Homework #3

RELEASE DATE: 05/02/2017

DUE DATE: 05/23/2017, 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.

Boosting

1. Assume that linear regression (for classification) is used within AdaBoost. That is, we need to solve the weighted-E_{in} optimization problem.

$$\min_{\mathbf{w}} E_{\text{in}}^{\mathbf{u}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} u_n(y_n - \mathbf{w}^T\mathbf{x}_n)^2.$$

The optimization problem above is equivalent to minimizing the usual E_{in} of linear regression on some "pseudo data" $\{(\tilde{\mathbf{x}}_n, \tilde{y}_n)\}_{n=1}^{N}$. Write down your pseudo data $(\tilde{\mathbf{x}}_n, \tilde{y}_n)$ and prove your answer. (*Hint: There is more than one possible form of pseudo data*)

2. Consider applying the AdaBoost algorithm on a binary classification data set where 99% of the examples are positive. Because there are so many positive 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 individual example weight of each negative example in the second iteration. What is $u_+^{(2)}/u_-^{(2)}$? Prove your answer.

Kernel for Decision Stumps

When talking about non-uniform voting in aggregation, we mentioned that $\boldsymbol{\alpha}$ can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

$$\boldsymbol{\phi}(\mathbf{x}) = \Big(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_T(\mathbf{x})\Big).$$

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product $(\boldsymbol{\phi}(\mathbf{x}))^T(\boldsymbol{\phi}(\mathbf{x}'))$. In this problem, we mix the two topics together using the decision stumps as our $g_t(\mathbf{x})$.

3. Assume that the input vectors contain only integers between (including) L and R.

$$\begin{aligned} & g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big), \\ \text{where} \quad & i \in \{1, 2, \cdots, d\}, d \text{ is the finite dimensionality of the input space,} \\ & s \in \{-1, +1\}, \theta \in \mathbb{R}, \text{ and } \text{sign}(0) = +1 \end{aligned}$$

Two decision stumps g and $\hat{g}$ are defined as the *same* if $g(\mathbf{x}) = \hat{g}(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$. Two decision stumps are different if they are not the same. How many different decision stumps are there for the case of $d = 2$, $L = 1$, and $R = 6$? Explain your answer.

4. Continuing from the previous question, let $\mathcal{G} = \{$ all different decision stumps for $\mathcal{X}$ $\}$ and enumerate each hypothesis $g \in \mathcal{G}$ by some index t. Define

$$\boldsymbol{\phi}_{ds}(\mathbf{x}) = \Bigg(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_t(\mathbf{x}), \cdots, g_{|\mathcal{G}|}(\mathbf{x})\Bigg).$$

Derive a simple equation that evaluates $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$ efficiently and prove your answer.

We would give full credit if your solution works for the specific (d, L, R) given by Question 3, and we would give 10 bonus points if your solution works for general (d, L, R). Besides, another 10 bonus points will be awarded if your solution works with the "non-integer" input vectors.

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.

5. 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.

6. Following Question 1, 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

Experiments with Adaptive Boosting.

For Questions 7–13, 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 $x_{[n],i} \leq x_{[n+1],i}$.

(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!!

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

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

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

10. (*) 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?

11. (*) Plot a figure for t versus ϵ_t. What is the minimum value of ϵ_t?

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

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

Experiments with Unpruned Decision Tree

Implement the simple C&RT algorithm without pruning using the Gini index as the impurity measure as introduced in the class. 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`

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

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

16. (*) Try pruning each leaf of the tree above. What is the lowest E_{in} that you can get from pruning one leaf? What is the corresponding E_{out}?

Power of Adaptive Boosting

In this problem, we will prove that AdaBoost can reach $E_{\text{in}}(G_T) = 0$ if T is large enough and every hypothesis g_t satisfies $\epsilon_t \leq \epsilon < \frac{1}{2}$. Let U_t be defined as in Question 10. 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 $E_{\text{in}}(G_T) \leq U_{T+1}$.

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

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