Homework #3

RELEASE DATE: 04/07/2023

DUE DATE: 04/27/2023, BEFORE 13:00 on Gradescope

QUESTIONS ARE WELCOMED ON DISCORD.

You will 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. 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 20 problems and a total of 500 points. For each problem, there is one correct choice. If you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get 0 points. For five of the secretly-selected problems, the TAs will grade your detailed solution in terms of the written explanations and/or code based on how logical/clear your solution is. Each of the five problems graded by the TAs counts as additional 20 points (in addition to the correct/incorrect choices you made). In general, each homework (except homework 0) is of a total of 500 points.

Linear Models and More

1. If some algorithm always takes a total CPU time of aN for training a binary classifier on a size-N binary classification data set. Consider a size-N K-class classification data set where each class is of size N/K. What is the total CPU time needed for training a K-class classifier via one-versus-one decomposition on the data set (ignoring the minor time needed for re-labeling the data set for the sub-problems)? Choose the correct answer; explain your choice.

[a] aN

[b] $a(K-1)N$

[c] aKN

[d] $a\binom{K}{2}N$

[e] none of the other choices

2. Consider six inputs $\mathbf{x}_1 = (2, 0)$, $\mathbf{x}_2 = (0, 2)$, $\mathbf{x}_3 = (-2, 0)$, $\mathbf{x}_4 = (0, -2)$, $\mathbf{x}_5 = (0, 0)$, $\mathbf{x}_6 = (1, 1)$. What is the biggest subset of those input vectors that can be shattered by the union of quadratic, linear, or constant hypotheses of $\mathbf{x}$? You can take $\text{sign}(0) = +1$ if needed. Choose the correct answer; explain your choice.

[a] $\mathbf{x}_1$, $\mathbf{x}_3$

[b] $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$

[c] $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$

[d] $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$, $\mathbf{x}_5$

[e] $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$, $\mathbf{x}_5$, $\mathbf{x}_6$

3. Consider the following data set:

$$\mathbf{x}_1 = (0,0), y_1 = -1 \qquad \mathbf{x}_2 = (4,0), y_2 = +1 \qquad \mathbf{x}_3 = (-4,0), y_3 = +1$$
$$\mathbf{x}_4 = (0,2), y_4 = -1 \qquad \mathbf{x}_5 = (0,-2), y_5 = -1$$

You can easily check that the data set is not linearly separable. Now, consider using a second order polynomial transformation $\mathbf{\Phi}(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$. Take $\text{sign}(0) = +1$ if needed. How many of the following weight vectors represent a linear classifier in the $\mathcal{Z}$-space that separates the five examples above after transformation? Choose the correct answer; explain your choice.

$$\begin{aligned}\mathbf{w}_1 &= (-1, 0, 0, 0.5, 0, -0.5)\\ \mathbf{w}_2 &= (-1, 0, 0, -0.5, 0, 0.5)\\ \mathbf{w}_3 &= (-2, 0, 0, 1, 0, 1)\\ \mathbf{w}_4 &= (-1, 0, 0, 0.2, 0, 0.1)\end{aligned}$$

[a] 0

[b] 1

[c] 2

[d] 3

[e] 4

4. Consider a feature transform $\mathbf{\Phi}(\mathbf{x}) = \Gamma\mathbf{x}$ where Γ is a $(d+1)$ by $(d+1)$ invertible matrix. For a training data set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, run linear regression on the original data set, and get $\mathbf{w}_{\text{lin}}$ while achieving the lowest $E_{\text{in}}(\mathbf{w}_{\text{lin}})$ on the original data. Then, run linear regression on the $\mathbf{\Phi}$-transformed data, and get $\tilde{\mathbf{w}}$ while achieving the lowest $E_{\text{in}}(\tilde{\mathbf{w}})$ on the transformed data. For simplicity, assume that the matrix X (with every row being $\mathbf{x}_n^T$) satisfies that X^TX is invertible. What are the relationships between and $\mathbf{w}_{\text{lin}}$ and $\tilde{\mathbf{w}}$? Choose the correct answer; explain your choice.

