Machine Learning I at TU Berlin

Assignment 7 - Group PTHGL

December 17, 2018

Exercise 1: The Dual SVM

a)

The Lagrange function for the hard margin SVM is:

$$\Lambda(\omega, \theta, \alpha) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\omega^T x_i + \theta) - 1)$$

b)

Minimize over ω and θ:

$$\frac{\partial \Lambda}{\partial \omega} = \omega - \sum_{i=1}^{N} \alpha_i y_i x_i = 0$$

$$\omega^* = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\frac{\partial \lambda}{\partial \theta} = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

Insert in $\Lambda(\omega, \theta, \alpha)$:

$$
\begin{aligned}
\Lambda^* &= \frac{1}{2}(\omega^*)^T\omega^* - \sum_{i=1}^{N} \alpha_i y_i (\omega^*)^T x_i \underbrace{- \sum_{i=1}^{N} \alpha_i y_i}_{=0} \theta + \sum_{i=1}^{N} \alpha_i \\
&= \frac{1}{2}(\omega^*)^T\omega^* - \sum_{i=1}^{N} \alpha_i y_i (\omega^*)^T x_i + \sum_{i=1}^{N} \alpha_i \qquad \longrightarrow \omega^* = \sum_{i=1}^{N} \alpha_i y_i x_i \\
&= \frac{1}{2}\left(\sum_{j=1}^{N} \alpha_j y_j x_j\right)^T \left(\sum_{k=1}^{N} \alpha_k y_k x_k\right) - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{l=1}^{N} \alpha_l y_l x_l\right)^T x_i + \sum_{i=1}^{N} \alpha_i \\
&= \frac{1}{2}\sum_{j,k=1}^{N} \alpha_j \alpha_k y_j y_k {x_j}^T x_k - \sum_{i,l=1}^{N} \alpha_i \alpha_l y_i y_l {x_l}^T x_i + \sum_{i=1}^{N} \alpha_i \\
&= -\frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j {x_i}^T x_j + \sum_{i=1}^{N} \alpha_i
\end{aligned}
$$

We solve the dual problem for α^* and therefore can also get our primal problem solution for ω^* with $\omega^* = \sum_{i=1}^{N} \alpha_i y_i x_i$.
If our ω^* and α^* satisfy the Karush-Kuhn-Tucker (KKT) conditions, then we have our solution for the primal and dual problem.

c)

The primal problem don't have a kernelized version. To solve a non-linear problem the data can be transformed in a feature space $x \mapsto \phi(x)$ and afterwards solve ω^* in the higher dimension:

$$
\begin{aligned}
&\min_{\omega,\theta} \|\omega\|^2 \\
&\text{subject to} y_i(\omega^T \phi(x_i) + \theta) \geq 1
\end{aligned}
$$

For the dual problem we can use the kernel trick $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ and get as optimization problem:

$$
\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)
$$

$$
\text{subject to } \alpha_i \geq 0, i = 1, \ldots N, \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0
$$

Exercise 2: SVMs and Quadratic Programming

a)

For the dual program

$$\min_{\alpha} \frac{1}{2}\alpha^T (yy^T K(X, X'))\alpha - 1^T \alpha$$
$$\text{subject to } \alpha_i \geq 0, i = 1, \dots N, \text{ and } y^T \alpha = 0$$

we will build the matrices P, q, G, h, A, b for the CVXOPT solver.

P can be constructed with the outer product of y and the element wise multiplication with the kernel matrix K. q realizes the summation of the alpha values. Since we originally had a maximazation problem, the signs are switched of the objective function. Therefore q is filled with -1.

$$P = yy^T \circ K, \qquad K = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,n} \\ k_{2,1} & k_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ k_{n,1} & \cdots & \cdots & k_{n,n} \end{bmatrix}, \qquad k_{i,j} = k(x_i, x_j), \quad q = \begin{bmatrix} -1^{(1)} \\ -1^{(2)} \\ \vdots \\ -1^{(n)} \end{bmatrix}$$

G and h realize the first inequality constraints. All alpha values need to be larger than zero. Since the orientation can of the element wise comparison can not be switched, G is a identity matrix scaled with -1,

$$G = \begin{bmatrix} -1^{(1)} & & \cdots & 0 \\ 0 & -1^{(2)} & & \\ \vdots & & \ddots & \\ 0 & & & -1^{(n)} \end{bmatrix}, \quad h = \begin{bmatrix} 0^{(1)} \\ 0^{(2)} \\ \vdots \\ 0^{(n)} \end{bmatrix}$$

A and b realize the second constraint A realizes the sum of the product alpha with the corresponding y, therefore A is simply the transpose of y. b is the condition which has to be met, therefore b is 0.

$$A = y^T, \quad b = 0$$