CS-EJ3211 Machine Learning with Python

Session 3 - Model validation and selection

Shamsi Abdurakhmanova

Aalto University
FITech

14.06.23

Machine Learning - recap

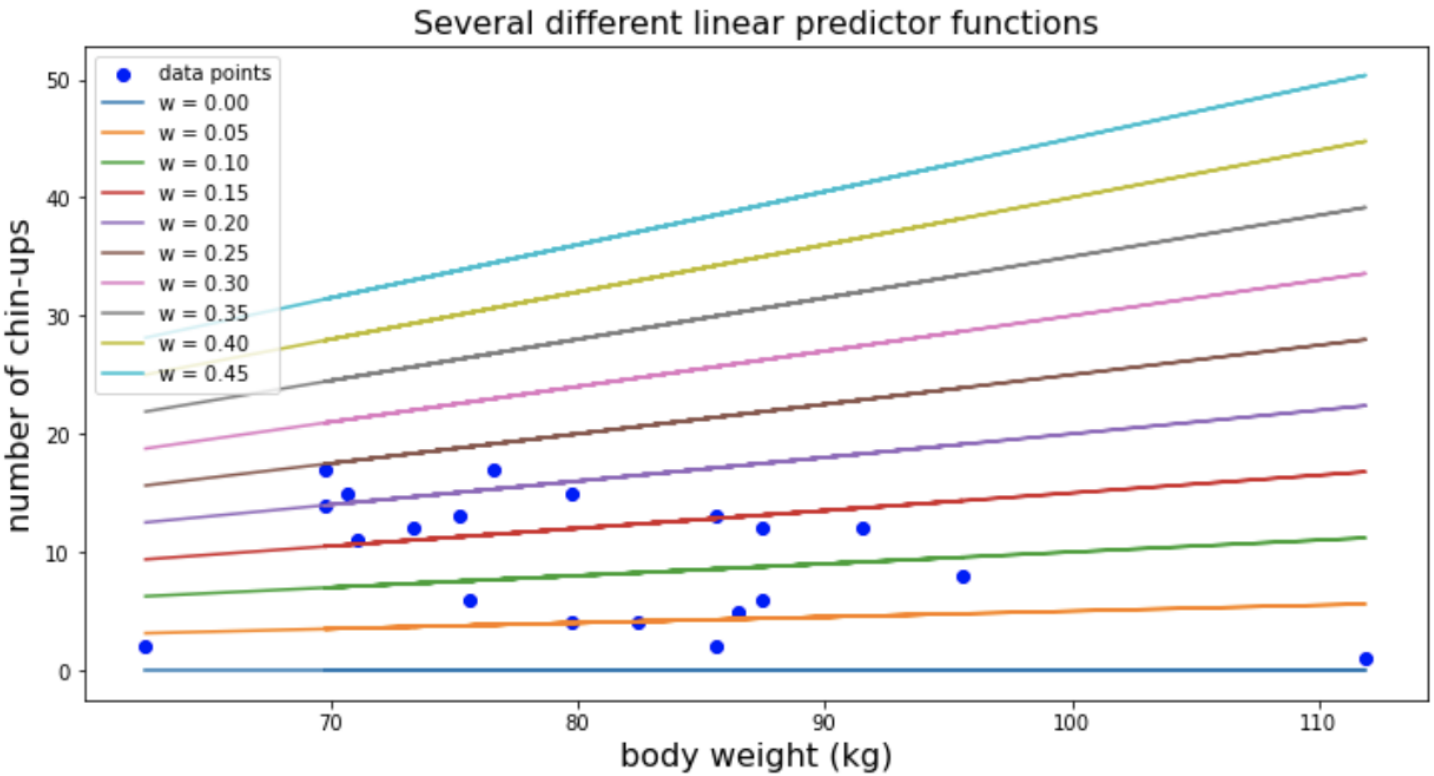
Hypothesis space $h(x) = wx$ is infinite set (collection) of all (linear) functions of form $h(x) = wx$. Examples:

- $h^{(1)}(x) = 0.05x$
- $h^{(2)}(x) = -10x$
- $h^{(3)}(x) = 0x = 0$

Hypotheses $h^{(1)}(x), h^{(2)}(x), h^{(3)}(x)$ are in the hypothesis space $h(x) = wx$.

Machine Learning - recap

Hypothesis space $h(x) = wx$.



