Statistical Machine Learning – Week 10

Sascha Jecklin

HS 2022/23

1 Task

In this lab we will evaluate `smoothing splines` and `polynomial regression`. In the first exercise we fit both models to an artificial dataset whose underlying population regression function exhibits a highly non-linear behaviour.

The flexibility of these two models is determined by a hyperparameter. In case of polynomial regression this is the polynomial degree and for smoothing spline it is the tollerance parameter s. In a second exercise we will try to determine the optimal value of these hyperparameters using grid search and cross validataion.

- Open the file `week10_exercise.py` and get an overview of the provided functions. The main function creates the dataset and calls the functions `exercise_1` and `exercise_2`, which perform the task of the corresponding exercise.
- Implement polynomial regression and smoothing spline in the function `exercise_1` and assume some reasonable values for the degree of freedom / smoothing parameter.
  Hints:
  - Sklearn does not provide one single function for polynomial regression. You have to use `PolynomialFeatures` in combination with `LinearRegression`.
  - Use scipy's `UnivariateSpline` function to implement a smoothing spline model. Make sure you fully understand the function parameters and their effect on the model flexibility.
- As part of the first exercise, plot the dataset as scatter plot and in the same figure the true function and the two trained models as line plots.
- Vary the hyperparameter of these two models and observe the effect on the model flexibility.
- Implement polynomial regression and smoothing spline in the function `exercise_2` and determine the best values for the degree of freedom / smoothing parameter by cross validation.
  Hints:
  - Use sklearn's `GridSearchCV` function to perform grid search over a defined parameter range.
  - Use the provided function `spline_cross_validation` to perform cross validation with a specified smoothing parameter.
- As part of the second exercise, plot the dataset as scatter plot and in the same figure the true function and the two trained models with the best parameters as line plots.
- Test your implementation with other data generating functions such as $f(x) = x^4$ or $f(x) = \frac{sin(2\pi x)}{x}$.

Comments

- See here for an explanation of sklearn's PolynomialFeatures function.
- See here for an explanation of scipy's UnivariateSpline function.
- This example explains how to use sklearn's GridSearchCV function.