

Problem solving session – Linear models for regression

Darko Zibar

August 31, 2022

Exercise 1

We consider a system with the output described by the following function: $y(t) = \sin(2\pi t)$, where t denotes time limited to the following interval $t \in [0 : 1]$. We perform a measurement of the system's output at distinct times t_k . The record signal, corrupted by the measurement noise, n_k , is now expressed as:

$$y_k = \sin(2\pi t_k) + n_k, \quad k = 1, \dots, L \quad (1)$$

where t_k is drawn from the uniform distribution in the range from 0 to 1, i.e. $t_k \sim \mathcal{U}[0; 1]$ and n_k is the measurement noise, drawn from the Gaussian distribution with zero mean and variance σ^2 , i.e. $n_k \sim \mathcal{N}(0, \sigma^2)$.

1. Plot $y(t)$ by assuming that $t = 0 : 0.01 : 1$.
2. Generate and plot y_k (when plotting do not connect the dots). Assume that $L = 100$ and $\sigma^2 = 0.09$. Comment on the graph!
3. Observe the impact of increasing the measurement noise variance from $\sigma^2 = 0$ to $\sigma^2 = 0.09$ on y_k .

Now, we are going to assume that we do not know the analytical expression for $y(t)$ and the objective is to learn it from the recorded data-set. The training data-set consist of the training input $\mathbf{X} = [t_1, t_2, \dots, t_L]^T$ and the output $\mathbf{Y} = [y_1, y_2, \dots, y_L]^T$, data, where T denotes the transpose. The data-set is denoted $\mathcal{D}^{L \times 2} = [\mathbf{X}, \mathbf{Y}]$. Since we do not know the exact analytical expression for the function, $y(t)$, we will assume the following model (polynomial):

$$\hat{y}(x_k, \mathbf{W}) = w_0 + w_1 t_k + w_2 t_k^2 + \dots + w_M t_k^M = \sum_{j=0}^M w_j t_k^j \quad (2)$$

where M and $\mathbf{W} = [w_0, w_1, \dots, w_M]$ are the unknown polynomial order and the weight vector, respectively.

1. Use the lecture slides and implement a gradient descent based iterative algorithm for determining the weights $\mathbf{W} = [w_0, w_1, \dots, w_M]$. Assume $L = 100$, $\sigma^2 = 0.09$, $M = 3$, $\eta = 0.1$. Use 3000 iterations for the gradient descent.
2. Plot the mean square error between the learned model, $\hat{y}(x_k, \mathbf{W})$ and the training data $\mathbf{Y} = [y_1, y_2, \dots, y_L]^T$ as a function of the number of iterations. What can be concluded by observing the evolution of the mean square error?
3. Evaluate the learned model, $\hat{y}(x_k, \mathbf{W})$, by testing it for $t = 0 : 0.01 : 1$. Plot $\hat{y}(x_k, \mathbf{W})$ together with $y(t) = \sin(2\pi t)$. Comment on the results!

4. Repeat task 3 by varying the length of L from 10 to 100 in steps of 10. Adjust η and the number of iterations to obtain the best fit. Assume $\sigma^2 = 0.09$ and $M = 3$. What is the impact of decreasing the training data set L ?

Instead of using the gradient descent to determine the weights \mathbf{W} , it is more convenient to use Moore-Penrose Pseudo-Inverse (MPPI). The weight vector \mathbf{W} can then be obtained in a single step without the need to adjust the step-size η .

1. Use the lecture slides and implement the MPPI method for determining the weights $\mathbf{W} = [w_0, w_1, \dots, w_M]$. Assume $L = 100$, $\sigma^2 = 0.09$, $M = 3$.
2. Evaluate the learned model, $\hat{y}(t_k, \mathbf{w})$, by testing it for $t = 0 : 0.01 : 1$ and plotting it against $y(t) = \sin(2\pi t)$. Compare the results with the ones obtained using the gradient descent.
3. Increase the length of the data-set L to 1000. Employ 10-fold cross-validation for learning and testing the model. This implies that you use 900 points for training and 100 for testing. By shuffling the test-set through the data-set you will obtain 10 test-sets and therefore you will obtain 10 different values for the weight vector \mathbf{W} . (Please consult the slides on how to perform cross-validation).
4. Evaluate and plot the root mean square error on the training and test-sets for each of the folds. The root mean square error is defined as:

$$E_{RMS} = \sqrt{\left(\frac{1}{L_{train/test}} \sum_{k=0}^{L_{train/test}} [y_{k,train/test} - \hat{y}(x_{k,train/test}, \mathbf{W})]^2 \right) / L_{train/test}} \quad (3)$$

where $L_{train/test}$ is the length of the training and the test-set, respectively. $x_{k,train/test}$ and $y_{k,train/test}$ are the input/output values belonging to the training and test-set, respectively.

5. Plot the training error and test root mean square error as a function of the polynomial order M ranging from 1 to 11. How do we find the right polynomial model order?

Exercise 2

The goal of the following exercise is to implement Rosenblatt perceptron and use it for learning logical gates functions.

1. Implement AND, OR and XOR table.
2. Generate a training and test-set by performing uniform sampling from AND table. Set the length of the training and test-set to 2000, respectively. HINT: to perform uniform sampling first generate a vector $\mathbf{X} = \text{randi}(4, 1, L)$, where L is the length of the vector.
3. Use the slides (book) and implement a Rosenblatt perceptron. Plot the mean square error computed on the training and the test-set. Use $\eta = 10^{-6}$. Comment on the results!
4. Repeat item 2–3 for OR and XOR gate. Can you learn OR and XOR gate? Comment on the results