

2018. 10. 8

基于机器学习和 BL 模型的 沪深 300 选股策略

“紫金港的火烈鸟”工程团队

本报告导读：

本报告围绕机器学习选股，阐述了如何将深度神经网络借鉴到选股研究中。通过遗传算法，将基本面指标和技术指标进行删减，降低噪声及过拟合的问题。同时，我们在选股输出值的基础上，结合 Black-Litterman 模型进行资产权重配置，尝试将深度学习和现代化的资产权重配置模型相结合，使用到我们的沪深 300 量化交易中。

摘要：

- 深度学习在语音、视频、自然语言处理等领域的成功使得越来越多的投资者开始尝试将其运用到投资中，然而，更高的噪音、更复杂的影响因素使得很多经典算法并不适用。本篇报告在尝试了多种算法的基础上，选出了深度神经网络算法，运用到选股策略中。
- 总体来说，深度组合策略的构建包括了三个步骤，即通过遗传算法筛选降维原始数据特征，再通过机器学习算法（深度神经网络和其他算法）学习筛选后的数据特征，转化为分类问题对下一期进行预测是否期望持有该股票，最后通过 Black-Litterman 模型对选取出来的股票进行资产配置，分配权重以确定持仓方案。最终，我们构建了 GA+DNN+BL（遗传算法+深度神经网络+Black-Litterman 模型）的策略模型。
- 基于上述模型和思路，我们构建了沪深 300 股票池中的选股策略，该策略从 2013 年中至 2018 年中的六年时间里，在 32.13% 的最大回撤下获得约 115.76% 的年化收益，未来 5 年内的总收益率达到 636.19%，领先于传统方法的市场收益率。根据检验指标，我们运用了不同的模型以及交易时间周期进行比较筛选，最终优化了选股策略，进一步提升了指标的可靠性。



团队成员：

陈栩淦

电话：18868116256

邮箱：chenxg014@gmail.com

胡宇峰

电话：18958086198

邮箱：iondex@163.com

施莹璐

电话：15061478352

邮箱：shiyinlu10@gmail.com

汤明远

电话：18868108397

邮箱：prctangmingyuan@163.com

张天昊

电话：13306519103

邮箱：3160105282@zju.edu.cn

目录

一、机器学习算法概述.....	3
1.1 机器学习现状.....	3
1.2 神经网络与深度学习.....	3
1.3 遗传算法.....	6
二、机器学习模型的测试流程.....	7
2.1 模型构建流程.....	7
2.2 时间.....	7
2.3 数据获取.....	8
2.4 特征预处理和标签计算.....	8
2.5 训练集和测试集成.....	10
2.6 GA 算法因子降维.....	10
2.7 ML 模型训练和测试.....	12
三、资产权重配置模型.....	15
3.1 Black-Litterman 模型概述.....	15
3.2 CAPM 模型.....	16
3.3 观点矩阵.....	17
3.4 结合机器学习和 BL 模型的资产权重分配.....	18
四、总结.....	21
4.1 策略总结：深度神经网络和 Black-Litterman 模型.....	21
附录.....	25
1. 其他机器学习算法的简要介绍和参数设置.....	25
2. 本文所用到的因子列表.....	29

一、机器学习算法概述

1.1 机器学习现状

自计算机问世以来,人们便试图探索机器究竟能在多大程度上取代人类。随着硬件与软件水平的共同发展,计算机已经能够帮助人类实现人脑无法完成的大规模运算,储存人脑无法储存的海量信息,然而这些仍离人类的智慧相距甚远。随着计算机科学的深入研究,人们发现若能使计算机如同人一样进行不断学习,就可能产生与人类水平相当的智能。由此,科学家们提出了人工智能这一概念,指被人类所制造的机器具有的智能,机器学习则是人工智能的一个重要分支。人工智能的研究历史有着一条自然清晰的脉络,从关注“推理”,到关注“知识”,再到如今的以“学习”为核心。

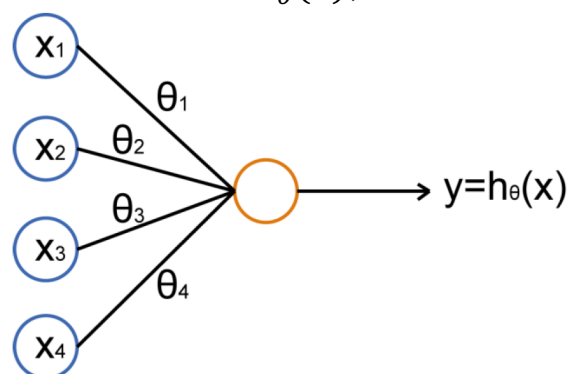
机器学习的对象是某种客观存在的“规律”。这种规律可以非常浅显,也可以极其复杂。机器学习的目标有图像识别、语音识别、自然语言理解、股价预测、天气预测、基因表达、内容推荐等等。上述的许多目标在目前已经得以初步实现,比如指纹识别系统就是一种图像识别,机器学习的是不同指纹图像之间的差异规律;自然语言的识别在智能语音识别系统中得以运用,机器学习的是自然语言的声信号和背后表达意义的规律;又如股价预测,则是学习各种金融指标因子和技术指标因子对于股价波动的影响规律。基于大数据的分析处理,机器学习所能掌握的规律有些甚至连人类自己都无法相比,比如 AlphaGo 通过学习围棋落子和胜负之间的规律,最后得以战胜人类。

1.2 神经网络与深度学习

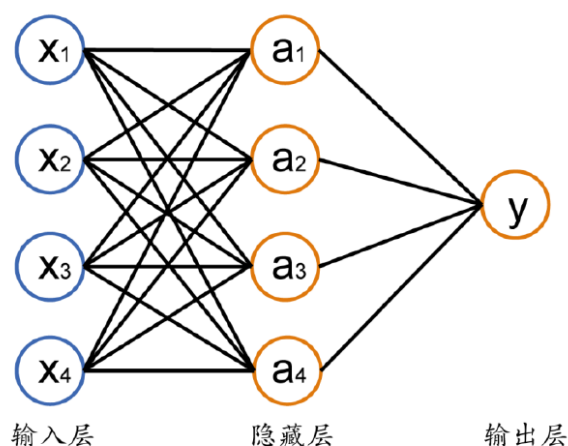
神经网络(Neural Networks, NN)算法中的神经元模拟了现实世界中人体神经元的架构,其理念最早可以追溯到于 1943 年提出的 McCulloch-Pitts 模型和于 1958 年提出的感知机模型。随后,1986 年由 Rumelhart 和 Hinton 提出的反向传播算法对神经网络架构进行了改进。1998 年 LeCun 针对手写字母识别提出的 LeNet-5 模型在应

用中取得巨大突破，该模型也是卷积神经网络（convolutional neural networks, CNN）的雏形。随着 2006 年 Hinton 在深度学习上取得的重要进展，神经网络再次受到人们的瞩目。

神经网络的结构类似于人脑中神经突触的连接，通过突触来进行信息的传递和处理。每个节点代表一种特定的输出函数，称为激励函数。常用的激励函数有逻辑函数，正切函数等。每两个节点间都存在一个权重值，用于表示每个输入对于输出信号的加权值。神经网络的输出值则根据网络间连接方式、权重值和激励函数的不同而不同。如图所示， x_1 到 x_4 四个节点组成输入层，代表输入的特征信息，相当于神经元的树突部分。 θ_1 到 θ_4 为连接权重，代表不同输入信息的重要程度，需要通过大量数据训练调节。输入层 x_1 到 x_4 的信息按权重加和，随后进入一个非线性的激活函数 $h_\theta(x)$ ，模拟神经元激活的过程。



单个神经元是一个线性模型，无法解决包括“异或”问题在内的非线性问题。如果将多个神经元层层连接，可以得到含有隐藏层的神经网络。含有多层的神经网络可以近似地拟合出任何一个函数。



深度神经网络 (Deep Neural Networks, DNN) 就是具备至少一个隐藏层的神经网络。上图中神经网络的数学表达式为

$$y = \sigma\left(\sum_{i=1}^D \theta_i x_i + \theta_0\right)$$

神经网络第 k 个输出的数学表达式为

$$y_k = \sigma\left\{\sum_{j=1}^M \left[\theta_{kj} h\left(\sum_{i=1}^D \theta_{ji} x_i + \theta_{j0}\right) + \theta_{k0}\right]\right\}$$

