一、数据：

训练集：AGTD 1-2月数据

验证集：AGTD 3月数据

测试集：AG2106 1-3月数据

二、数据处理：

1. 数据归一化：对白银价格和成交量进行均值方差归一化
2. 添加tick指标特征工程：添加中间价、资金流、一二三档量差、全体档位委托买卖量差、斜率等特征
3. 数据统计：将tick级别数据统计到秒，计算每秒白银价格的开高低收、成交量
4. 特征工程：
5. 选取过去20秒的秒统计数据
6. 选取当前时刻的盘口信息
7. tick指标特征工程
8. 技术指标特征工程
9. 选取白银期货预测标的
10. 处理后输入数据139维，输出数据1维

当前时刻的

买一到买五的价格 [best\_bid\_price\_i-0s i in range(1,6)]

卖一到卖五的价格 [best\_ask\_price\_i-0s i in range(1,6)]

买一到买五的挂单量 [best\_bid\_size\_i-0s i in range(1,6)]

卖一到卖五的挂单量 [best\_ask\_size\_i-0s i in range(1,6)]

共4\*5=20个特征

过去20s的

成交量 [qty\_-is i in range(0, 20)]

开 [open-is i in range(0, 20)]

高 [high-is i in range(0, 20)]

低 [low-is i in range(0, 20)]

收 [close-is i in range(0, 20)]

共5\*20=100个特征

Tick特征工程：7个特征

[mid\_price', 'cash\_flow\_qty', 'vol\_diff\_1', 'vol\_diff\_2', 'vol\_diff\_3', 'vol\_diff\_tot', 'sc ope']

技术指标特征工程：12个特征

['ma\_close\_-0s\_6', 'ema\_close\_-0s\_5', 'dif\_close\_-0s\_9\_12\_26', 'dea\_close\_-0s\_9\_12\_26', 'macd\_close\_-0s\_9\_12\_26', 'upper\_close\_-0s\_9\_2', 'middle\_close\_-0s\_9\_2', 'lower\_close\_-0s\_9\_2','rsi\_close\_-0s\_9', 'qty', 'obv\_qty-0s', 'bias\_close\_-0s\_9']

训练目标：5选1

[label+is for i in [5, 10, 20, 30, 60]]

1. 模型训练：
2. 遗传算法函数集设定：

gplearn自带的函数集 ['add', 'sub', 'mul', 'div', 'sqrt', 'log', 'abs', 'neg', 'inv', 'max', 'min']

自定义的函数集 [delta, delay, rank, scale, sma, stddev, product, ts\_rank, ts\_min, ts\_max, ts\_argmax, ts\_argmin, ts\_sum]

注意：使用自定义的函数集合的速度会大幅下降

1. 拟合方式

pearson: 即为IC，线性相关系数

spearman：同向相关系数

注意：使用spearman拟合的运行速度比使用pearman慢

1. 模型参数

feature\_names=features, 输入的120维数据

generations=30, 繁衍的代数

population\_size=10000, 种族的样本数

hall\_of\_fame=1000, 每次选取最优秀的样本数

n\_components=10, 模型最终输出的个体个数

function\_set=function\_set, 函数集合

p\_crossover=0.5, 交叉繁衍的概率

p\_subtree\_mutation=0.15, 子树变异的概率

p\_hoist\_mutation=0.15, hoist变异的概率

p\_point\_mutation=0.15, 点突变的概率，上述四个概率的和应该小于1

metric = 'spearman', 拟合方式

parsimony\_coefficient=0.00001, 公式复杂度的惩罚系数

max\_samples=1, 使用训练样本的比例

verbose=1, 是否在训练过程中输出信息

random\_state=0, 随机种子

n\_jobs=16 线程数

1. 挖掘到的因子

以下为实验中曾挖掘到的几个因子，IC值用fitness表示

sub(div(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_5-0s, min(div(div(div(sub(close\_-0s, best\_bid\_price\_1-0s), div(sub(close\_-0s, best\_bid\_price\_1-0s), min(best\_bid\_size\_5-0s, best\_ask\_size\_1-0s))), div(div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)), sub(best\_bid\_price\_4-0s, close\_-0s))), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(sub(best\_ask\_size\_3-0s, close\_-9s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(min(best\_ask\_size\_2-0s, open\_-13s), low\_-12s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.303426079554062, depth: 13, length: 77

