# 基于遗传规划的二级市场因子挖掘

## 一、问题背景

因子挖掘是证券分析领域中的热点问题，旨在寻找与高投资回报率相关的市场指标。有效的因子能够反映当前的市场属性，有利于宏观调控；同时也是机构和个人的重要投资参考。传统的因子挖掘通常采用演绎法，即从二级市场的量价、基本面信息入手，主观总结规律和经验进行因子挖掘和改进。常见的因子如估值因子、成长因子等都是由这种方法研究得出。随着量化投资的概念兴起，对证券量价、基本面等信息量化建模成为了一类有效的投资决策辅助方法。相较于演绎法，量化方法有着客观、无需先验知识、善于处理大数据的优势。受此启发，本研究将因子挖掘量化建模为并求解。以下是问题简述。

对于因子挖掘问题，每个因子的公式都是证券基本信息和算子的组合。例如，在WorldQuant公开的”alpha101“中，3号因子表示如下：

$$\alpha = -1 \times (\mathrm{correlation}(\mathrm{rank}(open), \mathrm{rank}(volume), 10)$$

式中，$open$(开盘价)、$volume$(成交量)为证券基本量价信息，correlation、rank、$\times$为算子。由于量价信息与算子的组合公式往往无法直接得出投资回报率，因子挖掘中通常利用因子的暴露度(即因子数量值)与投资回报率的信息系数(IC)评估因子的效用。因子挖掘便是要找出信息系数最高的因子公式。

## 二、问题建模

由以上介绍可知，因子挖掘试图发现某种最优的公式并以此利用特征变量(量价、基本面信息)反映目标变量(投资回报率等信息)，因此可以视为一类符号回归问题，适合利用遗传规划方法[1]对其近似求解。本问题的最终优化目标为：

$$\mathrm{maximize}\ \ \mathrm{RankIC}(\boldsymbol{\alpha}, \boldsymbol{y})$$

其中，$\boldsymbol{\alpha}$表示证券池的因子暴露度序列；$\boldsymbol{y}$表示目标序列，在本研究中我们将其设定为20日后的收益率序列，即调仓周期为20日。式中RankIC是信息的一类改进形式，表示为证券池中$n$只证券在特定时间截面$t$上的因子暴露度与相应收益率秩次差的相关系数，即：

$$\mathrm{RankIC} = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

其中，$d_i$即为$\alpha_i$秩次与$y_i$秩次的差值。相比于IC，RankIC能够避免由于原始量价信息数量级不同而导致因子值差距过大的问题。本研究中，适应度指标定义为在所有时间截面上RankIC的均值。

对于单个因子，其暴露度表示为：

$$\alpha = f(x) \text{ subject to } x \in \Omega$$

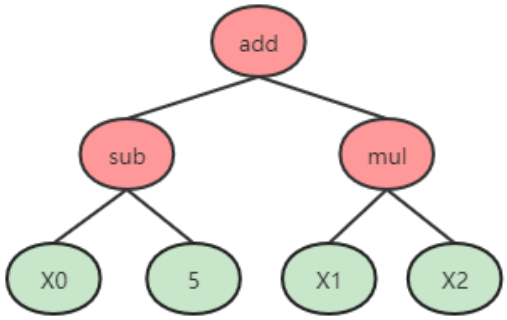
本研究中，可行解集$\Omega$考虑日频的证券基本量价信息，包括：$open$(开盘价)、$close$(收盘价)、$high$(最高价)、$low$(最低价)、$volume$(成交量)以及$vwap$(成交量加权平均价)。式中$f$则是由各类算子组合而成的公式。在一般的遗传规划中，算子包括基础的运算函数：add、sub、div、mul、abs、log等。考虑到证券量价信息的时序特性，本研究对遗传规划的算子集进行了扩展，加入了一系列自定义的时序运算函数。本研究用到的完整算子集见下表：

