实习报告2022-02-21

宋欣源

2022 年 2 月 21 日

1 综述

下面对于实习的工作做一个报告。

2 第一，期货策略

2.1 综述

从1月开始，构建基于L1和五档数据的tick级高频期权策略，取得了很大的成果，在各种策略搭建上都有了较大进展。

2.2 第一部分，数据清洗和采集

主要策略数据依赖于期权逐笔成交数据，部分数据在数据库中原先数据中处于缺失状态，采用其他数据库中L2高频数据转化而来。在读入数据时，期权的成交时间往往很不整齐，采用0.5s读入之间频率读入数据，保证每一条tick读入框架，整理成1min级别分钟交易数据。与此同时，按照同样的时间格式读入相应的期货数据，由于展期和掉期策略另外构建，所以只读取期货成交量前十名合约数据，而对于期权来说，非主要合约的成交量很小，所以按照昨日成交量进行排序，读取前五名的期权合约数据。最后得到的结果十对应每一种期货合约品种，有相应的十只期权合约数据，格式统一成日盘和夜盘总计570min分钟数据格式。同时，还建立了期权和期货映射关系，方便计算期权和期货的协同变化因子。

2.3 第二部分，因子计算

总计开发了716个基于不同逻辑的因子，主要逻辑有如下类别：

2.3.1 各类因子类别

1) 基于BS模型和期权希腊值，各类希腊值组合因子，在期权期货上的映射
2) 传统因子思路：期权CP力量对比，期权多空势力对比，期权期货势力对比，分歧，期权统计因子，各类单纯量价数据乖离，1，2，3，4阶矩，相关性，回归组合因子，基于tick的大小单，

3) 期权定价模型的应用：隐含波动率，隐含波动率组合拆分，基于隐含波动率计算的C,P价格，再套用平价公式和现实价格的对比，波动率对比，计价转换模组一二三

4) 期货期权对冲组合因子，远期期权和近期期权关系因子（拆分欧式和美式期权），收益率和远期利率动态过程

5) 各类早晚盘比例，日夜比例，差值，小时差，30min差，均值，日均值小时均线因子

6) 基于tick的持仓成交量分布因子，Vasicek模型（利用期权定价的概率分布反演收益率曲线），CIR模型

7) BDT二叉树因子，连续时间对等模型（拆解期权期货和现金的利率，风险，价值做二叉树，利用随机微分方程求瞬时波动率）

8) 其他乱七八糟基于复杂微分方程的积分求导因子，高斯HJM,马尔可夫HJM,BGM,

9) 叠式期权，跨式期权，互换期权，上下限期权等多期权期货组合产品协同因子（在截面上构建相互关系，共同操作，进行套利）

2.3.2 因子筛选和有效因子识别

因子发掘完毕，利用回测框架进行回测，回测框架主要依赖于我手动开发的因子回测框架，进行了如下几步操作：

1) 残差同方差化，采用统一方差。

2) 因子共线性分析，计算因子线性相关性，大于0.7的剔除，

3) 因子有效性识别，分别采用日月IR读图和NMI指标读图，取并集合。

4) 计算品种间相关性。只在某一两个品种有效的因子剔除。

2.3.3 特征工程

与此同时，还采取了机器学习方法特征工程方法，采用XGboost,lightGBM和catboost模型，经过1000次迭代，在验证集上认真调参，对因子进行另一种方式的筛选，筛选结果和手动筛选结果不尽相同，分别合成不用的因子。最后，手工筛选留下了40个超级因子，而机器学习筛选，经过grid search网格调参和贝叶斯优化，得到累计importance很高，留下了100个左右的因子，对这100个因子画PDP图分析，分析其中的非线性形，其中包含了40个R2良好的线性因子，还有其他60个非线性因子，尽管ICIR很低，但是经过逐个数据仔细对比，呈现出良好的非线性价值。

2.4 第三部分，计算收益率

计算收益率的过程分为三部分，合成信号，组合截面，和收益计算策略

2.4.1 合成信号

合成信号主要有机器学习合成和手工合成两种方法：

2.4.2 手工合成

手工合成信号操作主要如下

1) 根据过去一个月，一日,一小时的因子收益率和波动率，对因子筛选，前十名因子进因子池，其中最有效的选池方案是根据月度IR换因子池，每个月换一次

2) 根据因子池内的因子收益和波动率进行加权求和，周期比换因子池周期短，事件证明，根据日度收益率加权效果最好

3) 对合成的信号再次清清洗，去掉不合理的数值

2.4.3 机器学习

机器学习合成信号的方法，尝试了随机森林，GBDT,简单神经网络训练的方法，调参的时候，将labels设置成不同的return,(包括未来一小时平均return,未来一小时加权平均，未来一小时净值，未来log收益率很多）都进行了训练，其中未来一小时均线效果最好，但是训练还不成熟，信号效果目前和手工合成信号还有差距。然而，手工信号工作量巨大，不仅要读每一各因子的各类图表，还要读每一种模式的各类图表，数据更新以后。再做一遍手工信号实在太累，所以目前的主要精力以机器学习为主，尝试模型组合和模型堆叠，力求达到同样的效果。

