

**Homework #3**

RELEASE DATE: 10/30/2020

RED BUG FIX: 11/04/2020 09:45

BLUE BUG FIX: 11/12/2020 16:30

DUE DATE: 11/20 (THREE WEEKS, YEAH!!), BEFORE 13:00 on Gradescope

QUESTIONS ARE WELCOMED ON THE NTU COOL FORUM.

We will instruct you on how to use Gradescope to upload your choices and your scanned/printed solutions. For problems marked with (\*), please follow the guidelines on the course website and upload your source code to Gradescope as well. You are encouraged to (but not required to) include a README to help the TAs check your source code. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 400 points. For each problem, there is one correct choice. For most of the problems, if you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get  $-10$  points. That is, the expected value of random guessing is  $-20$  per problem, and if you can eliminate two of the choices accurately, the expected value of random guessing on the remaining three choices would be 0 per problem. For other problems, the TAs will check your solution in terms of the written explanations and/or code. The solution will be given points between  $[-20, 20]$  based on how logical your solution is.

**Linear Regression**

1. Consider a noisy target  $y = \mathbf{w}_f^T \mathbf{x} + \epsilon$ , where  $\mathbf{x} \in \mathbb{R}^{d+1}$  (including the added coordinate  $x_0 = 1$ ),  $y \in \mathbb{R}$ ,  $\mathbf{w}_f \in \mathbb{R}^{d+1}$  is an unknown vector, and  $\epsilon$  is an i.i.d. noise term with zero mean and  $\sigma^2$  variance. Assume that we run linear regression on a training data set  $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  generated i.i.d. from some  $P(\mathbf{x})$  and the noise process above, and obtain the weight vector  $\mathbf{w}_{\text{lin}}$ . As briefly discussed in Lecture 9, it can be shown that the expected in-sample error  $E_{\text{in}}(\mathbf{w}_{\text{lin}})$  with respect to  $\mathcal{D}$  is given by:

$$\mathbb{E}_{\mathcal{D}} [E_{\text{in}}(\mathbf{w}_{\text{lin}})] = \sigma^2 \left( 1 - \frac{d+1}{N} \right).$$

For  $\sigma = 0.1$  and  $d = 11$ , what is the smallest number of examples  $N$  such that  $\mathbb{E}_{\mathcal{D}} [E_{\text{in}}(\mathbf{w}_{\text{lin}})]$  is no less than 0.006? Choose the correct answer; explain your answer.

- [a] 25
- [b] 30
- [c] 35
- [d] 40

[e] 45

2. As shown in Lecture 9, minimizing  $E_{\text{in}}(\mathbf{w})$  for linear regression means solving  $\nabla E_{\text{in}}(\mathbf{w}) = 0$ , which in term means solving the so-called *normal equation*

$$\mathbf{X}^T \mathbf{X} \mathbf{w} = \mathbf{X}^T \mathbf{y}.$$

Which of the following statement about the normal equation is correct for any features  $\mathbf{X}$  and labels  $\mathbf{y}$ ? Choose the correct answer; explain your answer.

- [a] There exists at least one solution for the normal equation.
  - [b] If there exists a solution for the normal equation,  $E_{\text{in}}(\mathbf{w}) = 0$  [at such a solution](#).
  - [c] If there exists a *unique* solution for the normal equation,  $E_{\text{in}}(\mathbf{w}) = 0$  [at the solution](#).
  - [d] If  $E_{\text{in}}(\mathbf{w}) = 0$  [at some  \$\mathbf{w}\$](#) , there exists a *unique* solution for the normal equation.
  - [e] none of the other choices
3. In Lecture 9, we introduced the hat matrix  $\mathbf{H} = \mathbf{X}\mathbf{X}^\dagger$  for linear regression. The matrix projects the label vector  $\mathbf{y}$  to the “predicted” vector  $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$  and helps us analyze the error of linear regression. Assume that  $\mathbf{X}^T \mathbf{X}$  is invertible, which makes  $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ . Now, consider the following operations on  $\mathbf{X}$ . Which operation can possibly change  $\mathbf{H}$ ? Choose the correct answer; explain your answer.
- [a] multiplying the whole matrix  $\mathbf{X}$  by 2 (which is equivalent to scaling all input vectors by 2)
  - [b] multiplying each of the  $i$ -th column of  $\mathbf{X}$  by  $i$  (which is equivalent to scaling the  $i$ -th feature by  $i$ )
  - [c] multiplying each of the  $n$ -th row of  $\mathbf{X}$  by  $\frac{1}{n}$  (which is equivalent to scaling the  $n$ -th example by  $\frac{1}{n}$ )
  - [d] adding three randomly-chosen columns  $i, j, k$  to column 1 of  $\mathbf{X}$   
(i.e.,  $x_{n,1} \leftarrow x_{n,1} + x_{n,i} + x_{n,j} + x_{n,k}$ )
  - [e] none of the other choices (i.e. all other choices are guaranteed to keep  $\mathbf{H}$  unchanged.)

