Session 1:

Machine learning recap and sampling

Andreas Bjerre-Nielsen

Agenda

1. Who teach and what is this course
2. The why, what and how of machine learing - see SDS L11/L14
3. Recap of ML
    - Regularization - see SDS L12
    - Model building - see SDS L13
    - Model validation - see SDS L13/L14
4. Generalization error

The teachers

About Andreas

Assistant professor & Head of studies (MSc Social Data Science)

Research topics

- social networks and influence
- school choice and education
- human behavior

Check out my website (https://abjer.github.io/) or my Twitter profile (https://twitter.com/andbjn)

About Ulf and Kristian

Ulf

- Post doc at SODAS (sodas.ku.dk)
- Research topics: networks (social, biological), mobility patterns
- Check out Ulf's website (https://ulfaslak.com/about.html)

Kristian

- PhD student at CEBI (https://www.econ.ku.dk/cebi/)

This course

Course motivation

The course has two teaching teaching agendas:

- Part 1: Machine learning and econometrics
    - Advanced machine learning (inference, tree- and kernel based model)
    - Combination with econometrics
- Part 2: Networks and relations data
    - Handle complex networks: friendships, banks, and much more..
    - Investigate spatial relations and objects
    - Estimate models

This course has synergies with other fields:

- Economics: game theory, mechanism design, applied econometric policy evaluation etc.

Exam

- Not project based!
- Individual exam, 24 hour take home

Why use models?

(that are not causal...)

Value of modelling

Why are models useful?

Models are pursued with differens aims. Suppose we have a regression model, $y = X\beta + \epsilon$.

- Social science:
    - They teach us something about the world.
    - We want unbiased estimate $\hat{\beta}$ and distribution
- Data science:
    - To make optimal future decisions and precise predictions, i.e. $\hat{y}$.
    - Model flexibility
        - Universal Approximation (e.g. for handwriting recognition)
    - Secondary agenda: causality (Judea Pearl etc.)

Value of modelling (2)

Which street is from a wealthy neighborhood?

Street A

Street B

Value of modelling (3)

Do you think machine can learn this difference?

- Naik, N., Kominers, S. D., Raskar, R., Glaeser, E. L., & Hidalgo, C. A. (2017). Computer vision uncovers predictors of physical urban change. Proceedings of the National Academy of Sciences, 114(29), 7571-7576.

Value of modelling (4)

Why the hype about machine learning in Social Science?

- Deep ideas: model validation, non-linear estimation
- Used to construct input data
    - E.g. parse text data, unstructured image data, network data
- Combinination with causal methods:
    - E.g. Causal Forest
- Make predictions
    - Useful in finance, macroeconomics
    - Identifying candidates for policy that are *susceptible*
        - likely compliers
        - prediction policy (outcome after treatment)

Machine learning

What do we mean by machine learning (ML)?

ML consists of two related phenomena

- supervised learning
    - assume target that is to be predicted/inferred
    - scalar/number > regression
    - categorical > classification
- unsupervised learning (week 3)
    - no target for classification
    - includes clustering, component decomposition

Supervised learning models

Individual model

- Linear/logistic regression
    - no regularization (like econometrics)
    - with regularization (week 1/next slides)
- Tree and kernel based methods (week 2)

Combining models

- Ensemble, bagging (week 2)

Regularization

Regularization (1)

Why do we regularize?

- To mitigate overfitting > better model predictions

How do we regularize?

- We make models which are less complex:
    - reducing the **number** of coefficient;
    - reducing the **size** of the coefficients.

Regularization (2)

What does regularization look like?

We add a penalty term our optimization procedure:

$$\arg\min_\beta \underbrace{E[(y_0 - \hat{f}(x_0))^2]}_{\text{MSE}} + \underbrace{\lambda \cdot R(\beta)}_{\text{penalty}}$$

Introduction of penalties implies that increased model complexity has to be met with high increases precision of estimates.

Regularization (3)

What are some used penalty functions?

The two most common penalty functions are L1 and L2 regularization.

- L1 regularization (***Lasso***): $R(\beta) = \sum_{j=1}^{p} |\beta_j|$
    - Makes coefficients sparse, i.e. selects variables by removing some (if λ is high)
- L2 regularization (***Ridge***): $R(\beta) = \sum_{j=1}^{p} \beta_j^2$
    - Reduce coefficient size
    - Fast due to analytical solution

To note: The *Elastic Net* uses a combination of L1 and L2 regularization.

