HW 2

Matthew Jensen

2024-10-14

Part 1

Problem 1: PAC Learning

A.

i.

According to Occum’s Razor, the simpler explanation is the best one. Using this justification, L2 would be the better learning algorithm as its hypothesis space H2 is smaller that the hypothesis space of L1.

ii.

In the PAC formula, a bigger Hypothesis space would result in a bigger number from log(|H|). When keeping the value for the error and confidence level fixed, a bigger Hypothesis space would require a larger number of m training samples for generalization. Thus, using Occum’s Razor, it is better to have a smaller hypothesis space because it require less data to achieve the same accuracy as a bigger hypothesis space.

iii.

$$m > \frac{1}{\epsilon}(log(|H|) + \frac{1}{\delta})$$

$$m > \frac{1}{1-.90}(log(3^{10}) + log(\frac{1}{1-.95}) \approx 60.71$$

Around **61** training samples would be needed for L1.

Problem 2 Proof

$$\epsilon_t = \frac{1}{2} - \frac{1}{2}(\sum_{i=1}^{m} D(i) y_i h_t(x_i))$$

$$\sum_{i=1}^{m} D(i) y_i h_t(x_i) = \sum_{y_i = h_t(x)} D(t_i) - \sum_{y_i \neq h_t(x)} D(t_i)$$

$$\epsilon_t = \frac{1}{2} - \frac{1}{2}(\sum_{y_i = h_t(x)} D(t_i) - \sum_{y_i \neq h_t(x)} D(t_i))$$

$$\epsilon_t = \frac{1}{2} - \frac{1}{2}\sum_{y_i = h_t(x)} D(t_i) + \frac{1}{2}\sum_{y_i \neq h_t(x)} D(t_i)$$

$$\text{By definition, Weights sum to 1: } \sum_{i}^{m} D(t_i) = \sum_{y_i = h_t(x)} D(t_i) + \sum_{y_i \neq h_t(x)} D(t_i) = 1$$

$$\sum_{y_i = h_t(x)} D(t_i) = 1 - \sum_{y_i \neq h_t(x)} D(t_i)$$

$$\text{Subsitution: } \epsilon_t = \frac{1}{2} - \frac{1}{2}(1 - \sum_{y_i \neq h_t(x)} D(t_i)) + \frac{1}{2}\sum_{y_i \neq h_t(x)} D(t_i)$$

$$\epsilon_t = \frac{1}{2} - \frac{1}{2} + \frac{1}{2}(\sum_{y_i \neq h_t(x)} D(t_i)) + \frac{1}{2}(\sum_{y_i \neq h_t(x)} D(t_i))$$

$$= 0 + \frac{2}{2}\sum_{y_i \neq h_t(x)} D(t_i)$$

$$= \sum_{y_i \neq h_t(x)} D(t_i)$$

Problem 3 Linear Classifiers

a.

$$f(x_1, x_2, x_3) \text{ outputs 1 when } x_1 = 1, x_2 = 0, x_3 = 0$$

$$\text{linear classifiers are defined as : } \mathbf{w} \cdot x + b = 0$$

$$\text{When } \mathbf{w} \cdot x + b > 0 \text{ the output is 1, When } \mathbf{w} \cdot x + b \leq 0 \text{ the output is 0}$$

Cases

Postive output

$$w_1 \cdot 1 + w_2 \cdot 0 + w_3 \cdot 0 + b = 0$$

$$w_1 + b = 0, b = -w_1 \text{or } b = 0$$

Negative output

$$f(0, 0, 0) : w = [0, 0, 0], b \leq 0$$

$$f(1, 1, 0) : w = [1, 1, 0], w_1 + w_2 + b \leq 0, \text{ if b = 0, } w_1 = -w_2$$

$$f(1, 0, 1) : w = [1, 0, 1], w_1 + w_3 + b \leq 0, \text{ if b = 0, } w_1 = -w_3$$

$$f(1, 1, 1) : w = [1, 1, 1], w_1 + w_2 + w_3 + b \leq 0, \text{ if b = 0, } w_1 + w_2 + w_3 < 0$$

