数据挖掘第二次项目

并行决策树集成

陈铭涛
16340024

May 26, 2019

1 CART 算法

CART 包含了分类决策树和回归决策树的算法，

2 Gradient Boosting

使用的损失函数为均方误差：

$$L = \frac{1}{N}\sum_{u=1}^{n}(Y_i - \hat{Y_i}) \tag{1}$$

3 Random Forest

4 代码实现

出于内存和并行化实现的考虑，本次项目选择了使用 Rust 语言实现，Rust 的 RAII 机制使得临时资源可以及时地释放，提升内存利用率，由编译器提供的静态检查可以避免多线程时线程不安全的情况，降低 debug 难度。

程序实现中使用的第三方库如下：

1. rayon: 提供基于迭代器的便捷地编写并行代码的方法
2. rand: 提供随机数生成
3. csv: 提供对 csv 文件的读取
4. indicatif: 提供命令行进度条实现
5. ndarray: 提供类似 numpy 的多维数组的操作
6. num-traits: 提供数值类型上的一些实用方法，如最大最小值等

7. log: 程序日志
8. pretty_env_logger: 程序日志输出
9. num_cpus: 获取系统 CPU 核心数量
10. serde: 提供将结构体变量序列化的功能
11. serde_json: 用于以 json 格式将序列化后的模型变量保存至文件

项目中包括的主要代码文件如下

- data_frame.rs: 使用一个二维的 `ndarray` 作为程序使用的 `DataFrame` 类型，并定义了类型别名`V` 作为数据的存储类型，默认为 *f64*, 此外还包含了 csv 文件读写等其他实用函数
- learner.rs: 定义了一个 `Learner` trait，包括了类似于 sklearn 的`fit` 和`predict` 两个方法，决策树，Boosting 和 Random Forest 都需实现该 trait。
- tree.rs: 包括了 `DecisionTree`类型，实现了 CART 算法，可对单个决策树进行训练和预测。
- boosting.rs: 包括了`GradientBoosting` 类型的定义与训练和预测的实现。
- random_forest.rs: 包括了`RandomForest` 类型的定义与训练和预测的实现。
- utils 目录：包括了数个实用功能，如获取数据列排序序列，交叉验证，模型分数计算等。

5 并行化表现

6 验证

以下程序测试均在一台搭载 6 核 12 线程 CPU，运行 macOS 系统的笔记本电脑上运行。

验证的标准为 R^2, 其计算方法如下：

$$
\begin{aligned}
\bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i, \\
SS_{tot} &= \sum_i (y_i - \bar{y})^2, \\
SS_{res} &= \sum_i (y_i - f_i)^2, \\
R^2 &= 1 - \frac{SS_{res}}{SS_{tot}}
\end{aligned}
\tag{2}
$$

其中 y_i 为第 i 个样本的实际观察值，f_i 为第 i 个样本的模型预测值。R^2 的取值通常在 0 与 1 之间，越接近 1 代表预测值与真实值匹配程度越高。

使用 `LightGBM` 默认参数构建一个模型运行 3 折交叉验证进行对比：

```
In [6]: run_cross_validation(trains, labels, lgb.LGBMRegressor, params)

fit_time: [24.27433276 22.56340122 23.01793098]
Average fit_time: 23.285222
score_time: [10.66991425  9.20692468  8.52207994]
Average score_time: 9.466306
test_mse: [-0.33291203 -0.33487001 -0.34085721]
Average test_mse: -0.336213
train_mse: [-0.33630982 -0.33519583 -0.33240139]
Average train_mse: -0.334636
test_r2: [0.156102   0.15480654 0.15376245]
Average test_r2: 0.154890
train_r2: [0.15816958 0.15915648 0.15922097]
Average train_r2: 0.158849
```