sub(div(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(min(best\_ask\_size\_2-0s, open\_-13s), low\_-12s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.3041094341965509, depth: 9, length: 55

div(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_3-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.3040942525707506, depth: 8, length: 43

div(div(div(div(div(div(low\_-3s, high\_-17s), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), div(div(sub(close\_-0s, best\_bid\_price\_1-0s), sub(close\_-0s, best\_bid\_price\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))))

fitness: 0.30395080311415396, depth: 8, length: 57

div(div(div(div(div(div(sub(close\_-0s, best\_bid\_price\_1-0s), div(sub(close\_-0s, best\_bid\_price\_1-0s), min(best\_bid\_size\_5-0s, best\_ask\_size\_1-0s))), div(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), div(sub(best\_bid\_price\_4-0s, close\_-0s), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.30378812685279566, depth: 10, length: 73

div(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), sub(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(sub(best\_ask\_price\_4-0s, close\_-11s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.3039571900596223, depth: 9, length: 55

sub(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_5-0s, min(sub(best\_bid\_price\_4-0s, close\_-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), min(min(min(best\_ask\_size\_2-0s, open\_-13s), low\_-12s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.3038882410042205, depth: 8, length: 45

div(div(div(div(div(div(best\_bid\_price\_2-0s, open\_-3s), min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), div(div(div(div(sub(close\_-0s, best\_bid\_price\_3-0s), min(best\_ask\_size\_5-0s, best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), sub(close\_-0s, best\_ask\_price\_4-0s)))), sub(close\_-0s, best\_ask\_price\_4-0s))

fitness: 0.30370643070467945, depth: 8, length: 59

div(div(div(div(div(sub(best\_bid\_price\_4-0s, close\_-0s), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(min(best\_ask\_size\_3-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.30381206796869914, depth: 7, length: 45

div(div(div(div(div(div(sub(close\_-0s, best\_bid\_price\_1-0s), div(sub(best\_bid\_price\_4-0s, close\_-0s), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s)))), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))), min(best\_ask\_size\_5-0s, min(add(qty-10s, best\_ask\_size\_4-0s), best\_ask\_size\_1-0s))), div(sub(close\_-0s, best\_ask\_price\_4-0s), div(sub(best\_bid\_price\_4-0s, close\_-0s), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s))))), min(min(close\_-0s, min(min(best\_ask\_size\_2-0s, open\_-13s), best\_ask\_size\_1-0s)), best\_ask\_size\_1-0s)), div(sub(close\_-0s, best\_ask\_price\_4-0s), sub(best\_bid\_price\_4-0s, close\_-0s)))

fitness: 0.303549120557398, depth: 10, length: 65

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3042022987089461, depth: 5, length: 12

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(sqrt(sqrt(sqrt(sqrt(sqrt(best\_ask\_size\_1-0s))))))))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.30373422547205103, depth: 10, length: 17

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.30403405651307597, depth: 7, length: 14

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(sqrt(sqrt(sqrt(best\_ask\_size\_1-0s))))))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3039340329379419, depth: 8, length: 15

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(sqrt(best\_ask\_size\_1-0s))))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.30413394212251516, depth: 6, length: 13

sub(div(sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sub(best\_ask\_price\_2-0s, close\_-0s))), sub(best\_ask\_price\_2-0s, close\_-0s)), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3033194830118, depth: 6, length: 21

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3042022987089461, depth: 5, length: 12

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3042022987089461, depth: 5, length: 12

sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3042022987089461, depth: 5, length: 12

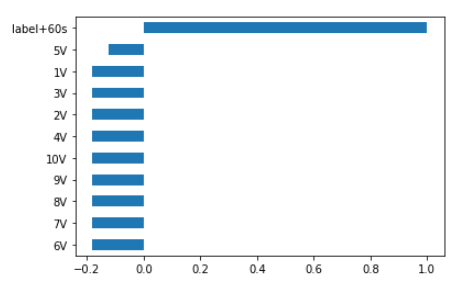
sub(div(sub(close\_-0s, best\_bid\_price\_2-0s), sqrt(sqrt(sqrt(best\_ask\_size\_1-0s)))), sub(best\_ask\_price\_2-0s, close\_-0s))

