

基于 vnpy 的回测使用说明

目前实现两个策略：strategyGirdTrading 和 ctaTradeTest 的回测。

具体流程：

1、strategyGirdTrading

1) 打开终端，切换工作目录：

```
cd ~/Downloads/raptor-dev/ctaAlgo
```

2) 相关回测参数可在当前文件夹下的 parameter_testS.json 中修改（如 buyPrice 等），参数文件具体内容如下：

```
{
  "PriceCoe": 1,
  "buyPrice": [
    3200,
    3300,
    3500,
    3600
  ],
  "closeFirst": false,
  "direction": "short",
  "isFilter": false,
  "isStop": true,
  "maxStpLos": 0,
  "openUnit": 1,
  "postoday": {
    "rb1710": 4
  },
  "receivers": [],
  "slippage": 5,
  "stoptime": "9999",
  "stpProfit": 2
}
```

3) 在终端输入 python 进入 python 环境。

4) 依次输入如下语句：

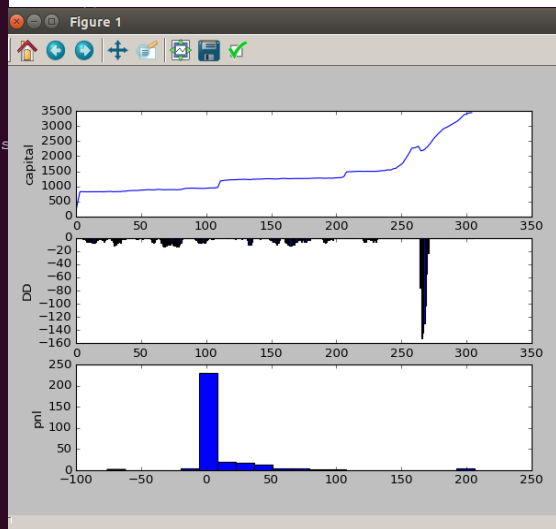
```
from ctaBacktesting import *
from strategyGirdTrading import *
engine = BacktestingEngine()
engine.setBacktestingMode(engine.TICK_MODE)
engine.setStartDate('20160401')
engine.loadHistoryData('backtesting', 'AG01')    #在此处修改合约名
engine.initStrategy(strategyGirdTrading, {"className": "theGirdTrading",
"name": "testS", "vtSymbol": "rb1710"})    #此策略中此处可不作修改
engine.runBacktesting()
engine.showBacktestingResult()
```

5) 运行结果样例（以合约 AG01 为例）

```

>>> from ctaBacktesting import *
>>> from strategyGirdTrading import *
>>> engine = BacktestingEngine()
>>> engine.setBacktestingMode(engine.TICK_MODE)
>>> engine.setStartDate('20160401')
>>> engine.loadHistoryData('backtesting', 'Y05')
开始载入数据
载入完成，数据量：27691
>>> engine.initStrategy(strategyGirdTrading, {"class
vtSymbol
vtSymbol
vtSymbol
vtSymbol ['rb1710']
>>> engine.runBacktesting()
开始回测
策略初始化完成
策略启动完成
开始回放数据
数据回放结束
>>> engine.showBacktestingResult()
显示回测结果
-----
第一笔交易时间：2016-04-18 22:29:54
最后一笔交易时间：2016-04-29 14:57:19
总交易次数：305
总盈亏：3440.0
最大回撤：-152.0

```



2、ctaTradeTest

- 1) 打开终端，切换工作目录：

`cd ~/Downloads/raptor-dev/ctaAlgo`

- 2) 相关回测参数可在当前文件夹下的 `parameter_testD.json` 中修改 (如 `buyPrice`、`vtSymbol` 等)，参数文件具体内容如下：

```

{
  "buyPrice": [
    3200,
    3300,
    3400,
    3500,
    3600
  ],
  "closeFirst": true,
  "isFilter": false,
  "longBuyUnit": 1,
  "longPriceCoe": 1,
  "postoday": {
    "AG05": 2,
    "AG01": 2
  },
  "receivers": [],
  "shortBuyUnit": 1,
  "shortPriceCoe": 1,
  "slippage": 5,
  "stoptime": "9999",
  "stpProfit": 2
}

```

- 3) 在终端输入 `python` 进入 `python` 环境。
- 4) 依次输入如下语句：

```

from ctaBacktesting import *
from ctaTradeTest import *
engine = BacktestingEngine()
engine.setBacktestingMode(engine.TICK_MODE)

```

```

engine.setStartDate('20160401')
engine.loadHistoryData2('backtesting', 'AG05', 'AG01') #在此处修改合约名
engine.initStrategy(tradeTest,{"className": "tradeTest", "longSymbol": "AG05", "name": "testD", "vtSymbol": "AG05,AG01", "shortSymbol": "AG01"})
# 此处 longSymbol、shortSymbol 和 vtSymbol 需要修改,注意需要与参数文件中对应
engine.runBacktesting()
engine.showBacktestingResult()

```

5) 运行结果样例（以合约 AG05 和 AG01 为例）

```

>>> from ctaBacktesting import *
>>> from ctaTradeTest import *
>>> engine = BacktestingEngine()
>>> engine.setBacktestingMode(engine.TICK_MODE)
>>> engine.setStartDate('20160401')
>>> engine.loadHistoryData2('backtesting', 'AG05', 'AG01')
开始载入数据
载入完成，数据量：4834
>>> engine.initStrategy(tradeTest,{"className": "tradeTest",
AG01", "shortSymbol": "AG01"})
vtSymbol
vtSymbol
vtSymbol
vtSymbol ['AG05', 'AG01']
>>> engine.runBacktesting()
开始回测
策略初始化完成
策略启动完成
开始回放数据
trade AG01 0 0
trade AG01 0 0
trade AG05 0 0
trade AG05 0 0
数据回放结束
>>> engine.showBacktestingResult()
显示回测结果
-----
第一笔交易时间：2016-04-19 21:22:33
最后一笔交易时间：2016-04-19 21:22:33
总交易次数：1
总盈亏：366.0
最大回撤：0.0

```