其中 σ 和 h 分别为输出层和隐含层的激励函数。神经网络的参数为各层的网络系数 θ_{ij} ，可以一并记为向量 θ 。

类似于浅层神经网络，深度神经网络也能够为复杂非线性系统建模，但更多的隐含层能够为模型提供更高的抽象层次，从而提高模型的拟合优度。在模型训练过程中，根据输入信号 x 的值可以计算出隐藏层的值 a ，再根据隐藏层的值算出输出层的预测值 y ，即前向传播算法。人们希望神经网络的预测值 y 和原始标签 t 越接近越好，而输出值和标签的拟合程度取决于每一层中的连接权重 θ 。如图的神经网络中，输入层 x 和隐藏层 a 之间共有 16 个连接权重，隐藏层 a 和输出层 y 之间共有 4 个连接权重，因此通过数据训练（训练集）所得到的自由参数值应该有 20 个（如果在隐藏层增加偏置节点，则模型包含 24 个自由参数）。模型训练的目标是寻取最优的权重参数 θ ，使得预测的均方误差最小，其误差如下式

$$E(\theta) = \sum_{n=1}^N E_n(\theta) = \sum_{n=1}^N \sum_{k=1}^K (y_{nk} - t_{nk})^2$$

一般使用梯度下降法来获取最优参数 θ

$$\theta_{ij} := \theta_{ij} - \alpha \frac{\partial}{\partial \theta_{ij}} E(\theta)$$

其中参数 α 为学习率，表示每一次迭代的步长。当神经网络训练集的数据样本量足够大时，梯度下降法效率很低，应该采用计算效率比较高的随机梯度下降 (Stochastic gradient descent) 方法或者是迷你批量方法 (Mini-Batch) 进行优化。深度学习中，神经网络的训练一般采取迷你批量的方式进行优化。迷你批量优化的方式即每次

根据部分样本（一个迷你批次内的样本，相对于全体样本集而言只是少量样本）进行梯度方向的计算与迭代优化，所有的迷你批次遍历一次时已经进行了很多次的迭代，因此计算效率比传统的批量梯度下降法要高很多。

理论上，隐藏层数越多，隐藏层的节点数就越多，模型对数据的拟合程度越好。

1.3 遗传算法

遗传算法（Genetic Algorithm）是一种由生物学中进化论规律（适者生存，优胜劣汰的遗传机制）演化而来的随机化搜索方法，用于解决最优化搜索的问题，是进化算法的一种。美国教授 J. Holland 首次于 1975 年提出该方法：对于一个最优化问题，一定数量的候选解（称为个体）可抽象成一定数量的染色体，使种群自我调整向更好的解进化。通常求得的解用二进制表示（即 0 和 1 构成的串），但也可以用其他表示方法。进化过程从完全随机个体的种群开始，之后一代一代继续。在每一代中评价整个种群的适应度，从当前种群中随机地选择多个个体，适应度越高，被选择的机会越高，而适应度低的，被选择的机会就低。通过自然选择和突变产生新的生命种群，该种群在算法的下一迭代中成为当前种群。

上述过程中选择，是一个周而复始的过程：评价每个个体，计算其适应度，两两交配，然后突变，产生下一代，直到终止条件满足为止。通常终止条件有以下几种：

- 个体满足最优值条件，即找到最优解；

- 进化次数达到限制条件；

- 适应度已经饱和，无法通过进化产生更优解；

- 资源限制（硬件条件或时间等）或人为干预等。

遗传算法通常包含有以下参数：

- 种群规模 P (population size)：种群中染色体的数目。

- 字符串长度 L (string length)：个体中染色体的长度。

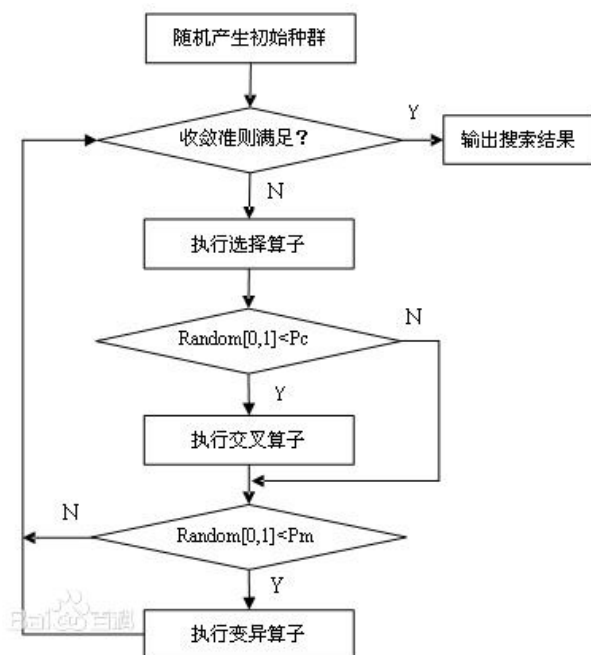
交叉概率 PC (probability of performing crossover):

控制交叉算子的使用频率。

变异概率 PM (probability of mutation):

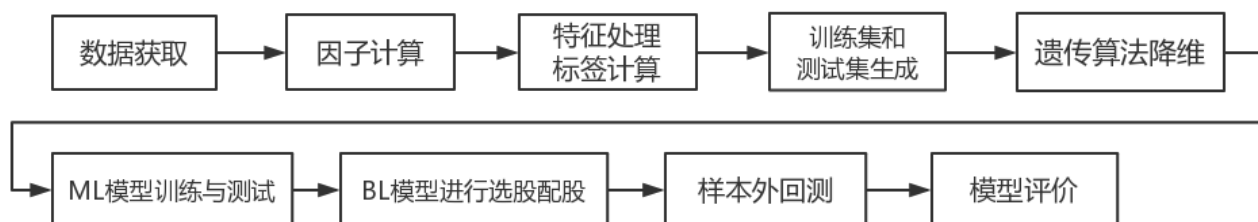
控制变异算子的使用频率。

中止条件 TC (termination criteria): 前文所述的中止条件



二、机器学习模型的测试流程

2.1 模型构建流程



2.2 时间

训练集：2005 年 7 月 1 日-2012 年 12 月 31 日

测试集：2012 年 12 月 31 日-2018 年 6 月 29 日

2.3 数据获取

主要来自 Tushare(<https://tushare.pro>)，使用 jaqs 包(<https://github.com/quantOS-org/JAQS>)进行整合下载，结合国泰安数据服务中心(CSMAR, <http://cn.gtadata.com>)。

其中，训练 ML 模型所用的沪深 300 指数相关信息、股票日行情数据(OHLC)和股票公司金融基本面指标来源于 Tushare；利用 BL 模型进行计算的股票日收益率、沪深 300 日收益率、无风险利率等数据来源于 CAMAR。

2.4 特征预处理和标签计算

(1) 采用的因子：

采用 123 个金融指标因子和 106 个技术指标因子，前者由 Tushare 直接下载，后者自己写代码进行计算，一共 229 个因子。

(2) 训练机器学习模型的数据格式：

对于每天每只股票，我们有

$$X_i(t) \in R^n, n \text{ 为因子数目,}$$

将每天每只股票的数据作为一个样本。

● 特征预处理

- 剔除非沪深 300 成分股，同时剔除停牌或者其他不可交易状态的股票。
- 缺失值处理：用这一天的所有股票的对应因子值的平均值替代。
- 标准化：将所有因子标准化到[0, 1]之间。

表：20151125 这一天 277 只股票的因子表格

	ACD20	ACD6	AD	ADOSC	ADTM	...	yoyocf	yoyocfps	yoyop	yoyroe
000001.SZ	0.001	0.039	0.427	0.392	0.632	...	0.491	0.571	0.979	0.929
000002.SZ	0.073	0.089	0.811	0.348	0.827	...	0.505	0.584	0.974	0.923
000009.SZ	0.000	0.039	0.792	0.000	0.802	...	0.512	0.594	0.980	0.929
000012.SZ	0.001	0.041	0.824	0.244	0.918	...	0.495	0.575	0.979	0.929
000016.SZ	0.000	0.039	0.637	0.235	0.714	...	0.505	0.584	0.974	0.923