fitness: 0.3042022987089461, depth: 5, length: 12

1. 因子评测

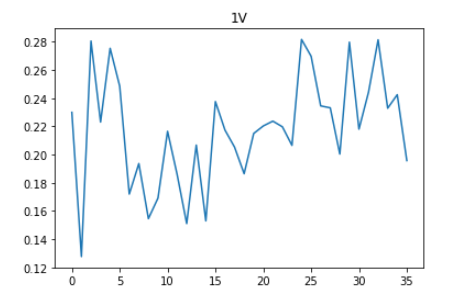
我们将单次遗传算法挖掘的到因子的前十名提取出来，并在测试集上进行因子评测

1. 因子相关性测试



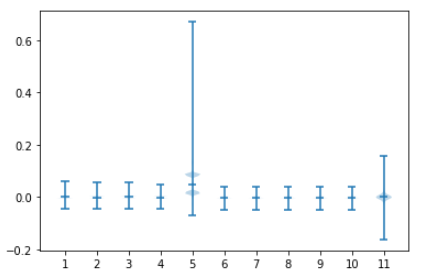
可以看出在测试集上，挖掘到的因子和实际的label值之间的IC值表现较好，呈现出较 好的相关性。其中，1V-10V表示单次遗传算法最后一批迭代中最优的十个个体在测试 集上面的IC表现。这十个样本的IC的绝对值都在0.1以上。

1. 因子时间衰减测试



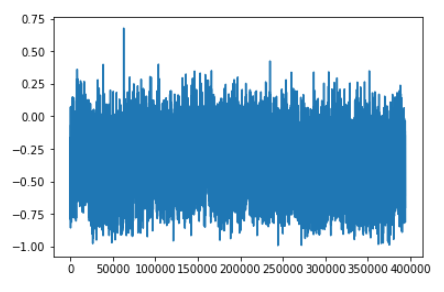
测试集共有35天的数据，我们单独取出每一天的数据，并对每一天的数据求一次当天 的日频IC，并且将这35天的日频IC绘图。可以发现该因子在这35天的时间内表现始 终较好，没有衰减趋势。

1. 因子分布箱型图

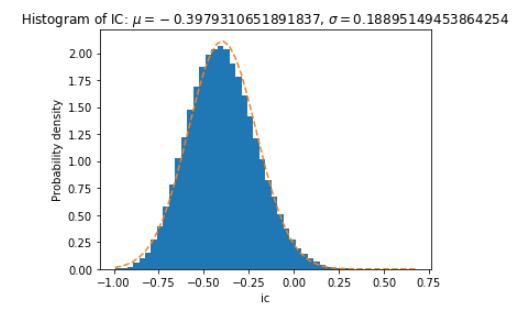


我们将挖掘到的十个因子在测试集上求值，并且将求出来的因子值统计成箱型图。可以 发现挖掘到的因子分布较为均匀。

1. 因子rolling IC及分布图

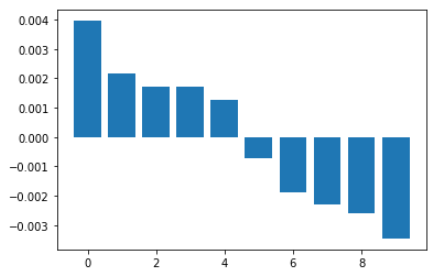
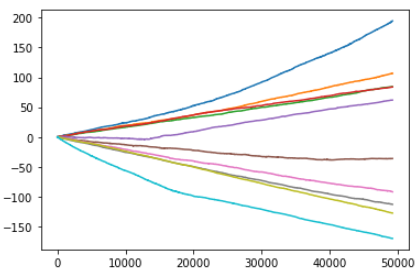


我们根据实际的使用情况，对该因子进行了rolling IC测试，即计算出了测试集每一秒 的因子值，并且和实际的label做了滑窗长度为50的 rolling IC计算。我们可以看出该 因子的rolling IC表现相对稳定，rolling IC的均值方差分布如下图：



挖掘到的因子的平均rolling IC为-0.398，与真实的label之间有很强的线性相关性；标 准差为0.189表明该因子十分稳定。

1. 分层回测



为了进一步了解挖掘到的因子的表现情况，我们对因子做了分层回测。首先对测试集上 的所有数据计算因子值，再按照因子值的大小排序，进行十等分，分别统计这十个区间 的label值作为分层回测的结果。我们可以发现该因子值的大小和实际的收益之间有很 强的相关性。