决策树和随机森林算法多因子策略报告

金滢

1 概述

利用现有的 50 个因子，采用决策树和随机森林算法选择股票。前期用了自己写的决策树所以性能很差，数据量较大导致运行慢。后面直接用 sk-learn 包里的决策树和随机森林算法来训练，速度快，但模式比较固定。

选股方法是使用每隔五天的数据，五天一调仓。使用上一天（5 天前）训练的决策树（或随机森林）预测今天的结果（分类树将得到 1 或 0，回归树将得到某个数值，具体训练方法后述）并选择 long 的股票和 short 的股票，两者总金额固定等权重买卖获得今日的 return，累乘得到 wealth curve。

使用所有 50 个 factor，设置最大树深为 4，用 2006 年到 2010 年的数据（间隔 5 天），回测收益从回归随机森林、回归树、分类随机森林、分类树递减。

分类树的代码文件为"CTree.py",分类随机森林的代码文件为"CForest.py",回归树的代码文件为"RTree.py",回归随机森林的代码文件为"RForest.py"

之后对算法做了两次改进，一次是对单棵决策树增加了交叉验证，一定程度上可以提升表现，消除单棵树的过拟合问题。代码文件为 "$Ctree\_multi.py$" 和 "$RTree\_multi.py$". 另一次是增加了训练时长，采用了多天的数据进行训练，很明显地提升了两种森林的表现，代码文件为 "$new\_CForest.py$" 和 "$new\_RForest.py$".

2 二分类决策树算法及结果

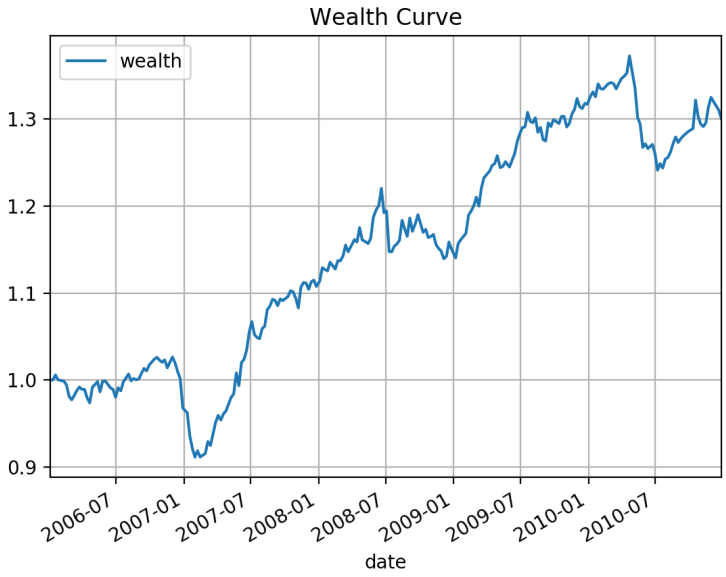
2.1 训练和选股方法

使用二分类决策树（CART 算法，训练依据为 gini），将所有股票按当天的 5-day forward return 从高到低排序，前 1/3 的股票标记为 1，后 1/3 的股票标记为 0，将这些股票当天所有 factor 的得分作为训练数据，训练的 target 为前面标记的 1 和 0。

