Sai Satya Charan

EECS 545

HW-4

Q 1)

$$\min_{x \in \mathbb{R}^2} \; (2x_1-1)^2 + (x_2-2)^2$$

Subject to $\quad 3x_1 + 2x_2 \leq 4$

$\quad x_2 \geq x_1$

1 (a) The original problem is convex and a constraint qualification holds $\Rightarrow$ all the inequality constraints are affine.

So, strong duality holds. $\Rightarrow p^* = d^*$

1 (b) The constrained optimization problem is differentiable, convex, and a constraint qualification hold $\Rightarrow$ all inequality constraints are affine, so, KKT conditions are necessary and sufficient for primal/dual optimality (with zero duality gap)

$f(x) = (2x_1-1)^2 + (x_2-2)^2$

$g_1(x) = 3x_1+2x_2-4 \leq 0$

$g_2(x) = x_1 - x_2 \leq 0$

There are No equality constraints.

The Lagrangian is given as.

$$\boxed{L(x, \lambda_1, \lambda_2) = (2x_1-1)^2 + (x_2-2)^2 + \lambda_1(3x_1+2x_2-4) + \lambda_2(x_1-x_2)}$$

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

We know that strong duality holds from part (a), so the primal optimal (x_1^*, x_2^*) and the dual optimal $(\lambda_1^*, \lambda_2^*)$ must satisfy KKT conditions

1) $\nabla_x f(x^*) + \sum_{i=1}^{r} \lambda_i^* \nabla_x g_i(x^*) + \underbrace{\sum_{j=1}^{s} \nu_j^* \nabla_x h_j(x^*)}_{0} = 0$

0 (No equality constraints)

$\nabla_{x_1}[L] = 2(2x_1^*-1)(2) + 3\lambda_1^* + \lambda_2^* = 0$

$\nabla_{x_2}[L] = 2(x_2^*-2) + 2\lambda_1^* - \lambda_2^* = 0$

$$\Rightarrow \left.\begin{aligned} 8x_1^* - 4 + 3\lambda_1^* + \lambda_2^* &= 0 \\ 2x_2^* - 4 + 2\lambda_1^* - \lambda_2^* &= 0 \end{aligned}\right\}$$ These equations constitute the 1st condition

2) $g_i(x^*) \le 0 \quad \forall i$

$\Rightarrow \quad 3x_1^* + 2x_2^* - 4 \le 0$

$x_1^* - x_2^* \le 0$

3) $h_j(x^*) = 0 \quad \forall j$

Not applicable as there are no equality constraints

4) $\lambda_i^* \ge 0$

$\Rightarrow \lambda_1^* \ge 0 \quad \lambda_2^* \ge 0$

5) $\lambda_i^* g_i(x^*) = 0 \quad \forall i$

$\Rightarrow \lambda_1^* (3x_1^* + 2x_2^* - 4) = 0$

$\lambda_2^* (x_1^* - x_2^*) = 0$

These are the 5 KKT Conditions that can be used to find the optimal variable.

1 (c) Due to strong duality proved in part (a) of the problem, it suffices to find $(x_1^*, x_2^*, \lambda_1^*, \lambda_2^*)$ satisfying the KKT conditions outlined in part (b)

From condition 1 of KKT we have

$$x_1^* = \frac{-(3\lambda_1^* + \lambda_2^*) + 4}{8} \Rightarrow \frac{-(3\lambda_1^* + \lambda_2^*)}{8} + \frac{1}{2} \quad -(1)$$

$$x_2^* = \frac{\lambda_2^* - 2\lambda_1^* + 4}{2} \Rightarrow \frac{(\lambda_2^* - 2\lambda_1^*)}{2} + 2 \quad -(2)$$

We can substitute eqn's (1) & (2) back into condition 5 of KKT

$$\Rightarrow \lambda_1^* (3x_1^* + 2x_2^* - 4) = 0$$

$$\lambda_2^* (x_1^* - x_2^*) = 0$$

$$\lambda_1^* \left(3\left[\frac{-(3\lambda_1^* + \lambda_2^*)}{8} + \frac{1}{2}\right] + 2\left[\frac{\lambda_2^* - 2\lambda_1^*}{2} + 2\right] - 4 \right) = 0 \quad -(3)$$

$$\lambda_2^* \left(\left[\frac{-(3\lambda_1^* + \lambda_2^*)}{8} + \frac{1}{2}\right] - \left[\frac{(\lambda_2^* - 2\lambda_1^*)}{2} + 2\right] \right) = 0 \quad -(4)$$