000021. SZ	0.001	0.039	0.661	0.218	0.970	...	0.490	0.569	0.979	0.928
...
600895. SH	0.000	0.039	0.693	0.314	0.717	...	0.483	0.561	0.979	0.929
600900. SH	0.000	0.039	0.631	0.449	0.521	...	0.505	0.584	0.974	0.923
600961. SH	0.000	0.039	0.730	0.207	0.957	...	0.505	0.584	0.974	0.923
600997. SH	0.000	0.039	0.661	0.237	0.660	...	0.505	0.584	0.974	0.923
601607. SH	0.000	0.039	0.786	0.220	0.558	...	0.557	0.648	0.980	0.932

计算未来 5 天的 Return-to-Std Ratio 以计算标签 $Y_i(t)$,

$$\text{Return to Std Ratio} = \frac{\text{Return of 5 days}}{\text{Std of 5 days' price}} = \frac{\frac{\text{price}_4}{\text{price}_0}}{\text{Std}(\text{price}_i, i = 0, \dots, 4)},$$

并且进行排序得到 Rank 值，选取 Rank 值前 15 的股票，记

$$Y_i(t) = 1$$

20151125 这一天 Rank 值前 15 的股票因子表格

(数据文件: 20051125head_data.xlsx)

选取 Rank 值后 15 的股票，记

$$Y_i(t) = 0$$

表: 20151125 这一天 Rank 值后 15 的股票因子表格

	ACD20	ACD6	AD	ADOSC	ADTM	...	yoyocf	yoyocfps	yoyop	yoyroe
600997. SH	0.000	0.039	0.661	0.237	0.660	...	0.505	0.584	0.974	0.923
600026. SH	0.002	0.039	0.722	0.157	0.941	...	0.505	0.584	0.974	0.923
000933. SZ	0.000	0.039	0.705	0.288	0.364	...	0.501	0.583	0.982	0.934
000831. SZ	0.000	0.039	0.753	0.263	0.254	...	0.488	0.567	0.929	0.601
...
600808. SH	0.000	0.039	0.782	0.247	0.559	...	0.493	0.572	0.978	0.925
600236. SH	0.000	0.039	0.696	0.188	0.517	...	0.492	0.572	0.983	1.000
600215. SH	0.000	0.039	0.700	0.243	0.901	...	0.487	0.566	0.982	0.934
600028. SH	0.000	0.039	0.505	0.244	0.654	...	0.505	0.584	0.974	0.923

忽略除此以外的股票。

由此，我们得到了

$$(X_i(t), Y_i(t)), X_i(t) \in R^n, Y_i(t) \in \{0,1\}$$

作为我们的训练模型的基本数据格式。

2.5 训练集和测试集成

经过特征预处理和标签计算的过程,我们可以得到训练集时间长度内共 54720 条数据,测试集时间长度内共 39930 条数据。

X_train	(54720, 229)	X_test	(39930, 229)
Y_train	(54720,)	Y_test	(39930,)

2.6 GA 算法因子降维

最开始我们采用了 229 个因子,但是由于因子数量太多,其中必然存在某些因子是存在线性相关或者有噪声的,这些因子是多余的,会增加我们的计算压力。

因此我们采用遗传算法 GA,在 229 个因子中寻找全集最优子集,以减少因子间的线性性,从而减低我们的计算压力。

(1) 生成染色体(New Population):

初始染色体个数为 100, 每个染色体服从

$$N_i \sim \text{Binomial}(229, 0.5), i = 1, \dots, 100$$

并且对染色体添加约束条件:

染色体中 1 的个数必须在 $(100/2, 229 - 100/2) = (50, 179)$ 之间

以使得最后 GA 得到的结果不会有太多或者太少的因子。

(2) 适应性函数(Fitness):

定义为:采用 Logistic Regression 在对应的因子集上训练并测试得到的 AUC 值作为对应染色体的适应度。

(3) 选择(Selection):

根据计算出来的适应度构建一个关于当前染色体集合的分布,其中更大适应度的人更有概率被选中。

(4) 交叉(Crossover):

对于每一个染色体，有 0.2 的可能性会进行交叉作用。

有三种随机化的交叉方式：

- ①相邻染色体一个位点交换，
- ②相邻染色体前面一部分进行交叉交换，
- ③相邻染色体多个位点交换

(5) 变异(Mutation):

对于每一个染色体，有 0.1 的可能性会进行变异作用，并且变异的位点数是随机化的。

最后，对于遗传算法的有效性，我们可以根据遗传算法前后的策略效果进行比较。

表：20151125 这一天 Rank 值前 15 的股票因子表格（经过 GA 降维）

	ADOSC	ADX	ARBR	AVGPRICE	DBCD	...	yoy_or	yoyassets	yoybps	yoyocfps	yoyroe
600997.SH	0.237	0.418	0.142	0.000	0.620	...	0.214	0.538	0.841	0.584	0.923
600026.SH	0.157	0.966	0.508	0.000	0.745	...	0.192	0.538	0.841	0.584	0.923
000933.SZ	0.288	0.393	0.098	0.001	0.799	...	0.192	0.605	0.888	0.583	0.934
000831.SZ	0.263	0.172	0.423	0.000	0.820	...	0.642	0.384	0.779	0.567	0.601
600115.SH	0.248	0.148	0.411	0.000	0.568	...	0.213	0.468	1.000	0.582	0.929
...
600808.SH	0.247	0.124	0.143	0.000	0.596	...	0.192	0.542	0.838	0.572	0.925
600236.SH	0.188	0.294	0.069	0.000	0.525	...	0.194	0.456	0.841	0.572	1.000
600215.SH	0.243	0.054	0.375	0.000	0.712	...	0.100	0.468	0.838	0.566	0.934
600028.SH	0.244	0.149	0.263	0.000	0.461	...	0.192	0.538	0.841	0.584	0.923

表：20151125 这一天 Rank 值后 15 的股票因子表格（经过 GA 降维）

	ADOSC	ADX	ARBR	AVGPRICE	DBCD	...	yoy_or	yoyassets	yoybps	yoyocfps	yoyroe
600309.SH	0.248	0.228	0.283	0.004	0.561	...	0.192	0.538	0.841	0.584	0.923
002024.SZ	0.216	0.082	0.131	0.003	0.811	...	0.192	0.538	0.841	0.584	0.923
600887.SH	0.269	0.237	0.151	0.003	0.475	...	0.192	0.593	0.875	0.577	0.961
600085.SH	0.226	0.715	0.682	0.000	0.485	...	0.192	0.548	0.846	0.574	0.930
000568.SZ	0.202	0.567	0.508	0.002	0.291	...	0.192	0.532	0.857	0.574	0.929

600519.SH	0.235	0.412	0.191	0.003	0.446	...	0.223	0.538	0.841	0.584	0.923
...
600621.SH	0.244	0.376	0.024	0.027	0.872	...	0.234	0.457	0.842	0.571	0.929
600601.SH	0.265	0.231	0.125	0.323	0.616	...	0.192	0.497	0.841	0.571	0.929
600653.SH	0.285	0.317	0.156	0.531	0.639	...	0.094	0.839	0.905	0.571	0.928

2.7 ML 模型训练和测试

我们采用使用深度神经网络（DNN）进行训练和测试。

1. 隐藏层 (hidden layer): 过多的隐藏层个数计算量过大, 且容易造成过拟合, 考虑到以上因素, 采用含有 2 层隐藏层的神经网络。
2. 神经元: 网络输入层神经元节点数就是系统的因子 (自变量) 个数, 输出层神经元节点数就是系统目标分类数。隐层节点选取按经验选取, 最后确定出最合理的网络结构。
3. 网络输入层节点个数是 84 (84 个因子), 最终输出层节点个数为分类数量, 我们采用二分类, 则输出层节点个数为 2, 第 1、2、3 层隐藏层节点个数分别取为 42、21, 最终我们构建了一个 84-42-21-3 的全连接神经网络模型。(如果输入层为 229 个因子的话, 则构建一个 229-114-57-2 的全连接神经网络模型)
4. 激活函数 (activation): 激活函数可以为神经网络加入非线性因素, 以弥补线性模型的不足。我们在隐藏层采用 ReLU 激活函数, 在输出层采用 Softmax 激活函数。
5. dropout: 使用 dropout 可以有效减小过拟合概率。dropout 的原理是每次迭代时都随机地从隐藏层上去除一部分神经元, 每个神经元都随机地与其他神经元进行组合, 减小彼此之间的相互影响。我们在设置了一层 0.5 的 Dropout 层
6. Batch Normalization: 该层在每个 batch 上将前一层的激活值重新规范化, 即使得其输出数据的均值接近 0, 其标准差接近 1。
7. 优化函数 (optimizer): 神经网络通过优化函数, 改善训练方式, 来最小化 (或最大
8. 化) 损失函数。我们采用随机递归下降优化函数的效果, 学习率