## Likelihood and Maximum Likelihood

4. Consider a coin with an unknown head probability  $\theta$ . Independently flip this coin  $N$  times to get  $y_1, y_2, \dots, y_N$ , where  $y_n = 1$  if the  $n$ -th flipping results in head, and 0 otherwise. Define  $\nu = \frac{1}{N} \sum_{n=1}^N y_n$ . How many of the following statements about  $\nu$  are true? Choose the correct answer; explain your answer by illustrating why those statements are true.

- $\Pr(|\nu - \theta| > \epsilon) \leq 2 \exp(-2\epsilon^2 N)$  for all  $N \in \mathbb{N}$  and  $\epsilon > 0$ .
- $\nu$  maximizes  $\text{likelihood}(\hat{\theta})$  over all  $\hat{\theta} \in [0, 1]$ .
- $\nu$  minimizes  $E_{\text{in}}(\hat{y}) = \frac{1}{N} \sum_{n=1}^N (\hat{y} - y_n)^2$  over all  $\hat{y} \in \mathbb{R}$ .
- $2 \cdot \nu$  is the negative gradient direction  $-\nabla E_{\text{in}}(\hat{y})$  at  $\hat{y} = 0$ .

(Note:  $\theta$  is similar to the role of the “target function” and  $\hat{\theta}$  is similar to the role of the “hypothesis” in our machine learning framework.)

- [a] 0
- [b] 1
- [c] 2
- [d] 3
- [e] 4

5. Let  $y_1, y_2, \dots, y_N$  be  $N$  values generated i.i.d. from a uniform distribution  $[0, \theta]$  with some unknown  $\theta$ . For any  $\hat{\theta} \geq \max(y_1, y_2, \dots, y_N)$ , what is its likelihood? Choose the correct answer; explain your answer.

- [a]  $\left(\frac{1}{\hat{\theta}}\right)^N$   
 [b]  $\sum_{n=1}^N \frac{y_n}{\hat{\theta}}$   
 [c]  $\prod_{n=1}^N \frac{y_n}{\hat{\theta}}$   
 [d]  $\frac{\max(y_1, \dots, y_N)}{\hat{\theta}}$   
 [e]  $\frac{\min(y_1, \dots, y_N)}{\hat{\theta}}$

(Hint: Those who are interested in more math [who isn't? :-)] are encouraged to try to derive the maximum-likelihood estimator.)

## Gradient and Stochastic Gradient Descent

6. In the perceptron learning algorithm, we find one example  $(\mathbf{x}_{n(t)}, y_{n(t)})$  that the current weight vector  $\mathbf{w}_t$  mis-classifies, and then update  $\mathbf{w}_t$  by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)} \mathbf{x}_{n(t)}.$$

A variant of the algorithm finds *all* examples  $(\mathbf{x}_n, y_n)$  that the weight vector  $\mathbf{w}_t$  mis-classifies (e.g.  $y_n \neq \text{sign}(\mathbf{w}_t^T \mathbf{x}_n)$ ), and then update  $\mathbf{w}_t$  by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \frac{\eta}{N} \sum_{n: y_n \neq \text{sign}(\mathbf{w}_t^T \mathbf{x}_n)} y_n \mathbf{x}_n.$$

The variant can be viewed as optimizing some  $E_{\text{in}}(\mathbf{w})$  that is composed of one of the following point-wise error functions with a fixed learning rate gradient descent (neglecting any non-differentiable spots of  $E_{\text{in}}$ ). What is the error function? Choose the correct answer; explain your answer.