Simplyfying (3)

$$\rightarrow \lambda_1^* \left(\frac{-9\lambda_1^* - 3\lambda_2^* + 12}{8} + \frac{2\lambda_2^* - 4\lambda_1^* + 8}{2} - 4 \right) = 0$$

$$\Rightarrow \lambda_1^* \left(-9\lambda_1^* - 3\lambda_2^* + 12 + 8\lambda_2^* - 16\lambda_1^* + \cancel{32} - \cancel{32} \right) = 0$$

$$\lambda_1^* (5\lambda_2^* - 25\lambda_1^* + 12) = 0 \quad -(5)$$

Simplyfying (4)

$$\lambda_2^* \left(\frac{-3\lambda_1^* - \lambda_2^* + 4}{8} - \frac{(\lambda_2^* - 2\lambda_1^* + 4)}{2} \right)$$

$$\Rightarrow \lambda_2^* \left(-3\lambda_1^* - \lambda_2^* + 4 - 4\lambda_2^* + 8\lambda_1^* - 16 \right) = 0$$

$$\Rightarrow \lambda_2^* \left(5\lambda_1^* - 5\lambda_2^* - 12 \right) = 0 \quad - (6)$$

Combining (5) & (6) we have

$$\left.\begin{aligned} \lambda_1^* \left(5\lambda_2^* - 25\lambda_1^* + 12 \right) &= 0 \\ \lambda_2^* \left(5\lambda_1^* - 5\lambda_2^* - 12 \right) &= 0 \end{aligned}\right\} \boxed{*}$$

Case 1 $\quad \lambda_1^* = 0$

Case 1.1 $\quad \lambda_2^* = 0 \quad \rightarrow$ Condition ~~Constraint~~ 4 Satisfied

Then we have $\quad x_1^* = \frac{4}{8} = \frac{1}{2}$

$$x_2^* = \frac{4}{2} = 2$$

Check if Condition (2) is Satisfied

$$\Rightarrow 3x_1^* + 2x_2^* - 4 \leq 0 \qquad\qquad x_1^* - x_2^* \leq 0$$

$$\Rightarrow 3(1/2) + 2(2) - 4 \qquad\qquad \frac{1}{2} - 2 \Rightarrow -3/2 \leq 0 \checkmark$$

$$\Rightarrow 3/2 > 0 \quad \times$$

— Condition 2 is violated

This Can't be the solution.

Case 1.2 $\cancel{\lambda}(5\lambda_1^* - 5\lambda_2^* - 12) = 0$

$\lambda_1^* = 0$ } assumption in all of Case-1's

So, $(-5\lambda_2^* - 12) = 0$

$\Rightarrow \lambda_2 = \frac{12}{5}$ ✗ → Condition 4 fails/ violated

So, No solution possible in this case 1.2

Case 2 $\cancel{\lambda_1^*}(5\lambda_2^* - 25\lambda_1^* + 12) = 0$

Case 2.1 $\lambda_2^* = 0$ ~~→ Condition 4 satisfied~~

$\Rightarrow -25\lambda_1^* + 12 = 0$

$\lambda_1^* = \frac{12}{25}$ $\lambda_1^* > 0$ $\lambda_2^* = 0$

→ Condition 4 satisfied

Check for condition (2)

$x_1^* = \frac{-3\lambda_1^* + 4}{8} \Rightarrow \frac{-3(12/25) + 4}{8} \Rightarrow \frac{-36/25 + 4}{8}$

$\Rightarrow 64/8 \times 25$

$x_1^* = 8/25$

$x_2^* = \frac{-2\lambda_1^* + 4}{2} \Rightarrow \frac{-2\left(\frac{12}{25}\right) + 4}{2} \Rightarrow \frac{-\frac{24}{25} + 4}{2}$

$x_2^* = 38/25$

Condition (2) requires:

$3x_1^* + 2x_2^* - 4 \leq 0$	$x_1^* - x_2^* \leq 0$
$\frac{24}{25} + \frac{76}{25} - 4 \leq 0$	$8/25 - 38/25$
$\Rightarrow 0$ ✓	$\Rightarrow -30/25 < 0$ ✓

→ Condition 2 satisfied

For Case 2.1 all the 5 conditions are satisfied and the obtained $\lambda_1^*, \lambda_2^*, x_1^*, x_2^*$ are as follows.

