Fitz Hoo MFE

证券投资

查看详细资料

关注他 发私信

动态 回答 0 提问 0 专栏 0 收藏 0 更多

他的分享 他的文章 按时间排序

AdaBoost Applied to Stocks Selection

Fitz Hoo
MFE

221 人赞了该文章

可能是由于自己在知乎上关注的话题比较少的原因，因此每次打开知乎首页的时候觉得大家好像都在聊机器学习，机器学习的各种算法大家相互之间也都能够谈笑风生。但机器学习领域鱼龙混杂的现象可能又是最明显的，反正每个人都能扯上几句，刷刷存在感，逢人必谈ML，DL，NLP，ANN，SVM，RF，NB等等，不一而足，顿时整个逼格的提升程度觉得用这句歌词会是比较合适

221 33 条评论 分享 举报 收起

关注了 22 关注者 792

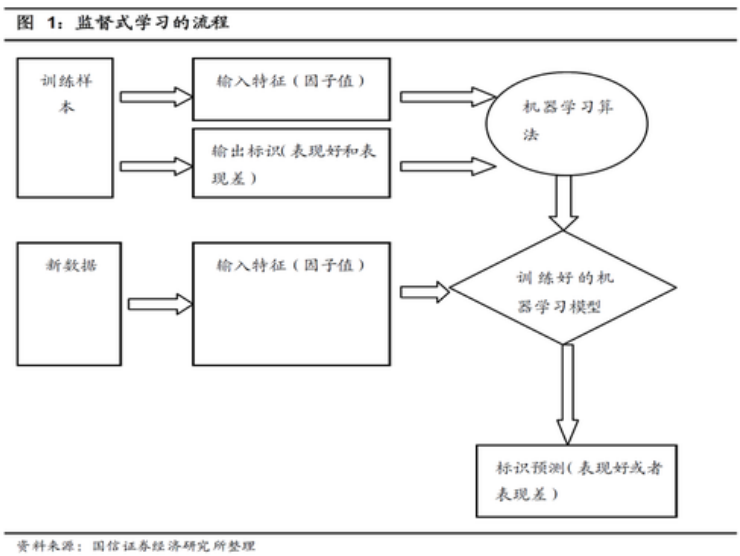
赞助的 Live 1
关注的话题 27
关注的专栏 13
关注的问题 0
关注的收藏夹 0

当然鄙人也没能免俗，作为一条机器学习领域并不等待翻身的咸鱼，也在静静地感受机器学习的热潮，只是偶尔出来透个气、冒个泡。这一段文字显然是会得罪一部分热爱机器学习的初学者的，但这并不是我本意，毕竟我自己也是鱼龙混杂之中的鱼而已，或许说自己是一只小虾米更贴切。

今天借着机器学习的东风，鄙人也想来凑凑热闹，而凑热闹的缘由则是上一篇文章——期权的动态复制反响程度确实让人始料未及，究竟是文章质量太差还是大家对简单的期权相关知识不感冒？

闲话勿赘，言归正传！

本文主题主要是关于机器学习中AdaBoost算法在量化交易中的运用。在开始之前，按套路是要介绍一下AdaBoost算法的基本知识和算法核心流程的。不过在此之前，我想先贴一幅图，按照这个流程图去理解算法的思想会更加有帮助，印象也更加深刻。



图片来源：国信证券经济研究所

机器学习包括监督学习、非监督学习、半监督学习以及强化学习，而其中的监督学习是机器学习中内容最丰富、应用最广泛的部分，也是目前社区中谈论最多的部分，所以此处贴图也是关于监督学习的。根据Wikipedia的资料，监督式学习是一种机器学习方法，其可以从训练资料中学到或建立一个模式（或函数），并以此模式推断新的实例。训练资料是由输入物件（通常是向量）和预期输出所组成的。函数的输出可以是一个连续的值（称为回归分析），或是预测一个分类标签（称为分类）。

