南開大學

《Python与数据分析》 课 程 论 文 （设 计）

中文题目： 多因子选股模型基于 LASSO 和 MLP 的改进与创新

外文题目： Improvement and Innovation of Multi-Factor Model based on LASSO and MLP

小　　组： 第七组

年　　级： 2016 级

指导教师： 王恺

完成日期： 2019 年 12 月

目录

小组全体成员：

学号	姓名	专业	学院
1610006	迟贺元	统计学	数学科学学院
1611980	郝若馨	经管法	经济学院
1612428	张瀚文	国际会计	商学院
1612435	胡濒午	国际会计	商学院

第一章　多因子模型理论及背景

在现代投资组合理论中，投资组合获得的收益可以分为两个部分，即来自市场的收益和超出市场的收益，也就是 *Beta* 和 *Alpha*。而预测金融市场的走势是较为困难的，学术界和业界的探索集中在了获取超额收益的方法，即如何准确地定义和寻找金融市场中的 *Alpha*。现在已有的模型包括资本资产定价模型 CAPM 、Fama-French 三因子模型及五因子模型等，而多因子模型 MFM 正是基于套利定价模型 APT 而建立的更为完善的定价模型。

多因子模型从其构建的目标上可分为两种，以 Barra 为代表的用于投资组合业绩归因的风险模型，和用于预测未来股票收益的 *Alpha* 模型，本篇论文主要讨论的是后者中的因子选择和预测收益的方式。

第一节　CAPM 与 Fama-French

资本资产定价模型 (CAPM) 是由威廉·夏普等人于 1964 年提出的，它是在马科·维茨的现代投资组合理论的基础上发展起来的：

$$E(r_p) = r_F + \beta_p \cdot (r_M - r_F) \tag{1.1}$$

其中：

- r_p 代表资产 p 的收益率
- r_F 代表无风险收益率
- r_M 代表市场组合的 (基准) 收益率

CAPM 模型假定，资产 p 的收益率只与其余市场的相关系数 β_p 有关，其中：

$$\beta_p = \frac{Cov(r_p, r_M)}{Cov(r_M)} \tag{1.2}$$

即资产 p 收益率与市场组合 M 收益率之间的协方差初一市场组合 M 收益率的方差。

因此我们可以将 CAPM 模型看做以市场组合 M 的收益率为因子的单因子模型。

但随着业界对股票市场研究的逐步深入，人们发现单一的因子无法很好地解释资产收益的来源，实际值与 (1.1) 的结果相差较大，包含更多因子的模型被相继提出。例如，Famma/French 在 1992 年证明了市净率 PB 与市值因子对股票的收益率有十分显著的影响，并基于此建立了 Famma-French 三因子模型。

模型认为，资产 p 的超额收益可由它对三个因子的暴露来解释，这三个因子是：市场资产组合 $r_M - r_f$、市值因子 SMB、账面市值比因子 HML。这个三因子均衡定价模型可以表示为：

$$E(r_{pt}) - r_{ft} = \beta_p \cdot [E(r_{Mt} - r_{ft})] + s_p \cdot E(SMB_t) + h_p \cdot E(HMI_t) \tag{1.3}$$

其中：

- r_{ft} 表示时间 t 下的无风险收益率
- r_{mt} 表示时间 t 下的市场组合的收益率
- r_{pt} 表示资产 p 在时间 t 下的收益率
- $E(r_{mt}) - r_{ft}$ 是市场风险溢价
- SMB_t 为时间 t 下的市值 (Size) 因子的模拟组合收益率 (Small Minus Big)
- HMI_t 为时间 t 下的账面市值比因子的模拟组合收益率 (High inus Low)

但是，我们应该看到，三因子模型并不代表资本定价模型的完结，其中还有很多未被解释的部分。2013 年 Famma/French 发现了除了上述风险，还有盈利水平风险、投资水平风险也能带来个股的超额收益，并发表了五因子模型，但五个因子依然是远远不够的。

第二节 基于 APT 理论的多因子模型 MFM

套利定价理论 APT 用多个因子来解释资产收益，得到资产收于与多个因子之间存在近似线性关系，从而将因子从 Fama-French 的五因子进一步拓展到多因子，这为多因子模型 MFM 提供了理论基础。

多因子模型 MFM 可以理解为，将每只股票的收益率分解为 M 个因子的线性组合与未被解释的残差项。而影响股票收益的因素主要来自于其对某一个因子的因子暴露，这里指的是其对应的因子值，例如 CAPM 模型中的 β_p。我们将

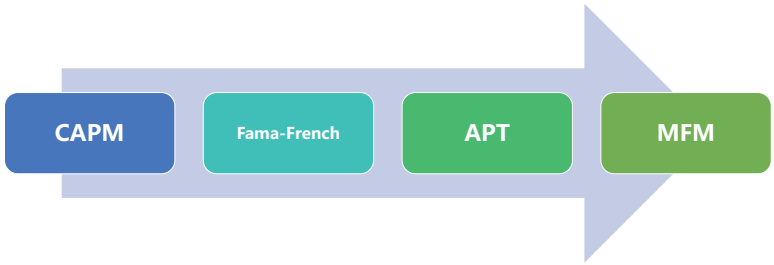


图 1.1 多因子模型 MFM 发展历程

多因子模型作如下表示：

$$\begin{aligned} r_i &= \beta_{i1} \cdot f_1 + \beta_{i2} \cdot f_2 + \cdots + \beta_{iM} \cdot f_M + \mu_i \\ &= \sum_{j=1}^{M} \beta_{ij} \cdot f_j + \mu_i \end{aligned} \tag{1.4}$$

也可以表示为向量形式：

$$\boldsymbol{r} = \boldsymbol{\beta} \cdot \boldsymbol{f} + \boldsymbol{\mu}$$

其中：

- β_{ij} 表示股票 i 在因子 j 上的因子暴露
- f_j 表示因子 j 的因子收益
- μ_i 表示股票 i 的残差收益

上式成立的前提假设为：

1. μ_j 之间两两互相独立，也就是说不同股票之间的收益率的相关性完全取决于 M 个因子的因子收益，这样的假设下我们很容易计算相关矩阵 Σ
2. 残差收益 μ_i 与 M 个因子之间均不存在相关性，残差收益间亦两两不相关，且均服从均值为 0 的正态分布

相较于 CAPM、Fama-French 等模型，多因子模型 MFM 的优势在于可以提供更为完整的风险暴露分析，更好地解释股票收益的来源，并分离出各个因子的影响，从而为投资决策提供更加详尽的分析。

第二章　多因子模型构建流程

多因子模型的构建流程包括以下几个步骤：

1. 样本的选取

   为了使模型测试的结果更具有实操性，符合实际的投资情况，我们需要对 ST、PT 及停牌等无法买入的股票进行剔除

2. 基础数据的获取与清洗

   根据因子计算的需求，获取全样本的基础数据，再进行数据的清洗。数据清晰地过程很容易被轻视或忽略，但异常值和缺失值对模型的影响是很显著的，数据清洗的步骤是很重要的，也是需要格外注意的。

3. 因子计算与单因子测试

   根据第 2 步中得到的数据，我们可以计算全体样本的全部因子的因子暴露。再进行标准化后，根据计算结果进行单因子测试，筛选出具有很好的单调性与稳定性的有效因子。具体的测试方法与框架将在下面章节中给出详细的定义。

4. 多因子模型

   作为模型构建中最重要的一个步骤，我们将利用多元线性回归利用筛选后的因子对股票进行预测。为了消除多重共线性的影响，我们还尝试了 LASSO 回归。此外，我们还将尝试 MLP
   多层感知器等一些机器学习算法，探寻一些线性模型外的方法。