2.4.4 组合截面

利用信号合成挑出的超级信号和机器学习模型信号进行组合构建，同样采用手工构建和集成学习方法构建两种。

2.4.5 手工合成

手工截面构建操作主要如下

1) 对信号进行平滑处理，wavelet滤波操作。

2) 根据品种收益率，品种波动率对55品种截面进行分钟加权，小时加权，除了波动率倒数和收益率以外，主要采用截面收益率协方差矩阵和协方差夏普比率，协方差波动率张量乘积加权，由于每分钟计算矩阵乘积计算量巨大，采用numpy分块矩阵爱因斯坦求和的巧妙方法，让计算时间缩短在5s以内。最后经过组合检验，根据小时协方差矩阵和return的乘积分钟加权效果最好

3) 采用不同的换权模式，每小时，每天对数据重新计算

4) 风险控制：根据日风险因子和收益率协方差和残差进行波动率控制和风险控制，对前50%的波动率和风险权重设置为0。

5) 风险控制：根据上下限设定，单个品种权重不超过阈值，控制风险。

6) 利用二次规划办法求解组合权重分配，考虑上述多个条件，求条件最大值。

2.4.6 机器学习

采用两层神经网络训练权重，labels设成根据周期变化的未来return,输入是过去的return和信号，经过训练获得了和手工合成相似的答案，但是手工截面经过波动率调整和风险控制，波动率更低，在收益率计算上拥有更低的手续费，所以最好的方法依然是手工截面组合。效果是组合收益是品种简单收益相加的两倍。机器学习方法可能由于调参问题，还需要进一步检验

2.4.7 收益计算

机器学习和手工方法用一套收益计算模板，有20多种计算方法，包括日，小时，分钟换仓等，包括

1) 普通分钟换仓，均线分钟换仓，普通小时，均线小时等

1) 分钟换仓，前90%,80%信号换仓

2) 分钟换仓，小时换仓，根据截面波动率决定是否触发换仓

3) 对信号分档，1，0，-1或者5档，应用不同周期换仓

4) 利用小波变换操作信号，滤去高阶频率

5) 最后减去基础数据集，用平均值方法设定基础持仓，计算超额收益。

其中，超级信号在小时换仓下表现最好可以战胜万4的手续费，部分品种可以战胜万6的手续费，大部分手工信号只能战胜万1的手续费，将机器学习方法截面，能战胜万1的手续费，还需要仔细操作。

2.5 第四部分，交易结构

1) 对于收益计算，目前采用的是price的做法，在交易结构上，使用对价方法更新策略净值曲线

1) 考虑换仓时刻的量有多少，根据成交量因子（在因子挖掘中计算得到），设定报价队列，根据量的大小逐步提高价格，当价格超过收益阈值的时候停下，这时得到的额外成本是冲击成本。经过优化对价和报价策略，在期权上目前还是没有办法战胜冲击成本，这是这项研究的一个瓶颈之处。

2) 构建流动性策略，主要参考华泰证券研报，用于降低冲击成本。降低风险

3) 展期策略。展期策略主要是日频策略，是另外一个人构建的，我看过内容，并且在我的交易框架上运行，做了展期日内日夜衔接程序。（主要想法是展期产品计算市场变量，在展期构建远期回购，利用我模拟的利率曲线进行远期回购和曲线拔靴，最后，再用定价模型预估手中的期权价格，择时卖出（变量计算-曲线构建-产品定价））

