



北京大學

# 计算概论股票交易大作业指导文档

# 目录

<b>1</b>	<b>项目概述</b>	<b>2</b>
1.1	项目目标 . . . . .	2
1.2	技术栈 . . . . .	2
1.3	项目结构 . . . . .	2
<b>2</b>	<b>项目要求</b>	<b>3</b>
2.1	股票市场 . . . . .	3
2.2	交易策略 . . . . .	3
2.2.1	turtle-strategy . . . . .	3
2.2.2	movingaverage-strategy . . . . .	4
2.2.3	abcd-strategy . . . . .	4
2.2.4	diy-strategy . . . . .	5
2.3	测评和调试 . . . . .	5
2.4	报告撰写 . . . . .	6
2.5	完成建议 . . . . .	6
<b>3</b>	<b>评分标准</b>	<b>6</b>
<b>4</b>	<b>提交方式</b>	<b>6</b>

# 1 项目概述

在本次大作业中，你将使用 Python 实现在一个简单的模拟股票市场中进行交易的策略，并且评估策略的收益（注：策略收益的多少并不作为评分的重点，请同学们不必过分纠结）。**请同学们先阅读次文档再开始本次作业，本次大作业不需要任何和股票相关的前置知识，按照文档的说明按部就班地完成作业，就可以获得 90% 以上的分数。**当然，在策略上采用了更高级的方法、对本次大作业提出了有效的建议，甚至在不破坏给定规则的前提下发现了股票市场中的漏洞从而赚到大量的钱都可以获得少量额外加分，**但本次大作业总分不会突破 100 分的上限，所以请同学们认真完成基础要求，根据自己的兴趣进行一些简单的探索，不必做无意义的内卷。**必做内容并不困难，请同学们按照文档，合理安排时间，相信自己的能力。

## 1.1 项目目标

本次大作业中，你不需要处理数据读取、交易过程以及策略收益测评等细节，这些在提供的代码中都已经实现，**你只需要补全 3 个已定义策略的有关函数以及 1 个自定义策略的有关函数即可**，这些函数都和交易的策略相关，在后续的内容中会对几种交易策略进行详细的介绍。

## 1.2 技术栈

本次作业不需要使用任何课程以外的内容，使用课程相关的知识即可完成，但是学习一些 pandas、matplotlib 以及 numpy 的使用方式能让你更加快速、简洁、优美地实现所需的功能（代码的风格不作为评分的标准，只需要正确实现，一定程度上保证可读性即可）。虽然本次作业不一定需要使用这些包，但为了正确运行程序我们仍需正确安装 pandas 和 matplotlib 以及 seaborn。在命令行中运行 `pip install pandas matplotlib seaborn` 即可。

## 1.3 项目结构

以下是下载压缩包中的项目结构

```
1 project/  
  |-- src/  
3   |   |-- trade.py  
   |   |-- visualize.py  
5   |   |-- strategy.py  
  |-- img/  
7  |-- data/  
   |   |-- dataset.csv  
9  |-- README.pdf  
  |-- main.py
```

其中 `main.py` 是项目的主要文件，你可通过调用该文件进行算法的测试或者获取策略的收益结果，结果会以图片的形式出现在 `img` 文件夹中。文件夹中包含了 `README.pdf`，即本文档。`data` 文件夹中包含了 `dataset.csv`，即本次测试所需使用的数据。`src` 文件夹中包含了 `trade.py`、`visualize.py`、`strategy.py` 三个文件，其中 `trade.py` 文件中实现了市场中交易的一些策略，**在完成作业的过程中不需要改动**；`visualize.py` 文件中实现有关结果可视化的一

些函数，在完成作业的过程中不需要改动；`strategy.py` 中有四个需要补全的函数，这就是本次作业中需要完成的部分。有关这四个函数补全的要求将在2项目要求中进行详细的说明。

