
Machine Learning

Answer Sheet for Homework 5

Da-Min HUANG

R04942045

Graduate Institute of Communication Engineering, National Taiwan University

November 24, 2015

Problem 1

The hard-margin support vector machine is with $d + 1$ variables. For soft-margin support vector machine, there are N more variables ξ_n , $1 \leq n \leq N$.

So soft-margin support vector machine is a quadratic programming problem with $N + d + 1$ variables.

□

Problem 2

I wrote a `Q02.py` to help me get the answer. By using Python package `cvxopt`^[2], with

$$\mathbf{z} = \begin{bmatrix} 1 & -2 \\ 4 & -5 \\ 4 & -1 \\ 5 & -2 \\ 7 & -7 \\ 7 & 1 \\ 7 & 1 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \\ +1 \\ +1 \\ +1 \end{bmatrix} \quad (1)$$

and

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2)$$

$$\mathbf{A}^T = \begin{bmatrix} -1 & -1 & 2 \\ -1 & -4 & 5 \\ -1 & -4 & 1 \\ 1 & 5 & -2 \\ 1 & 7 & -7 \\ 1 & 7 & 1 \\ 1 & 7 & 1 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (3)$$

To use this package, I gave `solvers.qp(Q, p, -AT, -c)` and got

$$b = -9, \mathbf{w} = [2, 0] \quad (4)$$

So the hyperplane is

$$2z_1 - 9 = 0 \Rightarrow z_1 = 4.5 \quad (5)$$

□

Problem 3

I wrote a `Q03.py` to help me get the answer. By using Python package `cvxopt`, with

$$\mathbf{Q} = \begin{bmatrix} 4 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 4 & 0 & -1 & -9 & -1 & -1 \\ 1 & 0 & 4 & -1 & -1 & -9 & -1 \\ 0 & -1 & -1 & 4 & 1 & 1 & 9 \\ -1 & -9 & -1 & 1 & 25 & 9 & 1 \\ -1 & -1 & -9 & 1 & 9 & 25 & 1 \\ -1 & -1 & -1 & 9 & 1 & 1 & 25 \end{bmatrix}, \mathbf{p} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad (6)$$

$$-\mathbf{A}^T = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

with

$$\mathbf{G} = \mathbf{y}^T = \begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ and } h = 0 \quad (8)$$

and To use this package, I gave `solvers.qp(Q, p, -AT, c, G, h)` and got

$$\alpha = [4.32 \times 10^{-9} \approx 0, 0.704, 0.704, 0.889, 0.259, 0.259, 5.27 \times 10^{-10} \approx 0] \quad (9)$$

where `cvxopt` needs conditions

$$-\mathbf{A}^T \boldsymbol{\alpha} \preceq \mathbf{c} \text{ and } \mathbf{G} \boldsymbol{\alpha} = h \quad (10)$$

□

Problem 4

I wrote a `Q04.py` to help me get the answer. By using python package `sympy` and

$$\mathbf{w} = \sum_{n=1}^N \alpha_n y_n K \left(\mathbf{x}_n, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) + b \quad (11)$$

$$b = y_s - \sum_{n=1}^N \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_s) \quad (12)$$

we have

$$\mathbf{w} = \frac{1}{9} (8x_1^2 - 16x_1 + 6x_2^2 - 15) \quad (13)$$

□

Problem 5

Since kernel function $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^2$ is different from $\mathbf{z} = (\phi(\mathbf{x}), \phi(\mathbf{x}'))$, the curves should be different in the \mathcal{X} space.

□

Problem 6

Since $\|\mathbf{x}_n - \mathbf{c}\|^2 \leq R^2, \forall n$, the constraint to maximize is

$$\|\mathbf{x}_n - \mathbf{c}\|^2 - R^2 \leq 0 \quad (14)$$

so $L(R, \mathbf{c}, \boldsymbol{\lambda})$ is

$$L(R, \mathbf{c}, \boldsymbol{\lambda}) = R^2 + \sum_{n=1}^N \lambda_n (\|\mathbf{x}_n - \mathbf{c}\|^2 - R^2) \quad (15)$$

□

Problem 7

At the optimal $(R, \mathbf{c}, \boldsymbol{\lambda})$,

$$\frac{\partial L}{\partial R} = 2R - 2R \sum_{n=1}^N \lambda_n = 0 \Rightarrow \sum_{n=1}^N \lambda_n = 1 \text{ or } R = 0 \quad (16)$$

$$\frac{\partial L}{\partial \mathbf{c}} = 2 \sum_{n=1}^N \lambda_n (\mathbf{c} - \mathbf{x}_n) = \mathbf{0} \Rightarrow \mathbf{c} = \left(\sum_{n=1}^N \lambda_n \mathbf{x}_n \right) / \left(\sum_{n=1}^N \lambda_n \right) \text{ if } \sum_{n=1}^N \lambda_n \neq 0 \quad (17)$$

So the KKT conditions are

1. primal feasible: $\|\mathbf{x}_n - \mathbf{c}\|^2 \leq R^2$.
2. dual feasible: $\lambda_n \geq 0$.
3. dual-inner optimal: if $R \neq 0$, $\sum_{n=1}^N \lambda_n = 1$ and $\mathbf{c} = \sum_{n=1}^N \lambda_n \mathbf{x}_n$.
4. primal-inner optimal: $\lambda_n (\|\mathbf{x}_n - \mathbf{c}\|^2 - R^2) = 0$.

