

# 回测系统说明 v2.0

tan

2015 年 9 月 7 日

## 1 策略文件

### 1.1 my\_strategy.py

```
from datetime import datetime,date
```

```
start = date(2015, 1, 1) 回测起始时间
```

```
end = date(2015, 6, 1) 回测结束时间
```

```
benchmark = '000010.sh' 策略参考标准
```

```
capital_base = 1e7 起始资金
```

```
short_capital = 1e7 做空资金
```

使用已有指数股票池则使用一下两句：

```
universe_code = '000300.SH'
```

```
self_defined = False
```

```
000002.SH 上证 A 指
```

```
000300.SH 沪深 300
```

```
000010.SH 上证 180
```

```
000016.SH 上证 50
```

```
000903.SH 中证 100
```

```
399330.SZ 深证 100
```

399312.SH 国证 300

399001.SZ 深证成指

A 全部 A 股

如果自定义股票池，则使用以下两句：

`universe_code = ['000002.sz', '000001.sz', '000004.sz']` 自定义的股票池

`self_defined = True`

`latest_universe = []` 最新的可交易股票池

```
if self_defined:
    latest_universe = universe_code
else:
    latest_universe = account.get_latest_universe()
```

更新股票池，自定义的就等于初始自定义

`def initialize(account):` 初始化属性

`def stop_loss(account):` 止损策略，在 `handle_data` 中调用

`def handle_data(account):` 这里写选股策略

以上列出的基本属性都要有，因为 `account` 类初始化会用到。

## 2 Account 类属性

1. `account.days_counts` 调用 `handle_data` 的次数，也就是第几个交易日
2. `account.current_time` 当前历史交易日的时间，类型为 `datetime`，可通过 `account.current_time.month` 等查看月份啊等等
3. `account.last_time` 类似 `current_time`，为上次交易日的时间
4. `account.tomorrow` 类似 `current_time`，为下一个交易日的时间
5. `account.all_universe` 初始化时指定的交易股票池，类型为 `list`
6. `account.universe` 当前交易日将 `all_universe` 中停牌，跌停等不可交易的股票剔除后剩余的股票
7. `account.cash` 当前的现金
8. `account.valid_secpos` 为一个字典，key 为股票代码，value 是对应手头上持有的该股票的数量，如果要得到所有持仓数目大于零的股票代码，可以这样写 `account.valid_secpos.index[account.valid_secpos>0]` 这是 `pandas.DataFrame` 的写法
9. `account.avgBuyprice` 为一个字典，key 为股票代码，value 是对应买该股票时的平均价格
10. `account.buyPosition` 为一个字典，key 为股票代码，value 是对应最近购买该股票时的时间单位，与 `days_counts` 对应，`=days_counts` 即为当天购买，`>`，`<`....

11. `account.is_upORdownLimit(code)` 判断该股票是否涨，跌停
12. `account.isOK2order(code)` 判断该股票当天是否能买卖
13. `account.getTodayOpen(code)` 该股票的当天开盘价，float 类型
14. `account.getTodayHigh(code)`
15. `account.getTodayLow(code)`
16. `account.getDailyHistory(code,length)` 得到历史数据，length 为时间长度，例如 `getDailyHistory('000007.sz',5)['close'].values[-1]` 得到前一交易日的 close 价格，`values[-1]` 为最后，也就是前一交易日，如果是 -2 则是前两个交易日，等等。。不过不能超过 length。除了 close，还有 `date_time,open,high,low` 等，就是数据库对应的表的几个属性
17. `account.order(code,num,price)` 下单函数，买多少手。code 股票代码，num 手数，>0 为买入，<0 为卖出，price 买卖价格
18. `account.order_to(code,num,price)` 下单函数，买或者卖到 num 手。code 股票代码，num 想让账户上持有该股票的数目
19. `account.isBeginOfMonth(t)` 判断是否月的第一天,t 为时间，类型是 `datetime.date`
20. `account.getBeginOfMonth(y,m)` 得到某年某月的第一个交易日，注意两个参数必须为 int 类型

21. `account.static_profit` 字典，对应股票的静态收益，如要清空，请这样
22. `account.DR` pandas 的 `DataFrame` 类型，具体有回测期间指定日期如“2015-01-04”的 `cash`, `capital`, `IF`, `alpha_capital`
23. `account.monthly_profit` 字典，记录月收益，获取上个月的收益可以用“`last_time`”或者指定的月份如“2015-02”

## 3 递推

### 3.1 my\_recursive.py

rate\_list = np.linspace(0,0.2,3) 设定止损率列表

rec = Recursive(train\_count=3,use\_count=2)

设定训练及使用日期区间如以上为使用前 3 个月的最佳止损参数应用于下 2 个月的止损。

需要递推时则运行此文件，否则默认运行 my\_test.py 文件

### 3.2 my\_recursive\_pp.py

此文件为递推的分布式多进程文件，数据多时间久就运行此文件。

## 4 指标计算

### 4.1 my\_performance.py

基准，策略年化收益率，基准年化收益率，alpha，beta，策略波动收益率，夏普率，信息率，最大回撤率，以及对比图