文章编号: 1000-5641(2019)04-0072-11

带交易成本的二阶在线投资组合选择策略

瞿菁晶[1], 郁顺昌[1], 黄定江[1,2]

(1. 华东理工大学 理学院, 上海 200237;
2. 华东师范大学 数据科学与工程学院, 上海 200062)

摘要: 针对基于在线牛顿步 (Online Newton Step, ONS) 算法的投资组合选择策略没有考虑交易成本的问题, 而交易成本是真实市场中不可或缺的部分, 提出了一种新的带交易成本的在线投资组合选择策略, 简称在线牛顿步交易成本策略 (Online Newton Step Transaction Cost, ONSC): 首先, 结合投资组合向量的二阶信息和交易成本惩罚项构造优化函数, 并推导得出投资组合的更新公式; 然后, 通过理论分析得到ONSC算法的次线性后悔边界 $O(\log(T))$. 实证研究表明, 与半常数再调整投资组合策略 (Semiconstant Rebalanced Portfolios, SCRP) 以及其他考虑交易成本的策略相比, 在SP500、NYSE(O)、NYSE(N) 和 TSE 这 4 个真实市场的数据集上, ONSC 获得了最高的累计净收益和最小的周转率, 表明了所提算法的有效性.

关键词: 投资组合选择; 在线牛顿步; 交易成本

中图分类号: TP399 **文献标志码**: A **DOI:** 10.3969/j.issn.1000-5641.2019.04.008

Second-order online portfolio selection strategy with transaction costs

QU Jing-jing[1], YU Shun-chang[1], HUANG Ding-jiang[1,2]

(1. *School of Science, East China University of Science and Technology*, *Shanghai* 200237, *China*;
2. *School of Data Science and Engineering, East China Normal University*, *Shanghai* 200062, *China*)

Abstract: Existing portfolio selection strategies based on the online Newton step (ONS) algorithm ignore the role of transaction costs, an indispensable factor in real markets. This paper proposes a new online portfolio selection strategy, the online Newton step transaction cost (ONSC) method, to address this issue. First, we constructed the optimal function by combining second order information of a portfolio with the transaction cost penalty term, and the portfolio was subsequently updated. Then, the sublinear regret bound $O(\log(T))$ was achieved by theoretical analysis. Empirical research on the data sets of four

收稿日期: 2018-08-07

基金项目: 国家自然科学基金 (11501204, U1711262); 上海市自然科学基金 (15ZR1408300)

第一作者: 瞿菁晶, 女, 硕士研究生, 研究领域为机器学习与金融投资组合选择.
E-mail: Jolinqu@hotmail.com.

通信作者: 黄定江, 男, 教授, 研究领域为机器学习与人工智能及其在计算金融、教育等跨领域的大数据解析和应用. E-mail: djhuang@ecust.edu.cn.

real markets–namely, SP500, NYSE(O), NYSE(N) and TSE–showed that in comparison to semiconstant rebalanced portfolios (SCRP) and other strategies with transaction costs, ONSC achieves the highest accumulated wealth and the smallest turnover. Hence, the research demonstrates the effectiveness of the algorithm.

Keywords: portfolio selection; online Newton step; transaction costs

0 引 言

在线投资组合选择[1-5]是人工智能和机器学习领域热门的研究课题, 其关键问题在于如何在不确定的市场环境下连续地选择最优投资组合, 以达到一定的目标, 如金融证券市场中的累计收益最大化或损失最小化.

在线牛顿步 (ONS) 算法[6]基于离线优化问题中的牛顿法[7], 近年来被广泛用于在线投资组合选择策略研究[6]. 不同于其他的投资组合选择算法[5], ONS 利用了投资组合向量的二阶信息进行更新; 相比只使用一阶信息, 它可以更快地收敛, 有最优的次线性后悔边界 $O(\log(T))$; 它的不足之处是忽略了金融市场中非常重要的元素——交易成本[8-9]. 最近的一些在线投资组合研究[10-12]试图解决交易成本的问题, 出现了如 SCRP[13]、半泛投资组合策略[14](Semi-Universal Portfolio, SUP) 等带交易成本的策略, 但其中没有使用投资组合向量的二阶信息策略.

针对上述问题, 本文提出了一种新的在线牛顿步交易成本策略 (ONSC): 在 ONS 的基础上结合交易成本, 利用投资组合向量的二阶信息进行更新, 并且在优化函数的设置上, 添加交易成本惩罚项, 控制交易量的大小. 本文还给出了 ONSC 算法的理论后悔边界, 并且通过在真实市场中对累计净收益和周转率的大量实验, 说明 ONSC 算法在保证收益最大化的同时降低了交易成本, 在净收益和稳定性两方面都优于 SCRP 和其他考虑交易成本的算法.

1 相关工作

研究投资组合选择主要有两个流派, 分别是均值方差理论[15]和资本增长理论[16], 其中资本增长理论侧重于多个周期或连续的投资组合选择, 适用于在线场景, 是本文研究的基础.

