Machine Learning: Regression

1 Introduction

Recommended programming language for conducting assignments is Python (but Matlab, R, or other language is also OK). Python libraries required to complete the assignments:

- NumPy
- Matplotlib
- scikit-learn

2 Linear regression

Linear regression model prediction equation:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n + \epsilon_i \tag{1}$$

- $\hat{y}$ — predicted value
- x_i - feature value, $X = [x_0, x_1, \ldots, x_n]$ is a feature vector
- θ_j - model parameter (θ_0 is bias term)

Assignments

- Generate data points using the following code:

```
import numpy as np
X = 0.4 * np.linspace(-3, 3, 500).reshape(500, 1)
y = 6 + 4 * X + np.random.randn(500, 1)
```

- Train two linear regression models based on two approaches: Least Squares and Stochastic Gradient Descent (SGD). Use classes from scikit-learn
- Perform predictions on X and plot the data and predictions on the same plot
- For SGD approach, experiment with hyperparameters

- From obtained model parameters (using least-squares model and SGD model) write the linear equation prediction equation
- Discuss the differences, pros and cons between two approaches

3 Polynomial regression

Polynomial regression model prediction equation:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_2 x_2^3 \ldots + \theta_n x_n^n + \epsilon_i \tag{2}$$

- $\hat{y}$ — predicted value
- x_i - feature value, $X = [x_0, x_1, \ldots, x_n]$ is a feature vector
- θ_j - model parameter (θ_0 is bias term)

Assignments

- Generate data points using the following code:

```
import numpy as np
X = 0.1 * np.linspace(-10, 10, 500).reshape(500, 1)
y = 3 * X**3 +  0.5 * X ** 2 + X
    + 2 + np.random.randn(500, 1)
```

- Train polynomial regression with different degrees of polynomial
- For selected degrees of polynomials calculate mean squared error (MSE)
- Select the best model based on MSE
- From obtained model parameters write the polynomial regression prediction equation
- Describe method