4) 整套策略被领导拿走，期货公司有返佣，所以实盘结果不得而知。。。

3 第二，深度学习

3.1 第一部分，综述

从10月28日开始，构建高频时间序列数据制作机器学习策略因子，到现在，已经取得了很大的成果，尝试了很多模型，对于最终结构，也有了很大进展，下面是报告内容。

3.2 第二部分，模型总结

研究了很多模型，从最简单的CNN,RNN, LSTM, resnet以及各种模型的组合，模型之间各有优劣势，基于目前的研究，利用日频数据，有如下表格（高频数据预测普遍都在0.6以上）：

model	mean_corr	sign_precise	sign_corr	主观评价	计算时间 (gpu)	计算时间(cpu)
mobileNet(1d)	0.13	0.42	0.3	差	0.84s	大约100s
mobileNet(2d)	0.11	0.2	0.44	差	1.99s	大约20min
mobileNetV2(1d)	0.12	不适合		差	<0.1s	15.84s
mobileNetV2(2d)	0.12	不适合		差	<0.1s	20.20s
LSTM	0.28	0.73	0.24	好	大约20s	太慢
CNN1d	0.14			差	3.55s	太慢
CNN3d	0.12	0.27		差	3.26s	太慢
CNN2d	0.14			差	8.0s	太慢
CNN1d*2+bottleneck	0.08			差	9.3s	太慢
CNN2d*2+bottleneck	0.08			差	16.08s	太慢
CNN1d+bottleneck	0.14	0.48	0.37	可以		可以
CNN2d+bottleneck	0.11	0.27	0.26	可以		可以
resCNN+bottleneck-n	0.14			可以		可以
resLSTM+bottleneck-n	0.28	0.66	0.22	可以		可以
RNN	0.24			差		太慢
GRU	0.26			差		太慢
resCNN	0.13			差	可以	
resLSTM	0.18	0.87	0.68	好	大约25s	很好
res-n	0.06	0.42		差	较少	可以
CNN1d-n	0.14			差		太慢
CNN2d-n	0.07			差		太慢
resLSTM-n	0.18			可以		可以

3.3 第三部分，各类研究

3.3.1 最终结构

最终结构参考因诺股票量价因子挖掘结构，有三个人在搭建（包括我），采用多级训练模型，输入是量价数据的分钟，半分钟均线，方差，残差，（高频结构算子挖掘而来），第一级输出是未来分钟，半分钟特征。第二级输入是拟合的第一级输出，第二级输出是未来5分钟，10分钟，复杂结构因子，如MACD，布林带，BM，动量等。本身这些指标已经可以作为很好的因子，经过之前的模型持仓和组合优化尝试，效果已经明显好于手工因子，和经过反复加权的超级因子相比稍差一点，但是这个模型还没有经过finetune，因自此三级模型是用上一级的特征，推理收益率曲线，主要方法还是深度学习。

实际模型一共6级，我们完成了四级，预测周期不断增长，预测指标越来越复杂，类似于预测因子再预测收益率曲线的过程。每一级模型经过resblocks保证每一级都有效，在每一级模型上要进行仔细优化，比如时间，复杂度优化，在此基础上，研究了mobilenet模型，deeplab模型，bottleneck模型，还有复杂门的时间序列循环时间网络（类似LSTM）。神经网络对于高频数据，拥有远超普通线性因子的优越性能，准确率和相关性，互信息率都有很好效果，将来一定能取代普通因子模型。下面对已经研究的东西进行举例说明。（神经网络研究模式大概是这样，一共研究了10种（每周一种））

3.3.2 bottleneck

bottleneck方法主要用来训练CNN神经网络，可以降低数据维度，先用bottleneck层作为人为升高数据的隐藏维度，在高维数据中提取信息，因为信息损失，就会减少高频噪声，就可以提取较好的特征，再用bottleneck层人为降低维度，再次进行模型简化，很大的避免了过拟合，算法如下：

Input: 1 channel 42 * 280000 time series dataloader; 3*3 线性卷积核; 5*5 线性卷积核;

Output: 1 channel 277200 return 或者277200其他高维特征;

1: 升维convolution(input:1 channel, output:100 channel,kernelsize: 1);
2: 维度转换convolution(input:100 channel, output:100 channel,kernelsize: 5);
3: 特征提取convolution(input:100 channel, output:100 channel,kernelsize: 3);
4: batchnorm2d;
5: relu(0.1);
6: 降维convolution(input:100 channel, output:50 channel,kernelsize: 1);
7: 残差层shortcut;
8: 特征提取convolution(input:100 channel, output:50 channel,kernelsize: 3);
9: batchnorm2d;
10: relu(0.5);

return residual bottleneck+ residual shortcut;

Algorithm 1: bottleneck structure

3.3.3 mobilenet

mobileNet模型原来用于汽车自动驾驶研究，其效果是能实时的进行训练，运行速度比传统模型大大加快，但是准确率确没有大幅下降，对于多通道数据冗余的数据源有很大意义。对于一般的深度学习模型，都是采用预训练加推理的步骤，首先预训练好各类模型，然后将模型更换配置（如高低频，股市量价，基本面，期权期货CTA)进行模型推理，一日为单位进行训的训练和强化学习（设定环境代理和数据产生他相互作用）。mobilenet的作用是省去模型推理和强化学习步骤，实时训练。