在线投资组合选择领域有很多出色的策略, 其中最经典的有常数再调整策略 (Constant Rebalanced Portfolio, CRP)[17], 它在每期都会重新调整投资组合, 保证每期分配到各个资产上的财富是一个固定的比例. Cover 在 1991 年提出了泛化投资组合策略 (Universal Portfolio, UP)[18], 采用所有 CRP 专家的加权平均进行投资. 不久之后, 交易成本开始被纳入考虑范畴. Gaivoronski 和 Stella 在 2000 年提出了半常数再调整策略 (SCRP), 相比 CRP 策略, “半”(semi) 的思想, 即只在整个交易期中选取 k 个适当的时期重新调整投资组合, 显著减少了交易成本. Das 等人在 2013 年提出了在线懒惰更新策略[19](Online Lazy Update, OLU), 它通过对优化问题中参数的懒惰更新来调整投资组合. 黄定江等人在 2015 年提出了基于交易成本的半泛投资组合策略 (SUP)[14], 也是用了“半”的概念来控制交易成本, 实证研究表明 SUP 算法在累计净收益和稳定性方面优于之前提到的所有算法.

2 问题设置

考虑一个有 n 只股票 T 个交易期的市场 (这里 1 d 为 1 个交易期). 相对价格向量 $\boldsymbol{x}_t = (x_{t,1}, \cdots, x_{t,n}) \in \mathbb{R}^n$ 表示股票价格的变化, 其中 $x_{t,j}$ 是第 j 只股票在第 t 天的收盘价与第 $(t-1)$ 天的收盘价的比. 记 $\boldsymbol{x}_1^T = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$ 为 T 个交易期的相对价格序列.

第 t 期的投资组合向量用 $\boldsymbol{b}_t = (b_{t,1}, b_{t,2}, \cdots, b_{t,n})$ 表示, $\boldsymbol{b}_t \in \Delta_n$, $b_{t,i}$ 为投资到第 i 只股票上的资产比例. 假设投资组合 $\boldsymbol{b}_t$ 不允许出现差价和卖空, 即 $\Delta_n = \{\boldsymbol{b}_t | \boldsymbol{b}_t \in \mathbb{R}^{n+}, \sum_{i=1}^{n} b_{t,i} = 1\}$. 记 $\boldsymbol{b}_1^T = (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_T)$ 为 T 个交易期的策略. 定义了相对价格向量和投资组合向量之后, 第 t 个交易期的收益因子可以用 $s_t = \boldsymbol{b}_t \cdot \boldsymbol{x}_t \triangleq \boldsymbol{b}_t^{\mathrm{T}} \boldsymbol{x}_t$ 表示.

投资组合的更新会产生交易成本. 设交易成本率为 c, 且 $0 \leqslant c < 1$, 这表示 1 个单位的资产在除去交易成本后剩下 $(1-c)$. 在第 $(t+1)$ 期开始时期, 交易成本向量可以用 $C_t = (C_{t,1}, C_{t,2}, \cdots, C_{t,n})^{\mathrm{T}} \in \mathbb{R}^{n+}$ 表示, 其中 $C_{t,i} = s_t |\widetilde{b}_{t,1} - b_{t,1,i}| c$ 为第 i 个资产上产生的交易成本, $\widetilde{b}_{t,i}$ 表示调整投资组合之前, 第 i 个资产在第 t 期结束时的资产分配比例, 即 $\widetilde{b}_{t,i} = \frac{b_{t,i} x_{t,i}}{\sum_{i=1}^{n} b_{t,i} x_{t,i}}$. 所以, 投资组合经理在第 $(t+1)$ 期的开始要支付的交易成本为 $\mathrm{Cost}_t = C_t^{\mathrm{T}} e$, 其中 $e = [1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^{n+}$.

第 t 期的净收益用 $s_t^C = s_t - \mathrm{Cost}_t$ 表示, 因此在 T 个交易期之后, 投资组合策略 $\boldsymbol{b}_1^T$ 产生的累计净收益为

$$S_T(\boldsymbol{b}_1^T, \boldsymbol{x}_1^T) = S_0 \prod_{t=1}^{T} s_t^C, \tag{1}$$

其中, S_0 为初始资产, 通常设为 1.

另外, 出于比较的方便, 经常考虑收益的对数增长率. 在 T 个交易期之后的对数增长率为 $\sum_{t=1}^{T} \log(\boldsymbol{b}_t \cdot \boldsymbol{x}_t)$, 而一个采用 CRP 选取投资组合 $\boldsymbol{b}$ 的投资者获得的对数增长率为 $\sum_{t=1}^{T} \log(\boldsymbol{b} \cdot \boldsymbol{x}_t)$. $\boldsymbol{b}^*$ 是最大化这个量的事后最优 CRP 的投资组合, 一个采用策略 $\boldsymbol{b}_1^T$ 的在线算法 Alg 的后悔边界定义为

$$\mathrm{Regret(Alg)} \triangleq \sum_{t=1}^{T} \log(\boldsymbol{b}^* \cdot \boldsymbol{x}_t) - \sum_{t=1}^{T} \log(\boldsymbol{b}_t \cdot \boldsymbol{x}_t), \tag{2}$$

即整个投资期中 Alg 与最优常数再调整投资组合策略[17] (Best Constant Rebalanced Portfolio, BCRP) 的对数收益之差. 后悔边界可以衡量一个算法与最优算法之间的差异, 希望得到尽可能小的后悔边界, 即对数收益增长率尽可能接近离线 BCRP 算法.

投资组合选择是在线进行的, 每一期都会利用新得到的相对价格向量, 根据优化函数重新调整投资组合. 它的目标是设计一个策略 $\boldsymbol{b}_1^T$ 使得交易成本最小, 累计收益最大. 后悔边界也是衡量一个算法性能的重要指标.