Model building

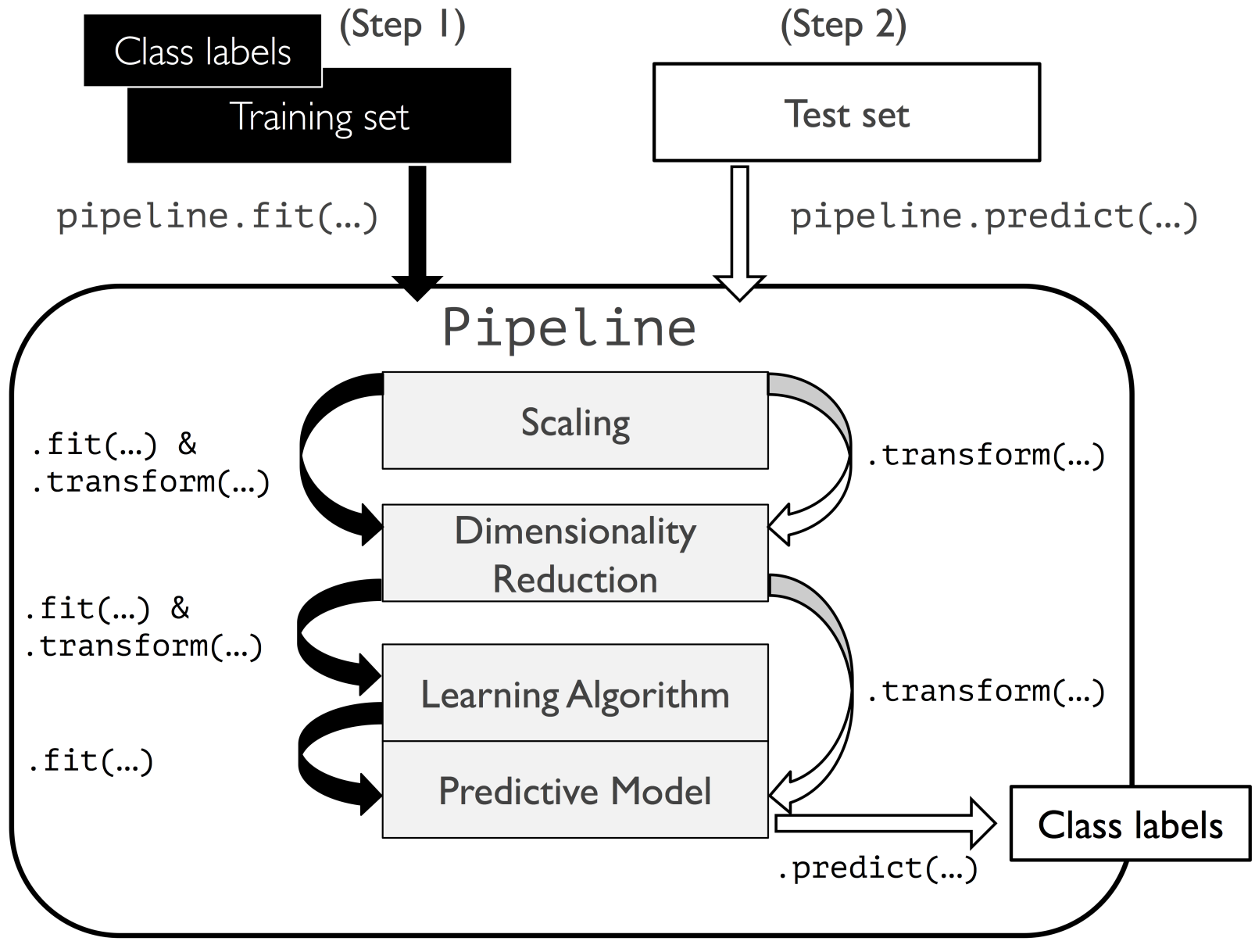
Model pipelines (1)

Is there a smart way to build supervised ML models?

Build pipeline:

- One step: preprocess data, estimate model
- Ensures good practice - we only build model using training data.
    - No data leakage

Model pipelines (2)


Class labels
(Step 1)
Training set
(Step 2)
Test set
pipeline.fit(...)
pipeline.predict(...)
Pipeline
Scaling
.fit(...) &
.transform(...)
.transform(...)
Dimensionality
Reduction
.fit(...) &
.transform(...)
Learning Algorithm
.transform(...)
.fit(...)
Predictive Model
Class labels
.predict(...)

