

# 量化研究与策略

iLampard

2017 年 1 月 11 日

## 目录

<b>1</b>	<b>量化选股基础</b>	<b>2</b>
1.1	因子的预处理 . . . . .	2
1.1.1	中性化 . . . . .	2
1.1.2	去极值化 . . . . .	3
1.1.3	标准化 . . . . .	3
1.2	主成分分析 . . . . .	3
1.3	策略表现的指标计算 . . . . .	4
1.3.1	对冲组合的净值计算方法 . . . . .	4
<b>2</b>	<b>量化选股策略</b>	<b>5</b>
2.1	动态情景多因子策略 . . . . .	5
2.1.1	概述 . . . . .	6
2.1.2	算法 . . . . .	6
<b>3</b>	<b>附录</b>	<b>8</b>
3.1	PCA推导 . . . . .	8

# 1 量化选股基础

## 1.1 因子的预处理

从各个数据源得到的因子数据称为原始因子。在构建Alpha策略前，要对截面原始因子进行一定的处理。

假设有 $k$ 个行业（一般使用申万一级行业），每个行业均有 $m$ 个股票(仅为计算方便如此假设)。用 $f_{i,t}^j$ 表示 $t$ 时刻第 $i$ 个行业的 $j$ 支股票的因子。因为是针对截面数据做处理( $t$ 作为一个固定值)，下面的表示将隐去 $t$ 角标，因子就以 $f_i^j$ 表示。

### 1.1.1 中性化

一般来说，因子需要对行业进行中性化处理，以去除行业对因子的影响。方法是将原始因子对行业虚拟变量进行回归，取得回归的残差作为因子值。

根据假设，将这些股票的因子合并成一个 $(km) \times 1$ 的向量， $f = (f_1^1, \dots, f_i^j \dots f_k^m)^T, i = 1, 2, \dots, k, j = 1, 2, \dots, m$ ，对应的 $(k \times 1)$  回归系数向量 $\beta = (\beta_1, \dots, \beta_j \dots \beta_k)^T$ 。

回归变量可以表示为 $(km) \times k$ 的矩阵 $X = (u_1^1, \dots, u_i^j \dots u_k^m)^T$ ，其中 $u_i^j = (0, \dots, 1, 0, \dots, 0)$  为第 $i$ 个元素为1总长度为 $k$ 的单位行向量。

$$\begin{pmatrix} f_1^1 \\ f_1^2 \\ \dots \\ f_i^j \\ \dots \\ f_k^m \end{pmatrix}_{(km) \times 1} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & & & \\ 0 & 0 & 1_{(i-1)m+j,i} & 0 \\ \dots & & & \\ 0 & \dots & 0 & 1 \end{pmatrix}_{(km) \times k} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_i \\ \dots \\ \beta_k \end{pmatrix}_{k \times 1} + \begin{pmatrix} \varepsilon_1^1 \\ \varepsilon_1^2 \\ \dots \\ \varepsilon_i^j \\ \dots \\ \varepsilon_k^m \end{pmatrix}_{(km) \times 1} \quad (1)$$

公式(1)中的残差向量 $(\varepsilon_1^1, \dots, \varepsilon_k^m)$ 就是行业中性化后的因子

[

公式(1)的含义是，行业中性化处理等价于每个股票的因子减去行业内股票该类因子的平均值。因为对指示函数 $\mathbf{1}_{industry}$ 回归，那么回归系数 $\beta$ 等于所有因变量的均值。

]

