Homework #3

RELEASE DATE: 10/12/2023

RED CORRECTION: 10/26/2023 06:00

YOU GET TWO MORE GOLD MEDALS THIS SEMESTER BECAUSE OF THE DELAYED RELEASE. YEAH!!

DUE DATE: 11/01/2023, BEFORE 13:00 on GRADESCOPE

QUESTIONS ARE WELCOMED ON DISCORD (INFORMALLY) OR NTU COOL (FORMALLY).

You will use Gradescope to upload 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 240 points and 20 bonus points. In general, every homework set would come with a full credit of 240 points, with some possible bonus points.

Noise and Error

1. (20 points) Consider a binary classification problem, where $\mathcal{Y} = \{-1, +1\}$. Assume a noisy scenario where the data is generated i.i.d. from some $P(\mathbf{x}, y)$. In class, we discussed that when the 0/1 error function (i.e. classification error) is considered, calculating the "ideal mini-target" on each $\mathbf{x}$ reveals the hidden target function of

$$f_{0/1}(\mathbf{x}) = \text{argmax}_{y \in \{-1,+1\}} P(y|\mathbf{x}) = \text{sign}\left(P(+1|\mathbf{x}) - \frac{1}{2}\right).$$

Instead of the 0/1 error, if we consider the CIA error function, where a false positive (classifying a negative example as a positive one) is 1000 times more important than a false negative, the hidden target should be changed to

$$f_{\text{CIA}}(\mathbf{x}) = \text{sign}\big(P(+1|\mathbf{x}) - \alpha\big).$$

Prove what the value of α should be.

2. (20 points) Consider a binary classification task, where God gives you some noiseless data i.i.d. from an unknown distribution $P(\mathbf{x})$ and an unknown target function $f(\mathbf{x})$ that maps from $\mathcal{X}$ to $\{-1, +1\}$. After you use the data to obtain some $g(\mathbf{x})$ that suffers

$$\begin{aligned} E_{\text{out}}(g) &= \mathcal{E}_{\mathbf{x} \sim P(\mathbf{x})} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!] \text{ (here } \mathcal{E} \text{ means expectation, as shown in class slides)} \\ &= \mathbb{E}_{\mathbf{x} \sim P(\mathbf{x})} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!] \text{ (or if you like the more beautiful font } \mathbb{E} \text{ for expectation).} \end{aligned}$$

Now, assume that $g(\mathbf{x})$ is put in a noisy test environment where

$$\begin{aligned} P(y = +f(\mathbf{x})|\mathbf{x}) &= 1 - \epsilon \\ P(y = -f(\mathbf{x})|\mathbf{x}) &= \epsilon. \end{aligned}$$

Derive

$$\mathbb{E}_{(\mathbf{x},y)\sim P(\mathbf{x},y)}\,[\![g(\mathbf{x}) \neq y]\!]$$

as a function of $E_{\text{out}}(g)$ and ϵ.

Linear Regression

3. (20 points) Consider a hypothesis set that contains hypotheses of the form $h(x) = wx$ for $x \in \mathbb{R}$. Combine the hypothesis set with the squared error function to minimize

$$E_{\text{in}}(w) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

on a given data set $\{(x_n, y_n)\}_{n=1}^N$. Derive the optimal w_{LIN} in terms of (x_n, y_n) and express the result *without* using matrix/vector notations. You can assume all denominators to be non-zero. *(Hint: This is linear regression in $\mathbb{R}$ without the added x_0.)*

4. (20 points) Consider the target function $f(x) = ax^2 + b$. Sample x uniformly from $[0, 1]$, and use all linear hypotheses $h(x) = w_0 + w_1 \cdot x$ to approximate the target function with respect to the squared error. For any given (a, b), derive the weights (w_0^*, w_1^*) of the optimal hypothesis as a function of (a, b).

5. (20 points) Consider running linear regression on $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n$ includes the constant dimension $x_0 = 1$ as usual. For simplicity, you can assume that $\mathrm{X}^T\mathrm{X}$ is invertible. Assume that the unique (why :-)) solution $\mathbf{w}_{\text{LIN}}$ is obtained after running linear regression on the data above. Now, consider an output transformation

$$y'_n = ay_n + b.$$

for some given constants (a, b). Run linear regression on $\{(\mathbf{x}_n, y'_n)\}_{n=1}^N$ to obtain the unique solution $\mathbf{w}'_{\text{LIN}}$. Derive $\mathbf{w}'_{\text{LIN}}$ as a function of $\mathbf{w}_{\text{LIN}}$ and (a, b).

