CSC311 Homework 1

Xuanqi Wei
1009353209

September 26, 2024

1 Question 1

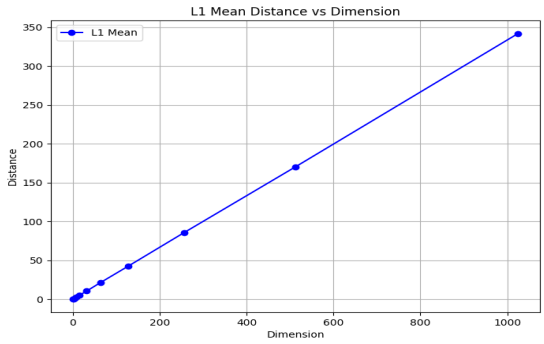
(a) We need a minimum of 50 data points to guarantee that any new test point is within 0.01 of an old point, the calculation is provided by

$$\frac{1}{0.02} = 50$$

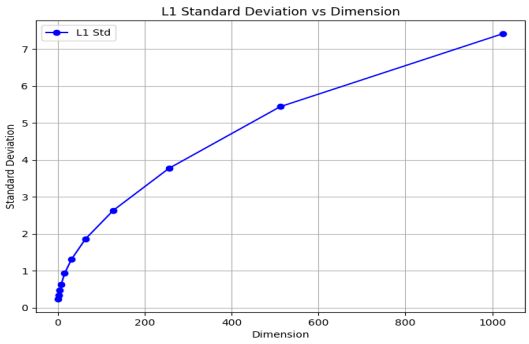
Since all 50 points in S are distributed in [0, 1] with same distances, all the points on the interval [0, 1] will have a distance within 0.01 of one of those points, which mean 50 points satisfy the requirement.

(b) When we are working on a problem with 10 features, we need to consider distance's calculations in higher dimension. In the case when 10 features are applied, there are 10 dimensions that we need to take into account which we need to ensure in each dimension, that the difference of distance between two points remain within 0.01, leading the calculation growing exponentially.

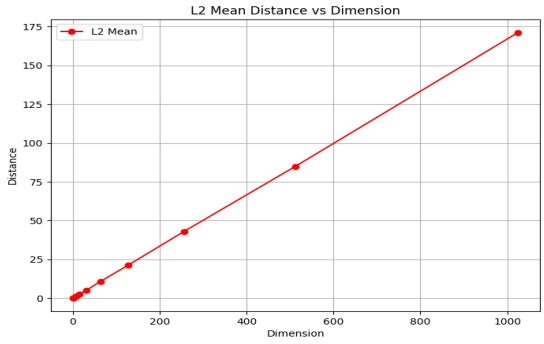
(c) I'll firstly provide the figures separately.



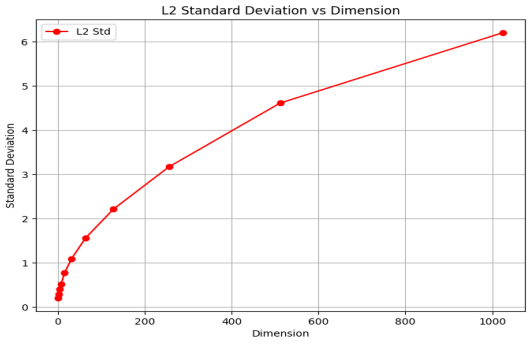
(a) l1 Distance Average



(b) l2 Distance Standard Deviation



(c) l2 Distance Average



(d) l2 Distance Standard deviation

Figure 1: Separate View of four figures

I'll then provide the distance average in one figure and distance standard deviation in one figure.

