## 可转债高频交易实战手册

Ricequant获取数据，BackTrader线下回测。

1. 数据获取

选择产品：煜邦转债，118039.XSHG。

使用Ricequant在本地获取行情数据。

调用all\_instruments获取类型为Convertible的可转债产品，共计1050个。

调用get\_price获取煜邦转债2023全年的日行情，8.15上市。

调用get\_price获取煜邦转债在2023.11.23的tick行情，存入文件conv\_bond\_tick.h5\_118039\_20231123。获取煜邦转债在2024.3.15的tick行情，存入文件conv\_bond\_tick.h5\_118039\_20240315。

编写：conv\_bond\_tick.py

校准代码：conv\_bond\_tick2.py

说明：Tick行情数据，是以datetime时间为索引的['open','high','low','last','volume', 'openinterest']序列。注意时间升序。注意open/high/low/last不是日线指标，要转义为最新价。openinterest被借用作为id索引，用于扩展字段。

1. 回测工具

使用BackTrader进行线下回测。

策略一：根据最新价加上价差，简单计算双边报价，成交后重新发布双边报价。

编写：conv\_bond.py

策略二：考虑库存风险、周期因素、波动率、订单密度，根据AS公式计算预定价格、价差，发布双边报价并在成交后刷新，增加价格检查、买卖限额。

编写：conv\_as.py

说明：下单逻辑，在next方法中。成交处理逻辑，在notify\_order方法的order.Completed分支。方法中的buy、sell、cancel指令，在下一回合才生效。

为防止未来函数，next方法中只能看到self.datas[0]，即已发生的历史行情，而不会把更多字段传进来。为方便编程，通过openinterest转义为id，访问全局df，以获得remaintime、midprice、买卖盘等更多参数。

1. 订单生命周期管理

每个tick在next函数中作下单判断，如果下单则在下个tick执行订单。因为是限价单，所以要等到价格合适才能成交。

根据AS公式计算预定价格rev\_price，价差设为0.25元，则基于预定价格加减价差，形成双边报价单，即一个较低价格的买单、一个较高价格的卖单。

任何一边报价成交，即刻取消另一边报价，并重新计算预定价格，重新放置双边报价单。

每次下单数量为10张，设置限额quota为50张，则持仓达到限额即停止该方向下报价单，即只做单边报价。为避免报价与最新价交叠导致疯狂扫单，要在报价前做价格检测。

即便双边报价一直不成交，每隔一分钟也要检测一下当前价格，如果预定价格发生较大偏移，则要撤消之前的双边报价、重新发出新的双边报价。

临近闭市，要作平仓处理。

1. 日志管理

发布双边报价时，记录CLOSE价格、QTY持仓、预定价格、买单CREATE、卖单CREATE。

买单/卖单执行时，记录方向、成交价格、金额、费率。

撤单或拒单，记录方向、价格。

1. 回测执行
2. 策略评估

全天tick数据回测完毕，进行策略执行评估。为保证统一基准，开市设置开仓，闭市统一平仓。假如平仓未达成，根据闭市价估算平仓后收益。

* 多少次发布双边报价，成交多少笔买单/卖单，撤单多少笔；
* 相对收益，基准收益（持有仓位、开盘到收盘），绝对收益。