- [a]  $\text{err}(\mathbf{w}, \mathbf{x}, y) = |1 - y\mathbf{w}^T \mathbf{x}|$   
 [b]  $\text{err}(\mathbf{w}, \mathbf{x}, y) = \max(0, -y\mathbf{w}^T \mathbf{x})$   
 [c]  $\text{err}(\mathbf{w}, \mathbf{x}, y) = -y\mathbf{w}^T \mathbf{x}$   
 [d]  $\text{err}(\mathbf{w}, \mathbf{x}, y) = \min(0, -y\mathbf{w}^T \mathbf{x})$   
 [e]  $\text{err}(\mathbf{w}, \mathbf{x}, y) = \max(0, 1 - y\mathbf{w}^T \mathbf{x})$

7. Besides the error functions introduced in the lectures so far, the following error function, exponential error, is also widely used by some learning models. The exponential error is defined by  $\text{err}_{\text{exp}}(\mathbf{w}, \mathbf{x}, y) = \exp(-y\mathbf{w}^T \mathbf{x})$ . If we want to use stochastic gradient descent to minimize an  $E_{\text{in}}(\mathbf{w})$  that is composed of the error function, which of the following is the update direction  $-\nabla \text{err}_{\text{exp}}(\mathbf{w}, \mathbf{x}_n, y_n)$  for the chosen  $(\mathbf{x}_n, y_n)$  with respect to  $\mathbf{w}_t$ ? Choose the correct answer; explain your answer.

- [a]  $+y_n \mathbf{x}_n \exp(-y_n \mathbf{w}^T \mathbf{x}_n)$   
 [b]  $-y_n \mathbf{x}_n \exp(-y_n \mathbf{w}^T \mathbf{x}_n)$   
 [c]  $+\mathbf{x}_n \exp(-y_n \mathbf{w}^T \mathbf{x}_n)$   
 [d]  $-\mathbf{x}_n \exp(-y_n \mathbf{w}^T \mathbf{x}_n)$   
 [e] none of the other choices

## Hessian and Newton Method

8. Let  $E(\mathbf{w}): \mathbb{R}^d \rightarrow \mathbb{R}$  be a function. Denote the gradient  $\mathbf{b}_E(\mathbf{w})$  and the Hessian  $\mathbf{A}_E(\mathbf{w})$  by

$$\mathbf{b}_E(\mathbf{w}) = \nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_1}(\mathbf{w}) \\ \frac{\partial E}{\partial w_2}(\mathbf{w}) \\ \vdots \\ \frac{\partial E}{\partial w_d}(\mathbf{w}) \end{bmatrix}_{d \times 1} \quad \text{and} \quad \mathbf{A}_E(\mathbf{w}) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_1 \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_d}(\mathbf{w}) \\ \frac{\partial^2 E}{\partial w_2 \partial w_1}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_2^2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_2 \partial w_d}(\mathbf{w}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_d \partial w_1}(\mathbf{w}) & \frac{\partial^2 E}{\partial w_d \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 E}{\partial w_d^2}(\mathbf{w}) \end{bmatrix}_{d \times d}.$$

Then, the second-order Taylor's expansion of  $E(\mathbf{w})$  around  $\mathbf{u}$  is:

$$E(\mathbf{w}) \approx E(\mathbf{u}) + \mathbf{b}_E(\mathbf{u})^T (\mathbf{w} - \mathbf{u}) + \frac{1}{2} (\mathbf{w} - \mathbf{u})^T \mathbf{A}_E(\mathbf{u}) (\mathbf{w} - \mathbf{u}).$$

Suppose  $\mathbf{A}_E(\mathbf{u})$  is positive definite. What is the optimal direction  $\mathbf{v}$  such that  $\mathbf{w} \leftarrow \mathbf{u} + \mathbf{v}$  minimizes the right-hand-side of the Taylor's expansion above? Choose the correct answer; explain your answer. (Note that iterative optimization with  $\mathbf{v}$  is generally called Newton's method.)

