Homework #3

RELEASE DATE: 05/31/2018

DUE DATE: 06/26/2018, BEFORE 14:00

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE FACEBOOK FORUM.

Unless granted by the instructor in advance, you must turn in a printed/written copy of your solutions (without the source code) for all problems.

For problems marked with (*), please follow the guidelines on the course website and upload your source code to designated places. 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 40 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 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.

1. The Gini index is $1 - \mu_+^2 - \mu_-^2$. What is the maximum value of the Gini index among all $\mu_+ \in [0, 1]$? Prove your answer.

2. Following the previous question, there are four possible impurity functions below. 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 normalization, which of the following impurity function is equivalent to the normalized Gini index? Prove your answer.

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

[**b**] the squared regression error (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 $0 \log 0 \equiv 0$.

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

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

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, if a tree with only one constant node is returned as g_1, and if $g_1(\mathbf{x}) = 2$, then after the first iteration, all s_n is updated from 0 to a new constant $\alpha_1 g_1(\mathbf{x}_n)$. What is s_n in terms of all the $\{(\mathbf{x}_m, y_m)\}_{m=1}^N$? Prove your answer.

6. For the gradient boosted decision tree, 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 is coupled with linear regression (without regularization) instead of decision trees. Prove or disprove that the optimal $g_2(\mathbf{x}) = 0$.

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 AdaBoost

For Questions 11–16, implement the AdaBoost-Stump algorithm as introduced in Lecture 208. Run the algorithm on the following set for training:

`hw3_train.dat`

and the following set for testing:

`hw3_test.dat`

Use a total of $T = 300$ iterations (please do not stop earlier than 300), and calculate E_{in} and E_{out} with the 0/1 error.

For the decision stump algorithm, please implement the following steps. Any ties can be arbitrarily broken.

(1) For any feature i, sort all the $x_{n,i}$ values to $x_{[n],i}$ such that $x_{[n],i} \le x_{[n+1],i}$.

(2) Consider thresholds within $-\infty$ and all the midpoints $\frac{x_{[n],i}+x_{[n+1],i}}{2}$. Test those thresholds with $s \in \{-1, +1\}$ to determine the best (s, θ) combination that minimizes E_{in}^{u} using feature i.

(3) Pick the best (s, i, θ) combination by enumerating over all possible i.

For those interested, step 2 can be carried out in $O(N)$ time only!!

11. (*) Plot a figure for t versus $E_{\text{in}}(g_t)$. What is $E_{\text{in}}(g_1)$ and what is α_1?

12. From the figure in the previous question, should $E_{\text{in}}(g_t)$ be decreasing or increasing? Write down your observations and explanations.

13. (*) Plot a figure for t versus $E_{\text{in}}(G_t)$, where $G_t(\mathbf{x}) = \text{sign}(\sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x}))$. That is, $G = G_T$. What is $E_{\text{in}}(G)$?

14. (*) Plot a figure for t versus U_t, where $U_t = \sum_{n=1}^{N} u_n^{(t)}$. What is U_2 and what is U_T?

15. (*) Plot a figure for t versus $E_{\text{out}}(g_t)$ estimated with the test set. What is $E_{\text{out}}(g_1)$?

16. (*) Plot a figure for t versus $E_{\text{out}}(G_t)$ estimated with the test set. What is $E_{\text{out}}(G)$?

Power of Adaptive Boosting

Next, we will prove that AdaBoost can reach $E_{\text{in}}(G_T) = 0$ if T is large enough and every hypothesis g_t satisfies $\epsilon_t \le \epsilon < \frac{1}{2}$. Let U_t be defined as in Question 14. It can be proved (see Lecture 11 of Machine Learning Techniques) that

$$U_{t+1} = \frac{1}{N} \sum_{n=1}^{N} \exp\left(-y_n \sum_{\tau=1}^{t} \alpha_\tau g_\tau(\mathbf{x}_n) \right).$$

and $E_{\text{in}}(G_T) \le U_{T+1}$.

17. (Bonus, 20 points) Prove that $U_1 = 1$ and $U_{t+1} = U_t \cdot 2\sqrt{\epsilon_t(1-\epsilon_t)} \le U_t \cdot 2\sqrt{\epsilon(1-\epsilon)}$.

18. (Bonus, 20 points) Using the fact that $\sqrt{\epsilon(1-\epsilon)} \le \frac{1}{2}\exp\left(-2(\frac{1}{2}-\epsilon)^2\right)$ for $\epsilon < \frac{1}{2}$, argue that after $T = O(\log N)$ iterations, $E_{\text{in}}(G_T) = 0$.