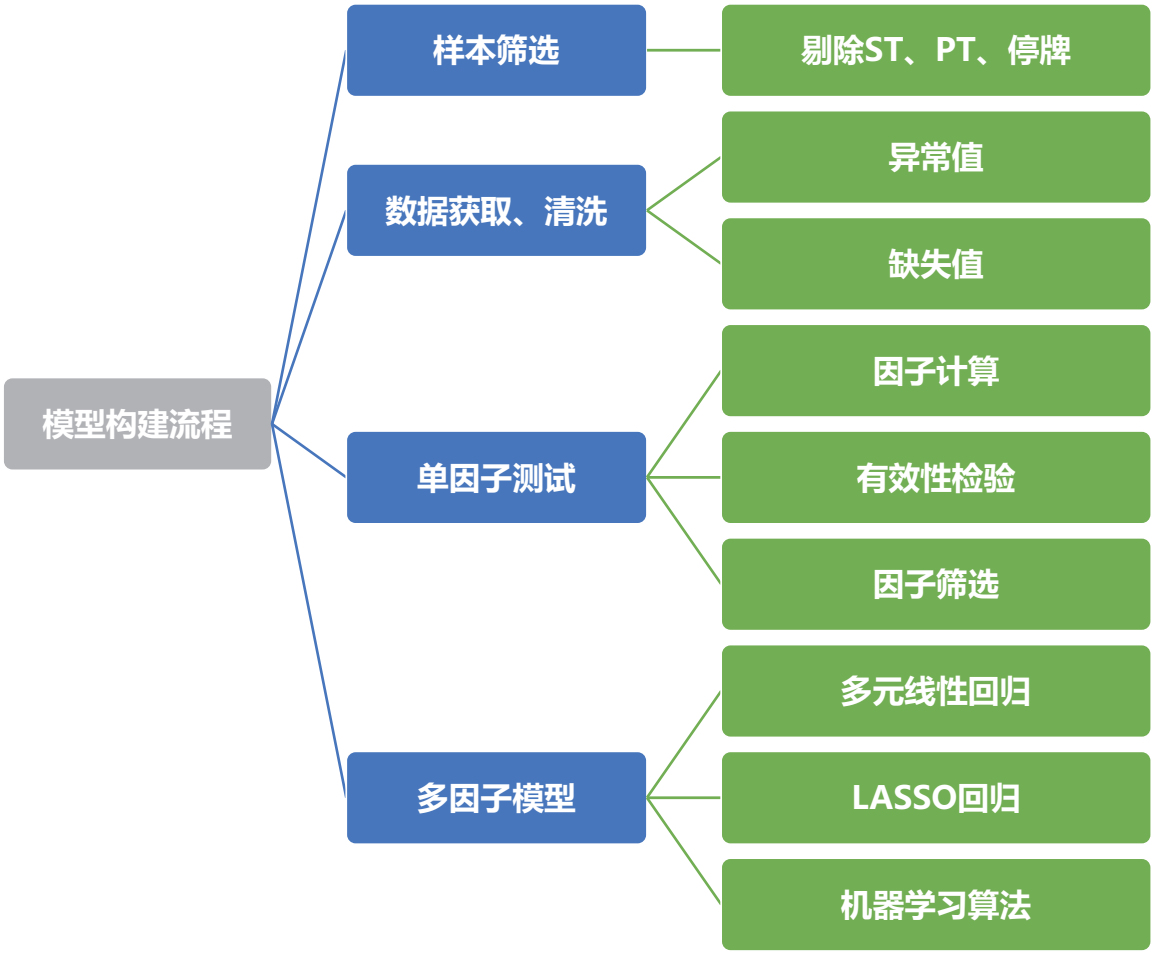

模型构建流程
样本筛选
剔除ST、PT、停牌
数据获取、清洗
异常值
缺失值
单因子测试
因子计算
有效性检验
因子筛选
多因子模型
多元线性回归
LASSO回归
机器学习算法

图 2.1 模型构建流程图

第三章　样本筛选、数据获取及单因子测试

第一节　样本筛选

- 样本范围：中国市场全体 A 股
- 样本期：2006-01-01 至 2019-03-01 共 183 个月
- 样本筛选规则：
  1. 剔除选股日的 ST、PT 股票；
  2. 剔除上市不满 3 年股票；
  3. 剔除选股日由于停牌等原因无法买入的股票；

第二节　数据清洗

数据清洗是指对数据进行的重新审查与校验的过程，目的是删除极端信息、纠正存在的错误、提供数据一致性，从而保证测试结果与模型建立的稳定性。我们从去除异常值、缺失值两方面来清理数据。

1. 异常值处理

   考虑到 3σ 去极值法在现实数据中可能存在的厚味问题，我们选择采用绝对中位数法 (MAD) 来保证模型的稳健。通过计算因子值的中位数 $median_{\beta_j}$，定义绝对中位值 MAD 为：

$$MAD_j = median(|\beta_{ij} - median_{\beta_j}|)$$

   根据 3σ 去极值法，位于

$$[median_{\beta_j} - 3 \cdot 1.4826 \cdot MAD_j, \quad median_{\beta_j} + 3 \cdot 1.4826 \cdot MAD_j]$$

   范围之外的被我们视为异常值。根据因子实际情况，我们通常将异常值设为区间上下限的数值。

2. 缺失值处理

在单因子测试中，我们针对缺失值不同的数据来源与逻辑关系，采取直接剔除或者替换为行业中位数的方法。对于缺失率小于 20% 的因子数据，我们用中信一级行业中位数代替；对于缺失率超过 20% 的因子数据，我们做剔除处理。

第三节 因子计算及标准化

利用上一节清洗后的数据进行因子计算下表中的全部因子，再进行标准化。

因子	定义	使用数据
F1	过去 12 个月营业收入/过去 12 个月平均总资产	营业收入，总资产
F2	过去 12 个月营业成本/过去 12 个月平均存货	营业成本，存活
F3	过去 12 个月净利润/过去 12 个月营业总收入	净利润，营业总收入
F4	（过去 12 个月营业收入-过去 12 个月营业支出）/过去 12 个月营业收入	营业收入，营业支出
F5	（销售费用 + 管理费用 + 财务费用）/营业收入	销售费用，管理费用，财务费用，营业收入
F6	管理费用/营业收入	管理费用, 营业收入
F7	财务费用/营业收入	财务费用, 营业收入
F8	销售费用/营业收入	销售费用, 营业收入
F9	过去十二个月的 roe	roe
F10	过去十二个月的 roa	roa
F11	过去十二个月的 roic	roic
F12	(归属母公司股东的权益-其他权益工具)/总股本	归属母公司股东的权益,其他权益工具, 总股本
F13	固定资产/归属母公司股东权益	固定资产, 归属母公司股东权益
F15	流动资产-流动负债/总资产	流动资产, 流动负债, 总资产
F16	现金/总资产	现金, 总资产
F27	经营活动产生的现金净流量/净利润	经营活动产生的现金净流量, 净利润
F28	企业自由现金流/总股本	息税前利润, 折旧与摊销, 营运资金增加, 购建固定资产, 无形资产和其他长期资产支付的现金, 总股本
F29	企业自由现金流/总市值	息税前利润, 折旧与摊销, 营运资金增加, 购建固定资产, 无形资产和其他长期资产支付的现金, 总股本
F30	流通股本/总股本	流通股本, 总股本

因子	定义	使用数据
F31	流动资产/流动负债	流动资产, 流动负债
F32	速动资产/流动负债	流动资产, 流动负债
F33	资产/负债	资产, 负债
F34	流动负债/总资产	流动负债, 总资产
F35	非流动负债/总负债	非流动负债, 总负债
F36	负债合计/有形资产	负债合计, 有形资产
F37	非流动负债/归属母公司股东的权益	非流动负债, 归属母公司股东的权益
F38	股东权益 (报告期)/总市值	股东权益 (报告期), 总市值
F39	市盈率倒数	市盈率
F40	未来 12 个月一致预期市盈率倒数	未来 12 个月一致预期市盈率
F44	市销率倒数	市销率
F45	peg	市盈率, 一致预期增长
F47	市现率倒数	市现率
F48	经营现金流/总市值	经营现金流, 总市值
F49	股息率	过去一年现金分红, 总市值
F50	调整过的总市值	总市值, 账面价值
F51	BPS 同比增长率	每股帐面价值
F52	股东权益同比增长率	股东权益
F53	总资产增长率	总资产
F54	每股营业收入同比增长率	每股营业收入
F55	单季度每股营业收入同比增长率	每股营业收入
F58	每股经营现金流同比增长率	每股经营现金流
F59	单季度每股经营现金流同比增长率	每股经营现金流
F60	净利润同比增长率	净利润
F61	单季度净利润同比增速	净利润
F62	营业收入同比增速	营业收入
F63	单季度营业收入同比增速	营业收入
F64	经营利润同比增速	经营利润
F65	单季度经营利润同比增速	经营利润

