSTRUCTIVE EXAMPLE

Taking again a look at the motivating example and adding a nested resampling outer loop, we get the expected behavior: