Cross Validated

What's the purpose of learning rate in sklearn AdaBoost implementation

Asked 3 months ago Active 3 months ago Viewed 99 times

1

We know that `sklearn` 's implemenation of AdaBoost algorithm uses DecisionTreeClassifier as the base learner.

Conceptually, `AdaBoost` assigns equal weights to all examples initially, where the weight is equal to `1/n` . `n` - the number of examples.

But then the AdaBoost documentantion includes a hyperparameter `learning_rate` defined as:

> **learning_rate float, default=1.**
>
> Weight applied to each classifier at each boosting iteration. A higher learning rate increases the contribution of each classifier. There is a trade-off between the learning_rate and n_estimators parameters.

But then Decision Trees do not use `learning_rate` parameter since it not a gradient-based model learning approach. Besides, `AdaBoost` already assigns `1/n` to each sample. How then the definition of `learning_rate` defined in the documentation fits here?

classification boosting supervised-learning ensemble-learning adaboost

Share Cite Edit Follow Flag

asked Jun 14 at 10:58

arilwan

153 6

1 Answer

Active Oldest Votes

2

The learning rate is just applied to each of the tree's predictions and has nothing to do with the tree model itself but the boosting 'meta' algorithm.

Since boosting is iteratively learning from the past model it can overfit so the learning rate is a simple way to control for that. So after each fit that round's predictions is equal to the previous round plus the learning rate multiplied by this iteration's fit. That is the point of the last sentence in the documentation:

> There is a trade-off between the learning_rate and n_estimators parameters.

If you have a lower learning rate such as .1 you could have more estimators since each individual estimator is contributing less.

As a side note I have seen it heavily used much more aggressively in gradient boosting.

Share Cite Edit Follow Flag

answered Jun 14 at 14:01

Tylerr

1,100 5 16