**表1. 算子列表**

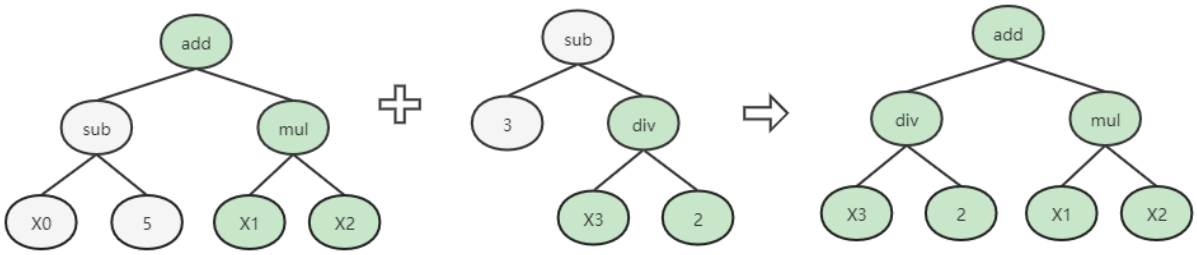
| 类型 | Aa 名称 | 定义 | Column |
|---|---|---|---|
| 基础算子 | add(X, Y) | X + Y | |
| 基础算子 | sub(X, Y) | X - Y | |
| 基础算子 | mul(X, Y) | X * Y | |
| 基础算子 | div(X, Y) | X / Y | |
| 基础算子 | abs(X) | 取绝对值 | |
| 基础算子 | log(X) | 取对数 | |
| 基础算子 | sqrt(X) | 开根号 | |
| 基础算子 | inv(X) | 取倒数 | |
| 自定义算子 | rank(X) | x在X中的分位数 | |
| 自定义算子 | delay(X, d) | d天以前的X值 | |
| 自定义算子 | correlation(X, Y, d) | 过去d天的X序列与Y序列的相关系数 | |
| 自定义算子 | covariance(X, Y, d) | 过去d天的X序列与Y序列的协方差 | |

| 类型 | Aa 名称 | 定义 | Column |
| --- | --- | --- | --- |
| 自定义算子 | delta(X, d) | X - delay(X, d) | |
| 自定义算子 | ts_min(X, d) | 过去d天中X的最小值 | |
| 自定义算子 | ts_max(X, d) | 过去d天中X的最大值 | |
| 自定义算子 | ts_argmin(X, d) | 过去d天中X的最小值的位置 | |
| 自定义算子 | ts_argmax(X, d) | 过去d天中X的最大值的位置 | |
| 自定义算子 | ts_rank(X, d) | d天以前x所处截面的分位数 | |
| 自定义算子 | ts_sum(X, d) | 过去d天中的时序数列之和 | |
| 自定义算子 | ts_prod(X, d) | 过去d天中的时序数列之积 | |
| 自定义算子 | ts_stddev(X, d) | 过去d天中的时序数列的标准差 | |

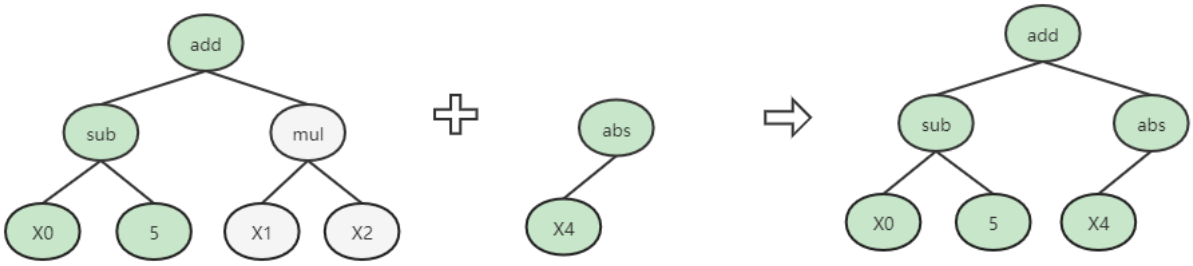
以上算子集作将为遗传规划的函数集。在遗传规划中，第一代的个体被随机生成，随后计算每一代个体的适应度，并以此选出合适的个体作为下一代的父类[2]。本研究中，每一个个体都是因子的公式。遗传过程中每一代都有概率产生交叉、突变操作以保证种群的多样性。为了适应这类进化操作，符号回归问题中的公式通常表示为二叉树形式。如$(x_0 - 5) + (x_1 \times x_2)$表示为：