为 0.001，动量为 0.9（用于加速 SGD 在相关方向上前进，并抑制震荡），衰变参数为 $1e-6$ （每次参数更新后学习率衰减值），并且采用 Nesterov 动量。

9. 损失函数 (loss function): 由于是二分类问题，我们采用分类交叉熵 (Categorical Cross Entropy) 损失函数。
10. 正则项 (Regularizer): 正则化器允许在优化过程中对层的参数或层的激活情况进行惩罚，以避免过拟合。我们在模型中采用施加在权重上的正则项，并采用 L1-L2 正则函数。
11. 迭代次数: 一般期望模型在最大迭代次数内收敛，但由于神经网络并不能保证在各种参数配置下迭代收敛，为了训练模型，我们设置训练代数 (Epoch) 为 100, Batch Size 为 20。

我们最后的输出结果为 0 到 1 区间的某个数，该数越接近于 1，代表我们越希望持有该只股票；越接近于 0，我们越希望将该只股票卖出。

下面是模型的参数各层参数情况：

采用 GA 之前：

隐含层	隐含层维度	参数个数
全连接层 1	114	26220
Dropout 层	114	0
全连接层 2	57	6555
批标准化层	57	228
全连接层 3	2	116
总参数量		33, 119
可训练参数		33, 005
不可训练参数		114

采用 GA 之后：

隐含层	隐含层维度	参数个数
全连接层 1	42	3570
Dropout 层	42	0

全连接层 2	21	903
批标准化层	21	84
全连接层 3	2	44
总参数量:	4,601	
可训练参数	4,559	
不可训练参数	42	

完成以上模型以后，我们也尝试了其他多种模型算法，来寻找最优结果。以下模型的结果对比会在报告中给出。

DNN	Deep Neural Networks
LR	Logistic Regression
SVM	Support Vector Machine
NB	Naive Bayes
RF	Random Forest
GBM	Gradient Boosting Machine
BAG	Bootstrap AGGREGATING
ET	Extra Trees
ADA	Ada Boost
EXS	Ensemble XGBoost Stack

图：除 DNN 之外的尝试的其他算法

采用 DNN 和 GA 算法之后，得到沪深 300 成分股对应的数据如下：

	000001.SZ	000002.SZ	601766.SH	...	601800.SH	601818.SH	601857.SH	601866.SH
2012-12-31	0.013		0.649	...		0.765	0.997	0.666
2013-01-31	0.012	0.009	0.702	...	0.777	0.710	0.996	0.707
2013-03-07	0.011	0.009	0.695	...	0.822	0.726	0.998	0.742
2013-04-08	0.011	0.009	0.758	...	0.796	0.771	0.995	0.914
2013-05-09	0.012	0.009	0.813	...	0.874	0.808	0.997	0.812
2013-06-06	0.012	0.009	0.755	...	0.870	0.800	0.998	0.814
2013-07-09	0.013	0.010	0.613	...	0.772	0.663	0.993	0.649
2013-08-06	0.013	0.010	0.550	...	0.763	0.671	0.996	0.647
2013-09-03	0.013	0.010	0.521	...	0.646	0.450	0.983	0.523
2013-10-10	0.012	0.011	0.564	...	0.701	0.573	0.993	0.503
...
2018-02-05	0.015	0.011	0.589	...	0.861	0.561	0.974	0.561
2018-03-12	0.016	0.011	0.674	...	0.879	0.552	0.981	0.631
2018-04-11	0.015	0.010	0.510	...	0.798	0.441	0.954	0.517
2018-05-11	0.016	0.011	0.814	...	0.837	0.746	0.988	0.766
2018-06-08	0.017	0.012	0.924	...	0.950	0.872	0.998	0.892

三、资产权重配置模型

3.1 Black-Litterman 模型概述

Fisher Black 和 Robert Litterman 于 1990 年最先提出 Black-Litterman 模型,将 Markowitz 的均值方差最优化理论和 Bayesian 的混合估计方法相结合,基于资本资产定价模型 (CAPM) 以及 Sharp 提出的逆最优化理论建立了 Black-Litterman 模型。传统的数量方法的局限在于当投资组合的个别资产预测报酬率变动时,会造成全局的不稳定,从而导致投资组合对参数的微幅变化具有相当的敏感性。此外,根据 Markowitz 的均值差最优化的理论:

$$\max_{\vec{w}} \{ \vec{w} \vec{m} - \lambda \vec{w}^T \Sigma \vec{w} \}$$

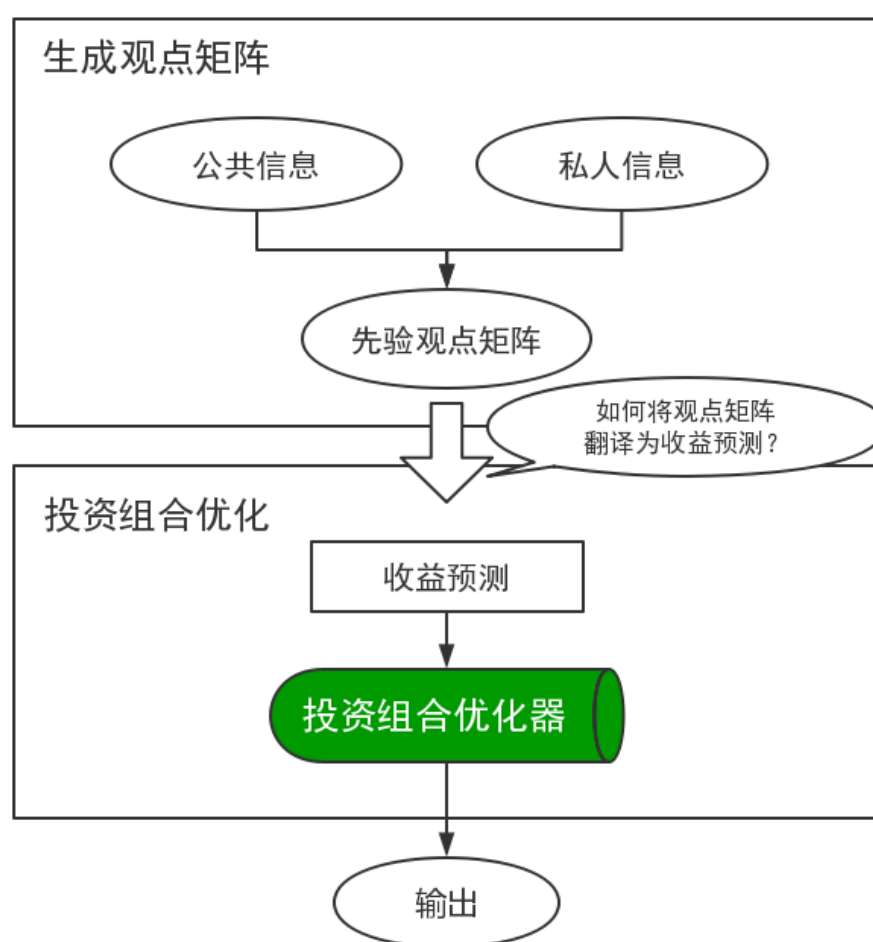
- \vec{m} 为组合的超额收益率矩阵
- Ω 为组合个股收益率的协方差矩阵

对矩阵求导,得到公式

$$\Pi = 2\lambda \Sigma W$$

由此解得资产资产配置结果 w 。

在这个模型中投资者必须利用长期的历史资料，提出预测报酬率和方差数，这些观点皆为 100%的信心水准，Black 和 Litterman 认为，投资者的看法在实际情况下，很难达到完全预测正确，因此传统模型不稳定。BL 模型基于 CAPM 模型，结合投资者主观观点，利用 Bayes 公式得到后验分布，克服了经典的 Markowitz-均值方差模型对于输入数据数量和质量要求太高，出现极端的投资组合权重以及对输入参数高度敏感性等缺点。