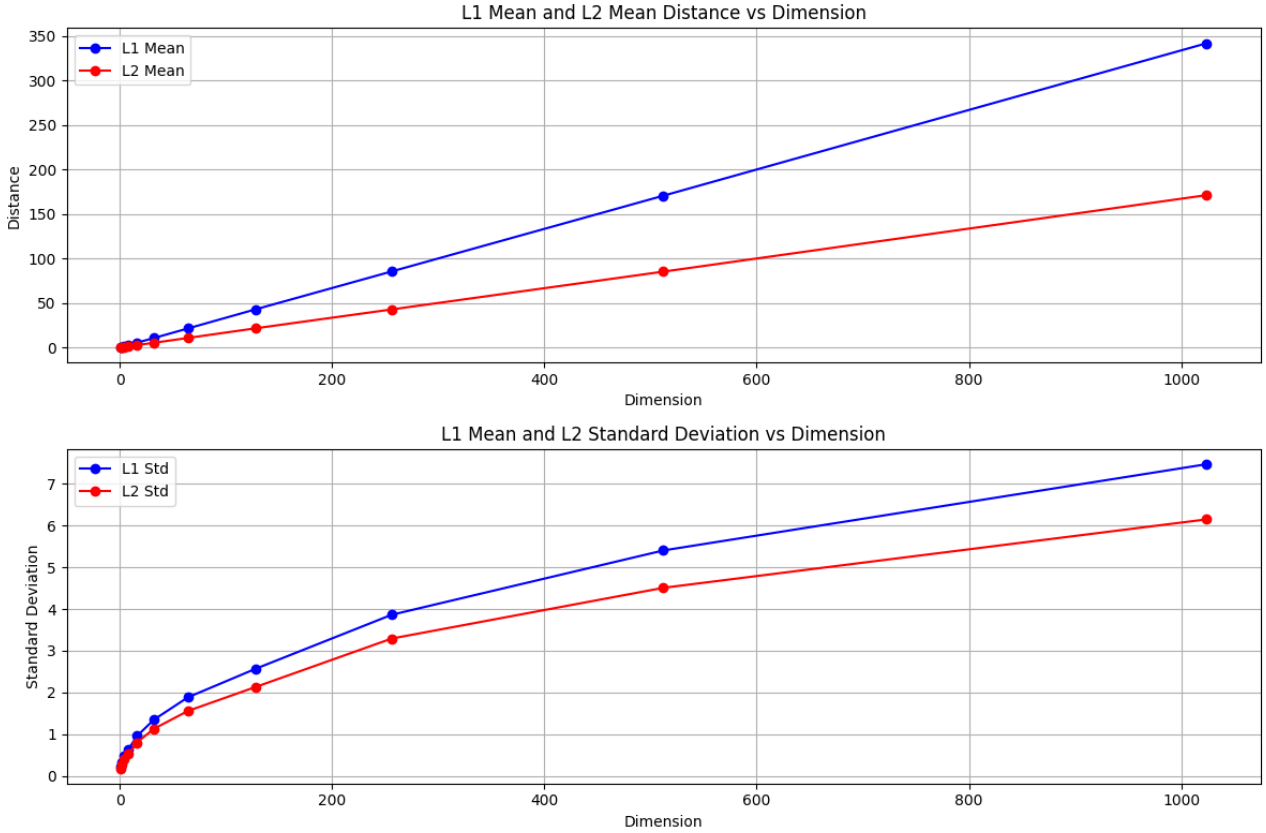


Figure 2: Comparing them for neatness

(d) From the basic rules, the expectation is given by:

$$E[R] = E[Z_1 + ... + Z_d] \tag{1}$$
$$= E[Z_1] + .. + E[Z_d] \tag{2}$$
$$= d \cdot E[Z] = \frac{3}{d} \tag{3}$$

Since $\forall i$, X_i and Y_i are sampled independently, if $i \neq j$, Z_i and Z_j are also independent. From the basic rules, the variance is given by:

$$Var[R] = Var[Z_1 + ... + Z_d] \tag{4}$$
$$= Var[Z_1] + ... + Var[Z_d] \tag{5}$$
$$= d \cdot Var[Z] = \frac{20}{23}d \tag{6}$$

(e) (i) E can be written in mathematics:

$$E = |R - E(r)| \geq r$$

(ii) According to Markov's Inequality:

$$P(E) = P(|R - E(r)| \geq r) \leq \frac{Var(R)}{r^2} = \frac{20d}{23r^2}$$

(iii) Given that r $= c \cdot d$. When d approaches ∞, gives:

$$\lim_{d\to\infty} P(E) = \lim_{d\to\infty} \frac{20d}{23r^2} = \lim_{d\to\infty} \frac{20d}{23(cd)^2} = \frac{20d}{23c^2} \cdot \lim_{d\to\infty} \frac{1}{d} = \frac{20}{23c^2} \cdot 0 = 0$$

Therefore, as the number of dimensions goes to infinity, P(E) goes to 0.