[a] $\mathbf{w}_{\text{lin}} = \Gamma^T\tilde{\mathbf{w}}, E_{\text{in}}(\tilde{\mathbf{w}}) = \text{trace}(\Gamma^T\Gamma) \cdot E_{\text{in}}(\mathbf{w}_{\text{lin}})$

[b] $\mathbf{w}_{\text{lin}} = \Gamma^T\tilde{\mathbf{w}}, E_{\text{in}}(\tilde{\mathbf{w}}) = E_{\text{in}}(\mathbf{w}_{\text{lin}})$

[c] $\mathbf{w}_{\text{lin}} = (\Gamma^T)^{-1}\tilde{\mathbf{w}}, E_{\text{in}}(\tilde{\mathbf{w}}) = \text{trace}(\Gamma^T\Gamma) \cdot E_{\text{in}}(\mathbf{w}_{\text{lin}})$

[d] $\mathbf{w}_{\text{lin}} = (\Gamma^T)^{-1}\tilde{\mathbf{w}}, E_{\text{in}}(\tilde{\mathbf{w}}) = E_{\text{in}}(\mathbf{w}_{\text{lin}})$

[e] none of the other choices

5. 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 $\mathbf{x}$: $\mathbf{\Phi}_{(k)}(\mathbf{x}) = (1, x_k)$. Let $\mathcal{H}_k$ be the set of hypothesis that couples $\mathbf{\Phi}_{(k)}$ with perceptrons. Yes, that is our old friend "decision stumps" in multiple dimensions. Among the following choices, which 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; prove your choice. *Hint: For any single $\mathcal{H}_k$, the growth function is $2N$. You can also use the fact that $\log_2 d \leq \frac{d}{2}$ for $d \geq 4$ if needed.*

[a] $2((\log_2 \log_2 d) + 1)$

[b] $2((\log_2 d) + 1)$

[c] $2((\log_2 d)^2 + 1)$

[d] $2(d+1)$

[e] $2((d \log_2 d) + 1)$

6. Assume that a transformer (no, not chat-Generative-Pretrained-Transformer!) peeks the data and decides the following transform $\mathbf{\Phi}$ "intelligently" from the data of size N. The transform maps $\mathbf{x} \in \mathbb{R}^d$ to $\mathbf{z} \in \mathbb{R}^N$, where

$$(\mathbf{\Phi}(\mathbf{x}))_n = z_n = [\![\mathbf{x} = \mathbf{x}_n]\!] .$$

For simplicity, assume that all $\mathbf{x}_n$ are different. Consider a learning algorithm that performs linear regression after the feature transform (for simplicity, please exclude $z_0 = 1$) to get a $g(\mathbf{x}) = \tilde{\mathbf{w}}^T\mathbf{\Phi}(\mathbf{x})$. Which of the following is not always true? Choose the incorrect statement; explain your choice.

[**a**] $\tilde{w}_n = y_n$

[**b**] $E_{\text{in}}(g) = 0$

[**c**] $g(2\mathbf{x}_n) = 2y_n$

[**d**] $g(\mathbf{x}) = 0$ on those $\mathbf{x} \neq \mathbf{x}_n$ for any n

[**e**] none of the other choices

Playing with Regularization

7. Given the following one-dimensional data set with $N = 3$ examples:

$$\begin{gathered}(x_1 = 2, y_1 = 1)\\(x_2 = 3, y_2 = 0)\\(x_3 = -2, y_3 = 2) \quad .\end{gathered}$$

Compute an L1-regularized hypothesis of the form $h(x) = w_0 + w_1 x$ by solving

$$\min_{\mathbf{w}\in\mathbb{R}^{1+1}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{3}\|\mathbf{w}\|_1$$

with the averaged squared error as $E_{\text{in}}(\mathbf{w})$ and $\lambda = 3$. What is the minimum value of $E_{\text{aug}}(\mathbf{w})$? Choose the closest value; explain your choice.

[**a**] 0.5

[**b**] 0.7

[**c**] 0.9

[**d**] 1.1

[**e**] 1.3

8. Given the following one-dimensional data set with $N = 2$ examples: $(x_1 = 2, y_1 = 9)$ and $(x_2 = -2, y_2 = -1)$. Compute an L2-regularized hypothesis of the form $h(x) = w_0 + w_1 x$ by solving

$$\min_{\mathbf{w}\in\mathbb{R}^{1+1}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\|\mathbf{w}\|_2^2$$

with the averaged squared error as $E_{\text{in}}(\mathbf{w})$. Then, run the resulting $\mathbf{w}_{\text{reg}}$ on a test example $(x = 1, y = 4)$. Which value of $\lambda > 0$ would make $y = \mathbf{w}_{\text{reg}}^T \begin{bmatrix} 1 \\ x \end{bmatrix}$? Choose the closest value; explain your choice.

[**a**] 0

[**b**] 2

[**c**] 4

[**d**] 6

[**e**] 8

9. Consider L2-regularization with a general $E_{\text{in}}(\mathbf{w})$, which comes with an augmented error of

$$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

with $\lambda > 0$. Optimize the augmented error using fixed-learning-rate gradient descent with learning rate η. That is, update the weight vector by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta\nabla E_{\text{aug}}(\mathbf{w}_t)$$

The update rule is equivalent to

$$\mathbf{w}_{t+1} \leftarrow \alpha\mathbf{w}_t - \eta\nabla E_{\text{in}}(\mathbf{w}_t).$$

with some α. What is the formula of α? Choose the correct answer; prove your choice.