3 动 机

许多在线投资组合策略在额外考虑交易成本后性能显著提升. 比如 SCRP, 在 CRP 的基础上考虑了交易成本, 加入"半"(semi) 的思想, 只在适当的时刻调整投资组合. 也就是说, 如果调整投资组合产生的交易成本高于收益, 就不进行调整. SUP 对 UP 的改进也是采取了同样的"半"的思想, 从模型上比较, UP 每期调整投资组合, 它的更新机制为

$$\boldsymbol{b}_t = \frac{\int_{\Delta_n} \boldsymbol{b} S_{t-1}(\boldsymbol{b}) \mathrm{d}\mu(\boldsymbol{b})}{\int_{\Delta_n} S_{t-1}(\boldsymbol{b}) \mathrm{d}\mu(\boldsymbol{b})}. \tag{3}$$

而 SUP 在整个 T 个交易期中选取 k 个适当的时期 $(t_1, \cdots, t_k)$ 将投资组合调整为 UP, 这 k 个时期把相对价格向量序列分成了 $(k+1)$ 个部分, 即

$$\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{t_1-1}\}\{\boldsymbol{x}_{t_1}, \cdots, \boldsymbol{x}_{t_2-1}\} \cdots \{\boldsymbol{x}_{t_k}, \cdots, \boldsymbol{x}_T\}.$$

假设只需在每个部分的开始支付交易成本, 于是每个部分 SUP 将获得累计收益

$$s_i = \boldsymbol{b}_i^{\mathrm{T}} \otimes_{t=t_{i-1}}^{t_i-1} \boldsymbol{x}_t, \tag{4}$$

其中, $\otimes$ 表示元素对应乘积, $\boldsymbol{b}_i = \frac{\int_{\Delta_n} \boldsymbol{b} S_{t_i-1}^c(\boldsymbol{b})\mathrm{d}\mu(\boldsymbol{b})}{\int_{\Delta_n} S_{t_i-1}^c(\boldsymbol{b})\mathrm{d}\mu(\boldsymbol{b})}$, $S_{t_i-1}^c(\boldsymbol{b}) = \prod_{j=1}^{t_i-1} \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}_j - \sum_{j=1}^{t_i-1} \mathrm{Cost}_j$. 扣除每部分的交易成本后, 将 $(k+1)$ 个部分的累计净收益求和就得到了 SUP 的累计净收益

$$\prod_{i=1}^{k+1}(s_i - \mathrm{Cost}_i).$$

这类带交易成本的策略 SCRP、SUP 在许多数据集上表现出高效的性能, 但它们有一些潜在的问题: SCRP 基于 CRP 的思想, 每次调整的投资组合为固定的比例, 不能很好地适应动态的市场; SUP 在选定时期采用 UP 的投资组合, 要计算所有 CRP 专家的加权平均, 在计算上非常复杂. 针对上述问题, 本文提出的新策略 ONSC 根据实时数据, 动态调整模型参数, 且每次迭代时计算量小、效率高.

“半” (semi) 的思想本身也有不足之处, 它需要在整个交易期中找到一个最优子集进行投资组合调整, 但找到这个最优子集在计算上也是非常困难的. 另外, 交易成本分为固定比例交易成本、固定值的交易成本, “半”的做法更适用于后者. 而 OLU 作为一种考虑交易成本的策略, 不同于 SCRP、SUP 这两个交易成本策略采用“半”的思想, OLU 给出了一个懒惰的投资组合向量

$$\boldsymbol{b}_{t+1} = \mathrm{argmin}_b - \eta \log(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}_t) + \alpha\|\boldsymbol{b}-\boldsymbol{b}_t\|_1 - \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{b}_t\|_2^2. \tag{5}$$

OLU 在优化模型中加入了 L_1 惩罚项[20], 计算调整投资组合需要的交易量, 并用参数 α 控制交易量的大小. 这个方法实际操作起来更加简便, 于是本文考虑在 ONS 的模型中也引入一个交易成本惩罚项. 之所以选择 ONS 算法来改进, 是因为它采用的策略中损失函数为对数收益的二阶泰勒展开, 利用了投资组合向量的二阶信息; 而 OLU 用的是对数收益, 即一阶信息. 多项关于二阶信息的研究工作[21-23]表明, 二阶信息可以达到更快的收敛速度, 且能提供投资组合向量的波动信息, 相较一阶信息更有助于投资组合选择任务.

本文在 ONS 的基础上考虑交易成本, 设计了一个新的策略 ONSC. 该策略利用投资组合向量的二阶信息进行更新, 并且通过交易成本惩罚项控制交易量的大小. 后面的理论证明与实证研究将说明 ONSC 的良好性能.

4　在线牛顿步交易成本策略 (ONSC)

4.1　模型设计

ONSC 考虑如下优化问题

$$\boldsymbol{b}_t = \underset{\boldsymbol{b}\in\Delta_n}{\mathrm{argmax}} \sum_{\tau=1}^{t-1}\left[f_\tau(\boldsymbol{b}) - \frac{\beta}{2}\|\boldsymbol{b}\|^2 - \frac{1}{2}\alpha\|\boldsymbol{b}-\boldsymbol{b}_\tau\|_2^2\right], \tag{6}$$

其中, argmax $f(\boldsymbol{b})$ 表示在 Δ_n 中使得 $f(\boldsymbol{b})$ 最大的 $\boldsymbol{b}$ 值; 等号右边括号内第一项为损失函数, 表示最大化对数收益, 第二项为正则项处理过拟合问题, 第三项用一个可变参数 α 来控制交易量和交易成本的大小, 保证相邻两项投资组合之间偏差最小, 即每期的交易量尽可能

小, 以此达到降低交易成本的目的, 不难看出 α 越大, 交易成本越大, 对交易的限制也越大, 当 $\alpha=0$ 时, 限制交易量的 L_2 范数没有作用.

