Session 1:

Machine learning recap and sampling

Andreas Bjerre-Nielsen

Agenda

1. This course
2. The why, what and how of machine learing - see SDS L11
3. Recap of ML
    - Regularization - see SDS L12
    - Model validation - see SDS L12-13
    - Model building - see SDS L13
    - Decision trees - see SDS L14
    - Ensemble learning - see SDS L14
4. Advanced model validation

This course

Course motivation

The course has two teaching teaching agendas:

- Learn tools for working with data
    - Handle complex networks: friendships, banks, and much more..
    - Process unstructured data, e.g. text, image, into model data
    - Investigate spatial relations and objects
- Learn modelling methods
    - Advanced machine learning (inference, deep learning)
    - Identification in networks and with spatial data

This course has synergies with other fields:

- Economics: game theory, mechanism design etc.
- Sociology: analysis of discourse

Course outline

We teach in four blocks:

- Block 1, machine learning: model validation, neural networks
- Block 2, networks: structure, behavior and identification
- Block 3, spatial data: theory and methods
- Block 4, text data: natural language processing

Each block builds on one or more of the previous.

Course projects (1)

- Rules:
  - At most four people
  - Anything goes (next slide)
  - More details on GitHub
- Advice:
  - Think about deep research questions. Which hypotheses?
  - Consider method: prediction or identification?
  - Talk to us in class or Kristian

Course projects (2)

- The world is your oyster:
    - Last year scraping: central banks, conflict data, twitter/reddit, football statistics

What about all the data which is not open online?

- Private data may be even more awesome.
    - Get in touch with your employer
    - Contact organizations, firms, goverment

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)

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. Athey, Wager (2018), Chernozhukov et al. 2017
- Make predictions
  - Useful in finance, macroeconomics
  - Identifying susceptible candidates for policy

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
    - no target for classification
    - includes clustering, component decomposition

Common supervised learning models

Individual model

- Linear/logistic regression (SDS)
    - no regularization (like econometrics)
    - with regularization > next slides
- Tree based methods (SDS/week 1)
- Deep learning (week 2-4)

Combining models

- Ensemble, bagging (SDS/week 1)

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 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.

Measuring the problem

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?

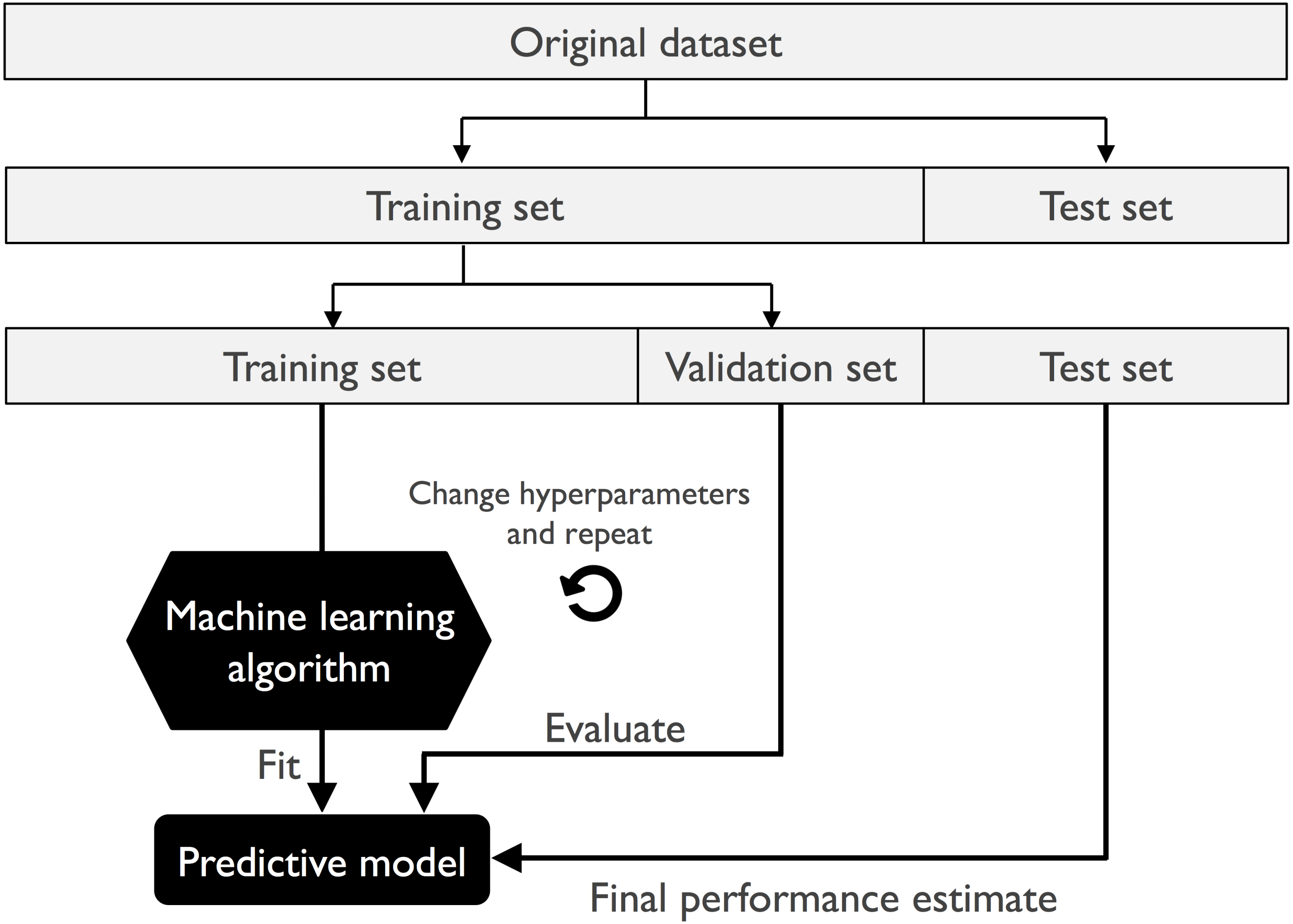
- Remember we cannot use the test set.