因子	定义	使用数据
F66	经营现金流同比增速	经营现金流
F67	单季度经营现金流同比增速	经营现金流
F68	roe 同比增速	roe
F69	(营业利润 ttm-过去八个季度的营业利润 ttm 均值)/过去八个季度的营业利润 ttm 标准差	营业利润
F70	(营业收入 ttm-过去八个季度的营业收入 ttm 均值)/过去八个季度的营业收入 ttm 标准差	营业收入
F71	(营业利润同比-过去八个季度的营业利润同比均值)/过去八个季度的营业利润同比标准差	营业利润
F72	(营业收入同比-过去八个季度的营业收入同比均值)/过去八个季度的营业收入同比标准差	营业收入
F73	(净利润同比-过去八个季度的净利润同比均值)/过去八个季度的净利润同比标准差	净利润
F74	(roe_ttm-过去八个季度的 roe_ttm 均值)/过去八个季度的 roe_ttm 标准差	roe
F75	(营业收入-过去八个季度的营业收入均值)/过去八个季度的营业收入标准差	营业收入
F76	(净利润 ttm-过去八个季度的净利润 ttm 均值)/过去八个季度的净利润 ttm 标准差	净利润
F77	(经营现金流净额 ttm-过去八个季度的经营现金流净额 ttm 均值)/过去八个季度的经营现金流净额 ttm 标准差	经营现金流净额
F78	(营业利润-过去八个季度的营业利润均值)/过去八个季度的营业利润标准差	营业利润
F79	(经营现金流净额-过去八个季度的经营现金流净额均值)/过去八个季度的经营现金流净额标准差	经营现金流净额
F80	(净利润-过去八个季度的净利润均值)/过去八个季度的净利润标准差	净利润
F106	过去一个月平均成交额/过去三个月平均成交额	成交额
F107	过去 21 交易日平均换手率	换手率
F108	过去 63 交易日平均换手率	换手率
F109	过去 252 交易日平均换手率	换手率
F110	收盘价和换手率相关系数	收盘价, 换手率

因子	定义	使用数据
F111	Fama French 回归的残差波动率	Fama French 回归三因子
F112	1-Fama French 回归的拟合优度	Fama French 回归三因子
F113	单位成交量下价格的波动	收盘价, 成交额
F114	过去一个月平均交易量	成交量
F115	过去一年平均交易量	成交量
F116	过去一个月交易量标准差	成交量
F117	过去一年交易量标准差	成交量
F118	过去一个月平均换手率	成交量
F119	过去一年平均换手率	成交量
F120	过去一个月换手率标准差	成交量
F121	过去一年换手率标准差	成交量
F122	过去一个月平均交易量/过去一年平均交易量	成交量
F123	过去一个月交易量标准差/过去一年交易量标准差	成交量
F124	过去一个月最高价/过去一个月最低价	收盘价
F125	过去一年最高价/过去一年最低价	收盘价
F126	过去一周 vwap/过去一个月 vwap	vwap 价格
F127	过去一个月 vwap/过去半年 vwap	vwap 价格
F128	过去一个月收益率第 10% 位数	收盘价
F129	过去一个月收益率第 90% 位数	收盘价
F130	过去一个月收益率标准差	收盘价
F131	过去一年收益率标准差	收盘价
F132	过去三个月收益率标准差	收盘价
F133	过去六个月收益率标准差	收盘价
F134	过去一年收益率	收盘价
F135	过去一个月收益率	收盘价
F136	过去三个月收益率	收盘价
F137	过去六个月收益率	收盘价
F138	过去三个月最高价/最低价	收盘价
F139	过去六个月最高价/最低价	收盘价
F140	过去一个月收益率最大值	收盘价

常见的因子标准化的方法有：Z-score 标准化、Rank 标准化、min-max 标准

化等，由于后两种标准化后的数据会丢失原始样本的一些重要信息，这里我们采用 Z-score 标准化：

$$\beta_{ij}^{*}=\frac{\beta_{ij}-\mu_{\beta_j}}{\sigma_{\beta_j}}$$

第四节 单因子测试模型

并不是所有的因子都是有效因子，有效的因子应有一定的逻辑支撑，此外很重要的一点就是其与股票收益率的相关性应较为显著。挖掘和测试单因子是多因子模型构建流程中很重要的一部分，我们的因子测试体系是基于最小二乘回归和分层回溯法实现的。

截面回归是目前业界较常用于因子测试的方法，不同于全样本面板回归，其更有利于对因子变化趋势的捕捉。同时，全样本面板数据量过于庞大，很容易到日回归模型更容易通过显著性检验；由于我们选取的样本为全体 A 股，因子但其界面的样本数据一般保持在 1500 个以上，在不影响模型有效性同时更有利于我们判断因子各项指标的优劣程度。

我们选择对每期全体样本分别作最小二乘回归，回归时因子暴露 β_{ij} 为已知值，回归得到每期的因子收益 f_i，通过 183 期的回归后我们可以得到因子收益序列，同时也可以得到 t 值序列，也就是因子收益与股票收益率相关性的 t 检验得到的 t 值。

进行截面回归判断每个单因子的收益情况和显著性时，需要特别关注 A 股市场中一些显著影响个股收益率的因素，例如**行业因素**和**市值因素**。市值因子在过去的很长一段时间内都是 A 股市场上影响股票收益显著性极高的一个因子，为了能够在单因子测试时得到因子真正收益情况，我们在回归测试时对市值因子也做了剔除。

单因子测试的回归方程如下：

$$\begin{bmatrix} r_{ti} \\ \vdots \\ r_{tn} \end{bmatrix} = \begin{bmatrix} \beta_{t11} & I_{t1u} & \cdots & I_{t1v} & m_{t1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta_{tn1} & I_{tnu} & \cdots & I_{tnv} & m_{tnm} \end{bmatrix} \cdot \begin{bmatrix} f_{ti} \\ \vdots \\ f_{tn} \end{bmatrix} + \begin{bmatrix} \mu_{ti} \\ \vdots \\ \mu_{tn} \end{bmatrix} \tag{3.1}$$

其中：

- β_{tiu} 代表股票 i 在测试因子上的因子暴露
- I_{tiu} 代表股票 i 的行业因子暴露 (行业哑变量，属于该行业则因子暴露为 1，不属于该行业则因子暴露为 0)
- m_{tiu} 代表股票 i 的市值因子暴露

在回归模型上，我们选择最常用和最简单的最小二乘回归 OLS。

第四章　单因子有效性检验

采用多期截面 OLS 回归后我们可以得到因子收益序列 f_i，以及每一期回归假设检验 t 检验的 t 值序列，针对这两个序列我们将通过以下几个指标来判断该因子的有效性以及稳定性：

- 因子收益序列 f_i 的假设检验 t 值
- 因子收益序列 f_i 大于 0 的概率
- t 值绝对值的均值
- t 值绝对值大于等于 2 的概率

第一节　t 值

我们通过取 t 值绝对值序列的均值进行 t 检验来分析因子的有效性以及稳定性，之所以要取绝对值，是因为只要 t 值显著不等于 0 即可以认为在当期，因子和收益率存在明显的相关性。但是这种相关性有的时候为正，有的时候为负，如果不取绝对值，则很多正负抵消，会低估因子的有效性。此外，通过检验 $|t|>2$ 的比例保证 $|t|$ 平均值的稳定性，避免出现少数数值特别大的样本值拉高均值。

第二节　IC 的定义和参数

IC 值 (信息系数) 是指个股第 t 期在因子 i 上的因子暴露 (剔除行业与市值后) 与 $t+1$ 期的收益率的相关系数。通过计算 *IC* 值可以有效的观察到某个因子收益率预测的稳定性和动量特征，以便在优化组合时用作筛选的指标。常见的计算 *IC* 值方法有两种：相关系数 (Pearson Correlation) 和秩相关系数 (Spearman Rank Correlation)。

由于 Pearson 相关系数计算时假设变量具有相等间隔以及服从正态分布，而这一假设往往与因子值和股票收益率的分布情况相左。因此我们将采用 Spearman 的方法计算因子暴露与下期收益率的秩相关性 *IC* 值。

Spearman 相关系数是衡量两个变量的依赖性的非参数指标。它利用单调方

程评价两个统计变量的相关性。被观测的两个变量的等级的差值

$$d_i = x_i - y_i$$

则

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{4.1}$$

Spearman 系数表明 X (独立变量) 和 Y (依赖变量) 的相关方向。如果当 X 增加时，Y 趋向于增加，Spearman 系数为正。如果当 X 增加时，Y 趋向于减少，Spearman 系数为负。系数为零表明 X 增减时 Y 没有任何趋向性。

