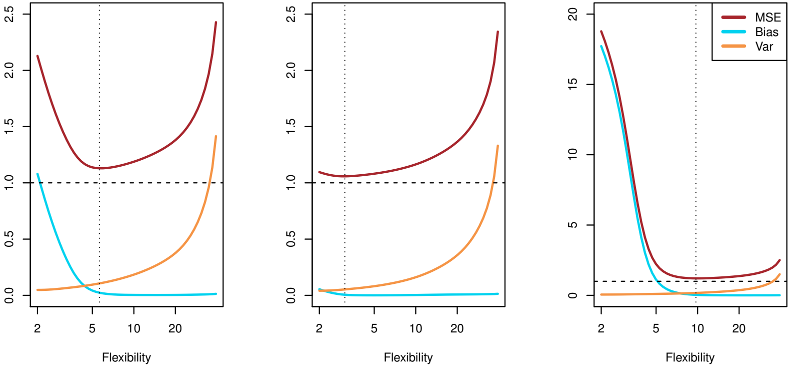
Model Assessment and Hyperparameter Tuning

Johanni Brea

Einführung Machine Learning

GYMINF 2022

Which Model Is Best?



The best model has smallest test MSE[†].

What if we do not know the true test error?

[†] Here the test MSE is exactly computed.

Some of the figures in this presentation are taken from “An Introduction to Statistical Learning, with applications in R” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Basic Idea: Split Available Data

Assumption: Hyperparameters fixed
Goal: Find best parameters

Use all data to estimate the parameters.

Assumption: Hyperparameters unknown
Goal: Find best hyperparameters

Split data into training and validation set(s).
Estimate parameters on the training set(s).
Estimate test errors for all hyperparameter choices on the validation set(s).
Select hyperparameters with lowest test error.

Assumption: Hyperparameters fixed
Goal: Estimate test error

Split data into training and test set(s).
Estimate parameters on the training set(s).
Estimate test error on the test set(s).

Assumption: Hyperparameters unknown
Goal: Estimate test error

Split data into training, validation and test set(s). Estimate parameters on the training set(s). Select hyperparameters with lowest test error estimated with the validations set(s). Estimate test error on the test set(s).

Table of Contents

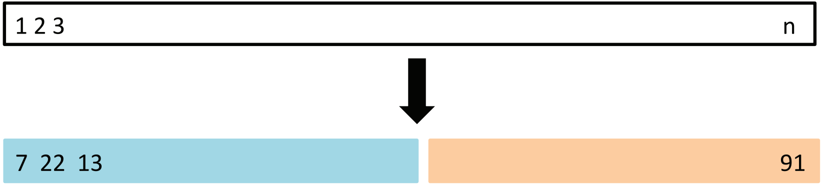
1. Training, Validation and Test Set

2. Cross-Validation

3. Tuning Models

4. A Recipe for Supervised Learning

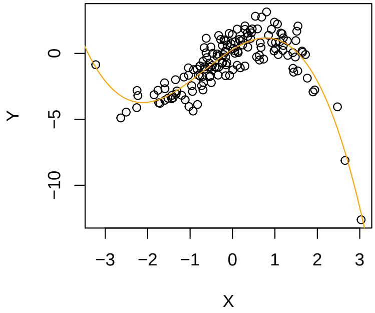
The Validation Set Approach

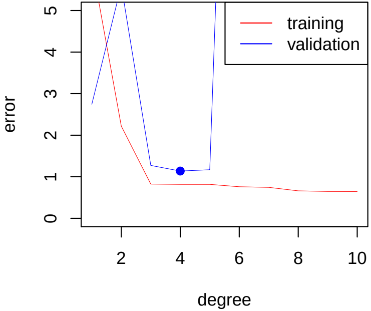


training set

validation set

Validation Set Approach Applied to Artificial Data





$$Y = 0.3 + 2X - 0.8X^2 - 0.4X^3 + \epsilon$$

polynomial fits with different degrees d

optimal $d = 4$

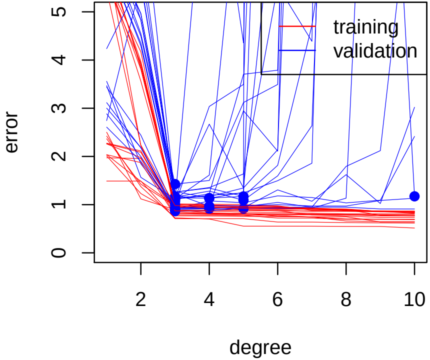
Training, Validation and Test Set

- **Training Set**: Subset of the full data used to find the parameters.
- **Validation Set**: Held-out subset of the full data used for model selection, i.e. finding the hyper-parameters.
- **Test Set**: Held-out subset of the data to estimate the test error of the best model.

Machine Learning Competitions e.g. on kaggle.com

1. Start of the competition: Participants obtain a data set, but not the test set.
2. Participants split the data set into training and validation sets as they want to fit the parameters and tune the hyper-parameters.
3. End of the competition: Organizers evaluate all submitted solutions on the test set.