Model validation

Model performance

How do check our model fit?

- One way is compute various measures of fit (R^2, accuracy etc.).
    - Issue: adding more variable $\Rightarrow$ higher R^2

How is this solved?

- Use some of our sample for model evaluation.
- Stagegy: divide into training data for estimation; remaining to test data for evaluation.

Classification performance metrics

How can we measure the performance in classification problems?

- Basic idea: how many correct. But how? Many ways to do this.
- More about this in exercises.

Calibrating the model

Does machine learning work out of the box?

- In some cases ML works quite well out of the box.
- Often ML requires making careful choices.
    - Note that automated machine learning packages and services exist.

Which choices are to be made?

- We need to pick model building **hyperparameters**.
- E.g. λ for Lasso, Ridge.

Model validation (1)

How do we measure our model's performance for different hyperparameters?

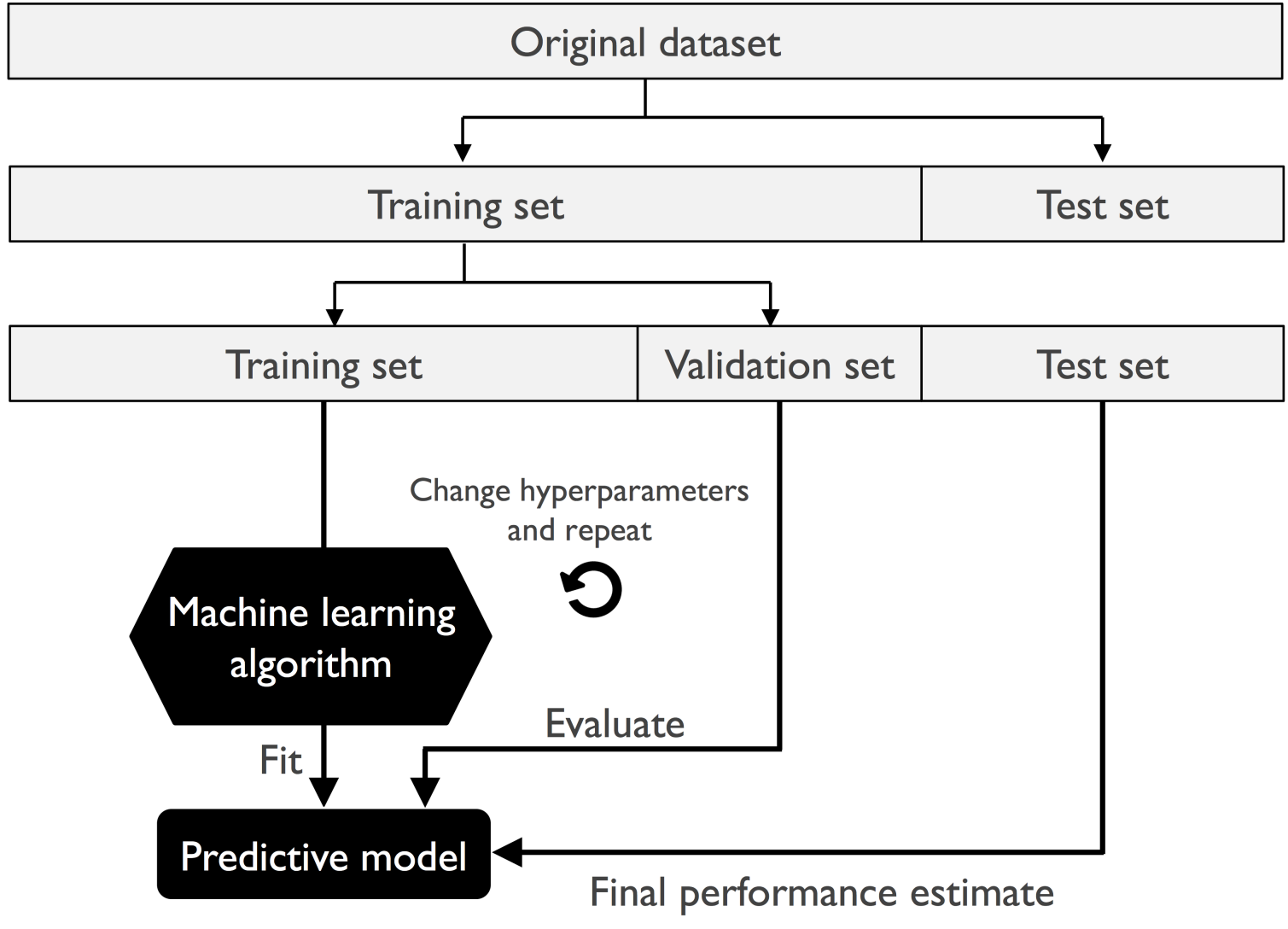
- Remember we cannot use the test set.