Figure 1: LightGBM 交叉验证结果

对单棵决策树进行交叉验证获得的结果如下：
在训练集上获得的平均 R^2 分数为 0.14947528261278345
在验证集上获得的平均 R^2 分数为 0.143995380629807
训练时间平均为 682423 ms.

```
    Finished release [optimized + debuginfo] target(s) in 0.84s
     Running `target/release/cv`
 INFO  cv > Train data shape: [10000004, 13]
 INFO  cv > Label data shape: [1, 10000004]
 INFO  cv > Load time: 6919ms
 INFO  ensembles_rs::utils::cross_validate > train time: 712503, predict time: 295
 INFO  ensembles_rs::utils::cross_validate > train: 0.14879519294338295, validation: 0.144
71150292761092
 INFO  ensembles_rs::utils::cross_validate > train time: 658188, predict time: 254
 INFO  ensembles_rs::utils::cross_validate > train: 0.1489242194641912, validation: 0.1453
8125132877167
 INFO  ensembles_rs::utils::cross_validate > train time: 676580, predict time: 341
 INFO  ensembles_rs::utils::cross_validate > train: 0.15070643543077622, validation: 0.141
89338763303838
CrossValidateScore { train_time: [712503, 658188, 676580], predict_time: [295, 254, 341],
train_score: [0.14879519294338295, 0.1489242194641912, 0.15070643543077622], validation_sc
ore: [0.14471150292761092, 0.14538125132877167, 0.14189338763303838] }
```

Figure 2: 单棵决策树训练交叉验证结果

使用 Gradient Boosting 训练 150 步，设置单棵树最大生长至 2 层，进行交叉验证获得的结果如下：
在训练集上获得的平均 R^2 分数为 0.14595633826167811
在验证集上获得的平均 R^2 分数为 0.14329480642795986
训练时间平均为 223031 ms.

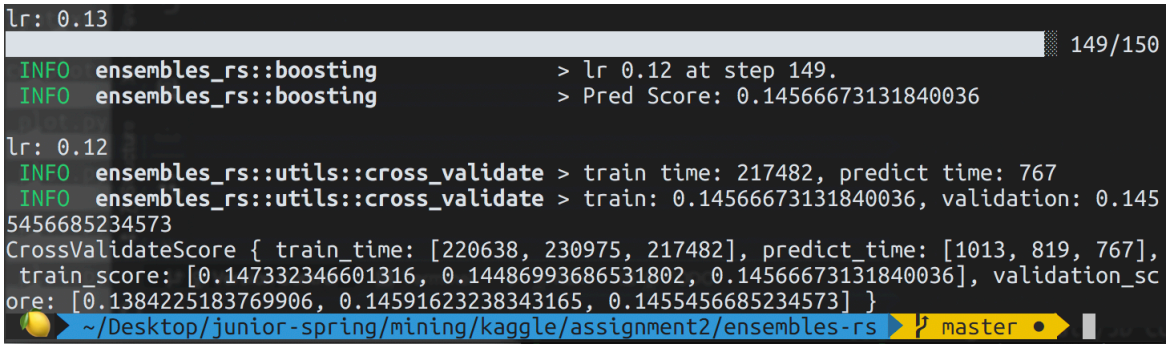

Figure 3: GBDT 训练交叉验证结果

使用 Random Forest 训练，决策树数量为 150，不限制决策树生长深度，进行交叉验证获得的结果如下：

在训练集上获得的平均 R^2 分数为 0.1496125066152948

在验证集上获得的平均 R^2 分数为 0.14889900127096856

训练时间平均为 444804 ms.