在损失函数的设置上, 考虑最大化对数收益. 仿照 ONS 的做法, 利用投资组合向量的二阶信息, 将损失函数 $f_t:\Delta_n\rightarrow\mathbb{R}$ 定义为

$$f_t(\boldsymbol{b})\triangleq\log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t)+\nabla_t^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_t)-\frac{\beta}{2}[\nabla_t^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_t)]^2, \tag{7}$$

其中, $\nabla_t=\nabla[\log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t)]=\frac{1}{\boldsymbol{b}_t\cdot\boldsymbol{x}_t}\cdot\boldsymbol{x}_t$, $\nabla^2[\log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t)]=-\frac{1}{(\boldsymbol{b}_t\cdot\boldsymbol{x}_t)^2}\cdot\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}=-\nabla_t\nabla_t^{\mathrm{T}}$, 且可见 $f_t(\boldsymbol{b}_t)=\log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t)$.

在每一期, 算法会利用历史信息和实时数据, 根据优化问题调整投资组合, 并支付交易成本. T 个交易期结束后, 获得累计净收益 $S_T(\boldsymbol{b}_1^T,\boldsymbol{x}_1^T)=S_0\prod_{t=1}^T s_t^C$, 用于对策略进行评价.

4.2 ONSC 算法

接下来求解第 4.1 节中的优化问题.

当 $t=1$ 时, 令所有 $f_t(\boldsymbol{b})$ 和 $-\frac{1}{2}\alpha\|\boldsymbol{b}-\boldsymbol{b}_t\|_2^2$ 为 0, 单位投资组合 $\boldsymbol{b}_1=\frac{1}{n}\mathbf{1}$ 最大化了 $-\frac{\beta}{2}\|\boldsymbol{b}\|_2^2$. 当 $t>1$ 时, 展开 $\boldsymbol{b}_t$ 中 $f_\tau(\boldsymbol{b})$ 的表达式, 为

$$\boldsymbol{b}_t=\mathop{\mathrm{argmax}}_{\boldsymbol{b}\in\Delta_n}\sum_{\tau=1}^{t-1}\Big[\log(\boldsymbol{b}_\tau\cdot\boldsymbol{x}_\tau)\nabla_\tau^{\mathrm{T}}+(\boldsymbol{b}-\boldsymbol{b}_\tau)-\frac{\beta}{2}[\nabla_\tau^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_\tau)]^2-\frac{\beta}{2}\|\boldsymbol{b}\|^2-\frac{1}{2}\alpha\|\boldsymbol{b}-\boldsymbol{b}_\tau\|_2^2\Big]. \tag{8}$$

不改变优化问题的解, 对式 (8) 等号右边乘以 $\frac{2}{\beta}$ 并舍弃常数, 有

$$\begin{aligned}\boldsymbol{b}_t&=\mathop{\mathrm{argmax}}_{\boldsymbol{b}\in\Delta_n}\Big[\sum_{\tau=1}^{t-1}\nabla_\tau^{\mathrm{T}}\boldsymbol{b}-\frac{\beta}{2}[\nabla_\tau^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_\tau)]^2-\frac{\beta}{2}\|\boldsymbol{b}\|^2-\frac{1}{2}\alpha\|\boldsymbol{b}-\boldsymbol{b}_\tau\|_2^2\Big]\\&=\mathop{\mathrm{argmax}}_{\boldsymbol{b}\in\Delta_n}\sum_{\tau=1}^{t-1}\Big[\frac{2}{\beta}\nabla_\tau^{\mathrm{T}}\boldsymbol{b}-\boldsymbol{b}^{\mathrm{T}}\nabla_\tau\nabla_\tau^{\mathrm{T}}\boldsymbol{b}+2\boldsymbol{b}_\tau^{\mathrm{T}}\nabla_\tau\nabla_\tau^{\mathrm{T}}\boldsymbol{b}-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b}-\frac{\alpha}{\beta}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b}+2\frac{\alpha}{\beta}\boldsymbol{b}_\tau^{\mathrm{T}}\boldsymbol{b}\Big].\end{aligned} \tag{9}$$

于是原优化问题就减弱为

$$\mathop{\mathrm{argmax}}_{\boldsymbol{b}\in\Delta_n}\sum_{\tau=1}^{t-1}\Big[-\boldsymbol{b}^{\mathrm{T}}\nabla_\tau\nabla_\tau^{\mathrm{T}}\boldsymbol{b}-\Big(1+\frac{\alpha}{\beta}\Big)\boldsymbol{b}^{\mathrm{T}}\boldsymbol{b}+2\Big(\boldsymbol{b}^{\mathrm{T}}\nabla_\tau\nabla_\tau^{\mathrm{T}}+\frac{1}{\beta}\nabla_\tau^{\mathrm{T}}+\frac{\alpha}{\beta}\boldsymbol{b}_\tau^{\mathrm{T}}\Big)\boldsymbol{b}\Big]. \tag{10}$$

令 $\sum_{\tau=1}^{t-1}-\nabla^2(\log\boldsymbol{x}_\tau\cdot\boldsymbol{b}_\tau)+\big(1+\frac{\alpha}{\beta}\big)\boldsymbol{I}_n=A_{t-1}$, $\big(1+\frac{1}{\beta}\big)\sum_{\tau=1}^{t-1}\big[\nabla(\log\boldsymbol{b}_\tau\boldsymbol{x}_\tau)+\alpha\cdot\frac{1}{\beta}\boldsymbol{b}_{\tau-1}^{\mathrm{T}}\big]=\boldsymbol{p}_{t-1}$, 则式 (10) 可简化为

$$\mathop{\mathrm{argmax}}_{\boldsymbol{b}\in\Delta_n}(-\boldsymbol{b}^{\mathrm{T}}A_{t-1}\boldsymbol{b}+2\boldsymbol{p}_{t-1}^{\mathrm{T}}\boldsymbol{b}),$$