Machine Learning

The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs—not just those on which our model was trained. The ability to perform well on previously unobserved inputs is called generalization.

"Deep Learning" I. Goodfellow.

Machine Learning workflow

- Data (features & labels), Model, Loss
- Train (fit) model using training set
- Evaluate model performance on training set (average training error)

Is an average training error a good estimate of a generalization capability of the model?

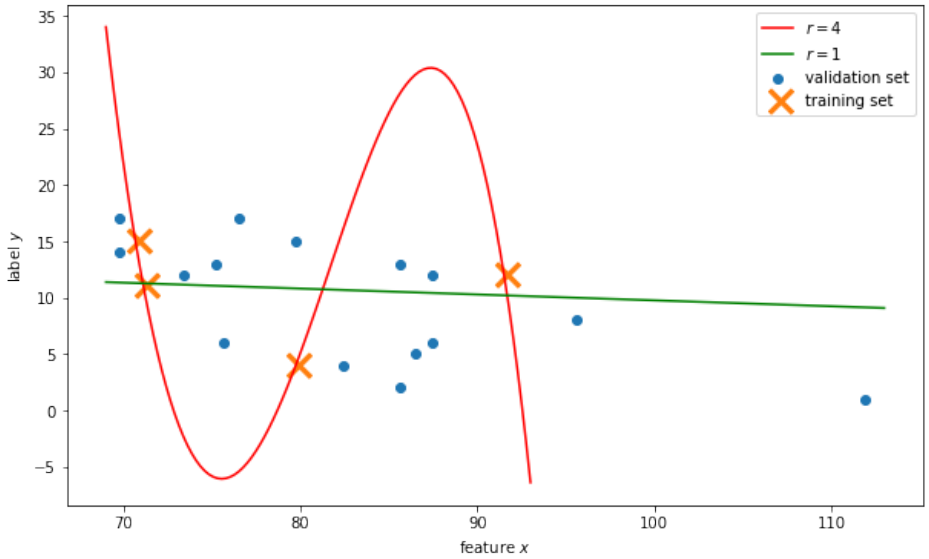
Machine Learning workflow

- Data (features & labels), Model, Loss
- Train (fit) model using training set
- Evaluate model performance on training set (average training error)

Is an average training error a good estimate of a generalization capability of the model?

No, an average training error is too optimistic. Usually, model performs worse on unseen data (overfitting) $\rightarrow$ Need to choose model that fit unseen "new" data well.

Overfitting on training data



- Data: how representative is training set?
- Hypothesis: how complex is the model?

Ideally: n.o. data points $\gg$ n.o. model's parameters.
More in the coursebook Ch.2.2.2 "The Size of a Hypothesis Space".

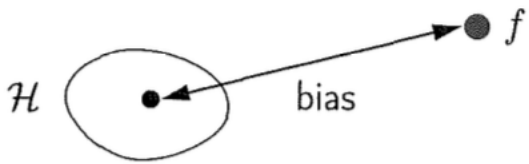
Bias - Variance Tradeoff

Expected loss of the hypothesis $\bar{L}(h) := E\{L((\mathbf{x}, y), h)\}$ can be decomposed into the sum of three fundamental quantities*:

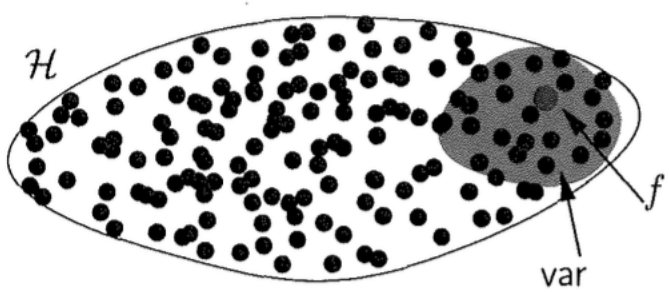
- the variance
- the bias (the limitation of the hyp.space)
- the variance of the error terms (irreducible error, noise)

*read details in the course book Ch.6.4.

Bias - Variance Tradeoff



Very small model. Since there is only one hypothesis, both the average function $\bar{g}$ and the final hypothesis $g^{(\mathcal{D})}$ will be the same, for any data set. Thus, var $= 0$. The bias will depend solely on how well this single hypothesis approximates the target f, and unless we are extremely lucky, we expect a large bias.



