Machine Learning

Exercise Sheet 5

Winter Term 2023 — Available: 01.12.2023
Prof. Dr. Niels Landwehr — Hand in until: 08.12.2023 11:59am
Dr. Ujjwal — Exercise sessions: 11.12.2023/13.12.2023

Task 1 – Bias-Variance Decomposition of Error Estimator [10 points]

In this exercise, we prove the bias-variance decomposition for the error of the error estimator of a fixed model $f_{\boldsymbol{\theta}^*}$ (Slide 13 in lecture).
In the lecture, we defined the error estimator

$$\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*}) = \frac{1}{\bar{N}} \sum_{n=1}^{\bar{N}} \ell_{eval}(\bar{y}_n, f_{\boldsymbol{\theta}^*}(\bar{\mathbf{x}}_n)) \tag{1}$$

and the true risk

$$R(f_{\theta^*}) = \iint \ell_{eval}(y, f_\theta(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy. \tag{2}$$

To reduce notational clutter, define $\hat{R} := \hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}^*})$ and $R := R(f_{\boldsymbol{\theta}}{}^*)$. Prove the bias-variance decomposition for the expected squared error of the error estimator:

$$\mathbb{E}[(\hat{R} - R)^2] = Bias[\hat{R}]^2 + Var[\hat{R}] \tag{3}$$

where according to the definitions in the lecture

$$Bias[\hat{R}]^2 = (\mathbb{E}[\hat{R}] - R)^2 \tag{4}$$
$$Var[\hat{R}] = \mathbb{E}[(\hat{R} - \mathbb{E}[\hat{R}])^2]. \tag{5}$$

Hint: remember that $\hat{R}$ is a random variable, because it depends on a random sample $\mathcal{T}$ of data, while R is a simple scalar value. The model $f_{\boldsymbol{\theta}^*}$ is considered fixed.

Task 2 – Cross-Validation and Hyperparameter Tuning (Programming) [20 points]

In this programming task, we take a look at the toy sine data set discussed in the regularization lecture and implement cross-validation and hyperparameter tuning for this data set.
In the notebook *Exercise05_Task2.ipynb* you find example code that implements the toy sine data set, a polynomial feature map, and a function for learning a regularized polynomial regression of degree d on the data. There is also code for plotting the data and the learned model. By changing the value of the variable d, you can fit polynomial models of different degrees to the data. You can observe how the model underfits, reasonably fits or overfits the data depending on d. Alternatively, you can leave the polynomial degree at $d = 10$ and use the regularization weight λ to prevent the model from overfitting (last argument in call to *fit_ridge_regression*).

a) In the notebook *Exercise05_Task2.ipynb*, complete the method *crossval_split* such that it returns the training and test set in the k-th fold (iteration) of a K-fold cross-validation.

b) Now write a method for tuning the hyperparameter d, keeping $\lambda = 0$ fixed. For each $d \in \{0, 1, ..., 10\}$, your method should run a cross-validation on the training data using a model of degree d. The method should then pick the value of d that has resulted in the lowest error estimate from the cross-validation, and retrain the model on all of the data using this d. Plot the resulting model using the example code for plotting provided.

c) Now write a method for tuning the hyperparameter λ, keeping $d = 10$ fixed. For each $\lambda \in \{10^0, 10^{-1}, ..., 10^{-9}\}$, your method should run a cross-validation on the training data using a model with regularization λ. The method should then pick the value of λ that has resulted in the lowest error estimate from the cross-validation, and retrain the model on all of the data using this λ. Plot the resulting model using the example code for plotting provided.

Task 3 – Confidence Interval [10 points]

Assume we have trained a binary classification model $f_{\boldsymbol{\theta}}$ and evaluate it on independent test data $\mathcal{T} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{10}, y_{10})\}$. The result of the evaluation is as follows:

$f_{\boldsymbol{\theta}}(\mathbf{x}_n)$	y_n
0	0
1	1
1	0
1	1
0	1
1	0
1	1
0	0
0	1
1	1

Compute the error estimate $\hat{R}_{\mathcal{T}}(f_{\boldsymbol{\theta}})$. Compute a two-sided confidence interval around the error estimate with a confidence level of 95%. That is, choose a confidence level such that if we repeat the probabilistic process of drawing the data $\mathcal{T}$ and computing the confidence interval, the interval would contain the true risk in approximately 95% of the repetitions.

A table for looking up the inverse cumulative distribution function Φ^{-1} of the standard normal distribution can be found here: `https://faculty.biu.ac.il/~shnaidh/zooloo/library/normal.3.pdf`.