当 X 和 Y 越来越接近完全的单调相关时，Spearman 系数会在绝对值上增加。当 X 和 Y 完全单调相关时，Spearman 系数的绝对值是 1。完全的单调递增关系意味着任何两对数据 (X_i, Y_i) 和 (X_j, Y_j) ，有 $X_i - Y_i$ 和 $X_j - Y_j$ 总是同号；完全的单调递减关系意味着任何两对数据 (X_i, Y_i) 和 (X_j, Y_j)，有 $X_i - Y_i$ 和 $X_j - Y_j$ 总是异号。

当数据大致分布并没有明显的离群点时，Pearson 系数和 Spearman 系数是相似的。

类似回归法的因子测试流程，我们在计算 IC 时同样考虑剔除了行业因素与市值因素。同样我们会得到一个 IC 值序列，类似的，我们将关注以下几个与 IC 值相关的指标来判断因子的有效性和预测能力：

- IC 值的均值
- IC 值的标准差
- IC 大于 0 的比例
- IC 绝对值大于 0.02 的比例
- IR (IR = IC 均值/IC 标准差)

由于单因子回归法所得到的因子收益值序列并不能直观的反应因子在各期的历史收益情况以及单调性，为了同时能够展示所检验因子的单调性，我们将通过分层打分回溯的方法作为补充。

在进行分层回溯法时，我们在各期期末将全市场 A 股按照因子值得大小分成 5 等分，在分组时同样做行业中性处理，即在中信一级行业内做 5 等分组。同时为了使回溯结果具有可比性，我们在回溯测试每组的历史收益情况时采取了市值加权的方法。

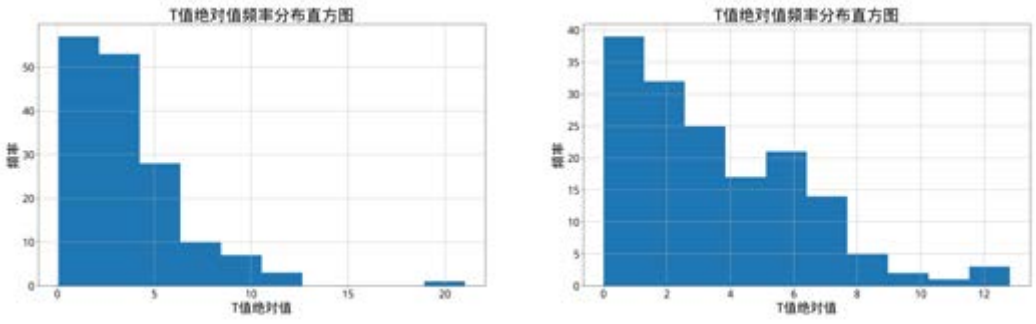
第三节 单因子测试举例

我们以 F38 和 F129 因子为例来展示本节提到的单因子测试流程。因子 F38 是 PB 的倒数，F129 是过去一个月的收盘价的 90% 分位数。

4.3.1 OLS 回归测试

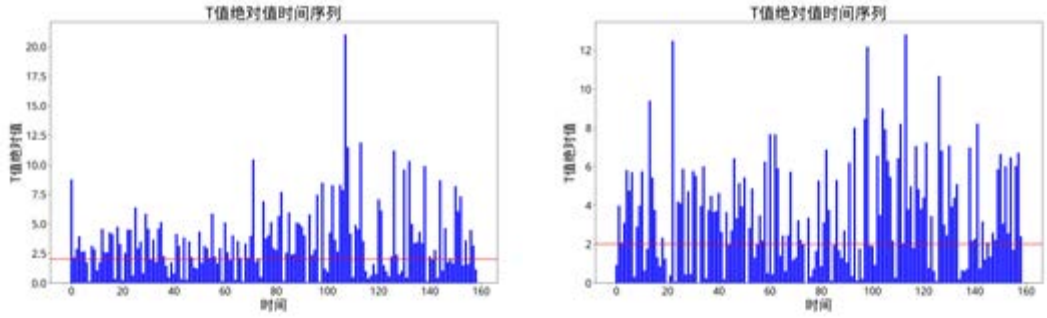
指标名称	F38	F129
$\|T\|$ 均值	3.54	3.56
$\|T\| \geq 2$ 比例	68.6%	64.8%
$T > 0$ 比例	61.6%	25.2%
IC 均值	5.29%	-7.46%
$\|IC\|$ 均值	12.0%	11.1%
IC 标准差	14.1%	11.9%
$IC > 0$ 比例	66.0%	33.3%
$\|IC\| > 0.02$ 比例	88.7%	87.4%
IR	37.21%	-62.7%

由上表可知，F38 和 F129 的因子收益序列显著性均大于 0，其中 F38 因子收益情况良好，同时 IC 值高达 5.23%，IR 值达到 5.23%。与之相反，F129 是一个负面因子，IC 值是 -7.46%，IR 值达到 -62.7%，也就代表着过去一期价格的 90% 分位数越高的股票，下期的收益越低。