Conclusion

Thus, a possible answer for this systems of equations is w = [1,-1,-1], b = 0 and the hyperpane is $x_1 - x_2 - x_3 = 0$

b.

$$f(x_1, x_2, x_3) \text{ outputs 0 when } x_1 = 1, x_2 = 1, x_3 = 1, \text{ and 1 for all other examples(at least one 0)}$$

Lets use another method and set w = [-1,-1,-1]

Cases

Negative output

$$w \cdot x + b \leq 0$$

$$w \cdot x + b = -1 \cdot 1 - 1 \cdot 1 - 1 \cdot 1 + b \leq 0 = -3 + b \leq 0 = b \leq 3$$

We know the bias is at most 3

Conclusion

Thus, a possible answer is w=[-1,-1,-1], b =3, and hyperplane: $-x_1 - x_2 - x_3 + 3 = 0$ or $x_1 + x_2 + x_3 = 3$

c.

$$f(x_1, x_2, x_3, x_4) \text{ outputs 1 when } x_{1 \cup 2} \cap x_{3 \cup 4} \text{ is 1}$$

Because at least 2 values need to be 1 to be positive, we know that $x_i + x_j + b = 0$ where i = 1 or 2 and j = 3 or 4

Let w = [1,1,1,1] , $w \cdot x + b = 0$ is then at least $1 \cdot x_i + 1 \cdot x_j + b = 0, 2 + b = 0, b = -2$

Thus a solution is w =[1,1,1,1], b = -2, and the hyperplane: $x_1 + x_2 + x_3 + x_4 - 2 = 0$

d.

$$f(x_1, x_2) \text{ outputs 1 when } (x_1 = x_2 = 1) \cup (x_1 = x_2 = 0)$$

$$f(x_1, x_2) \text{ outputs 0 when } x_1 \neq x_2$$

Cases

Positive

$$x_1 = 0 \cap x_2 = 0, x_1 = 1 \cap x_2 = 1$$

Negative

$$x_1 = 0 \cap x_2 = 1, x_1 = 1 \cap x_2 = 0$$

This is the famous **XOR** problem, which is a dataset that is not linearly seperable

Let's create a new feature z = $x_1 \cdot x_2$

$$f(x1, x2, z) \text{ outputs 1 when } (x_1 = x_2 = x_1 \cdot x_2 = 1) \cup (x_1 = x_2 = x_1 \cdot x_2 = 0), \text{ and outputs 0 otherwise}$$

If f(x1,x2,z) = [0,0,0] then $\mathbf{w} \cdot 0 + b > 0$ to be correctly classified

Working Out the Weights and Bias:

Positive

$$(x_1, x_2, z) = (0,0,0), (1,1,1)$$

$$\text{For } (0,0,0), \quad w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 0 + b > 0 \quad \Rightarrow \quad b > 0$$

$$\text{For } (1,1,1), \quad w_1 \cdot 1 + w_2 \cdot 1 + w_3 \cdot 1 + b > 0 \quad \Rightarrow \quad w_1 + w_2 + w_3 + b > 0$$

Negative

$$(x_1, x_2, z) = (0,1,0), (1,0,0)$$

$$\text{For } (0,1,0), \quad w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 0 + b < 0 \quad \Rightarrow \quad w_2 + b < 0$$

$$\text{For } (1,0,0), \quad w_1 \cdot 1 + w_2 \cdot 0 + w_3 \cdot 0 + b < 0 \quad \Rightarrow \quad w_1 + b < 0$$

$$\text{From } b > 0 \quad \Rightarrow \quad \text{Choose } b = 0.5$$

$$w_1 + b < 0 \quad \Rightarrow \quad w_1 < -0.5 \quad \text{and} \quad w_2 + b < 0 \quad \Rightarrow \quad w_2 < -0.5$$

$$\text{Try choosing } w_1 = w_2 = -1$$

$$\text{Now, solve for } w_3:$$

$$w_1 + w_2 + w_3 + 0.5 > 0 \quad \Rightarrow \quad -1 + (-1) + w_3 + 0.5 > 0 \quad \Rightarrow \quad w_3 > 1.5$$

$$\text{Set } w_3 = 2$$

Conclusion

Thus a solution is w = [-1,-1,-2], b = .5, and the hyperplane: $-x_1 - x_2 + 2z + 0.5 = 0$

Problem 4: Feature Mapping for Inner Product Powers

a.