- [a]  $+(\mathbf{A}_E(\mathbf{u}))^{-1} \mathbf{b}_E(\mathbf{u})$
  - [b]  $-(\mathbf{A}_E(\mathbf{u}))^{-1} \mathbf{b}_E(\mathbf{u})$
  - [c]  $+(\mathbf{A}_E(\mathbf{u}))^{+1} \mathbf{b}_E(\mathbf{u})$
  - [d]  $-(\mathbf{A}_E(\mathbf{u}))^{+1} \mathbf{b}_E(\mathbf{u})$
  - [e] none of the other choices
9. Following the previous problem, considering minimizing  $E_{\text{in}}(\mathbf{w})$  in linear regression problem with Newton's method. For any given  $\mathbf{w}_t$ , what is the Hessian  $\mathbf{A}_E(\mathbf{w}_t)$  with  $E = E_{\text{in}}$ ? Choose the correct answer; explain your answer.

- [a]  $\frac{2}{N} \mathbf{X}^T \mathbf{X} \mathbf{w}_t \mathbf{w}_t^T$
- [b]  $\frac{2}{N} \mathbf{X}^T \mathbf{X}$
- [c]  $\frac{2}{N} \mathbf{X} \mathbf{X}^T$
- [d]  $\frac{2}{N} \mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}$
- [e] none of the other choices

## Multinomial Logistic Regression

- 10.** In Lecture 11, we solve multiclass classification by OVA or OVO decompositions. One alternative to deal with multiclass classification is to extend the original logistic regression model to Multinomial Logistic Regression (MLR). For a  $K$ -class classification problem, we will denote the output space  $\mathcal{Y} = \{1, 2, \dots, K\}$ . The hypotheses considered by MLR can be indexed by a matrix

$$W = \begin{bmatrix} | & | & \cdots & | & \cdots & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_k & \cdots & \mathbf{w}_K \\ | & | & \cdots & | & \cdots & | \end{bmatrix}_{(d+1) \times K},$$

that contains weight vectors  $(\mathbf{w}_1, \dots, \mathbf{w}_K)$ , each of length  $d+1$ . The matrix represents a hypothesis

$$h_y(\mathbf{x}) = \frac{\exp(\mathbf{w}_y^T \mathbf{x})}{\sum_{i=1}^K \exp(\mathbf{w}_i^T \mathbf{x})}$$

that can be used to approximate the target distribution  $P(y|\mathbf{x})$  for any  $(\mathbf{x}, y)$ . MLR then seeks for the maximum likelihood solution over all such hypotheses. For a given data set  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  generated i.i.d. from some  $P(\mathbf{x})$  and target distribution  $P(y|\mathbf{x})$ , the likelihood of  $h_y(\mathbf{x})$  is proportional to  $\prod_{n=1}^N h_{y_n}(\mathbf{x}_n)$ . That is, minimizing the negative log likelihood is equivalent to minimizing an  $E_{\text{in}}(W)$  that is composed of the following error function

$$\text{err}(W, \mathbf{x}, y) = -\ln h_y(\mathbf{x}) = -\sum_{k=1}^K \mathbb{I}[y = k] \ln h_k(\mathbf{x}).$$

When minimizing  $E_{\text{in}}(W)$  with SGD, we need to compute  $\frac{\partial \text{err}(W, \mathbf{x}, y)}{\partial W_{ik}}$ . What is the value of the partial derivative? Choose the correct answer; explain your answer.

- [a]  $(h_k(\mathbf{x}) + \mathbb{I}[y = k])x_i$
- [b]  $(h_k(\mathbf{x}) - \mathbb{I}[y = k])x_i$
- [c]  $(-h_k(\mathbf{x}) + \mathbb{I}[y = k])x_i$
- [d]  $(-h_k(\mathbf{x}) - \mathbb{I}[y = k])x_i$
- [e] none of the other choices

- 11.** Following the previous problem, consider a data set with  $K = 2$  and obtain the optimal solution from MLR as  $(\mathbf{w}_1^*, \mathbf{w}_2^*)$ . Now, relabel the same data set by replacing  $y_n$  with  $y'_n = 2y_n - 3$  to form a binary classification data set. Which of the following is an optimal solution for logistic regression on the binary classification data set? Choose the correct answer; explain your answer.

