Homework #7

RELEASE DATE: 12/02/2024

DUE DATE: 12/16/2024, BEFORE 13:00 on GRADESCOPE

QUESTIONS ARE WELCOMED ON DISCORD (INFORMALLY) OR VIA EMAILS (FORMALLY).

You will use Gradescope to upload your scanned/printed solutions. 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 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 200 points and 20 bonus points. In general, every homework set would come with a full credit of 200 points, with some possible bonus points.

Experimentally, we will allow each grading TA to give up to 2 clarity-bonus points for every human-graded problem if the TA believes that the answer is delivered with exceptional clarity, in addition to being correct. We hope that this encourages everyone to think about how to express your ideas clearly.

1. (10 points, auto-graded) Suppose we have a data set of size $N = 1126$, and we use bootstrapping to sample N' examples. Within the choices, what is the minimum N' such that the probability of getting at least one duplicated example (with number of copies ≥ 2) is larger than 70%? Choose the correct answer.

[a] 40

[b] 45

[c] 50

[d] 55

[e] 60

2. (10 points, auto-graded) Impurity functions play an important role in decision tree branching. For binary classification problems, let μ_+ be the fraction of positive examples in a data subset, and $\mu_- = 1 - \mu_+$ be the fraction of negative examples in the data subset. We can normalize each impurity function by dividing it with its maximum value among all $\mu_+ \in [0, 1]$. For instance, the classification error is simply $\min(\mu_+, \mu_-)$ and its maximum value is 0.5. So the normalized classification error is $2\min(\mu_+, \mu_-)$. After normalizing the following impurity function, which one is equivalent to the Gini index $1 - \mu_+^2 - \mu_-^2$ (after normalization)? Choose the correct answer.

[a] the classification error $\min(\mu_+, \mu_-)$

[b] the squared error (when used for branching in classification data sets), which is by definition $\mu_+(1 - (\mu_+ - \mu_-))^2 + \mu_-(-1 - (\mu_+ - \mu_-))^2$

[c] the entropy, which is $-\mu_+ \ln \mu_+ - \mu_- \ln \mu_-$, with $0 \ln 0 \equiv 0$

[d] the closeness, which is $1 - |\mu_+ - \mu_-|$

[e] none of the other choices

3. (10 points, auto-graded) Consider applying the AdaBoost algorithm on Page 17 of Lecture 208 to a binary classification data set where 87% of the examples are negative. Because there are so many negative examples, the base algorithm within AdaBoost returns a constant classifier $g_1(\mathbf{x}) = -1$ in the first iteration. Let $u_+^{(2)}$ be the individual example weight of each positive example in the second iteration, and $u_-^{(2)}$ be the example weight of each negative example in the second iteration. What is $\frac{u_+^{(2)}}{u_-^{(2)}}$? Choose the correct answer.

[a] 13/87

[b] 74/87

[c] 1

[d] 87/74

[e] 87/13

4. (10 points, auto-graded) Consider a Neural Network with $d^{(0)} + 1 = 20$ input units, 3 output units, and 50 hidden units (each $x_0^{(\ell)}$ is also counted as a unit). The hidden units can be arranged in any number of layers $\ell = 1, ..., L-1$. That is,

$$\sum_{\ell=1}^{L-1} \left(d^{(\ell)} + 1 \right) = 50.$$

Each layer is fully connected to the layer above it. What is the maximum possible number of weights that such a network can have? Choose the correct answer.

[a] 875

[b] 1179

[c] 1219

[d] 1327

[e] 1130

5. (20 points, human-graded) Consider an aggregation binary classifier G constructed by uniform blending on $2M+1$ binary classifiers $\{g_t\}_{t=1}^{2M+1}$, where M is a positive integer. That is,

$$G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{2M+1} g_t(\mathbf{x})\right).$$

Assume that each g_t is of test 0/1 error $E_{\text{out}}(g_t) = e_t$. Derive the tightest upper bound of the test 0/1 error $E_{\text{out}}(G)$ as a function of M and $\{e_t\}_{t=1}^{2M+1}$.

