

3.1 读写数据的应用案例

PYTHON 支持的各种各样的数学计算，而 TBquant 平台具有完备的期货证券数据，有完善的策略评估体系，更是有方便的量化交易策略编写和交易体系。所以 TBPY 将 TBQuant 和 PYThon 的结合，将会给量化交易者带来极大方便。

下面我将介绍一个完整的案例来讲解如何把 TBquant 的数据、策略评估和交易的优势与 PYTHON 的数学处理的优势相结合，来对量化交易策略进行提升。

策略思路简介：

在 python 上调用 TBQuant 的 RB000 的历史数据，使用 python 上用户自定义的滤波处理函数 fttfilter2 对未来价格走势进行预测，最后将预测后的价格走势写回到 TBQuant。TBQuant 根据预测的价格走势展开进一步的交易。

1) 基础数据的概念

BAR 也叫 K 线。简单的说基础数据就是 BAR 的属性数据，比如常规 BAR 都有 open\high\low\close 这些基础数据。TBquant 的基础数据就是用户可以自己给 BAR 增加个性化的属性。你可以把自己计算的指标比如均线保存到对应的 BAR 上面，这样 BAR 就增加了一个均线属性。复杂的你可以在 PYTHON 里面计算更加复杂的指标，然后将这些指标写在对应的 BAR 上面，这样 BAR 就增加了另一个指标属性。

所以基础数据的基本操作就是读写和清空。

2) PYTHON 读取 TB 的历史行情

2.1 首先我们在 TBquant 平台新建一个交易单元，插入品种 RB000 的日线。并且对数据源进行设置添加一个 RB905 的 5mins 图表。



2.2 然后我们启动 PYTHON 的一种编译器 JUPYTER，在 JUPYTER 上我们开始读取 TBquant 的历史数据 close。然后对 close 进行傅里叶提取主要波形并对未来 close 走势进行预测，将预测后的时间序列作为新的一个指标 wave。

PYTHON 代码如下：

#初始化 TBPY 之前 必须先启动 tbquant

```
import tbpy
```

```
tbpy.init()
```

#设定品种周期, 从 TBquant 读取历史行情数据

```
symbols=['rb000.SHFE']
```

```
freq='1d'
```

```
begintime=datetime.datetime.strptime('20181001','%Y%m%d')
```

```
endtime=datetime.datetime.strptime('20190101','%Y%m%d')
```

```
bars=tbpy.get_history(symbols, freq,begintime, endtime, fields=None, timeout='30s')
```

```
code=symbols[0]
```

```
ddrb=bars[code]
```

#调用 PYTHON 函数完成滤波处理和价格预测

```
tn=10
```

```
tx=ddrb['close'][:-tn]
```

```
[a,b,c]=fttfilter2(tx,3,tn,2)
```

3) PYTHON 将数学模型处理的结果写回 TBquant。

我们将 wave 指标写到对应的 BAR 上。PYTHON 代码如下：

#将预测的价格序列写回 TBquant,便于在 TBquant 上做交易

```
for i in range(ddrb['time'].shape[0]):
```

```
    tbpy.write_fundamental(code, "wave",ddrb['time'][i],float(c[i]))
```

4) TBquant 读取基础数据, 进一步完成交易。

4) 在 TBquant 里面读取 wave 指标, 对下一天的交易方向给出判断, 然后在 5mins 上完成日内交易。

4.1 首先我们要编写一个 bartrade 的公式, 代码如下。

Params

```
Numeric hllen(80);
```

Vars

```
Dic<Numeric> fft("wave"); //定义基础数据变量 fft,接收 PYTHON 写入的基础数据
```

wave

```
Numeric highline; //5mins 的高点通道
```

```
Numeric lowline; //5mins 的低点通道
```

```
Series<Numeric> fft2; //5mins 的 fft2 变量, 接收日线图的 fft
```

```
Global Numeric flag; //交易方向
```

Events

```
OnBar(ArrayRef<Integer> indexes)
```

```
{
```

```
    //在日线图上画出 fft 指标
```

```
    range[0:0]
```

```
{
```

```
    if(fft<>InvalidNumeric)
```

```
        PlotNumeric("fft",fft);
```

```
}
```

```
    //在 5mins 图上进行小波段交易
```

```

range[1:1]
{
    //在 5mins 图上画出 fft2 指标
    if(data[0].fft<>InvalidNumeric)
        fft2=data[0].fft;
    PlotNumeric("fft2",fft2);
    if(fft2>fft2[1]) flag=1;
    if(fft2<fft2[1]) flag=-1;
    //高低点通道的计算
    highline=Highest(high[1],hllen);
    lowline=lowest(low[1],hllen);
    PlotNumeric("highline",highline);
    PlotNumeric("lowline",lowline);
    //大小周期共振的波段交易
    if(high>=highline and MarketPosition== -1)
        BuyToCover(0,Max(Open,highline));
    If(Low<=lowline and MarketPosition==1)
        Sell(0,Min(Open,lowline));
    if(MarketPosition<>1 and High>=highline and flag==1)
        Buy(1,Max(Open,highline));
    If(MarketPosition<>-1 and Low<=lowline and flag== -1)
        SellShort(1,Min(Open,lowline));
}
}

```

4.2 我们把 bartrade 加载到最初的交易单元基础数据上。并且打开 K 线。结果如下图所示：



