Systems analysis and decision support methods in Computer Science

Lab – Python – Assignment 1

Linear regression

authors: A. Gonczarek, J.M. Tomczak, S. Zaręba, J. Kaczmar

translation: P. Klukowski

Assignment goal

Your task is to implement the linear least squares without regulation and with ℓ_2 regularization in context of polynomial fitting.

Linear least squares

We are going to use the following model:

$$\overline{y} = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w} \tag{1}$$

where $\mathbf{w} = (w_0\, w_1 \ldots w_{M-1})^{\mathrm{T}}$ is a vector of parameters, and $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x})\, \phi_1(\mathbf{x}) \ldots \phi_{M-1}(\mathbf{x}))^{\mathrm{T}}$ is a feature vector. For example a model can be defined as a polynomial of degree M with features defined as the consecutive powers of the argument.

We are interested in fitting a model to observations $\mathbf{y} = (y_1\, y_2 \ldots y_N)^{\mathrm{T}}$ and $\mathbf{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots \mathbf{x}_N]$. Let $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1)\, \boldsymbol{\phi}(\mathbf{x}_2) \ldots \boldsymbol{\phi}(\mathbf{x}_N)]^{\mathrm{T}}$ denote a design matrix of features calculated for input data $\mathbf{X}$. The goal of model fitting procedure is to estimate values parameters $\mathbf{w}$. In order to do so, we are going to minimize the loss function, which defines a mismatch between the observed values and the model output. An example of such loss function is the sum of squared differences between the model predictions and the target values:

$$Q(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|_2^2. \tag{2}$$

This is known as **linear least squares problem**.

Assuming that rank $r(\boldsymbol{\Phi}) = M$, we can compute the gradient with respect to the parameters and set it to zero. This leads to the unique solution:

$$\mathbf{w} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y}. \tag{3}$$

Linear least squares with ℓ_2 regularization

An important problem associated with fitting a model with linear least squares is to decide how many features should be used (i.e. select a degree of polynomial). Selecting inadequate number of features results in a model, which do not reflect observed dependencies between inputs and outputs. To mitigate this issue we can set a fixed number of features, which is big enough to model large variety of dependencies, and modify the loss function by introducing a **regularization term**, precisely the ℓ_2 **regularization**[1]:

$$Q(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{4}$$

where $\lambda > 0$ is a regularization coefficient.

Calculating the gradient with respect to the parameters and setting it to zero gives the unique solution:

$$\mathbf{w} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{y}. \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix. In this case no assumption about the rank of the matrix $\mathbf{\Phi}$ is required.

Model selection

Further, we want to determine the degree of the polynomial M. This problem can be approached as follows:

1. Using high degree polynomial and apply ℓ_2 regularization.
2. Using different models (different degrees polynomials) and then select the one with the lowest error throughout **model selection procedure**

To assess the quality of the model in the model selection procedure, we are going to use the *mean squared error*:

$$E(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \overline{y}(\mathbf{x}_n)\right)^2. \tag{6}$$

Choosing a degree of the polynomial

We are going to investigate different values of $M \in \mathcal{M}$. For example $\mathcal{M} = \{0, 1, 2, 3, 4, 5, 6, 7\}$, so polynomials of orders from $M = 0$ to $M = 7$ are considered. We train the model (determine the parameters $\mathbf{w}$) using equation (3) and training set ($\mathbf{X}$ and $\mathbf{y}$). After that, a comparison of different polynomials degrees is done with the use of separate validation set $\mathbf{X}_{val}$ and $\mathbf{y}_{val}$. Detailed model selection procedure is following:

[1] ℓ_2 regularization is sometimes called Tikhonov regularization.

1. For each degree of polynomial $M \in \mathcal{M}$ calculate the parameters $\mathbf{w}_M$ (eq. 3) with the use of training data $\mathbf{X}$ and $\mathbf{y}$.

2. Having $\mathbf{w}_M$ calculated for each degree of polynomial, calculate error E_M (eq. 6) with the use of validation set ($\mathbf{X}_{val}$ and $\mathbf{y}_{val}$).

3. Select the order of the polynomial M that has the smallest error E_M.

Calibrating the regularization coefficient

In regularization approach, a polynomial with high degree (i.e. $M = 7$) is selected. Then the model is trained with different regularization coefficients $\lambda \in \Lambda$. For example, $\Lambda = \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300\}$. Model parameters $\mathbf{w}$ are calculated using equation (5) and training set. A comparison of models trained with different λ values is done on validation set ($\mathbf{X}_{val}$ and $\mathbf{y}$). The entire model selection procedure is following:

0. Set high value of M.

1. For each regularization coefficient $\lambda \in \Lambda$ calculate the parameters $\mathbf{w}_\lambda$ (eq. 5) using training data ($\mathbf{X}$ and $\mathbf{y}$).

