

Backtrader (十五) - Order订单 - 自定义Sizer类控制下单量

2022-01-26 22:29:10 【roy1126_】

简介

通过向cerebro传入自定义Sizer, 修改buy/sell的执行逻辑, 控制下单量等。

在自定义的Sizer类中, 可以访问策略对象`self.strategy`, 以及策略的经纪行对象`self.strategy.broker(self.broker)`, 通过broker可以访问仓位 (`self.broker.getposition(data)`) 和市值(`self.broker.getvalue()`)等信息。

在Sizer类中的 `def _getsizing(self, comminfo, cash, data, isbuy):`

方法中, 可通过其参数访问更多信息, 包括佣金信息, 现金, 行情数据。

其中isbuy参数确定订单是买单还是卖单。该方法需要返回 `int`类型表示下单量, 如果返回 0. 则不执行订单操作。

将自定义的Sizer注入Cerebro

有两种方法注入sizer到cerebro

1、所有策略都会使用该Sizer

```
cerebro.addsizer(LongOnly) # 全部策略使用自定义sizer
```

2、按策略id设置Sizer, 给不同的策略设置不同的Sizer

```
idx = cerebro.addstrategy(SmaCross)
cerebro.addsizer_byidx(idx, maxRiskSizer)
```

示例

```
'''自定义sizer'''
import math

import backtrader as bt
from feed import feed
from logger import lg

class maxRiskSizer(bt.Sizer):
    params = (
        ('risk', 0.03),
    )

    def __init__(self):
        if self.p.risk > 1 or self.p.risk < 0:
            raise ValueError(
                'The risk parameter is a percentage which must be enter as a float: e.g. 0.5'
            )

    def _getsizing(self, comminfo, cash, data, isbuy):
        if isbuy:
            size = math.floor(
                (cash * self.p.risk) / data[0]
            )
        else:
            size = math.floor(
                (cash * self.p.risk) / data[0]
            ) * -1

        return size
```

边栏推荐

- 1 DVWA靶场通关
- 2 贪吃蛇js部分 框架
- 3 低代码平台竞品分
- 4 ESB项目培训样
- 5 Mysql SQLyog连
- 6 Docker 下 Redis
- 7 笃行致远, 人大
- 8 软考高项笔记 | 项
- 9 Handler全解
- 10 平均年薪60.8万,

猜你喜欢



【阅读笔记】
Personalized



LeetCode
配对次数[

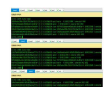
es6--箭头函数



Go 入门很
文本文件



生产者消费
问题



Redis学习

pyhton XGBoost交叉验
——极简源码

程序设计基础综合实验
管理系统

字符串的定义和输入、

C语言指针与二维数组

随机推荐

CTFHub - 密码口令

科研踩坑: 角度制三
乘上π/180

```

class LongOnly(bt.Sizer):
    '''自定义订单管理类 1、默认股数 100 2、不能做空'''
    params = (
        ('stake', 100),
    )

    def _getsizing(self, comminfo, cash, data, isbuy):
        if isbuy:
            return self.p.stake
        # 处理卖单情况, 先获取仓位对象
        position = self.broker.getposition(data)
        # 返回卖单的下单量, 防止形成做空
        return min(self.p.stake, position.size)

class SmaCross(bt.Strategy):
    params = (
        ('period', 5),
    )

    def __init__(self):
        self.move_average = bt.ind.MovingAverageSimple(
            self.data.close,
            period=self.p.period
        )
        self.crossover = bt.ind.CrossOver(
            self.data.close,
            self.move_average
        )

    def log(self, txt, dt=None):
        dt = dt or self.data.datetime.date(0)
        if isinstance(dt, float):
            dt = bt.num2date(dt)
        lg.info(
            '%s, %s' % (dt.isoformat(), txt)
        )

    def notify_order(self, order):
        if order.status in [order.Submitted, order.Accepted]:
            return
        if order.status in [order.Completed]:
            if order.isbuy():
                self.log(f'买单执行, price {
                    order.executed.price}, size {
                    order.executed.size}')
            elif order.issell():
                self.log(f'卖单执行, price {
                    order.executed.price}, size {
                    order.executed.size}')

        else:
            self.log(f'订单状态 {
                order.getstatusname(order.status)}')

    def notify_trade(self, trade):
        if trade.isclosed:
            self.log(
                f'毛收益 {
                    trade.pnl}, 扣佣后收益 {
                    trade.pnlcomm}, 佣金 {
                    trade.commission}'
            )

    def next(self):
        if self.crossover > 0:
            self.buy()
        if self.crossover < 0:

```

ConvNeXt网络模型

未知轴的四元数变換及从b到a

Socket笔记(一)

GameFi会是Polygo吗? | Footprint Ana

2021年WAX处在2022年能否继续维持Footprint Analytics

游戏公会并喷, 谁在YGG? | Footprint A

webStorage(浏览器

快到年底了, 现在可产品?

Redis实现分布式锁

数据库设计 3 个范

数据库中主键、外键

JDBC, Statement、PreparedStatement

零基础自学, 谁说自

Fiddler 环境搭建及

Jmeter 线程组/配置

830. 单调栈

力扣 167. 两数之和数组

C基础(一)数据类型符、语句

操作表DML相关语

WPF 使用async 作

【蛇形填空】03

【七段码】05

每周全球科技十大看(2022.1.17-1.23)

【既约分数】02

锅都不敢背, 凭什干?

一文看懂回归问题、度下降与神经元模型深度学习笔记》第2整理系列)

Mysql -- opérations

Mysql -- extraction dans le champ

科研踩坑: 角度制三乘上 $\pi/180$ Recherche scientifique dérivation de la fonction trigonométrique du d'angle de 0° à 180°

Ctfhub - mot de pas

Pointeur de langage bidimensionnel

Définition, entrée et des chaînes

```
self.close()
```

```
if __name__ == '__main__':
    cerebro = bt.Cerebro()
    cerebro.adddata(feed)
    idx = cerebro.addstrategy(SmaCross)
    cerebro.broker.setcash(10000.0)
    cerebro.broker.setcommission(0.001)
    cerebro.broker.set_slippage_fixed(0.05)
    # cerebro.addsizer(LongOnly) # 全部策略使用自定义sizer
    cerebro.addsizer_byidx(idx, maxRiskSizer)
    lg.warning(f'初始市值 {
        cerebro.broker.get_value()}')
    cerebro.run(stdstats=False)
    lg.warning(f'最终市值 {
        cerebro.broker.get_value()}')

    cerebro.plot()
```

版权声明

本文为[roy1126_]所创，转载请带上原文链接，感谢

https://blog.csdn.net/weixin_44785098/article/details/122675199

【七段碼】 05

【蛇形填空】 03

[code à sept paragr

[remplissage des vi
de serpent] 03

WPF utilise async v
effectuer des opéra
asynchrones

友情链接

资讯整合: IT数据的整合 副头条: IT资讯、技术知识

Java人: 专注Java开发 前端人: 专注前端开发

编程人: 编程的世界, 专注各种开发 Python人: 专注Python开发

摄影圈子: 专业摄影知识, 旅游摄影分享

免责声明

本站以网络数据为基准, 引入优质的垂直领域内容。本站内容仅代表作者观点, 与本站立场无关, 本站不对

如有内容侵犯了您的权益, 请告知, 本站将及时删除。联系邮箱: chxpostbox@gmail.com

Copyright © 2020 文章整合 All Rights Reserved.