Cross Validation

Benoit Gaüzère

INSA Rouen Normandie - Laboratoire LITIS

October 23, 2023

Introduction

How to learn a "good" model ?

- We want good performance
- Simple as possible
- Able to predict unseen data

Empirical Risk

Error on learning set

- Empirical risk:

$$R_emp(f) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(\mathbf{x}_i), y_i)$$

- $\mathcal{L}$ evaluates the performance of prediction $f(\mathbf{x}_i)$
- Error is computed on the training set
- The model can be too specialized on this particular dataset

Generalisation

Tentative of Definition

- Ability of the model to predict well unseen data
- Hard to evaluate
- Real objective of a model

Regularisation

- Regularization term control the model
- Balances between empirical risk and generalization ability
- Need to tune the balance (λ)

How to evaluate to ability to generalize ?

Evaluate on unseen data

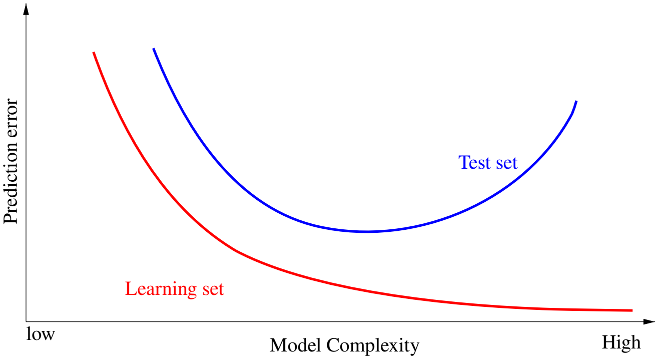
- Define and isolate a test set
- Evaluate on the test set

Bias

- Avoid to use same data in train and test
- Test set must be totally isolated

Overfitting vs Underfitting

- Overfitting: low R_{emp}, high generalization error
- Underfitting: high R_{emp}, medium generalization error



Hyperparameters

Parameters outside the model

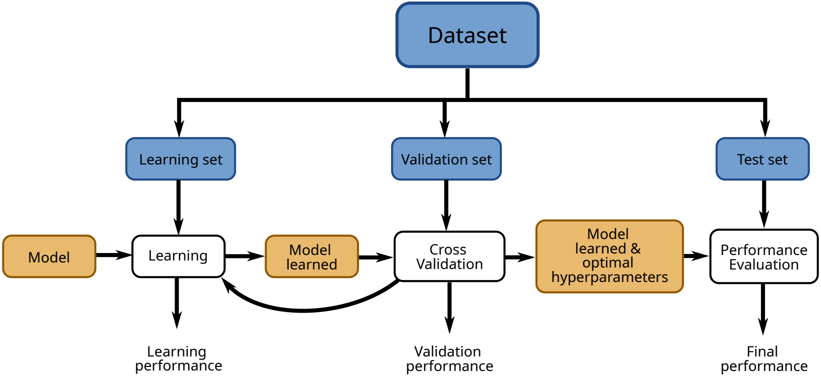
- Some parameters are not learned by the model
- They are “hyperparameters” and must be tuned
- ⚠ Tuned on data outside the test set
- Example: λ in Ridge Regression

How to tune the hyperparameters ?

Validation set

- Split train set into validation and learning set
- Learn model parameters using the learning set
- Evaluate the performance on validation set
- Validation set simulates the test set, aka unseen data

General framework



Validation strategies

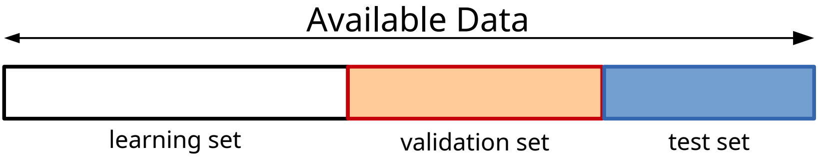
How to split validation/training set

- Need of a strategy to split between training and validation sets
- Training is used to tune the parameters of the model
- Validation is used to evaluate the model according to hyperparameters

Train/Validation/Test

Single split

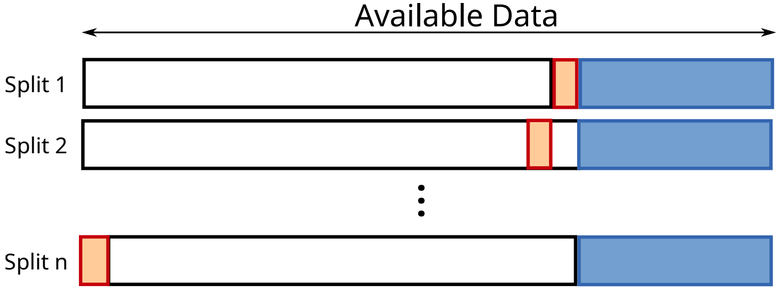
- ✚ An unique model to learn
- ▬ May be subject to split bias
- ▬ Only one evaluation of performance



Leave one out

N splits

- N models to learn
- Validation error is evaluated on 1 data



KFold Cross validation

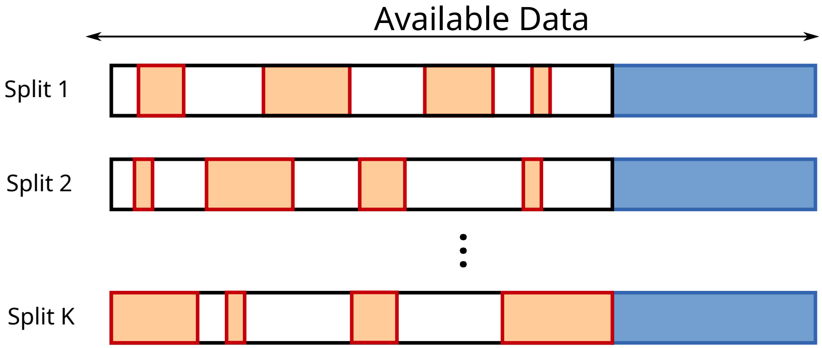
K splits

- K models to learn
- Validation error is evaluated on N/K data
- Some splits may be biased

Shuffle Split Cross validation

K splits

- Learn/Valid sets are randomly splited
- ✚ K models to learn
- ✚ Avoid bias
- ▬ Some data may not be evaluated



With scikit-learn

- sklearn.model_selection.train_test_split
- sklearn.model_selection.KFold
- sklearn.model_selection.ShuffleSplit
- sklearn.model_selection.GridSearchCV

Recommandation

Size of splits

- How many splits ?
- How many element by split ?
- Depends on the number of data
- Tradeoff between learning and generalization

Stratified splits

- Splitting may induce to imbalanced datasets
- Take care that the distribution of $\boldsymbol{y}$ is the same for all sets

Conclusion

- A good protocol avoid bias
- Test is never used during tuning of (hyper)parameters
- Perfect protocol doesn't exists