https://www.myquant.cn/docs/python\_strategyies/426

**菲阿里四价(期货)**

分享

 阅读 35968  更新 2022-05-09 16:56:48

**菲阿里四价**

**1. 原理**

菲阿里四价同R Breaker一样，也是一种日内策略交易，适合短线投资者。

菲阿里四价指的是：昨日高点、昨日低点、昨天收盘、今天开盘四个价格。

菲阿里四价上下轨的计算非常简单。昨日高点为上轨，昨日低点为下轨。当价格突破上轨时，买入开仓；当价格突破下轨时，卖出开仓。

**2. 策略逻辑**

第一步：获取昨日最高价、最低价、收盘价、开盘价四个数据。  
第二步：计算上轨和下轨。当价格上穿上轨时，买入开仓；当价格下穿下轨时，卖出开仓。  
第三步：当日平仓。

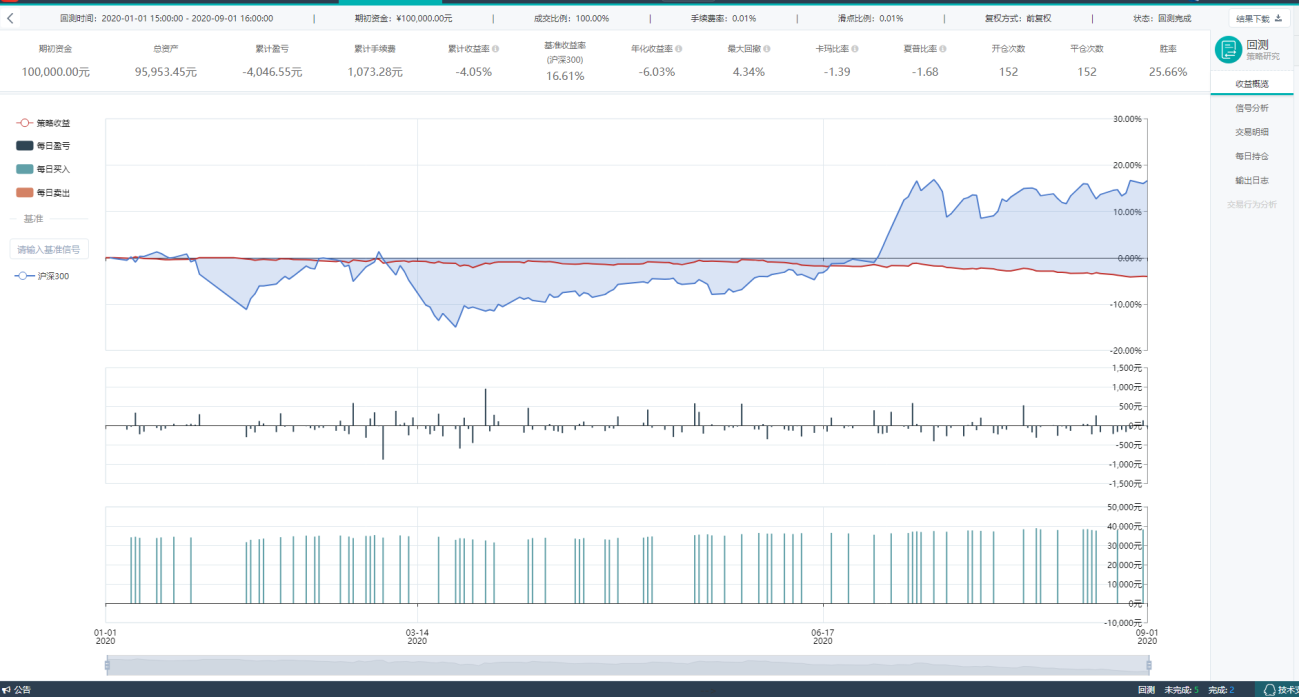
回测标的：SHFE.rb2010  
回测期：2020-02-07 至 2020-04-15  
回测初始资金：200万  
**注意：若修改回测期，需要修改对应的回测标的。**

