Backtrader量化平台教程（三）Indicator

2019-01-28 阅读 2.4K

AD：（本人录制的backtrader视频课程，大家多多支持哦~ https://edu.csdn.net/course/detail/9040）

前面两篇文章，讲了大致的框架，接下来涉及的更多的是细节。本文介绍了backtrader中的indicator，并讲述了一些别的细节的代码。所谓indicator就是技术指标，比如MA，RSI

1.预备

在介绍backtrader的indicator之前，我们先配置一下我们的平台，也就是cerebro。

```
if __name__ == '__main__':
    # Create a cerebro entity
    cerebro = bt.Cerebro()
    # Add a strategy
    cerebro.addstrategy(TestStrategy)
    # 本地数据，笔者用Wind获取的东风汽车数据以csv形式存储在本地。
    # parase_dates = True是为了读取csv为dataframe的时候能够自动识别
    # 注意，这里最后的pandas要符合backtrader的要求的格式
    dataframe = pd.read_csv('dfqc.csv', index_col=0, parse_dat
    dataframe['openinterest'] = 0
    data = bt.feeds.PandasData(dataname=dataframe,
                               fromdate = datetime.datetime(2015,
                               todate = datetime.datetime(2016, 1
                               )
    # Add the Data Feed to Cerebro
    cerebro.adddata(data)
    # Set our desired cash start
    cerebro.broker.setcash(100.0)
    # 设置每笔交易交易的股票数量
    cerebro.addsizer(bt.sizers.FixedSize, stake=10)
    # Set the commission
    cerebro.broker.setcommission(commission=0.0)
    # Print out the starting conditions
    print('Starting Portfolio Value: %.2f' % cerebro.broker.ge
    # Run over everything
    cerebro.run()
    # Print out the final result
    print('Final Portfolio Value: %.2f' % cerebro.broker.getva
    cerebro.plot()
```

这里出现了

```
# 设置每笔交易交易的股票数量
    cerebro.addsizer(bt.sizers.FixedSize, stake=10)
```

这个函数，用来设置每次下单的时候买卖的股票数量。

```
bt.sizers.FixedSize
```

就告诉平台，我们是每次买卖股票数量固定的，stake=10就是10股。当然，实际过程中，我们不可能如此简单的制定买卖的数目，而是要根据一定的规则，这就需要自己写一个sizers，这是后话。

目录

作者介绍

钱塘小甲子

关注 专栏

文章	阅读量	获赞	作者排名
197	167.6K	623	902

策略的完整生命周期如下：

0.__init__

这个是肯定的，任何类在生成的时候都是先调用这一初始化构造函数。也就是说，在实例生成的时候，这个函数将被调用。

1.Birth: start

start方法在cerebro告诉strategy，是时候开始行动了，也就是说，通知策略激活的时候被调用。

2.Childhood: prenext

有些技术指标，比如我们提到的MA，存在一个窗口，也就是说，需要n天的数据才能产生指标，那么在没有产生之前呢？这个prenext方法就会被自动调用。

3.Adulthood: next

这个方法是最核心的，就是每次移动到下一的时间点，策略将会调用这个方法，所以，策略的核心往往都是写在这个方法里的。

4.Death: stop

策略的生命周期结束，cerebro把这一策略退出。

2.2策略当中的回调函数

Strategy 类就像真实世界的交易员一样，当交易执行的时候，他会得到一些消息，譬如order是否执行，一笔trader赚了多少钱，等等。这些消息都将在Strategy类中通过回调函数被得以知晓。这些回调函数如下：

notify_order(order)：下的单子，order的任何状态变化都将引起这一方法的调用

notify_trade(trade)：任何一笔交易头寸的改变都将调用这一方法

notify_cashvalue(cash, value)：任何现金和资产组合的变化都将调用这一方法

notify_store(msg, *args, **kwargs)：可以结合cerebro类进行自定义方法的调用

那么问题接踵而至，这里我们只关注前2种方法中监测对象的可变化方式。

trade指的是一笔头寸，trdae是open的状态指当前时刻，这一标的的头寸从0变到某一非零值。trade是closed则刚好相反。

trade大概有如下常用属性

ref: 唯一id
size (int): trade的当前头寸
price (float): trade资产的当前价格
value (float): trade的当前价值
commission (float): trade的累计手续费
pnl (float): trade的当前pnl
pnlcomm (float): trade的当前pnl减去手续费
isclosed (bool): 当前时刻trade头寸是否归零
isopen (bool): 新的交易更新了trade
justopened (bool): 新开头寸
dtopen (float): trade open的datetime
dtclose (float): trade close的datetime

Orders

order是strategy发出的指令，让cerebro去执行。

strategy自身有buy, sell and close方法来生成order，cancel方法来取消一笔order。下单的方式有很多，后续会介绍，这里主要讲回调函数中，咱们可以获得哪些信息。