[**a**] $1 - \frac{\eta\lambda}{N}$

[**b**] $1 - \frac{2\eta\lambda}{N}$

[**c**] $1 - \frac{\eta\lambda}{2N}$

[**d**] $1 - \frac{\eta}{N}$

[**e**] none of the other choices

Virtual Examples and Regularization

10. Consider linear regression with virtual examples. That is, we add K virtual examples

$$(\tilde{\mathbf{x}}_1, \tilde{y}_1), (\tilde{\mathbf{x}}_2, \tilde{y}_2), \ldots, (\tilde{\mathbf{x}}_K, \tilde{y}_K)$$

to the training data set, and solve

$$\min_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N+K}\left(\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \sum_{k=1}^{K}(\tilde{y}_k - \mathbf{w}^T\tilde{\mathbf{x}}_k)^2\right).$$

Let $\tilde{\mathrm{X}} = [\tilde{\mathbf{x}}_1\tilde{\mathbf{x}}_2\ldots\tilde{\mathbf{x}}_K]^T$, and $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_K]^T$. Also, let I_{d+1} denote a $(d+1)$ by $(d+1)$ identity matrix. For what $\tilde{\mathrm{X}}$ and $\tilde{\mathbf{y}}$ will the solution of this linear regression be equal to

$$\mathbf{w}_{\text{reg}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N}\|\mathrm{X}\mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{N}\|\mathbf{w}\|^2?$$

Choose the correct answer; prove your choice.

[**a**] $\tilde{\mathrm{X}} = \lambda\mathrm{I}_{d+1}$, $\tilde{\mathbf{y}} = \mathbf{0}$

[**b**] $\tilde{\mathrm{X}} = \sqrt{\lambda}\mathrm{I}_{d+1}$, $\tilde{\mathbf{y}} = \mathbf{0}$

[**c**] $\tilde{\mathrm{X}} = \lambda\mathrm{I}_{d+1}$, $\tilde{\mathbf{y}} = \mathbf{y}$

[**d**] $\tilde{\mathrm{X}} = \sqrt{\lambda}\mathrm{I}_{d+1}$, $\tilde{\mathbf{y}} = \mathbf{y}$

[**e**] none of the other choices

11. We discussed about adding "virtual examples" (hints) to help combat overfitting. One way of generating virtual examples is to add a small noise to the input vector $\mathbf{x} \in \mathbb{R}^{d+1}$ (including the 0-th component x_0) For each $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$ in our training data set, assume that we generate virtual examples $(\tilde{\mathbf{x}}_1, y_1), (\tilde{\mathbf{x}}_2, y_2), \ldots, (\tilde{\mathbf{x}}_N, y_N)$ where $\tilde{\mathbf{x}}_n$ is simply $\mathbf{x}_n + \boldsymbol{\epsilon}$ and the noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^{d+1}$ is generated i.i.d. from a multivariate uniform distribution between $[-r, r]$ in each of the $(d+1)$ dimensions. Recall that when training the linear regression model, we need to calculate $\mathrm{X}^T\mathrm{X}$ first. Define the hinted input matrix

$$\mathrm{X}_h = \begin{bmatrix} | & \ldots & | & | & \ldots & | \\ \mathbf{x}_1 & \ldots & \mathbf{x}_N & \tilde{\mathbf{x}}_1 & \ldots & \tilde{\mathbf{x}}_N \\ | & \ldots & | & | & \ldots & | \end{bmatrix}^T.$$

What is the expected value $\mathbb{E}(\mathrm{X}_h^T\mathrm{X}_h)$, where the expectation is taken over the (uniform)-noise generating process above? Choose the correct answer; prove your choice.

[a] $\mathrm{X}^T\mathrm{X} + \frac{1}{3}r^2\mathrm{I}_{d+1}$

[b] $2\mathrm{X}^T\mathrm{X} + r^2\mathrm{I}_{d+1}$

[c] $2\mathrm{X}^T\mathrm{X} + \frac{N}{3}r^2\mathrm{I}_{d+1}$

[d] $2\mathrm{X}^T\mathrm{X} + Nr^2\mathrm{I}_{d+1}$

[e] none of the other choices

(*Note: The choices here "hint" you that the expected value is related to the matrix being inverted for regularized linear regression. That is, data hinting "by noise" is closely related to regularization. If* $\mathbf{x}$ *contains the pixels of an image, the virtual example is a noise-contaminated image with the same label. Adding some noise is a very common technique to generate virtual examples for images.*)