Could we somehow mimick what we do with test data?

- Yes, we can split the remaining non-test data into training and validation data:
    - we train model for various hyperparameters on training data;
    - pick the hyperparameters which performs best on validation data.

Model validation (2)

The non-test data is split into training and validation


Original dataset
Training set
Test set
Training set
Validation set
Test set
Change hyperparameters
and repeat
Machine learning
algorithm
Evaluate
Fit
Predictive model
Final performance estimate

Cross validation

The holdout method

How do we got the more out of the data?

We reuse the train-test data split in reverse:

- Rotate which parts of data is used for test and train.

Advantage: We test on all the data; little extra computation.

Disadvantage: Depends on the split; still only 50 pct. used for training model.

Leave-one-out CV

How do we got the most of the data?

Procedure:

- Each single observation as test data; remaining for training.
- Also known as Jackknife

Advantage: Robust, does not depend on random numbers!

Disadvantages:

- Very computing intensive: One model per observation.
- Not good for hypothesis testing.

K fold method (1)

How do balance computing time vs. overfitting?

We split the sample into K even sized test bins.

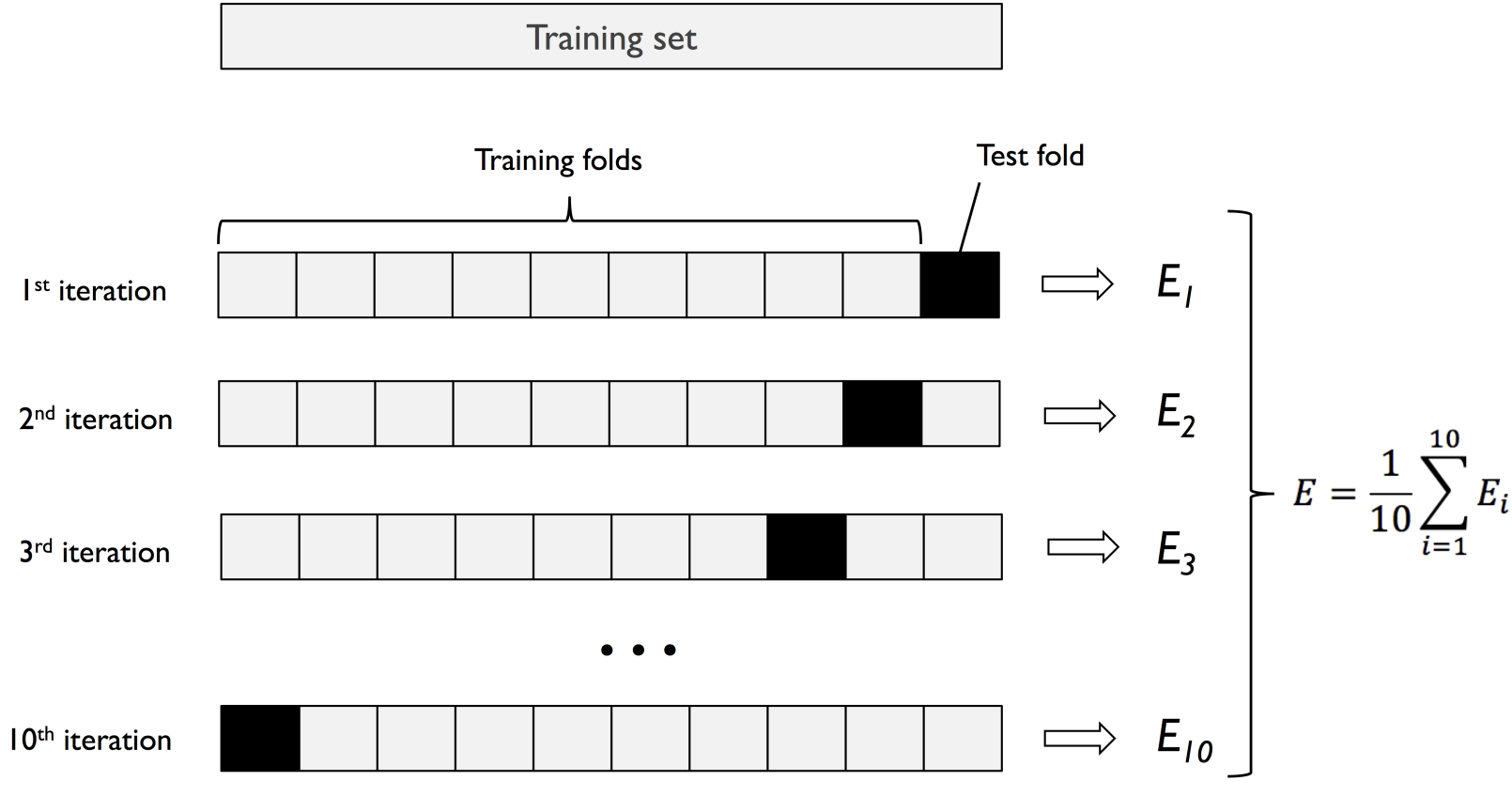
- For each test bin k we use the remaining data for training.