$$\boxed{\begin{gathered} \lambda_1^* = \frac{12}{25} \qquad \lambda_2^* = 0 \\ x_1^* = 8/25 \qquad x_2^* = 38/25 \\ \text{optimal value} \Rightarrow (2x_1^* - 1)^2 + (x_2^* - 2)^2 \\ = \left(16/25 - 1\right)^2 + \left(\frac{38}{25} - 2\right) \\ = 9/25 \end{gathered}}$$

$\rightarrow$ Ans

I am doing Case 2.2 just to check

Case 2.2 $5\lambda_1^* - 5\lambda_2^* - 12 = 0$

$\Rightarrow$ Case 2 assumption is $(5\lambda_2^* - 25\lambda_1^* + 12) = 0$

$$\lambda_1^* = \left[\frac{5\lambda_2^* + 12}{5}\right] \Rightarrow 5\lambda_2^* - \cancel{25}^{5}\left[\frac{5\lambda_2^* + 12}{\cancel{5}}\right] + 12$$

$$\Rightarrow 5\lambda_2^* - 25\lambda_2^* - 60 + 12 = 0$$

$$\Rightarrow -20\lambda_2^* - 48 = 0$$

$$\lambda_2^* = -\frac{\cancel{48}\ 12}{\cancel{20}\ 5}$$

$$\lambda_2^* = -12/5 < 0$$

$\rightarrow$ Condition 2 violated

$\Rightarrow$ This case can't have the solution to the problem.

~~Min~~

Q1) (d)

Lagrangian Dual function

$$L_D(\lambda_1, \lambda_2) = \min_{x_1, x_2} L(x_1, x_2, \lambda_1, \lambda_2)$$

$$\boxed{L_D(\lambda_1, \lambda_2) = \min_{x_1, x_2} (2x_1 - 1)^2 + (x_2 - 2)^2 + \lambda_1(3x_1 + 2x_2 - 4) + \lambda_2(x_1 - x_2)} \quad \#$$

The ~~dual is~~ minimization to find the dual function can be done using $\nabla_{x_1}, \nabla_{x_2}[L(x_1, x_2, \lambda_1, \lambda_2)]$ and setting them to 0.

$$\nabla_{x_1}[L(x_1, x_2, \lambda_1, \lambda_2)] \Rightarrow 4(2x_1 - 1) + 3\lambda_1 + \lambda_2 = 0$$

$$\nabla_{x_2}[L(x_1, x_2, \lambda_1, \lambda_2)] = 2(x_2 - 2) + 2\lambda_1 - \lambda_2 = 0$$

Solving for x_1, x_2 gives.

$$x_1 = \frac{-3\lambda_1 - \lambda_2 + 4}{8} \quad \Big| \quad x_2 = \frac{-2\lambda_1 + \lambda_2 + 4}{2} \quad \text{— } (0)$$

back substitute x_1, x_2 in $\#$

I will build the $L_D(\lambda_1, \lambda_2)$ term by term.

$$(2x_1 - 1)^2 \Rightarrow \left(\cancel{2}\frac{-3\lambda_1 - \lambda_2 + 4}{\cancel{8}4} - 1\right)^2 \Rightarrow \cancel{\left(\frac{-3\lambda_1 - \lambda_2 - 4}{84}\right)^2}$$

$$\Rightarrow \cancel{(3\lambda_1 + \lambda_2 + 4)^2 / 16}$$

$$(2x_1 - 1)^2 \Rightarrow \frac{(3\lambda_1 + \lambda_2)^2}{16} \Rightarrow \frac{9\lambda_1^2}{16} + \frac{3\lambda_1\lambda_2}{8} + \frac{\lambda_2^2}{16} \quad \text{— } (1)$$

$(x_2-2)^2 = \frac{\lambda_1^2 - \lambda_1\lambda_2 + \lambda_2^2}{4}$ — (2)

$\lambda_1(3x_1+2x_2-4) = \frac{-25}{8}\lambda_1^2 + \frac{5\lambda_1\lambda_2}{8} + \frac{3\lambda_1}{2}$ (3)

$\lambda_2(x_1-x_2) = \frac{5\lambda_1\lambda_2}{8} - \frac{5\lambda_2^2}{8} - \frac{3\lambda_2}{2}$ — (4)

Summing up all 4 terms (1)+(2)+(3)+(4)

we get,

$$\boxed{L_D(\lambda_1,\lambda_2) = -\frac{25}{16}\lambda_1^2 - \frac{5}{16}\lambda_2^2 + \frac{5}{8}\lambda_1\lambda_2 + \frac{3\lambda_1}{2} - \frac{3\lambda_2}{2}}$$

dual optimization problem

$$\max_{\lambda_1,\lambda_2} L_D(\lambda_1,\lambda_2) \quad \lambda_1 \geq 0, \quad \lambda_2 \geq 0 \implies \min -L_D(\lambda_1,\lambda_2) \quad -\lambda_1 \leq 0, \quad -\lambda_2 \leq 0$$