利用前一天训练的决策树分类器对今天的股票进行选择，预测为 1 的股票作为 long，预测为 0 的股票作为 short，两者按相同的总额，每支股票等权重买卖。

2.2 回测结果

仅用一棵决策树的收益情况不是很好，wealth curve 如图：



3 二分类随机森林算法及结果

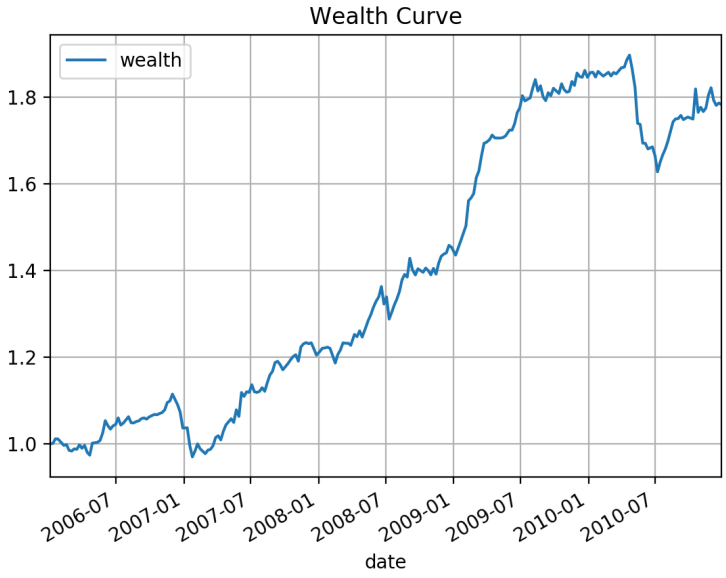
3.1 训练和选股方法

与之前的二分类决策树算法类似，只是使用随机森林增加稳定性，设置每个森林有 50 棵树，对训练数据的处理与二分类决策树相同，选股和买卖方式也相同。

3.2 回测结果

回测结果的趋势与之前二分类决策树大致类似，但曲线更稳定，收益也更高。

wealth curve 如图：



4 回归树算法及结果

4.1 训练和选股方法

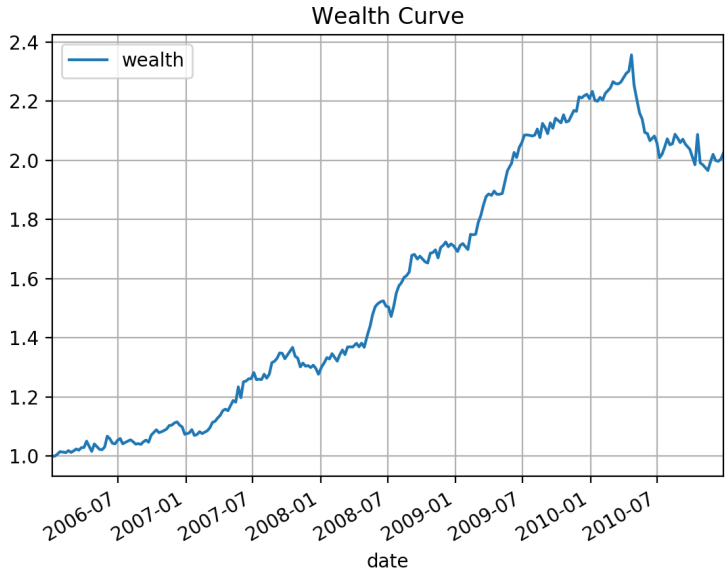
使用回归树算法，训练判据是最小方差，将所有股票按当天的 5-day forward return 从高到低排序，前 1/3 的股票标记为 1，后 1/3 的股票标记为 0，将这些股票当天所有 factor 的得分作为训练数据，训练的 target 为选定的股票的 5-day forward return。

使用前一天训练的回归树来对今天的股票预测 forward return。之后对所有股票按照预测值排序，取前 1/5 作为 long，后 1/5 作为 short 等额对冲，各自内部股票等权重买卖。

4.2 回测结果

回归树的结果比分类树和分类森林更好，回测的年化收益 15%。我觉得是因为在分类算法中最终给出的是 1 和 0，而我的选股方法是买 1 卖 0，范围太大，或许可以尝试买 1 中的一部分、卖 0 中的一部分。而对回归算法来说，可以很容易地对预测结果进行区分（排序），取其中一部分来做 long-short 对冲。