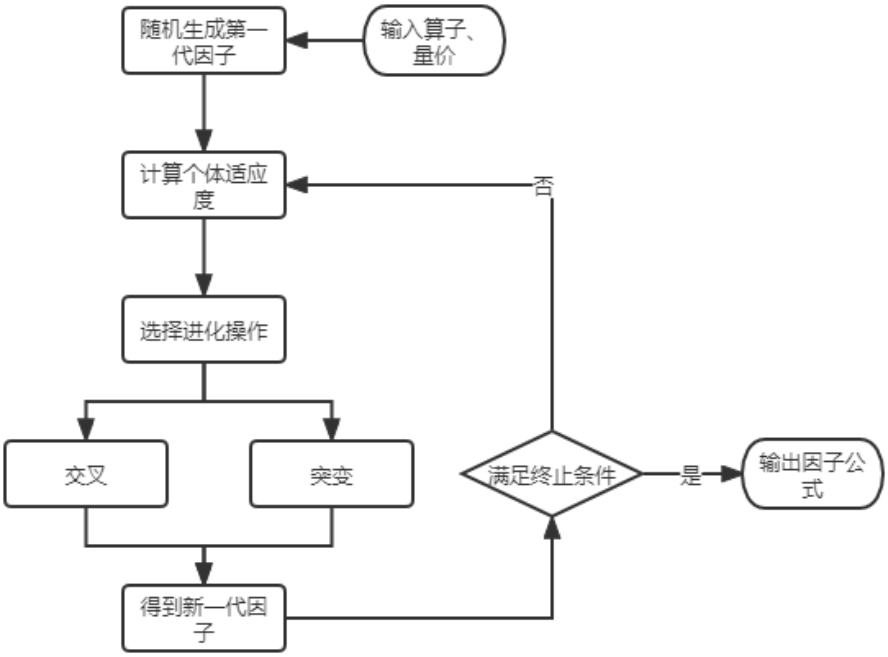

交叉操作将一个适应度较高的公式树的随机子树替换为另一轮中适应度较高的公式树的随机子树，以此形成新的后代。该方法是较为常用的进化方式，过程如下图示意：




突变操作将随机选择公式树中的某一个结点，并使用一个完全随机生成的公式树或单个结点代替。该方法可以较好的维持种群多样性，过程如下图示意：




通过以上进化操作，利用遗传规划优化因子公式的总体流程如下图所示：



# 三、实验结果与分析

本研究从聚宽数据源[3]获取国内A股市场的五支证券作为证券池，包括：`300750`(宁德时代)、`601919`(中远海控）、`600795`(国电电力)、`300052`(中青宝)、`600522`(中天科技)。该股票池由雪球论坛话题中随机选取，对应用遗传规划方法而言不失一般性。

数据训练区间设置为：2020年1月1日~2020年12月1日期间所有交易日，回测区间设置为：2021年1月1日至2021年12月1日期间所有交易日。

实验中使用的基本量价信息和算子定义如第二节所述。由于自定义算子涉及时间序列操作，且适应度函数涉及各个时间截面上因子值的计算，定义算子的输入变量都为$m \times n$的矩阵。其中$m$为证券池大小，$n$为时间截面数量。算子操作不改变矩阵的尺寸，因此所有输出矩阵尺寸仍为$m \times n$。最终由评估函数聚合为RankIC均值作为最终的适应度。