This is again a constrained minization problem,

with objective $-L_D(\lambda_1,\lambda_2)$ is convex and differentiable, and also

Constraint qualification holds $\Rightarrow$ all inequalities are affine

So KKT conditions are necessary and sufficient

Let us build Lagrangian, (treat λ as normal primal variable, $k_1, k_2 \to$ dual variable) For new problem

$$L(\lambda_1,\lambda_2,k_1,k_2) = -L_D(\lambda_1,\lambda_2) + k_1(-\lambda_1) + k_2(\lambda_2)$$

The optimal $\lambda_1^*, \lambda_2^*, k_1^*, k_2^*$ should satisfy the KKT conditions.

1) $\nabla f(x^*) + \sum_{i=1}^{r} k_i^* \nabla_x g_i(x_i^*) = 0$

$$\nabla_{L_1}[L] = +\frac{25}{8}\lambda_1^* - 5/8\,\lambda_2^* - 3/2 - k_1^* = 0$$

$$\nabla_{L_2}[L] = 5/8\,\lambda_2^* - 5/8\,\lambda_1^* + 3/2 - k_2^* = 0$$

2) $g_i(x^*) \leq 0 \;\forall i$

$$-\lambda_1^* \leq 0 \qquad -\lambda_2^* \leq 0$$

3) Not applicable

4) $k_1^* \geq 0 \qquad k_2^* \geq 0$

5) $\left.\begin{array}{l} k_1^*(-\lambda_1^*) = 0 \\ k_2(-\lambda_2^*) = 0 \end{array}\right\}$ Complimentary Slackness.

These are the 5 KKT conditions that should be satisfied

Case:-1 $k_1^* = 0$

Case 1.1 $k_2^* = 0$ → condition 4 satisfied

Check for condition (2) by solving condition (1)

$$\left.\begin{aligned} &\frac{25}{8}\lambda_1^* - \frac{5}{8}\lambda_2^* - \frac{3}{2} = 0 \quad \& \\ &\frac{5}{8}\lambda_2^* - \frac{5}{8}\lambda_1^* + \frac{3}{2} = 0 \end{aligned}\right\} \quad \begin{aligned} &\lambda_1^* = 0 \\ &\lambda_2^* = -12/5 \end{aligned}$$

→ Condition (2) violated

Solution doesn't exist in this case

Case 1.2 $\lambda_2^* = 0$

Solving eqn's in condition (1)

$$\left.\begin{aligned} &\frac{25}{8}\lambda_1^* - \frac{3}{2} = 0 \\ &-\frac{5}{8}\lambda_1^* + \frac{3}{2} - k_2^* = 0 \end{aligned}\right\}$$

$\lambda_1^* = 12/25$ → condition (2) satisfied

$k_2^* = -\frac{\cancel{5}}{\cancel{8}_2} \times \frac{\cancel{12}^3}{\cancel{25}_5} + 3/2$

$= -3/10 + 3/2$

$k_2^* = 12/10 = 6/5 > 0$

→ condition (4) satisfied.

$$\boxed{\begin{array}{l} \text{This case satisfies all conditions} \\ \lambda_2^* = 0 \quad \lambda_1^* = 12/25 \\ k_1^* = 0 \quad k_2^* = 6/5 \end{array}}$$

→ Answer

But I will prove the other cases also.

Case 2 $\lambda_1^* = 0$

Case 2.1 $k_2^* = 0$

Case 2.1 Continued.

Solving eqn's in condition ①

$$\left.\begin{aligned} -5/8\,\lambda_2^* - 3/2 - k_1^* &= 0 \\ 5/8\,\lambda_2^* + 3/2 &= 0 \end{aligned}\right\} \quad \lambda_2^* = -\frac{12}{5} \ \} \rightarrow \text{condition (2) is violated}$$

This case doesnot contain solution.

Case 2.2 $\lambda_2^* = 0$

Solving eqns in condition (1)

$$\left.\begin{aligned} -3/2 - k_1^* &= 0 \\ 3/2 - k_2^* &= 0 \end{aligned}\right\} \quad \begin{aligned} k_1^* &= -3/2 \\ k_2^* &= 3/2 \end{aligned} \rightarrow \} \text{ condition (4) is violated}$$

This case does not contain the solution.