Advantages:

- We use all our data for testing.
- Training is done with 100-(100/K) pct. of the data, i.e. 90 pct. for K=10.

K fold method (2)

In K-fold cross validation we average the errors.



Advanced model validation

Nested cross validation

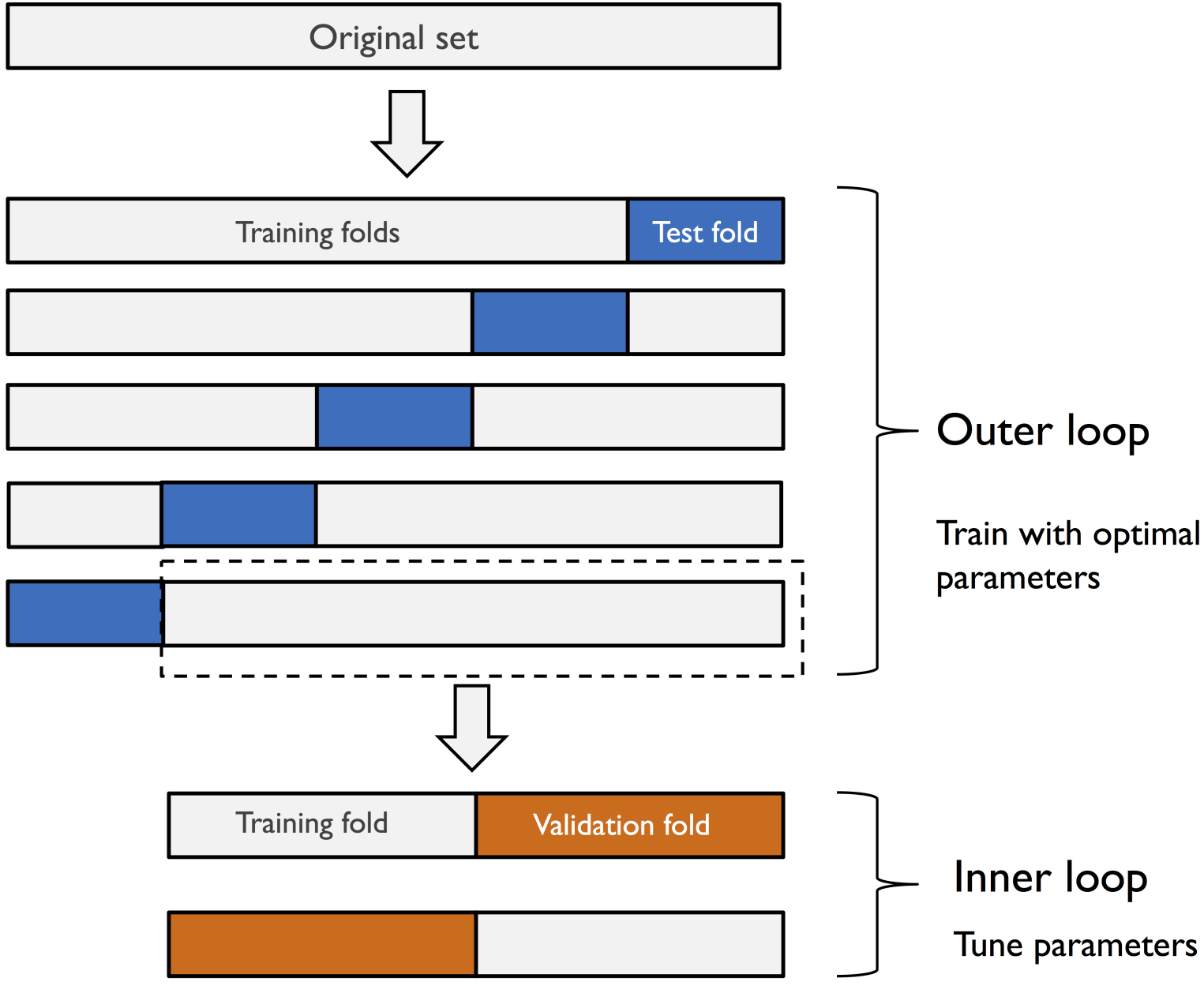
What should we do if we have more than one model that we test? Is it okay to take the one that performs best on the test set?