Very large model. The target function is in $\mathcal{H}$. Different data sets will lead to different hypotheses that agree with f on the data set, and are spread around f in the red region. Thus, bias ≈ 0 because $\bar{g}$ is likely to be close to f. The var is large (heuristically represented by the size of the red region in the figure).

"Learning From Data" Yaser S. Abu-Mostafa

Machine Learning workflow - updated

- Split Data into **training** and **test** sets
- Train (fit) model using **training** set
- Evaluate model performance on **test** set (average test error)

Computed test loss/ metrics is our estimate on how model will generalize to unseen data.

Machine Learning workflow - updated

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Split Data into training and validation sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# model
reg = LinearRegression()

# fit on training set
reg.fit(X_train, y_train)

# compute predictions
ypred_train = reg.predict(X_train)
ypred_val = reg.predict(X_val)

# compute mse
error_train = mean_squared_error(y_train, ypred_train)
error_val = mean_squared_error(y_val, ypred_val)
```

Machine Learning problem types

- Case 1. Choose the best hypothesis from hyp.space.
- Case 2a. Choose the best hypothesis space: Model selection.
  Example: choose between linear and decision tree regressor.
- Case 2b. Choose the best hypothesis space: Hyperparameter tuning.
  Example: choosing ε in Huber regressor.

Hyperparameters tuning

Hyperparameters: degree of polynomial, number of units in ANN, parameter ε in Huber Loss, parameter α in Lasso, ... etc.

- Split Data into **training** and **test** sets
- Fit models with different hyperparameter values to training set.
- Estimate how well models fit test set
- Select the model which fits best to test set

How well this final test loss estimate generalization capabilities of a selected model?

Hyperparameters tuning

Hyperparameters: degree of polynomial, number of units in ANN, parameter ε in Huber Loss, parameter α in Lasso, ... etc.

- Split Data into **training** and **test** sets
- Fit models with different hyperparameter values to training set.
- Estimate how well models fit test set
- Select the model which fits best to test set

How well this final test loss estimate generalization capabilities of a selected model?

The model is selected based on test set will overfit test set $\rightarrow$ Use another set for final evaluation.

Hyperparameters tuning - updated

- Split Data into **Training**, **Validation** and **Test** sets
- Fit models with different hyperparameter values to **Training** set
- Select the model which smallest **Validation** error
- Re-train chosen model on **Training** + **Validation** set
- Estimate model performance on **Test** set

Hyperparameters tuning - Huber regressor example

- Split Data into **Training**, **Validation** and **Test** sets
- fit training data to Huber regressor $\varepsilon = 1$ and to Huber regressor $\varepsilon = 2$
- compute average validation error for (trained) Huber regressor $\varepsilon = 1$ and for (trained) Huber regressor $\varepsilon = 2$
- choose model with smallest average validation error, e.g. Huber regressor $\varepsilon = 1$
- combine train and val data (trainval) and re-fit Huber regressor $\varepsilon = 1$ to the trainval data
- use re-trained Huber regressor $\varepsilon = 1$ to predict on test data and compute average test loss.

Terminology alert

- **Training** set - find best hypothesis out of the given hyp. space; to learn (fit) model parameters.
- **Validation** set - choose best hyp. space from several. For example, linear regression vs decision tree regressor; polynomial of degree 3 vs polynomial of degree 10.
- **Test** set - estimate generalization capabilities of a chosen (selected) hypothesis.

Summary

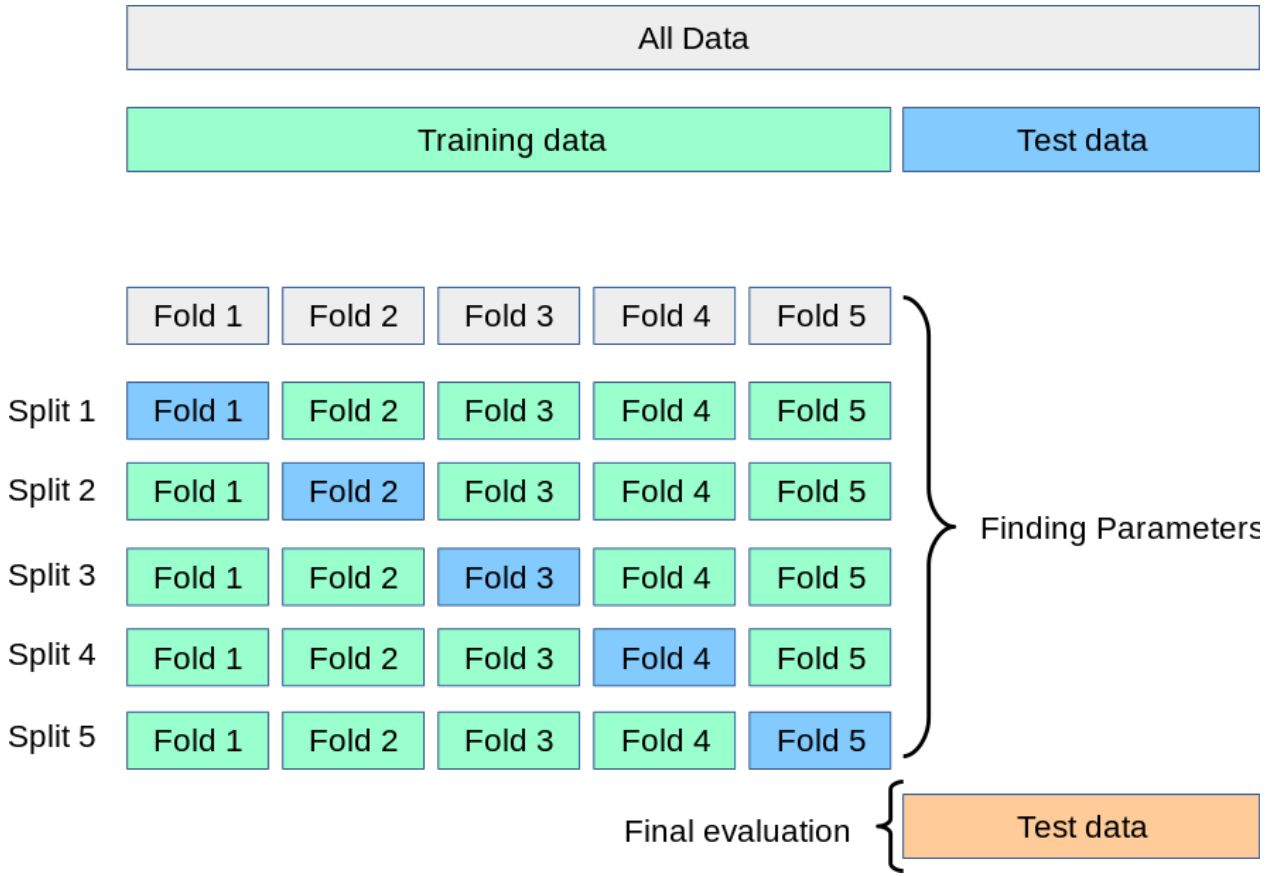
One model:

Data $\rightarrow$ Training, Test

Compare several models, hyperparameter tuning:

Data $\rightarrow$ Training, Validation, Test

Cross -Validation



Student task. 5-Fold Cross Validation.