2 Question 2

(a) (i) Since the matrix A is a dimension of $d \times p$, the squared Euclidean distance as a quadratic form is given by:

$$\begin{aligned} d_A(x_1, x_2)^2 &= ||A^T(x_1 - x_2)||_2^2 && (7)\\ &= (A^T(x_1 - x_2))^T \cdot (A^T(x_1 - x_2)) && (8)\\ &= x_1^T AA^T x_1 - 2x_1^T AA^T x_2 + x_2^T AA^T x_2 && (9)\\ &= (x_1 - x_2)^T AA^T (x_1 - x_2) && (10) \end{aligned}$$

(ii) (a) For non-negativity, since the norm of a vector is always non-negative, we have:

$$d_A(x_1, x_2) = ||A^T(x_1 - x_2)||_2 \geq 0$$

(ii) (b) For symmetry, we have:

$$d_A(x_1, x_2) = ||A^T(x_1 - x_2)|| = || - A^T(x_2 - x_1)|| = ||A^T(x_2 - x_1)|| = d_A(x_2, x_1)$$

(ii) (c) For Triangle Inequality, we want to show that: $d_A(x_1, x_3) \leq d_A(x_1, x_2) + d_A(x_2, x_3)$. I'll take $u = x_3 - x_2$, $v = x_2 - x_1$, gives:

$$\begin{aligned} & d_A(x_1, x_3) \leq d_A(x_1, x_2) + d_A(x_2, x_3) && (11)\\ \iff & ||A^T(u + v)||_2 \leq ||A^T u||_2 + ||A^T v||_2 && (12)\\ \iff & ||A^T(u + v)||_2^2 \leq (||A^T u||_2 + ||A^T v||_2)^2 && (13) \end{aligned}$$

According to the Cauchy-Schwarz Inequality, we have

$$\begin{aligned} ||A^T(u + v)||_2^2 &= ||A^T u + A^T v||^2 && (14)\\ &\leq ||A^T u||^2 + 2||A^T u|| \cdot ||A^T v|| + ||A^T v||^2 && (15)\\ &= (||A^T u||_2 + ||A^T v||_2)^2 && (16) \end{aligned}$$

Thus, since the norm of a vector is always non-negative, and from (11), (12), (13), gives,

$$\begin{aligned} & ||A^T(u + v)||_2^2 \leq (||A^T u||_2 + ||A^T v||_2)^2 && (17)\\ \iff & ||A^T(u + v)||_2 \leq ||A^T u||_2 + ||A^T v||_2 && (18)\\ \iff & d_A(x_1, x_3) \leq d_A(x_1, x_2) + d_A(x_2, x_3) \text{ as needed.} && (19) \end{aligned}$$

(b) (i) p is the hyperparameter that is present in this algorithm but that is not present in the standard KNN algorithm.

(ii) This approach is parametric since the embedding matrix introduces new parameters, and the embedding matrix A contains the learned parameters.

(iii) The inner optimization aligns with the KNN inference procedure, ensuring that the k-nearest neighbors in the embedded space belong to the same majority class. Instead of assigning a class, it selects the embedding matrix A that maximizes the count of neighbors with the

clearest outcomes, focusing on finding the configuration that results in the most cohesive class representation.

(iv) The bi-level optimization problem solves the embedded nearest neighbors problem by letting the outer optimization yielding an optimal embedding matrix that maximizes the inner optimization for all data points, which in this case is A. This can produce an optimal embedded space for the KNN probalm, which transforms the original data and optimize the nearest neighbors as needed.

(v) When $p = d$, the data is embedded into a $d-2$ dimensional space, which $d-2$ non-zero eigenvalues exist. Since there are d dimensions in the original space, and currently the data is embedded into a lower-dimensional, i.e., $d-2$, this implies that 2 dimensions are ignored.