More on Linear Models

6. (20 points) Let $E(\mathbf{w})\colon \mathbb{R}^d \to \mathbb{R}$ be a function. Denote the gradient $\mathbf{b}_E(\mathbf{w})$ and the Hessian $\mathrm{A}_E(\mathbf{w})$ by

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

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

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

Suppose $\mathrm{A}_E(\mathbf{u})$ is positive definite. 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 is simply $-(\mathrm{A}_E(\mathbf{u}))^{-1}\mathbf{b}_E(\mathbf{u})$.

Hint: Homework 0! :-)

An iterative optimization algorithm using the "optimal direction" above for updating $\mathbf{w}$ is called **Newton's method**, which can be viewed as "improving" gradient descent by using more information about E. Now, consider minimizing $E_{\text{in}}(\mathbf{w})$ in logistic regression problem with Newton's method on a data set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ with the cross-entropy error function for E_{in}:

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\ln(1 + \exp(-y_n\mathbf{w}^T\mathbf{x}_n)).$$

For any given $\mathbf{w}_t$, let

$$h_t(\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}_t^T \mathbf{x})}.$$

Express the Hessian $\mathrm{A}_E(\mathbf{w}_t)$ with $E = E_{\text{in}}$ as $\mathrm{X}^T\mathrm{DX}$, where D is an N by N diagonal matrix. Derive what D should be in terms of h_t, $\mathbf{w}_t$, $\mathbf{x}_n$, and y_n.

Note: The h_t that we use here is slightly different from that being used in class—this particular h_t predicts $P(-1|\mathbf{x})$ instead of $P(+1|\mathbf{x})$. It was a harmless typo (sorry!!). By default, the TAs will grade by this definition of h_t. But if you really want to choose to use the in-class definition, you are allowed to do so ***as long as you state your choice clearly for the TAs****.*

7. (20 points) The truncated squared loss

$$\mathrm{err}(s, y) = (\max(0, 1 - ys))^2$$

can be easily shown to be an upper bound on the 0/1 error. Assume that s is generated from a linear scoring function $s = \mathbf{w}^T\mathbf{x}$ like Page 3/25 of Lecture 11. Derive a "perceptron learning algorithm" by applying SGD on the truncated squared loss. Compare the resulting algorithm with the original PLA. Discuss the similarities and differences using 5 to 10 sentences.

Multinomial Logistic Regression

8. 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 $\mathcal{Y} = \{1, 2, \cdots, K\}$. The hypotheses considered by MLR can be indexed by a matrix

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

that contains weight vectors $(\mathbf{w}_1, \cdots, \mathbf{w}_K)$, 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 $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ 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 $E_{\text{in}}(\mathrm{W})$ that is composed of the following error function

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

Consider minimizing $E_{\text{in}}(\mathrm{W}) = \frac{1}{N}\sum_{n=1}^N \mathrm{err}(\mathrm{W}, \mathbf{x}_n, y_n)$ with gradient descent. Derive $\nabla E_{\text{in}}(\mathrm{W})$. Your result should simply be a matrix with the same size as W. *(Note: the hypothesis that transforms the scores $\{\mathbf{w}_i^T\mathbf{x}\}_{i=1}^K$ to $h_y(\mathbf{x})$ is often called a softmax function in (multiclass) deep learning.)*

Experiments with Linear Models

Next, we will play with linear regression, logistic regression, and their use for binary classification. We will also learn how their different robustness to outliers. We will generate artificial $2D$ data for training and testing in the next problems. Each example $(\mathbf{x}, y)$ is assumed to be generated from the following process:

- Flip a fair coin to get either $y = +1$ or $y = -1$.

- If $y = +1$, generate $\mathbf{x} = (1, x_1, x_2)$ where (x_1, x_2) comes from a normal distribution of mean $[3, 2]$ and covariance $\begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}$.
- If $y = -1$, generate $\mathbf{x} = (1, x_1, x_2)$ where (x_1, x_2) comes from a normal distribution of mean $[5, 0]$ and covariance $\begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$.