2. For each model with parameters $\mathbf{w}_\lambda$ calculate error E_λ (eq. 6) using validation set ($\mathbf{X}_{val}$ and $\mathbf{y}_{val}$).

3. Select those parameters $\mathbf{w}_\lambda$ which result in the smallest error E_λ.

Dataset

A dataset used in this assignment was generated using formula:

$$y = \sin(2\pi x) + \varepsilon, \tag{7}$$

where $\varepsilon \sim \mathcal{N}(\varepsilon|0, \sigma^2)$ is Gaussian noise, i.e., a normally-distributed random variable with zero-mean. The dataset was divided into two disjunctive training sets $\mathbf{X}$, $\mathbf{y}$ (composed of 8 and 50 observations) and one validation set $\mathbf{X}_{val}$, $\mathbf{y}_{val}$ (20 examples). Figure 1 presents exemplary training (blue points) and validation sets (red points), as well as true dependency (eq. 7, green line) and a learned model (red line).

Testing the code

To check the code correctness use the function `main` (file `main.py`).

File `main.py` must remain unchanged. Modification of this file is not allowed.

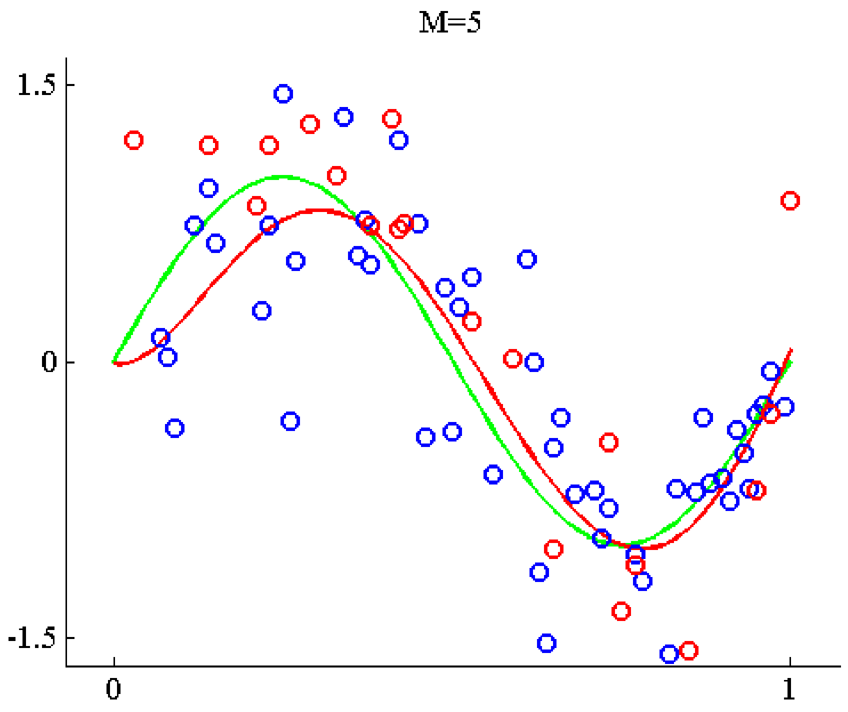

Figure 1: Visualization of assignment dataset with true dependency (green) and the model (red).

Tasks

Supplementary functions are implemented in utils.py:

- `y = polynomial( x, w )` – function returns prediction `y` for a given `x` and parameters `w`.

Instruction:

Implement all functions in file `content.py`

1. Implement `mean_squared_error` function that calculates the mean squared error (Eq. 6) .
2. Implement `design_matrix` function that calculates a design matrix Φ.
3. Implement `least_squares` function that solves the linear least squares problem.
4. Implement `regularized_least_squares` function that solves the regularized linear least squares problem.
5. Implement `model_selection` function that performs model selection for different polynomial degrees $\mathcal{M}$ in the file .
6. Implement `regularized_model_selection` function that performs a model selection for a regularized model.

REMARK! All functions and variables names in the file `content.py` must remain unchanged.

Control questions

1. Derive equation (3) (a solution of the linear least squares problem).
2. Derive equation (5) (solution of the linear least squares problem with the ℓ_2 regularization).
3. What is *overfitting*? Explain using an example of polynomial fitting.
4. What is *underfitting*? Explain using an example of polynomial fitting.
5. What are training, validation and test set? Why we need those sets?
6. What is model selection? How to make model selection? Can quality metrics for model selection be different than training criteria?
7. Which of the presented model selection procedures is easier to use in practice and why?
8. In what cases the linear least squares problem have a unique solution, when multiple solutions exist? How about regularized linear least squares?
9. Write a feature vector $\boldsymbol{\phi}$ for the polynomial of the order M.
10. What is the role of λ parameter? How its value affects the solution?