## 2 项目要求

本章节将对模拟的股票市场、所需实现的策略、以及如何进行测试和获得结果进行详细的说明，请认真阅读此部分。

### 2.1 股票市场

为了简化本次大作业，我们采用一个非常简单的市场进行模拟。首先我们假设市场中只有一只股票和一个投资者（你自己），并且我们以股票的真实的收盘价作为股票当天的价格，该价格在当日是不会波动的。同时我们也对交易的部分做了一些简化，首先投资者每天只能进行三个动作买入一手股票、卖出一手股票或者什么都不做。当然，当你的资金不足时你无法买入股票，当你没有任何持股时你也不能卖出股票。并且最终计算收益时，只关注手中的资金，并不会将手中的股票折合成资金再进行计算。我们定义了四种市场，分别是 `poor`（贫穷），`normal`（正常），`rich`（富有），`inexhaustible`（用之不竭），分别代表了不同的起始资金。

作为一个真实的投资人，我们并不知道交易会在哪一天结束（虽然我们在作业中以一个上帝视角得知了这一点），所以我们不能采用毫无道理地在交易结束前的若干天抛售掉自己手中股票的策略。除此之外，作为一个绝对理性的投资人，我们的策略只和股票的价格有关，也就是说，我们设计的策略不应该与自己手中的资金数目以及当前交易是哪一天相关。同时，在一个真实的场景中，我们无法预知未来，所以我们的策略应该只和当前的股价以及历史的股价相关。违反以上黑体字的三点规则的策略将不被认为是一个正确的策略（即使你取得了很高的收益）。

### 2.2 交易策略

首先，所有的策略函数的返回值都是一个 `list`，`list` 的长度等于交易的天数，`list` 每个位置的取值（也就是每天的行动）只有三种，分别是 0（什么都不做）、1（买入一手股票）、-1（卖出一手股票）。在你自己设计的策略中，返回的 `list` 也应该满足这个要求，不满足要求的 `list` 会在检查时报错。对于一个满足要求的 `list`，我们将其称为 `signal`，1 称为买入信号，-1 称为卖出信号。再次强调，在确认每天是买入还是卖出时，不需要考虑自己的资金状况（例如资金不足时你仍可以设置买入信号，只是当天不会交易）或者持股状况（例如没有持股时仍可以设置卖出信号，只是当天不会交易），因为你的策略，或者说买入信号和卖出信号的设置只和股票的当日价格以及历史价格有关。

#### 2.2.1 turtle-strategy

海龟策略有一个参数  $w$ ，代表了 `window_size`。简单来说，海龟策略比较的是今日的价格和上一个窗口的最高价和最低价。那么如何定义上一个窗口，上一个窗口一般是指，以昨日为结束的窗口。例如今天是第 6 个交易日， $w = 3$ ，那么上一个窗口中包括了第 3、第 4、第 5 日的历史价格。如果历史的天数小于  $w$ ，则一直取到第一天即可。因为第一天时没有任何历史信息，我们在第一天不进行任何操作。

所以，海龟策略总体来说可以描述为，如果今天的价格大于上一个窗口中的最高价，则当天为卖出信号，如果今天的价格小于上一个窗口中的最低价，则当天为买入信号。也就是：

$$\text{if } RW_{t-1}.min > Price_t, \text{Signal}_t = \text{buying}$$

$$\text{if } RW_{t-1}.max < Price_t, \text{Signal}_t = \text{selling}$$

### 2.2.2 movingaverage-strategy

移动平均策略有两个参数  $sw$  和  $lw$ ，分别代表 short\_window 和 long\_window 的窗口长度，此处窗口的定义与海龟策略中的相同。我们对于每个窗口中的价格求平均值，就得到了每天的短期平均价格和长期平均价格，将他们连起来就得到了短期均线和长期均线，如下图所示：

