

Exercise Sheet 6

Notes on Dual Problem, KKT-Optimality and Slater's Condition

Consider an optimization problem in the **canonical** form:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

The **Lagrange function** $\mathcal{L}: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ is defined as a weighted sum of the objective and constraint functions:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p \mu_i h_i(\mathbf{x}),$$

where \mathbf{x} is called **primal** and $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ the **dual** variables.

The (Lagrange) **dual function** $g: \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ is defined as:

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{x} \in \text{domain}(f_0)} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}).$$

The convex optimization problem

$$\begin{aligned} & \underset{(\boldsymbol{\lambda}, \boldsymbol{\mu})}{\text{maximize}} && g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ & \text{subject to} && \boldsymbol{\lambda} \succeq \mathbf{0} \end{aligned}$$

is called the (Lagrange) **dual problem**.

In the Lagrange optimization framework the KKT-conditions are used to find the primal and dual optimal solutions.

Theorem 1 (Optimality Conditions) *For any optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ must satisfy KKT-conditions:*

$$\begin{aligned} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= 0, && (\text{stationarity}) \\ f_i(\mathbf{x}^*) &\leq 0, && (\text{primal feasibility}) \\ h_i(\mathbf{x}^*) &= 0, && (\text{primal feasibility}) \\ \lambda_i^* &\geq 0, && (\text{dual feasibility}) \\ \lambda_i^* \cdot f_i(\mathbf{x}^*) &= 0 && (\text{complementary slackness}) \end{aligned}$$

For any convex problem, the KKT-conditions are sufficient for $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ to be optimal with zero duality gap.

Definition 2 (Slater's Condition) We say that a convex optimization problem

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

satisfies the Slater's condition if there exists a strictly feasible point \mathbf{x} , i.e., $f_i(\mathbf{x}) < 0$ and $h_j(\mathbf{x}) = 0$ for all i, j .

Theorem 3 (Slater's Theorem) For any convex problem for which Slater's condition holds, the KKT-conditions provide necessary and sufficient condition for $(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ to be primal and dual optimal with zero duality gap.

Exercise 1: Dual formulation of the Soft-Margin SVM

(a) Soft-margin SVM is given by a convex optimization problem: the objective is convex and the inequality constraints are linear (therefore also convex). Furthermore, the Slater's Theorem guarantees that if there is a feasible point $(\mathbf{w}, b, \boldsymbol{\xi})$ which strictly satisfies the inequality constraints, then strong duality holds. Here, for any (\mathbf{w}, b) we can always choose sufficiently large values for the slack variables $\boldsymbol{\xi}$ such that all inequality constraints are strictly satisfied. Therefore, strong duality (in contrast to the hard-margin) holds always for the soft-margin formulation.

(b) First we rewrite the optimization problem in the canonical form:

$$\begin{aligned} & \underset{\mathbf{w}, b, \boldsymbol{\xi}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{subject to} && 1 - \xi_i - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \leq 0, \quad i = 1, \dots, n \\ & && -\xi_i \leq 0, \quad i = 1, \dots, n \end{aligned}$$

The Lagrangian is given as

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - \xi_i - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b)) + \sum_{i=1}^n \beta_i (-\xi_i),$$

where $(\mathbf{w}, b, \boldsymbol{\xi})$ are the primal and $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are the dual variables. The corresponding dual function is given as

$$\begin{aligned}
g(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - \xi_i - y_i (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b)) + \sum_{i=1}^n \beta_i (-\xi_i) \\
&= \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \left(\frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) \right) + \left(- \sum_{i=1}^n \alpha_i y_i b \right) + \left(C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n \beta_i \xi_i \right) + \sum_{i=1}^n \alpha_i \\
&= \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \left(\frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) \right) + \left(-b \sum_{i=1}^n \alpha_i y_i \right) + \left(\sum_{i=1}^n \xi_i (C - \alpha_i - \beta_i) \right) + \sum_{i=1}^n \alpha_i \\
&= \inf_{\mathbf{w}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) \right\} + \inf_b \left\{ -b \sum_{i=1}^n \alpha_i y_i \right\} + \inf_{\boldsymbol{\xi}} \left\{ \sum_{i=1}^n \xi_i (C - \alpha_i - \beta_i) \right\} + \sum_{i=1}^n \alpha_i
\end{aligned}$$

Note that the minimization over b and $\boldsymbol{\xi}$ is completely unrestricted. Therefore, the only way for the infimum to be bigger than $-\infty$ if the constraints $\sum_{i=1}^n \alpha_i y_i = 0$ and $C - \alpha_i - \beta_i = 0$ are satisfied. This is in agreement with the results below. To find the minimizing arguments $(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*)$ we set the gradient of the corresponding terms to zero as follows:

$$\begin{aligned}
\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i) &= \mathbf{0} \quad \implies \quad \mathbf{w}^* = \sum_{i=1}^n \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i) \\
\frac{\partial \mathcal{L}}{\partial b} = - \sum_{i=1}^n \alpha_i y_i &= 0 \quad \implies \quad \sum_{i=1}^n \alpha_i y_i = 0 \\
\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \beta_i &= 0 \quad \xRightarrow{\beta_i \geq 0} \quad 0 \leq \alpha_i \leq C
\end{aligned}$$