对于股市量价数据而言，对于CNN1d模型，输入的量价类特征，大部分都很类似，具有重复性和数据冗余星，将每一个特征作为输入的channel,很适合使用mobileNet代替CNN模型，或者用mobile blocks代替原始模型中的每一个CNN blocks。大幅降低运算时间，（模型复杂度降低n^2，期中n是模型通道数）。操作如下：
在传统CNN中

{卷积核的channel} = {输入特征矩阵的channel} , {输出的新特征矩阵的channel} = {卷积核的个数}

在mobilenet中，将CNN分解成deepwise和pointwise两部分，对于deepwise部分

{卷积核的个数} = {输入特征矩阵的channel} = {输出的新特征矩阵的channel}

{卷积核的个channel } = 1

对于pointwise部分

{1*1卷积核的个数} = {输出特征矩阵的channel} , {输出的新特征矩阵的channel} = {1*1卷积核的个数}

Input: 1 channel 42 * 280000 time series dataloader; 3*3 线性卷积核; 5*5 线性卷积核;

Output: 1 channel 277200 return 或者277200其他高维特征;

1: 42 个 1 channel CNN(input:1 channel, output:42， channel,kernelsize: 3);
2: batchnorm2d(1 feature);
3: relu;
4: residual net;
5: relu(0.1);
6: 1 个 42 channel CNN(input:42 channel, output:1， channel,kernelsize: 1);
7: batchnorm2d(42 features);
8: relu(0.5);
9: residual shortcut;

return residual net+ residual shortcut;

Algorithm 2: mobile structure

前面说的mobilenet的基准模型，但是有时候你需要更小的模型，就是mobilenetV2模型。这里引入了两个超参数：width multiplier和resolution multiplier。第一个参数width multiplier主要是按比例减少通道数，第二个参数resolution multiplier主要是按比例降低数据长度。

降低后的计算精度肯定会下降，但是进一步提高了计算效率，对于时间序列这种超大hidden size的模型来说，非常可取。因此加入了这两个参数进行尝试。

现在自动驾驶策略和手机算法推荐普遍采用的是mobilenetV3模型，是mobilenetV2的进一步变化，首先，使用bottleneck代替CNN参数，第二，去掉了拖慢时间的relu，第三，纺锤形 bottleneck和mobilenet本质上存在一层的计算重复，使用一层1*1 CNN就可以实现。

总网络不算dropout和残差网络，一共15层，使用mobileNet进行训练，训练速度比CNN提高了80%左右,训练精度几乎没有下降。我觉得完全可以实现，在训练精度保持不变的情况下，计算时间大幅下降，训练精度大大提高，性能超过bottleneck+CNN的训练效果。

3.4 第四，神经科学方法研究

attention，神经科学的主要方法，目前最主流的是基于transformer体系encoder，decoder模型，围绕transformer时间序列研究展开，已经做了很多尝试。

基于encoder直接进行预测，效果比较差，将中间向量直接连接全连接神经网络，训练相关性大约在8%，如果在encoder的attention层和forward层加入residuel机制进行尝试，在中间向量C中每一层都加入dropout。

在中间向量和前馈神经网络中加入RNN，LSTM，作为decoder，发现效果比原来好。那么合理的想法就是在transformer的框架内，在中间向量，前馈神经网络中加入新的attention系统（即multi-head attention机制），再用残差神经网络连接，最后做layer normalize，这样做有数据结构性困难，效果也不太好，还需要进一步调整神经结构。另外，transformer结构在高频数据上不尽如人意，但是在股票基本面数据上，效果很好，可能和股票基本面数据的金融逻辑并非number逻辑而是金融文本逻辑有关。

3.5 第五，高性能计算

对GPU底层构架有一定了解，深度学习离不开高性能计算。大规模的深度学习依赖超算，集群计算机，能申请到大量的计算资源是关键，第二还要对这方面很了解，比如slurm，跳板机，多卡训练，分布式计算（dusk），multiprocessing，这些我平时都有使用，GPU训练还要注重服务器通信，宽带系统，内存泄露，这些知识是必要的，目前我每个部分都有一定尝试，还需要系统的学习。高性能计算非常重要，一个高性能计算水平高的人是深度学习团队的基石。

4 第三，其他学习

4.1 阅读论文

阅读研报华泰人工智能系列1-54。微软qlib论文

参考文献

[1] K.Jensen, V.M.Acosta, J.M.Higbie, et al. Cancellation of nonlinear Zeeman shifts with light shifts[J]. Physical Review A, 2009, 79(2):023406.

[2] 华泰人工智能系列