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

Model building

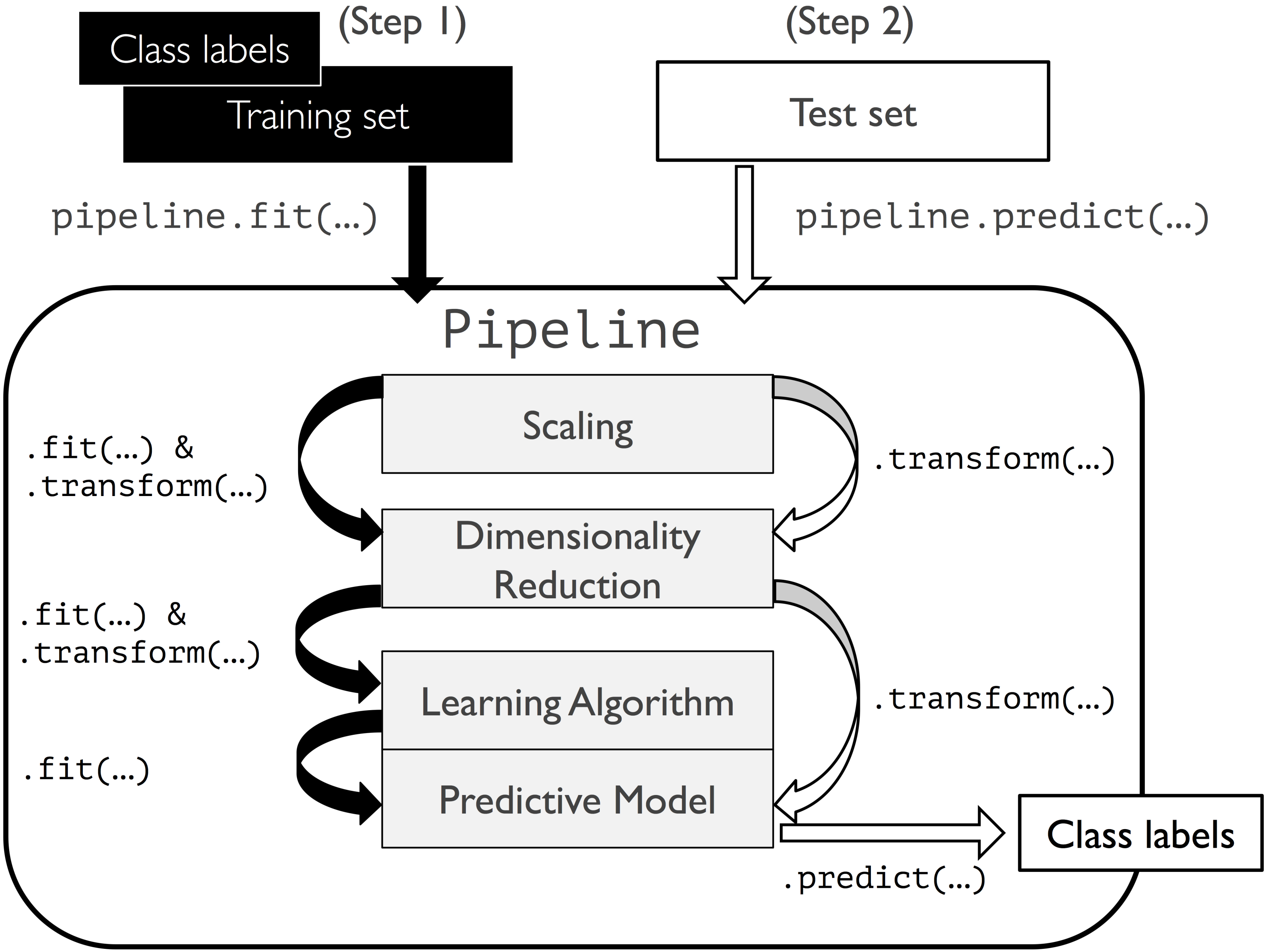
Model pipelines (1)