The answer obtained by solving dual problem is

$$\lambda_1^* = 12/25 \qquad \lambda_2^* = 0$$

Back substituting in eqn ⓪ to get back x_1^*, x_2^*

$$\left.\begin{aligned} x_1^* &= \frac{-3\lambda_1^* - \lambda_2^* - 4}{8} \rightarrow 8/25 \\ x_2^* &\rightarrow \frac{-2\lambda_1^* + \lambda_2 + 4}{2} \rightarrow 38/25 \end{aligned}\right\}$$

Hence the primal solution is inferred

Q2) SVM without offset

a(a)

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C/n \sum_{i=1}^{n} \xi_i$$

(s.t) $y_i(w^T x_i) \geq 1-\xi_i$ for $i=1,2,3\ldots n$.

$\xi_i \geq 0$ for $i=1,2,\ldots n$.

$\alpha = (\alpha_1,\ldots \alpha_n)$ $\beta = (\beta_1 \ldots \beta_n)$

$$L(w,\xi,\alpha,\beta) = \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i) - 1+\xi_i) - \sum_{i=1}^{n}\beta_i\xi_i \quad -(\#)$$

The dual problem is

$$\max_{\alpha\geq 0,\beta\geq 0} L_D(\alpha,\beta)$$

where

$$L_D(\alpha,\beta) = \min_{w,\xi} L(w,\xi,\alpha,\beta)$$

This problem is an unconstrained minimization problem with a convex differentiable objective function. Therefore, for fixed α,β, values of w,ξ achieving the maximum satisfy

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum \alpha_i y_i x_i = 0 \quad -(1)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \frac{C}{n} - \alpha_i - \beta_i = 0 \quad \forall i \quad -(2)$$

Therefore if $\alpha_i+\beta_i = C/n$ $\forall i$ then pulging in (1) into (#) gives us

$$L_D(\alpha,\beta) = \frac{1}{2}\|\sum\alpha_i y_i x_i\|^2 + C/n\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\sum\alpha_i y_i x_i\right)^T x_i - 1+\xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i$$

$$L_D(\alpha,\beta) = \frac{1}{2}\|\sum \alpha_i y_i x_i\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i - \sum \alpha_i y_i \left(\sum_j \alpha_j y_j x_j\right)^T x_i + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i\xi_i - \sum \beta_i\xi_i$$

$$L_D(\alpha,\beta) = \frac{1}{2}\|\sum \alpha_i y_i x_i\|^2 - \sum \alpha_i y_i \langle \sum_j \alpha_j y_j x_j, x_i\rangle + \sum \alpha_i + \sum_{i=1}^{n}\xi_i\left[\frac{C}{n} - \alpha_i - \beta_i\right]$$

0

$= 0$ from (2)

$$\therefore L_D(\alpha,\beta) = \frac{1}{2}\|\sum_i \alpha_i y_i x_i\|^2 - \sum_i \alpha_i y_i \langle \sum \alpha_i y_i x_i, x_i\rangle + \sum_i \alpha_i$$

$$= \frac{1}{2}\langle \sum_i \alpha_i y_i x_i, \sum_j \alpha_j y_j x_j\rangle - \langle \sum_i \alpha_i y_i x_i, \sum_j \alpha_j y_j x_j\rangle + \sum_i \alpha_i$$

$$= -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i$$

So the quadratic dual function

$$L_D(\alpha,\beta) = \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i & \text{if } \alpha_i+\beta_i = \frac{C}{n}\ \forall i \\ -\infty & \text{otherwise} \end{cases}$$

this is because ξ_i can be send to $\xi_i = \pm\infty$

~~2(b)~~ Therefore the dual optimization problem,

$$\max_{\alpha\geq 0\ \beta\geq 0} L_D(\alpha,\beta)$$

may be written

$$\max_{\alpha,\beta} -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle + \sum \alpha_i$$

* since this is an inner product we can kernelize this

$$\alpha_i + \beta_i = C/n\ \forall i$$

$$\alpha_i \geq 0 \quad \beta_i \geq 0 \quad \forall i$$

The problem can be simplified by eliminating β_i

Since $\alpha_i + \beta_i = C/n$

$\alpha_i \geq 0 \quad \beta_i \geq 0$

we can write it as $0 \leq \alpha_i \leq C/n \quad \forall i$

This leads to an alternate form of the optimal- soft-margin dual

$$\boxed{\begin{aligned} &\max_{\alpha} \; -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle + \sum_{i=1}^{n} \alpha_i \\ &\text{s.t.} \quad 0 \leq \alpha_i \leq C/n \quad \forall i \end{aligned}}$$

we only have this as b is omitted

~~$\langle x_i, x_j\rangle$ is an inner product and hence can be kernelized~~

~~$\langle x_i, x_j\rangle = k(x_i, x_j)$ where k is an innerproduct kernel.~~

~~$\therefore$ The SVM classifier is~~

~~$f(x) = \text{sign}$~~

2(b)