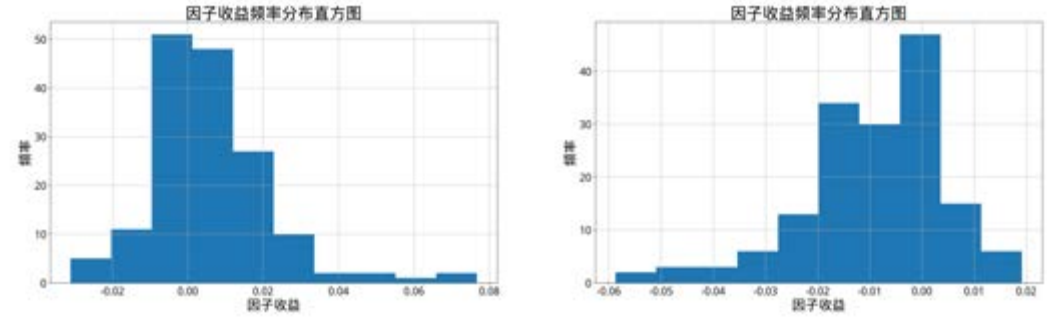
(a) F38 (b) F129

图 4.1 F38 & F129-T 值绝对值频率分布直方图



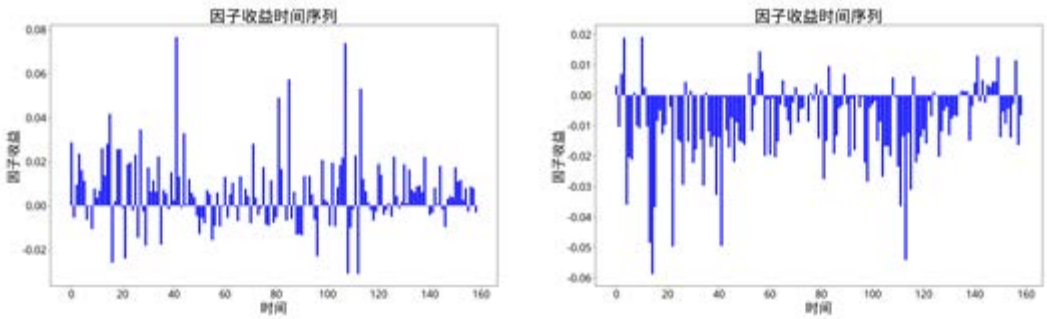
(a) F38 (b) F129

图 4.2 F38 & F129-*T* 值绝对值时间序列



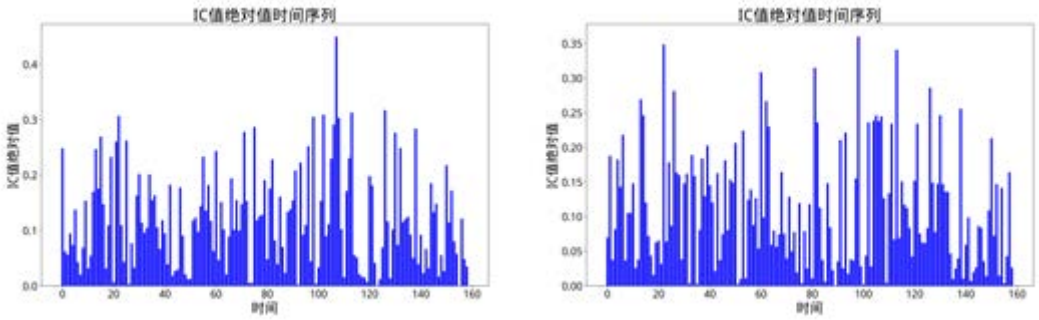
(a) F38 (b) F129

图 4.3 F38 & F129-因子收益频率分布直方图



(a) F38 (b) F129

图 4.4 F38 & F129-因子收益时间序列



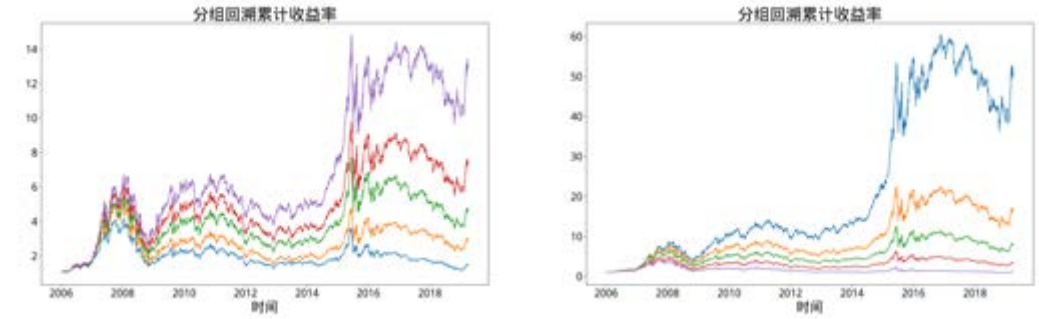
(a) F38 (b) F129

图 4.5 F38 & F129-*IC* 值绝对值时间序列

然而，通过回归测试我们可以得到的仅是因子在各期的因子收益和因子预测能力的历史表现和变化情况，所以接下来我们将会通过分层回溯的方法来检验因子的单调性。

4.3.2 分层法回溯测试

如下图，紫色、红色、绿色、黄色与蓝色分别对应着因子暴露由高至低的五个投资组合的累计收益曲线。由此可见，F38 和 F129 均具有较好的单调性，在 2006-2009 年期间 F38 的单调性较差，而 F129 始终保持了很好的单调性。容易发现，F38 的因子暴露越高的股票，累计收益率越高，而 F129 正相反，这与回归测试得到的结果也是吻合的。



(a) F38 (b) F129

图 4.6 F38 & F129-分组回溯累计收益率曲线

第四节 因子筛选

完成单因子测试后，我们根据有效性检验的结果对因子进行筛选，筛选后的因子及其各指标如下表：

在下一章中，我们将根据因子收益和每个股票的因子暴露计算出个股的预期收益率，并利用预测结果构建投资组合。

因子	T_{mean}	$\|T\| \geq 2$	$T > 0$	IC_{mean}	$\|IC\|_{mean}$	IC_{std}	$IC > 0$	$\|IC\| > 0.02$	IR
F9	3.07	56.60%	37.11%	0.94%	10.00%	13.07%	42.14%	82.39%	-7.18%
F10	2.99	57.23%	40.25%	-0.87%	9.01%	11.51%	43.40%	84.28%	-7.52%
F11	2.91	55.35%	39.62%	-0.75%	9.18%	11.68%	44.03%	88.05%	-6.40%
F35	1.62	32.70%	42.77%	-1.13%	5.43%	6.84%	42.14%	71.70%	-16.52%
F37	2.36	48.42%	51.57%	-0.96%	7.67%	9.47%	47.17%	81.13%	-16.16%
F38	3.54	68.55%	61.64%	5.29%	12.04%	14.05%	66.04%	88.68	37.21%
F39	1.46	24.52%	51.57%	1.55%	6.91%	8.58%	57.86%	84.91%	18.13%
F44	2.73	49.06%	63.52%	3.67%	9.05%	10.63%	61.01%	89.31%	34.54%
F47	3.28	64.15%	61.01%	1.88%	8.62%	10.41%	58.50%	82.39%	18.08%
F48	2.10	44.03%	59.12%	2.30%	7.00%	8.40%	61.01%	80.50%	27.32%
F50	4.76	67.92%	35.22%	-4.87%	15.72%	18.54%	35.85%	94.34%	-26.26%
F52	2.22	45.91%	43.40%	-0.49%	7.46%	9.84%	45.28%	74.21%	-4.99%
F54	1.71	34.56%	67.30%	2.25%	4.76%	5.66%	69.18%	72.96%	39.72%
F60	1.80	35.22%	53.46%	1.28%	6.49%	7.80%	56.60%	80.50%	15.98%
F63	1.91	44.65%	65.41%	1.75%	6.17%	7.58%	55.35%	77.99%	23.11%
F68	1.55	27.67%	54.72%	1.51%	5.78%	7.12%	56.60%	75.47%	21.23%
F70	2.11	41.51%	56.60%	1.39%	6.71%	8.37%	55.97%	82.39%	16.55%
F71	1.56	29.56%	74.84%	2.74%	4.68%	5.26%	71.07%	72.33%	52.16%
F73	1.72%	34.59%	73.58%	2.80%	5.10%	5.72%	67.92%	74.21%	48.88%
F106	3.76	64.78%	25.79%	-5.45%	10.37%	12.27%	28.30%	84.28%	-44.41%
F107	4.07	64.78%	37.11%	-2.73%	14.43%	17.56%	40.25%	90.57%	-15.57%
F114	3.71	64.15%	35.22%	-5.05%	12.06%	14.10%	38.99%	90.56%	-35.83%
F115	3.12	59.12%	42.14%	-2.29%	10.17%	12.39%	42.77%	88.68%	-18.47%
F116	3.55	59.75%	33.33%	-6.31%	11.78%	13.25%	32.70%	89.94%	-47.61%

因子	T_{mean}	$\|T\| \geq 2$	$T > 0$	IC_{mean}	$\|IC\|_{mean}$	IC_{std}	$IC > 0$	$\|IC\| > 0.02$	IR
F117	2.93	55.35%	40.88%	-2.74%	9.73%	12.08%	42.14%	87.42%	-22.74%
F118	4.07	64.78%	37.11%	-2.73%	14.44%	17.56%	40.25%	90.57%	-15.57%
F120	3.70	66.04%	32.70%	-3.99%	13.35%	15.65%	36.48%	94.34%	-25.51%
F122	4.48	69.81%	26.41%	-5.84%	12.27%	14.40%	29.56%	88.05%	-40.53%
F123	3.70	66.04%	22.01%	-6.67%	10.70%	11.51%	24.53%	88.68%	-57.87%
F124	5.17	73.58%	40.25%	-2.41%	15.23%	18.43%	40.225%	93.71%	-13.10%
F125	4.74	72.33%	38.99%	-2.98%	14.16%	1736%	37.74%	89.94%	-17.29%
F126	4.40	66.67%	36.48%	-4.53%	11.24%	13.96%	35.85%	89.31%	-32.48%
F127	4.40	66.04%	34.59%	-4.79%	11.37%	14.24%	38.36%	84.58%	-33.67%
F128	3.53	66.67%	24.53%	-7.28%	10.60%	11.31%	33.96%	84.91%	-64.38%
F129	3.56	64.78%	25.16%	-7.46%	11.12%	11.89%	33.33%	87.42%	-62.70%
F130	4.26	70.44%	33.96%	-6.45%	14.09%	16.29%	33.33%	89.31%	-39.57%
F131	3.72	68.55%	33.96%	-6.22%	12.40%	13.86%	34.59%	91.19%	-44.87%
F132	4.24	69.18%	32.08%	-5.59%	13.70%	15.90%	35.85%	90.57%	-35.16%
F133	4.06	68.55%	30.82%	-5.85%	13.24%	14.98%	37.11%	92.45%	-39.01%
F134	4.28	74.21%	35.22%	-5.13%	14.08%	16.18%	37.74%	95.60%	-31.70%
F135	3.19	61.63%	29.56%	-7.20%	12.56%	14.61%	31.45%	87.42%	-49.28%
F136	4.22	69.18%	30.19%	-7.40%	13.79%	15.47%	31.45%	89.94%	-47.84%
F137	4.12	69.81%	32.70%	-5.97%	13.22%	15.61%	33.96%	90.57%	-38.20%
F138	5.56	75.47%	42.14%	-1.56%	15.14%	19.04%	45.28%	86.79%	-8.19%
F139	5.15	75.47%	39.62%	-2.20%	14.69%	18.45%	40.25%	89.94%	-11.92%
F140	4.32	73.58%	30.82%	-6.05%	12.10%	13.28%	29.56%	93.71%	-45.59%