图：模型描述

3.2 CAPM 模型

利用 CAPM 模型公式，对于个股，其收益率和风险满足公式

$$E(R_i) - R_f = \beta_i(E(R_m) - R_f)$$

$$\beta_i = \frac{\text{cov}(R_i, R_m)}{\sigma_m^2}$$

$$\pi_i = E(R_i) - R_f = \beta_i(E(R_m) - R_f)$$

$$\pi = [\pi_1, \pi_2, \pi_3, \dots]^T$$

其中：

- R_i 为个股收益率
- R_f 为无风险利率
- R_m 为市场组合收益率，在报告中指沪深 300 指数的收益率
- Π 为超额收益率矩阵

3.3 观点矩阵

Black 和 Litterman 通过综合投资者观点作为一个条件分布，很好的解决了如何刻画投资者观点的问题

假设投资人关于这 N 个资产持有 K 个观点，并且这 K 个观点是互不相关的，这就保证了条件分布的协方差是对角矩阵。

假设投资人对于 N 个资产的 K 个观点用下面的矩阵表示。

- P , $K \times N$ 维矩阵，其每一行代表一个观点所涉及资产的权重，相对观点的权重和为 0，绝对观点的权重和为 1；
- Q , $K \times 1$ 维向量表示每个观点的期望收益
- Ω , $K \times K$ 维对角阵，表示观点的协方差矩阵，代表观点的信心或者不稳定性

这样可以将投资者观点描述如下：

$$P = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad Q = \begin{bmatrix} 2\% \\ 3\% \end{bmatrix} \quad \Omega = \begin{bmatrix} w_{11} & 0 \\ 0 & w_{22} \end{bmatrix}$$

将观点矩阵带入模型，可以得到更新后的公式：

$$\vec{m} = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\vec{\pi} + P^T\Omega^{-1}\vec{q}]^1 \quad (*)$$

$$\vec{V} = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}$$

- M 是更新后的超额收益率矩阵
- τ 表示投资者观点信心
- V 是更新后的投资组合协方差矩阵

最后带入 $M = 2\delta V W$ 求得资产分配 W，作为最优资产权重分配结果，其中 δ 为风险偏好系数，可以根据投资者不同的风险偏好进行设定。

在此基础上，Petter N. KOLM, Sergio M. Focardi 和 Frank J. Fabozzi 在论文 *Incorporating Trading Strategies in the Black-Litterman Framework* 中进一步深化了观点矩阵的研究，将跨期动量交易策略、多因子模型等作为观点矩阵加入 BL 模型，进行了实证研究。

3.4 结合机器学习和 BL 模型的资产权重分配

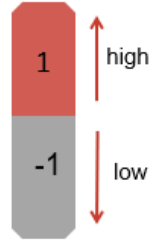
对于机器学习选股步骤中给出的个股结果，每支股票对应 0 至 1 区间中的某个值。越接近 1 表示个股目前行情越好。在此基础上，采用跨期动量交易模型，在沪深 300 成分股中分配资产权重，构建观点矩阵。

将沪深 300 成分股个股按照 0 至 1 区间值从大到小排序，采用动量交易模型作为观点矩阵：

排序处于前一半的股票，在 P 矩阵中相应位置赋值 1；后一半赋值为 -1。得到相应的 P 矩阵：

$$P = [1 - 1 \ 1 \ 1 \dots 1 - 1]_{300 \times 1}$$

¹ *具体证明参考：The Black-Litterman Model Explained WingCHEUNG† February, 2009



将个股每日收益率作为 R 矩阵，得到：

$$R = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{300} \end{bmatrix}$$

根据不同的策略周期日期长度（different frequency），得到 R_1 , R_2 , R_3

观点收益率矩阵 $q = P \times R_i$ ，得到 q_1, q_2, q_3, \dots

最终观点矩阵为：

$$\vec{\hat{q}} = E(q)$$

$$\Omega = var(q)$$

带入 (*) 式中，得到更新后的组合协方差矩阵和超额收益率矩阵，求解效用函数最大值。

$$\max_{\vec{w}} \{ \vec{w} \vec{m} - \lambda \vec{w}^T \Sigma \vec{w} \}$$

为了防止资产在某些股票上配比过大造成严重损失，在求解上式函数最大值时，增加约束条件：

$$-0.036 \leq w_i \leq 0.036, i = 1, \dots, n$$

$$\sum_{i=1}^n w_i = 1$$

将上述优化问题转化成标准的二次规划约束问题，如下：

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} w^T (2\lambda \Sigma) x - m^T w \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

$$\text{where } G = \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}, h = \begin{bmatrix} 0.036 \\ \vdots \\ 0.036 \\ 0.036 \\ \vdots \\ 0.036 \end{bmatrix}$$

$$A = [1 \quad \cdots \quad 1], \quad b = [1]$$

使用内点罚函数法（内点法）解决这个二次规划的约束问题，基本思想是通过构造惩罚函数把约束问题转化为一系列无约束最优化问题，进而用无约束最优化方法求解。

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T Px + q^T x \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

内点法保持在可行域内部搜索，因此内点法仅适用于有下列约束条件的情况下。

定义障碍函数：

$$G(x, r) = f(x) + rH(x)$$

其中 $H(x)$ 是连续函数，当点 x 趋向于可行区域边界时， $H(x) \rightarrow +\infty$ ，因此可以构造函数：

$$H(x) = \sum_{i=1}^m \frac{1}{g_i(x)}$$

$$\text{或 } H(x) = -\sum_{i=1}^m \log g_i(x)$$

取一个极小的正数 r ，则当 x 趋向于可行区域边界时， $G(x) \rightarrow +\infty$ ；否则有 $G(x) \approx f(x)$ ，故只需求解 $\min G(x, r)$ 即可得到所需要的解。

四、总结

4.1 策略总结：深度神经网络和 Black-Litterman 模型

我们的策略测试时间选择为 2012 年 12 月 31 日至 2018 年 6 月 29 日，主要采用深度神经网络的算法得到沪深 300 成分股的个股对应输出，输出值为 0 至 1 区间中的某个数，该数越接近 1 代表越倾向于持有；越接近 0 越倾向于卖出。

Black-Litterman 模型通过 Bayes 理论，将 CAPM 和投资人观点结合得到资产配置权重分配。其中，投资人观点采用深度神经网络的输出值，将个股根据输出值排序，再结合动量交易的理念，构建投资者观点矩阵作为 BL 模型的输入值，得到最后的资产配置。

(1) 机器学习算法训练情况

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>TPR</i>	<i>FPR</i>	<i>AUC</i>
<i>LR_beforeGA</i>	0.974	0.953	0.998	0.975	0.998	0.050	0.999
<i>RF_beforeGA</i>	0.985	1.000	0.970	0.985	0.970	0.000	0.997
<i>SVM_beforeGA</i>	0.955	0.999	0.911	0.953	0.911	0.001	0.999
<i>DNN_beforeGA</i>	0.981	0.999	0.964	0.981	0.964	0.001	1.000
<i>NB_beforeGA</i>	0.706	0.866	0.488	0.625	0.488	0.076	0.783
<i>GBM_beforeGA</i>	0.983	0.983	0.984	0.983	0.984	0.017	0.992
<i>BAG_beforeGA</i>	0.972	0.968	0.976	0.972	0.976	0.032	0.995
<i>ET_beforeGA</i>	0.997	1.000	0.993	0.997	0.993	0.000	1.000
<i>ADA_beforeGA</i>	0.992	1.000	0.984	0.992	0.984	0.000	1.000
<i>EXS_beforeGA</i>	0.984	0.997	0.972	0.984	0.972	0.003	0.998
<i>LR_afterGA</i>	0.980	0.962	0.999	0.980	0.999	0.040	0.999
<i>RF_afterGA</i>	0.985	1.000	0.971	0.985	0.971	0.000	0.998
<i>SVM_afterGA</i>	0.962	1.000	0.924	0.960	0.924	0.000	0.999
<i>DNN_afterGA</i>	0.957	1.000	0.914	0.955	0.914	0.000	1.000
<i>NB_afterGA</i>	0.709	0.876	0.487	0.626	0.487	0.069	0.752
<i>GBM_afterGA</i>	0.991	0.998	0.984	0.991	0.984	0.002	0.996
<i>BAG_afterGA</i>	0.979	0.973	0.984	0.979	0.984	0.027	0.996
<i>ET_afterGA</i>	0.997	1.000	0.994	0.997	0.994	0.000	1.000
<i>ADA_afterGA</i>	0.992	1.000	0.984	0.992	0.984	0.000	0.999
<i>EXS_afterGA</i>	0.991	0.998	0.985	0.991	0.985	0.002	0.998