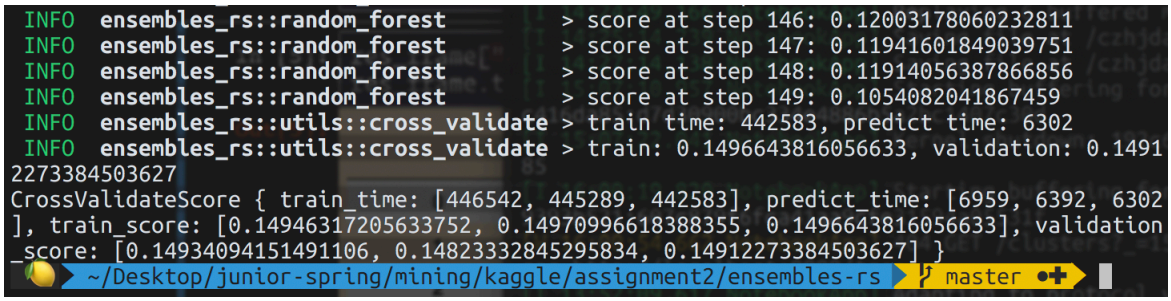

Figure 4: 随机森林训练交叉验证结果

四个训练中使用取样工具查看内存占用值分别如下：

1. LightGBM: 5.2G

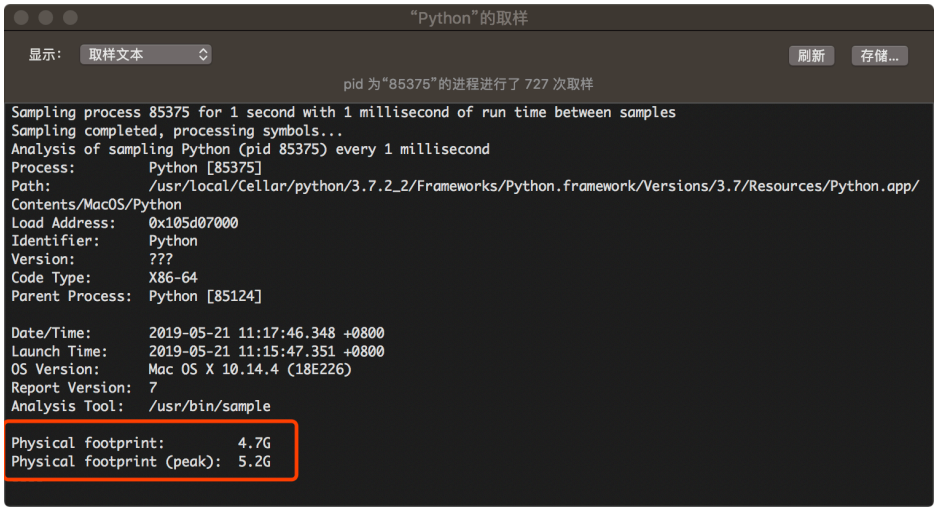

Figure 5: LightGBM 内存占用

2. 单决策树：3.2G

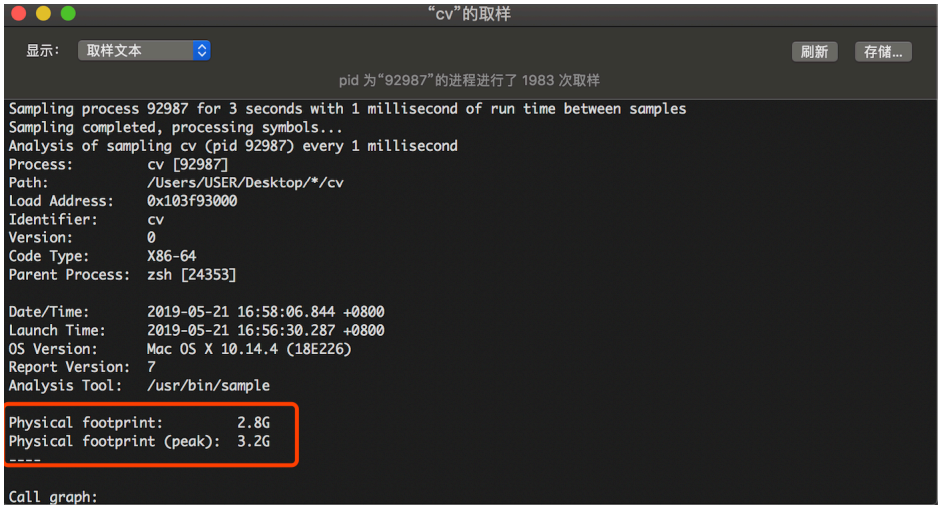

Figure 6: 单棵决策树训练内存占用

3. Gradient Boosting: 3.4G

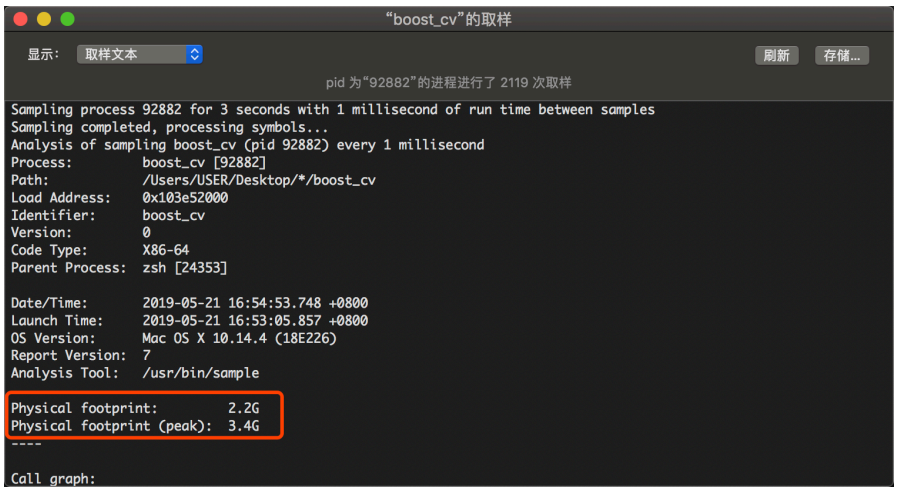

Figure 7: Gradient Boosting 训练内存占用

4. Random Forest: 2.1G