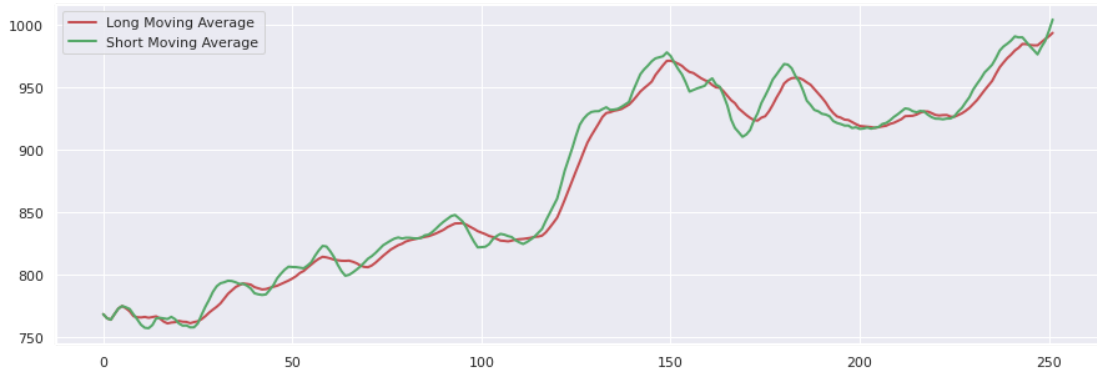


图 1: 移动均线的例子

移动平均策略可以描述为，当短期均线从下方穿过长期均线来到上方时，最近的下一个交易日设置为买入信号，当短期均线从上方穿过长期均线来到下方时，最近的下一个交易日设置为卖出信号。也就是：

$$\text{if } SMA_{t-1} < LMA_{t-1} \text{ and } SMA_t > LMA_t, \text{Signal}_t = \text{buying}$$

$$\text{if } SMA_{t-1} > LMA_{t-1} \text{ and } SMA_t < LMA_t, \text{Signal}_t = \text{selling}$$

### 2.2.3 abcd-strategy

abcd 策略有两个参数  $w$  和  $skip\_loop$ ， $w$  和其他策略一样，代表窗口长度， $skip\_loop$  代表了向后查找时每一步的步长。

在 abcd 策略中，我们先以  $w$  为窗口长度，和移动平均策略一样获得一条均线  $MA$ 。而 abcd 策略就是在均线上查找满足条件的 abcd 四元组，并设置交易信号。abcd 四元组的含义如下：a 是起始点，所以 a 需要遍历每一个交易日，b 满足  $b \in \{a + i * skip\_loop | i = 0, 1, 2, \dots, MA_b > MA_a\}$ ，c 满足  $c \in \{b + i * skip\_loop | i = 0, 1, 2, \dots, MA_c < MA_b, MA_c > MA_a\}$ ，d 满足  $d \in \{c + i * skip\_loop | i = 0, 1, 2, \dots, MA_d > MA_b\}$ ，查找完所有的四元组之后，我们可以得到四个集合  $Set_a, Set_b, Set_c, Set_d$  分别代表 a, b, c, d 的集合。信号设置规则如下：

$$\forall t \in (Set_a \cap Set_c) - (Set_b \cap Set_d), \text{Signal}_t = \text{buying}$$

$$\forall t \in (Set_b \cap Set_d) - (Set_a \cap Set_c), \text{Signal}_t = \text{selling}$$

### 2.2.4 diy-strategy

在 diy 策略中，你需要自己设计一个策略，在报告中讲明白自己的设计思路即可，策略不需要很高收益。但需要注意，策略需要满足市场的规则以及最终的返回值应该是与交易天数相等的一个 list，每个位置是每个交易日的 signal。

## 2.3 测评和调试

本次作业可以通过运行 main.py 文件来进行测评和调试，main.py 文件有如下几个参数：

- --strategy: 交易策略，有四种取值，分别是 turtle, movingaverage, abcd, diy，代表了不同的交易策略。
- --market\_type: 市场类型，有四种取值，分别是 poor, normal, rich, inexhaustible，代表了不同的起始资金。
- --verbose: 是否输出调试信息，有两种取值，0 和 1，默认是 0，不输出调试信息。
- --arg1: 策略的第一个参数
- --arg2: 策略的第二个参数（某些策略可能不需要）
- --arg3: 策略的第三个参数（某些策略可能不需要）
- --arg4: 策略的第四个参数（某些策略可能不需要）