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.

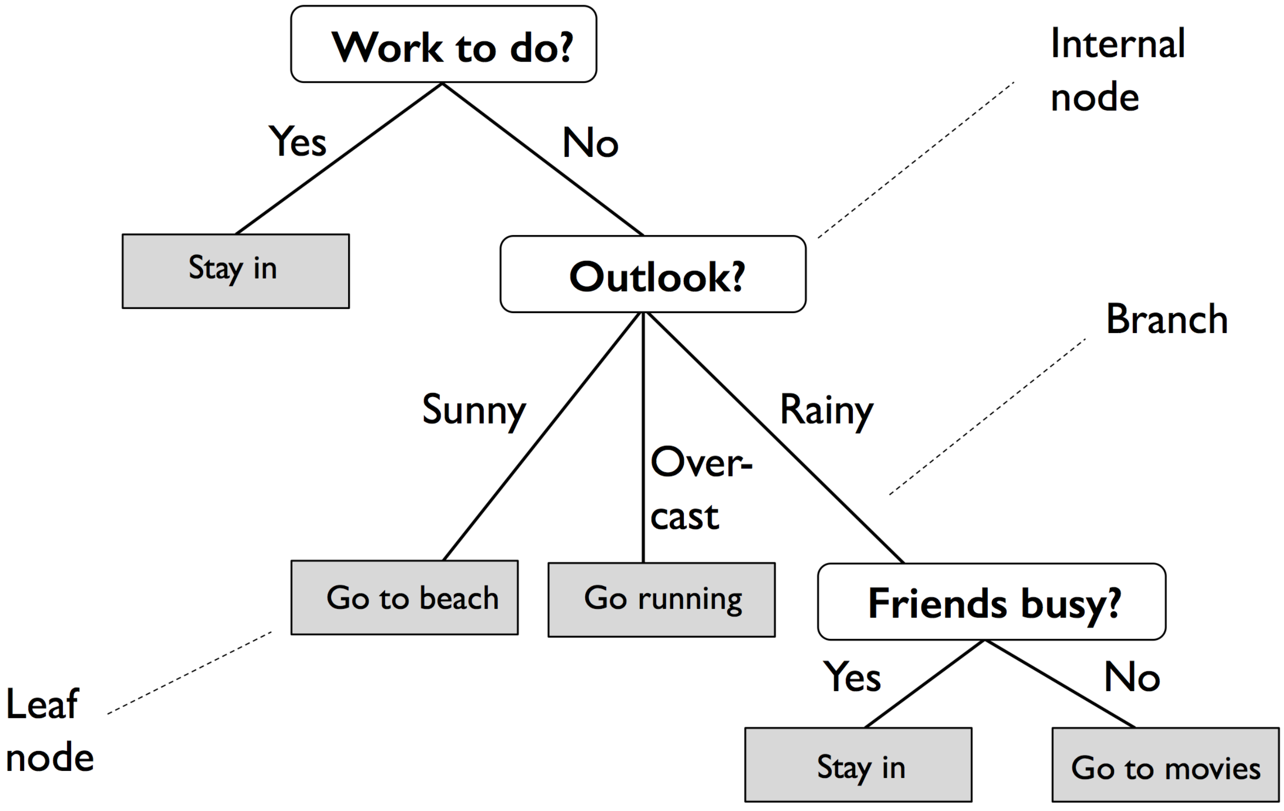
Model pipelines (2)


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

Decision trees

A hierarchal structure

What does a decision tree look like?



Evaluating decision trees

What can we conclude about the decision trees?

- Can fit anything ~ Universal Approximation
  - *little* underfitting (~low bias)
  - **LARGE** overfitting (~large variance)
- What can we say about linear and logistic regression?

Ensemble learning

Ensemble of models

Models can be combined into one; this is called **ensemble learning**.