Please generate $N = 256$ examples from the process as your training data set $\mathcal{D}$. Then, generate 4096 more examples from the process as your test data set (for evaluating E_{out}).

Hint: Be sure to check whether your normal distribution function needs you to provide the variance, which would be like 0.4 *for the* $y_n = +1$ *cases, or the standard deviation, which would be like* $\sqrt{0.4}$.

9. (20 points, *) Implement the linear regression algorithm taught in the lecture. Run the algorithm for 128 times, each with a different random seed for generating the two data sets above. Plot a histogram to visualize the distribution of $E_{\text{in}}^{\text{sqr}}(\mathbf{w}_{\text{LIN}})$, where $E_{\text{in}}^{\text{sqr}}$ denotes the *averaged* squared error over N examples. What is the median $E_{\text{in}}^{\text{sqr}}$ over the 128 experiments?

10. (20 points, *) Following the previous problem, plot a histogram to visualize the distribution of $E_{\text{in}}^{0/1}(\mathbf{w}_{\text{LIN}})$, where $E_{\text{in}}^{0/1}$ denotes the averaged 0/1 error over N examples (i.e. using $\mathbf{w}_{\text{LIN}}$ for binary classification). What is the median $E_{\text{in}}^{0/1}$ over the 128 experiments?

(Note: You can choose to run 128 *new experiments in this problem, or just re-use the* 128 *hypotheses* $\mathbf{w}_{\text{LIN}}$ *and test data sets obtained from the previous problem.)*

11. (20 points, *) Consider two algorithms. The first one, $\mathcal{A}$, is the linear regression algorithm above. The second one $\mathcal{B}$ is logistic regression, trained with fixed learning rate gradient descent with $\eta = 0.1$ for $T = 500$ iterations, starting from $\mathbf{w}_0 = \mathbf{0}$. Run the algorithms on the same $\mathcal{D}$, and record $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D})), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))]$. Repeat the process for 128 times, each with a different random seed for generating the training and test data sets above. Plot a scatter plot for $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D})), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))]$. What is the median of $E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}))$ and what is the median of $E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}))$?

12. (20 points, *) Following the previous problem, in addition to the 256 examples in $\mathcal{D}$, add 16 outlier examples generated from the following process to your training data (but not to your test data). All outlier examples will be labeled $y = +1$ and $\mathbf{x} = [1, x_1, x_2]$ where (x_1, x_2) comes from a normal distribution of mean $[0, 6]$ and covariance $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.3 \end{bmatrix}$. Name the new training data set $\mathcal{D}'$. Run the algorithms on the same $\mathcal{D}'$, and record $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}')), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))]$. Repeat the process for 128 times, each with a different random seed for generating the training and test data sets above. Plot a scatter plot for $[E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}')), E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))]$. What is the median of $E_{\text{out}}^{0/1}(\mathcal{A}(\mathcal{D}'))$ and what is the median of $E_{\text{out}}^{0/1}(\mathcal{B}(\mathcal{D}'))$? Compare your results to the previous problem. Describe your findings.

Bonus: Logistic Regression and Linear Regression

13. (Bonus 20 points) When using Newton's method for solving logistic regression, as discussed in Problem 6, each update $\mathbf{v}$ is calculated by

$$\mathbf{v} = -(\mathrm{X}^T \mathrm{D} \mathrm{X})^{-1} \nabla E_{\text{in}}(\mathbf{w}_t)$$

when $(\mathrm{X}^T \mathrm{D} \mathrm{X})$ is invertible. In linear regression, when $\mathrm{X}^T \mathrm{X}$ is invertible, the optimal

$$\mathbf{w}_{\text{LIN}} = (\mathrm{X}^T \mathrm{X})^{-1} \mathrm{X}^T \mathbf{y}.$$

If we can express $-\nabla E_{\text{in}}(\mathbf{w}_t)$ as some $\tilde{\mathrm{X}}^T \tilde{\mathbf{y}}$, and $\mathrm{X}^T \mathrm{D} \mathrm{X}$ as some $\tilde{\mathrm{X}}^T \tilde{\mathrm{X}}$, then each iteration of Newton-solving logistic regression is performing an internal linear regression! State the internal linear regression problem—in particular, what are $\tilde{\mathrm{X}}$ and $\tilde{\mathbf{y}}$?