CSCI 5521 – Introduction to Machine Learning

Homework – 1

Mourya Karan Reddy Baddam - 5564234

Q1. i) $E(w_1, w_0 \mid \mathcal{X}_{train}) = \frac{1}{N} \sum_{t=1}^{N} (r^t - (w_1 x^t + w_0))^2$

To minimize $\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial w_0} = 0$

$$\frac{\partial E}{\partial w_0} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} (r^t - w_1 x^t - w_0) = 0.$$

Dividing by 2

$$\Rightarrow \frac{\sum_{t=1}^{N} r^t}{N} - w_1 \frac{\sum_{t=1}^{N} x^t}{N} = \frac{\sum_{t=1}^{N} w_0}{N}$$

Let $\frac{\sum_{t=1}^{N} r^t}{N} = \bar{r}$ & $\frac{\sum_{t=1}^{N} x^t}{N} = \bar{x}$.

$$\Rightarrow w_0 = \bar{r} - w_1 \bar{x} \quad \text{——} \quad (1).$$

$$\frac{\partial E}{\partial w_1} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} (r^t - w_1 x^t - w_0) x^t = 0.$$

Dividing by 2.

$$\Rightarrow \frac{\sum_{t=1}^{N} r^t x^t}{N} - w_1 \frac{\sum_{t=1}^{N} (x^t)^2}{N} - w_0 \frac{\sum_{t=1}^{N} x^t}{N} = 0.$$

Substituting w_0 from eqn (1)

$$\Rightarrow \frac{\sum_{t=1}^{N} r^t x^t}{N} - w_1 \frac{\sum_{t=1}^{N} (x^t)^2}{N} - (\bar{r} - w_1 \bar{x}) \bar{x} = 0.$$

$$\Rightarrow \frac{\sum_{t=1}^{N} (r^t x^t - \bar{r}\bar{x})}{N} = w_1 \frac{\sum_{t=1}^{N} ((x^t)^2 - \bar{x}^2)}{N}.$$

$$\Rightarrow w_1 = \frac{\frac{1}{N} \sum_{t=1}^{N} (r^t x^t - \bar{r}\bar{x})}{\frac{1}{N} \sum_{t=1}^{N} ((x^t)^2 - \bar{x}^2)}$$

$$w_0 = \bar{r} - w_1 \bar{x}$$

The above values are optimal for w_1 & w_0.

ii) $E(v_2, v_1, v_0 | \mathcal{X}_{train}) = \frac{1}{N} \sum_{t=1}^{N} \left(r^t - (v_2 (x^t)^{2020} + v_1 x^t + v_0)\right)^2$

To minimize, $\frac{\partial E}{\partial v_0} = \frac{\partial E}{\partial v_1} = \frac{\partial E}{\partial v_2} = 0.$

$$\frac{\partial E}{\partial v_0} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} (r^t - v_2 (x^t)^{2020} - v_1 x^t - v_0) = 0$$

$$\Rightarrow v_2 \sum_{t=1}^{N} (x^t)^{2020} + v_1 \sum_{t=1}^{N} x^t + N v_0 = \sum_{t=1}^{N} r^t \quad \text{— (1)}$$

$$\frac{\partial E}{\partial v_1} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} (r^t - v_2 (x^t)^{2020} - v_1 x^t - v_0) x^t = 0.$$

$$\Rightarrow v_2 \sum_{t=1}^{N} (x^t)^{2021} + v_1 \sum_{t=1}^{N} (x^t)^2 + v_0 \sum_{t=1}^{N} x^t = \sum_{t=1}^{N} r^t x^t \quad \text{— (2)}$$

$$\frac{\partial E}{\partial v_2} = 0 \Rightarrow \frac{2}{N} \sum_{t=1}^{N} (r^t - v_2 (x^t)^{2020} - v_1 x^t - v_0)(x^t)^{2020} = 0.$$

$$\Rightarrow v_2 \sum_{t=1}^{N} (x^t)^{4040} + v_1 \sum_{t=1}^{N} (x^t)^{2021} + v_0 \sum_{t=1}^{N} (x^t)^{2020} = \sum_{t=1}^{N} r^t (x^t)^{2020} \quad \text{— (3)}$$