- Regression: use mean/median over model predictions
- Classification: use mode of model predictions (i.e. most common)

The wisdom of the crowd (1)

What can we do to reduce overfitting of a decision tree?

- We create multiple trees where for each tree
    - We draw a subsample of observation
    - We draw a subsample of features

These combined decision trees are called a random forest.

- The predicted value is the mode (most common) predicted by the trees
- Extension to regression where the mean over trees is computed.
- Works almost like magic out of the box (has hyperparamters).

The wisdom of the crowd (2)

The underlying process of Random Forest is called **bagging**, short for *bootstrap aggregating*. Possible dimensions:

- Bagging *observations*, i.e. 1st dimension of data
- Bagging *features* (variables), i.e. 2nd dimension of data

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.

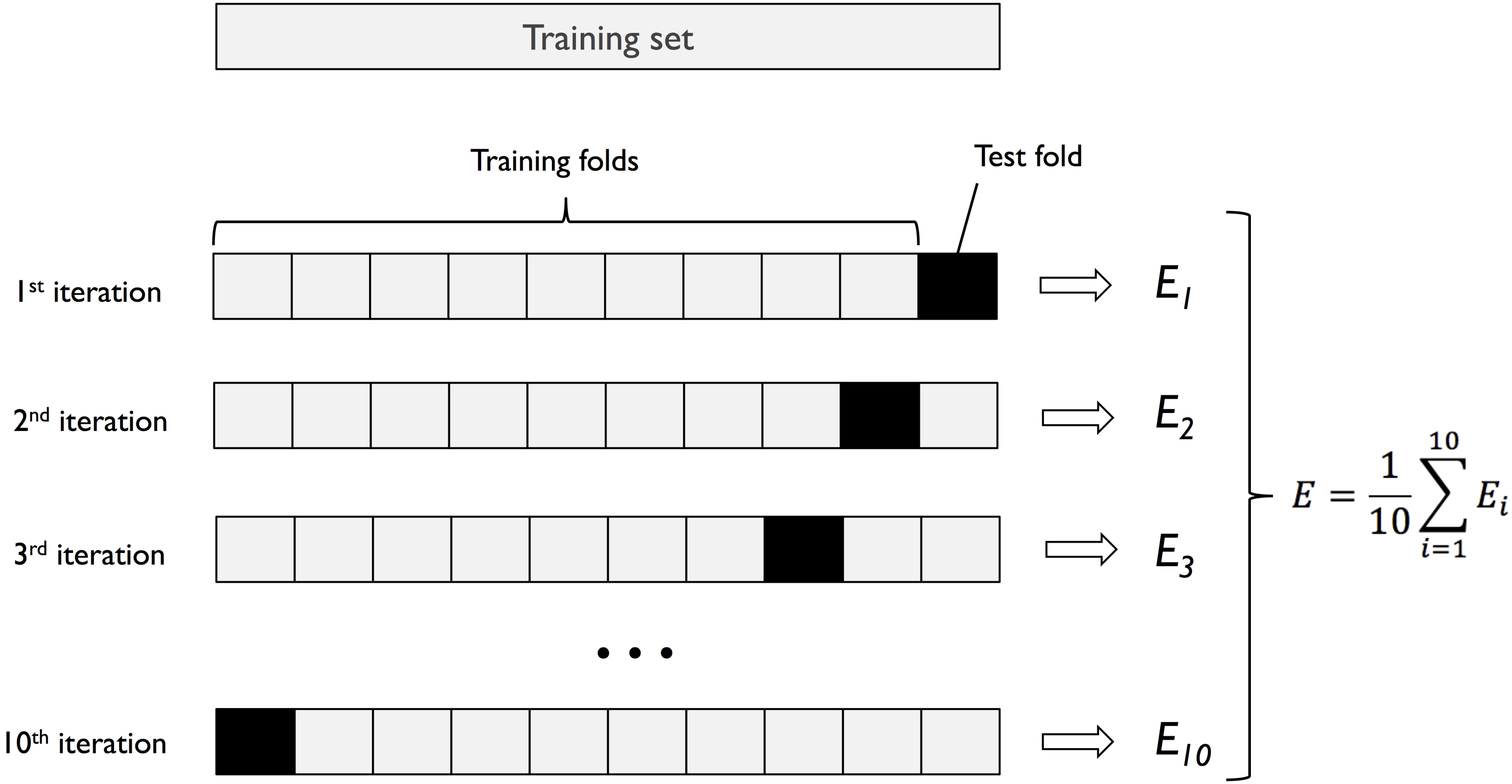
- For each test bin k we use the remaining data for training.