Given two vectors $x = [x_1, x_2]$ and $y = [y_1, y_2]$, find the feature mapping $\phi(x)$ and $\phi(y)$ such that:

$$(x^\top y)^2 = \phi(x)^\top \phi(y)$$

Feature Mapping:

$$\phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix}, \quad \phi(y) = \begin{bmatrix} y_1^2 \\ \sqrt{2} y_1 y_2 \\ y_2^2 \end{bmatrix}$$

Verification:

$$\phi(x)^\top \phi(y) = x_1^2 y_1^2 + 2 x_1 x_2 y_1 y_2 + x_2^2 y_2^2 = (x^\top y)^2$$

b.

Find the feature mapping for the cubic inner product:

$$(x^\top y)^3 = \phi(x)^\top \phi(y)$$

Feature Mapping:

$$\phi(x) = \begin{bmatrix} x_1^3 \\ \sqrt{3} x_1^2 x_2 \\ \sqrt{3} x_1 x_2^2 \\ x_2^3 \end{bmatrix}, \quad \phi(y) = \begin{bmatrix} y_1^3 \\ \sqrt{3} y_1^2 y_2 \\ \sqrt{3} y_1 y_2^2 \\ y_2^3 \end{bmatrix}$$

Verification:

$$\phi(x)^\top \phi(y) = x_1^3 y_1^3 + 3 x_1^2 x_2 y_1^2 y_2 + 3 x_1 x_2^2 y_1 y_2^2 + x_2^3 y_2^3 = (x^\top y)^3$$

c.

General Case:

For any positive integer k, we need to find the feature mapping such that:

$$(x^\top y)^k = \phi(x)^\top \phi(y)$$

Using the multinomial expansion:

$$(x_1 y_1 + x_2 y_2)^k = \sum_{i+j=k} \binom{k}{i} (x_1^i x_2^{k-i})(y_1^i y_2^{k-i})$$

Feature Mapping:

$$\phi(x) = \left[\sqrt{\binom{k}{i}} \cdot x_1^i x_2^{k-i} \right]_{\forall i+j=k}, \quad \phi(y) = \left[\sqrt{\binom{k}{i}} \cdot y_1^i y_2^{k-i} \right]_{\forall i+j=k}$$

Verification:

$$\phi(x)^\top \phi(y) = \sum_{i+j=k} \binom{k}{i} (x_1^i x_2^{k-i})(y_1^i y_2^{k-i}) = (x^\top y)^k$$

Problem 5: LMS

a.

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (w_1 x_1 + w_2 x_2 + w_3 x_3 + b))^2$$

b.

$$\frac{\nabla J}{\nabla w} = \frac{1}{n} \sum_{i=1}^{n} ((y_i - (w_1 x_1 + w_2 x_2 + w_3 x_3 + b)) \cdot x_i)$$

$$\frac{\nabla J}{\nabla b} = \frac{1}{n} \sum_{i=1}^{n} ((y_i - (w_1 x_1 + w_2 x_2 + w_3 x_3 + b)))$$

$$w = [-1, 1, -1]^T, b = -1 y = w^T x_1 + b$$

$$x_1 = [1, -1, 2], \hat{y}_1 = w^T x_1 + b = -1 - 1 - 2 - 1 = -5 x_2 = [1, 1, 3], \hat{y}_2 = w^T x_2 + b = -1 + 1 - 3 - 1 = -4 x_3 = [-1, 1, 0], \hat{y}_3 = w^T x_3 + b = 1 + 1 - 0 - 1 = 1 x_4 = [1, 2, -4]$$