```
# For loop, iterate hyperparamter values:
    ...
    # For loop, iterate indices with kf object:

        # Create a model
        # Fit the model on the current training set

        # Calculate the predicted labels of the current training set
        # Calculate the predicted labels of the current val set

        # Add the training error to the list of errors
        # Add the val error to the list of errors

    # Compute the mean of training errors across splits
    # Compute the mean of validation errors across splits
```

GridSearchCV

```
# data
iris = datasets.load_iris()

# hyperparameters
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}

# cross-validation
kf = KFold(n_splits=6, shuffle=True)

# model
svc = svm.SVC()

# grid search
clf = GridSearchCV(svc, parameters, cv=kf)
clf.fit(iris.data, iris.target)
```

Penalized Optimization.

Ridge Regression:

$$\mathcal{E}(\mathbf{w}, w_0) = (1/m_t) \sum_{\left(\mathbf{x}^{(i)}, y^{(i)}\right) \in \mathbb{X}^{(t)}} \left(y^{(i)} - w_0 - \mathbf{w}^T \mathbf{x}^{(i)}\right)^2 + \alpha \|\mathbf{w}\|_2^2. \quad (1)$$

Lasso Regression:

$$\mathcal{E}(\mathbf{w}, w_0) = (1/m_t) \sum_{\left(\mathbf{x}^{(i)}, y^{(i)}\right) \in \mathbb{X}^{(t)}} \left(y^{(i)} - w_0 - \mathbf{w}^T \mathbf{x}^{(i)}\right)^2 + \alpha \|\mathbf{w}\|_1. \quad (2)$$