System Programming - 2018 Fall
Programming Assignment #4

R302 – IR Lab

Task Description

Introduction

在「系統程式設計」玩玩「機器學習」

根據「Training Data」建出「Random Forest Model」
接著套用在「Testing Data」上面並找出解答

File://training_data

第 1 欄為「ID」
第 2-34 欄為 Feature
最後 1 欄為 Label

File://testing_data

第 1 欄為「ID」
第 2-34 欄為 Feature
你的工作就是要決定最後一欄

Implementation

Random Forest - Training

1. 從 training_data 讀出資料，記作 training_dataset
   切記不要把 ID 丟進去

2. 從 training_dataset 裡隨機取出跟等量的資料
   取後放回，所以可能會取到重覆的資料

3. 拿第二步取出的資料，當做 Input
   拿去做一棵 Decision Tree

4. 重複前兩個步驟數次
   把做出來的一堆 Tree 集合起來就是 Random Forest

Implementation

Random Forest - Testing

1. 從 testing_data 讀出資料，記作 testing_dataset
   切記不要把 ID 丟進去

2. 把 testing_dataset 裡面的每一筆資料
   丟進去你剛剛做出來的每一顆 Tree
   每一棵 Decision Tree 會告訴你它是好人或是壞人

3. 用 Forest 中的 Tree 們，投票來決定最後的答案

Implementation

Decision Tree - Training

1. 把剛剛的 Input Data 丟進 Root Node 中

2. 從各維度中尋找最佳切點 - 第 x 維尋找最佳切點的方式：
   (a) 先對第 x 維的資料排序
   (b) 從最小的數值開始切，計算出 Gini Impurity
   (c) 有最小 Gini Impurity 的切點即該維的最佳切點

3. 比較哪個維度切點的 Gini Impurity 最小
   即為這個 Node 的最佳切點，記錄維度及 Threshold

4. 把比最佳切點 Threshold 小的資料丟給左子樹，其他丟給右邊

5. 重複步驟二到四直到 Node 中的 Data 都是好人或壞人

Implementation

Gini Impurity

J 是指有幾種 Label，f_i 則是該 Label 在 Data 中的比例

$$I_G(f) = \sum_{i=1}^{J} f_i(1 - f_i)$$

ID	Weight	Speed	Label
1	1000	300	1
2	500	500	1
3	300	150	0
4	200	100	0

假設切在 Weight 這一維，用 750 當 Threshold
右樹的 Gini Impurity 為 $0(1-0) + 1(1-1) = 0$
左樹的 Gini Impurity 為 $0.66(1-0.66) + 0.33(1-0.33) = 0.44$
整體的 Gini Impurity 即為 $0 + 0.44 = 0.44$

Implementation

Decision Tree - Testing

1. 將 Random Forest 丟進來的 Test Data
   依照每個 Node 所記錄的維度及 Threshold
   看是走左邊還是右邊

2. 一直走到底
   看最後的 Node 的 Label 為何，即為所求

Instructions

What is Instructions

簡單來說，Instructions 數量跟工作量成正比

```
                             ; Section __text
                             ; Range 0x100000aa0 - 0x100000ea9 (1033 bytes)
                             ; File offset 2720 (1033 bytes)
                             ; Flags : 0x80000400
                             ;
                        _main:
0000000100000aa0        push    rbp                                     ; XREF=0x1000000d0
0000000100000aa1        mov     rbp, rsp
0000000100000aa4        sub     rsp, 0xa0
0000000100000aab        lea     rdi, qword [ds:0x100000f28]             ; "Please input y:\\n", a
0000000100000ab2        xorps   xmm0, xmm0
0000000100000ab5        mov     dword [ss:rbp+var_4], 0x0
0000000100000abc        mov     dword [ss:rbp+var_1C], 0x0
0000000100000ac3        mov     dword [ss:rbp+var_20], 0x64
0000000100000aca        mov     byte [ss:rbp+var_29], 0x30
0000000100000ace        movsd   qword [ss:rbp+var_38], xmm0
0000000100000ad3        mov     al, 0x0
0000000100000ad5        call    imp___stubs__printf
0000000100000ada        lea     rdi, qword [ds:0x100000f39]             ; "%lf", argument "format
0000000100000ae1        lea     rsi, qword [ss:rbp+var_10]
0000000100000ae5        mov     dword [ss:rbp+var_54], eax
0000000100000ae8        mov     al, 0x0
0000000100000aea        call    imp___stubs__scanf
0000000100000aef        lea     rdi, qword [ds:0x100000f3d]             ; "Please input cashflow:
0000000100000af6        mov     dword [ss:rbp+var_58], eax
0000000100000af9        mov     al, 0x0
0000000100000afb        call    imp___stubs__printf
```

Instructions

Perf

Perf 是一個 Linux 系統效能評估的工具

此工具在系上工作站上就有了，不必額外下載

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

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

perf list 可以看更多

Submission

Files You Get

1. training_data
2. testing_data
3. sample_submission.csv – 你輸出的格式
4. 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

Submission

Submission Format

命名為 hw4_你的學號.zip，例如 hw4_b03902015.zip

解壓縮後會生出名為你的「學號」的資料夾，資料夾中需包含

1. hw4.c
2. Makefile
   執行 make 可正確編譯你的程式，不得使用 -O2 –Os 等加速
   執行 make run 可以執行你的程式
   data_dir 預設為”../data”
   output 預設為”./submission.csv”
   tree_number 及 thread_number 自行設定為最好的參數
   其中 thread_number 需大於等於二，時限為三分鐘
3. report.pdf

Submission

Score - Program

1 Point

你的程式碼能被正確編譯，可以執行行以下指令
在時限內跑出結果且正確率需大於85%

./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 代表開的執行緒數量。因為你有可能開在很多地方
所以此數字是同一時間所有 thread 的數量，此數量需大於等於二

Submission

Score – Report

1 Point
試說明你將執行緒開在哪裡，是分工在哪裡

2 Points
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

1 Point
試畫出或以表格做出 Thread 數量與 Instructions 數量的比較，並說明此圖表

1 Point
試畫出或以表格做出樹的數量與 Instructions 數量的比較，並說明此圖表

1 Point
說說你的其他發現，可以是與正確率的比較啦，或是哪個函式會造成大量 Cache Miss 啦
都可以！都來都來！

Any Questions ?