机器学习在比特币市场的应用模型

赖星霖

数据选择

本文选择2015年9月25日20:35起至2020年3月14日9:30在Bitmex平台比特币现货交易5分钟级别K线及成交量数据。经检查没有缺失值。成交量在2016年12月以前有较多为0的数据，因此在计算与成交量相关指标而分母分子同时为0时，记为1。

特征工程&因子构造

输入变量共有5个：收盘价$close$、开盘价$open$、最高价$high$、最低价low和成交量$volume$。由于判别器需要每个因子的尺度基本相近，一般数据清洗采用两种方式：。归一化(分位数)转换的公式是$\frac{y-y_{min,k}}{y_{max,k}-y_{min,k}}$，其中$y_{min,k},y_{max,k}$分别表示$y$的前k个数据的最小值和最大值。从5个基础变量还可以延伸出不同的变量，主要为最大差价$maxgap=high-low$,开收盘差价$gap=close-open$，K线实柱部分$linepart=\frac{close-open}{high-low}$，价量弹性$elastic=\frac{volume-volume_{-1}}{volume_{-1}}/\frac{close-close_{-1}}{close_{-1}}$

考虑到信号可能存在滞后效应，在以上当期变量的基础上，考虑加入滞后最高100期的滞后，因此共有16*100=1600个因子。对每个因子分别采用分位数转换，同时k分别取20~200。参数k各自对应1600个因子，总共180个检验结果。

信号生成

区别于传统的直接预测价格，本文首先根据价格背后的行情生成相应的信号：从三种基本状态0观望，1开多，-1开空可以推出共9种可能的决策：0->0继续观望，0->1从观望开多,0->-1从观望开空,1->0平多，1->1，持续持有多仓,1->-1平多转空,-1->0平空转观望,-1->1平空转开多,-1->-1继续持有空仓。

分别定义:

交易信号代码	交易信号含义	多空信号强度
(-1->1)	空仓转多仓	[4]
(1->1)	维持多仓	[3]
(0->1)	观望转多仓	[2]
(1->0)	多仓转观望	[1]
(0->0)	观望	[0]
(-1->0)	空仓转观望	[-1]
(0->-1)	观望转空仓	[-2]
(1->-1)	多仓转空仓	[-3]
(-1->-1)	空仓转空仓	[-4]

9种信号的判定标准由以下算法确定：假设现在处于时期点t，给定一个前向搜索范围，例如$S>0$,首先找出$[t,t+S]$期间的最高价和最低价，计算最高价和最低价的差值$high_S-low_S$，如果高低差价除以最低价大于交易成本c，$\frac{high_S-low_S}{low_S}>c$，那么如果最高价出现比最低价早，那么起始点$t$为0->1 if $\frac{high-start}{start}>c$ else 0->0，$t+1$到最高价点t_{high}期间都为1->1或0->0，在最高价点为1->-1或0->-1，自t_{high+1}到t_{low-1}为-1->-1，在t_{low}为-1->1 if $\frac{end-low}{low}>c$ else -1->0 ，自t_{low+1}到结尾点t_{t+S}为1->1或者0->0，在结尾处为1->0或0->0；如果最低价出现在前，那么上述的起始点应为0->-1 if $\frac{start-low}{start}>c$ else 0->0，t+1到最低点t_{low}期间都为-1->-1或0->0，在最低点为-1->1或0->1，自t_{low+1}到t_{high-1}期间都为1->1，在t_{high}为1->-1 if$\frac{high-low}{high}>c$ else 1->0，自t_{high+1}到结尾点t_{t+S}为-1->0或者0->0。

根据以上规则，取不同的S参数，在回测中测试其开仓正确率(开仓盈利次数/开仓次数)以及总盈利率,结果如下：

参数S	开仓正确率	总盈利率
100	0.6671916010498687	8.135290101310904
200	0.695928080381	1893.24316095
500	0.69708994709	754.869638476

根据结果，基本可以确认所生成的正确买卖信号比较可靠，可以进一步作为训练集的目标变量。

模型选择

回归

将数据集分为训练集和测试集两部分，测试集由数据集中随机抽取20%样本组成。本文使用训练集进行分类训练或者回归，然后再在测试集中检验。检验标准主要为两个：收益率、胜率。收益率为在测试集涉及范围内，根据判别器给出的信号计算的模拟交易收益率；胜率则只计算做出的交易决策盈利的概率。