order.status可以返回order的当前状态

order.isbuy可以获得这笔order是否是buy

order.executed.price
order.executed.value

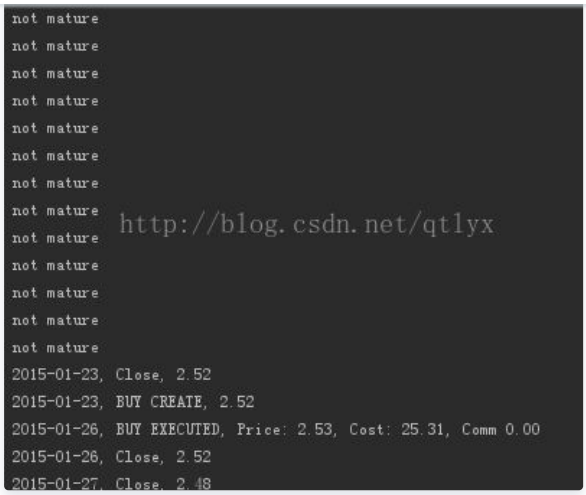
目录

2.3代码

```
class TestStrategy(bt.Strategy):
    params = (
        ('maperiod', 15),
    )

    def log(self, txt, dt=None):
        ''' Logging function fot this strategy'''
        dt = dt or self.datas[0].datetime.date(0)
        print('%s, %s' % (dt.isoformat(), txt))

    def __init__(self):
        # Keep a reference to the "close" line in the data[0]
        self.dataclose = self.datas[0].close

        # To keep track of pending orders and buy price/commis
        self.order = None
        self.buyprice = None
        self.buycomm = None

        # Add a MovingAverageSimple indicator
        self.sma = bt.indicators.SimpleMovingAverage(
            self.datas[0], period=self.params.maperiod)
    def start(self):
        print("the world call me!")

    def prenext(self):
        print("not mature")

    def notify_order(self, order):
        if order.status in [order.Submitted, order.Accepted]:
            # Buy/Sell order submitted/accepted to/by broker -
            return

        # Check if an order has been completed
        # Attention: broker could reject order if not enougth
        if order.status in [order.Completed]:
            if order.isbuy():
                self.log(
                    'BUY EXECUTED, Price: %.2f, Cost: %.2f, Co
                    (order.executed.price,
                     order.executed.value,
                     order.executed.comm))

                self.buyprice = order.executed.price
                self.buycomm = order.executed.comm
            else:  # Sell
                self.log('SELL EXECUTED, Price: %.2f, Cost: %.
                         (order.executed.price,
                          order.executed.value,
                          order.executed.comm))

            self.bar_executed = len(self)

        elif order.status in [order.Canceled, order.Margin, or
            self.log('Order Canceled/Margin/Rejected')

        self.order = None
```

大家可以看到打印出来的结果中，有start和prenext，最后当然也有death

目录



3.Backtrader的indicator

上面的代码中，我们单独拿出init这一部分，因为这里涉及了一个新的东西，indicator，也是本文想重点介绍的。

```
def __init__(self):
    # Keep a reference to the "close" line in the data[0]
    self.dataclose = self.datas[0].close

    # To keep track of pending orders and buy price/commis
    self.order = None
    self.buyprice = None
    self.buycomm = None

    # Add a MovingAverageSimple indicator
    self.sma = bt.indicators.SimpleMovingAverage(
        self.datas[0], period=self.params.maperiod)
```

这里的最后，我们使用了一个backtrader内置的indicator，后续我们将尝试自己编写一个indicator。

编程算法 举报

目录

相关文章

Backtrader量化平台教程（八）TimeFrame

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程（七）Optimizer

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程（五）Signal

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程（六）Analyzer

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程（二）：Strategy类

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程（一）:backtrader的整体框架

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csd...

钱塘小甲子

Backtrader量化平台教程（四）SSA策略实际案例

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

Backtrader量化平台教程-跟踪止损单（十）

AD：（本人录制的backtrader视频课程，大家多多支持哦 ~ https://edu.csdn.net/course/detail/9040）

钱塘小甲子

目录

~ https://edu.csdn.net/course/detail/9040)

钱塘小甲子

Backtrader量化平台教程-作者的一篇博客（十一）

Backtrader的作者在他的博客上写了一篇很有意思的文章。这个哥们从csdn上面找了backtrader的代码，然后改写了一下，提高了可读性，觉...

钱塘小甲子

Backtrader来啦：可视化篇（重构）

量化投资与机器学习公众号 独家撰写 前言 今天的《可视化篇》先会介绍与可视化相关的观测器模块 observers ， 然后介绍 Backtrader 自带的绘图...

量化投资与机器学习微信公...

Backtrader来啦：策略篇

公众号为全网读者带来Backtrader系列自推出第一期以来，
受到了众多读者的喜爱与点赞，QIML也会继续把这个系列...

量化投资与机器学习微信公...

pyalgotrade教程1--第一个demo

之前一直使用backtrader作为回测的平台，但是近来觉得，backtrader虽然在有些设计上很精妙，但是官方demo中都有很多细节性的错误...

钱塘小甲子

一个alpha量化的开源项目--Signal_Report_Platform（单因子...

目前，网上其实有很多量化的回测平台，比如之前笔者写过教程的backtrader和pyalgotrade，当然还有大名鼎鼎的zipline。但是值得注意的是，这些...

钱塘小甲子

Backtrader 来了！

Backtrader 是 2015 年开源的 Python 量化回测框架（支持
实盘交易），功能丰富，操作方便灵活：

量化投资与机器学习微信公...

Backtrader来啦：数据篇

此系列将由浅入深，每期1~2周，大家敬请期待！ 前言 阅读
完上一篇Backtrader 来了后，不知大家心里是否有如下疑...

量化投资与机器学习微信公...

量化投资教程：用R语言打造量化分析平台

？概述 和Python计算环境中的tushare包一样，在R中我们使
用quantmod包接入第三方数据源，实现自定义量化分析平...

CDA数据分析师

目录

近来忙于毕业找工作，也不知道能不能继续在量化界混了。周末比较闲，抽空研究了一下vn.py。有人说，为什么学那么多的回测平台呀。其实我个人觉得，做...

钱塘小甲子

【推荐收藏】倾心整理的Python量化资源大...

随着Python编程语言的流行和普及，越来越多人对如何应用Python做金融数据分析和量化交易充满兴趣。但是不少人...

量化小白

更多文章

目录