还有一种中性化的方法是同时对行业以及风格(如市值对数)进行中性化，用 $(km) \times 1$  的向量 $(\log M_1^1, \log M_1^2 \dots \log M_k^m)^T$ 表示所有股票在该截面时刻的市值对数，原始因子对行业虚拟矩阵和市值对数向量同时做回归可得

$$\begin{pmatrix} f_1^1 \\ \dots \\ f_i^j \\ \dots \\ f_k^m \end{pmatrix}_{(km) \times 1} = \begin{pmatrix} 1 & \dots & 0 \\ \dots & & \\ 0 & 1_{(i-1)m+j,i} & 0 \\ \dots & & \\ 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \dots \\ \beta_i \\ \dots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \log M_1^1 & \dots & 0 \\ \dots & \dots & \\ \dots & \log M_i^j & \dots \\ \dots & \dots & \\ 0 & \dots & \log M_k^m \end{pmatrix} \begin{pmatrix} \gamma_1^1 \\ \dots \\ \gamma_i^j \\ \dots \\ \gamma_k^m \end{pmatrix} + \begin{pmatrix} \varepsilon_1^1 \\ \dots \\ \varepsilon_i^j \\ \dots \\ \varepsilon_k^m \end{pmatrix} \quad (2)$$

公式(2)中的残差向量 $(\varepsilon_1^1, \dots, \varepsilon_k^m)$ 就是行业中性化以及市值中性化后的因子。在实际处理中通常要把公式(2)转换成公式(3)表示的形式：

$$\begin{pmatrix} f_1^1 \\ f_1^2 \\ \dots \\ f_i^j \\ \dots \\ f_k^m \end{pmatrix}_{(km) \times 1} = (A, B)_{(km) \times k(m+1)} \begin{pmatrix} \beta_1 \\ \dots \\ \beta_k \\ \gamma_1^1 \\ \dots \\ \gamma_k^m \end{pmatrix}_{k(m+1) \times 1} + \begin{pmatrix} \varepsilon_1^1 \\ \varepsilon_1^2 \\ \dots \\ \varepsilon_i^j \\ \dots \\ \varepsilon_k^m \end{pmatrix}_{(km) \times 1} \quad (3)$$

其中,

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & & & \\ 0 & 0 & 1_{(i-1)m+j,i} & 0 \\ \dots & & & \\ 0 & \dots & 0 & 1 \end{pmatrix}_{(km) \times k}$$

$$B = \begin{pmatrix} \log M_1^1 & 0 & \dots & 0 \\ 0 & \log M_1^2 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \log M_i^{(i-1)m+j} & 0 \\ \dots & & & \\ 0 & \dots & 0 & \log M_k^m \end{pmatrix}_{(km) \times (km)}$$

### 1.1.2 去极值化

因子序列中要去除某些极值（异常值）的影响，一般做法是将原始因子值调整到3倍绝对偏离中位数（或者是均值）的范围内。

### 1.1.3 标准化

对于原始因子序列 $\{f_i, i = 1, \dots, N\}$ , 用 $\mu, \sigma$ 表示这一序列的均值和标准差, 则标准化后的因子序列为

$$\{f_i^s = \frac{f_i - \mu}{\sigma}, i = 1, \dots, N\}$$

## 1.2 主成分分析

[主成分分析原理详解, <http://blog.csdn.net/yansmile1/article/details/46954089>]

[主成分分析简介及其python实现, <http://blog.csdn.net/yansmile1/article/details/46954089>]

[Peter Harrington, 《机器学习实战》, 人民邮电出版社, 2013]

主成分分析 (Principal Component Analysis, PCA) 是一种常见的数据降维方法, 其目的是在信息量损失较小的前提下, 将高维的数据转换到低维, 从而减小计算量。实质就是找到一些投影方向, 使得数据在这些投影方向上包含的信息量最大, 而且这些投影方向是相互正交的。选择其中一部分包含最多信息量的投影方向作为新的数据空间, 同时忽略包含较小信息量的投影方向, 从而达到降维的目的。

样本的“信息量”可以理解为是样本在特征方向上投影的方差。方差越大，则样本在该特征上的差异就越大，因此该特征就越重要。参见《机器学习实战》上的图，在分类问题里，样本的方差越大，越容易将不同类别的样本区分开。