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.



$$E = \frac{1}{10}\sum_{i=1}^{10} E_i$$

Advanced model validation

Nested cross validation

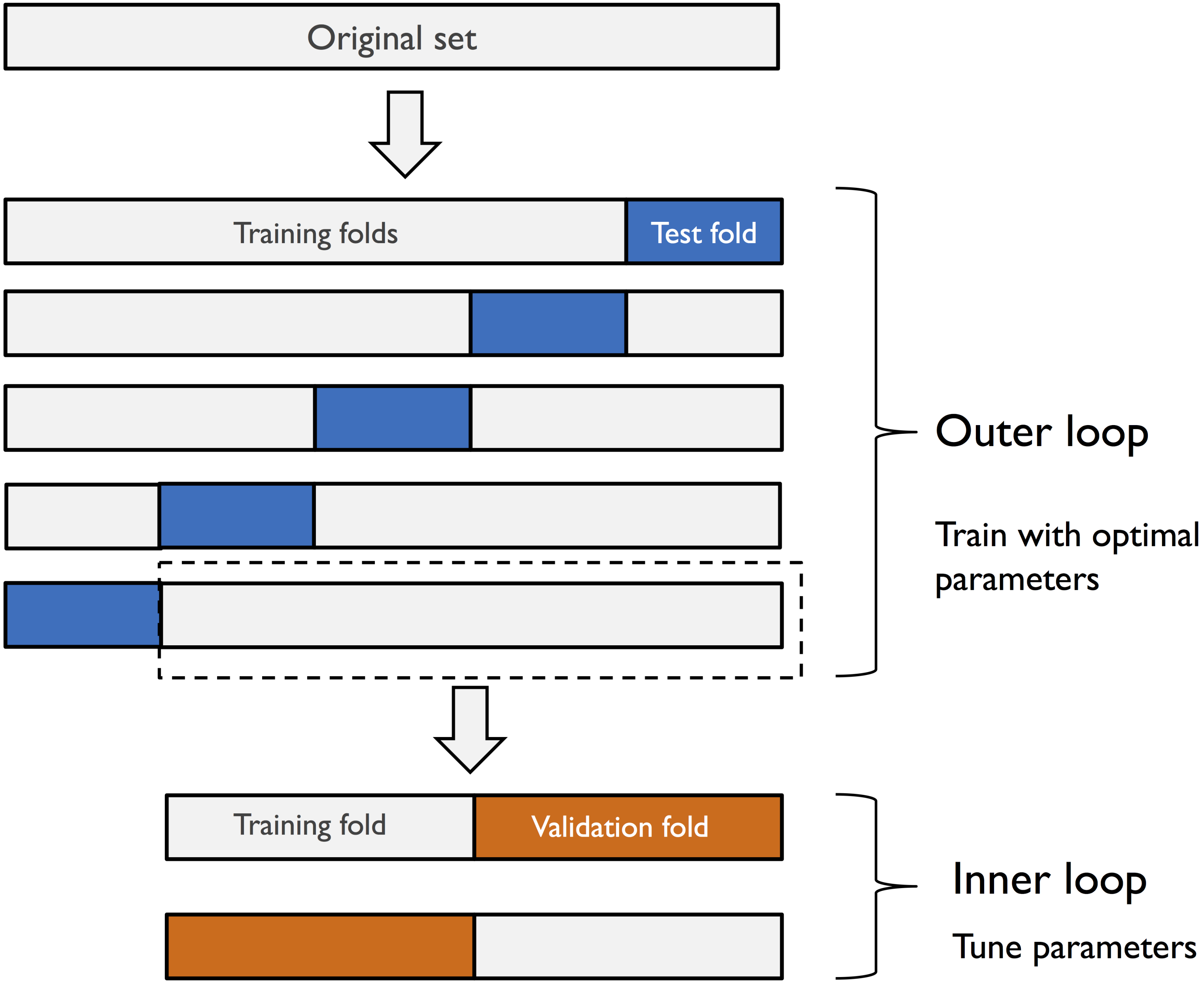
What should we do if we have more than one model that we test? Is it okay to take the one that performs best on the test set?

- 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

Model comparison tests

If we want to make reproducible research we should repeat do this many times.

- Repeated subsampling: use corrected resampled t-test
    - Nadeau, Claude, and Yoshua Bengio. "Inference for the Generalization Error." Machine Learning 52.3 (2003): 239.
- 5x2 nested cross validation: use resampled t-test
    - Dietterich, Thomas G. "Approximate statistical tests for comparing supervised classification learning algorithms." Neural computation 10.7 (1998): 1895-1923.