Responsible TA:
Asbjörn Rasmussen
asbjoern.rasmussen@chem.ethz.ch

Statistical and Numerical Methods
Exercise 10, HS24

03/12/2024

## 1 Dataset with Gaussian noise (core)

We can generate a trial dataset by adding Gaussian noise to a deterministic function so that :

$$y_i = f(x_i) + \epsilon_i \tag{1}$$

where $\epsilon_i \sim N(f(x_i), \sigma^2)$
We will fit the generated data using a linear regression model with polynomial basis functions :

$$y = \sum_{i=0}^{M} \beta_i \Phi^i(x) \tag{2}$$

where

$$\Phi^i(x) = x^i$$

1. Find online the function-file Ex1_dataset.m that will generate the dataset for you.
2. Call the function in your script by specifying the following input arguments:

   a. The size of the dataset, e.g. 200

   b. The variance of the noise distribution, e.g. 1

   c. A function handle to $f(x) = sin(2\pi x) + cos(4\pi x)$
3. Create the dataset and inspect it by plotting it against the underlying 'true' function.
4. Perform a stepwise linear regression with polynomial features on the data using the *stepwiselm* function:

   a. Use AIC as the stopping criterion

   b. Set the upper limit for the adding of fitting features/variables to a polynomial of degree 9, i.e. by setting the 'Upper' argument to 'poly9'.

   c. Display the model output and plot the model.

   d. How many polynomials are added? Is the model employing all the accessible features? Why? Keep track of the regression steps.
5. Generate the Tukey-Anscombe and the QQplot and comment the distribution of residuals.
6. Find online the dataset *dataset1.txt* and import the data into your workspace using the *importdata* function.
7. Perform an ordinary least squares (OLS) linear regression with a polynomial of degree 9 by using the function *regularized_polyfit* that you can find online. Read carefully the explanation in the function file about its usage.
8. Find online the function *fit_eval* and use it with the estimated coefficients from step 7. to plot your fitting results against the original data :

   *Y_fitted = fit_eval(linspace(0, 1), fit_coeff)*
9. Inspect carefully the vector of estimated coefficient and the plot. What do you observe? Why the behavior has changed so dramatically?

10. Now perform ridge regression on the same dataset by specifying a regularization parameter $\lambda$ , e.g. $\sim 1e-3$, as the input of *regularized_polyfit*. Repeat step 8.-9.

11. What was the effect of regularization? Was it helpful? Do you see any issue with this approach?

## 2 Stepwise Model

1. Load *hald*. This data set includes the variables *ingredients* and *heat*. The matrix ingredients contains the percent composition of four chemicals present in the cement. The vector *heat* contains the values for the heat hardening after 180 days for each cement sample.

2. Create a linear model to predict heat based on the ingredients and use it as a benchmark for the next steps.

3. Create a stepwise model using *stepwiselm* and different variable selection criteria (i.e. SSE, BIC, AIC and RSquared). Compare the variables retained in the model and the model performance compared to 2.

4. How many regression coefficients would a model have accounting for all interactions (and quadratic terms) across the different independent variables? Can these be reasonably estimated?

5. Repeat 2. and 3. with a model a stepwise fitting also using interaction terms (and quadratic terms). Which variables are chosen? Compare model performance?

6. Randomly split the data into 3 runs for testing and 10 runs for training the model. Repeat procedure in a for loop 10 times.

7. Compare for each iteration the performance of a purely linear, stepwise linear and stepwise interaction models (with the three criteria listed in 3.) based on the RMSE during training and during prediction. Store both errors in arrays and visualize the distribution of both two arrays based on *boxplot*. What do you observe across the six models as well as comparing results in training and testing? Why?

   Hint: use predict to apply the trained model to the test set and calculate the prediction RMSE in an external function