AdaBoost的原理

1. AdaBoost概述

AdaBoost，即Adaptive Boosting，译为自适应增强，其1995年由Freund和Schapire提出的。AdaBoost算法是提升方法（Boosting）的一种，那什么是提升方法呢？提升方法是一种将弱学习算法提升为强学习算法的一种统计学习方法，在分类问题中，它通过反复修改训练样本的权值分布，构建一系列基本分类器（弱分类器），并将这些基本分类器线性组合起来，构建一个强分类器，以提高分类性能。而AdaBoost算法便是提升方法中的代表性算法，另外一个是提升树方法（Boosting Tree）。

2. AdaBoost算法流程

a. 初始化训练数据的权值分布。如果有N个样本，则每一个训练样本最开始时都被赋予相同的权重：1/N

b. 训练弱分类器。在训练过程中，如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它的权重就被降低；相反，如果某个样本点没有被准确地分类，那么它的权重就得到提高。然后，权重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

c. 将各个训练得到的弱分类器组合成强分类器。各个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性作用。换言之，误差率低的弱分类器在最终分类器中占

•输入：训练数据集T =
$\{(x_1, y_1), (x_2, y_2), ... ,(x_N, y_N)\}$, $x_i \in \chi \subseteq R^n$, $y_i \in \Upsilon = \{-1, +1\}$
•输出：最终分类器G(x)
1. 初始化训练数据的权值分布，等权分布
$$D_1 = (w_{11}, ..., w_{1i}, ..., w_{1N}), w_{1i} = \frac{1}{N}, i = 1, 2, ..., N$$
2. 使用具有权值分布 D_m 的训练数据集学习，得到基本的弱分类器 $G_m(x) : \chi \to \{-1, +1\}$
3. 计算 $G_m(x)$在训练数据集上的分类误差率:
$$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i)$$
4. 计算第m个弱分类器 $G_m(x)$ 的系数:
$$\alpha_m = \frac{1}{2} log(\frac{1 - e_m}{e_m})$$
5. 更新训练数据集的权值分布
$$D_m = (w_{m+1,1}, ..., w_{m+1,i}, ..., w_{m+1,N})$$
$$w_{m+1,i} = \frac{w_{m,i}}{Z_m} * exp(-\alpha_m y_i G_m(x_i)), i = 1, 2, ..., N$$
$$Z_m = \sum_{i=1}^{N} w_{m,i} exp(-\alpha_m y_i G_m(x_i))$$
6. 构建基本分类器的线性组合，并得到最终的分类器
$$f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$$
$$G(x) = sign(f(x)) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$$

关于算法第五步权重更新这一块，其也是整个算法的核心，有必要补充解释一下：如果根据输入数据，算法预测的结果 $G_m(x_i)$ 与真正的结果 y_i 一致，即 $G_m(x_i) = y_i$，则 $y_i G_m(x_i) = 1$，所以此时权重为 $w_{m+1,i} = \frac{w_{m,i}}{Z_m} exp(-\alpha_m)$，反之，若预测错误则权重为 $w_{m+1,i} = \frac{w_{m,i}}{Z_m} exp(\alpha_m)$，其目的是在下一轮训练中加大分类错误的数据的权重，降低分类正确的数据的权重。
而算法流程第三步关于分类误差率的计算，直观上便能感受到若分类误差率大，即 e_m 偏大，则 $\alpha_m = \frac{1}{2} log(\frac{1 - e_m}{e_m})$的值就会减小，即该弱分类器的权重下降，对整个分类结果的影响将下降。

看到这里我们不难发现AdaBoost算法核心内容其实就是两块：
•如何改变训练数据的权重或概率分布?
AdaBoost算法提高那些被前一轮弱分类器错误分类的样本的权重，而降低那些被正确分类的权重，这样做的好处是在下一轮的的分类过程中错误的分类由于权重加大而受到更大的关注

•如何将弱分类器组合为一个强分类器?
AdaBoost采用加权多数表决的方法。具体说来，即加大分类误差率小的弱分类器的权重，使其在表决中起较大作用，减小分类误差率较大的弱分类器的权重，使其在表决中起较小的作用

4. AdaBoost算法举例

为了更加直观深入地了解AdaBoost算法的思想，我觉得是很有必要举一个实际的例子的。因为纯粹的语言说明不足以使理论清晰，而纯粹的数学解释又让人觉得有点生硬并难于理解。在此推荐台湾大学资讯工程系研究所的这篇关于AdaBoost介绍的例子，其也是很多教科书和博客中所列示的实例的来源。