Recovery of the primal solution

(α^*, β^*) be dual optimal, and let (w^*, ξ^*) be primal optimal. Since the primal is convex and satisfies constraint qualification, strong duality holds The KKT necessity theorem then implies that $(w^*, \xi^*, \alpha^*, \beta^*)$ satisfies the KKT conditions.

So to obtain w^*, we use the 1st KKT condition derived in part (a) of this problem.

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

$\downarrow$

This information is sufficient to find the Classifier

The final classifier is therefore

$$f(x) = \text{sign}(\langle w^*, x \rangle)$$

$$= \text{sign}\left(\left\langle \sum_i \alpha_i^* y_i x_i, x \right\rangle\right)$$

$$= \text{sign}\left(\sum_i \alpha_i y_i \langle x_i, x \rangle\right)$$

$\uparrow$

since $\langle x_i, x \rangle$ is an inner product

we can use an inner product kernal to kernalize this method

$$\langle x_i, x \rangle = k(x_i, x)$$

$\therefore$ The final classifier is given as.

$$\boxed{f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i^* y_i \, k(x_i, x)\right)}$$

Q3) $$\min_{w,b,\xi^+,\xi^-} \frac{1}{2}\|w\|_2^2 + C/n \sum_{i=1}^{n} (\xi_i^+ + \xi_i^-)$$

s.t $\quad y_i - w^T x_i - b \leq \varepsilon + \xi_i^+ \quad \forall i \quad$ - ①

$\quad w^T x_i + b - y_i \leq \varepsilon + \xi_i^- \quad \forall i \quad$ - ②

$\quad \xi_i^+ \geq 0 \quad \forall i \quad$ - ③

$\quad \xi_i^- \geq 0 \quad \forall i \quad$ - ④

where $w \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\xi^+ = (\xi_1^+ \cdots \xi_n^+)^T$ and $\xi^- = (\xi_1^- \cdots \xi_n^-)^T$

$C > 0 \quad \varepsilon > 0$

From conditions ① & ③, we can write

$$\left.\begin{array}{l} \xi_i^+ \geq y_i - w^T x_i - b - \varepsilon \\ \text{and} \quad \xi_i^+ \geq 0 \end{array}\right\} \forall i$$

We can combine both them and write as

$$\xi_i^+ \geq \max\{0, y_i - w^T x_i - b - \varepsilon\}$$

Similarly from constraints ② & ④ we can write

$$\xi_i^- \geq \max\{0, w^T x_i + b - y_i - \varepsilon\}$$

So the problem now reduces to

$$\left.\begin{array}{l} \min_{w,b,\xi^+,\xi^-} \frac{1}{2}\|w\|_2^2 + \frac{C}{n}\sum (\xi_i^+ + \xi_i^-) \\ \xi_i^+ \geq \max\{0, y_i - (w^T x_i + b) - \varepsilon\} \\ \xi_i^- \geq \max\{0, (w^T x_i + b) - y_i - \varepsilon\} \end{array}\right\} \rightarrow \#$$

If we examine the objective function, we have a $\left[\xi_i^+ + \xi_i^-\right]$ term.

We know that in a minimization case both ξ_i^+ and ξ_i^- should have equality as we can always reduce the values more to satisfy the condition given below with equality

$\therefore$

@ optimal value $\rightarrow$

$$\left.\begin{aligned} \xi_i^+ &= \max\{0,\ y_i - w^T x_i - b - \varepsilon\} \\ \xi_i^- &= \max\{0,\ (w^T x_i + b) - y_i - \varepsilon\} \end{aligned}\right\}$$

These both should hold @ optimal values

Now we have $\xi_i^+ + \xi_i^-$

Let us analyse this in cases:

Case 1) $y_i - w^T x_i - b - \varepsilon \geq 0$

$\Rightarrow \xi_i^+ = y_i - (w^T x_i + b) - \varepsilon.$

$\Rightarrow w^T x_i + b - y_i \leq -\varepsilon$

$(w^T x_i + b) - y_i - \varepsilon \leq -2\varepsilon \qquad \varepsilon > 0$

$\Rightarrow$ (-ve)