可以看到，大部分的模型训练结果都很好，超过 95% 的准确率。

(2) 不同交易周期对比

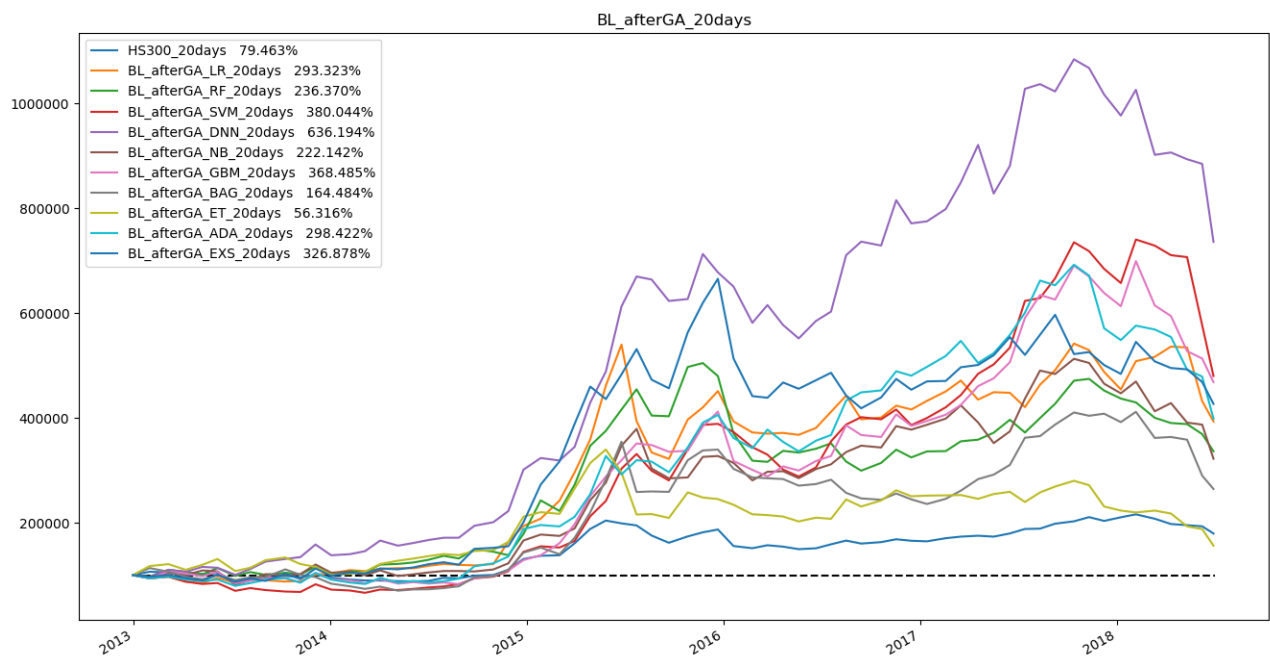
交易 周期	是否使用 GA	总 收益率	年化 收益率	历史 最大回撤	历史 夏普率	历史 年化波动率	信息 比率	初始 净资产	最终 净资产
10days	beforeGA	0.31	0.06	0.57	1.01	0.36	-0.20	100000	130839.71
10days	afterGA	3.66	0.67	0.41	3.29	0.33	2.79	100000	465637.91
12days	beforeGA	0.00	0.00	0.38	0.64	0.36	-0.48	100000	99507.95
12days	afterGA	2.79	0.51	0.33	2.92	0.37	2.12	100000	379259.09
15days	beforeGA	2.30	0.42	0.41	2.99	0.38	2.85	100000	330161.02
15days	afterGA	0.66	0.12	0.60	1.72	0.37	1.27	100000	166222.81
18days	beforeGA	0.38	0.07	0.43	1.42	0.35	0.58	100000	137546.07
18days	afterGA	1.66	0.30	0.40	2.86	0.36	2.71	100000	265586.81
20days	beforeGA	2.30	0.42	0.44	3.33	0.41	2.43	100000	330324.48
20days	afterGA	6.36	1.16	0.32	5.71	0.33	5.07	100000	736194.15
25days	beforeGA	0.90	0.16	0.46	2.62	0.34	1.35	100000	189760.27
25days	afterGA	1.23	0.22	0.57	2.82	0.41	2.26	100000	222856.31
30days	beforeGA	0.39	0.07	0.51	1.95	0.45	0.70	100000	138629.76
30days	afterGA	0.99	0.18	0.54	2.84	0.39	1.91	100000	198554.25

图：不同交易周期对比

根据对比可见，交易周期取 20 个交易日所得到的结果最好，所以在我们的策略中，最终选择 20 个交易日为策略交易周期。每 20 个交易日进行一次选股和资产配比变化，在 2012 至 2018 的测试时间内得到收益率结果及具体指标值。

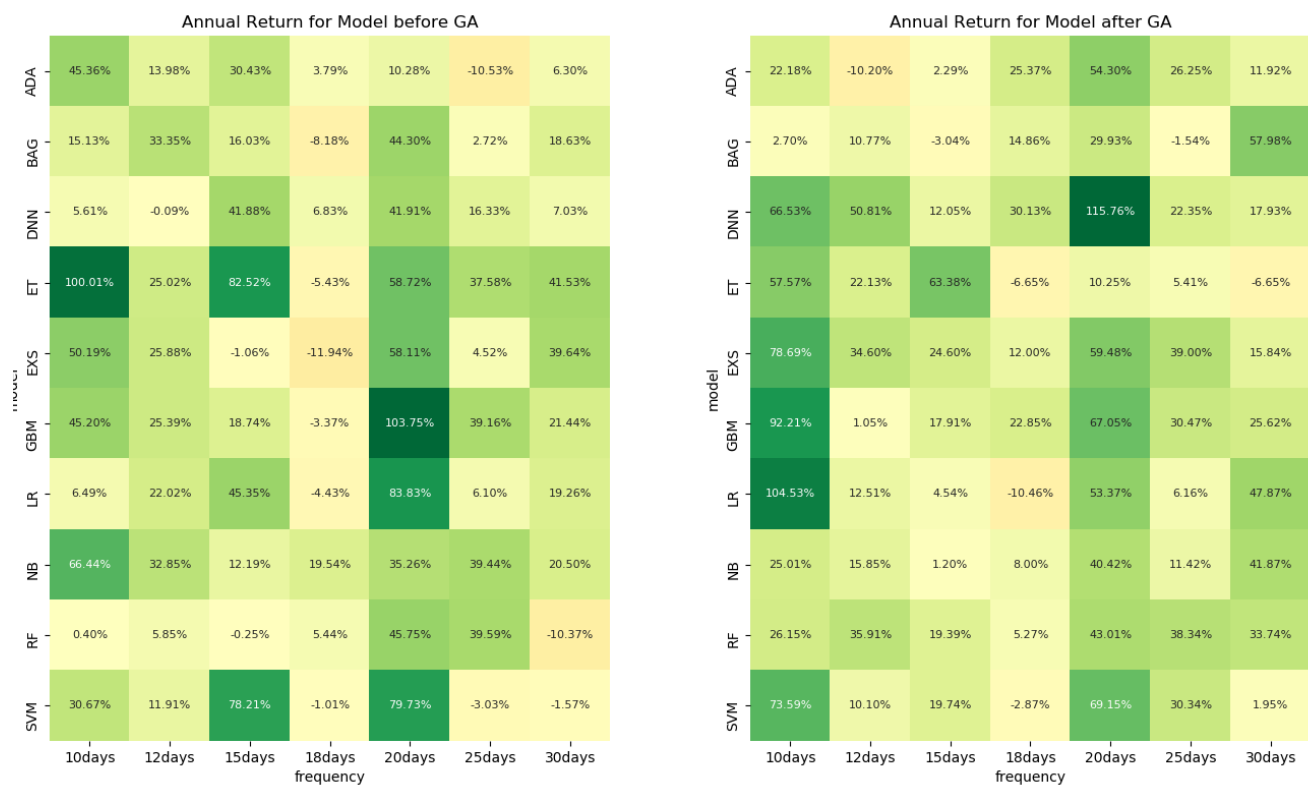
(3) 测试收益情况

图：各机器学习算法结果比较



取交易周期为 20 个交易日，各机器学习算法结果比较图可以看出，深度神经网络（DNN）收益结果最好，因此，这也是本报告最终采取的机器学习算法。

热力对比图也可以很清晰的看出，深度神经网络对应 20 个交易日为交易周期的量化策略年化收益率最高。



图：不同交易周期、不同算法对应的年化收益率热力图

在使用遗传算法之前，我们共有 229 个指标，容易造成过拟合和噪声，在测试结果中，使用遗传算法（afterGA）得到的收益结果比不使用遗传算法（beforeGA）的收益结果更好，说明遗传算法删减指标个数的环节有效降低了数据噪声和过拟合。