- [a]  $\mathbf{w}_2^* + \mathbf{w}_1^*$
- [b]  $\mathbf{w}_1^* - \mathbf{w}_2^*$
- [c]  $\frac{1}{2}(\mathbf{w}_2^* - \mathbf{w}_1^*)$
- [d]  $2(\mathbf{w}_1^* - \mathbf{w}_2^*)$
- [e]  $\mathbf{w}_2^* - \mathbf{w}_1^*$

## Nonlinear Transformation

12. Given the following training data set:

$$\begin{aligned} \mathbf{x}_1 &= (0, 1), y_1 = -1 & \mathbf{x}_2 &= (1, -0.5), y_2 = -1 & \mathbf{x}_3 &= (-1, 0), y_3 = -1 \\ \mathbf{x}_4 &= (-1, 2), y_4 = +1 & \mathbf{x}_5 &= (2, 0), y_5 = +1 & \mathbf{x}_6 &= (1, -1.5), y_6 = +1 & \mathbf{x}_7 &= (0, -2), y_7 = +1 \end{aligned}$$

Using the quadratic transform  $\Phi_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$ , which of the following weights  $\tilde{\mathbf{w}}^T$  in the  $\mathcal{Z}$ -space can separate all of the training data correctly? Choose the correct answer; (no, you don't need to explain your answer :-)).

- [a]  $[-9, -1, 0, 2, -2, 3]$
- [b]  $[-5, -1, 2, 3, -7, 2]$
- [c]  $[9, -1, 4, 2, -2, 3]$
- [d]  $[2, 1, -4, -2, 7, -4]$
- [e]  $[-7, 0, 0, 2, -2, 3]$

13. Consider the following feature transform, which maps  $\mathbf{x} \in \mathbb{R}^d$  to  $\mathbf{z} \in \mathbb{R}^{1+1}$ , keeping only the  $k$ -th coordinate of  $\mathbf{x}$ :  $\Phi_{(k)}(\mathbf{x}) = (1, x_k)$ . Let  $\mathcal{H}_k$  be the set of hypothesis that couples  $\Phi_{(k)}$  with perceptrons. Among the following choices, which of is the tightest upper bound of  $d_{\text{vc}}\left(\bigcup_{k=1}^d \mathcal{H}_k\right)$  for  $d \geq 4$ ? Choose the correct answer; explain your answer. (Hint: You can use the fact that  $\log_2 d \leq \frac{d}{2}$  for  $d \geq 4$  if needed.)

- [a]  $2((\log_2 d) + 1)$
- [b]  $2((\log_2 d) + 1)$
- [c]  $2((d \log_2 d) + 1)$
- [d]  $2(d + 1)$
- [e]  $2(d^2 + 1)$

## Experiments with Linear and Nonlinear Models

Next, we will play with linear regression, logistic regression, non-linear transform, and their use for binary classification. Please use the following set for training:

[https://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw3/hw3\\_train.dat](https://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw3/hw3_train.dat)

and the following set for testing (estimating  $E_{\text{out}}$ ):

[https://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw3/hw3\\_test.dat](https://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw3/hw3_test.dat)

Each line of the data set contains one  $(\mathbf{x}_n, y_n)$  with  $\mathbf{x}_n \in \mathbb{R}^{10}$ . The first 10 numbers of the line contains the components of  $\mathbf{x}_n$  orderly, the last number is  $y_n$ , which belongs to  $\{-1, +1\} \subseteq \mathbb{R}$ . That is, we can use those  $y_n$  for either binary classification or regression.

14. (\*) Add  $x_{n,0} = 1$  to each  $\mathbf{x}_n$ . Then, implement the linear regression algorithm on page 11 of Lecture 9. What is  $E_{\text{in}}^{\text{sqf}}(\mathbf{w}_{\text{lin}})$ , where  $E_{\text{in}}^{\text{sqf}}$  denotes the averaged squared error over  $N$  examples? Choose the closest answer; provide your code.