关于AdaBoost算法的核心内容到此就基本介绍完毕了，而这些内容基本上大家也能够很方便地从网站上搜索到，因此这里只是简单地内容搬运工。虽然机器学习的内容博大精深，但是对于我们接下来进入实战的课题，即利用**AdaBoost算法来预测涨跌**以及**利用AdaBoost算法思想选股**，此处的内容介绍也已经完全足够了。

一、利用AdaBoost算法预测涨跌

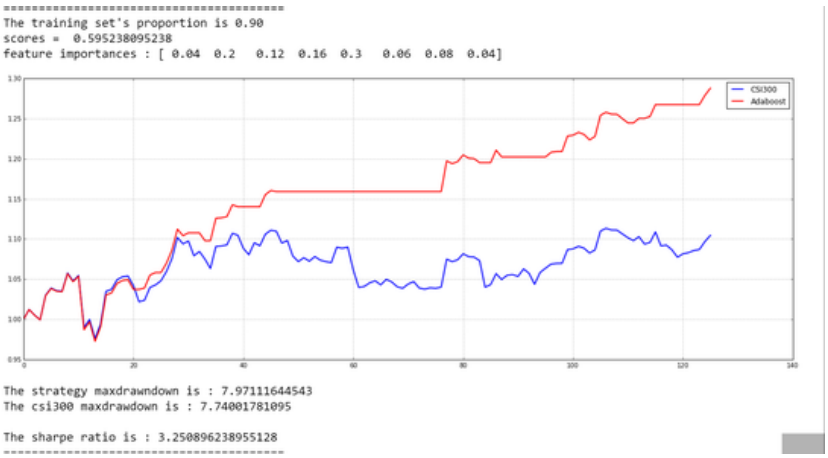
根据前面对AdaBoost算法的介绍，我们在此基础上来厘清一下其具体的实施思路。我们首先通过算法对训练数据进行学习， 训练数据是一个包含输入与输出的数据对，输入数据通常是矩阵的形式，而输出数据则是一列向量。根据训练出来的模型或函数，我们继续用测试数据对模型的优劣进行评估，即使用测试数据中的输入数据来预测输出数据的结果，并与真实的输出数据进行比较，观察算法的预测成功率。

身就是一个二元分类问题，要么涨要么跌。

那什么指标能够很好地预测大盘的涨跌呢，不同的流派会有不同的看法，趋势交易者认同技术分析能够在短时间内预测大盘走势，价值投资者则主张基本面分析。我们况且不论这些派别之争，仅为了说明方便这里我自己仅选取了技术指标作为输入数据，而为了使选取的指标更合理，我根据TA-Lib对技术指标的分类，分别选取了不同类别下的技术指标，而相应的的输出数据是下一个交易日的大盘涨跌情况，根据下一个交易日大盘涨跌情况进行标记，如果大盘上涨则标记为+1，大盘下跌标记为-1。数据集选取好后，我们将前90%作为训练集，后10%作为测试集，来验证我们的效果好坏！

我从Ricequant平台提取了2010-01-01到2016-08-09的沪深300指数价格数据（OCHL），根据收盘价价格时间序列求出了对数收益率，并根据收益率是否大于0分别标记为+1和-1，作为输出数据集。而作为输入数据集的技术指标则是根据各种价格依据TA-Lib求出的，本人在这里选取的技术指标分别为：X = [ema, macd, linreg, momentum, rsi, var, cycle, atr]。

利用AdaBoost算法对前90%的数据集进行训练后，我们**基于预测的输出集**来进行指数的买卖操作策略：如果本交易日预测下一个交易日的label为+1，且本交易日无持仓，则默认在本交易日收盘时买入沪深300指数，若原始持有仓位，则继续保持仓位不动，持仓待涨；反之，若预测的label为-1，则在持有仓位的情况下进行清仓，而若本来就不持有仓位，则继续空仓等待。整个买卖的逻辑是这样的，那我们来看看效果如何。具体代码请戳：来呀！互相伤害呀！



