

# CS-E3210- Machine Learning Basic Principles

## Home Assignment - “Validation”

Your solutions to the following problems should be submitted as one single pdf which does not contain any personal information (student ID or name). The only rule for the layout of your submission is that for each problem there has to be exactly one separate page containing the answer to the problem. You are welcome to use the L<sup>A</sup>T<sub>E</sub>X-file underlying this pdf, available under <https://version.aalto.fi/gitlab/junga1/MLBP2017Public>, and fill in your solutions there.

## Problem 1: The Training Error is not the Generalization Error

Consider a folder  $\mathbb{X} = \{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}\}$  constituted by  $N$  webcam snapshots  $\mathbf{z}^{(i)}$ , each characterized by the features  $\mathbf{x}^{(i)} \in \mathbb{R}^d$  and labeled by the local temperature  $y^{(i)} \in \mathbb{R}$  during the snapshot. We would like to find out how to predict the temperature based solely from the feature vector  $\mathbf{x}$ . To this end, we will use linear predictors of the form:  $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$  with some weight vector  $\mathbf{w} \in \mathbb{R}^d$ .

Let us assume that the features  $\mathbf{x}$  and label  $y$  are related by a simple linear regression model:

$$y = \bar{\mathbf{w}}^T \mathbf{x} + \varepsilon \quad (1)$$

with some non-random weight vector  $\bar{\mathbf{w}} \in \mathbb{R}^d$  and random noise  $\varepsilon$ . We assume that the feature vector and noise are jointly normal with zero mean and covariance matrix  $\mathbf{C}$ , i.e.,  $(\mathbf{x}^T, \varepsilon)^T \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ . The feature vectors  $\mathbf{x}^{(i)}$  and labels  $y^{(i)}$  are independent and identically distributed (i.i.d.) realizations of  $\mathbf{x}$  and  $y$ .

- (a) Consider the predictor  $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$  for a particular fixed weight vector  $\mathbf{w} \in \mathbb{R}^d$ . What is the relation between the empirical risk (training error):

$$\mathcal{E}(\mathbf{w}|\mathbb{X}) := \frac{1}{N} \sum_{i=1}^N \left( y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} \right)^2$$

and the generalization error

$$\mathcal{E}(\mathbf{w}) = \mathbb{E}\{(y - \mathbf{w}^T \mathbf{x})^2\}?$$

- (b) Find a closed-form expression for the generalization error which involves the true (but unknown) weight vector  $\bar{\mathbf{w}}$  and covariance matrix  $\mathbf{C}$ .
- (c) According to your results in (b), how should we choose the weight vector  $\mathbf{w}$  such that the predictor  $h^{(\mathbf{w})}$  has small generalization error?

**Answer.**

## Problem 2: Overfitting in Linear Regression

Consider the problem of predicting a real-valued label (target)  $y \in \mathbb{R}$  based on the features  $\mathbf{x} \in \mathbb{R}^d$ . Given a labeled dataset  $\mathbb{X}$  consisting of  $N$  labeled data points with feature vectors  $\mathbf{x}^{(i)} \in \mathbb{R}^d$  and labels  $y^{(i)} \in \mathbb{R}$ , we learn a linear predictor  $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$  by minimizing the empirical risk:

$$\mathcal{E}(h^{(\mathbf{w})}(\cdot)|\mathbb{X}) := \frac{1}{N} \sum_{i=1}^N (y^{(i)} - h^{(\mathbf{w})}(\mathbf{x}^{(i)}))^2 = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2.$$

If the dataset  $\mathbb{X}$  is small compared to the number  $d$  of features, i.e.,  $N \leq d$ , the feature vectors  $\{\mathbf{x}^{(i)}\}_{i=1}^N$  are typically linearly independent. Show that in this case, there exists a weight vector  $\mathbf{w}_0$  so that  $\mathcal{E}(h^{(\mathbf{w}_0)}(\cdot)|\mathbb{X}) = 0$ .