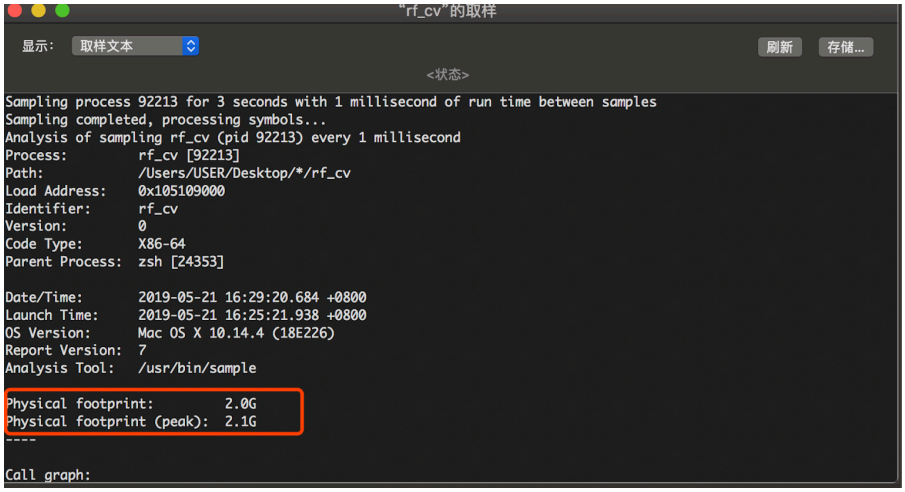

Figure 8: Random Forest 训练内存占用

7 Kaggle 分数

使用单棵决策树训练至 10 层后提交至 Kaggle 获得的分数为 0.16087:

Figure 9: 单棵决策树分数

使用 Gradient Boosting, Learning Rate 固定为 0.25，基学习器最大训练至 3 层，训练步数为 400 时的分数为 0.16957：

Figure 10: lr=0.25, GBDT

使用 Random Forest, 不限制单棵决策树生长，使用决策树总数为 350，时的分数为 0.17210：

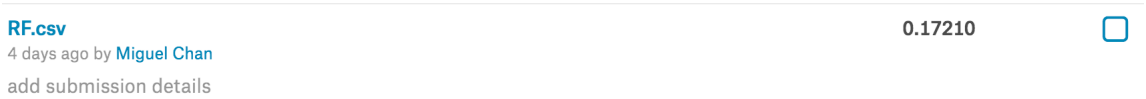

Figure 11: 随机森林