实验参数中，设定最初种群大小为100，进化10代终止；交叉和突变的概率分别设置为0.5和0.1。进化完成后，取适应度最高的5个因子观察，实验结果如下：

| | | fitness | | | | | | size | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gen | nevals | avg | gen | max | min | nevals | std | avg | gen | max | min | nevals | std |
| 0 | 100 | 0.0568045 | 0 | 0.37781 | -0.250648 | 100 | 0.115779 | 4.1 | 0 | 8 | 2 | 100 | 1.55242 |
| 1 | 56 | 0.267661 | 1 | 0.390547 | -0.145711 | 56 | 0.129976 | 3.51 | 1 | 8 | 1 | 56 | 1.16185 |
| 2 | 58 | 0.34266 | 2 | 0.390547 | 0 | 58 | 0.0902167 | 3.76 | 2 | 8 | 3 | 58 | 1.04038 |
| 3 | 55 | 0.328553 | 3 | 0.390547 | 0.0208955 | 55 | 0.113832 | 5.05 | 3 | 8 | 3 | 55 | 0.829156 |
| 4 | 67 | 0.28945 | 4 | 0.394566 | -0.013516 | 67 | 0.1444 | 4.99 | 4 | 8 | 1 | 67 | 1.17043 |
| 5 | 58 | 0.310354 | 5 | 0.394566 | -0.061691 | 58 | 0.13171 | 4.93 | 5 | 8 | 3 | 58 | 0.919293 |
| 6 | 42 | 0.312482 | 6 | 0.394566 | 0.019327 | 42 | 0.13468 | 5.06 | 6 | 9 | 3 | 42 | 1.07536 |
| 7 | 51 | 0.283839 | 7 | 0.394566 | -0.018782 | 51 | 0.14215 | 5.07 | 7 | 12 | 3 | 51 | 1.20212 |
| 8 | 44 | 0.327657 | 8 | 0.394566 | -0.085672 | 44 | 0.128853 | 5.06 | 8 | 9 | 3 | 44 | 0.957288 |
| 9 | 62 | 0.325783 | 9 | 0.398497 | -0.034328 | 62 | 0.121656 | 5.04 | 9 | 9 | 3 | 62 | 0.760526 |
| 10 | 56 | 0.318091 | 10 | 0.398497 | 0.020895 | 56 | 0.12776 | 5.43 | 10 | 8 | 3 | 56 | 1.15113 |

输出前六名的因子为：

```
sma(ts_rank(ts_sum(vwap, 5), 60), get5())
sma(ts_rank(add(close, close), 60), get5())
sma(ts_rank(close, get60()), get5())
sma(ts_rank(close, 60), 5)
sma(ts_rank(close, 60), get5())
```

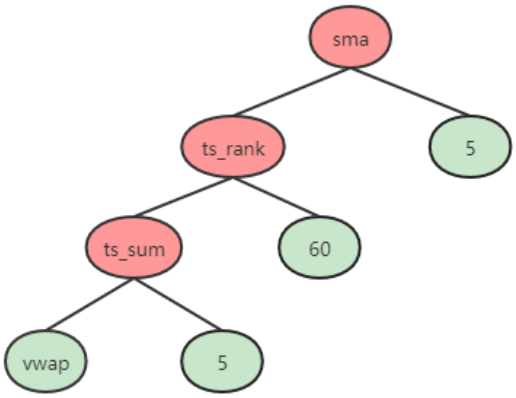
从实验日志中可以看出，随着进化轮次的增加，适应度平均值(avg)稳定至0.32附近，最大值(max)于最后两轮增加至0.398。因子公式树的平均深度也保持在5左右，复杂度没有陡增。优化过程处于比较理想的状况。

由于代码层面的实现原因，第3至第5名的因子表达式等价，这里取第1名的因子进一步分析。如有需求可以输出更多的因子。

第一名的因子表达为：

$$\alpha_1 = \text{sma}(\text{ts\_rank}(\text{ts\_sum}(vwap,\ 5),\ 60))$$

其中，sma、ts_rank、ts_sum为算子，$vwap$为量价信息。该因子的公式树形如：




接下来对第1名的因子进行测试。本研究不简单在其他时间选用RankIC测试，而是在多重指标下测试实际收益情况，包括：调仓周期分别为1、5、20的平均收益分析。测试证券池保持不变。