如果你的 diy 函数需要更多的参数可自行添加例如，当你在 rich 市场下测试 abcd 策略，并且想看到调试信息，将 abcd 策略的第一个参数 w 设置为 7，第二个参数 skip\_loop 设置为 4，你需要在命令行中执行以下命令：python main.py --strategy abcd --market\_type rich --verbose 1 --arg1 7 --arg2 4，当程序正确运行完毕后，你将会在 img 文件夹下找到策略名 + 市场类型.png，上面这个例子中会生成 abcd\_rich.png，该图片的内容类似于下图：

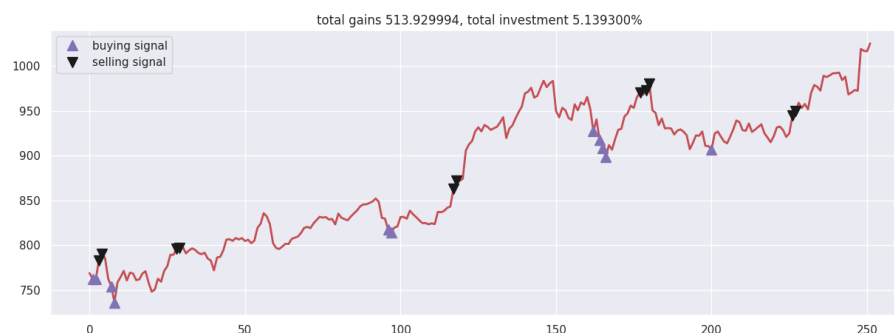


图 2: 运行结果的例子

图中红色的线是股票的价格波动，紫色三角形代表了一次成功的买入（signal 中设置了买入信号可能因为资金不足买入失败），黑色三角形代表了一次成功的卖出（signal 中设置了卖出信号可能因为没有持股卖出失败）。上面记录了总收益和总收益率。

## 2.4 报告撰写

报告中需要包括以下部分：

- 关于以上 4 种策略实现的介绍，必要时可结合代码截图，并对其中关键语句进行说明。
- 关于以上 4 种策略的结果的展示和分析，分析主要包括不同初始资金对参数选取的影响，分析部分可以畅所欲言，不一定需要正确也可以给分。

报告对格式无要求，你可以任何喜欢的方式组织实验报告。报告的主要目的是为了判断本次作业是否由自己独立完成，可以采用一边写代码一边记录的流水账方式或者像一份正式的实验报告那样有严谨的格式。在实验报告里你也可以写其他任何你想写的内容：本次作业有什么困难、有什么建议、有什么想对助教说的话都可以写出来。

## 2.5 完成建议

可以看到，三个策略的核心都和 window 有关，所以在开始完成策略的代码之前，可以考虑如何维护一个窗口，以及计算这个窗口的最大值、最小值和均值。此处你可以使用一个 list 并通过删除和添加元素来维护该滑动窗口，当然你也可以使用 pandas 中的 rolling 函数，或者任何你喜欢的方式。

## 3 评分标准

本次大作业的评分分为两部分，分别为代码（80 分）和报告（20）分。其中代码部分，正确实现海龟策略（25 分），正确实现移动平均策略（25 分），正确实现 abcd 策略（25 分），正确实现一个 diy 策略（5 分）；报告部分，介绍海龟策略实现的方法并展示结果，讨论不同初始资金对参数选取的影响（5 分），介绍移动平均策略实现的方法并展示结果，讨论不同初始资金对参数选取的影响（5 分），介绍 abcd 策略实现的方法并展示结果，讨论不同初始资金对参数选取的影响（5 分），介绍 diy 策略实现的方法并展示结果（5 分）。

如果策略实现错误将根据实现的具体情况酌情给分，但违反以上三点规则只能在该策略上最多拿到 60% 的分数。

## 4 提交方式

本次作业只需要提交 strategy.py 文件和 pdf 格式的报告，报告请命名为学号 + 姓名 + 大作业报告.pdf，将这两个文件放在一个压缩包中，压缩包命名为股票交易 + 学号 + 姓名。请同学们独立完成本次作业，不要抄袭。