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 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^{\mathrm{T}}$ with respective regression targets $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)^{\mathrm{T}}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$.

Alice fits a linear regression model $f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i$ 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, $g(\boldsymbol{x}_i) = \boldsymbol{v}^T \boldsymbol{\Phi}(\boldsymbol{x}_i)$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\Phi}(x_i) = \boldsymbol{A}^T \boldsymbol{x}_i$, 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.

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

P1 $x_i \sim \text{Uniform}(-10, 10)$

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

underfitting

① Linear regression : High bias / Low variance ⇔ Linear model is too rigid to represent the uniform and high-dim data

② Polynomial 3 : Low bias / Low variance ⇔ because y_i is also a polynomials over 3 dim

③ Polynomial 10 : Low bias / High variance ⇔ Polynimal with 10 degree model is too flexible to fit the y_i, bring the fit noise

P2 $X = (x_1, \cdots, x_n)^T \quad y = (y_1 \cdots y_n)^T \quad x_i \in R^{p\times 1} \quad y_i \in R, \quad X \in R^{n\times p}$

A $f(x_i) = w^T x_i$

$E_{LS} = \frac{1}{2}\sum_{i=1}^{N} (w^T x_i - y_i)^2$

$= \frac{1}{2}(Xw-Y)^T(Xw-Y)$

$w^* = (X^TX)^{-1}X^Ty$

B $y(x_i) = v^TA^Tx_i$

$E_{LS} = \frac{1}{2}\sum_{i=1}^{N} (v^TA^Tx_i - y_i)^2$

$= \frac{1}{2}(XAv - Y)^T(XAv - y)$

$\frac{1}{2}(A^Tv^Tx^TXvA - 2A^Tv^Tx^Ty + c)$

$A^Tx^TXAv - A^Tx^Ty \doteq 0$

$v^* = (A^Tx^TxA)^{-1}A^Tx^Ty$

$= A^{-1}(x^Tx)^{-1}\cancel{A^{-T}}\cancel{A^T}x^Ty$

$= A^{-1}(x^Tx)^{-1}x^Ty$

$= A^{-1}w^*$

$g(x_i) = w^{*T}\cancel{A^{-T}}\cancel{A^T}x_i$

$= w^{*T}x_i$

$f(x) = w^{*T}x_i$

$\therefore f(x) = g(x)$

Because $v^* = A^{-1}w^*$

So v^* is a subset of w^* can't achieve a lower error.

P4 $E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N} t_i \left[\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - y_i\right]^2$

$\frac{1}{2}\left[w^T\Phi^T\Phi w - 2w^T\Phi^Ty + y^Ty\right]$

$w^* = \arg\min_w E_W(w) = \frac{d\,\frac{1}{2}[\Phi w - y]^T[\Phi w - y]}{dw}$

assume $T = \text{diag}(t_1 \cdots t_N)$

$E_W(w) = \frac{1}{2}(\Phi w - y)^T T(\Phi w - y)$

$= \frac{1}{2}\left[w^T\Phi^TT - y^TT\right]\left[\Phi w - y\right]$

$= \frac{1}{2}\left[w^T\Phi^T T\Phi w - 2w^T\Phi^TTy + y^TTy\right]$

$= T\Phi^T\Phi w - T\Phi^Ty \doteq 0$

$\frac{dE_W}{dw} = \Phi^TT\Phi w - \Phi^TTy \doteq 0$

$w^* = (\Phi^T\Phi)^{-1}\Phi^Ty$

$w^* = (\Phi^TT\Phi)^{-1}\Phi^TTy$

1) $y_i \sim \mathcal{N}(w^T\phi(x_i), \beta^{-1})$

$p(Y|X, w, \beta) = \prod_{i=1}^{N} p(y_i|x, w, \beta)$

$\text{argmax}\, P(Y|X, w, \beta) = \text{argmin} -\ln\prod_{i=1}^{N} p(y_i|x, w, \beta) = E_{LS}(w)$

$= -\sum_{i=1}^{N}\ln\sqrt{\frac{\beta}{2\pi}} + \frac{1}{2}\sum_{i=1}^{N}\beta(w^T\phi x_i - y_i)^2$ ⇔ $E_{\text{weighted}}(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N} t_i\left[\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - y_i\right]^2$

$\Rightarrow \beta = t_i \Rightarrow y_i \sim \mathcal{N}(y_i|\, w^T\phi(x_i), t_i^{-1})$

t_i seen as the effective number of replicated observation

$$\mathrm{p}(\boldsymbol{x} \mid y = c) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}) \quad (3)$$

$$= \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_c)\right\} \quad (4)$$

P5 $\bar{\Phi} \in R^{N \times M}$ 入

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

Ridge regression

$y_i = w^T \phi(x_i)$

$E_{LS}(w) = \frac{1}{2} \sum_{i=1}^{N} (w^T \phi(x_i) - y_i)^2 + \frac{\lambda}{2} \|w\|_2^2$

$= \frac{1}{2} \left[\bar{\Phi} w - y \right]^T \left[\bar{\Phi} w - y \right] + \frac{\lambda}{2} \|w\|_2^2$