各个因子分位数(以0.2为一档)的因子值范围、均值和方差统计如下表：

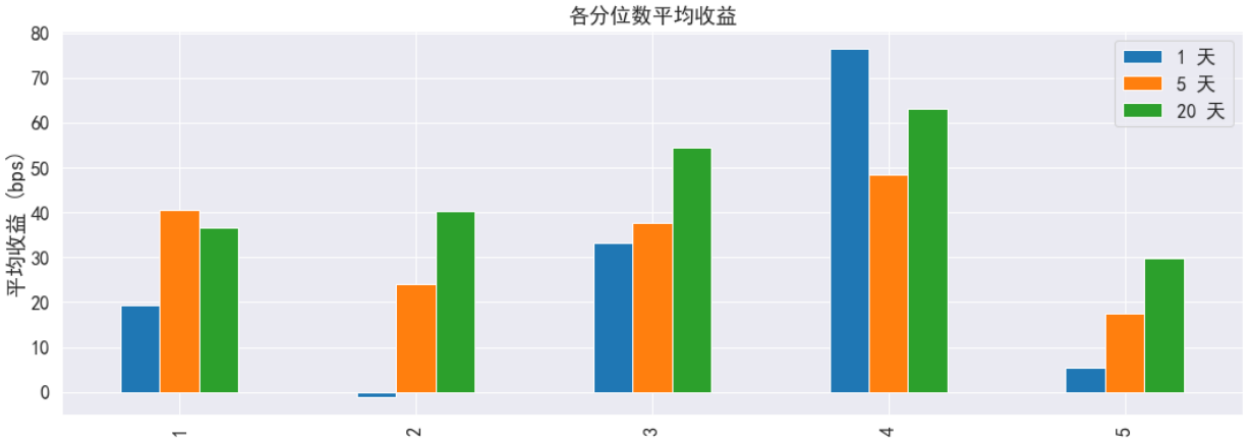
| factor_quantile | min | max | mean | std | count | count % |
|---|---|---|---|---|---|---|
| 1 | 1.0 | 24.8 | 9.603315 | 6.083182 | 181 | 20.000000 |
| 2 | 3.6 | 56.8 | 22.140331 | 13.193752 | 181 | 20.000000 |
| 3 | 18.8 | 58.6 | 39.862637 | 9.360846 | 182 | 20.110497 |
| 4 | 21.8 | 59.8 | 45.847778 | 9.154597 | 180 | 19.889503 |
| 5 | 35.0 | 60.0 | 57.604420 | 4.778608 | 181 | 20.000000 |

调仓周期分别为1日、5日、20日的收益指标统计如下表。其中，alpha代表超额收益，即超越市场预期的收益、beta代表系统性风险，即相对于市场的波动性比例、Mean Period Wise Spread是分位数收益差，越大则区分越明显。

| | period_1 | period_5 | period_20 |
|---|---|---|---|
| Ann. alpha | 0.053 | 0.080 | 0.022 |
| beta | 1.031 | 0.917 | 1.082 |
| Mean Period Wise Return Top Quantile (bps) | 5.356 | 17.553 | 29.920 |
| Mean Period Wise Return Bottom Quantile (bps) | 19.433 | 40.523 | 36.762 |
| Mean Period Wise Spread (bps) | -14.077 | -13.755 | -0.335 |

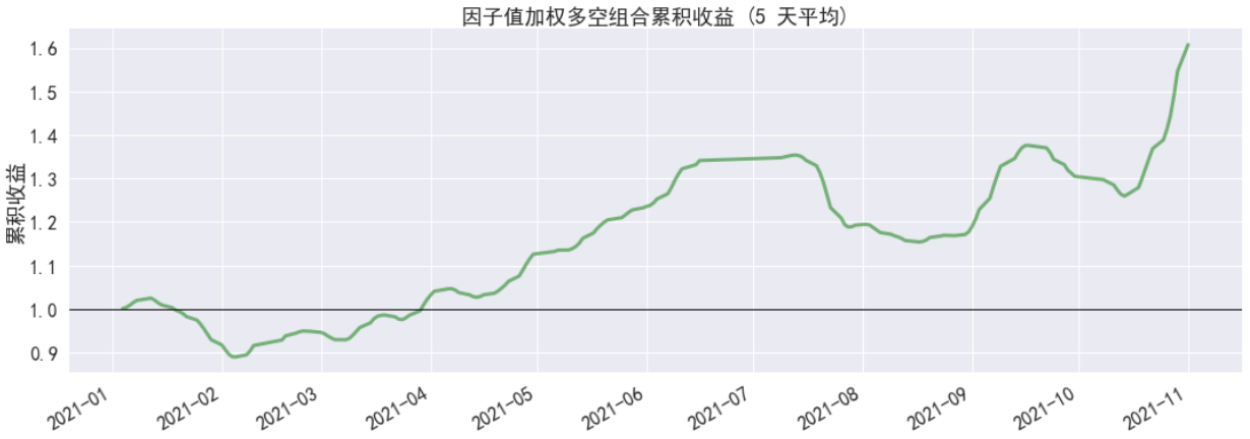
表中alpha值为正，表示利用该因子可以取得超额收益；beta值稳定在1附近，与市场总体波动持平；分位数收益差区分性不明显。