第五章　线性回归模型

本章我们将利用最小二乘 OLS 和套索 LASSO 两种算法，分别对训练集中股票的因子暴露与其下一期的收益进行回归，得到因子收益序列，再对测试集中的股票收益进行预测，根据预测结果构建多空投资组合。此外，为了使预测结果更加稳定，我们还尝试了利用 Bootstrap 方法，每次从训练集中抽取一定数据进行回归预测，将多次预测结果取平均值作为最终的预测结果。

- 样本范围：中国市场全体 A 股；
- 训练集范围：2006-01-01 至 2016-12-01 共 132 个月，约 23 万条数据；
- 测试集范围：2017-01-01 至 2019-03-01 共 27 个月，约 7 万条数据；

第一节　最小二乘估计 OLS

5.1.1　OLS 回归原理

在统计学中，线性最小二乘法 OLS 是最常用的一种用于在线性回归模型中估计未知参数的方法。OLS 通过最小化所观察的因变量 (所预测的变量的值)，在给定数据集中的残差 ε_i 的平方和：

$$SSR = \sum_{i=1}^{N} \varepsilon_i^2 = \sum_{i=1}^{N} (r_i - \sum_{j=1}^{M} \beta_{ij} \cdot f_j)^2 \tag{5.1}$$

来选择一组由线性函数预测的参数即因子收益 f_j。

在几何上，这可以看作是到平行于因变量的轴的距离平方的总和，该集合中的每个数据点与回归线上的相应点之间的残差越小，模型的拟合度越高。当误差是同方差和连续不相关的时候，OLS 在各种线性无偏估计中是最优的。在这些条件下，当误差具有有限的方差时，OLS 方法提供了最小方差均值的无偏估计。在误差正态分布的假设下，OLS 是最大似然估计量。

考虑残差平方和 SSR (5.1)，即罚函数

$$S(\boldsymbol{f}) = \sum_{i=1}^{N} (\boldsymbol{r_i} - \boldsymbol{\beta}_i^T \boldsymbol{f})^2 = (\boldsymbol{r} - \boldsymbol{X}\boldsymbol{f})^T (\boldsymbol{r} - \boldsymbol{X}\boldsymbol{f}) \tag{5.2}$$

其中 $\boldsymbol{X}$ 为因子暴露矩阵

$$\boldsymbol{X}=\begin{bmatrix}\beta_{11} & \beta_{12} & \cdots & \beta_{1M}\\ \vdots & \vdots & \ddots & \vdots\\ \beta_{N1} & \beta_{N2} & \cdots & \beta_{NM}\end{bmatrix}$$

所求的 $\hat{\boldsymbol{f}}$ 即为使罚函数最小时的 $\boldsymbol{f}$，即

$$\hat{\boldsymbol{f}}=\underset{\boldsymbol{f}\in\boldsymbol{R}^M}{\arg\max}\, S(\boldsymbol{f}) \tag{5.3}$$

于是有

$$\frac{\partial S}{\partial \boldsymbol{f}}(\hat{\boldsymbol{f}})=\frac{\partial}{\partial \boldsymbol{f}}(\boldsymbol{r}'\boldsymbol{r}-\boldsymbol{f}'\boldsymbol{X}'\boldsymbol{r}-\boldsymbol{r}'\boldsymbol{X}\boldsymbol{f}+\boldsymbol{f}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{f})\big|_{\boldsymbol{f}=\hat{\boldsymbol{f}}}=0$$

$$\Rightarrow \hat{\boldsymbol{f}}=(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{r} \tag{5.4}$$

即为所求的回归系数。

第二节 LASSO

LASSO 回归在是在最小二乘法估计的基础上对回归系数进行压缩，使部分绝对值较小的系数被直接压缩为 0。LASSO 方法的表达式可以写成：

$$\hat{\boldsymbol{f}}=\underset{\boldsymbol{f}\in\boldsymbol{R}^M}{\arg\max}\,||\boldsymbol{r}-\boldsymbol{X}\boldsymbol{f}||^2$$

$$s.t.\sum_{i=1}^{M}|f_i|\le t$$

$$t_0=\sum_{i=1}^{M}|\hat{f}_i|$$

当 $t\le t_0$ 时部分回归参数就会被压缩到 0。另一种常见的表达方式为：

$$\hat{\boldsymbol{f}}=\underset{\boldsymbol{f}\in\boldsymbol{R}^M}{\arg\max}\{||\boldsymbol{r}-\boldsymbol{X}\boldsymbol{f}||^2+\alpha\sum_{i=1}^{M}|f_i|\}$$

LASSO 的解为：

$$\hat{\boldsymbol{f}}_{\boldsymbol{i}}=\begin{cases}sgn(\hat{\boldsymbol{f}}_{\boldsymbol{i}})(|\hat{\boldsymbol{f}}_{\boldsymbol{i}}|-\gamma) & ,|\hat{\boldsymbol{f}}_{\boldsymbol{i}}|>\gamma\\ 0 & ,|\hat{\boldsymbol{f}}_{\boldsymbol{i}}|\le\gamma\end{cases}$$

其中 γ 由条件 $\sum_{i=1}^{M}|f_i| = t$ 决定。

也就是说，对于最小二乘法估计的绝对值大于 γ 的参数，LASSO 的解将这些值向原点压缩了 γ 个单位，而对于参数估计绝对值小于等于 γ 的系数，LASSO 的解直接将其压缩为 0。

5.2.1 LASSO 回归优点

我们构建的多因子模型中，由于因子数量很多，如果只简单使用 OLS 方法进行回归容易发生过拟合，同时因子之间可能存在复共线性导致回归结果方差过大。因此，为了使模型具有更好的解释能力，同时降低了无效因子的干扰，我们使用 LASSO 方法通过将一些因子的系数降为 0，筛选出较好的因子使投资组合模型保持稳定的结果，近而从多因子中提取出有效的资产组合。即在多因子模型上对因子权重施加了一个惩罚函数进行约束，从而使权重变得稀疏并且具有了更好的稳定性。

第三节 Bootstrap 方法

Bootstrap 是利用少量实验数据的样本信息及计算机仿真，去模拟未知分布并获得所需感兴趣的未知分布的某一特征的方法，是数据分析中常用的非参数统计方法，由 Efron 在 20 世纪 70 年代后期建立的。这种方法不需要对对未知分布做任何假设，只需利用计算机对已知数据进行再抽样来模拟未知分布，近而估计所求未知变量，大大节约了成本。

我们将尝试利用 Bootstrap 方法对线性回归模型进行优化，步骤如下：

1. 自训练集 $x_1, x_2, \ldots, x_N$ 按放回抽样的方法，抽的容量为 n 的 Bootstrap 样本；
2. 相继地、独立地抽得 B 个 Bootstrap 样本，分别利用线性回归，求得对应的因子收益 f_i 的估计为 $\hat{f_{iB}}$，$i = 1, 2, \cdots, B$；
3. 计算 $\hat{f_i} = \frac{\sum_{j=1}^{B} f_{ij}}{B}$
4. 这就是 $\boldsymbol{f}$ 的 Bootstrap 估计。