□

Problem 8

From Problem 6, we have

$$L(R, \mathbf{c}, \boldsymbol{\lambda}) = R^2 + \sum_{n=1}^N \lambda_n (\|\mathbf{x}_n - \mathbf{c}\|^2 - R^2) = R^2 + \sum_{n=1}^N \lambda_n \|\mathbf{x}_n - \mathbf{c}\|^2 - \sum_{n=1}^N \lambda_n R^2 \quad (18)$$

$$= R^2 - R^2 + \sum_{n=1}^N \lambda_n \|\mathbf{x}_n - \mathbf{c}\|^2 = \sum_{n=1}^N \lambda_n \|\mathbf{x}_n - \mathbf{c}\|^2 \quad (19)$$

where $\sum_{n=1}^N \lambda_n = 1$ since $R \neq 0$.

Also, from (17), we have $\mathbf{c} = \sum_{n=1}^N \lambda_n \mathbf{x}_n$. Hence

$$\text{Objective}(\boldsymbol{\lambda}) = \sum_{n=1}^N \lambda_n \left\| \mathbf{x}_n - \sum_{m=1}^N \lambda_m \mathbf{x}_m \right\|^2 \quad (20)$$

□

Problem 9

We have

$$\sum_{n=1}^N \lambda_n \|\mathbf{x}_n - \mathbf{c}\|^2 = \sum_{n=1}^N \lambda_n (\mathbf{x}_n^T \mathbf{x}_n - \mathbf{x}_n^T \mathbf{c} - \mathbf{c}^T \mathbf{x}_n + \mathbf{c}^T \mathbf{c}) \quad (21)$$

$$= \sum_{n=1}^N \lambda_n \left(\mathbf{x}_n^T \mathbf{x}_n - \mathbf{x}_n^T \sum_{m=1}^N \lambda_m \mathbf{x}_m - \left(\sum_{m=1}^N \lambda_m \mathbf{x}_m \right)^T \mathbf{x}_n + \left\| \sum_{m=1}^N \lambda_m \mathbf{x}_m \right\|^2 \right) \quad (22)$$

So

$$\sum_{n=1}^N \lambda_n \|\phi(\mathbf{x}_n) - \phi(\mathbf{c})\|^2 \quad (23)$$

$$= \sum_{n=1}^N \lambda_n K(\mathbf{x}_n, \mathbf{x}_n) - 2 \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m K(\mathbf{x}_n, \mathbf{x}_m) + \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m K(\mathbf{x}_n, \mathbf{x}_m) \quad (24)$$

$$= \sum_{n=1}^N \lambda_n K(\mathbf{x}_n, \mathbf{x}_n) - \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m K(\mathbf{x}_n, \mathbf{x}_m) \quad (25)$$

□

Problem 10

From primal-inner optimal condition, pick some $\lambda_i > 0$, we have

$$\|\mathbf{x}_i - \mathbf{c}\|^2 = R^2 \quad (26)$$

so

$$R^2 = \mathbf{x}_i^T \mathbf{x}_i - \mathbf{x}_i^T \sum_{m=1}^N \lambda_m \mathbf{x}_m - \left(\sum_{m=1}^N \lambda_m \mathbf{x}_m \right)^T \mathbf{x}_i + \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m \mathbf{x}_n^T \mathbf{x}_m \quad (27)$$

$$= K(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum_{m=1}^N \lambda_m K(\mathbf{x}_i, \mathbf{x}_m) + \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m K(\mathbf{x}_n, \mathbf{x}_m) \quad (28)$$

$$\Rightarrow R = \sqrt{K(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum_{m=1}^N \lambda_m K(\mathbf{x}_i, \mathbf{x}_m) + \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m K(\mathbf{x}_n, \mathbf{x}_m)} \quad (29)$$

where $R > 0$.

□

Problem 11

☐

Problem 12

☐

Problem 13

☐

Problem 14

☐

Problem 15

☐

Problem 16

☐

Problem 17

☐

Problem 18

☐

Problem 19



Problem 20



Reference

- [1] Lecture Notes by Hsuan-Tien LIN, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan.
- [2] Quadratic Programming with Python and CVXOPT
<https://courses.csail.mit.edu/6.867/wiki/images/a/a7/Qp-cvxopt.pdf>