Drawback of Validation Set Approach



The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set ⇒ high variance in model selection.

Quiz

Which of the following statements are correct?

- After finding in a model comparison the best performing model on the validation set, we compute the error on the validation set and the error on the test set.

  1. The test error is usually larger than the validation error.
  2. Test error and validation error are roughly equal.
  3. The test error is usually smaller than the validation error.

- The error on unseen data tends to be lower for a model trained on all available data compared to a model trained on a training set with 80% of all available data.

- To have the most accurate models for model comparison it is acceptable to fit the models on all available data and compare them on the validation set consisting of 50% of the data.

Table of Contents

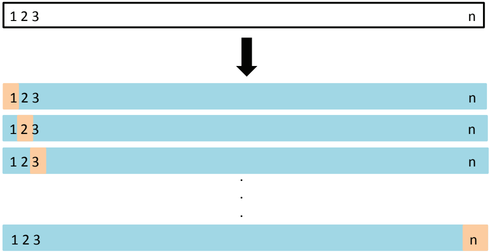
1. Training, Validation and Test Set

2. **Cross-Validation**

3. Tuning Models

4. A Recipe for Supervised Learning

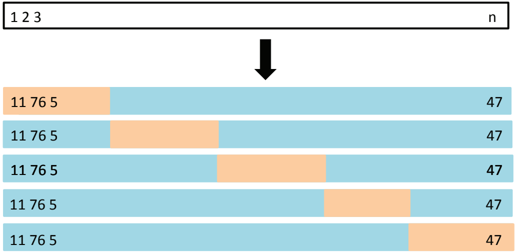
Leave-One-Out Cross-Validation (LOOCV)



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i \qquad \mathrm{MSE}_\mathrm{i} = (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the prediction obtained by fitting without (x_i, y_i).

K-Fold Cross-Validation



1 2 3 n

11 76 5 47

11 76 5 47

11 76 5 47

11 76 5 47

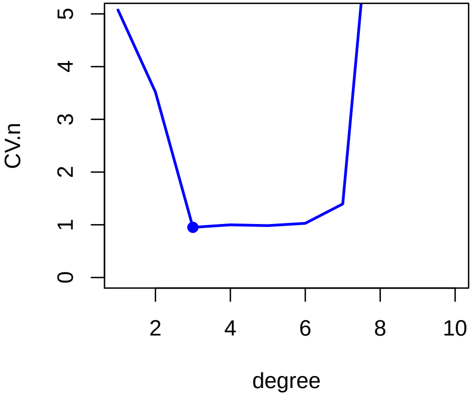
11 76 5 47

$$\mathrm{CV}_{(k)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k \qquad \mathrm{MSE_k} = \frac{1}{n_k} \sum_{i \in C_k} (y_i - \hat{y}_i)^2$$

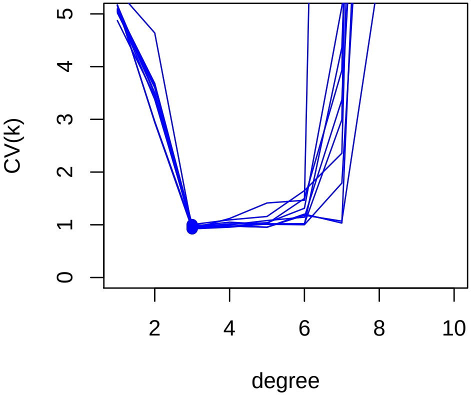
where $\hat{y}_i$ are predictions obtained by fitting without the data in part C_k.

Cross-Validation Applied to the Artificial Data

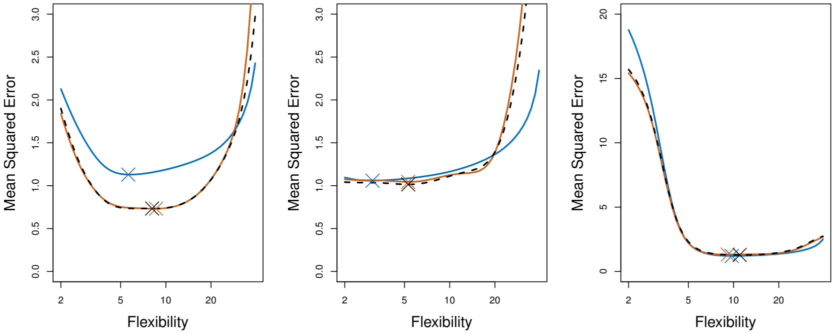
Leave-one-out Cross Validation



5-Fold Cross Validation



True versus Estimated Test Error



LOOCV (black dashed) and 10-fold CV (orange solid) find almost the same optimal flexibility as the true test error (blue). Crosses indicate the minima of the MSE curves.