**3. 策略代码**

1. # coding=utf-8
2. from \_\_future\_\_ import print\_function, absolute\_import
3. from gm.api import \*
4. """
5. 上轨=昨日最高点；
6. 下轨=昨日最低点；
7. 止损=今日开盘价;
8. 如果没有持仓，且现价大于了昨天最高价做多，小于昨天最低价做空。
9. 如果有多头持仓，当价格跌破了开盘价止损。
10. 如果有空头持仓，当价格上涨超过开盘价止损。
11. 选取 进行了回测。
12. 注意：
13. 1：为回测方便，本策略使用了on\_bar的一分钟来计算，实盘中可能需要使用on\_tick。
14. 2：实盘中，如果在收盘的那一根bar或tick触发交易信号，需要自行处理，实盘可能不会成交。
15. """
16. def init(context):
17. # 设置标的
18. context.symbol = 'SHFE.rb2010'
19. # 订阅一分钟线
20. subscribe(symbols = context.symbol,frequency = '60s',count = 1)
21. # 记录开仓次数，保证一天只开仓一次
22. context.count = 0
23. # 记录当前时间
24. time = context.now.strftime('%H:%M:%S')
25. # 如果当前时间点是交易时间段，则直接执行algo获取历史数据，以防直接进入on\_bar()导致context.history\_data未定义
26. if '09:00:00' < time < '15:00:00' or '21:00:00' < time < '23:00:00':
27. algo(context)
28. # 如果是非交易时间段，等到上午9点或晚上21点再执行algo()
29. schedule(schedule\_func = algo, date\_rule = '1d', time\_rule = '09:00:00')
30. schedule(schedule\_func = algo, date\_rule = '1d', time\_rule = '21:00:00')
31. def algo(context):
32. # 获取历史的n条信息
33. context.history\_data = history\_n(symbol=context.symbol, frequency = '1d', end\_time = context.now,
34. fields='symbol,open,high,low',count=2, df=True)
35. def on\_bar(context,bars):
36. # 取出订阅的一分钟bar
37. bar = bars[0]
38. # 提取数据
39. data = context.history\_data
40. # 现有持仓情况
41. position\_long = context.account().position(symbol=context.symbol, side=PositionSide\_Long)
42. position\_short = context.account().position(symbol=context.symbol, side=PositionSide\_Short)
43. # 如果是回测模式
44. if context.mode == 2:
45. # 开盘价直接在data最后一个数据里取到,前一交易日的最高和最低价为history\_data里面的倒数第二条中取到
46. open = data.loc[1, 'open']
47. high = data.loc[0, 'high']
48. low = data.loc[0, 'low']
49. # 如果是实时模式
50. else:
51. # 开盘价通过current取到
52. open = current(context.symbol)[0]['open']
53. # 实时模式不会返回当天的数据，所以history\_data里面的最后一条数据是前一交易日的数据
54. high = data.loc[-1, 'high']
55. low = data.loc[-1, 'low']
56. # 交易逻辑部分
57. if position\_long: # 多头持仓小于开盘价止损。
58. if bar.close < open:
59. order\_volume(symbol=context.symbol, volume=1, side=OrderSide\_Sell,
60. order\_type=OrderType\_Market, position\_effect=PositionEffect\_Close)
61. print('以市价单平多仓')
62. elif position\_short: # 空头持仓大于开盘价止损。
63. if bar.close > open:
64. order\_volume(symbol=context.symbol, volume=1, side=OrderSide\_Buy,
65. order\_type=OrderType\_Market, position\_effect=PositionEffect\_Close)
66. print('以市价单平空仓')
67. else: # 没有持仓。
68. if bar.close > high and not context.count: # 当前的最新价大于了前一天的最高价
69. # 开多
70. order\_volume(symbol=context.symbol, volume=1, side=OrderSide\_Buy,
71. order\_type=OrderType\_Market, position\_effect=PositionEffect\_Open)
72. print('以市价单开多仓')
73. context.count = 1
74. elif bar.close < low and not context.count: # 当前最新价小于了前一天的最低价
75. # 开空
76. order\_volume(symbol=context.symbol, volume=1, side=OrderSide\_Sell,
77. order\_type=OrderType\_Market, position\_effect=PositionEffect\_Open)
78. print('以市价单开空仓')
79. context.count = 1
80. # 每天收盘前一分钟平仓
81. if context.now.hour == 14 and context.now.minute == 59:
82. order\_close\_all()
83. print('全部平仓')
84. context.count = 0
85. if \_\_name\_\_ == '\_\_main\_\_':
86. '''
87. strategy\_id策略ID,由系统生成
88. filename文件名,请与本文件名保持一致
89. mode实时模式:MODE\_LIVE回测模式:MODE\_BACKTEST
90. token绑定计算机的ID,可在系统设置-密钥管理中生成
91. backtest\_start\_time回测开始时间
92. backtest\_end\_time回测结束时间
93. backtest\_adjust股票复权方式不复权:ADJUST\_NONE前复权:ADJUST\_PREV后复权:ADJUST\_POST
94. backtest\_initial\_cash回测初始资金
95. backtest\_commission\_ratio回测佣金比例
96. backtest\_slippage\_ratio回测滑点比例
97. '''
98. run(strategy\_id='strategy\_id',
99. filename='main.py',
100. mode=MODE\_BACKTEST,
101. token='token\_id',
102. backtest\_start\_time='2020-01-01 15:00:00',
103. backtest\_end\_time='2020-09-01 16:00:00',
104. backtest\_adjust=ADJUST\_PREV,
105. backtest\_initial\_cash=100000,
106. backtest\_commission\_ratio=0.0001,
107. backtest\_slippage\_ratio=0.0001)

**4. 回测结果与稳健性分析**

设定初始资金10万，手续费率为0.01%，滑点比率为0.01%。回测结果如图所示：



回测期累计收益率为-4.05%，年化收益率为-6.03%。沪深300指数收益率为16.61%，整体跑输指数。最大回撤为4.34%，胜率为25.66%。

为了验证策略的稳健性，更改标的，回测结果如下：

| **标的** | **回测期** | **年化收益率** | **最大回撤** |
| --- | --- | --- | --- |
| SHFE.rb2010 | 2020.02.07-2020.04.15 | -6.03% | 4.34% |
| SHFE.ag2010 | 2020.02.07-2020.04.15 | 38.03% | 7.97% |
| SHFE.ni2010 | 2020.02.07-2020.04.15 | -22.46% | 22.85% |
| SHFE.zn2010 | 2020.02.07-2020.04.15 | -15.93% | 12.01% |
| SHFE.rb2010 | 2020.04.07-2020.06.15 | -3.40% | 1.37% |
| SHFE.rb2010 | 2020.06.07-2020.08.15 | -13.96% | 2.32% |
| SHFE.rb2010 | 2020.08.07-2020.10.15 | -11.38% | 2.18% |

由上表可以看出，随着品种的变化，策略的收益变化差异较大。ag2010的收益率能达到38.03%，但ni2010的收益仅为-22.46%。对于rb2010，随着回测期的变化，年化收益率均为负值，收益较不稳定。

**注：此策略只用于学习、交流、演示，不构成任何投资建议。**