好吧，是不是觉得Unbelievable？此处突然觉得张韶涵的那句歌词实在太贴切了：

> 你说过牵了手就算约定
> 但亲爱的那并不是爱情
> 就像来不及许愿的流星
> 再怎么美丽也只能是曾经
>
> 太美的承诺因为太年轻
> 但亲爱的那并不是爱情
> 就像是精灵住错了森林
> 那爱情错的很透明

好吧，其实我想表达的意思是这回测应该是有问题的，欢迎小伙伴去debug！这里预测数据集和测试数据集是固定的，实际的预测值是基于训练集的训练结果得来的，所以这是一个固定时间段的预测，其实也是可以放宽到每一天都更新训练集然后每天更新预测结果，但是我个人实验了一下，效果并不好，欢迎感兴趣的小伙伴主动尝试。

说完利用AdaBoost算法来预测涨跌这部分内容了，那么我们继续来看看如何利用它的算法原理来选股。这也是我们介绍的重点！

二、AdaBoost算法在选股中的应用

在我进行这个课题的时候，我查阅了一些相关资料，发现AdaBoost算法在实际可查阅到的资料中的应用大部分都只是局限于上面的预测涨跌，而这个真实效果其实是不好的，但是对于AdaBoost算法在其他方面的应用，相关资料少之又少。后来无意中查阅到了国信金工的研报，其关于AdaBoost算法在选股中的应用发了两篇研报，分别是《机器学习法选股》以及《AdaBoost算法下的多因子选股》，而后面一篇只是前面一篇的优化版，思想都一样。

持向量机以及神经网络其实都是可以作为AdaBoost的弱分类器的。因此若我们直接拿AdaBoost算法来选股会便会有点水土不服。

既然现成的弱分类器不好用，那为了对股票进行分类，我们就要想着自己构造一种分类规则，使得不同的股票在该规则下有明显的归属。这里我们借鉴国信金工研报中的处理方法，使用简单的概率统计分类——以T-1期的每个因子暴露分档与T期的收益统计值作为分类方式，AdaBoost依然起到一个分类增强的作用。

这里我着重介绍一下算法在选股中的应用流程，依然按照前面介绍的算法流程来介绍：

1. 初始化权重

提取沪深300指数成分股某一月份不同因子下的数据值，并计算成分股的下一个月的收益率，构成训练集合 $S=\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$，其中 x_i 表示第i个股票对应的所有因子值构成的向量，n表示股票数目（默认为300支，但是排除掉一些缺少因子数据的股票，往往少于300支），y_i 依然是股票的收益标识，+1代表正收益，-1代表负收益。因子值的选取为了方便个人仅仅只是从**Ricequant**平台上通过它的API调取的数据，所以因子的选取可能存在不合理的地方。初始化权重时，将每一支股票的权重设置为一样的。处理后大致是这样一个Dataframe：

	market_cap	pe_ratio	earnings_per_share	return_on_equity	\
600369.XSHG	0.493056	0.555556	0.204861	0.326389	
601669.XSHG	0.621528	0.111111	0.496528	0.760417	
600030.XSHG	0.940972	0.479167	0.690972	0.347222	
601800.XSHG	0.902778	0.097222	0.708333	0.569444	
600705.XSHG	0.576389	0.413194	0.840278	0.854167	

	net_profit	total_equity	monthly returns	weight
600369.XSHG	0.503472	0.600694	1.0	0.008772
601669.XSHG	0.802083	0.809028	1.0	0.008772
600030.XSHG	0.875000	0.902778	1.0	0.008772
601800.XSHG	0.906250	0.927083	1.0	0.008772
600705.XSHG	0.725694	0.645833	1.0	0.008772