6. (20 points, human-graded) For the AdaBoost algorithm introduced in Lectures 208 and 211, let $U_t = \sum_{n=1}^{N} u_n^{(t)}$. That is, $U_1 = 1$ (you are very welcome! ;-)). Assume that $0 < \epsilon_t < \frac{1}{2}$ for each hypothesis g_t. Prove that $\frac{U_{t+1}}{U_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

(Hint: The result is the backbone of proving that AdaBoost will converge within $O(\log N)$ iterations.)

7. (20 points, human-graded) If gradient boosting is coupled with linear regression (without regularization) instead of decision trees. Prove or disprove that the optimal $\alpha_1 = 1$.

8. (20 points, human-graded) For the gradient boosted decision tree, after updating all s_n in iteration t using the steepest η as α_t, prove that $\sum_{n=1}^{N}(y_n - s_n)g_t(\mathbf{x}_n) = 0$.

(Note: This special value may tell us some important physical property on the relationship between the vectors $[y_1 - s_1, y_2 - s_2, \ldots, y_N - s_N]^T$ and $[g_t(\mathbf{x}_1), g_t(\mathbf{x}_2), \ldots, g_t(\mathbf{x}_N)]^T$.

9. (20 points, human-graded) For a Neural Network with one hidden layer and $\tanh(s)$ as the transformation functions on all neurons **including the output neuron**, prove that for the backprop algorithm (with mini-batch gradient descent), when all the initial weights $w_{i,j}^{(\ell)}$ are set to 0.5, then $w_{i,j}^{(1)} = w_{i,j+1}^{(1)}$ for all i and $1 \le j < d^{(1)}$.

10. (20 points, human-graded) For Problems 10-12, implement the AdaBoost-Stump algorithm as introduced in Lecture 208. Run the algorithm on the following set for training:

`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/madelon`

and the following set for testing:

`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/madelon.t`

Use a total of $T = 500$ iterations (please do not stop earlier than 500), and calculate E_{in} and E_{out} with the 0/1 error.

For the decision stump algorithm, please extend your implementation in homework 2 to the multi-dimensional version (Problem 13 of homework 2) that takes example weights into account. Any algorithm that minimizes $E_{\text{in}}^{\mathbf{u}}$ is acceptable, but the simplest approach would be to obtain the best decision stump for each dimension using your implementation in homework 2 (with example weights), and then choose the "best of the best" across all dimensions. Plot $E_{\text{in}}(g_t)$ (0/1 error) and ϵ_t (the normalized $E_{\text{in}}^{\mathbf{u}^{(t)}}(g_t)$) as a function of t for $t = 1, 2, \ldots, 500$ on the same figure. Describe your findings. Please include screenshots of the first page of your AdaBoost-Stump code. We require the code snapshot for this problem only, but not the following two problems, given that they are similar in nature.

11. (20 points, human-graded) Define $G_t(\mathbf{x}) = \text{sign}\left(\sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x})\right)$. Plot $E_{\text{in}}(G_t)$ and $E_{\text{out}}(G_t)$ as a function of t for $t = 1, 2, \ldots, 500$ on the same figure. Describe your findings.

12. (20 points, human-graded) Plot U_t defined in Problem 6 and $E_{\text{in}}(G_t)$ as a function of t for $t = 1, 2, \ldots, 500$ on the same figure. Describe your findings.

13. (Bonus 20 points, human-graded) The following chatGPT answer says that it is possible to implement $\texttt{XOR}\big((x)_1, (x)_2, \ldots, (x_d)\big)$ with a d-$(d-1)$-1 feed-forward neural network with $\text{sign}(s)$ as the transformation function (usually called linear-threshold units).

`https://chatgpt.com/share/67482991-8544-8002-9c9c-c41f77e5c004`

But our 2023 fall bonus homework asked the students to prove that it is impossible to implement $\texttt{XOR}\big((x)_1, (x)_2, \ldots, (x_d)\big)$ with any d-$(d-1)$-1 feed-forward neural network with $\text{sign}(s)$ as the transformation function. Point out where chatGPT's argument diverges from our 2023 fall bonus homework, and prove the impossibility mathematically.