Machine Learning

Answer Sheet for Homework 5

Da-Min HUANG

R04942045

Graduate Institute of Communication Engineering, National Taiwan University

November 23, 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.

$\square$

Problem 2

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

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

and

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

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

To use this package, I gave `solvers.qp(`$\mathbf{Q}, \mathbf{p}, -\mathbf{A}^T, -\mathbf{c}$`)` and got

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

So the hyperplane is

$$2z_1 - 9 = 0 \Rightarrow z_1 = 4.5 \tag{5}$$

$\square$

Problem 3

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

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

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

with

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

and To use this package, I gave `solvers.qp(`$\mathbf{Q}, \mathbf{p}, -\mathbf{A}^T, \mathbf{c}, \mathbf{G}, h$`)` and got

$$\alpha = \left[4.32 \times 10^{-9} \approx 0, 0.704, 0.704, 0.889, 0.259, 0.259, 5.27 \times 10^{-10} \approx 0\right] \tag{9}$$

where `cvxopt` needs conditions

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

□

Problem 4

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

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

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

we have

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

□

Problem 5

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

□

Problem 6

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

$$\left\|\mathbf{x}_n - \mathbf{c}\right\|^2 - R^2 \leq 0 \tag{14}$$

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

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

□

Problem 7

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

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

$$\frac{\partial L}{\partial \mathbf{c}} = 2 \sum_{n=1}^{N} \lambda_n (\mathbf{c} - \mathbf{x}_n) = \mathbf{0} \tag{17}$$

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 $R \neq 0$, $\sum_{n=1}^{N} \lambda_n = 1$; $\sum_{n=1}^{N} \lambda_n (\mathbf{c} - \mathbf{x}_n) = \mathbf{0}$.
4. primal-inner optimal: $\lambda_n \left(\|\mathbf{x}_n - \mathbf{c}\|^2 - R^2\right) = 0$.

□

Problem 8

□

Problem 9

□

Problem 10

□

Problem 11

□

Problem 12

$\square$

Problem 13

$\square$

Problem 14

$\square$

Problem 15

$\square$

Problem 16

$\square$

Problem 17

$\square$

Problem 18

$\square$

Problem 19

$\square$

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