

HOMEWORK-3

PROBLEM 5.5

a) According to linear regression model

$$d_i = w^T \phi_i + \epsilon_i \quad i=1, 2, \dots, N \quad - (1)$$

$$\text{Let } \phi = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_N^T \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}.$$

We can write (1) as

$$d = \phi w + \epsilon \quad - (2)$$

We obtain the least square estimate by minimizing the loss function as below

$$L(w) = \sum_{i=1}^N \epsilon_i^2 = \sum_{i=1}^N (d_i - w^T \phi_i)^2$$

$$= \|d - \phi w\|^2$$

$$= (d - \phi w)^T (d - \phi w)$$

$$= d d^T + w^T \phi^T \phi w - 2 w^T \phi^T d \quad - (3)$$

Taking gradient of $L(w)$ w.r.t w and equating to zero, we get

$$2 \phi^T \phi w - 2 \phi^T d = 0$$

$$\hat{w} = (\phi^T \phi)^{-1} \phi^T d$$

$$= (\phi^T \phi)^{-1} \phi^T (\phi w + \epsilon)$$

$$= w + (\phi^T \phi)^{-1} \phi^T \epsilon.$$

- (4)

Eq (4) is the required result.

b) Using (4), we know

$$\hat{\omega} = \omega + (\phi^T \phi)^{-1} \phi^T \varepsilon$$

$$\begin{aligned} \mathbb{E}[(\omega - \hat{\omega})(\omega - \hat{\omega})^T] &= \mathbb{E}\left[\left((\phi^T \phi)^{-1} \phi^T \varepsilon\right)\left((\phi^T \phi)^{-1} \phi^T \varepsilon\right)^T\right] \\ &= \mathbb{E}\left[(\phi^T \phi)^{-1} \phi^T \varepsilon \varepsilon^T \phi (\phi^T \phi)^{-1}\right] \quad - (5) \end{aligned}$$

$\mathbb{E}[\varepsilon \varepsilon^T]$ is the covariance of the white noise process with mean zero and component variance σ^2 .

$$\mathbb{E}[\varepsilon \varepsilon^T] = \sigma^2 \mathbf{I} \quad - (6)$$

Using (6) in (5), after simplifying

$$\mathbb{E}[(\omega - \hat{\omega})(\omega - \hat{\omega})^T] = \sigma^2 (\phi^T \phi)^{-1} \quad - (7)$$

$$\phi^T \phi = \sum_{j=1}^N \phi_j \phi_j^T = R \quad - (8)$$

Using (8) in (7),

$$\begin{aligned} \mathbb{E}[(\omega - \hat{\omega})(\omega - \hat{\omega})^T] &= \sigma^2 (R)^{-1} \\ &= \sigma^2 R^{-1} \\ &= \sigma^2 P. \end{aligned}$$

PROBLEM 5.6:

$$a) \quad \mathcal{E}_{\text{aw}}(\underline{w}) = \frac{1}{2} \sum_{i=1}^N \left(d(i) - \underline{w}^T \phi(i) \right)^2 + \frac{1}{2} \lambda \|\underline{w}\|^2 \quad - (9)$$

$$\text{Let } \underline{d} = \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}, \quad \underline{e} = \underline{d} - \phi \underline{w} \quad \phi = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi(N)^T \end{bmatrix}$$

$$\begin{aligned} E(\underline{w}) &= \frac{1}{2} \sum_{i=1}^N e_i^2 + \frac{1}{2} \lambda \|\underline{w}\|^2 \\ &= \frac{1}{2} \underline{e}^T \underline{e} + \frac{1}{2} \lambda \|\underline{w}\|^2 \\ &= \frac{1}{2} \left[(\underline{d} - \phi \underline{w})^T (\underline{d} - \phi \underline{w}) + \frac{1}{2} \lambda \|\underline{w}\|^2 \right] \end{aligned}$$

Equating gradient to zero, we get

$$(\phi^T \phi + \lambda \mathbf{I}) \hat{\underline{w}} = \phi^T \underline{d} \quad - (10)$$

$$\text{Let } \mathbf{R} = \phi^T \phi + \lambda \mathbf{I}$$

$$\underline{r} = \sum_{i=1}^N d_i \phi_i = \phi^T \underline{d}$$

At time n , (10) is of the form

$$\mathbf{R}(n) \hat{\underline{w}}(n) = \underline{r}(n) \quad - (13)$$

$$\mathbf{R}(n) = \sum_{i=1}^n \phi_i \phi_i^T + \lambda \mathbf{I} \quad - (11)$$

$$\underline{r}(n) = \sum_{i=1}^n d_i \phi_i \quad - (12)$$

$$\text{From (12), } \underline{r}(n) = \underline{r}(n-1) + d_n \underline{\phi}_n$$

$$\text{From (13), } \underline{r}(n) = \mathbf{R}(n-1) \hat{\underline{w}}(n-1) + d_n \underline{\phi}_n$$

Adding and subtracting $\underline{\phi}_n \underline{\phi}_n^T \hat{\underline{w}}(n-1)$,

$$\begin{aligned} r(n) &= R(n-1) \hat{\underline{w}}(n-1) + \underline{\phi}_n \underline{\phi}_n^T \hat{\underline{w}}(n-1) + d_n \underline{\phi}_n - \underline{\phi}_n \underline{\phi}_n^T \hat{\underline{w}}(n-1) \\ &= [R(n-1) + \underline{\phi}_n \underline{\phi}_n^T] \hat{\underline{w}}(n-1) + \underline{\phi}_n [d_n - \underline{\phi}_n^T \hat{\underline{w}}(n-1)] \quad \text{--- (14)} \end{aligned}$$