我们将在下一节中对比 Bootstrap 方法与普通线性回归得到的结果。

第四节 OLS 与 LASSO 的对比

5.4.1 OLS 应用

我们首先用全部训练集进行回归，得到回归参数，即因子收益：

```
In [1]: model = LassoCV(alphas = np.arange(-0.1, 0.1 , 0.0001))
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print(model.coef_)
        print(model)
        print(model.alpha_)

Out [1]:[-3.51014659e-05  1.82696611e-03  9.29291249e-04 -2.90408529e-03
         -2.74646718e-05  4.16366358e-03  3.26895515e-04  4.47577364e-05
          4.74909769e-03 -3.21869648e-03 -2.29457926e-03 -1.46291839e-03
         -2.90087056e-03  7.22242702e-04  6.72217006e-03 -1.25924078e-03
          1.07952028e-03  1.11284686e-03  3.14294805e-03  2.28034175e+01
          9.28580953e-04  6.09875392e-03 -4.87004691e-03 -9.59504720e-04
         -5.81984418e-03 -2.28039130e+01  1.69447744e-03 -6.66640992e-03
         -6.98341747e-03  1.00153259e-03 -5.18207138e-03 -4.34446954e-03
         -5.19747948e-04 -2.43265467e-02  5.92038621e-03  3.05878159e-03
          3.03491507e-03  4.77113124e-03  1.71980354e-03 -1.48989944e-03
         -1.89175551e-03 -3.59573592e-03 -1.24938729e-03  2.42755581e-03
         -2.01773756e-04  1.08718901e-03]

        LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

图 5.1 OLS code

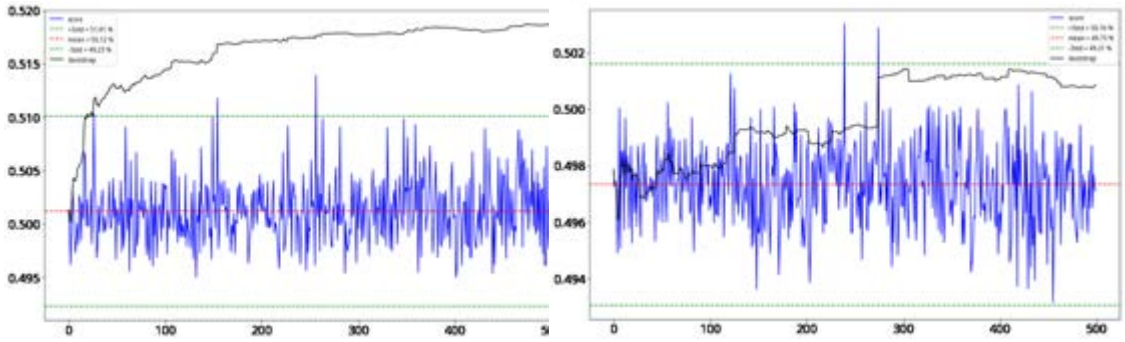
再利用 Bootstrap 算法迭代 500 次，模型拟合优度随次数的增多有明显的提升。

5.4.2 LASSO 的应用

我们用训练集中的数据进行回归，利用交叉验证 Cross Validation 的方法计算得到最优的 α 值，即 LASSO 回归模型中的惩罚参数：

根据计算结果，$\alpha = 0.004$ 为模型最优解，有 24 个因子收益为 0，保留了 24 个因子。

再利用 Bootstrap 算法迭代 500 次，模型拟合优度随次数的增多有一定的提升，但提升不大。



(a) OLS (b) LASSO

图 5.2 OLS & LASSO-拟合优度

5.4.3 结果分析及对比

我们分别用以上四组预测值进行选股策略，选择每一期预测值最高的 20 只股票买进，作为多头组合 Portfolio 1；预测值最低的 20 只股票卖空作为空头组合 Portfolio 2；以及买入 Portfolio 1，卖出 Portfolio 2，构造多空组合，其收益曲线与基准合约（沪深 300）的收益曲线如下图：

四种策略的空头组合都有明显的下跌趋势，获得稳定的负超额收益，而只有普通 LASSO 和 Bootstrap 后 OLS 的空头组合跑赢了指数，但综合来看，多头组合与空头组合明显分离，多空组合均获得了比较稳定的超额收益，这说明线性回归是可以在一定范围内预测股价变化的大体趋势的。

指标	OLS Normal 多空	OLS Bootstrap 多空	LASSO Normal 多空	LASSO Bootstrap 多空
年化收益率	12.0%	22.7%	49.9%	9.58%
基准收益率	6.76%	6.76%	6.76%	6.76%
超额收益	5.19%	15.9%	43.2%	2.82%
α	0.04	0.07	0.15	0.03
β	0.09	0.20	0.25	0.07
最大回撤	12.0%	19.8%	15.1%	19.4%
平均波动率	9.57%	9.56%	1.33%	9.82%
Sharpe	0.44	0.85	1.33	0.35

再结合下面的图片可以发现，对于 OLS 算法，Bootstrap 后的结果相较普

```
In [2]: model = LinearRegression()
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print(model.coef_)
        print(model)

Out [2]:[ 0.00000000e+00  1.51201252e-03  2.19052511e-04 -2.61456168e-03
         -0.00000000e+00  3.50983214e-03  3.71829328e-05  0.00000000e+00
          2.02933301e-03 -4.42552141e-04 -2.29214113e-03 -3.67121068e-04
         -0.00000000e+00  0.00000000e+00  3.72505539e-03 -6.46884252e-05
          7.83728533e-04  8.31703682e-04  2.57377381e-03  2.14403242e-04
         -0.00000000e+00  0.00000000e+00 -6.38435320e-04  0.00000000e+00
         -4.91047097e-03  1.75994312e-05  0.00000000e+00 -3.47040511e-03
         -5.53491956e-03  1.56675810e-03 -2.39448258e-03 -4.34502122e-03
         -5.73955523e-04 -1.36277038e-02 -0.00000000e+00  2.42202344e-03
          0.00000000e+00  3.91465966e-03  0.00000000e+00 -1.32398449e-03
         -1.23653045e-03 -3.66836029e-03 -1.29844360e-03  2.04552176e-03
          0.00000000e+00  9.06507635e-06 -0.00000000e+00  0.00000000e+00
         -0.00000000e+00  0.00000000e+00  0.00000000e+00 -0.00000000e+00
          6.82991961e-04 -0.00000000e+00 -0.00000000e+00  0.00000000e+00
         -0.00000000e+00  0.00000000e+00]

        LassoCV(alphas=array([0.0001, 0.0002, ..., 0.0998, 0.0999]), copy_X=True,
            cv=None, eps=0.001, fit_intercept=True, max_iter=1000, n_alphas=100,
            n_jobs=1, normalize=False, positive=False, precompute='auto',
            random_state=None, selection='cyclic', tol=0.0001, verbose=False)