$\Rightarrow \xi_i^- = 0$

$\Rightarrow \xi_i^+ + \xi_i^- \Rightarrow y_i - (w^T x_i + b) - \varepsilon$

This can be written as $\xi_i^+ + \xi_i^- = \max\{0, (y_i - (w^T x_i + b)) - \varepsilon\}$

0 occurs when $y_i - (w^T x_i + b) - \varepsilon \geq 0$

Case 2) $w^T x_i + b - y_i - \varepsilon \geq 0$

a. $\xi_i^+ = 0$

$\xi_i^- = (w^T x_i + b) - y_i - \varepsilon$

So $\xi_i^+ + \xi_i^- = \max\{0, ((w^T x_i + b) - y_i - \varepsilon\}.$

Combining Case -1 and Case -2 one can write

$$\xi_i^+ + \xi_i^- = \max\{0, |y_i - (w^T x_i + b)| - \varepsilon\}$$

$\therefore$ The minimization problem (#) can be reduced by eliminating ξ_i^+, ξ_i^- completely

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\left(\max\{0, |y_i - (w^T x_i + b)| - \varepsilon\}\right)$$

taking C common from the above

we get

$$\min_{w,b} \frac{1}{n}\sum_{i=1}^{n}\underbrace{\left[\max\{0, |y_i - (w^T x_i + b)| - \varepsilon\}\right]}_{l_\varepsilon(y_i,\, w^T x_i + b)} + \underbrace{\frac{1}{2C}}_{\frac{\lambda}{2}}\|w\|_2^2$$

where

$$l_\varepsilon(y, t) = \max\{0, |y - t| - \varepsilon\}$$

and $\frac{\lambda}{2} = \frac{1}{2C}$ (*)

So the SVR solves.

$$\min_{w,b} \frac{1}{n}\sum_{i=1}^{n} l_\varepsilon(y_i, w^T x_i + b) + \frac{\lambda}{2}\|w\|_2^2$$

Q4)

(a) RMSE on testdata SVR: 36.152137

(b) RMSE on test data for KRR: 37.8432

(c) SVR - optimal parameters.

regularization term = 10.

kernel parameter = 0.01

RMSE on test data: 31.264725

KRR - optimal Parameters.

regularization term = 10

kernel parameter = 0.01

RMSE on test data = 33.36136

Problem 4

In [1]:
```python
import numpy as np

from matplotlib import pyplot
import matplotlib.pyplot as plt

# You have have to install the libraries below.
# sklearn, csv
import csv

from sklearn.metrics import mean_squared_error
from sklearn.svm import SVR
from sklearn.kernel_ridge import KernelRidge
```

In [2]:
```python
# The csv file air-quality-train.csv contains the training data.
# After loaded, each row of X_train will correspond to CO, NO2, O3, SO2.
# The vector y_train will contain the PM2.5 concentrations.
# Each row of X_train corresponds to the same timestamp.
X_train = []
y_train = []

with open('air-quality-train.csv', 'r') as air_quality_train:
    air_quality_train_reader = csv.reader(air_quality_train)
    next(air_quality_train_reader)
    for row in air_quality_train_reader:
        row = [float(string) for string in row]
        row[0] = int(row[0])

        X_train.append([row[1], row[2], row[3], row[4]])
        y_train.append(row[5])

# The csv file air-quality-test.csv contains the testing data.
# After loaded, each row of X_test will correspond to CO, NO2, O3, SO2.
# The vector y_test will contain the PM2.5 concentrations.
# Each row of X_train corresponds to the same timestamp.
X_test = []
y_test = []

with open('air-quality-test.csv', 'r') as air_quality_test:
    air_quality_test_reader = csv.reader(air_quality_test)
    next(air_quality_test_reader)
    for row in air_quality_test_reader:
        row = [float(string) for string in row]
        row[0] = int(row[0])

        X_test.append([row[1], row[2], row[3], row[4]])
        y_test.append(row[5])

X_train = np.array(X_train)
y_train = np.array(y_train)
X_test = np.array(X_test)
y_test = np.array(y_test)
```

