Machine Learning – SS 2024 – Exercise Sheet 4

# Supervised Learning I

**Deadline:** 06. May 2024, 23:55 (CET)

**Important.** Upload a single PDF file and a single .ipynb notebook with your homework solution to ILIAS by 06. May 2024, 23:55 (CET). We recommend typesetting your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it wont be graded and you waive your right to dispute that. You only need to submit homework problems. The in-class problems will be discussed in the tutorial but will not be graded.

**Policy.** Any resources that you use to solve the homework must be cited:

- When taking inspiration from online forums (like StackOverflow), you should include the URL where you found the information;
- If using large language models (like ChatGPT) for formulation/editing your answers, you should either provide the link when available (ChatGPT supports this feature) or detail how you did use it, whether for rephrasing, explaining an unclear concept, or seeking assistance in proving a point, among other uses.
- When taking inspiration from previous years' homeworks, you should include both the number of the exercise and the year of the homework along with how you did use the solution, e.g. "analogous exercise from the 2023 homework 4 ex. 2, differently from that we [...]. Similarly, to the exercise we [...]".
- If you discuss the homework with another team, you should include the name of the team and the members you discussed with – note that this does not mean that the solutions should be the same, just that you discussed the homework together. Write your own solution.

## Homework Problems

### Ridge regression

**Problem 1.** Show that ridge regression on a design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$ with regularization strength $\lambda$ is equivalent to ordinary least squares regression with an augmented design matrix and target vector

$$\hat{\boldsymbol{\Phi}} = \begin{pmatrix} \boldsymbol{\Phi} \\ \sqrt{\lambda} \boldsymbol{I}_M \end{pmatrix} \quad \text{and} \quad \hat{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \mathbf{0}_M \end{pmatrix}.$$

### Implementation

**Problem 2.** John Doe is a data scientist, and he wants to fit a polynomial regression model to his data. For this, he needs to choose the degree of the polynomial that works best for his problem.
Unfortunately, John didn't attend the lecture, so he wrote the following code for choosing the optimal degree of the polynomial:

```
X, y = load_data()
best_error = -1
best_degree = None

for degree in range(1, 50):
    w = fit_polynomial_regression(X, y, degree)
```

```
        y_predicted = predict_polynomial_regression(X, w, degree)
        error = compute_mean_squared_error(y, y_predicted)
        if (error <= best_error) or (best_error == -1):
            best_error = error
            best_degree = degree

    print("Best degree is " + str(best_degree))
```

Assume that the functions are implemented correctly and do what their name suggests.

a) Explain briefly why this code doesn't do what it's supposed to do.

b) Describe a possible way to fix the problem with this code. (You don't need to write any code, just describe the approach.)

**Problem 3.** Derive the closed-form solution for the ridge regression error function

$$\mathcal{L}_{\text{ridge}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) - y_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

Additionally, discuss the scenario when the number of training samples $N$ is smaller than the number of basis functions $M$. What computational issues arise in this case? How does regularization address them?

**Problem 4.** Using singular value decomposition of the design matrix $\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ show that predicted target $\hat{\boldsymbol{v}}$ for the training set when using $\boldsymbol{w}^\star_{\text{ridge}}$ can be written as

$$\hat{\boldsymbol{v}} := \boldsymbol{\Phi}\boldsymbol{w}^\star_{\text{ridge}} = \sum_{j=1}^{M}\left(\frac{\sigma_j^2}{\sigma_j^2 + \lambda}\boldsymbol{u}_j\boldsymbol{u}_j^\top\right)\boldsymbol{v}$$

where $\boldsymbol{u}_j$ are the columns of $\boldsymbol{U}$, $\sigma_j$ the elements of diagonal matrix $\boldsymbol{S}$ and $\lambda$ the strength of the $L_2$ regularization. What is the interpretation of this formula?

## Comparison of Linear Regression Models

**Problem 5.** We want to perform regression on a dataset consisting of $N$ samples $\boldsymbol{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{y} \in \mathbb{R}^N$).
Assume that we have fitted an $L_2$-regularized linear regression model and obtained the optimal weight vector $\boldsymbol{w}^\star \in \mathbb{R}^D$ as

$$\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \frac{\lambda}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

*Note that there is no bias term.*
Now, assume that we obtained a new data matrix $\boldsymbol{X}_{new}$ by scaling all samples by the same positive factor $a \in (0, \infty)$. That is, $\boldsymbol{X}_{new} = a\boldsymbol{X}$ (and respectively $\boldsymbol{x}_i^{new} = a\boldsymbol{x}_i$).

a) Find the weight vector $\boldsymbol{w}_{new}$ that will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^\star$ produces on $\boldsymbol{X}$.

b) Find the regularization factor $\lambda_{new} \in \mathbb{R}$, such that the solution $\boldsymbol{w}^\star_{new}$ of the new $L_2$-regularized linear regression problem

$$\boldsymbol{w}^\star_{new} = \arg\min_{\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{X}_i^{new} - y_i)^2 + \frac{\lambda_{new}}{2}\boldsymbol{w}^\top \boldsymbol{w}$$

will produce the same predictions on $\boldsymbol{X}_{new}$ as $\boldsymbol{w}^\star$ produces on $\boldsymbol{X}$.

Provide a mathematical justification for your answer.

## Programming problems

**Problem 6.** Download `exercise_notebook_04.ipynb` from ILIAS. Fill in the missing code and run the notebook. Use the given docstring of each function as a hint on how you should implement the function and what is going to be the input and the output. Upload the final `.ipynb` along with the PDF to ILIAS.

# In-Class Problems

**Problem 7.** Assume that we are given a dataset, where each sample $x_i$ and regression target $y_i$ is generated according to the following process

$$x_i \sim \text{Uniform}(-10, 10)$$
$$y_i = ax_i^3 + bx_i^2 + cx_i + d + \epsilon_i, \quad \text{where} \quad \epsilon_i \sim \mathcal{N}(0, 1) \quad \text{and} \quad a, b, c, d \in \mathbb{R}.$$

The 3 regression algorithms below are applied to the given data. Your task is to say what the bias and variance of these models are (low or high). Provide a 1-2 sentence explanation to each of your answers.

a) Linear regression

b) Polynomial regression with degree 3

c) Polynomial regression with degree 10

**Problem 8.** Given a training set consisting of samples $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^\top$ with respective regression targets $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)^\top$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$.
Alice fits a linear regression model $f(\boldsymbol{x}_i) = \boldsymbol{w}^\top \boldsymbol{x}_i$ to the dataset using the closed form solution for linear regression (normal equations).
Bob has heard that by transforming the inputs $\boldsymbol{x}_i$ with a vector-valued function $\boldsymbol{\Phi}$, he can fit an alternative function, $g(\boldsymbol{x}_i) = \boldsymbol{v}^\top \boldsymbol{\Phi}(\boldsymbol{x}_i)$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\Phi}(x_i) = \boldsymbol{A}^\top \boldsymbol{x}_i$, where $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ has full rank.

a) Show that Bob's procedure will fit the same function as Alice's original procedure, that is $f(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^D$ (given that $\boldsymbol{w}$ and $\boldsymbol{v}$ minimize the training set error).

b) Can Bob's procedure lead to a lower training set error than Alice's if the matrix $\boldsymbol{A}$ is not invertible? Explain your answer.