We now eliminate \mathbf{w} by inserting the value \mathbf{w}^* back into the equation and respecting the corresponding constraints:

$$\begin{aligned}
g(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \inf_{\mathbf{w}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) \right\} + \inf_b \left\{ -b \sum_{i=1}^n \alpha_i y_i \right\} + \inf_{\boldsymbol{\xi}} \left\{ \sum_{i=1}^n \xi_i (C - \alpha_i - \beta_i) \right\} + \sum_{i=1}^n \alpha_i \\
&= \frac{1}{2} \|\mathbf{w}^*\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^{*\top} \boldsymbol{\phi}(\mathbf{x}_i) + \inf_b \left\{ -b \sum_{i=1}^n \alpha_i y_i \right\} + \inf_{\boldsymbol{\xi}} \left\{ \sum_{i=1}^n \xi_i (C - \alpha_i - \beta_i) \right\} + \sum_{i=1}^n \alpha_i \\
&= \begin{cases} -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j) + \sum_{i=1}^n \alpha_i, & \text{if } \sum_{i=1}^n \alpha_i y_i = 0 \text{ and } C - \alpha_i - \beta_i = 0 \\ -\infty, & \text{otherwise} \end{cases}
\end{aligned}$$

where in the last equation we used

$$\begin{aligned}
\frac{1}{2} \|\mathbf{w}^*\|^2 - \sum_{i=1}^n \alpha_i y_i \mathbf{w}^{*\top} \boldsymbol{\phi}(\mathbf{x}_i) &= \frac{1}{2} \left\langle \sum_{i=1}^n \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i), \sum_{i=1}^n \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i) \right\rangle - \sum_{i=1}^n \alpha_i y_i \left\langle \sum_{j=1}^n \alpha_j y_j \boldsymbol{\phi}(\mathbf{x}_j), \boldsymbol{\phi}(\mathbf{x}_i) \right\rangle \\
&= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j) - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j) \\
&= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j)
\end{aligned}$$

as well as

$$\inf_b \left\{ -b \sum_{i=1}^n \alpha_i y_i \right\} = \begin{cases} 0, & \text{if } \sum_{i=1}^n \alpha_i y_i = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

and

$$\inf_{\xi} \left\{ \sum_{i=1}^n \xi_i (C - \alpha_i - \beta_i) \right\} = \begin{cases} 0, & \text{if } C - \alpha_i - \beta_i = 0 \text{ for all } i \\ -\infty, & \text{otherwise.} \end{cases}$$

That is, the **dual function** is given as

$$g(\alpha, \beta) = \begin{cases} -\frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) + \sum_{i=1}^n \alpha_i, & \text{if } \sum_{i=1}^n \alpha_i y_i = 0 \text{ and } C - \alpha_i - \beta_i = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

Given an optimization problem in the canonical form, in the corresponding dual we maximize the dual function (here $g(\alpha, \beta)$) subject to the non-negativity constraints on the dual variables (here $\alpha, \beta \succeq \mathbf{0}$). Observe that the dual function has implicit constraints. Since we aim to maximize g over the dual variables – we do not care about the case where $g(\alpha, \beta) = -\infty$. Therefore, we can write the implicit constraints in the definition of g as explicit constraints of the optimization problem. This gives the the following formulation of the dual problem (for the soft-margin SVM):

$$\begin{array}{ll} \underset{\alpha_1, \dots, \alpha_n}{\text{maximize}} & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \\ \text{subject to} & \forall i: 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^n \alpha_i y_i = 0 \end{array}$$

In particular, from $C - \alpha_i - \beta_i = 0$ we get $C - \alpha_i = \beta_i$. That is, β is uniquely determined by the values of α and is effectively eliminated from the above optimization problem. Furthermore, due to $\alpha_i, \beta_i \geq 0$ we get a box constraint $0 \leq \alpha_i \leq C$.

(c) From the previous solution in (b) we know that

$$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* y_i \phi(\mathbf{x}_i).$$

To find b^* we use the KKT condition (complementary slackness) " $\lambda_i \cdot f_i(\mathbf{x}) = 0$ ". Note that the data points \mathbf{x}_i with $\alpha_i = 0$ do not contribute to the decision boundary. All other points with $\alpha_i > 0$ constitute the support vectors. Points with $\alpha_i = C$ lie inside the margin (or even on the wrong side of the decision boundary). Consider a support vector with $0 < \alpha_i < C$. Such support vectors lie exactly on the margin boundary! This follows from the complementary slackness:

$$\begin{aligned} \alpha_i \cdot (1 - \xi_i - y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b)) &= 0 & \xRightarrow{\alpha_i > 0} & b = y_i(1 - \xi_i) - \mathbf{w}^\top \phi(\mathbf{x}_i) \\ \beta_i(-\xi_i) &= 0 & \xRightarrow{\beta_i = C - \alpha_i > 0} & \xi_i = 0, \end{aligned}$$

which together implies

$$0 < \alpha_i < C \implies b = y_i - \mathbf{w}^\top \phi(\mathbf{x}_i) = y_i - \sum_{j=1}^n \alpha_j y_j \phi(\mathbf{x}_j)^\top \phi(\mathbf{x}_i).$$