注：此处仅选取了前五行，并且所有因子值都以其序数位置填充后转化为[0,1]之间的数值

2. 构建弱分类器

我们依次将因子库中的每一个因子构建一个弱分类器。而该弱分类器可以理解为一个从因子空间到分类信心空间的一个函数，然后依据该函数值来判断弱分类器的优先使用顺序。那该函数如何定义才能更好地区分弱分类器的分类效果好坏呢？直觉上来讲，如果一个分类中表现好的权重越多，即该分类器的效果越好，如果未来某个股票的因子落在这个分类中，则该股票将有更大的概率表现更好。例如，如果有50支股票下一个月的收益率都大于0，即标记为1，而在pe_ratio这个因子下，这50支股票中有40支股票其pe值都是小于均值的，只有10支是pe大于均值的，则说明越小的pe未来越有可能获得正收益。那么这10支股票某种程度上可以理解为是错误分类的，在下一轮训练中我们加大其权重。如果大于与小于均值的股票各25支左右，则说明该分类器效果并不好。

具体可以表示为：首先我们将每一个因子中的值（注意是向量）根据其是否大于该因子均值来将该因子分为两类，标记大于均值的为1，反之为0.此时弱分类器的值可以表示为：

$h(x)=\frac{1}{2}ln(\frac{w_a^j+\varepsilon}{w_b^j+\varepsilon})$，$j=0,1$，则 w_a^1 表示收益率为正时那些股票的因子值大于均值的权重之和，以此类推，$\varepsilon=1/n$ 主要是为了防止分类器的数值等于0，而起不到区分的作用（a表示收益率为正，b表示收益率为负）。

weightedUpdated :

	market_cap	pe_ratio	earnings_per_share	return_on_equity	net_profit
\					
wb_0	0.27	0.20	0.20	0.22	0.28
wb_1	0.23	0.30	0.30	0.28	0.22
wa_0	0.24	0.28	0.31	0.32	0.26
wa_1	0.26	0.22	0.19	0.18	0.24

	total_equity
wb_0	0.29
wb_1	0.21
wa_0	0.23
wa_1	0.27

我们接着根据不同收益率（a或b）不同阶段(1或0)对应的权重之和(w)来判断哪一个弱分类器（因子）效果最好

```
Z:
          market_cap  pe_ratio  earnings_per_share  return_on_equity  \
z_score     0.499099  0.493548            0.487745          0.489829

          net_profit  total_equity
z_score     0.499597      0.496381

Classifier: earnings_per_share
```

3. 更新分类器的权重，并计算分类器的系数

生成第一个弱分类器后，我们重新分配数据权重，对那些错误分类的数据我们增加其权重，对那些正确分类的数据减小其权重，再将权重之和转变为1。接着继续进行第二轮分类器的训练。权重的更新我们依据的公式可以表示为 $w_{m+1}^{x_i} = w_m^{x_i} e^{-y_i h_m^{x_i}}$，m表示第m层弱分类器，h表示弱分类器的值，相当于常规 daboost中的弱分类器的系数。

```
h_1 = -0.2200, h_2 = 0.2100
a = 1, b = -1
权重调整后...
                                  weight
monthly returns earnings_per_share
-1.0            -1.0                0.20
                 1.0                0.30
 1.0            -1.0                0.31
                 1.0                0.19
```

4. 构建基本分类器的线性组合

等到所有因子都作为弱分类器训练后，我们将所有的弱分类器线性组合，此时便得到了一个强分类器，$H(x) = \sum_{m=1}^{M} h_m(x)$ 我们来看一下强分类器的最后结果：