该优化问题等价于

$$\mathop{\mathrm{argmin}}_{\boldsymbol{b}\in\Delta_n}(\boldsymbol{b}-A_{t-1}^{-1}\boldsymbol{p}_{t-1})^{\mathrm{T}}A_{t-1}(\boldsymbol{b}-A_{t-1}^{-1}\boldsymbol{p}_{t-1}).$$

令

$$\mathop{F}_{K}^{A_{t-1}}(y)\triangleq\mathop{\mathrm{argmin}}_{x\in K}(x-y)^{\mathrm{T}}A_{t-1}(x-y),$$

于是得到所求优化问题的解为

$$\boldsymbol{b}_t = \underset{\Delta_n}{\overset{A_{t-1}}{F}}(\delta A_{t-1}^{-1}\boldsymbol{p}_{t-1}). \tag{11}$$

这就是ONSC的更新机制. 下面给出ONSC的算法, 见算法1, 其中α、β、η用于理论分析, δ是为了后续的实验需要而设置的一个调试参数.

算法1　ONSC 算法

输入: $\boldsymbol{x}_n$, α, β, η, δ

输出: b

1: 初始化 $\boldsymbol{b}_1 = [1/n, \cdots, 1/n]$

2: 对于 $t > 1$, 执行 $\boldsymbol{b}_t = \underset{\Delta_n}{\overset{A_{t-1}}{F}}(\delta A_{t-1}^{-1}\boldsymbol{p}_{t-1})$,

其中

$$\boldsymbol{p}_{t-1} = \left(1 + \frac{1}{\beta}\right)\sum_{\tau=1}^{t-1}\left[\nabla(\log \boldsymbol{b}_\tau \boldsymbol{x}_\tau) + \alpha \cdot \tfrac{1}{\beta}\boldsymbol{b}_{\tau-1}^{\mathrm{T}}\right]$$

$$\underset{K}{\overset{A_{t-1}}{F}}(y) = \underset{x\in K}{\operatorname{argmin}}(x-y)^{\mathrm{T}}A_{t-1}(x-y)$$

$$A_{t-1} = \sum_{\tau=1}^{t-1} -\nabla^2\left(\log \boldsymbol{x}_\tau \boldsymbol{b}_\tau\right) + \left(1 + \frac{1}{\beta}\right)\boldsymbol{I}_n$$

3: $\boldsymbol{w}_t = (1-\eta)\boldsymbol{b}_t + \frac{\eta}{n}\boldsymbol{1}$

ONSC算法第一期使用均匀投资, 即$\boldsymbol{b}_1 = [1/n, \cdots, 1/n]$, 之后每一期把$\boldsymbol{b}_t$投影到一个单纯型上, 最后用$\boldsymbol{w}_t = (1-\eta)\boldsymbol{b}_t + \frac{\eta}{n}\boldsymbol{1}$对投资组合进行平滑处理[6]. ONSC算法利用对数增长函数的梯度(定义为∇)和海塞矩阵(定义为∇^2)进行投资组合的更新. 从算法1可以看出它只需在每次迭代时计算一个n阶矩阵的逆、一个矩阵向量的乘积和一个到单纯型上的映射. 映射这一项可以通过梯度下降法快速得到, 另外两项计算的时间复杂度都可利用矩阵逆引理[24]控制在$O(n^2)$, 所以ONSC算法的计算效率非常高.

4.3　理论分析

本节给出ONSC算法的后悔边界.

定理 1　假设市场中有一个可变参数α, 令$\eta = 0$, $\beta = \frac{\alpha}{8\sqrt{n}}$, $\delta = 1$则ONSC算法有以下后悔边界,

$$\mathrm{Regret(ONSC)} \leqslant \frac{1}{\beta}\log\left[\frac{nT}{\alpha^2}\right] + \frac{\beta}{2} + 2\eta T.$$

证　明　首先, 将对数函数$\log(\boldsymbol{b}\cdot\boldsymbol{x}_t)$在$\boldsymbol{b}_t$点作二阶泰勒展开, 得

$$\log(\boldsymbol{b}\cdot\boldsymbol{x}_t) = \log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t) + \nabla_t^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_t) - \frac{1}{2}[\nabla_t^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_t)]^2.$$

又根据损失函数$f_t(\boldsymbol{b})$的定义可以得出, 对所有$\boldsymbol{b}\in\Delta_n$, $\log(\boldsymbol{b}\cdot\boldsymbol{x}_t) \leqslant f_t(\boldsymbol{b})$, 则

$$\max_{\boldsymbol{b}\in\Delta_n}\sum_t \log(\boldsymbol{b}\cdot\boldsymbol{x}_t) - \log(\boldsymbol{b}_t\cdot\boldsymbol{x}_t) \leqslant \max_{\boldsymbol{b}\in\Delta_n}\sum_t f_t(\boldsymbol{b}) - f_t(\boldsymbol{b}_t). \tag{12}$$

所以要计算后悔边界只需计算等式(12)右边的式子的上界.