**Answer.**

The solution to minimization problem can be written in matrix as  $\mathbf{X}\mathbf{w} = \mathbf{y}$  where the rows of the matrix  $\mathbf{X}$  are the features  $\mathbf{x}^i$ . This can be further written in form  $\mathbf{X}'\mathbf{w}' = 0$  where  $\mathbf{X}' = [\mathbf{X}, -\mathbf{y}]$  is a  $N \times (d+1)$  matrix and  $\mathbf{w}' = [\mathbf{w}, 1]^T$ . Because the feature vectors were considered to be independent the rank of matrix  $\mathbf{X}'$  is  $N$  and thus there are  $d+1-N$  dependent columns in  $\mathbf{X}'$  and the  $\mathbf{X}'$  can be written as  $\mathbf{X}' = [\mathbf{A}, \mathbf{AB}]$  where  $\mathbf{A}$  is a  $N \times N$  matrix and  $\mathbf{B}$  is a  $N \times (d+1-N)$  matrix. Now the solution to  $\mathbf{X}'\mathbf{w}' = 0$  can be written as  $\mathbf{w}' = [-\mathbf{B}, \mathbf{I}]^T$  where  $\mathbf{I}$  is a  $(d+1-N) \times (d+1-N)$  identity matrix. Now any linear combination of columns of  $\mathbf{w}'$  is a solution to  $\mathbf{X}'\mathbf{w}' = 0$  and thus  $\mathbf{w}_0$  is obtained also as linear combinations of those columns and removing the last element.

### Problem 3: Probability of Sampling Disjoint Datasets

Consider a dataset  $\mathbb{X}$  which contains  $N=10$  different labeled webcam snapshots. We then create a training dataset  $\mathbb{X}^{(\text{train})}$  by copying 3 randomly selected elements of  $\mathbb{X}$ . Moreover, we create a validation dataset  $\mathbb{X}^{(\text{val})}$  by copying another 2 randomly selected elements of  $\mathbb{X}$ . What is the probability that the training set and the validation set are disjoint, i.e., they have no snapshot in common?

**Answer.**

For disjoint training and validation sets the training set can be chosen in  $\binom{10}{3}$  ways and then the disjoint validation set in  $\binom{7}{2}$  way resulting in total  $\binom{10}{3}\binom{7}{2}$  ways. Number of all possible ways to select training and validation sets is  $\binom{10}{3}\binom{10}{2}$  and thus the probability that the sets are disjoint is

$$P = \frac{\binom{10}{3}\binom{7}{2}}{\binom{10}{3}\binom{10}{2}} = \frac{\binom{7}{2}}{\binom{10}{2}}, \quad (2)$$

which is approximately 0.45 .

## Problem 4: The Histogram of the Prediction Error

Consider the dataset  $\mathbb{X}$  available at <https://version.aalto.fi/gitlab/junga1/MLBP2017Public/tree/master/Validation/p3data>. For your convenience, this dataset is already split into a training dataset  $\mathbb{X}^{(\text{train})} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N^{(\text{train})}}$  (features  $\mathbf{x}^{(i)} \in \mathbb{R}^5$  stored in the file “X\_train.txt”, labels  $y^{(i)} \in \mathbb{R}$  stored in “y\_train.txt”) and the validation dataset  $\mathbb{X}^{(\text{val})}$  (stored in the files “X\_validation.txt” and “y\_validation.txt”). We want to predict the label  $y$  given the features  $\mathbf{x}$  using a linear predictor  $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ .

- (a) Learn a linear predictor  $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$  by choosing the weight vector  $\mathbf{w}$  such that the empirical risk (using squared error loss)

$$\mathcal{E}(h^{(\mathbf{w})}(\cdot) | \mathbb{X}^{(\text{train})}) = (1/|\mathbb{X}^{(\text{train})}|) \sum_{(\mathbf{x}, y) \in \mathbb{X}^{(\text{train})}} (y - h^{(\mathbf{w})}(\mathbf{x}))^2$$

obtained for the training dataset  $\mathbb{X}^{(\text{train})}$  is as small as possible. Denote this optimal weight vector by  $\mathbf{w}_{\text{opt}}$ .