$= \frac{1}{2} \left[w^T \bar{\Phi}^T \bar{\Phi} w - 2 w^T \bar{\Phi}^T y + y^T y \right] + \frac{\lambda}{2} \|w\|_2^2$

$\frac{dE_{LS}}{dw} = \bar{\Phi}^T \bar{\Phi} w - \bar{\Phi}^T y + \lambda w \overset{!}{=} 0$

$(\bar{\Phi}^T \bar{\Phi} + \lambda I) w^* = \bar{\Phi}^T y$

$w^* = (\bar{\Phi}^T \bar{\Phi} + \lambda I)^{-1} \bar{\Phi}^T y$

Linear Regression

$y_i = w^T x_i$

$\hat{y} = w^T \hat{\Phi}(x_i)$

$E_{LS} = \frac{1}{2} \sum_{i=1}^{N} (w^T \phi(x_i) - y_i)^2$

$= \frac{1}{2} \left[\left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right) \cdot w - \begin{pmatrix} y \\ 0_M \end{pmatrix} \right]^T \left[\left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right) w - \begin{pmatrix} y \\ 0_M \end{pmatrix} \right]$

$= \frac{1}{2} \left[w^T \left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right)^T \left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right) w - 2 w^T \left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right)^T \begin{pmatrix} y \\ 0_M \end{pmatrix} + c \right]$

$\frac{dE_{LS}}{dw} = \left(\frac{\bar{\Phi}^T}{\sqrt{\lambda} I_M} \right) \left(\frac{\bar{\Phi}}{\sqrt{\lambda} I_M} \right) \cdot w - \left(\frac{\bar{\Phi}^T}{\sqrt{\lambda} I_M} \right) \begin{pmatrix} y \\ 0_M \end{pmatrix} \overset{!}{=} 0$

$(\bar{\Phi}^T \bar{\Phi} + \lambda I_M) \cdot w = \bar{\Phi}^T y$

$w^* = (\bar{\Phi}^T \bar{\Phi} + \lambda I_M)^{-1} \bar{\Phi}^T y$

P6 Because the training error will decrease if the model (with high degree polynomial) perfectly fit all the data

Best is 4th

Split data into training and val data

then compute the validation error

P7 $p(w, \beta | D) \propto p(Y | \bar{\Phi}, w, \beta) \, p(w | \beta)$

$\log p(w, \beta | D) = \sum_{i=1}^{N} \ln \left[\sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2} (w^T \phi(x_i) - y_i)^2 \right) \right] + \sum_{i=1}^{M} \ln \left[\sqrt{\frac{\beta}{2\pi}} \right]$

$+ \ln \frac{b_0^{a_0}}{\Gamma(a_0)} + (a_0 - 1) \ln \beta - b_0 \beta$

$N = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (x - \mu)^2 \right)$

$\sqrt{\beta^{-1} S_0} = \sigma$

$\frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{\beta^{-1} S_0}}$

$\log \sqrt{\frac{\beta}{2\pi \cdot S_0}}$

$= \frac{1}{2} \log \left(\frac{\beta}{2\pi \cdot S_0} \right)$

$= \frac{1}{2} \left[\log \beta - \log 2\pi - \log S_0 \right]$

$p(w, \beta | D) = \mathcal{N}(w | m_N, \beta^{-1} S_N) \, \text{Gamma}(\beta | a_N, b_N)$

Homework

Least squares regression

Problem 4: Let's assume we have a dataset where each datapoint, $(\boldsymbol{x}_i, y_i)$ 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:

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

Find the equation for the value of $\boldsymbol{w}$ 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 $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$ with regularization strength λ is equivalent to ordinary least squares regression with an augmented design matrix and target vector

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

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:

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

Then the conjugate prior for both $\boldsymbol{w}$ and β is

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

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

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

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

Hint: Expand $\log \mathrm{p}(\boldsymbol{w}, \beta \mid \mathcal{D})$ 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

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

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?

Comparison of Linear Regression Models

Problem 9: 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 $\boldsymbol{w}^* \in \mathbb{R}^D$ as

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

P8 $E_{ridge} = \frac{1}{2}\sum_{i=1}^{N}(w^T\phi(x_i) - y_i)^2 + \frac{\lambda}{2}w^Tw$

$\arg\min \quad \frac{dE}{dw} = (\Phi^T\Phi + \lambda I)w = \Phi^T y$

$w^* = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T y$

$\uparrow$

$\in R^{N\times M}$

If $N < M$ not inversible singularity

$\Phi^T\Phi \in M\times M$ ←

add λI to eliminate the

P9 ① $\hat{y} = w^* x_i = w^*_{new} x_{i new}$

$X w^* = X_{new} w^*_{new} = a X w_{new}$

$w^*_{new} = \frac{1}{a} w^*$

② $w^*_{new} = a(a^2 X^T X + \lambda_{new} I)^{-1} \cdot X^T y$

$= a(a^2 X^T X + a^2\lambda I)^{-1} X^T y$

$= \frac{1}{a}(X^T X + \lambda I)^{-1} X^T y$

$= \frac{1}{a} w^* = w^*_{new}$

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 $\boldsymbol{w}^*$ produces on $\boldsymbol{X}$.

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

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

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

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.