12. Additive smoothing (https://en.wikipedia.org/wiki/Additive_smoothing) is a simple yet useful technique in estimating discrete probabilities. Consider the technique for estimating the head probability of a coin. Let $y_1, y_2, \ldots, y_N$ denotes the flip results from a coin, with $y_n = 1$ meaning a head and $y_n = 0$ meaning a tail. Additive smoothing adds $2K$ "virtual flips", with K of them being head and the other K being tail. Then, the head probability is estimated by

$$\frac{(\sum_{n=1}^N y_n) + K}{N + 2K}$$

The estimate can be viewed as the optimal solution of

$$\min_{y \in \mathbb{R}} \frac{1}{N} \sum_{n=1}^{N} (y - y_n)^2 + \frac{\lambda}{N}\Omega(y),$$

where $\Omega(y)$ is a "regularizer" to this estimation problem. What is $\Omega(y)$? Choose the correct answer; prove your choice.

[a] $\frac{2K}{\lambda}(y + 0.5)^2$

[b] $\frac{2K}{\lambda}(y - 0.5)^2$

[c] $\frac{K}{\lambda}(y + 0.5)^2$

[d] $\frac{K}{\lambda}(y - 0.5)^2$

[e] none of the other choices

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/ml23spring/hw3/hw3_train.dat

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

https://www.csie.ntu.edu.tw/~htlin/course/ml23spring/hw3/hw3_test.dat

The files are of the same format as the one you received in Homework 1.

13. (*) Add $x_{n,0} = 1$ to each $\mathbf{x}_n$. Then, implement the linear regression algorithm on page 14 of Lecture 5. What is $E_{\text{in}}^{\text{sqr}}(\mathbf{w}_{\text{lin}})$, where $E_{\text{in}}^{\text{sqr}}$ denotes the *averaged* squared error over N examples? Choose the closest answer; provide your code.

- [a] 0.76
- [b] 0.78
- [c] 0.80
- [d] 0.82
- [e] 0.84

14. (*) Add $x_{n,0} = 1$ to each $\mathbf{x}_n$. Then, implement the SGD algorithm for linear regression using a similar derivation like page 49 of Lecture 5. 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 800 iterations. Repeat the experiment 1000 times, each with a different random seed. What is the average $E_{\text{in}}^{\text{sqr}}(\mathbf{w}_{800})$ over the 1000 experiments, where $E_{\text{in}}^{\text{sqr}}$ denotes the *averaged* squared error over N examples? Choose the closest answer; provide your code.

- [a] 0.76
- [b] 0.78
- [c] 0.80
- [d] 0.82
- [e] 0.84

15. (*) 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 49 of Lecture 5. 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 800 iterations. Repeat the experiment 1000 times, each with a different random seed. What is the average $E_{\text{in}}^{\text{ce}}(\mathbf{w}_{800})$ 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] 0.60
- [b] 0.63
- [c] 0.66
- [d] 0.69
- [e] 0.72

16. (*) Repeat the previous problem, but with $\mathbf{w}$ initialized by $\mathbf{w}_0 = \mathbf{w}_{\text{lin}}$ of Problem 13 instead. Repeat the experiment 1000 times, each with a different random seed. What is the average $E_{\text{in}}^{\text{ce}}(\mathbf{w}_{800})$ over the 1000 experiments? Choose the closest answer; provide your code.

[a] 0.60
[b] 0.63
[c] 0.66
[d] 0.69
[e] 0.72

17. (*) Following the previous problem, what is the average $\left|E_{\text{in}}^{0/1}(\mathbf{w}_{800}) - E_{\text{out}}^{0/1}(\mathbf{w}_{800})\right|$ over 1000 experiments, where $E_{\text{in}}^{0/1}$ denotes the *averaged* 0/1 error (i.e. using $\mathbf{w}_{800}$ 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] 0.03
[b] 0.04
[c] 0.05
[d] 0.06
[e] 0.07

18. (*) Following Problem 13, what is $\left|E_{\text{in}}^{0/1}(\mathbf{w}_{\text{lin}}) - E_{\text{out}}^{0/1}(\mathbf{w}_{\text{lin}})\right|$, where $E_{\text{in}}^{0/1}$ denotes the *averaged* 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] 0.03
[b] 0.04
[c] 0.05
[d] 0.06
[e] 0.07

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

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

Transform the training and testing data according to $\mathbf{\Phi}(\mathbf{x})$ with $Q = 2$, and again implement the linear regression algorithm like Problem 13. 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] 0.06
[b] 0.07
[c] 0.08
[d] 0.09
[e] 0.10

20. (*) Repeat the previous problem, but with $Q = 8$ 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] 0.1
[b] 0.2
[c] 0.3
[d] 0.4
[e] 0.5