        0.0004
```

图 5.3 OLS code

通的回归有明显的提升，但 Bootstrap 后 LASSO 的投资组合表现却远不及普通 LASSO，这与前两节的拟合优度结果相符。

最后，再对两种回归算法最好的的多空组合对比，可以发现 LASSO 的预测结果要优于 OLS，这是由于其能消除一些相关性高的因子，更准确地对模型做出解释。但其 Bootstrap 后结果较差，这是因为每次的样本选择很大程度上影响了最终的因子选择，LASSO 算法在不同的训练集中选择消去的复共线性因子有可能不同，平均后的预测结果使其丧失了消除共线性优点，所以预测不准确也是可以理解的。

(a) OLS (b) Bootstrap 后的 OLS

图 5.4 Bootstrap 前后 OLS 的投资组合表现

(a) LASSO (b) Bootstrap 后的 LASSO

图 5.5 Bootstrap 前后 LASSO 的投资组合表现

(a) OLS (b) LASSO

图 5.6 OLS & LASSO Bootstrap 前后多空组合对比

图 5.7 OLS 与 LASSO 的多空组合对比

第六章　多感知神经网络

第一节　模型介绍

多层感知机（MLP，Multilayer Perceptron）也叫人工神经网络（ANN，Artificial Neural Network）除了输入输出层，它中间可以有多个隐层，最简单的 MLP 只含一个隐层，即三层的结构。

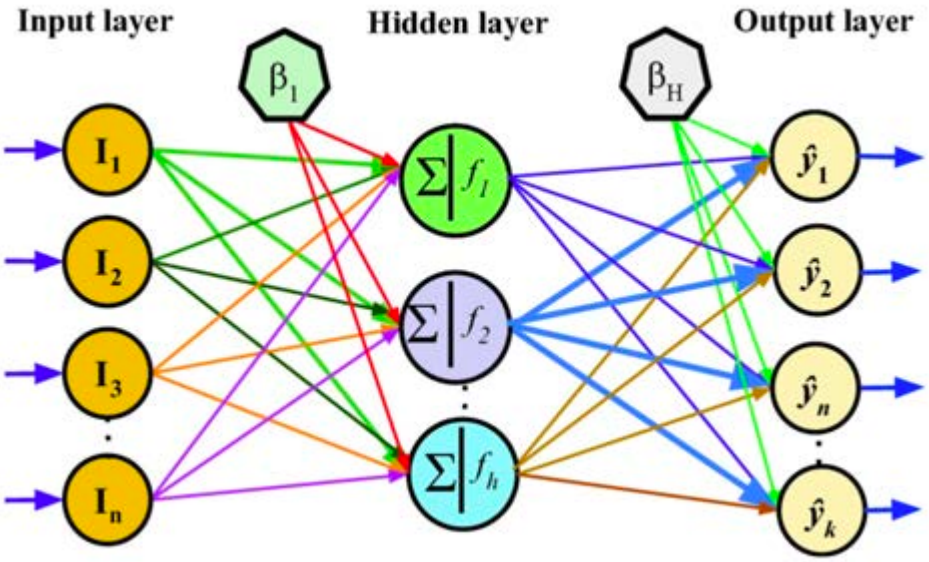
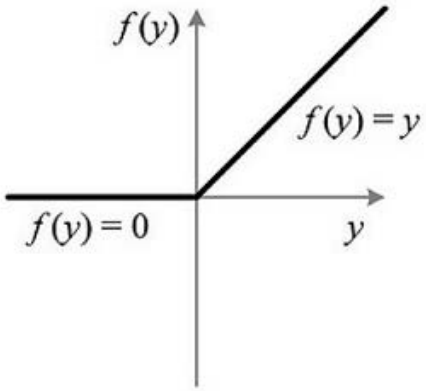


图 6.1　三层结构的 MLP

6.1.1 ReLU：用在隐藏层的激活函数



$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases} \tag{6.1}$$

优点：使用 ReLU 得到的 SGD 的收敛速度会比 sigmoid/tanh 快很多。相比于 sigmoid/tanh，ReLU 只需要一个阈值就可以得到激活值，而不用去算一大堆复杂的运算。虽然 ReLU 函数大于零的部分和小于零的部分分别都是线性函数，但是整体并不是线性函数，所以仍然可以做为激活函数，ReLU 函数其实是分段线性，把所有的负值都变为 0，而正值不变，这种操作被成为单侧抑制。

在训练数据的时候，由于对于不同的任务，可能某些神经元的输出影响就比较大，有些则小，甚至有些则无，类似于人的大脑，左脑和右脑分别管理逻辑能力与想象力，当使用右脑的时候，就抑制左脑，当使用左脑的时候，抑制右脑，RELU 函数正好可以实现小于 0 的数直接受到抑制，这就使得神经网络的激活更接近于生物学上的处理过程，给神经网络增加了生命。

6.1.2 SoftMax 函数：用在输出层的激活函数

归一化指数函数，目的是将多分类的结果以概率的形式展现出来。

6.1.3 Dropout：训练深度神经网络的一种供选择的方式

在每个训练批次中，通过忽略一半的特征检测器（让一半的隐层节点值为 0），可以明显地减少过拟合现象。这种方式可以减少特征检测器（隐层节点）间的相互作用，检测器相互作用是指某些检测器依赖其他检测器才能发挥作用。Dropout 的实现是在前向传导的时候，让某个神经元的激活值以一定的概率 p，

使其停止工作。之前神经网络的计算公式是：

$$z_i^{l+1} = w_i^{l+1} y^l + b_i^{l+1}$$

$$y_i^{l+1} = f(z_i^{l+1})$$

采用 dropout 后计算公式就变成了：

$$r_j^l \sim Bernoulli(p)$$

$$\tilde{\mathbf{y}}^l = \mathbf{r}^l * \mathbf{y}^l$$

$$z_i^{l+1} = \mathbf{w}_i^{l+1} \tilde{\mathbf{y}}^l + b_i^{l+1}$$

$$y_i^{l+1} = f(z_i^{l+1})$$

上面公式中 Bernoulli 函数，是为了以概率 p，随机生成一个 0、1 的向量。

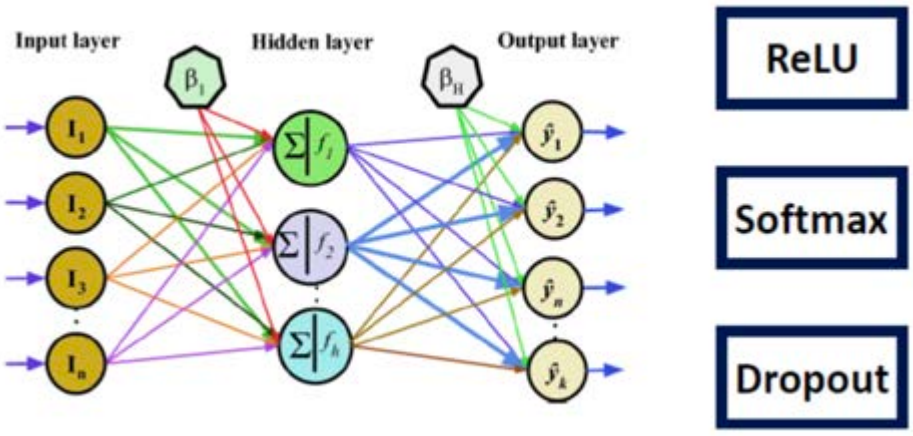


图 6.2

第二节　训练结果和模型对比

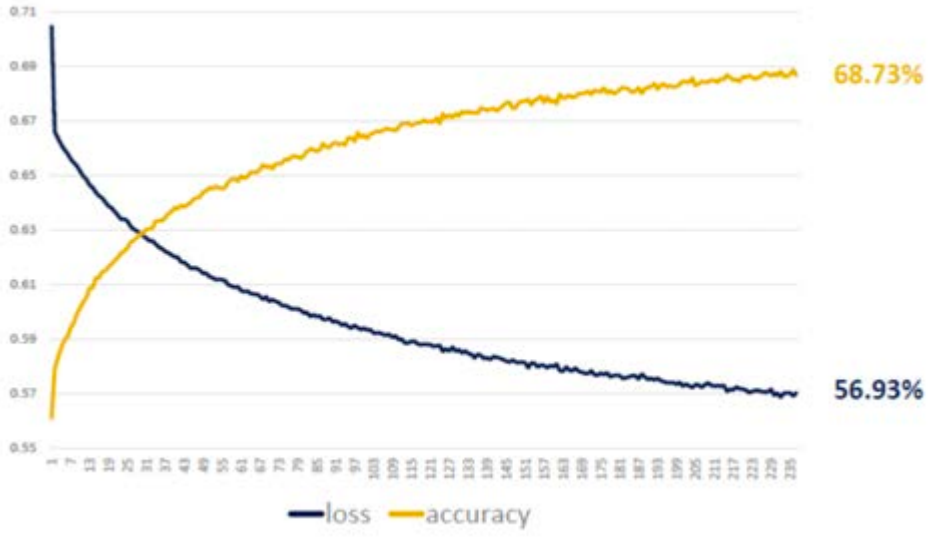


图 6.3 通过 MLP 模型训练，预测准确率达到了 68.73%

Portfolio	
Accumulated Return	42.8%
Abnormal Return	52.2%
Alpha	0.412
Beta	0.063
Sharp Ratio	3.262
Maximum Drawdown	4.48%
Average Volatility	0.670%

图 6.4 回测结果

模型在回测期获得总收益为 42.8%，夏普比率达到 3.262，最大回撤为 4.48%。

Portfolio	OLS	LASSO	MLP
Accumulated Return	45.9%	55.0%	42.8%
Abnormal Return	55.3%	64.4%	52.2%
Alpha	0.465	0.541	0.412
Beta	0.223	0.095	0.063
Sharp Ratio	2.278	2.077	3.262
Maximum Drawdown	7.40%	8.15%	4.48%
Average Volatility	1.134%	1.468%	0.670%

图 6.5 OLS、LASSO、MLP 结果对比

MLP 相对较 OLS 和 LASSO 较优，得到的策略更加稳健，通过夏普比率、最大降深、平均波动率等可以看出。可见神经网络在分析预测复杂、非线性显著的股票数据中具有一定的优势，相较于中国市场这样发展不成熟的弱有效市场来说，神经网络具有更好的预测能力。

第三节 总结

我国量化投资传入的时间并不久，在中国的发展的空间将非常巨大。从目前中国的证券市场现状来看，随着 A 股市场上股票数量的不断增加，传统的定性分析方法的分析成本正在不断上升，仅仅依靠传统的基本面和技术面研究获得超额收益的难度加大。在这种情况下，市场上对多元化的投资理念的需求变得异常迫切，量化投资作为一种新兴的投资方式，能够很好的弥补传统分析的不足，最大程度上避免市场的非理性投资。量化投资方法，有效的避免了投资者的主观依赖以及对技术指标的过度依赖，以科学的研究方法和实证分析来预测股价的未来走势。制定合理的量化投资策略，需要我们运用数学能力以及强大的计算机手段进行辅助，量化能力的提升将推动量化投资的大力发展。