Artificial Intelligence II
Winter term 2020/2021
January 8, 2021

Daniel Wulff
wulff@rob.uni-luebeck.de

Exercise sheet 8

Submission deadline: 10:00, January 15, 2021 - *Remember to submit Exercise sheet 7 as well!*

Task 1: Support Vector Regression (SVR) (20 points)

The dual QP of the kernel SVR is given by:

$$\text{Maximize} \quad -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m}\varepsilon(\alpha_i + \alpha_i') + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i')$$

$$\text{subject to} \quad \sum_{i=1}^{m}(\alpha_i - \alpha_i') = 0$$

$$\text{with} \quad 0 \le \alpha, \alpha' \le C.$$

Download the given MATLAB code snippets from the Moodle course.

- `svrTrain.m`

  ```
  function [alphas, alphas_dash, d] = svrTrain( X, y, K, epsilon )
  ```

  For a given data point matrix X, a given vector of function values **y**, a given kernel $K(\mathbf{x}, \mathbf{y})$, and the SVR parameter ε, this function uses the kernel SVR for finding a regression plane for the given data points and returns the vectors of the Lagrange multipliers α_i and α_i', and the distance d. It uses the MATLAB `quadprog` command. Implement the needed functionality.

- `svrProduce.m`

  ```
  function f = svrProduce( X, K, alphas, alphas_dash, d, x1, x2 )
  ```

  For a given data point matrix X, a given kernel $K(\mathbf{x}, \mathbf{y})$, some trained vectors of Lagrange multipliers α_i and α_i', a trained distance d, and some given domain(s) $\mathbf{x}_1$ and $\mathbf{x}_2$, this function approximates and returns the function values $f(x)$ in the given domain(s). Implement the needed functionality.

- `svrPlot.m`

  Auxiliary function. **Do not edit this function.**

- `svrTest_1D.m`

  This script tests your implementations. Some data points of the function $f(x) = (x-5)^2 + 2 + \eta$, where η is white noise, are given. Set up a proper kernel function and SVR parameter ε, and use your implementations to approximate $f(x)$ in the given domain. The script generates one figure which will automatically be saved as a PNG file.

- `svrTest_2D.m`

  This script tests your implementations. Some data points of the two-dimensional standard normal distribution function are given. Set up a proper kernel function and SVR parameter ε, and use your implementations to approximate $f(x_1, x_2)$ in the given domains. The script generates one figure which will automatically be saved as a PNG file.

a) Before implementing, set up the QP in its standard form. Thus, derive the matrices Q, A and A_{equ} as well as the vectors **c**, **b** and $\mathbf{b}_{equ}$. Remember the upper and lower bounds. How does the variable vector of the QP look like in this scenario? **Derive** does not mean "just write them down". How does the vector **x** look like in this scenario? (6 points)

b) Implement the missing functionality in `svrTrain.m`. Your implementation should be able to handle data of any finite dimension. (5 points)

c) Implement the missing functionality in `svrProduce.m`. Your implementation should be able to handle one-dimensional and two-dimensional data. (3 points)

d) Have a look at the two test instances. Find a proper combination of the SVR parameters ε and C. To help you with this task, in the title of each plot, the root-mean-square (RMS) error is shown. For each test scripts, find **one** combination of parameters which returns the best results from your point of view. Briefly comment on your results. Zip your implementations and the generated PNG files from the test scripts and upload your archive to the Moodle course. (6 points)