Robot Learning

Winter Semester 2020/2021, Homework 3

Prof. Dr. J. Peters, J. Watson, J. Carvalho, J. Urain and T. Dam

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 35
Due date: Midnight, 28 December 2020

Name, Surname, ID Number

Problem 3.1 Machine Learning in a Nutshell [35 Points]

For this exercise you will use a dataset, divided into training set and validation set (download them in Moodle). The first row is the vector $\boldsymbol{x}$ and the second row the vector $\boldsymbol{y}$.
Based on this data, we want to learn a function mapping from $\boldsymbol{x}$ values to $\boldsymbol{y}$ values, of the form $\boldsymbol{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x})$.

Please upload the code you developed in the corresponding section in Moodle.

a) Supervised vs Unsupervised Learning [2 Points]

Briefly explain the differences between supervised and unsupervised learning. Is the above a supervised or unsupervised learning problem? Why?

b) Regression vs Classification [2 Points]

Supervised learning is typically divided into regression and classification tasks. Briefly explain what are the differences between regression and classification.

c) Linear Least Squares [4 Points]

Consider the training set above to calculate features $\boldsymbol{\phi}(x)$ of the form $[\sin(2^i x)]_{i=0\ldots n-1}$. Compute the feature values when n is 2, 3 and 9 (i.e., when using 2, 3 and 9 features). Use the linear least squares (LLS) method to predict output values y for input values $x \in \{0, 0.01, 0.02, \ldots, 6\}$ using the different numbers of features. Attach a single plot showing the three resulting predictions when using 2, 3 and 9 features (i.e., having x and y as axes).

d) Training a Model [2 Points]

The root mean square error (RMSE) is defined as RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i^{\text{true}} - y_i^{\text{predicted}}\right)^2}$, where N is the number of data points. Using the LLS algorithm implemented in the previous exercise, train a different model for each of the number of features between 1 and 9, i.e., [1,2,3...,9]. For each of these models compute the corresponding RMSE for the training set. Attach a plot where the x-axis represents the number of features and the y-axis represents the RMSE.

e) Model Selection [4 Points]

Using the models trained in the previous exercise, compute the RMSE of each of these models for the validation set.
Compare in one plot the RMSE on the training set and on the validation set. How do they differ? Can you explain what is the reason for these differences? (Hint: remember the plot from Exercise c)) What is the number of features that you should use to achieve a proper modeling?

f) Cross Validation [8 Points]

K-fold cross validation is a common approach to estimate the test error when the dataset is small. The idea is to randomly divide the training set into K different datasets. Each of these datasets is then used as validation set for the model trained from the remaining $K-1$ datasets. The resulting vector of errors $\boldsymbol{E} = [e_1 ... e_K]$ can now be used to compute a distribution (typically by fitting a Gaussian distribution). When K is equal to the number of data points, K-fold cross validation takes the name of leave-one-out cross validation (LOO).

Apply LOO using only the training set and compute the mean/variance of the RMSE for the learned models. Repeat for the models with the number of features between 1 and 9, i.e., [1,2,3...,9]

Attach a plot showing the mean/variance (as a distribution) of the RMSE computed using LOO and having on the x-axis the number of features and on the y-axis the RMSE. Which is the optimal number of features now? Discuss the results obtained and compare against model selection using train/validation set.

g) Kernel Functions [2 Points]

A kernel function $k\left(\mathrm{x}_i, \mathrm{x}_j\right)$ is given by the inner product of two feature vectors. Write out the kernel function for the previous set of features where $n = 3$.

h) Kernel Regression [6 Points]

The kernel function in the previous question required explicit definition of the type and number of features, which is often difficult in practice. Instead, we can use a kernel that defines an inner product in a (possibly infinite dimensional) feature space.
Using the training set and an exponential squared kernel $k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(-\frac{1}{\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$ with $\sigma = 0.15$, predict output values y for input values $x \in \{0, 0.01, \ldots, 6\}$. Attach a plot of your results.
(Hint: use standard kernel regression: $f(\boldsymbol{x}) = \boldsymbol{k}^\top \boldsymbol{K}^{-1} \boldsymbol{y}$ with $\boldsymbol{K}_{ij} = k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ and $\boldsymbol{k}_i = k(\boldsymbol{x}, \boldsymbol{x}_i)$).

Compute the RMSE on the validation set for the kernel regression model. Compare it with the RMSE of the best LLS model you found.

i) Derivation [5 Points]

Explain the concept of ridge regression and why/when it is used. Derive its final equations presented during the lecture.
(Hint: remind that for normal linear regression the cost function is $J = \frac{1}{2}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2$)
(Hint 2: use matrix notation)