errors

$$y = [1, 4, -1, -2, 0] \hat{y} = [-5, -4, 1, 4, -4] e_1 = 1 - -5 = 6 e_2 = 4 - -4 = 8 e_3 = -1 - 1 = -2 e_4 = -2 - 4 = -6 e_5 = 0 - -4 = 4$$

gradients

$$\frac{\nabla J}{\nabla w} = -\frac{1}{n} \sum_{i=1}^{n} e_i \cdot xi$$

$$\frac{\nabla J}{\nabla w_1} = -\frac{1}{5}(6(1) + 8(1) + -2(-1) + -6(1) + 4(3) = -\frac{6 + 8 + 2 - 6 + 12}{5} = -\frac{22}{5}$$

$$\frac{\nabla J}{\nabla w_2} = -\frac{1}{5}(6(-1) + 8(1) + -2(1) + -6(2) + 4(-1) = -\frac{-6 + 8 - 2 - 12 - 4}{5} = \frac{16}{5}$$

$$\frac{\nabla J}{\nabla w_3} = -\frac{1}{5}(6(3) + 8(3) + -2(0) + -6(-4) + 4(-1) = -\frac{18 + 24 + 24 - 4}{5} = -\frac{56}{5}$$

$$\frac{\nabla J}{\nabla w} = -\frac{1}{n} \sum_{i=1}^{n} e_i$$

$$\frac{\nabla J}{\nabla b} = -\frac{1}{5}(6 + 8 + -2 + -6 + 4) = -\frac{10}{5} = -2$$

Final

$$\frac{\nabla J}{\nabla w} = [-\frac{22}{5}, \frac{16}{5}, -\frac{56}{5}], \frac{\nabla J}{\nabla b} = -2$$

c.

$$\theta = (X'^T X')^{-1} X'^T y$$

$$X' = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & -1 & 1 & 0 \\ 1 & 1 & 2 & -4 \\ 1 & 3 & -1 & -1 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 1 \\ 4 \\ -1 \\ -2 \\ 0 \end{bmatrix}$$

We first compute:

$$X'^T X' = \begin{bmatrix} 5 & 5 & 2 & 0 \\ 5 & 13 & -2 & -2 \\ 2 & -2 & 8 & -6 \\ 0 & -2 & -6 & 30 \end{bmatrix}$$

and

$$X'^T y = \begin{bmatrix} 2 \\ 4 \\ -2 \\ 22 \end{bmatrix}$$

Using the normal equation:

$$\theta = (X'^T X')^{-1} X'^T y$$

After solving, we get:

$$\theta = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Thus, the optimal parameters are:

$$b^* = -1, \quad w_1^* = 1, \quad w_2^* = 1, \quad w_3^* = 1 \textbf{or} \theta = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

d

Iteration 1

$$x_1 = [1, -1, 2], w = [0, 0, 0]^T, b = 0 \hat{y}_1 = 1(0) + -1(0) + 2(0) + 0 e_1 = 1 - 0 = 1 \frac{\nabla J}{\nabla w_1} = -1(1) = -1 \frac{\nabla J}{\nabla w_2} = -1(-1) = 1 \frac{\nabla J}{\nabla w_3} = -1(2) = -2 \frac{\nabla J}{\nabla b} = -1$$

$$\textbf{Weight Updates: } w_1 = 0 - .1(-1) = .1 w_2 = 0 - .1(1) = -.1 w_3 = 0 - .1(-2) = -.2 b = 0 - .1(-1) = .1$$

#### Iteration 2

$$x_2 = [1, 1, 3], w = [.1, -.1, -.2]^T, b = .1 \hat{y}_2 = 1(.1) + 1(-.1) + 3(.2) + .1 = .7 e_2 = 4 - .7 = 3.3 \frac{\nabla J}{\nabla w_1} = -3.3(1) = -3.3 \frac{\nabla J}{\nabla w_2} = -3.3(1) = -3.3 \frac{\nabla J}{\nabla w_3} = -3.3(3) = -9.9 \frac{\nabla J}{\nabla b} =$$