一般多元线性回归

使用OLS估计多元线性回归方程，因变量为上述给出的不同信号，输出的信号为预测值的四舍五入整数。本文使用测试集对OLS预测结果进行检验，没有采用传统基于全样本的统计检验，原因是传统更注重估计量的统计特性和相关性分析，而量化交易更倾向于预测性能。

验证集准确率	测试集准确率
0.0025389	0.0021639

一般多元线性回归在同等因子数量条件下无论是拟合准确率还是在测试集中的泛化准确率都相比下文的机器学习分类器算法低很多，因此多元线性因子在实际量化投资中作用有限。

Ridge 回归

由于含有滞后项和由相同基础变量计算的相似因子，多重共线性会导致统计检验的检验功效降低，此外，多重共线性同样弱化了模型的泛化能力，一种解决方法是使用岭回归牺牲一定的参数估计无偏性得到稳健的估计参数。岭回归惩罚系数的设置三个水平$\lambda = 0.3; 0.6; 0.9$，λ的系数大小决定正则化项和最小误差平方的权衡之和。

λ	验证集准确率	测试集准确率
0.3	0.0025408	0.0021625
0.6	0.0025396	0.0021608
1.2	0.0025349	0.0021556
2.4	0.0025193	0.0021402

Lasso 回归

为了防止过拟合问题，在OLS目标函数的基础上加入由λ控制的惩罚项，即Lasso回归。

λ	验证集准确率	验证集准确率
0.3	0.0	-0.0004820
0.6	0.0	-0.0004820
1.2	0.0	-0.0004820
2.4	0.0	-0.0004820

以上分析表明，回归方法不适用于目标为预测的应用场景，回归方法更多用于因果关系的推断。

分类

朴素贝叶斯

朴素贝叶斯是基于贝叶斯公式和极大后验概率准则计算的模型，本质上是一种非参数模型超参数α设为默认的1。

验证集准确率	测试集准确率
0.7555382	0.7545504

多项Logistic

多项Logistic是广义线性模型的一种，通过假定多项条件Logistic分布和使用极大似然估计估计系数向量。

验证集准确率	测试集准确率
0.7555181	0.7545257

支持向量机

多项条件Logistic实质上是找到了一个平面在特征空间里将分别属于不用类别的样本点分开，但是该平面并非是最大间隔的，此外，Logiastic模型并不能直接推广到非线性特征空间。引入最大间隔分隔平面作为目标决策函数，以分类正确条件作为约束条件，就得到了支持向量机。如果将特征矩阵与系数向量得法乘积纳入核函数中便能实现一定的非线性化，以下使用高斯核函数进行训练：

模型	验证集准确率	测试集准确率
支持向量机-高斯核函数	0.7555382	0.7545504
支持向量机-线性核函数	0.7555382	0.7545504

决策树

决策树有熵值和Gini值两种判断准则。现行比较成熟的算法是基于Gini值的CART。

判断准则	训练集准确率	测试集准确率
Gini值	0.9955160	0.6190245
熵值	0.9955160	0.6219158

虽然决策树对训练集的训练准确率较高，但是在测试集中表现一般，泛化能力差，这是因为决策树本身是构建了一个复杂的阶梯状边界——容易导致过拟合。

K近邻

k参数	验证集准确率	测试集准确率
10	0.7608552	0.7453840
50	0.7555583	0.7544929
200	0.7555382	0.7545504
1000	0.7555382	0.7545504
2000	0.7555382	0.7545504

K近邻的方法与SVM本质上比较相似，都是使用了非参数方法，所以性能上两者差不多。

多层感知机

隐藏层参数	验证集准确率	测试集准确率
1	0.7555382	0.7545504
2	0.7555382	0.7545504
3	0.7555382	0.7545504
4	0.7555238	0.7545586
5	0.7555439	0.7545011
6	0.7555382	0.7545504
7	0.7555238	0.7545586

多层感知机是神经网络的基础，而神经网络更适用于高维数据，这里因子数量较少，因此优势并不明显。

组合分类器/回归

Adaboost

AdaBoost是英文"Adaptive Boosting"（自适应增强）的缩写，它的自适应在于：前一个基本分类器被错误分类的样本的权值会增大，而正确分类的样本的权值会减小，并再次用来训练下一个基本分类器。同时，在每一轮迭代中，加入一个新的弱分类器，直到达到某个预定的足够小的错误率或达到预先指定的最大迭代次数才确定最终的强分类器。

Adaboost算法可以简述为三个步骤：
（1）首先，是初始化训练数据的权值分布D1。假设有N个训练样本数据，则每一个训练样本最开始时，都被赋予相同的权值：w1=1/N。
（2）然后，训练弱分类器hi。具体训练过程中是：如果某个训练样本点，被弱分类器hi准确地分类，那么在构造下一个训练集中，它对应的权值要减小；相反，如果某个训练样本点被错误分类，那么