由文献[25]中简单的推导可知, 对任意$\boldsymbol{b}$, 有$-\frac{\beta}{2}\|\boldsymbol{b}_1\|^2 + \sum_t f_t(\boldsymbol{b}_{t+1}) \geqslant \sum_t f_t(\boldsymbol{b}) - \frac{\beta}{2}\|\boldsymbol{b}\|^2$; 再由文献[6]中的引理4可知$\sum_{t=1}^T[f_t(\boldsymbol{b}_{t+1}) - f_t(\boldsymbol{b}_t)] \leqslant -\frac{1}{\beta}n\log\left[\frac{nT}{\alpha^2}\right]$; 又因为$\frac{\beta}{2}[\|\boldsymbol{b}\|^2 - \|\boldsymbol{b}_1\|^2] \leqslant \frac{\beta}{2}$, 于是可以得到

$$\mathrm{Regret(Alg)} \leqslant \frac{1}{\beta}\log\left[\frac{nT}{\alpha^2}\right] + \frac{\beta}{2}. \tag{13}$$

最后对投资组合进行平滑处理, 由文献 [1] 的定理 2 推导得出

$$\mathrm{Regret(ONSC)} \leqslant \frac{1}{\beta}\log\left[\frac{nT}{\alpha^2}\right]+\frac{\beta}{2}+2\eta T, \tag{14}$$

其中 $\alpha \geqslant \frac{\eta}{n}$.

从定理 1 的结果可以看出, ONSC 算法获得了次线性后悔边界 $O(\log(T))$, 说明随着交易期数的增加, 后悔值会逐渐趋于 0, 即 ONSC 的累计收益增长率将逐渐接近事后最优的 BCRP. 也就是说, 随着时间的推移, 不断得到新的数据使得 ONSC 和选定的最优策略 BCRP 之间的差异越来越小, 逐步逼近离线最优解.

5 实证研究

本节用 ONS、CRP、SCRP、UP、SUP 与 ONSC 作比较, 根据平均净收益和周转率来评估算法的性能, 比较 ONS 与 ONSC 可以看到是否考虑交易成本对一个算法性能的影响. CRP 和 UP 是在线投资组合选择领域的经典策略, SCRP 和 SUP 分别是这两个策略考虑交易成本的版本, 其中 SUP 代表了本领域的最优结果.

5.1 数据集

实验在 4 个真实市场的数据集上进行, 分别是标普 500 指数 SP500, 纽交所两个时间段的数据 NYSE(O) 和 NYSE(N)、德黑兰指数 TSE, 具体信息见表 1. 纽交所的数据在时间范围上更广, 有新、老两个时间段, 新一阶段的股票数量根据市场变化也有所更新, 是值得用于实证研究的数据集. 这里选取 1 d 作为一个投资期.

表 1 实验数据集

Tab. 1 Databases used for experiments

数据集	时间范围	天数/d	股票数/只
SP500	1998.01.02−2003.01.30	1 276	25
NYSE(O)	1962.07.03−1984.12.31	2 826	36
NYSE(N)	1985.01.01−2010.06.30	6 431	23
TSE	1994.01.04−1998.12.31	1 259	88

5.2 评价指标

本文采用了两种指标来评估 ONSC 与其他策略对比的实验效果.

(1) 平均净收益: 净收益表示累计收益扣除交易成本, 本文每次选取 3 只股票, 平均 50 次实验的净收益. 这是度量一个算法的标准指标, 显然平均净收益越大越好.

(2) 周转率: 每期交易的平均资产百分比. 可以用来做投资算法的稳定性分析, 周转率越小表示策略的稳定性越高.

5.3 实验及结果

实验的做法是从每个数据集中随机选取 3 只股票, 这样随机选取 50 次, 并采用 ONSC、ONS、CRP、SCRP、UP、SUP 等 6 种策略来做投资. 这里 ONS 与 ONSC 的参数设置为 $\eta = 0$, $\beta = 1$, $\delta = 0.125$, ONSC 的交易成本控制参数 $\alpha = 10\ 000$. 表 2、表 3、表 4 和表 5 分别表示了当交易成本率分别为 $c = 0.05$, 0.02, 0.01, 0.001, 0 时, 6 种不同的策略在 4 个数据集上获得的平均净收益. 在 4 个数据集的结果中, 除了在 NYSE(O) 上当交易成本率为 0 时 ONSC 获得的平均净收益略低于 ONS 以外, 其他情况下 ONSC 的表现比其他策略都要好, 获得的平均净收益最高.

图 1、图 2、图 3 和图 4 展示了当 $c = 0.05$ 时 6 种策略在 4 个数据集上的周转率. 从 4 这张图可以看出 ONSC 得到了最小的周转率, 远低于 UP 和 CRP. 表 6 展示了 4 个图的数值结果, 可以

看出在数据集NYSE(O)与NYSE(N)上, ONSC的周转率比剩余策略中周转率最低的SCRP还要低一个数量级. 而在SP500中, ONSC的周转率只是剩余策略中周转率最低的SUP的1/4. 在TSE中, ONSC的周转率也只是剩余策略中周转率最低的SCRP的一半不到. 结合以上两个实验, ONSC有较小的周转率和较大的净收益, 而且只需较低的交易成本.

