



Boosting a decision stump

6 试题

1
point

1.

Are you using GraphLab Create? Please make sure that

1. You are using version 1.8.3 of GraphLab Create. Verify the version of GraphLab Create by running

```
graphlab.version
```

inside the notebook. If your GraphLab version is incorrect, see this post (<https://www.coursera.org/learn/ml-classification/supplement/LgZ3I/installing-correct-version-of-graphlab-create>) to install version 1.8.3. **This assignment is not guaranteed to work with other versions of GraphLab Create.**

2. You are using the IPython notebook named module-8-boosting-assignment-2-blank.ipynb obtained from the associated reading.

This question is ungraded. Check one of the three options to confirm.

- ☒ I confirm that I am using the right version of GraphLab Create and the right IPython notebook.
- ☐ I am using SFrame and NumPy only.
- ☐ I am using other tools, and I understand that I may not be able to complete some of the quiz questions.

1
point

2.

Recall that the **classification error for unweighted data** is defined as follows:

$$\text{classification error} = \frac{\# \text{ mistakes}}{\# \text{ all data points}}$$

Meanwhile, the **weight of mistakes for weighted data** is given by

$$\text{WM}(\alpha, \hat{y}) = \sum_{i=1}^n \alpha_i \times 1[y_i \neq \hat{y}_i].$$

If we set the weights $\alpha=1$ for all data points, how is the weight of mistakes $\text{WM}(\alpha, \hat{y})$ related to the classification error?

- ☐ $\text{WM}(\alpha, \hat{y}) = [\text{classification error}] * [\text{total weight of all data points}]$
- ☐ $\text{WM}(\alpha, \hat{y}) = [\text{classification error}] * [\text{weight of correctly classified data points}]$
- ☒ $\text{WM}(\alpha, \hat{y}) = N * [\text{classification error}]$
- ☐ $\text{WM}(\alpha, \hat{y}) = 1 - [\text{classification error}]$

1
point

3.

Refer to section **Example: Training a weighted decision tree**.

Will you get the same model as **small_data_decision_tree_subset_20** if you trained a decision tree with only 20 data points from the set of points in **subset_20**?

- ☒ Yes
- ☐ No

1

point

4.

Refer to the 10-component ensemble of tree stumps trained with Adaboost.

As each component is trained sequentially, are the component weights monotonically decreasing, monotonically increasing, or neither?

- ☐ ~~Monotonically decreasing~~
 - ☐ Monotonically increasing
 - ☒ Neither
-

1
point

5.

Which of the following best describes a **general trend in accuracy** as we add more and more components? Answer based on the 30 components learned so far.

- ☐ Training error goes down monotonically, i.e. the training error reduces with each iteration but never increases.
 - ☒ Training error goes down in general, with some ups and downs in the middle.
 - ☐ Training error goes up in general, with some ups and downs in the middle.
 - ☐ Training error goes down in the beginning, achieves the best error, and then goes up sharply.
 - ☐ None of the above
-

1
point

6.

From this plot (with 30 trees), is there massive overfitting as the # of iterations increases?

☐ Yes

☒ No

提交测试