(d) By replacing $\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$ we get kernelized version:

$$\begin{array}{ll} \underset{\alpha_1, \dots, \alpha_n}{\text{maximize}} & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & \forall i: 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^n \alpha_i y_i = 0. \end{array}$$

The corresponding decision function is given as

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \phi(\mathbf{x}) + b) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Exercise 2: SVMs and Quadratic Programming

Consider first the objective

$$\underset{\alpha_1, \dots, \alpha_n}{\text{minimize}} \quad \underbrace{\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)}_{=\boldsymbol{\alpha}^\top P \boldsymbol{\alpha}} - \underbrace{\sum_{i=1}^n \alpha_i}_{=\mathbf{1}^\top \boldsymbol{\alpha}}$$

where $P_{i,j} = y_i y_j K_{i,j}$ with K being the kernel matrix (reminder: $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$). That is,

$$P = \text{diag}(y_1, \dots, y_n) \cdot K \cdot \text{diag}(y_1, \dots, y_n),$$

where y_1, \dots, y_n are the labels of x_1, \dots, x_n , respectively. Furthermore, $\mathbf{q} = -\mathbf{1} \in \mathbb{R}^n$, where $\mathbf{1} = (1, \dots, 1)$ is a vector of one's. The inequality constraints $0 \leq \alpha_i \leq C$ can be represented as follows:

$$G = \begin{bmatrix} -I \\ I \end{bmatrix}, h = \begin{bmatrix} \mathbf{0} \\ C \cdot \mathbf{1} \end{bmatrix},$$

where $I \in \mathbb{R}^{n \times n}$ denotes the identity matrix, that is, $G \in \mathbb{R}^{2n \times n}$ and $h \in \mathbb{R}^{2n}$. The equality constraint $\sum_{i=1}^n \alpha_i y_i = 0$ can be represented as

$$A = \mathbf{y}^\top, b = 0,$$

where $\mathbf{y} = (y_1, \dots, y_n)$.

```

1 import numpy as np
2 import scipy, scipy.spatial
3 import cvxopt, cvxopt.solvers
4
5 def getGaussianKernel(X1, X2, scale):
6     D = scipy.spatial.distance.cdist(X1,X2,'sqeuclidean')
7     return np.exp(-D/(2*scale**2))
8
9 def getQPMatrices(K, Y, C):
10     # Prepare matrices
11     n = Y.shape[0]
12     P = Y[:,np.newaxis]*K*Y[np.newaxis,:]
13     # # alternatively
14     # #-----
15     # diag = np.diag(Y)
16     # P = np.matmul(diag, np.matmul(K, diag))
17     # #-----
18     q = -np.ones([n])
19     G = np.concatenate([-np.identity(n), np.identity(n)])
20     h = np.concatenate([np.zeros([n]), C * np.ones([n])])
21     A = np.reshape(Y, (1,n))
22     b = np.array([0.0])
23
24     # Convert to CVXOPT matrices
25     P = cvxopt.matrix(P)
26     q = cvxopt.matrix(q)
27     G = cvxopt.matrix(G)
28     h = cvxopt.matrix(h)
29     A = cvxopt.matrix(A)
30     b = cvxopt.matrix(b)
31
32     return P,q,G,h,A,b
33

```

```

33
34 def getTheta(K, Y, alpha, C):
35     # First we need to find a support vector with  $0 < \alpha_i < C$ .
36     # Instead of looking at all possible alpha's in a loop, we use the midpoint heuristic.
37     # Note: the value lying closer to  $C / 2$  is more likely to satisfy this condition.
38     # Considering the absolute difference np.abs ensures that the value \alpha_i does
39     # not lie too close to the boundaries 0 or C improving upon the numerical stability.
40     sv = np.argmin(np.abs(alpha - C / 2.0))
41     theta = Y[sv] - np.dot(K[sv,:], alpha * Y)
42     return theta
43
44 def fit(self, X, Y):
45     K = getGaussianKernel(X, X, self.scale)
46     P, q, G, h, A, b = getQPMatrices(K, Y, self.C)
47
48     alpha = np.array(cvxopt.solvers.qp(P, q, G, h, A, b)['x']).flatten()
49     th = 1e-6 * alpha.mean()
50     ind = alpha > th # determine (robust) support vectors (alternatively set th = 0)
51     self.X, self.Y, self.alpha = X[ind], Y[ind], alpha[ind]
52
53     self.theta = getTheta(K, Y, alpha, self.C)
54
55 def predict(self, X):
56     K = getGaussianKernel(X, self.X, self.scale)
57     Y = np.sign(np.dot(K, self.alpha * self.Y) + self.theta)
58     return Y
59

```