表 2　在数据集 SP500 上 50 次独立试验 (c=0.05, 0.02, 0.01, 0.001, 0) 的平均净收益

Tab. 2　Average net wealth for 50 independent trails (c = 0.05, 0.02, 0.01, 0.001, 0) on the SP500 dataset

策略	c				
	0.05	0.02	0.01	0.001	0
ONSC	2.146	2.235	2.265	2.293	2.296
ONS	0.582	1.133	1.419	1.740	1.823
UP	1.196	1.603	1.804	2.016	2.091
SUP	1.790	1.836	1.880	1.915	1.987
CRP	0.855	1.448	1.727	2.024	2.060
SCRP	1.651	1.789	1.841	1.890	1.895

表 3　在数据集 NYSE(O) 上 50 次独立试验 (c=0.05, 0.02, 0.01, 0.001, 0) 的平均净收益

Tab. 3　Average net wealth for 50 independent trails (c = 0.05, 0.02, 0.01, 0.001, 0) on the NYSE(O) dataset

策略	c				
	0.05	0.02	0.01	0.001	0
ONSC	38.562	38.866	40.315	40.724	40.770
ONS	4.379	16.158	25.557	38.959	40.786
UP	22.911	23.705	27.583	29.520	35.111
SUP	29.079	30.640	32.393	34.221	36.702
CRP	3.562	14.532	23.587	2.024	40.125
SCRP	19.381	23.708	25.673	27.429	27.752

表 4　在数据集 NYSE(N) 上 50 次独立试验 (c=0.05, 0.02, 0.01, 0.001, 0) 的平均净收益

Tab. 4　Average net wealth for 50 independent trails (c = 0.05,0.02,0.01,0.001,0) on the NYSE(N) Dataset

策略	c				
	0.05	0.02	0.01	0.001	0
ONSC	27.014	28.327	28.778	29.190	29.236
ONS	1.468	6.659	11.238	18.128	19.125
UP	10.521	12.385	16.091	20.400	22.137
SUP	14.931	15.943	16.849	17.867	19.636
CRP	1.187	6.823	12.266	20.828	22.092
SCRP	11.679	14.592	15.930	17.372	17.549

表 5 在数据集 TSE 上 50 次独立试验 (c=0.05, 0.02, 0.01, 0.001, 0) 的平均净收益

Tab. 5 Average net wealth for 50 independent trails ($c = 0.05, 0.02, 0.01, 0.001, 0$) on the TSE dataset

策略	c				
	0.05	0.02	0.01	0.001	0
ONSC	2.812	2.931	2.971	3.008	3.012
ONS	0.430	0.887	1.131	1.408	1.443
UP	1.149	1.297	1.591	1.816	2.143
SUP	1.671	1.841	1.952	2.090	2.219
CRP	0.781	1.216	1.414	1.621	1.646
SCRP	1.694	1.815	1.860	1.899	1.907

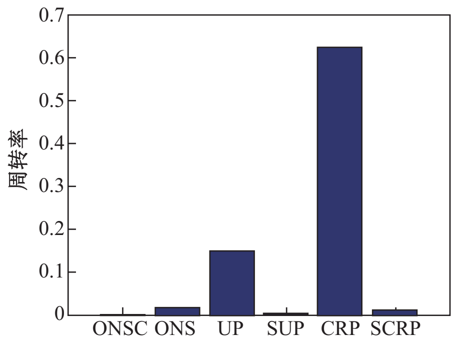

图 1 6 个策略在数据集 SP500 上的周转率结果 ($c = 0.05$)

Fig. 1 The turnover results for six strategies on the SP500 dataset ($c = 0.05$)

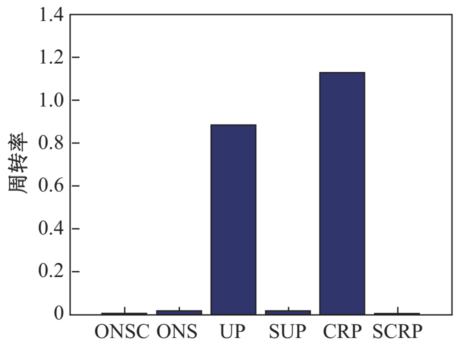

图 2 6 个策略在数据集 NYSE(O) 上的周转率结果 ($c = 0.05$)

Fig. 2 The turnover results for six strategies on the NYSE(O) dataset ($c = 0.05$)

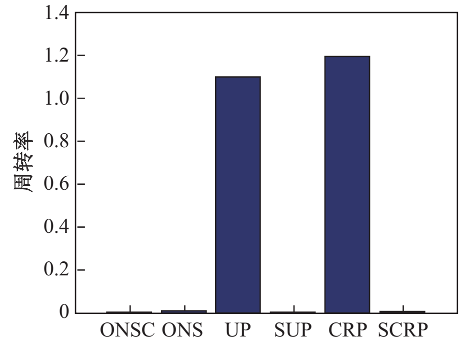

图 3　6 个策略在数据集 NYSE(N) 上的周转率结果 ($c = 0.05$)

Fig. 3　The turnover results for six strategies on the NYSE(N) dataset ($c = 0.05$)

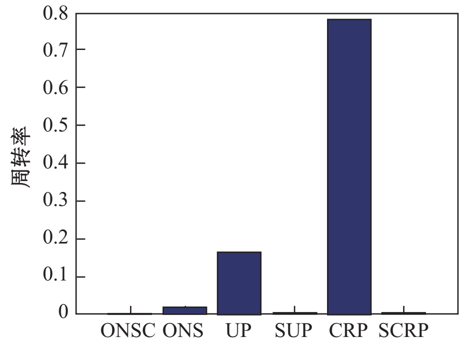

图 4　6 个策略在数据集 TSE 上的周转率结果 ($c = 0.05$)

Fig. 4　The turnover results for six strategies on the TSE dataset ($c = 0.05$)