$$\textbf{Weight Updates: } w_1 = .1 - .1(-3.3) = .43 w_2 = -.1 - .1(-3.3) = .23 w_3 = .2 - .1(-9.9) = 1.19 b = .1 - .1(-3.3) = .43$$

iteration 3

$$x_3 = [-1, 1, 0], w = [.43, .23, 1.19]^T, b = .43 \hat{y}_3 = -1(.43) + 1(.23) + 0(1.19) + .43 = .23 e_3 = -1 - .23 = -1.23 \frac{\nabla J}{\nabla w_1} = -(-1.23)(-1) = -1.23 \frac{\nabla J}{\nabla w_2} = -(-1.23)(1) = 1.2$$

$$\textbf{Weight Updates: } w_1 = .43 - .1(-1.23) = .553 w_2 = .23 - .1(1.23) = .107 w_3 = 1.19 - .1(0) = 1.19 b = .43 - .1(1.23) = .307$$

iteration 4

$$x_4 = [1, 2, -4], w = [.553, .107, 1.19]^T, b = .307 \hat{y}_4 = 1(.553) + 2(.107) + -4(1.19) + .307 = -3.686 e_3 = -2 - -3.686 = 1.686 \frac{\nabla J}{\nabla w_1} = -1.686(1) = -1.686 \frac{\nabla J}{\nabla w_2} = -1.686$$

Processing math: 100%

$\textbf{Weight Updates: } w_1 = .553 - .1(-1.23) = 0.7216 w_2 = .107 - .1(-3.372) = 0.4442 w_3 = 1.19 - .1(6.744) = 0.5156 b = .307 - .1(-1.686) = 0.4756$

iteration 5

$x_4 = [3, -1, -1], w = [.7216, .4442, .5156]^T, b = .4756 \hat{y}_4 = 3(.7216) + -1(.4442) + -1(.5156) + .4756 = 1.6806 e_3 = 0 - 1.6806 = -1.6806 \frac{\nabla J}{\nabla w_1} = -(-1.6806(3)) = 5.041$

$\textbf{Weight Updates: } w_1 = .7216 - .1(5.0418) = 0.21742 w_2 = .4442 - .1(-1.6806) = 0.61226 w_3 = .5156 - .1(-1.6806) = 0.68366 b = .4756 - .1(1.6806) = 0.30754$

Part 2

NOTE

Run the run.sh file in Ensemble Learning to run code for problem 2 and 4

Problem 2

a.

Generally, the test error for the Adaboost is about half of that of from the decision tree from homework 1 and did a lot better at generalization. The training error for the decision tree in hw 1 was practically 0, meaning it was overfitting, and the results of the adaboost do not show that.

b.

The graphs between the bagged trees and single trees look very similar, however the numbers show that the errors for the Adaboost model is on average smaller than the errors for the Decision Tree model.

c:

Yes, Random Forests generally perform better than Bagged Trees because the added randomness in feature selection reduces overfitting and improves generalization. However, 2c takes over 3 hours to run, and is very computationally inefficient, even with parallelism, so I decided not to include it in the .sh as it is a waste of your time and mine to wait that long.

Did not do, takes to long to run

Problem 4

a.

Learned weights: [-0.02228739 0.5541137 0.42924586 0.46526175 1.02221117 0.07198982 1.10686754 0.64512634] Learning Rate 0.5 Test data cost: 0.4598029050984202

see graph for 4a loss over each step

b.

Learning Rate 0.01 Learned weights: [-0.02574642 0.3349877 0.21747145 0.22641376 0.8310834 0.04085715 0.8449247 0.40547703] Test data cost: 0.4783825602905353

see graph for 4b loss over each step

c.

Optimal weight vector (analytical solution): [-0.01519667 0.90056451 0.78629331 0.85104314 1.29889413 0.12989067 1.57224887 0.99869359] Test data cost (analytical solution): 0.4672352895987766

The weights for a and b are practically the same but a little different from part c. Part c’s weights are the most accurate as it solves the system of equations directly. Batch gradient descent and stochastic gradient descent are better for approximating weights on large datasets.

Processing math: 100%