

## Machine Learning Exercise Sheet 04

### Linear Regression

---

Exercise sheets consist of two parts: In-class exercises and homework. The in-class exercises will be solved and discussed during the tutorial. The homework is for you to solve at home and further engage with the lecture content. There is no grade bonus and you do not have to upload any solutions. Note that the order of some exercises might have changed compared to last year's recordings.

---

#### In-class Exercises

**Problem 1:** 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 2:** Given is a training set consisting of samples  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^T$  with respective regression targets  $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$  where  $\mathbf{x}_i \in \mathbb{R}^D$  and  $y_i \in \mathbb{R}$ .

Alice fits a linear regression model  $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$  to the dataset using the closed form solution for linear regression (normal equations).

Bob has heard that by transforming the inputs  $\mathbf{x}_i$  with a vector-valued function  $\Phi$ , he can fit an alternative function,  $g(\mathbf{x}_i) = \mathbf{v}^T \Phi(\mathbf{x}_i)$ , using the same procedure (solving the normal equations). He decides to use a linear transformation  $\Phi(\mathbf{x}_i) = \mathbf{A}^T \mathbf{x}_i$ , where  $\mathbf{A} \in \mathbb{R}^{D \times D}$  has full rank.

- a) Show that Bob's procedure will fit the same function as Alice's original procedure, that is  $f(\mathbf{x}) = g(\mathbf{x})$  for all  $\mathbf{x} \in \mathbb{R}^D$  (given that  $\mathbf{w}$  and  $\mathbf{v}$  minimize the training set error).
- b) Can Bob's procedure lead to a lower training set error than Alice's if the matrix  $\mathbf{A}$  is not invertible? Explain your answer.

**Problem 3:** See Jupyter notebook `inclass_04_notebook.ipynb`.

---

## Homework

### Least squares regression

**Problem 4:** Let's assume we have a dataset where each datapoint,  $(\mathbf{x}_i, y_i)$  is weighted by a scalar factor which we will call  $t_i$ . We will assume that  $t_i > 0$  for all  $i$ . This makes the sum of squares error function look like the following:

$$E_{\text{weighted}}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N t_i [\mathbf{w}^T \phi(\mathbf{x}_i) - y_i]^2$$

Find the equation for the value of  $\mathbf{w}$  that minimizes this error function.

Furthermore, explain how this weighting factor,  $t_i$ , can be interpreted in terms of

- 1) the variance of the noise on the data and
- 2) data points for which there are exact copies in the dataset.

### Ridge regression

**Problem 5:** Show that ridge regression on a design matrix  $\Phi \in \mathbb{R}^{N \times M}$  with regularization strength  $\lambda$  is equivalent to ordinary least squares regression with an augmented design matrix and target vector

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

### Implementation

**Problem 6:** 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 hasn't attended IN2064, so he writes 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.)

## Bayesian linear regression

Bishop 3.12

In the lecture we made the assumption that we already knew the precision (inverse variance) for our Gaussian distributions. What about when we don't know the precision and we need to put a prior on that as well as our Gaussian prior that we already have on the weights of the model?

**Problem 7:** It turns out that the conjugate prior for the situation when we have an unknown mean and unknown precision is a normal-gamma distribution (See section 2.3.6 in Bishop). This means that if our likelihood is as follows:

$$p(\mathbf{y} \mid \Phi, \mathbf{w}, \beta) = \prod_{i=1}^N \mathcal{N}(y_i \mid \mathbf{w}^T \phi(\mathbf{x}_i), \beta^{-1})$$

Then the conjugate prior for both  $\mathbf{w}$  and  $\beta$  is

$$p(\mathbf{w}, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_0, \beta^{-1} \mathbf{S}_0) \text{Gamma}(\beta \mid a_0, b_0)$$

Show that the posterior distribution takes the same form as the prior, i.e.

$$p(\mathbf{w}, \beta \mid \mathcal{D}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \beta^{-1} \mathbf{S}_N) \text{Gamma}(\beta \mid a_N, b_N)$$

Also be sure to give the expressions for  $\mathbf{m}_N$ ,  $\mathbf{S}_N$ ,  $a_N$ , and  $b_N$ .

*Hint:* Expand  $\log p(\mathbf{w}, \beta \mid \mathcal{D})$  once with the prior and likelihood and once with the presumed posterior form. The resulting expressions have to be equal, so you should be able to match all terms in the two expansions against each other and then read off the parameters of the posterior distribution.

**Problem 8:** Derive the closed form solution for ridge regression error function

$$E_{\text{ridge}}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \phi(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

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?

## Comparison of Linear Regression Models

**Problem 9:** We want to perform regression on a dataset consisting of  $N$  samples  $\mathbf{x}_i \in \mathbb{R}^D$  with corresponding targets  $y_i \in \mathbb{R}$  (represented compactly as  $\mathbf{X} \in \mathbb{R}^{N \times D}$  and  $\mathbf{y} \in \mathbb{R}^N$ ).

Assume that we have fitted an  $L_2$ -regularized linear regression model and obtained the optimal weight vector  $\mathbf{w}^* \in \mathbb{R}^D$  as

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - y_i)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

Note that there is no bias term.

Now, assume that we obtained a new data matrix  $\mathbf{X}_{new}$  by scaling all samples by the same positive factor  $a \in (0, \infty)$ . That is,  $\mathbf{X}_{new} = a\mathbf{X}$  (and respectively  $\mathbf{x}_i^{new} = a\mathbf{x}_i$ ).

- a) Find the weight vector  $\mathbf{w}_{new}$  that will produce the same predictions on  $\mathbf{X}_{new}$  as  $\mathbf{w}^*$  produces on  $\mathbf{X}$ .
- b) Find the regularization factor  $\lambda_{new} \in \mathbb{R}$ , such that the solution  $\mathbf{w}_{new}^*$  of the new  $L_2$ -regularized linear regression problem

$$\mathbf{w}_{new}^* = \arg \min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^N (\mathbf{w}^T \mathbf{X}_i^{new} - y_i)^2 + \frac{\lambda_{new}}{2} \mathbf{w}^T \mathbf{w}$$

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

Provide a mathematical justification for your answer.

## Programming Task

**Problem 10:** Download the notebook `exercise_04_linear_regression.ipynb` from Moodle. Fill in the missing code and follow the instructions in the notebook to append the solution to your PDF submission.

*Note:* We suggest that you use [Anaconda](#) for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks and how to convert them to other formats, consult the [Jupyter documentation](#) and [nbconvert documentation](#).