Homework #3

RELEASE DATE: 04/30/2019

DUE DATE: 05/21/2019, BEFORE 14:00 ON GRADESCOPE

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE FACEBOOK FORUM.

Please upload your solutions (without the source code) to Gradescope as instructed.

For problems marked with (*), please follow the guidelines on the course website and upload your source code to CEIBA. 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 160 points and 20 bonus points. In general, every homework set would come with a full credit of 160 points, with some possible bonus points.

Decision Tree

Impurity functions play an important role in decision tree branching. For multi-class classification problems, let $\mu_1, \mu_2, \ldots, \mu_K$ be the fraction of each class of examples in a data subset, where each $\mu_k \geq 0$ and $\sum_{k=1}^{K} \mu_k = 1$.

1. The Gini impurity is $1 - \sum_{k=1}^{K} \mu_k^2$. What is the maximum value of the Gini impurity among all possible $[\mu_1, \mu_2, \ldots, \mu_K$ that satisfies $\mu_k \geq 0$ and $\sum_{k=1}^{K} \mu_k = 1$? Prove your answer.

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.

2. Prove or disprove that the squared regression error when using binary classification, which is by definition $\mu_+(1-(\mu_+-\mu_-))^2 + \mu_-(-1-(\mu_+-\mu_-))^2$ is simply a scaled version of the Gini impurity $1 - \mu_+^2 - \mu_-^2$.

Random Forest

3. If bootstrapping is used to sample $N' = pN$ examples out of N examples and N is very large, argue that approximately $e^{-p} \cdot N$ of the examples will not be sampled at all.

4. Consider a Random Forest G that consists of K binary classification trees $\{g_k\}_{k=1}^{K}$, where K is an odd integer. Each g_k is of test 0/1 error $E_{\text{out}}(g_k) = e_k$. Prove or disprove that $\frac{2}{K+1} \sum_{k=1}^{K} e_k$ upper bounds $E_{\text{out}}(G)$.

Gradient Boosting

5. For the gradient boosted decision tree (with squared error), if a tree with only one constant node is returned as g_1, and if $g_1(\mathbf{x}) = 11.26$, then after the first iteration, all s_n is updated from 0 to a new constant $\alpha_1 g_1(\mathbf{x}_n) = 11.26\alpha_1$. What is α_1 in terms of all the $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$? Prove your answer.

6. For the gradient boosted decision tree (with squared error), after updating all s_n in iteration t using the steepest η as α_t, what is the value of $\sum_{n=1}^{N} s_n g_t(\mathbf{x}_n)$? Prove your answer.

7. If gradient boosting (with squared error) is coupled with squared-error polynomial regression (without regularization) instead of decision trees. Prove or disprove that the optimal $\alpha_1 = 1$.

Neural Network

8. Consider Neural Network with sign(s) instead of tanh(s) as the transformation functions. That is, consider Multi-Layer Perceptrons. In addition, we will take +1 to mean logic TRUE, and −1 to mean logic FALSE. Assume that all x_i below are either +1 or −1. Write down the weights w_i for the following perceptron

$$g_A(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i\right).$$

to implement

$$\texttt{OR}\,(x_1, x_2, \ldots, x_d)\,.$$

Explain your answer.

9. For a Neural Network with at least one hidden layer and tanh(s) as the transformation functions on all neurons (including the output neuron), when all the initial weights $w_{ij}^{(\ell)}$ are set to 0, what gradient components are also 0? Justify your answer.

10. Multiclass Neural Network of K classes is typically done by having K output neurons in the last layer. For some given example $(\mathbf{x}, y)$, let $s_k^{(L)}$ be the summed input score to the k-th neuron, the joint "softmax" output vector is defined as

$$\mathbf{x}^{(L)} = \left[\frac{\exp(s_1^{(L)})}{\sum_{k=1}^{K}\exp(s_k^{(L)})}, \frac{\exp(s_2^{(L)})}{\sum_{k=1}^{K}\exp(s_k^{(L)})}, \ldots, \frac{\exp(s_K^{(L)})}{\sum_{k=1}^{K}\exp(s_k^{(L)})}\right].$$

It is easy to see that each $x_k^{(L)}$ is between 0 and 1 and the the components of the whole vector sum to 1. That is, $\mathbf{x}^{(L)}$ defines a probability distribution. Let's rename $\mathbf{x}^{(L)} = \mathbf{q}$ for short.

Define a one-hot-encoded vector of y to be

$$\mathbf{v} = [[\![y = 1]\!], [\![y = 2]\!], \ldots, [\![y = K]\!]].$$

The cross-entropy loss function for the Multiclass Neural Network, much like an extension of the cross-entropy loss function used in logistic regression, is defined as

$$e = -\sum_{k=1}^{K} v_k \ln q_k.$$

Prove that $\frac{\partial e}{\partial s_k^{(L)}} = q_k - v_k$ which is actually the $\delta_k^{(L)}$ that you'd need for backprop.

Experiments with Decision Trees

Implement the simple C&RT algorithm without pruning using the Gini impurity as the impurity measure as introduced in the class. You need to implement the algorithm by yourself without using sophisticated pacakges. For the decision stump used in branching, if you are branching with feature i and direction s, please sort all the $x_{n,i}$ values to form (at most) $N+1$ segments of equivalent θ, and then pick θ within the median of the segment.

Run the algorithm on the following set for training:

`hw3_train.dat`

and the following set for testing:

`hw3_test.dat`

11. (*) Draw the resulting tree (by program or by hand, in any way easily understandable by the TAs).

12. (*) Continuing from the previous problem, what is E_{in} and E_{out} (evaluated with 0/1 error) of the tree?

13. (*) Assume that the tree in the previous question is of height H. Try a simple pruning technique of restricting the maximum tree height to $H-1$, $H-2$, ..., 1 by terminating (returning a leave) whenever a node is at the maximum tree height. Call g_h the pruned decision tree with maximum tree height h. Plot curves of h versus $E_{\text{in}}(g_h)$ and h versus $E_{\text{out}}(g_h)$ using the 0/1 error in the same figure. Describe your findings.

Now implement the Bagging algorithm with $N' = 0.8N$ and couple it with your fully-grown (without-pruning) decision tree above to make a preliminary random forest G_{RF}. Produce $T = 30000$ trees with bagging. Compute E_{in} and E_{out} using the 0/1 error.

14. (*) Plot a histogram of $E_{\text{in}}(g_t)$ over the 30000 trees.

15. (*) Let G_t = "the random forest with the first t trees". Plot a curve of t versus $E_{\text{in}}(G_t)$.

16. (*) Continuing from Question 15, and plot a curve of t versus $E_{\text{out}}(G_t)$. Briefly compare with the curve in Question 15 and state your findings.

Bonus: Crazy XOR

17. (10%) Construct a d-d-1 feed-forward neural network with $\text{sign}(s)$ as the transformation function (such a neural network is also called a Linear Threshold Circuit) to implement $\texttt{XOR}\big((x)_1, (x)_2, \ldots, (x_d)\big)$.

18. (10%) 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.