The above 3 equations can be represented in matrix form as below:

$$\begin{bmatrix} \sum_{t=1}^{N} (x^t)^{2020} & \sum_{t=1}^{N} x^t & N \\ \sum_{t=1}^{N} (x^t)^{2021} & \sum_{t=1}^{N} (x^t)^2 & \sum_{t=1}^{N} x^t \\ \sum_{t=1}^{N} (x^t)^{4040} & \sum_{t=1}^{N} (x^t)^{2021} & \sum_{t=1}^{N} (x^t)^{2020} \end{bmatrix} = A.$$

$$\begin{bmatrix} v_2 \\ v_1 \\ v_0 \end{bmatrix} = v$$

$$\begin{bmatrix} \sum_{t=1}^{N} r^t \\ \sum_{t=1}^{N} r^t x^t \\ \sum_{t=1}^{N} r^t (x^t)^{2020} \end{bmatrix} = b.$$

and $Av = b$.

* A unique solution exists if A^{-1} exists and it is given by $v = A^{-1}b$. If A^{-1} does not exists then a unique solution doesn't exist.

iii. Yes. Professor Gopher’s claim is correct.
For a training set, given a sufficiently large N value, a higher degree basis function always fits better or same as a linear basis function. So, the training error for a polynomial basis function is either same or less than the linear basis function.

*But the actual generalized performance might be the opposite if the model overfits the given training data.

Q2. Using numpy.trace() and numpy.matmul(), the following values are computed:

i. $tr(A) = tr(A_T) = 76$
$tr(A.\ A_T) = tr(A_T.A) = 5278$

ii. |A| is the volume of the 4-dimensional parallelepiped formed by column vectors or row vectors of A.
|A| can also be found by dividing it into three 3-dimensional matrices as below (which is easier and practical to imagine):

$$|A| = 1 \text{ x } \begin{vmatrix} 2 & 4 & 8 \\ 3 & 9 & 27 \\ 4 & 16 & 64 \end{vmatrix} - 1 \text{ x } \begin{vmatrix} 1 & 4 & 8 \\ 1 & 9 & 27 \\ 1 & 16 & 64 \end{vmatrix} + 1 \text{ x } \begin{vmatrix} 1 & 2 & 8 \\ 1 & 3 & 27 \\ 1 & 4 & 64 \end{vmatrix} - 1 \text{ x } \begin{vmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{vmatrix}$$

Now, the determinant of each 3-dimentional square matrix is the volume of the parallelepiped formed by its column vectors or row vectors.

iii. Yes. The rows of A are linearly independent.
A is a Vandermonde matrix (Each row is a Geometric Progression).
$|A| = \prod (a_j-a_i)$, where a_j is the common ratio of j_{th} row and $1 \le i \le j \le n$.
Therefore $|A| = (4-1)*(4-2)*(4-3)*(3-1)*(3-2)*(2-1) = 3*2*2 = 12$.

Since $|A| \neq 0$, the rows of A are linearly independent.

Q.3.(i): Summary:

Error rates LinearSVC with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
15.69%	13.73%	13.73%	11.76%	17.65%	15.69%	22.0%	34.0%	44.0%	16.0%	20.42%	9.9%

Error rates LinearSVC with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
45.1%	27.45%	9.8%	13.73%	7.84%	9.8%	12.0%	18.0%	10.0%	14.0%	16.77%	10.86%

Error rates LinearSVC with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
2.78%	5.56%	5.0%	9.44%	7.22%	7.78%	2.78%	5.03%	3.35%	6.15%	5.51%	2.1%

Error rates SVC with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
27.45%	27.45%	19.61%	23.53%	31.37%	19.61%	28.0%	18.0%	10.0%	28.0%	23.3%	6.13%