它的权值就应该增大。权值更新过的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去。

（3）最后，将各个训练得到的弱分类器组合成一个强分类器。各个弱分类器的训练过程结束后，加大分类误差率小的弱分类器的权重，使其在最终的分类函数中起着较大的决定作用，而降低分类误差率大的弱分类器的权重，使其在最终的分类函数中起着较小的决定作用。

换而言之，误差率低的弱分类器在最终分类器中占的权重较大，否则较小。

基准模型	训练集准确率	测试集准确率
决策树	0.7555411	0.7545257
Logistic	0.7555282	0.7545504

Voting

对于分类问题的预测，我们通常使用的是投票法。假设我们的预测类别是 ，对于任意一个预测样本 ，我们的 个弱学习器的预测结果分别是 。最简单的投票法是相对多数投票法（plurality voting），也就是我们常说的少数服从多数，也就是 n个弱学习器的对样本 的预测结果中，预测数量最多的类别 为最终的分类类别。如果不止一个类别获得最高票，则随机选择一个做最终类别。稍微复杂的投票法是绝对多数投票法（majority voting），也就是我们常说的要票过半数。在相对多数投票法的基础上，不光要求获得最高票，还要求票过半数。否则会拒绝预测。更加复杂的是加权投票法（weighted voting），和加权平均法一样，每个弱学习器的分类票数要乘以一个权重，最终将各个类别的加权票数求和，最大的值对应的类别为最终类别。投票机制是根据已经训练好的多个模型，通过计算每个模型的准确率进行投票，给予准确率高的模型更多权重(软分类)或者选出最高准确率的模型。

模型组合	训练集准确率	测试集准确率
'naive bayes', 'tree', 'svm'	0.7555382	0.7545504
'naive bayes', 'knn', 'logistic'	0.7555382	0.7545504
'svm', 'logistic', 'knn'	0.7555382	0.7545504
'logistic', 'naive bayes', 'tree'	0.7556100	0.7544847
'svm', 'tree', 'logistic'	0.7556100	0.7544765
'tree', 'knn', 'logistic'	0.7557192	0.7544765
'tree', 'svm', 'knn'	0.7556473	0.7545340
'svm', 'naive bayes', 'knn'	0.7555382	0.7545504
'naive bayes', 'logistic', 'svm'	0.7555382	0.7545504
'tree', 'knn', 'naive bayes'	0.7556473	0.7545422

随机森林

随机森林是一种Bagging方法，通过boostrap的方法提高单个决策树的泛化能力，从下表可以发现，模型对测试集的预测能力比单个决策树上升了不少。

决策树数量	训练集准确率	测试集准确率
10	0.7555382	0.7545504
50	0.9951081	0.7497454
200	0.9954413	0.7512485

模型选择总结

综上所述，前10个在测试集准确率为以下几个分类器，考虑到组合模型的迁移性会受到影响，因此优先考虑单个模型：

决策树数量	测试集准确率
多层感知机-7层神经层	0.7545586
多层感知机-4层神经层	0.7545586
SVM 支持向量机-高斯核函数	0.7545504
SVM 支持向量机-线性核函数	0.7545504
朴素贝叶斯	0.7545504
K近邻-参数200以上	0.7545504
多层感知机1-3层	0.7545504
Logistic	0.7545504

其中，多层感知机不能很好地移植到其他语言上，需要一定的参数化；K近邻是一种非参数方法，我们只能通过定义距离方法例如不同p参数的$(\sum_{u=1}^{n}|x_{iu}-x_{ju}|^{p})^{\frac{1}{p}}$的闵可夫斯基距离寻找最近的k个点，然后求它们的目标变量的均值，需要储备一个自带的历史数据库进行查找匹配。能够直接参数化和单独程序化的只有使用支持向量机、Logisitic回归、朴素贝叶斯三种。剩余的组合分类器在本地平台能够很好地使用，但是很难迁移到指标平台。

