System Programing HW4
Fall 2018

Due: 23:59 Tue, Jan 15th (No Delay submission is allowed)

1. 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 be read into process memory. How to efficiently train such huge data set is critical to application performance.

In this assignment, your job is to write a random forest 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 parallel programing should be significantly less than with sequential programing.

2. Data Description

你將擁有兩個dataset，一個是training_data，一個是testing_data，其中trianing_data會標明誰是壞人誰是好人，而你們的目標是從training_data train出一個Random forest，並使用它抓出testing_data中，誰是好人誰是壞人，並輸出至Submission.csv(詳細作法見Section3 & 4)

第一欄代表 ID

第二到三十四 欄代表他們的feature，表示你一共會有33 維的feature

最後一欄，0代表好人，1代表壞人(只有training_data有，你的目標在於抓出testing_data中的壞人)

3. Random Forests(詳情見wiki)

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:

(a). 從training_data中讀出資料在以下記為training_dataset，切記不要把ID放進去

(b). 從training_dataset中隨機取出等量的資料，切記為取後放回，所以你有可能取到相同的資料

(c). 拿自step (b)中取出的資料作為input，即可種出一顆decision tree(section 4中有其作法)

(d). 重複step(b)跟step(c)若干次，將你所種出的樹集合起來，即為random forest

Testing:

(a). 自testing_data中取出資料

(b). 將你想要預測那一筆資料丟進剛剛的train出來的每一棵decision tree，每一棵decision tree會告訴你那筆資料應該是壞人或是好人

(c). 把每一棵decision tree的結果進行投票，多者勝出(假設你種出九棵樹，五棵樹認為他是壞人，四個認為他是好人，則他為壞人)

4. Decision Tree(詳情見wiki)

   Some of you may not be familiar with decision tree, we will show how to train a random forest with training data step by step as following

   Training:

   (a). 將自Section 3中取出的data，當做input丟進root node中

   (b). 從各個維度中尋找各個維度的最佳切點

   第X維尋找最佳切點的方式：

   (1). 先依照第X維的數字sort過data

   (2). 從第X維數值最小的data開始試切，計算切下去後的Gini impurity

   (3). 找出切在哪個兩點之間的時候Gini impurity為最小，即為該維度的最佳切點

   (c). 比較各個維度的最佳切點，找出最讚最棒的那個切下去，記錄threshold及其維度(假設做出為在第25維的第10筆data及第11筆data之間，threshold即為第10筆data的第25維數值跟第11筆data的第25維數值的平均)，將第25維數值比threshold小的data丟給左邊的node，比threshold大的data丟給右邊的node

   (d). 每個node重複步驟(b)及(c)直至某node內的data全部label為0或是1

   Testing:

   (a). 將Section3中丟進來的資料，依照每個node記錄的維度及threshold，看是走左邊或右邊

   (b). 重複步驟(a)直至走到底，看終點的node為1還是為0，即為此decision tree所算出的答案

5. Perf(看instruction數量的方法)

   Perf 是一個Linux 系統效能評估的工具，請使用以下指令來得到你從讀檔開始到預測出答案所使用的instruction 數量，此工具在工作站上就有了不必額外下載

   perf stat -e intructions:u -v ./hw4

6. You will get

   (a). training_data：你拿來train random forest的input

   (b). testing_data：你拿來test 你做出的random forest的input

   (c). sample_submission.csv：你output的範例格式

   (d). ans.csv：testing_data的解答(給你衡量自己做出來的正確性，因為不是Machine Learning課，所以給此解答，但你的程式執行時不得使用任何此解答的資訊，也不得嵌入程式內)

   (e). spec.pdf：此說明文件

7. Submission:

   命名：hw4_你的學號.zip(Ex: hw4_b01902020.zip)

   其中需包含

   (a). hw4.c

   (b). Makefile

   執行make可正確compile你的code，不得使用-O1 -O2 -O3 -Os

   執行make run可以run 你的code，data_dir預設為"../data"，output預設為"./submission.csv"，tree_number及thread_number為你自行設定最好的參數，其中thread_number需大於等於2，時限為3分鐘

   (c). report.pdf

8. Score(7%)
   (a). [Code]: 你的code能被compile，可以執行以下指令，在時限(3分鐘)內跑出結果，結果正確率需大於85% (1%)

   ./hw4 -data data_dir -output submission.csv -tree tree_number -thread thread_number

   其中：
   -data：data_dir代表training_data testing_data所在的資料夾
   -output：submission.csv代表結果的輸出檔案
   -tree：tree_number代表種幾顆樹
   -thread：thread_number代表開的thread數量(因為你有可能開thread開在很多地方，所以此數字是同一時間所有thread的數量，此數量需大於等於2)

   切記要是此項沒拿到分，後面的項次皆不予計分

   (b). [Report]: 試說明你將thread開在哪裡，是分工在哪裡？(1%)
   (c). [Report]: 試畫出或以表格做出thread數量與時間的比較，以紅色標出時間最快的位置，並說明此圖表(2%)
   (d). [Report]: 試畫出或以表格做出thread數量與intructions數量(詳見Section 5)的比較，並說明此圖表(1%)
   (e). [Report]: 試畫出或以表格做出樹的數量與intructions數量(詳見Section 5)的比較，並說明此圖表(1%)
   (f). [Report]: 說說你的其他發現! (可以是與正確率的比較啦，或是哪個function會造成大量cache miss啦 都可以 都來都來!) (1%)

9. Other rules：
   (a). 不得遲交
   (b). 不得抄襲，發現抄襲者成績一律以F計算