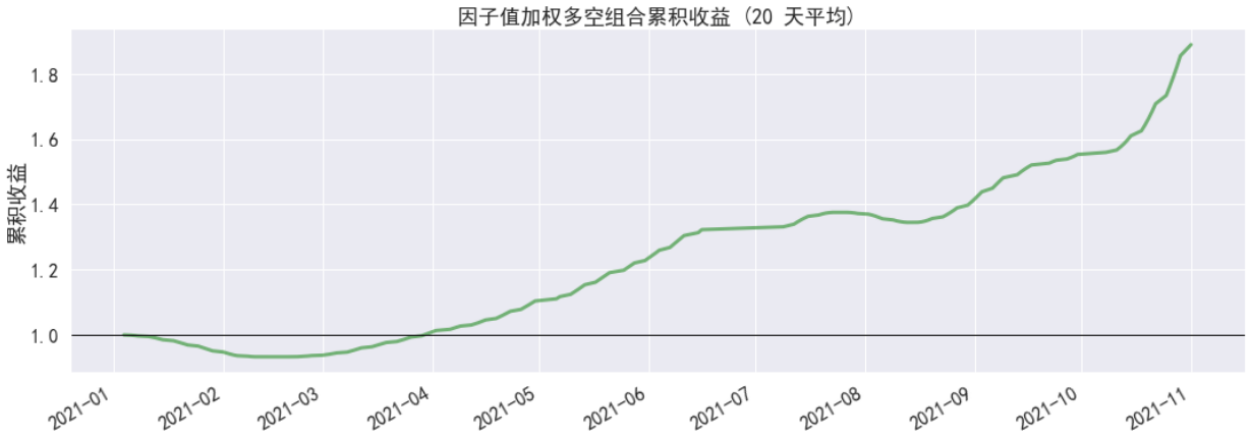
调仓周期分别为1日、5日、20日的各分位数平均收益如下图：




图中，除了因子0.2分位处1日收益外，其余周期和分位点的平均收益都为正且十分客观，其中0.8分位处平均表现最佳。

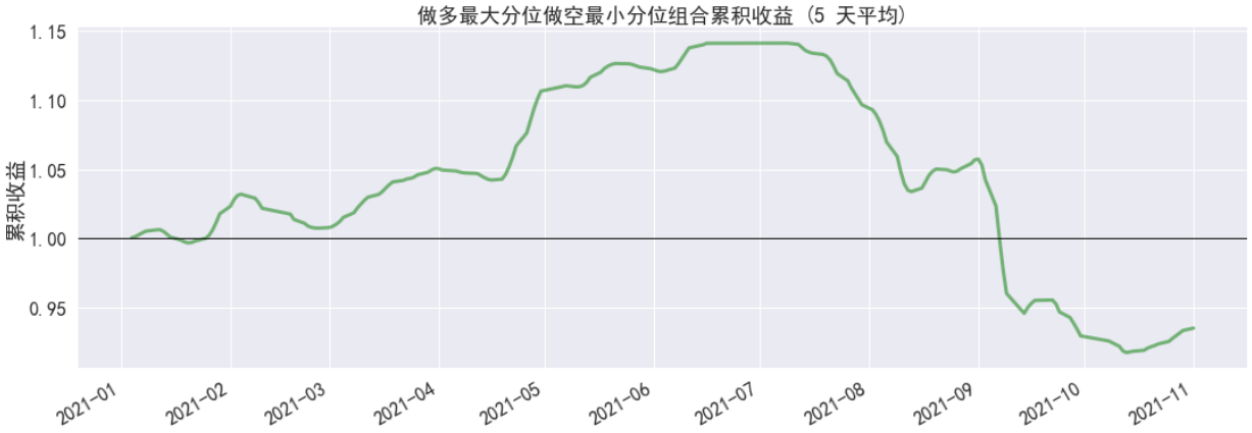
按因子值加权多空组合每日累积收益：

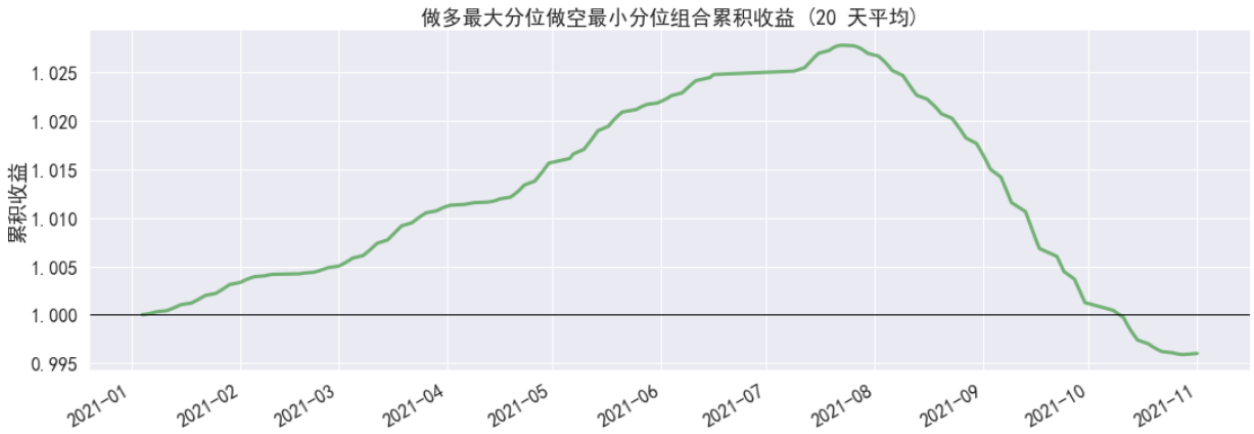



图中，各调仓周期的因子加权累积随着时间推移稳步增加，20日周期能达到至80%的收益率，且回撤小。说明该策略效果良好。

调仓周期为1日、5日、20日的做多最大分位数、做空最小分位数组合累积收益统计如下图：




图中，各调仓周期的做多最大分位数、做空最小分位数累积收益10个月以后回落，1日周期最大回撤有120%，训练使用的20日周期最大回撤为3%，差距较大。

调仓周期为1日的分位数累积收益统计如下图：


图中，0.8分位处1天累积收益最为突出，与各分位数平均收益图的结论相同。

# 四、结论

从第三节的结果与分析可以得出，由第二节设置的遗传规划方法筛选出的第一名因子基本有效。其中，按因子值加权多空组合的策略累积收益最佳，且因子值0.8分位处最有效。各调仓周期下都能获取超额收益(alpha)且基本不引入额外的风险(beta)。基于遗传规划方法的因子挖掘仍有改进的潜力。限于数据源和算力，本研究的证券池数量仅设置为5支证券，而非全部A股的股池。另一方面，遗传规划的种群数量以及进化次数也设置得较小，一定程度上可能影响到该方法的性能。算子集也仍有扩展的空间，有待引入更多有效的时间序列操作。

# 五、参考文献

[1] Chong E K P. An Introduction to Optimization[M]. 4th. 电子工业出版社, 2015-10.

[2] Trevor Stephens. gplearn’s documentation[EB/OL]. [2019.4.26]. https://gplearn.readthedocs.io.

[3] JoinQuant. JQData使用说明[EB/OL]. [2021]. https://www.joinquant.com/help/api/help#name:JQData.

[4] F'elix-Antoine Fortin. DEAP: Evolutionary Algorithms Made Easy[J]. Journal of Machine Learning Research, 2012(13):2171-2175.