wealth curve 如图：



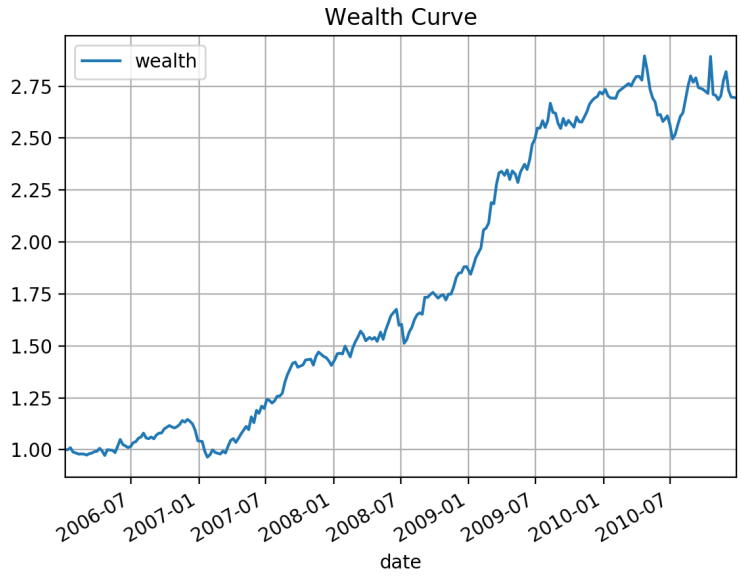
5 回归森林算法及结果

5.1 训练和选股方法

数据选取和训练、选股方法与回归树相同，只是将回归树改为回归森林以减少单棵树的随机性。

5.2 回测结果

回归森林的收益曲线趋势与回归树大致相同，但收益进一步提高到 2.75，年化收益有 22.4%。wealth curve 如图：



6 改进 1：关于贪心算法和交叉验证

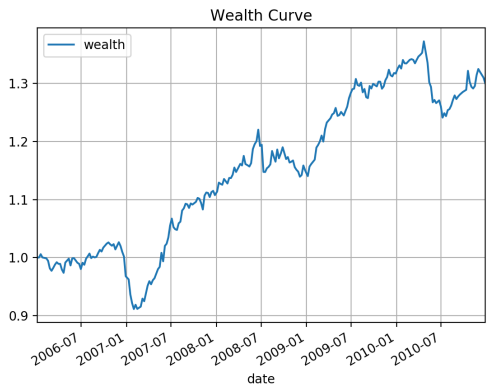
决策树的训练方法是贪心算法，对于每个节点直接选取 gini 系数最小的 factor 及其分类方法。虽然在决策树的训练中有剪枝（比如后剪枝的依据是如果该节点不能提高分类性能则将节点删除），但这样的剪枝思路还是要求最后决策树使得分类最纯。（我的理解是这样的，不知道有没有错）

所以在我的理解里这里还是存在对数据的过拟合（只考虑所有样本的分类），可能在泛化时会有问题，因为当天的决策树使用了所有的数据直接训练，而不像其他机器学习算法一样有交叉验证的过程。或许每次训练单棵决策树（无论是分类还是回归）时先对所有股票随机分出 80% 来训练决策树，再用剩下 20% 来验证决策树的分类结果。一共训练多棵决策树，取里面验证结果最好的。

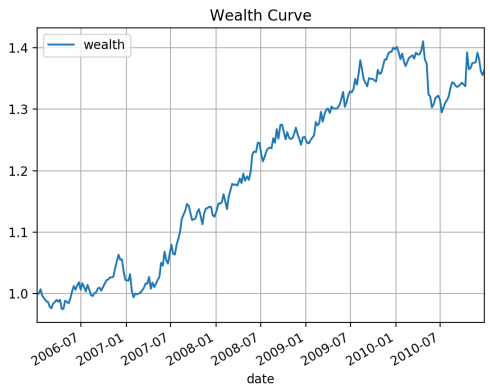
对分类树和做这个改进，具体分类结果比较时，计算如下 score：

$$score = \frac{\text{预测为 1 且实际为 1 的股票数}}{\text{预测为 1 的股票数}} * \frac{\text{预测为 0 且实际为 0 的股票数}}{\text{预测为 0 的股票数}}$$