（4）主要指标结果

模型	是否使用GA	2013年	2014年	2015年	2016年	2017年	2018年
DNN	beforeGA	0.209150	0.871980	0.619193	0.108736	0.437937	-0.366879
DNN	afterGA	0.633594	1.184956	1.095345	0.185487	0.312206	-0.246667

使用深度神经网络，配合 Black-Litterman 模型，得到该策略每自然年收益率。

测试时间总收益率、年化收益率、最大回撤、年化波动率等重要指标数据在下表给出。

模型	是否使用GA	总收益率	年化收益率	历史最大回撤	历史夏普率	历史年化波动率	信息比率	初始净资产	最终净资产
DNN	beforeGA	2.303245	0.419085	0.437879	3.327376	0.411110	2.426851	100000.0	330324.481640
DNN	afterGA	6.361941	1.157582	0.321251	5.706082	0.332679	5.073871	100000.0	736194.147034

附录

1. 其他机器学习算法的简要介绍和参数设置

1. Logistic Regression

Logistic 回归可以看成单层神经网络模型，因此我们用构建神经网络的方式来构建 Logistic 回归模型。

输入神经元数为系统的因子（自变量）个数，输出为单变量输出（0/1），并采用 Softmax 激活函数，优化函数采用 rmsprop 优化函数，损失函数采用 binary_crossentropy 函数，正则项采用施加在权重上的正则项，并采用 L1-L2 正则函数。

隐含层	隐含层维度	参数个数
全连接层 1	1	230
总参数量：230 可训练参数：230 不可训练参数：0		

2. 支持向量机(Support Vector Machine)

支持向量机（SVM）模型采用 RBF 核函数，惩罚参数 C 为 1.0。

```
1. SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,  
2.   decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',  
3.   max_iter=-1, probability=True, random_state=None, shrinking=True,  
4.   tol=0.001, verbose=0)
```

3. 朴素贝叶斯 Naive Bayes

朴素贝叶斯模型设置平滑参数为 0.6，并且学习分类的先验概率。

```
1. BernoulliNB(alpha=0.6, binarize=0.0, class_prior=None, fit_prior=True)
```

4. 随机森林 Random Forest

随机森林模型设置森林里的树木数量为 100, 树的最大深度为 4, 拆分内部节点所需的最小样本数为 2, 叶节点所需的最小样本数为 1。

```
1. RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
2.                        max_depth=4, max_features='auto', max_leaf_nodes=None,
3.                        min_impurity_decrease=0.0, min_impurity_split=None,
4.                        min_samples_leaf=1, min_samples_split=2,
5.                        min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
6.                        oob_score=False, random_state=None, verbose=0,
7.                        warm_start=False)
```

5. Gradient Boosting Machine

模型设置学习率为 0.1, boosting stages 为 100, 最大深度是 4 层, loss 函数为 deviance, criterion 函数为弗里德曼的改进的均方误差“friedman_mse”, 拆分内部节点所需的最小样本数为 2, 叶子节点所需的最小样本数为 1。

```
1. GradientBoostingClassifier(criterion='friedman_mse', init=None,
2.                            learning_rate=0.1, loss='deviance', max_depth=4,
3.                            max_features=None, max_leaf_nodes=None,
4.                            min_impurity_decrease=0.0, min_impurity_split=None,
5.                            min_samples_leaf=1, min_samples_split=2,
6.                            min_weight_fraction_leaf=0.0, n_estimators=100,
7.                            presort='auto', random_state=None, subsample=1.0, verbose=0,
8.                            warm_start=False)
```

6. Bagging

模型设置估算器为决策树, 估算器的数量为 100, 训练每个基本估算器的样本数为全体样本, 特征数为全体特征。

```
1. BaggingClassifier(base_estimator=None, bootstrap=True,
2.                  bootstrap_features=False, max_features=1.0, max_samples=1.0,
3.                  n_estimators=100, n_jobs=1, oob_score=False, random_state=None,
4.                  verbose=0, warm_start=False)
```

7. Extra Trees

模型设置树木数量为 100, 最大深度是 4 层,, criterion 函数为 gini 函数, 用于评价分裂质量, 拆分内部节点所需的最小样本数为 2, 叶子节点所需的最小样本数为 1。

```
1. ExtraTreesClassifier(bootstrap=False, class_weight=None, criterion='gini',
2.                       max_depth=4, max_features='auto', max_leaf_nodes=None,
3.                       min_impurity_decrease=0.0, min_impurity_split=None,
4.                       min_samples_leaf=1, min_samples_split=2,
5.                       min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1,
6.                       oob_score=False, random_state=None, verbose=0, warm_start=False)
```

8. Ada Boost

模型设置基本估算器为单层决策树, 增强算法为 SAMME.R 算法, 学习率为 1.0, 最大估算量为 100。

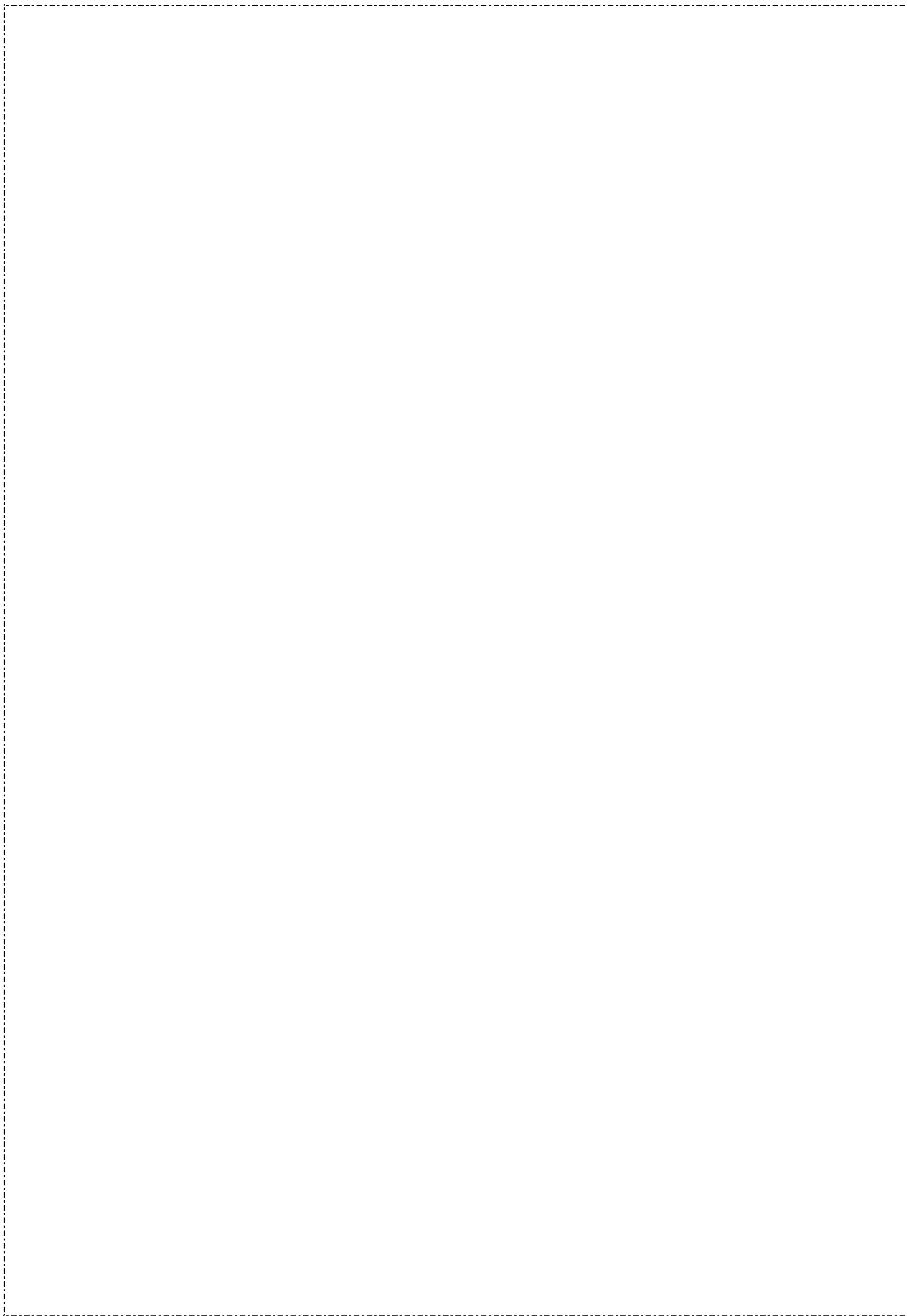
```
1. model = AdaBoostClassifier(n_estimators=100)
2. AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
3.                     learning_rate=1.0, n_estimators=100, random_state=None)
```