- [a] 0.00
- [b] 0.20
- [c] 0.40
- [d] 0.60
- [e] 0.80

15. (\*) Add  $x_{n,0} = 1$  to each  $\mathbf{x}_n$ . Then, implement the SGD algorithm for linear regression using the results on pages 10 and 12 of Lecture 11. Pick one example uniformly at random in each iteration, take  $\eta = 0.001$  and initialize  $\mathbf{w}$  with  $\mathbf{w}_0 = \mathbf{0}$ . Run the algorithm until  $E_{\text{in}}^{\text{sqr}}(\mathbf{w}_t) \leq 1.01E_{\text{in}}^{\text{sqr}}(\mathbf{w}_{\text{lin}})$ , and record the total number of iterations taken. Repeat the experiment 1000 times, each with a different random seed. What is the average number of iterations over the 1000 experiments? Choose the closest answer; provide your code.
- [a] 600
  - [b] 1200
  - [c] 1800
  - [d] 2400
  - [e] 3000
16. (\*) Add  $x_{n,0} = 1$  to each  $\mathbf{x}_n$ . Then, implement the SGD algorithm for logistic regression by replacing the SGD update step in the previous problem with the one on page 10 of Lecture 11. Pick one example uniformly at random in each iteration, take  $\eta = 0.001$  and initialize  $\mathbf{w}$  with  $\mathbf{w}_0 = \mathbf{0}$ . Run the algorithm for 500 iterations. Repeat the experiment 1000 times, each with a different random seed. What is the average  $E_{\text{in}}^{\text{ce}}(\mathbf{w}_{500})$  over the 1000 experiments, where  $E_{\text{in}}^{\text{ce}}$  denotes the *averaged* cross-entropy error over  $N$  examples? Choose the closest answer; provide your code.
- [a] 0.44
  - [b] 0.50
  - [c] 0.56
  - [d] 0.62
  - [e] 0.68
17. (\*) Repeat the previous problem, but with  $\mathbf{w}$  initialized by  $\mathbf{w}_0 = \mathbf{w}_{\text{lin}}$  of Problem 14 instead. Repeat the experiment 1000 times, each with a different random seed. What is the average  $E_{\text{in}}^{\text{ce}}(\mathbf{w}_{500})$  over the 1000 experiments? Choose the closest answer; provide your code.
- [a] 0.44
  - [b] 0.50
  - [c] 0.56
  - [d] 0.62
  - [e] 0.68
18. (\*) Following Problem 14, what is  $\left|E_{\text{in}}^{0/1}(\mathbf{w}_{\text{lin}}) - E_{\text{out}}^{0/1}(\mathbf{w}_{\text{lin}})\right|$ , where  $0/1$  denotes the 0/1 error (i.e. using  $\mathbf{w}_{\text{lin}}$  for binary classification), and  $E_{\text{out}}^{(0/1)}$  is estimated using the test set provided above? Choose the closest answer; provide your code.
- [a] 0.32
  - [b] 0.36
  - [c] 0.40
  - [d] 0.44
  - [e] 0.48

- 19.** (\*) Next, consider the following *homogeneous* order- $Q$  polynomial transform

$$\Phi(\mathbf{x}) = (1, x_1, x_2, \dots, x_{10}, x_1^2, x_2^2, \dots, x_{10}^2, \dots, x_1^Q, x_2^Q, \dots, x_{10}^Q).$$

Transform the training and testing data according to  $\Phi(\mathbf{x})$  with  $Q = 3$ , and again implement the linear regression algorithm on page 11 of lecture 9. What is  $\left|E_{\text{in}}^{0/1}(g) - E_{\text{out}}^{0/1}(g)\right|$ , where  $g$  is the hypothesis returned by the transform + linear regression procedure? Choose the closest answer; provide your code.

- [a] 0.32
- [b] 0.36
- [c] 0.40
- [d] 0.44
- [e] 0.48

- 20.** (\*) Repeat the previous problem, but with  $Q = 10$  instead. What is  $\left|E_{\text{in}}^{0/1}(g) - E_{\text{out}}^{0/1}(g)\right|$ ? Choose the closest answer; provide your code.

- [a] 0.32
- [b] 0.36
- [c] 0.40
- [d] 0.44
- [e] 0.48