Homework #4

B05902120 / Yu-Ting, TSENG

Jan 17, 2018

Problem description

Random Forest is one of the most fundamental algorithms and frequently used models for machine learning. In many real world applications, like bad-guy classifier, the size of the data file can range from hundreds of megabytes to terabytes. As a result, the data to be trained may be too large to read into process memory. How to efficiency train such huge data set is critical to application performance.

In this assignment, your job is to write a forest random model using multiple threads, which reads training data from file, trains the model, and reads testing data from file, predict them and outputs your prediction. The execution time using random forest with parellel programming should be significantly less than with sequential programming.

Data description

Your get two datasets, one is `training_data` and the other is `testing_data`. In `training_data`, we will specify who is a good guy and who is a bad guy. What you should do is construct a random forest by training over `training_data`, catch the bad guys in the `testing_data`, and eventually output to `submission.csv`

The first column means the IDs. The second to the thirty-forth columns tell the features of the ID, in other words, you will have thirty three kinds of features. And the last column specifies whether it is a good guy or a bad guy, 0 for good and 1 for bad (only in `trainging_data`.

Random Forest

Some of you may not be familiar with random forest, we will show how to train a random forest with `training_data` step by step as following:

Training

1. Get `training_dataset` by reading from `training_data`, and note that do not put the IDs into the dataset.
2. Get equivalence number of data randomly.
3. Construct a decision tree by the data you just chose.
4. Repeat Step3 and Step4 for several times and you will have a random forest.

Testing

1. Get `data` by reading from `testing_data`
2. Let the data be input of each decision trees and the tree will tell you whether it is a good guy or a bad guy.
3. Vote to decide the final result by the answer given by each decision trees.

Decision Tree

Some of you may not be familiar with decision tree, we will show you how to construct a decision tree with data step by step as following:

Training

1. Input the data to the root node.
2. Find the best separate point in each dimension.
   The way to find the separate point:
   (a) Sort the data according to the specific dimension.
   (b) Separate the data from the minimum value of the specific dimension and calculate `Gini_Impurity`.
   (c) Find the separate point which has the minimum value of the `Gini_Impurity`.
3. Compare the separate point in each dimension, find the best one and separate it. Record the `threshold` and the dimension. And the put the data smaller than the threshold on the left node and the data larger than the threshold on the right node.

4. Repeat Step2 and Step3 until the labels in the node become all 0s or all 1s.

Testing

1. Get `data` by reading from `testing_data`
2. Go through the tree decided by the threshold and the dimension, and find the label of the bottom node.

Perf

`Perf` is a tool to evaluate the efficiency on Linux. Please type the command below to get the number of the instructions you used from you read the data.

```
perf stat -e instructions:u -v ./random_forest
```

Format for input and output

The request random forest:

```
./random_forest -data data_dir -output submission.csv
                        -tree tree_number -thread thread_number,
```

where `data_dir` is the directory in which `training_data` and `testing_data` is placed.

Tasks and Scoring

There are 6 subtasks in this assignment. By finishing all subtasks you earn the full 7 points.

1. Successfully compile and execute the code and get the result within 3 minutes with the accuracy higher than 80%. (1 point)
2. Thread using. (1 point)
   - Specify where you use threads and how do they share the results.
3. The number of the threads versus the used time. (2 point)
   - Plot a diagram or fill a form over the number of the threads versus the used time, and mark the fastest one.
4. The number of the threads versus the number of the used instructions. (1 point)

- Plot a diagram or fill a form over the number of the thread versus the used instruction.

5. The number of the trees versus the number of the used instructions. (1 point)

    - Plot a diagram or fill a form over the number of the trees versus the used instruction.

6. Others. (1 point)

    - Specify what other interesting things you have found.