回测检验

Logistic 分类器

开仓正确率	总盈利率	交易次数	盈利次数
0.3945	0.02347	9	23

SVM分类器

开仓正确率	总盈利率	交易次数	盈利次数
0	0	0	0

回测发现，直接使用三种信号进行分类，模型的盈利能力和准确率都比较低或者无交易信号，同样的现象也出现在**K近邻和朴素贝叶斯**两种方法上：

	开仓正确率	总盈利率	交易次数	盈利次数
K近邻-参数100	0	0	0	0
朴素贝叶斯	0	0	0	0

这主要由两个原因组成，一个是类别不平衡问题,由于通过随机抽取的方式划分了测试集和验证集，随机抽取的样本含有更多0，于是分类器将更多地偏向预测0；

信号类别	开多1	观望0	开空-1
原样本比例	0.2300483	0.2930745	0.4768772
抽取样本比例	0.1702611	0.5597637	0.2699752

另一个原因是由于这里采取了三种信号而非两种，因此这里都是多项分类器而不是二分类，多项分类器都使用one VS rest 或者one VS one 两种方式，本质上和分别训练分类器再投票组合的方式是等同的，而这与一般先判断是否开仓然后再决定买卖方向的交易逻辑是冲突的，解决第一个问题，只需要进行全样本回归，解决第二个问题，则需要对目标信号做进一步处理，分别采取两个二分类器，一个判断是否交易，一个判断方向。

最终模型

训练两个Logistic分类器分别作为买卖信号和买卖方向的判断，回测检验的利润率和成功率如下：

开仓正确率	总盈利率	交易次数	盈利次数	平均调仓频率
0.541958041958042	1210.5086899921359%	572	310	14天

具体参数如下：

变量名字	买卖信号分类器	买卖方向分类器
当期最大差价	0.013840160457448	0.6807768431534419

当期最大差价	0.9138401604574489	-0.6807768421534419
当期开收盘差价	-14.9199363556557	11.946779923521724
当期K线实柱部分	0.007699558879138878	-0.05453491892666999
当期价量弹性	0.08417033137877325	0.07904332496795
滞后一期最大差价	0.9138401604574489	-0.6807768421534416
滞后一期开收盘差价	-14.9199363556557	11.946779923521719
滞后一期K线实柱部分	0.007699558879138878	-0.05453491892666999
滞后一期价量弹性	0.08417033137877325	0.07904332496795
滞后二期最大差价	0.28061558501938183	0.09011936100952744
滞后二期开收盘差价	-20.894848909637307	14.593789514524852
滞后二期K线实柱部分	-0.06382239337758644	-0.0902477430403969
滞后二期价量弹性	0.0823026130368366	0.11537253007319548
滞后三期最大差价	0.848420865771053	0.725439808893585
滞后三期开收盘差价	-18.35294220268937	11.052718711959944
滞后三期K线实柱部分	-0.028558632953437607	-0.08275578576260392
滞后三期价量弹性	0.058768071897383256	0.10106482838526655
滞后四期最大差价	1.1192776837596299	0.9051776569581801
滞后四期开收盘差价	-16.050717781482604	8.148366661547932
滞后四期K线实柱部分	-0.030426899245194282	-0.044628917665259664
滞后四期价量弹性	0.06971204662555436	0.0330868759842468
滞后五期最大差价	1.3730916411909928	1.4264271548025154
滞后五期开收盘差价	-13.973989010542779	5.998841208911537
滞后五期K线实柱部分	-0.014583191710017164	-0.0045460287619859185
滞后五期价量弹性	0.02090761776372556	0.021278288694665793
滞后六期最大差价	2.567065807238473	2.0472041818630125
滞后六期开收盘差价	-11.623691903300623	4.261390723491782
滞后六期K线实柱部分	0.00012059057406468322	0.008122891711337166
滞后六期价量弹性	0.09343990471715774	0.09318679651883113
截距项	51.176514201594244	-32.037327953543596

模型改进展望

以上的分析表明，使用未来函数生成买卖信号再通过分类器回归的做法比直接对价格进行回归更有效，机器学习中Logistic的优良表现说明复杂的非线性分类器的边际效果其实有限，而且会一定程度上影响模型的泛化能力，例如决策树，Logistic很容易地能适用于高维数据而不会对模型训练速度和泛化能力有太大影响，此外，Logitic也能用于多层感知机并进一步组成神经网络模型对更多的非结构化数据进行反应，因此Logitic能作为、性能优良的基准模型和组合模型的原料。组合模型方面，决策树和随机森林具有较好的训练性能，随机森林能够提高单个决策树泛化能力差的弊端，进一步地，随机森林和决策树能方便地转移到其他平台上，具有良好的解释性，但是需要注意的是，信号生成必须分为买卖信号的买卖方向两部分分别进行训练。因子方面，上述分析仅仅使用了K线数据和成交量数据，还可以引入盘口数据/成交情况/大单分析、资金费率等因子，这都将进一步改善模型的性能。