作为对这棵树分类效果的评价函数。回测结果比较如下：



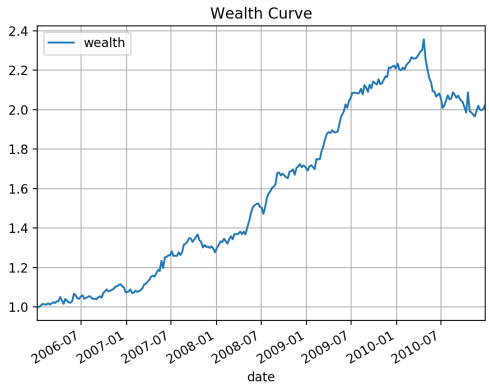
简单决策分类树



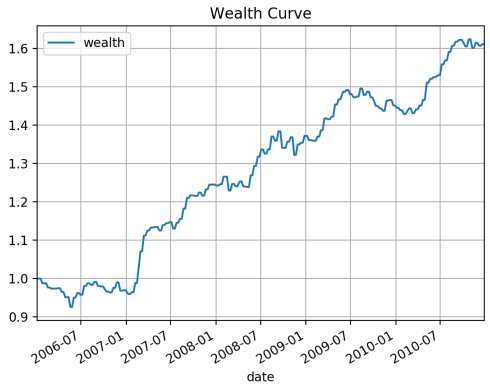
加入验证的决策分类树

可以看到分类树加入验证后回测收益率更高，且更稳定。

对回归树做同样处理，这里将 return 排名前 1/2 的标为 1，后 1/2 的标为 0，其余与分类树相同。回测结果比较如下：



简单决策回归树

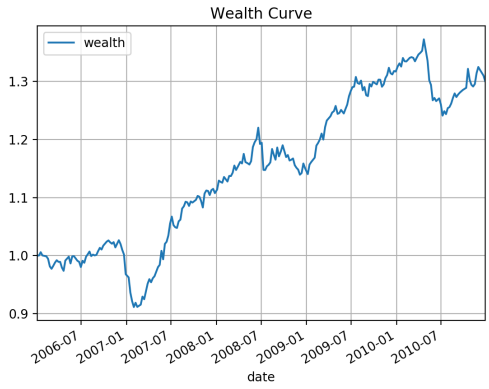


加入验证的决策回归树

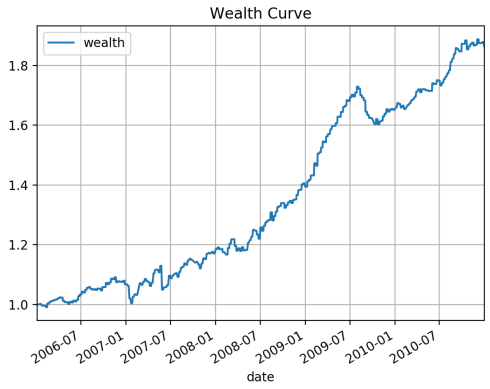
发现回归树加入验证后回撤变小，但收益率不如简单回归树，更加不如回归随机森林。

7 改进 2：增加窗口期

在之前策略的基础上，将训练数据向前取多天（之前只取当天），用参数 $back\_days$ 表示，由于计算速度等原因，这里回测取 $back\_days = 5$。将当天与之前共 $back\_days$ 天，每天的所有股票按 return 排序，取当天前 10% 标记为 1，后 10% 标记为-1，全部取出的记录作为训练数据。对分类决策树、分类随机森林、回归决策树和回归随机森林都作了改动。其中分类树有一些提升，两种随机森林的平滑性和收益率都有很好的提升，将结果总结对比如下。（两个森林的代码文件分别为 "$new\_CForest.py$"，"$new\_RForest.py$"

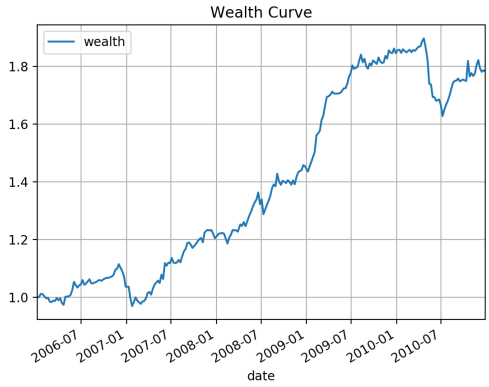


简单决策分类树

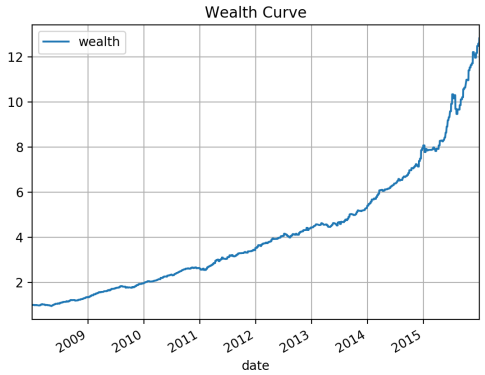


用多天数据训练的分类树

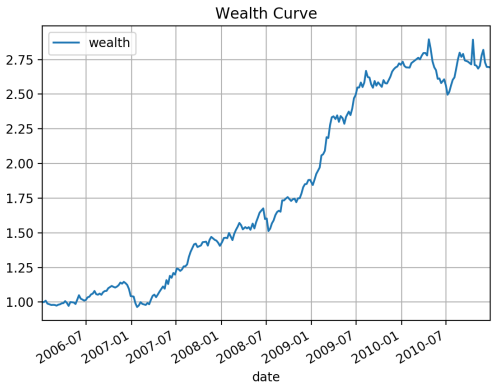
发现随机森林的表现由于增加训练天数而有了很明显的提升，回归森林的年化收益率达到 47%。可能是因为随机森林本身将过拟合问题处理得较好，用更长时间数据得到了更好的效果。不过单棵决策树的表现还是一般。本想对 20 天甚至 200 天的数据训练随机森林，但个人电脑跑不起来这么多数据。



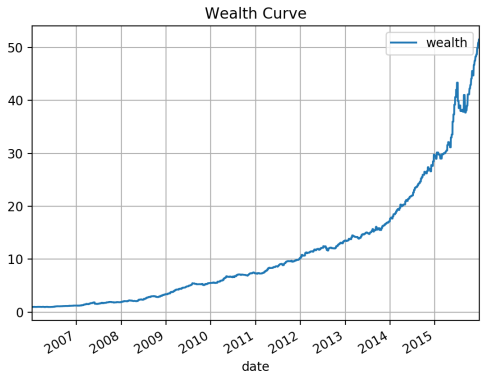
单天数据训练的分类随机森林



5 天数据训练的分类随机森林



单天数据训练的回归森林



5 天数据训练的回归森林