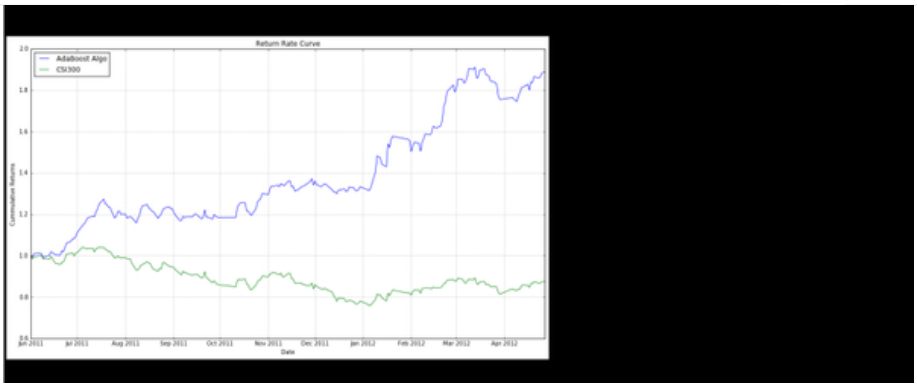
```
            market_cap pe_ratio earnings_per_share return_on_equity  \
601668.XSHG       0.06     0.22              0.21             0.12
601333.XSHG       0.06     0.22              0.21             0.12
601117.XSHG       0.06     0.22              0.21             0.12
600048.XSHG       0.06     0.22              0.21             0.12
600900.XSHG       0.06     0.22              0.21             0.12

            net_profit total_equity   sum
601668.XSHG       0.02            0  0.63
601333.XSHG       0.02            0  0.63
601117.XSHG       0.02            0  0.63
600048.XSHG       0.02            0  0.63
600900.XSHG       0.02            0  0.63
```

```
            market_cap pe_ratio earnings_per_share return_on_equity  \
002310.XSHE      -0.04    -0.21             -0.22            -0.15
002069.XSHE      -0.04    -0.21             -0.22            -0.15
002299.XSHE      -0.04    -0.21             -0.22            -0.15
002007.XSHE      -0.04    -0.21             -0.22            -0.15
600267.XSHG      -0.04    -0.21             -0.22            -0.15

            net_profit total_equity   sum
002310.XSHE      -0.04        -0.02 -0.68
002069.XSHE      -0.04        -0.02 -0.68
002299.XSHE      -0.04        -0.02 -0.68
002007.XSHE      -0.04        -0.02 -0.68
600267.XSHG      -0.04        -0.02 -0.68
```

我们依据DataFrame最后一列sum的值进行操作，做多前10%的股票，做空后10%的股票，构建一个alpha策略组合。但A股市场不支持做空，所以我们这里只是做多前10%的股票，最后的收益大致如下：



由于是自己写的回测，所以可能存在一些实际交易中无法买卖而本回测可以的的情况，导致真实收益会有一些折扣，但是总体趋势应该不会发生变化！

好吧，废话了辣么多，总算扯完了，其实这个过程中还有很多地方需要改进，后期我还是会不断修改这篇文章的，如果小伙伴自身有什么疑问，欢迎在评论区留言。详情请戳：有种点我，有本事赞我呀

最后的最后，你的留言与点赞是我持续分享的不竭动力！**<这么厚颜要赞真的好嘛o(╯□╰)o>**

编辑于 2016-08-25

论如何优雅地复制期权之OBPI策略

Fitz Hoo
MFE

57 人赞了该文章

在机器学习、数据挖掘大行其道的今天，我再在这里给大家分享六、七年前股指期货刚推出时国内比较火爆的策略似乎是比较LOWBEE的一件事，一开始我也是拒绝的，但是拒绝并不代表拒绝成功，正如我抗议，抗议无效一… 阅读全文

57 11 条评论 分享 举报

ARMA+GARCH交易策略在沪深300指数上的应用

Fitz Hoo
MFE

155 人赞了该文章

在金融时间序列分析中，经常会涉及到对资产收益率以及波动率建模，并依据构建模型的预测值来为未来的决策提供依据。而在我接触这门课程以来，也一直对将收益率模型和波动率模型应用到实际

221 33 条评论 分享 举报 收起

策略不给力？来一发卡尔曼滤波

Fitz Hoo
MFE

778 人赞了该文章

我和卡尔曼滤波的渊源要从我开始尝试做配对交易开始说起。 配对交易是八十年代中期华尔街著名投行Morgan Stanley的数量交易员Nunzio Tartaglia成立的一个数量分析团队提出的一种市场中…
阅读全文

778 55 条评论 分享 举报

Kalman Filter Applied to Pair Trading

Fitz Hoo
MFE

58 人赞了该文章

配对交易是八十年代中期华尔街著名投行Morgan Stanley的数量交易员Nunzio Tartaglia成立的一个数量分析团队提出的一种市场中性投资策略。具体说来，其是指从市场上找出历史股价走势相近的股票进行配对，当配对的股票价格差（Spreads）偏离历史均值时，则做空股价较高的股票同时买进股价较低的股票，等待他… 阅读全文

58 7 条评论 分享 举报

221 33 条评论 分享 举报 收起