9. Ensemble XGBoost Stack

此模型将 sklearn.ensemble 中的 5 个模型 (AdaBoostClassifier、BaggingClassifier、ExtraTreesClassifier、GradientBoostingClassifier、RandomForestClassifier) 和 xgboost 中的模型利用堆栈的方式进行模型合成, 以期能提高模型效果。

Ensemble 中的 5 个模型参数设置与上文提到的一样, xgboost 模型设置树木数量为 500, 最大深度为 7 层, 目标函数为 reg:logistic 函数, 助推器 booster 为 gbtrees, 学习率为 0.5。

```
1. XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
2.               colsample_bytree=1, gamma=0, learning_rate=0.5, max_delta_step=0,
3.               max_depth=7, min_child_weight=1, missing=None, n_estimators=500,
4.               n_jobs=1, nthread=None, objective='reg:logistic', random_state=0,
5.               reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
6.               silent=True, subsample=1)
```



2. 本文所用到的因子列表

管理费用/营业总收入	非流动资产/总资产	基本每股收益同比增长率(%)	绝对价格振荡器
应收账款周转率	净债务	稀释每股收益同比增长率(%)	Aroon down 指标
应收账款周转天数	销售净利率	归属母公司的股东权益相对年初增长率(%)	Aroon up 指标
权益乘数	营运流动资产	归属母公司股东的净利润同比增长率(%)	Aroon 指标
总资产周转率	营业外收支净额/利润总额	归属母公司股东的净利润-扣除非经常损益同比增长率(%)	
每股净资产	非营业利润	经营活动产生的现金流量净额同比增长率(%)	权力平衡
资本支出/折旧和摊销	非营业利润 / 利润总额	每股经营活动产生的现金流量净额同比增长率(%)	
保守速动比率	每股经营活动产生的现金流量净额	营业利润同比增长率(%)	Chande Momentum Oscillator
货币资金 / 流动负债	经营活动产生的现金流量净额/负债合计	净资产收益率(摊薄)同比增长率(%)	定向运动指数
货币资金 / 带息流动负债	经营活动产生的现金流量净额/带息债务	指数移动平均线	移动平均线收敛/发散
流动资产/总资产	经营活动产生的现金流量净额/净债务	指数移动平均线	移动平均线收敛/发散 signal 指标
流动资产周转率	经营活动产生的现金流量净额/经营活动净收益	指数移动平均线	移动平均线收敛/发散 hist 指标
每股现金流量净额	经营活动产生的现金流量净额/营业收入	指数移动平均线	动力
销售成本率	经营活动产生的现金流量净额 / 营业利润	移动平均线	百分比价格振荡器
流动比率	经营活动产生的现金流量净额/流动负债	移动平均线	变化率：（（price / prevPrice）-1）* 100
流动负债/负债合计	经营活动净收益	变化率百分比：（ price-prevPrice ） / prevPrice	
资产负债率	经营活动净收益/利润总额	多空指数	变化率：（price/prevPrice）
产权比率	营业利润 / 负债合计	多空指数比收盘价	变化率 100：（price/prevPrice）
扣除非经常性损益后的净利润	营业利润/营业总收入	三重指数移动平均线	相对强弱指数
扣除非经常损益后的净利润/净利润	营业利润 / 流动负债	布林带 upper 指标	三重平滑 EMA 的 1 天变化率（ROC）
权益乘数(杜邦分析)	每股营业收入	布林带 lower 指标	终极振荡器
总资产净利率(杜邦分析)	扣除财务费用前营业利润	随机指标 slowK 指标	威廉姆斯%R
息税前利润	净利润/营业总收入	随机指标 slowD 指标	Chaikin A / D 线
息税折旧摊销前利润	利润总额 / 营业收入	随机指标 slowJ 指标	Chaikin A / D 振荡器
息税折旧摊销前利润/负债合计	速动比率	随机快速 faskK 指标	平衡音量
每股息税前利润	研发费用	随机快速 fastD 指标	平均真实范围
息税前利润/营业总收入	留存收益	随机快速 fastJ 指标	归一化平均真实范围
已获利息倍数(EBIT/利息费用)	每股留存收益	随机相对强弱指数 faskK 指标	真实范围
基本每股收益	总资产报酬率	随机相对强弱指数 faskD 指标	平均价格
稀释每股收益	年化总资产净利率	随机相对强弱指数 faskJ 指标	中间价格
期末摊薄每股收益	年化总资产报酬率	相对活力指数指标 UpRVI 指标	典型价格
归属于母公司的股东权益/负债合计	净资产收益率	相对活力指数指标 DownRVI 指标	加权收盘价
归属于母公司的股东权益/带息债务	平均净资产收益率(增发条件)	相对活力指数指标	希尔伯特变换 - 主导周期
归属于母公司的股东权益/全部投入资本	净资产收益率(扣除非经常损益)	异同离差乖离率	希尔伯特变换 - 主导循环阶段
无息流动负债	年化净资产收益率	资金流量指数	希尔伯特变换 - 相量分量（同相）
无息非流动负债	投入资本回报率	价格动量指标	希尔伯特变换 - 相量分量（正交）
销售期间费用率	年化投入资本回报率	梅斯线指标	希尔伯特变换 - SineWave（正弦）
非经常性损益	销售费用/营业总收入	艾达透视指标	希尔伯特变换 - SineWave（引线）
固定资产周转率	销售商品提供劳务收到的现金/营业收入	用户生命周期价值指标	希尔伯特变换 - 趋势与循环模式
股权自由现金流量	每股资本公积	佳庆离散指标	真实波幅指标
每股股东自由现金流量	每股盈余公积	简易波动指标	终极指标中间指标
每股企业自由现金流量	有形资产/带息债务	减去方向指示器	终极指标中间指标
财务费用/营业总收入	有形资产	减去定向运动	终极指标中间指标
营业总成本/营业总收入	有形资产/总资产	正方向指示器	终极指标中间指标
毛利	有形资产/负债合计	正方向运动	终极指标
销售毛利率	所得税/利润总额	双指数移动平均线	修正指标

每股营业总收入	固定资产合计周转率	考夫曼自适应移动平均线	价格动量之差
资产减值损失/营业总收入	营业周期	MESA 自适应移动平均线 MAMA 指标	估波指标
带息债务/全部投入资本	每股未分配利润	MESA 自适应移动平均线 FAMA 指标	平滑移动平均
带息债务	加权平均净资产收益率	MidPoint 超期	方向标准离差指数
全部投入资本	营运资金	中点价格超过期间	情绪指标
利息费用	净资产同比增长率	抛物线 SAR	心理线
价值变动净收益/利润总额	营业收入同比增长率(%)	抛物线 SAR - 扩展	威廉变异离散量
存货周转率	营业总收入同比增长率(%)	三重指数移动平均线（T3）	平滑威廉变异离散量
存货周转天数	资产总计相对年初增长率(%)	三角移动平均线	收集派发
非流动负债/负债合计	每股净资产相对年初增长率(%)	平均方向运动指数	收集派发
长期债务与营运资金比率	利润总额同比增长率(%)	平均方向运动指数评级	收集派发
移动平均线	Aroon 振荡器	商品通道指数	（成交量）移动平均线收敛/发散