Further Considerations

- The choice of the number of folds K is somewhat arbitrary.
  Typical choices are $K = 5$ or $K = 10$.
- LOOCV has higher computational costs, since n fits are made instead of K
  (Except for least squares linear or polynomial regression.)
- To estimate the test error with the validation set approach: fit the winner of model comparison to all data except the test set and evaluate it on the test set.
- To estimate the test error with cross-validation (nested cross-validation): repeat the approach above for multiple folds.
- To have the best model for predictions of future data: fit the model on all data you have.
- If you do not care about an estimate of the test error, you run cross-validation on the full data, without first splitting off a test set.

Cross-Validation on Classification Problems

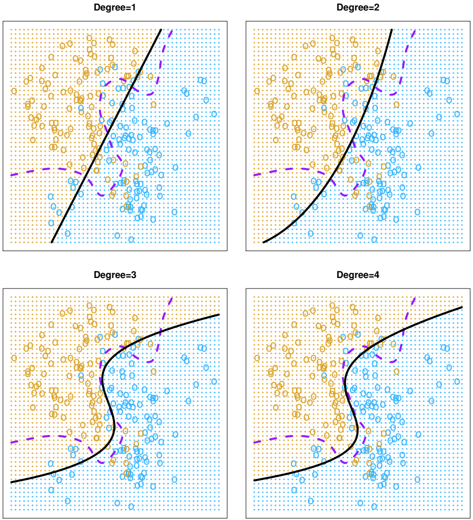
Instead of the log-likelihood one can also use the average misclassification rate on the held-out sample for cross-validation in classification problems.

$$\mathrm{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

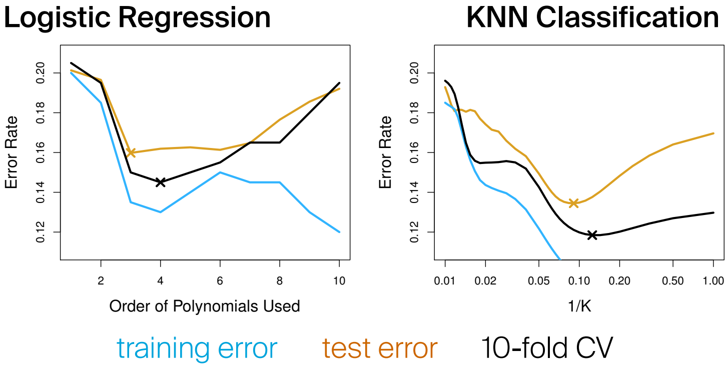
Optimal decision boundary (purple)

Estimated decision boundary (black)

for polynomial degrees 1-4.



Cross-Validation on Classification Problems



Logistic regression maximizes the likelihood of the parameters β_i given the data $\Rightarrow$ training likelihood is monotonically increasing with order of polynomials, but the training error (misclassification rate) is not necessarily decreasing.

Quiz

Which of the following statements are correct?

- Estimates of the test error with the validation set approach have lower variance than those with LOOCV.
- In a binary classification task we could use the AUC instead of the error rate to perform cross-validation.

Table of Contents

1. Training, Validation and Test Set

2. Cross-Validation

3. **Tuning Models**

4. A Recipe for Supervised Learning

Tuning Models

Hyper-parameter tuning, i.e. finding the best model for the given data, **is an art.**

Common recipes:

- **Grid Search**: Perform cross-validation on a grid of hyper-parameter values. E.g. pick 10 different values of K and pick the best one with cross-validation.
- Use more sophisticated sampling methods for the hyper-parameters to be evaluated, see e.g. `https://www.automl.org/` or `https://github.com/baggepinnen/Hyperopt.jl`

Model assessment and hyper-parameter tuning with MLJ

```julia
# Estimating the test error
evaluate!(mach, resampling = Holdout(fraction_train = 0.5), measure = rmse) # validation set
evaluate!(mach, resampling = CV(nfolds = 5), measure = rmse) # cross-validation

# Hyperparameter tuning
model = KNNRegressor()
self_tuning_model = TunedModel(model = model,      # model to be tuned
                               resampling = CV(),  # or Holdout
                               tuning = Grid(),    # tuning strategy
                               range = range(model, :K, values = 1:10),
                               measure = rmse)
self_tuning_mach = machine(self_tuning_model, X, y)
fit!(self_tuning_mach)

# Nested Cross-Validation
evaluate!(self_tuning_mach, resampling = CV(), measure = rmse)
```

Table of Contents

1. Training, Validation and Test Set

2. Cross-Validation

3. Tuning Models

4. **A Recipe for Supervised Learning**

A Recipe for Supervised Learning

1. Collect (a lot of) data.
2. Look at the raw data; clean it if necessary.
3. Select relevant features from the raw data,
   i.e. choose a suitable representation of the raw data.
4. Select a machine learning method.
5. Fit the data and tune hyperparameters, e.g. with cross-validation.
6. - ▶ training loss: high, test loss: high (underfitting?): select a more flexible method.
   - ▶ training loss: low, test loss: high (overfitting?): select a less flexible method.
7. Repeat 4-6 until the lowest test loss is found.
8. If unhappy with the lowest test loss, repeat 2-7 or collect more data.
9. Fit the best model on all available data for best performance on unseen data.