- (b) Select a test set  $\mathbb{X}^{(\text{test})}$  by copying  $N^{(\text{test})} = 10$  randomly selected data points  $(\mathbf{x}^{(i)}, y^{(i)})$  out of the validation dataset  $\mathbb{X}^{(\text{val})}$ . Evaluate the prediction error of  $h^{(\mathbf{w}_{\text{opt}})}$  by computing the empirical risk

$$\mathcal{E}(h^{(\mathbf{w}_{\text{opt}})}(\cdot) | \mathbb{X}^{(\text{test})}) = (1/|\mathbb{X}^{(\text{test})}|) \sum_{(\mathbf{x}, y) \in \mathbb{X}^{(\text{test})}} (y - h^{(\mathbf{w}_{\text{opt}})}(\mathbf{x}))^2$$

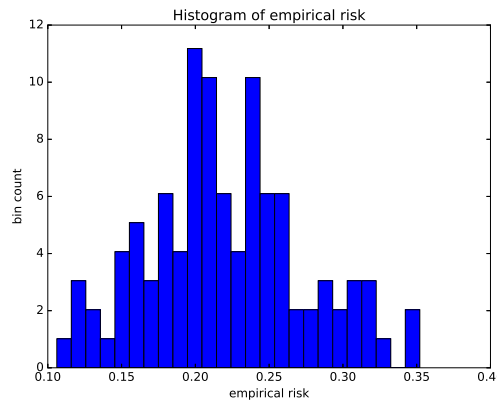
obtained for the test dataset  $\mathbb{X}^{(\text{test})}$ .

- (c) Repeat step (b)  $K = 100$  times, involving another test dataset  $\mathbb{X}^{(\text{test})}$  each time due to randomness, and generate a histogram of the prediction error. In view of the obtained histogram, is it a good idea to evaluate the error only for one single test dataset ?

**Answer. (a)** The optimal weight vector  $\mathbf{w}_{\text{opt}}$  is obtained using the basic least squares solution which gives  $\mathbf{w}_{\text{opt}} = (-0.18298954, -0.18298954, 1.51048792, 0.84094088, 0.1355004, 0.43844228)$ , where the first element is the constant term.

**(b)** The prediction error obtained using the above  $\mathbf{w}_{\text{opt}}$  and ten random samples is 0.21 .

**(c)** The generated histogram is shown in figure 1. According to histogram evaluating error based on only single set may easily under- or overestimate the true error.



**Figure 1:** Histogram of prediction error corresponding to 100 samples

## Problem 5: K-fold Cross Validation

Consider a dataset  $\mathbb{X} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$  containing a total of  $N = 20$  snapshots ("winter??.jpg" or "autumn??.jpg" available at <https://version.aalto.fi/gitlab/junga1/MLBP2017Public/tree/master/Validation/WinterFall>) which are either taken either during winter ( $y^{(i)} = -1$ ) or autumn ( $y^{(i)} = 1$ ). We aim at finding a classifier which classifies an image as "winter" ( $\hat{y} = -1$ ) if  $h^{(\mathbf{w})}(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) < 1/2$  or as "autumn" ( $\hat{y} = 1$ ) if  $h^{(\mathbf{w})}(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) \geq 1/2$ . Let us collect the image pixels  $i$  which belong to the top-left square of size  $r \times r$  pixels by  $\mathcal{R}_r$ . For a given model size  $r$ , define the hypothesis space

$$\mathcal{H}^{(r)} := \{h^{(\mathbf{w})}(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}), \text{ with } w_i = 0 \text{ for } i \notin \mathcal{R}_r\}.$$

In order to find the best choice for  $r$ , we will use "K-fold cross validation" (with  $K = 5$ ) in order to assess the quality of the hypothesis space  $\mathcal{H}^{(r)}$  for each  $r \in \{1, 10, 20, 50, 100, 200\}$ . This works as follows:

- step 1: randomly partition the dataset  $\mathbb{X}$  into  $K = 5$  equal-size subsets  $\mathbb{X}^{(1)}, \dots, \mathbb{X}^{(K)}$ .
- step 2: choose one of the subsets  $\mathbb{X}^{(t)}$  as validation set
- step 3: choose the remaining subsets as the training set  $\mathbb{X}^{(\text{train}),t} = \mathbb{X} \setminus \mathbb{X}^{(t)}$
- step 4: find optimal classifier  $h^{(\mathbf{w}_{\text{opt},t})}(\cdot) \in \mathcal{H}^{(r)}$  which minimizes empirical risk

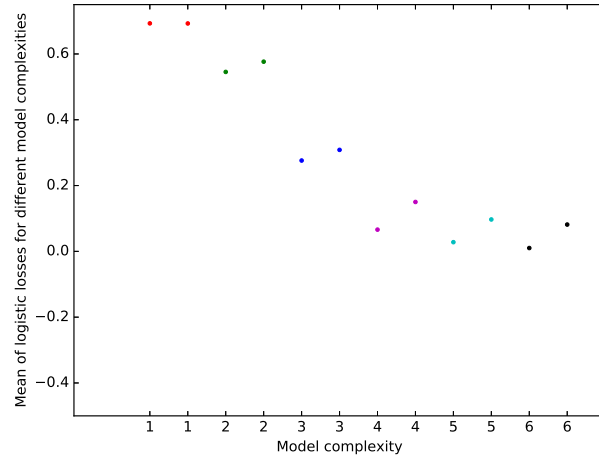
$$\mathcal{E}\{h^{(\mathbf{w})}|\mathbb{X}^{(\text{train}),t}\} = (5/N) \sum_{(\mathbf{x},y) \in \mathbb{X}^{(\text{train}),t}} L((\mathbf{x}, y), h^{(\mathbf{w})}(\cdot))$$

using logistic loss  $L((\mathbf{x}, y), h^{(\mathbf{w})}(\cdot)) = \ln(1 + \exp(-y(\mathbf{w}^T \mathbf{x})))$ . You might use gradient descent for determining the optimal weight vector  $\mathbf{w}_{\text{opt},t}$ . (see HA3)

- step 5: compute validation error  $\mathcal{E}\{h^{(\mathbf{w}_{\text{opt},t})}|\mathbb{X}^{(t)}\}$
- step 6: repeat from step 2 until every subset  $\mathbb{X}^{(t)}$  has been used exactly once for validation
- step 7: compute the average training error  $E^{(\text{train})}(r) = (1/5) \sum_{t=1}^5 \mathcal{E}\{h^{(\mathbf{w}_{\text{opt},t})}|\mathbb{X}^{(\text{train}),t}\}$   
and the average validation error  $E^{(\text{val})}(r) = (1/5) \sum_{t=1}^5 \mathcal{E}\{h^{(\mathbf{w}_{\text{opt},t})}|\mathbb{X}^{(t)}\}$

Implement this procedure for each choice  $r \in \{1, 10, 20, 50, 100, 200\}$ . Plot the average training error  $E^{(\text{train})}(r)$  and the average validation error  $E^{(\text{val})}(r)$  as functions of the model complexity  $r$ . What is the best model complexity for the classification problem at hand? Justify your answer.

**Answer.** The average of training and validation errors for different model complexities is shown in figure 2. According to figure 2 the best choice for model complexity is  $r = 50$  because for higher model complexities  $r$  there is only a small improvement in average validation error and occasionally the validation error might get even larger while the computational complexity increases significantly.



**Figure 2:** Average of training and validation error for model complexities  $r = 1$  (red),  $r = 10$  (green),  $r = 20$  (blue),  $r = 50$  (magenta),  $r = 100$  (cyan),  $r = 200$  (black)