表 6　6 个策略在 4 个数据集上的周转率的数值结果

Tab. 6　The numerical turnover results of six strategies on the four datasets

策略	数据集			
	SP500	NYSE(O)	NYSE(N)	TSE
ONSC	0.001 18	0.000 54	0.000 25	0.001 09
ONS	0.017 32	0.014 21	0.007 80	0.019 00
UP	0.148 63	0.882 00	1.096 39	0.164 89
SUP	0.004 62	0.015 71	0.002 73	0.004 44
CRP	0.622 26	1.123 00	1.187 33	0.780 97
SCRP	0.011 05	0.002 83	0.003 63	0.002 69

6　结论与展望

本文提出了一种新的在线投资组合选择策略——在线牛顿步交易成本策略 (ONSC), 基于 ONS 的思想, 充分利用投资组合向量的二阶信息构造优化函数, 并在优化函数中添加了交易成本惩罚项, 控制了交易量和交易成本的大小. 该方法具备 ONS 的二阶信息优势, 且能够适应存在交易成本的真实市场. 本文对 ONSC 对应的算法进行了理论分析, 得到后悔边界 $O(\log(T))$, 并在 4 个真实数据集上对多个股票进行了大量实验, 发现 ONSC 得到的累计净收益明显高于其他在线投资组合选择策略, 并且获得了最低的周转率, 其对应的算法表现出良好的稳定性.

[参 考 文 献]

[1] HELMBOLD D P, SCHAPIRE R E, SINGER Y, et al. Online portfolio selection using multiplicative updates [J]. Mathematical Finance, 1998, 8(4): 325-347.

[2] GYÖRFI L, LUGOSI G, UDINA F. Nonparametric kernel-based sequential investment strategies [J]. Mathematical Finance, 2010, 16(2): 337-357.

[3] THEODOROS TSAGARIS, AJAY JASRA, NIALL ADAMS. Robust and adaptive algorithms for online portfolio selection [J]. Quantitative Finance, 2012, 12(11): 1651-1662.

[4] LI B. PAMR: Passive aggressive mean reversion strategy for portfolio selection [J]. Machine Learning, 2012, 87(2): 221-258.

[5] LI B, HOI S C H, SAHOO D, et al. Moving average reversion strategy for on-line portfolio selection [J]. Artificial Intelligence, 2015, 222(1): 104-123.

[6] AGARWAL A, HAZAN E, KALE S, et al. Algorithms for portfolio management based on the Newton method [C]//Proceedings of the 23rd International Conference on Machine Learning. 2006: 9-16.

[7] ORDENTLICH E, COVER T M. Online portfolio selection [C]//Proceedings of the 9th Annual Conference on Computational Learning Theory. 1996: 310-313.

[8] 胡海鸥. 货币理论与货币政策 [M]. 上海: 上海人民出版社, 2004.

[9] DAVIS M H A, NORMAN A R. Portfolio selection with transaction costs [J]. Mathematics of Operations Research, 1990, 15(4): 676-713.

[10] ALBEVERIO S, LAO L J, ZHAO X L. Online portfolio selection strategy with prediction in the presence of transaction costs [J]. Mathematical Methods of Operations Research, 2001, 54(1): 133-161.

[11] LI B, WANG J L, HUANG D J, et al. Transaction cost optimization for online portfolio selection [J]. Quantitative Finance, 2018, 18(8): 1411-1424.

[12] BLUM A, KALAI A. Universal portfolios with and without transaction costs [J]. Machine Learning, 1999, 35(3): 193-205.

[13] KOZAT S S, SINGER A C. Universal semiconstant rebalanced portfolios [J]. Mathematical Finance, 2011, 21(2): 293-311.

[14] HUANG D J, ZHU Y, LI B, et al. Semi-universal portfolios with transaction costs [C]// Proceedings of the 24th International Conference on Artificial Intelligence. AAAI Press, 2015: 178-184.

[15] MARKOWITZ H. Portfolio selection [J]. Journal of Finance, 1952, 7(1): 77-91.

[16] KELLY J L. A new interpretation of information rate [J]. Bell System Technical Journal, 1956, 35 (4): 917-926.

[17] LI B, HOI S C H. Online portfolio selection: A survey [J]. Papers, 2012, 46(3):1-36.

[18] COVER T M. Universal portfolios [J]. Mathematical Finance, 1991(1): 1-29.

[19] DAS P, JOHNSON N, BANERJEE A. Online lazy updates for portfolio selection with transaction costs [C]//27th AAAI Conference on Artificial Intelligence. AAAI Press, 2013: 202-208

[20] BOYD S, PARIKH N, CHU E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers [J]. Foundations & Trends in Machine Learning, 2010, 3(1): 1-122.

[21] FAN R E, CHEN P H, LIN C J, et al. Working set selection using second order information for training support vector machines[J]. Journal of Machine Learning Research, 2005, 6(4):1889-1918.

[22] LI B, HOI S C H, ZHAO P, et al. Confidence weighted mean reversion strategy for online portfolio selection [J]. ACM Transactions on Knowledge Discovery From Data, 2013, 7(1): 1-38.

[23] HOI S C H, SAHOO D, LU J, et al. Online learning: A comprehensive survey [J]. arXiv: 1802. 02871v2[cs.LG]22 Oct 2018.

[24] BROOKES M. The matrix reference manual [R/OL]. [2018-06-30]. http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html.

[25] HAZAN E, AGARWAL A, KALE S. Logarithmic regret algorithms for online convex optimization [J]. Machine Learning, 2007, 69(2/3): 169-192.

(责任编辑: 李 艺)