

Random forest README

目錄

- [介紹](#)
- [組件](#)
- [TreeQueue 類別](#)
- [RandomForest 類別](#)
- [平行處理](#)
- [OpenMP 平行化](#)
- [一般工作流程](#)
- [其他注意事項](#)
- [可能的改進](#)

Introduction(介紹)

所提供的程式碼似乎涉及使用 OpenMP 和 CUDA 進行平行處理的隨機森林演算法實作。本 README 將代碼分解為其組件，解釋主要功能，並討論所使用的平行化技術。

組件

TreeQueue 類別

```
void tree_queue::push(Tree&& t) {
    std::unique_lock<std::mutex>
locker(write_mutex_m);
    base_queue_m.push(std::move(t));
}

bool tree_queue::pop(Tree& t) {
    std::unique_lock<std::mutex>
locker(write_mutex_m);
    if (base_queue_m.empty()) { return
false; }

    t = std::move(base_queue_m.front());
    base_queue_m.pop();
    return true;
}
```

tree_queue 類別似乎代表樹的隊列。它提供了將樹推入隊列和從中彈出的方法，使用互斥鎖 (**write_mutex_m**) 確保線程安全性。

RandomForest 類別

cppCopy code

```
RandomForest::RandomForest(int n_estimators,
double entropy_threshold, int max_depth, int
random_state, int n_jobs)
    : n_estimators_m(n_estimators)
    , max_depth_m(max_depth)
    , random_state_m(random_state)
    , entropy_threshold_m(entropy_threshold)
    , n_jobs_m(n_jobs)
{}
```

RandomForest 類別是實現隨機森林演算法的主要類別。它初始化參數，如估算器的數量、最大深度、隨機種子、熵閾值和并行作業的數量。

```
void RandomForest::fit(dataset& tr_ds) {
}
```

平行處理

OpenMP 平行化

cppCopy code

```
#pragma omp parallel num_threads(n_jobs_m)
{
    #pragma omp for schedule(dynamic)
    nowait
    for (int i = 0; i < n_estimators_m;
        ++i) {
        build_trees(tr_ds, todo_trees,
out_trees);
    }
}
```

fit 方法使用 OpenMP 將多個決策樹 (**n_estimators_m**) 的訓練並行化。它創建了一個 **todo_trees** 隊列來存儲待構建的樹，以及一個 **out_trees** 隊列來存儲完成的樹。然後並行執行 **build_trees** 方法來構建樹。

一般工作流程

初始化： 在對象創建期間設置隨機森林的參數。

訓練： **fit** 方法使用 OpenMP 將多個決策樹並行訓練。樹被同時構建並存儲在 **trees_m** 向量中。

預測： **predict** 方法使用已訓練的隨機森林對新數據進行預測。

其他注意事項

代碼假設數據集具有特徵 (**x**) 和標籤 (**y**) 的結構。構建決策樹的過程涉及為每棵樹創建數據和特徵的子集。

可能的改進

進一步的註釋和標記可以改善對特定部分和步驟的理解。

錯誤處理和日誌記錄可以增強對訓練和預測過程中例外情況的處理。

請注意，對代碼的全面理解可能需要訪問特定類別和函數的具體實現，如 **Tree** 和 **dataset**。

重要指令

cmake: 生成Makefile。

make: 編譯代碼。

./cpp_2020_random_forest: 執行可執行文件。

以上三個指令皆須在/cpp-2020-random-forest-main/build

python3 sklearn_test.py:觀察訓練model time(包含build tree time etc...)

而此指令在/cpp-2020-random-forest-main