University of Colorado at Colorado Springs

Home Work Assignment 2

Out 03-17-2021, Due 04-05-2021

Preface

In this homework assignment, you will answer questions as asked and write programs as necessary. You can do more if you want for extra credit. Consider the homework as open-ended and as a result, you should do the basics and extend it any way you can. Extra credits will be given for *substantial* additional work. Please read on your own if a topic has not been discussed in class to your satisfaction. *Make sure you have a demo scheduled with me the week the homework is due.* Please note that you will have to keep working on your semester project as you work on this and future homeworks; so please manage time properly.

Introduction

We have studied decision and regression trees in class. Although a tree is easy and fast to build, the predictive power or generalization ability of an individual tree is not high. It is likely to overfit to the training data and not produce good results on unseen or test data. That is why researchers have come up with ways to classify with a collection of trees, not just one. In general, there are two ways to develop such a collection: Bagging and Boosting. Bagging samples the dataset to pick a random subset of the training dataset, and builds a tree from the selected subset. It builds a number of trees in this manner, and these trees make a collective decision on unseen or test examples by following a voting or averaging procedure, weighted or unweighted. A bagged collection of trees, when a small number of features are randomly selected to build each individual tree, is called a Random Forest. The topic of this homework is Boosting Trees that are discussed next.

Boosting Trees

In contrast to bagging, where each tree is grown randomly and independently of each other, boosting trees are grown in sequence, and therefore, there are dependencies among the trees. Each tree is usually quite simple, possibly a constant node or a stump with only one decision node. A subsequent tree is built to address some "weakness" of the previous tree. A number of such trees, possibly several hundred, are constructed. The decision of the entire collection of boosting trees is usually obtained by adding the decisions of individual trees. That is why such trees are called additive. Boosting trees can be built by considering the cumulative loss (or error)[1] incurred by the trees in sequence, each tree attempting to reduce the total error or loss, considering the entire training dataset. Such trees are usually called Forward Stagwise (Loss) Trees.

The first well-known boosting tree algorithm that was proposed is called AdaBoost [1]. We have discussed AdaBoost in class. AdaBoost builds a sequence of simple additive trees for a classification

[1] The cumulative loss, computed from the dataset is called Empirical Loss by some. It is empirical because the dataset is a bunch of observations, i.e., they are empirical as opposed to being something theoretical.

problem. When building a subsequent tree, it attempts to handle better the examples where the previous tree made mistakes. Thus, it builds a sequence of simple trees to locally improve the additive performance. In building the tree that follows immediately, it weighs the mistaken examples from this round, a little higher, and weighs the correctly handled examples a little lower. The trees are also differently weighted. Each tree it builds must perform better than random. For example, if it is a binary classifier, each tree must have accuracy higher than 50%. AdaBoost was designed for binary classification. Researchers have modified it for regression as well as for multi-class classification.

To Do

Boosting Trees are very good classifiers and regressors, often better than Random Forests. In this assignment, you will write programs in a language of your choice that learn to build boosting classification trees, experiment with them, obtain results, analyze them and write a report. In particular, you would do the following.

1. Write a builder for a single-node tree or *stump*. You will have to develop a way to choose a feature to use for splitting, and then decide on the point of split considering this feature. In your report, state how the splitting decision is made.
2. Write the AdaBoost algorithm using an algorithmic style in LaTeX. Discuss this algorithm, paying attention to how local improvements are made as new trees are built, how examples are weighted to improve cumulative performance, and how individual trees are weighted.
3. Write a program to build a sequence of decision trees using AdaBoost. Test with the Pima Indian Diabetes dataset and the Sonar dataset. You can download the~~se~~ dataset~~s~~ from the UCI Machine Learning Repository.
4. Report your results for the two datasets with appropriate metrics. Perform analysis of results.
5. Perform research into how AdaBoost can be extended to perform multi-class classification. Describe such an algorithm. Implement from scratch and test it on two multi-label classification datasets from the UCI Repository. It is possible that you can implement a general multi-class version of AdaBoost and use it for binary classification as well.

Note that you have to implement Boosting Trees from *"scratch."* You can use numerical libraries, but *not* libraries that build Boosting Trees.

Playing with Tools

Learn to use algorithms with single trees and collection of trees in R or Python. Learn how to change parameters for these algorithms in these languages. In particular, learn to run AdaBoost in R or Python. Solve the regression ~~and classification~~ problem~~s~~ you solve above using Boosting Trees in R or Python.

What to Hand in

You will submit a 2-4 page paper with a title and your name. Use the AAAI Author style you have been using for the semester project papers you have been writing for the class. In the paper, you will have a short section with an appropriate heading for each question asked and any extra work you perform. You must have a section called *Results*, in addition to other sections you deem appropriate. Here, you will present your findings in terms of discussions, tables, graphs and any other visuals as you deem appropriate. In the report, *compare* the results you obtain with the Boosting Trees you implement yourself, and the Boosting Trees you build using libraries in both R or Python.

Please keep to the paper length requested above and use the right format. You will be penalized for writing too little or too much, and for not following the format.

References

References

[1] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." icml. Vol. 96. 1996.