3 Question 3

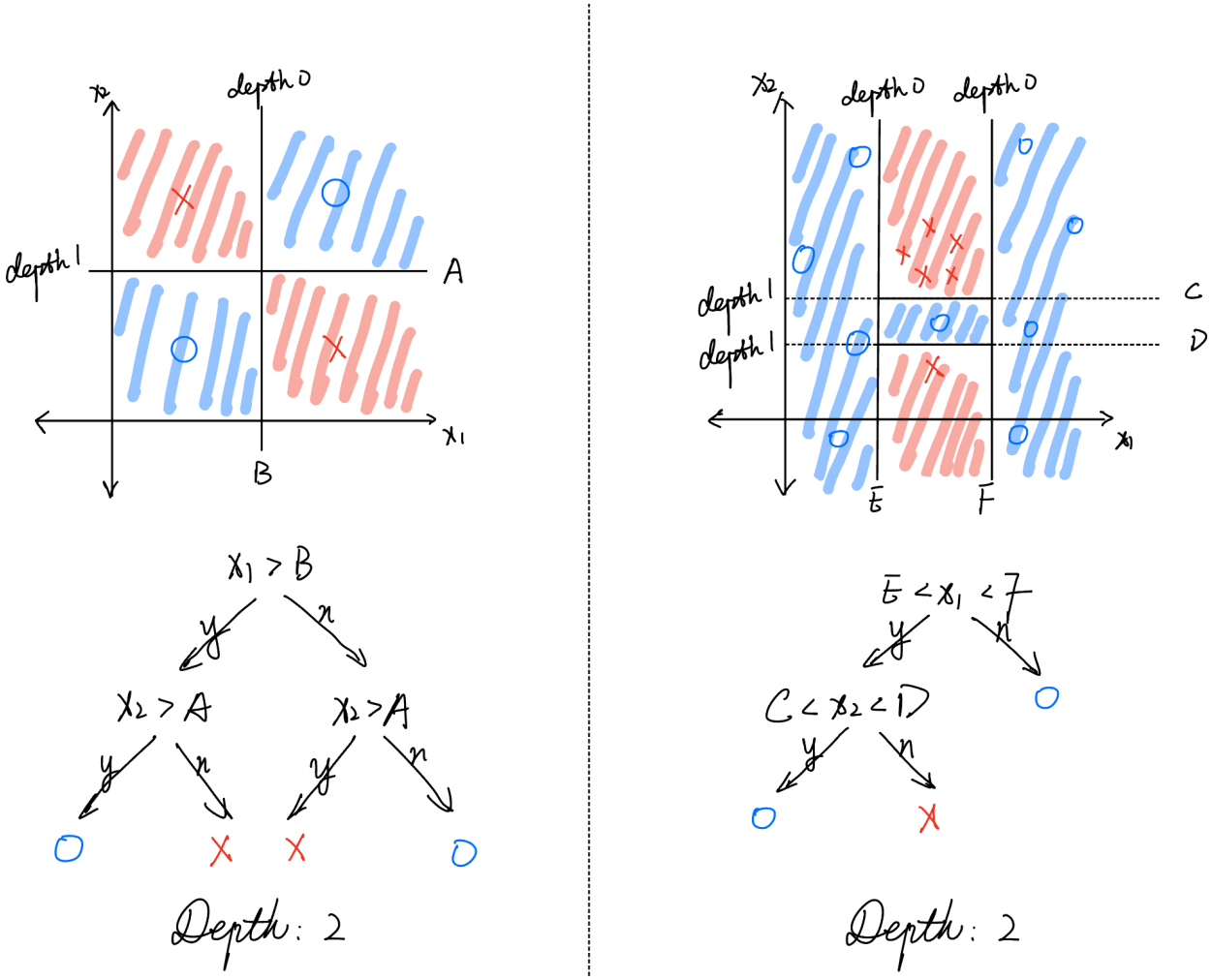


Figure 3: Optimal (Binary) Decision Tree

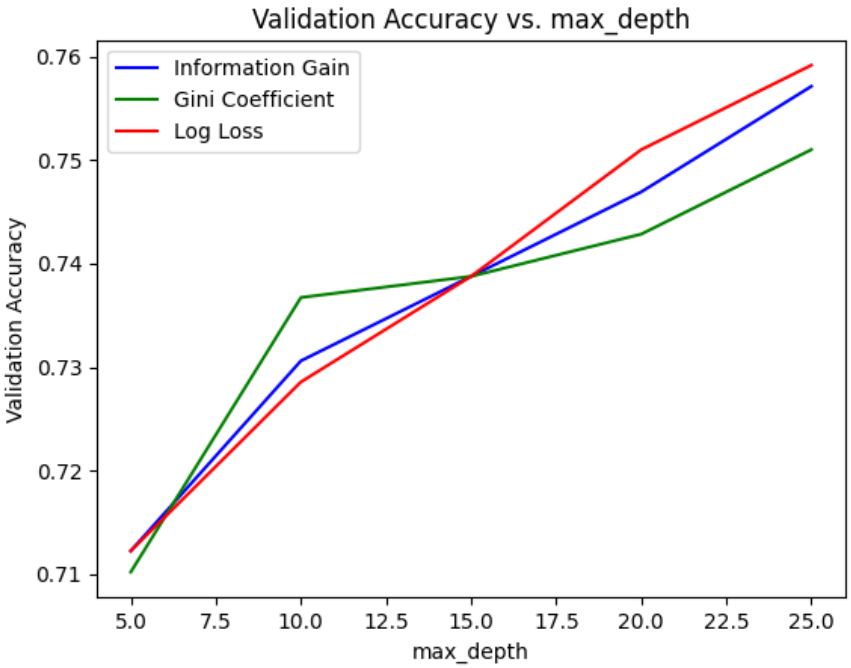
4 Question 4

(b)

```
/opt/homebrew/anaconda3/envs/ML/bin/python /Users/henry-wxq/Documents/Programming/CS_Notes/Python/04_CSC311_Intro_ML/HW/hw1_q4.py
Depth 5:
Validation Accuracy using entropy: 0.7122448979591837
Validation Accuracy using Gini: 0.710204081632653
Validation Accuracy using log loss: 0.7122448979591837
Depth 10:
Validation Accuracy using entropy: 0.7306122448979592
Validation Accuracy using Gini: 0.736734693877551
Validation Accuracy using log loss: 0.7285714285714285
Depth 15:
Validation Accuracy using entropy: 0.7387755102040816
Validation Accuracy using Gini: 0.7387755102040816
Validation Accuracy using log loss: 0.7387755102040816
Depth 20:
Validation Accuracy using entropy: 0.746938775510204
Validation Accuracy using Gini: 0.7428571428571429
Validation Accuracy using log loss: 0.7510204081632653
Depth 25:
Validation Accuracy using entropy: 0.7571428571428571
Validation Accuracy using Gini: 0.7510204081632653
Validation Accuracy using log loss: 0.7591836734693878

Process finished with exit code 0
```

(a) Output of the Function



(b) the Validation Accuracy v.s. max_depth

(c)

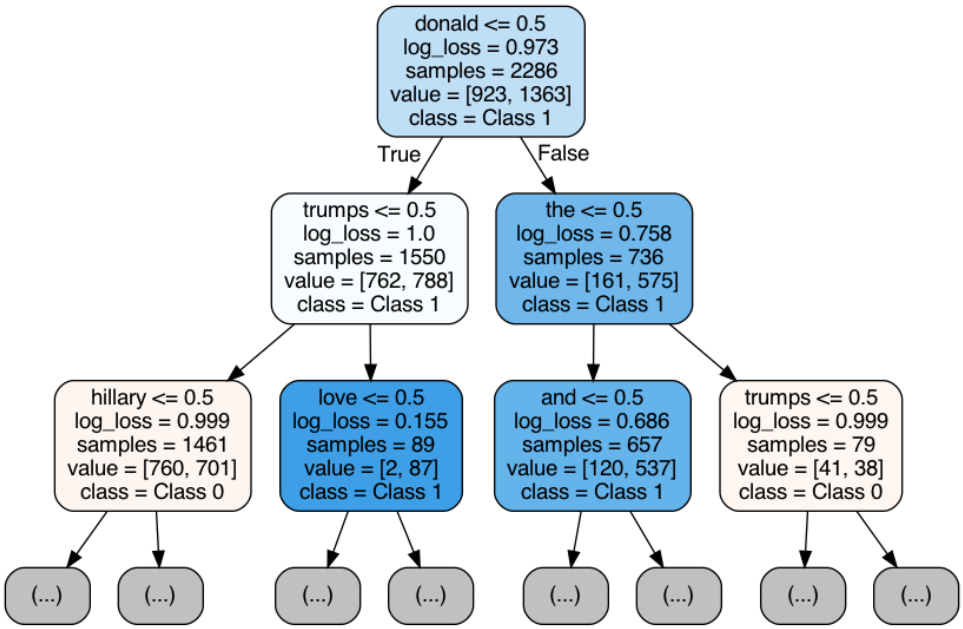


Figure 5: The Information Gain

(d)

```
Information Gain for keyword donald: 0.05119976458379005
Information Gain for keyword trumps: 0.04345964433678118
Information Gain for keyword the: 0.0475553113996825
Information Gain for keyword hillary: 0.035325902154559774
Information Gain for keyword coal: 0.00029934606170567424
```