Error rates SVC with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
23.53%	27.45%	31.37%	25.49%	15.69%	27.45%	14.0%	26.0%	24.0%	24.0%	23.9%	5.04%

Error rates SVC with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
0.0%	1.11%	0.0%	0.56%	2.22%	2.22%	0.56%	1.68%	0.56%	0.0%	0.89%	0.83%

Error rates LogisticRegression with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
21.57%	15.69%	13.73%	13.73%	19.61%	11.76%	14.0%	4.0%	16.0%	12.0%	14.21%	4.52%

Error rates LogisticRegression with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
9.8%	7.84%	9.8%	13.73%	7.84%	5.88%	4.0%	12.0%	8.0%	10.0%	8.89%	2.68%

Error rates LogisticRegression with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
3.89%	5.56%	1.67%	3.89%	3.33%	5.56%	1.67%	2.79%	3.35%	3.91%	3.56%	1.27%

Q.3.(ii): Summary:

Error rates LinearSVC with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
12.7%	11.11%	29.37%	31.75%	39.68%	20.63%	30.16%	19.84%	26.19%	24.6%	24.6%	8.35%

Error rates LinearSVC with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
12.7%	12.7%	23.02%	18.25%	7.94%	7.94%	23.81%	38.1%	14.29%	13.49%	17.22%	8.66%

Error rates LinearSVC with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
6.68%	6.24%	5.57%	5.35%	4.68%	4.68%	4.68%	5.35%	5.35%	6.24%	5.48%	0.68%

Error rates SVC with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
26.98%	23.02%	18.25%	20.63%	17.46%	20.63%	28.57%	30.95%	28.57%	24.6%	23.97%	4.46%

Error rates SVC with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
23.02%	29.37%	15.08%	24.6%	26.98%	21.43%	16.67%	22.22%	24.6%	19.05%	22.3%	4.22%

Error rates SVC with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
0.45%	0.89%	0.0%	0.89%	1.11%	1.34%	1.11%	0.22%	0.67%	0.89%	0.76%	0.4%

Error rates LogisticRegression with Boston50:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
11.11%	11.11%	19.84%	17.46%	16.67%	12.7%	13.49%	15.08%	11.9%	11.9%	14.13%	2.86%

Error rates LogisticRegression with Boston75:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
11.11%	9.52%	11.11%	11.11%	7.94%	7.14%	10.32%	10.32%	10.32%	7.94%	9.68%	1.41%

Error rates LogisticRegression with Digits:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
4.45%	1.78%	3.79%	4.68%	4.23%	4.23%	4.45%	4.45%	4.01%	3.12%	3.92%	0.83%

Q4. Summary:

Error rates LinearSVC with $\tilde{X}_1$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
10.56%	12.78%	20.56%	9.44%	11.11%	9.44%	10.56%	8.94%	9.50%	12.85%	11.57%	3.26%

Error rates LinearSVC with $\tilde{X}_2$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
1.67%	1.11%	1.67%	0.56%	0.0%	0.56%	1.11%	0.56%	0.56%	1.68%	0.95%	0.56%

Error rates SVC with $\tilde{X}_1$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
2.22%	0.56%	2.22%	1.67%	2.22%	1.11%	2.22%	0.56%	3.35%	3.35%	1.95%	0.94%

Error rates SVC with $\tilde{X}_2$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
0.56%	1.11%	2.78%	0.56%	1.11%	0.56%	1.67%	0.0%	0.56%	2.23%	1.11%	0.83%

Error rates LogisticRegression with $\tilde{X}_1$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
7.78%	7.22%	11.67%	6.67%	5.56%	6.11%	6.67%	6.7%	8.38%	7.82%	7.46%	1.62%

Error rates LogisticRegression with $\tilde{X}_2$:

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Mean	Std
2.22%	1.11%	2.22%	1.67%	1.67%	0.56%	0.56%	0.56%	0.56%	1.12%	1.22%	0.65%