- No, the performance of model may be biased.

Solution:

- idea: perform cross-validation (CV) multiple times on different parts of data.
- **outer CV**:
    - split data like in cross validation
    - for each training dataset perform **inner CV** to tune hyperparameters

Nested cross validation (2)


Original set
Training folds
Test fold
Outer loop
Train with optimal parameters
Training fold
Validation fold
Inner loop
Tune parameters

Nested cross validation (3)

Improved measure of the uncertainty by re-doing cross-validation again and again.

- called **Repeated k-fold Cross validation**.

Nested resampling

If we want to make reproducible research we should make repeated samples. Some possibilities:

- Subsampling:
    - We randomly split data into train and test. Train data obs. are unique.
- The bootstrap:
    - Draw training data with replacemtent from all data - same sample size.
    - Unused data will be test data.
    - Issue: Binder (2008) "Adapting prediction error estimates for biased complexity selection in high-dimensional

Generalization error

Measure

How do we expect our model to perform on unseen data? This is sometimes known as the **generalization error**.

How can we measure the generalization error? We can compute the variability on the test set.

We can use the error on the test set(s) to bound the error on unseen data

- note: requires that we assume that new and existing data come from the **exact same distribution**.

Getting scientific

Often we have a dataset and we want to test whether one of two models are better. We want to do this scientifically. This may be whole model building procedure including optimization of hyperparameters etc.

How can do this?

Model inference: single test set

Situation: we have a model that we have trained/estimated on some part of the data (optimized etc.). The other part is NEVER tried. We test our performance on this "new" dataset.

We can compare the two models with standard tests, e.g. we pair observations from the models with

- paired t-tests
- Wilcoxon signed-rank test.
- bootstrap / permutation stuff

Problem: requires that there is **no** variability from training data

- strong assumption, especially for non-linear e.g. tree based and neural
- sometimes known as model stability

Model inference: multiple test sets

If we want to make sure that we take into account model tuning uncertainty we should use more than one test set. These sets can be drawn either from:

- subsampling, e.g. n random, independent subsets
- repeatedly dividing data into k folds p times

Should we just compare the distribution of mean performance across bins (i.e. folds/subsamples)?

Model inference: solution with multiple test sets

If we want to make sure that we take into account model tuning uncertainty we should use cross validation. How can we compare algorithms across cross validation bins?

Problem: bias from training data may be too large! Nadeau and Bengio (2003) argues that we need to correct the std. of distributions as follows.

- Step 1: compute J iterations of subsampling and let the set of subsamples be denoted $\mathcal{J}$.
- Step 2: compute the paired mean difference, $\hat{\mu}_j$ for scoring function for each bin j.
- Step 3: compute the mean and standard error across bins ($\hat{\mu}_{\mathcal{J}}$, $S^2_{\hat{\mu}_{\mathcal{J}}}$)
- Step 4: compute the corrected standard error across bins = (($\frac{1/|J|}{+}$ $\frac{n_{test}}{n_{train}}$) $\cdot$ $S^2_{\hat{\mu}_J}$)

Recommended by review in [Bouckaert, Frank (2004) (https://link.springer.com/chapter/10.1007/978-3-540-24775-3_3)](https://link.springer.com/chapter/10.1007/978-3-540-24775-3_3).