In [3]:
```python
# TODOs for part (a)
#    1. Use SVR loaded to train a SVR model with rbf kernel, regularizer (C) set to 1 and rbg kernel parameter (gamma) 0.1
#    2. Print the RMSE on the test dataset
svr_rbf = SVR(kernel = 'rbf', C = 1, gamma = 0.1)
y_pred_svr = svr_rbf.fit(X_train,y_train).predict(X_test)
rmse_svr = np.sqrt(mean_squared_error(y_test,y_pred_svr))
print('RMSE obtained for SVR: {}'.format(rmse_svr))

# TODOs for part (b)
#    1. Use KernelRidge to train a Kernel Ridge  model with rbf kernel, regularizer (C) set to 1 and rbg kernel parameter (ga
#    2. Print the RMSE on the test dataset
krr_rbf = KernelRidge(kernel = 'rbf', alpha = 0.5, gamma = 0.1)
y_pred_krr = krr_rbf.fit(X_train,y_train).predict(X_test)
rmse_krr = np.sqrt(mean_squared_error(y_test,y_pred_krr))
print('RMSE obtained for KRR: {}'.format(rmse_krr))


# Use this seed.
seed = 0
np.random.seed(seed)

K = 5 #The number of folds we will create

# TODOs for part (c)
#   1. Create a partition of training data into K=5 folds
#   Hint: it suffice to create 5 subarrays of indices
idx = np.arange(X_train.shape[0])
```

```python
np.random.shuffle(idx)
idx_split = np.array_split(idx,K)

# Specify the grid search space
reg_range = np.logspace(-1,1,3)     # Regularization paramters
kpara_range = np.logspace(-2, 0, 3) # Kernel parameters

# TODOs for part (d)
#    1.  Select the best parameters for both SVR and KernelRidge based on k-fold cross-validation error estimate (use RMSE a:
#    2.  Print the best paramters for both SVR and KernelRidge selected
#    3.  Train both SVR and KernelRidge on the full training data with selected best parameters
#    4.  Print both the RMSE on the test dataset of SVR and KernelRidge

best_rmse_svr = float('inf')
best_rmse_krr = float('inf')

for i in range(len(reg_range)):
    reg = reg_range[i]
    for j in range(len(kpara_range)):
        kpara = kpara_range[j]
        sum_rmse_svr = 0
        sum_rmse_krr = 0
        for k in range(K):
            train_idx = np.concatenate(idx_split[:k]+idx_split[(k+1):])
            validation_idx = np.array(idx_split[k])

            # svr
            svr_rbf = SVR(kernel = 'rbf', C = reg, gamma = kpara)
            y_pred_svr = svr_rbf.fit(X_train[train_idx[:],:] , y_train[train_idx[:]]).predict(X_train[validation_idx[:],:])
            sum_rmse_svr += np.sqrt(mean_squared_error(y_train[validation_idx[:]],y_pred_svr))

            # krr
            krr_rbf = KernelRidge(kernel = 'rbf', alpha = 1/(2*reg), gamma = kpara)
            y_pred_krr = krr_rbf.fit(X_train[train_idx[:],:] , y_train[train_idx[:]]).predict(X_train[validation_idx[:],:])
            sum_rmse_krr += np.sqrt(mean_squared_error(y_train[validation_idx[:]],y_pred_krr))

        if (sum_rmse_svr < best_rmse_svr):
            best_rmse_svr = sum_rmse_svr
            best_reg_svr = reg
            best_kpara_svr = kpara

        if (sum_rmse_krr < best_rmse_krr):
            best_rmse_krr = sum_rmse_krr
            best_reg_krr = reg
            best_kpara_krr = kpara

# With Optimal Parameters
svr_rbf = SVR(kernel = 'rbf', C = best_reg_svr, gamma = best_kpara_svr)
y_pred_svr = svr_rbf.fit(X_train,y_train).predict(X_test)
optimal_rmse_svr = np.sqrt(mean_squared_error(y_test,y_pred_svr))

krr_rbf = KernelRidge(kernel = 'rbf', alpha = 1/(2*best_reg_krr), gamma = best_kpara_svr)
y_pred_krr = krr_rbf.fit(X_train,y_train).predict(X_test)
optimal_rmse_krr = np.sqrt(mean_squared_error(y_test,y_pred_krr))

print('\nOptimal SVR\nregualarization term: {} \nkernel parameter: {}'.format(best_reg_svr,best_kpara_svr))
print('RMSE on test data: {}'.format(optimal_rmse_svr))

print('\nOptimal KRR\nregualarization term: {} \nkernel parameter: {}'.format(best_reg_krr,best_kpara_krr))
print('RMSE on test data: {}'.format(optimal_rmse_krr))
```

```
RMSE obtained for SVR: 36.152137059139214
RMSE obtained for KRR: 37.84320015147227

Optimal SVR
regualarization term: 10.0
kernel parameter: 0.01
RMSE on test data: 31.26472587569616

Optimal KRR
regualarization term: 10.0
kernel parameter: 0.01
RMSE on test data: 33.36136004426222
```