UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ROBOTICS AND COGNITIVE SYSTEMS

Artificial Intelligence II
Winter term 2019/2020
December 11, 2020

Daniel Wulff
wulff@rob.uni-luebeck.de

Exercise sheet 6

Submission deadline: 10:00, December 18, 2020

Task 1: Polynomial kernels (10 points)

The general form of a polynomial kernel is given by

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + \gamma)^d.$$

a) Assume $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$. For $d = 2$, $\gamma = 1$ and $F : \mathbb{R}^2 \to \mathbb{R}^6$ with

$$F(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)^T,$$

show that $K(\mathbf{x}, \mathbf{z}) = F(\mathbf{x})^T \cdot F(\mathbf{z})$. (5 points)

b) Assume $\mathbf{x}, \mathbf{z} \in \mathbb{R}^2$. For $d = 3$ and $\gamma = 1$, find a mapping F, so that $K(\mathbf{x}, \mathbf{z}) = F(\mathbf{x})^T F(\mathbf{z})$. (5 points)

Task 2: Separating planes. . . you know. . . (10 points)

In the following, the dual QP given below should be solved by using a kernel $K(\mathbf{x}_i, \mathbf{x}_j)$

$$\text{Maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j F(\mathbf{x}_i)^T F(\mathbf{x}_j) \quad \text{s.t.} \quad \alpha_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

Considering the optimal Lagrange multipliers $\hat{\alpha}_i$, the *solution function* is given by

$$\hat{f}(\mathbf{u}) = \sum_{i=1}^{\hat{n}} \hat{\alpha}_i y_i K(\mathbf{u}, \mathbf{x}_i) + d_0.$$

Following the Karush-Kuhn-Tucker conditions, d_0 can be determined by solving $y_j \hat{f}(\mathbf{u}_j) = 1$ for any $\mathbf{u}_j$ for which $\hat{\alpha}_j > 0$. For numerical stability, the average of all these solutions is used for d_0. A classification rule (or classification function) for an arbitrary data point $\mathbf{u}$ is then given by

$$\hat{G}(\mathbf{u}) = \text{sign}(\hat{f}(\mathbf{u})).$$

Download the given MATLAB code snippets from the Moodle course.

- `maxMarg06.m`

  ```
  function [alphas, idx] = maxMarg06( X, y, K )
  ```

  For a given data point matrix $\mathbf{X}$, a given vector of class labels $\mathbf{y}$ with $y_i \in \{-1, +1\}$, and a given kernel $K(\mathbf{x}, \mathbf{y})$, this function uses dual Quadratic Programming for finding a separating plane between the data points of the two classes and returns the vector of the Lagrange multipliers α_i, and the indices of the non-zero α_i. Implement the needed functionality. (Attention: For `quadprog`, set upper boundaries (`ub`) of α_i to 1000.)

- `maxMargTest06.m`

  This script tests your implementation. From your obtained $\hat{\alpha}_i$, the classification function $\hat{G}(\mathbf{u})$ is calculated for a given grid of two-dimensional input data. The script generates one figure which will automatically be saved as a PNG file. **This time, you have to add some functionality in the test script, too.** Mainly, you have to add the calculation of d_0 and the values of $\hat{G}(\mathbf{u})$. Implement the needed functionality.

Implement the missing functionality in `maxMarg06.m` and `maxMargTest06`. Zip your implementations and the generated PNG file from the `maxMargTest06.m` script and upload your archive to the Moodle course. (10 points)