Figure 6: The Information Gain

5 Question 5

(a) Given that the formulation,

$$J_{reg}^{\alpha\beta}(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})^2 + \sum_{j=1}^{D}\alpha_j|w_j| + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2$$

I'll calculate the gradient descent rules for each by separating them into two parts:

$$J = \frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})^2 \tag{20}$$

$$R = \sum_{j=1}^{D}\alpha_j|w_j| + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2 \tag{21}$$

For (20), we have,

$$\frac{\partial J}{\partial w_j} = \frac{\partial(\frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})^2)}{\partial w_j} = \frac{1}{2N}\cdot\sum_{i=1}^{N}\frac{\partial(y^{(i)} - t^{(i)})^2}{\partial w_j} \tag{22}$$

$$= \frac{1}{2N}\sum_{i=1}^{N}2(y^{(i)} - t^{(i)})\cdot\frac{\partial(y^{(i)} - t^{(i)})}{\partial w_j} \quad \text{since } y = \sum_{j=1}^{D}w_j x_j + b, \text{ gives,} \tag{23}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})\frac{\partial(w^T x^{(i)} + b - t^{(i)})}{\partial w_j} \tag{24}$$

$$= \frac{1}{N}\sum_{i=1}^{N}2(y^{(i)} - t^{(i)})\cdot x_j^{(i)} \tag{25}$$

For (21), we have,

$$\frac{\partial R}{\partial w_j} = \frac{\partial(\sum_{j=1}^{D}\alpha_j|w_j| + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2)}{\partial w_j} \tag{26}$$

$$= \alpha_j\frac{\partial|w_j|}{\partial w_j} + \beta_j w_j \tag{27}$$

$$= \alpha\frac{w_j}{|w_j|} + \beta_j w_j \tag{28}$$

$$= \alpha_j\cdot\text{sgn}(x) + \beta_j w_j \tag{29}$$

where sgn is the sign function for w_j,

$$\text{sgn}(w_j) = \begin{cases} 1, & \text{if } w_j > 0, \\ 0, & \text{if } w_j = 0, \\ -1, & \text{if } w_j < 0. \end{cases}$$

Thus, from (25) and (29), we obtain that,

$$\frac{J_{reg}^{\alpha\beta}}{\partial w_j} = \frac{\partial(J+R)}{\partial w_j} = \frac{\partial J}{\partial w_j} + \frac{\partial R}{\partial w_j} \tag{30}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) \cdot x_j^{(i)} + \alpha_j \cdot sgn(w_j) + \beta_j w_j \tag{31}$$

Also, for b, we have,

$$\frac{\partial J_{reg}^{\alpha\beta}}{\partial b} = \frac{\partial(J+R)}{\partial b} = \frac{\partial J}{\partial b} + 0 \tag{32}$$

$$= \frac{\partial(\frac{1}{2N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)})^2)}{\partial b} \tag{33}$$

$$= \frac{\partial(\frac{1}{2N}\sum_{i=1}^{N}(\sum_{j=1}^{D} w_j x_j + b - t^{(i)})^2)}{\partial b} \tag{34}$$

$$= \frac{1}{2N} \cdot \sum_{i=1}^{N} 2(\sum_{j=1}^{D} w_j x_j^{(i)} + b - t^{(i)}) \tag{35}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(\sum_{j=1}^{D} w_j x_j^{(i)} + b - t^{(i)}) \tag{36}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - t^{(i)}) \tag{37}$$

Let $\alpha > 0$ be the learning rate. According to (31), (37), we have:
If $w_j > 0$:

$$w_j \leftarrow w_j - \alpha\left(\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)x_j^{(i)} + \alpha_j + \beta_j w_j\right)$$

$$\iff w_j \leftarrow (1-\alpha\beta_j)w_j - \frac{\alpha}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)x_j^{(i)} - \alpha\alpha_j$$

$$b \leftarrow b - \alpha\left(\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)\right)$$

If $w_j = 0$:

$$w_j \leftarrow w_j - \alpha\left(\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)x_j^{(i)}\right)$$

$$b \leftarrow b - \alpha\left(\frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)\right)$$

If $w_j < 0$:

$$w_j \leftarrow w_j - \alpha \left(\frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right) x_j^{(i)} - \alpha_j + \beta_j w_j \right)$$

$$\iff w_j \leftarrow (1 - \alpha\beta_j) w_j - \frac{\alpha}{N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right) x_j^{(i)} + \alpha\alpha_j$$

$$b \leftarrow b - \alpha \left(\frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - t^{(i)} \right) \right)$$