From (11),

$$\begin{aligned} R(n) &= \sum_{i=1}^{n-1} \phi_i \phi_i^T + \lambda I + \phi_n \phi_n^T \\ &= R(n-1) + \phi_n \phi_n^T \quad \text{--- (15)} \end{aligned}$$

Defining innovation as $\alpha_n = d_n - \hat{\underline{w}}^T(n-1) \underline{\phi}_n$ --- (16)

$$r(n) = R(n) \hat{\underline{w}}(n) + \phi_n \alpha_n \quad \text{--- (17)}$$

$$R(n) \hat{\underline{w}}(n) = R(n) \hat{\underline{w}}(n-1) + \phi_n \alpha_n$$

$$\hat{\underline{w}}(n) = \hat{\underline{w}}(n-1) + R^{-1}(n) \phi_n \alpha_n$$

Computing R^{-1} by matrix inversion formula

$$R^{-1}(n) = R^{-1}(n-1) - \frac{R^{-1}(n-1) \phi_n \phi_n^T R^{-1}(n-1)}{1 + \phi_n^T R^{-1}(n-1) \phi_n}$$

Let $P(n) = R^{-1}(n)$

$$P(n) = P(n-1) - \frac{P(n-1) \phi_n \phi_n^T P(n-1)}{1 + \phi_n^T P(n-1) \phi_n}$$

$$\therefore \hat{\underline{w}}(n) = \hat{\underline{w}}(n-1) + P(n) \phi_n \alpha_n.$$

Therefore, the regularization term $\frac{1}{2} \lambda \|\underline{w}\|^2$ has no effect on the composition of the RLS algorithm.

b) The only effect of the regularization term is the modification of the correlation matrix as

$$R(n) = \sum_{i=1}^n \phi_i \phi_i^T + \lambda I.$$

The correlation matrix with regularizer is symmetric and positive definite, therefore it is invertible.

The practical benefit of introducing regularization term is avoiding singularity of $R(n)$ which improves stability of $R^{-1}(n)$ computation.

PROBLEM 6-17:

a) Solving the problem using radial basis functions (RBF):

The RBF algorithm corresponds to choosing a function

F of the form

$$F(x) = \sum_{i=1}^N w_i \phi(\|x - x_i\|) \quad [N=8 \text{ in this problem}]$$

• Choose a gaussian function with $\sigma=1$ as

$$\phi(\|x - x_i\|) = \exp\left(-\frac{1}{2} \|x - x_i\|^2\right)$$

• Form the matrix Φ such that $\Phi(i,j) = \phi(\|x_j - x_i\|)$.

• We want $\Phi \underline{w} = \underline{d}$ where \underline{d} is the vector with desired output.

• Solving the system of linear equations $\Phi \underline{w} = \underline{d}$, one can obtain the optimal weight values.

b) Solving the problem using support vector machines:

PROCEDURE:

The given problem is not linearly separable.

• Choosing a Mercer kernel of the form $k(\underline{x}, \underline{x}_i) = (1 + \underline{x}^T \underline{x}_i)^p$

• Let $p=3$ for this problem.

$$k(\underline{x}, \underline{x}_i) = (1 + \underline{x}^T \underline{x}_i)^3$$

• Based on the values of the truth table, compute the values of the matrix $K(i, j) = k(\underline{x}_i, \underline{x}_j)$.

• We obtain an 8×8 matrix.

The objective function for the dual problem is given by

$$Q(\alpha) = \sum_{i=1}^8 \alpha_i - \frac{1}{2} \sum_{i=1}^8 \sum_{j=1}^8 \alpha_i \alpha_j d_i d_j k(\underline{x}_i, \underline{x}_j)$$

• Let $B(i, j) = d_i d_j k(\underline{x}_i, \underline{x}_j)$.

• Then $Q(\alpha) = \alpha^T \mathbf{1} - \frac{1}{2} \alpha^T B \alpha$

• To find optimal α , differentiate w.r.t α and equate to 0:

$$B \alpha = \mathbf{1}.$$

• Solving this based on computation, we get $\alpha_i = \frac{1}{48} \forall i$.

• Further, extract $\underline{\phi}(\underline{x})$ from $k(\underline{x}_i, \underline{x}_j)$.

• The optimum weight vector is obtained as

$$\underline{w}_0 = \sum_{i=1}^8 \alpha_i d_i \underline{\phi}(\underline{x}_i)$$

• Plugging d_i values and $\alpha_i = \frac{1}{48}$, the optimal weight can be obtained.

- The optimal hyperplane is given by $\underline{w}_0^T \underline{\phi}(x) = 0$.
 - Based on the above computations, we get $x_1 x_2 x_3 = 0$.
 - This satisfies the given problem of assigning $+1$ if $x_1 x_2 x_3 > 0$ and -1 if $x_1 x_2 x_3 < 0$.
- ⇒ You can verify the above by both hand calculations as well as by writing a code.

In this problem, all the N samples are support vectors.

The computational complexity of both the procedures for this problem is the same of complexity $O(N^3)$ as both of them require matrix inversion.

PROBLEM 6.25:

- Generate the data as shown in the figure.
- Choose an appropriate kernel like the RBF for training SVM.
- Perform experiments with various combinations of hyperparameter and value of C .
- Compute the classification error rate and check for overfitting.