(b) $\lambda_1 = 0$ means $\mathcal{J}_{\text{reg}}^{\beta}(w) = \frac{1}{2N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)})^2 + \frac{1}{2} \sum_{j'=1}^{D} \beta_{j'} w_{j'}^2$.

$$\frac{\partial \mathcal{J}_{\text{reg}}^{\beta}(w)}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - t^{(i)}) x_j^{(i)} + \beta_j w_j \tag{38}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} - t^{(i)} \right) x_j^{(i)} + \beta_j w_j \tag{39}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} x_j^{(i)} - \frac{1}{N} \sum_{i=1}^{N} t^{(i)} x_j^{(i)} + \beta_j w_j \tag{40}$$

$$= \sum_{j'=1}^{D} \left(\frac{1}{N} \sum_{i=1}^{N} x_{j'}^{(i)} x_j^{(i)} \right) w_{j'} + \delta(j = j') \beta_j w_j - \frac{1}{N} \sum_{i=1}^{N} t^{(i)} x_j^{(i)} \tag{41}$$

$$= \sum_{j'=1}^{D} \left(\frac{1}{N} \sum_{i=1}^{N} x_{j'}^{(i)} x_j^{(i)} + \delta(j = j') \beta_j \right) w_j - \frac{1}{N} \sum_{i=1}^{N} t^{(i)} x_j^{(i)} \tag{42}$$

$$= \sum_{j'=1}^{D} A_{jj'} w_{j'} - c_j. \tag{43}$$

Therefore, we have

$$A_{jj'} = \frac{1}{N} \sum_{i=1}^{N} x_{j'}^{(i)} x_j^{(i)} + I(j = j') \beta_j \quad \text{and} \quad c_j = \frac{1}{N} \sum_{i=1}^{N} t^{(i)} x_j^{(i)}.$$

(c) From (b), $A_{jj'} = \frac{1}{N}\sum_{i=1}^{N} x_{j'}^{(i)} x_j^{(i)} + I(j = j')\beta_j$ and $c_j = \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_j^{(i)}$. Thus, the formulas for A and c are as follows:

$$
\begin{aligned}
A &= \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} x_1^{(i)} x_1^{(i)} & \cdots & \frac{1}{N}\sum_{i=1}^{N} x_1^{(i)} x_D^{(i)} \\ \vdots & \ddots & \vdots \\ \frac{1}{N}\sum_{i=1}^{N} x_D^{(i)} x_1^{(i)} & \cdots & \frac{1}{N}\sum_{i=1}^{N} x_D^{(i)} x_D^{(i)} \end{bmatrix} + \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_D \end{bmatrix} \\
&= \frac{1}{N} \begin{bmatrix} \sum_{i=1}^{N} x_1^{(i)} x_1^{(i)} & \cdots & \sum_{i=1}^{N} x_1^{(i)} x_D^{(i)} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} x_D^{(i)} x_1^{(i)} & \cdots & \sum_{i=1}^{N} x_D^{(i)} x_D^{(i)} \end{bmatrix} + \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_D \end{bmatrix} \\
&= \frac{1}{N} X^T X + \operatorname{diag}(\beta) \\
c &= \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_1^{(i)} \\ \vdots \\ \frac{1}{N}\sum_{i=1}^{N} t^{(i)} x_D^{(i)} \end{bmatrix} \\
&= \frac{1}{N} \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(N)} \\ \vdots & \ddots & \vdots \\ x_D^{(1)} & \cdots & x_D^{(N)} \end{bmatrix} \begin{bmatrix} t^{(1)} \\ \vdots \\ t^{(N)} \end{bmatrix} \\
&= \frac{1}{N} X^T t
\end{aligned}
$$

The closed-form solution for the parameter w is:

$$
\begin{aligned}
Aw - c &= 0 \\
\left(\frac{1}{N} X^T X + \operatorname{diag}(\beta)\right) w - \frac{1}{N} X^T t &= 0 \\
\left(X^T X + N \operatorname{diag}(\beta)\right) w &= X^T t \\
w &= \left(X^T X + N \operatorname{diag}(\beta)\right)^{-1} X^T t
\end{aligned}
$$

where I'm assuming $X^T X + N \operatorname{diag}(\beta)$ is invertible.