PCA的数学原理，就是对原始的空间中顺序地找一组相互正交的坐标轴，第一个轴是使得方差最大的，第二个轴是在与第一个轴正交的平面中使得方差最大的，第三个轴是在与第1、2个轴正交的平面中方差最大的，这样假设在N维空间中，可以找到N个这样的坐标轴，取前r个去近似这个空间，这样就从一个N维的空间压缩到r维的空间了，但是最终选择的r个坐标轴能够使得数据的损失最小。

假设存在一组原始数据 $\{z_1, z_2, \dots, z_n\}$ ，其中 $z_i$ 为 $p \times 1$ 维向量，即每个数据有p个特征。

- (i) 去除平均值，即中心化，得到新的数据 $\{x_1, x_2, \dots, x_n\}$ ， $x_i = z_i - \mu$ ， $\mu = \frac{1}{n} \sum_{i=1}^n z_i$
- (ii) 计算 $\{x_i\}$ 协方差矩阵 $\Sigma_{p \times p}$
- (iii) 计算协方差矩阵的特征向量 $\{\xi_j\}$ 和特征值 $\{\lambda_j\}$ ， $j = 1..p$
- (iv) 将特征值排序
- (v) 保留前q个特征向量，假设为 $\{\xi_j^*\}$ ， $j = 1..q$ ，对应的特征向量构成的矩阵为 $\Lambda = [\xi_1^* \dots \xi_q^*]_{p \times q}$
- (vi) 将数据转换到上述N个特征向量构建的新的空间中，得到新的数据集 $X^* = X\Lambda = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_i \\ \dots \\ x_n \end{pmatrix}_{n \times p} [\xi_1^* \dots \xi_q^*]_{p \times q}$

数学原理请见附录(3.1)。

## 1.3 策略表现的指标计算

### 1.3.1 对冲组合的净值计算方法

一个对冲组合的净值计算主要涉及到调仓的设定。此处调仓是指使得组合的股票多头和对应期货（以指数代替）的空头市值相同，即完全对冲。在下一个调仓日之前，组合的股票多头和期货空头数量不再调整，随着市场的波动，组合会出现多空头寸的暴露。

假设有 $t = 0, 1, 2, \dots, N$ 共 $(N + 1)$ 个交易日，调仓日为 $\{t_i^*, i = 1, 2 \dots k\}$ 共k个。调仓时间为调仓日的收盘前。

用 $\{r_t^s\}$ 和 $\{r_t^f\}$ 表示策略和对冲标的的日简单收益序列。用 $V_t, V_t^s, V_t^f$ 表示t时刻的组合价值，组合中股票账户市值，以及期货账户市值。

假设组合的收益为累计复利计算方式，另外假设组合中期货账户占用资金比例为p（用于保证金账户），那么组合中 $1 - p$ 的部分可投资于股票账户。对冲组合的净值NPV<sub>t</sub>变化可以演示为

- 在 $t = 0$ 时刻建立组合，该时刻也是第一个调仓日。此时 $V_0^s = V_0^f$ ， $NPV_0 = 1.0$
- 在 $t = 1$ 时刻，组合的价值变动为 $V_0^s(1 + r_1^s) - V_0^f(1 + r_1^f)$ ，此时

$$NPV_1 = NPV_0 \left( 1 + (1 - p)[(1 + r_1^s) - (1 + r_1^f)] \right)$$

[

*Proof.* 根据股票账户以及期货账户的资金比例假设以及  $V_0^s = V_0^f$  可知

$$\begin{aligned}
 \text{NPV}_1 &= \text{NPV}_0 \left( 1 + \frac{V_0^s(1+r_1^s) - V_0^f(1+r_1^f)}{V_0} \right) \\
 &= \text{NPV}_0 \left( 1 + \frac{V_0(1-p)(1+r_1^s) - V_0(1-p)(1+r_1^f)}{V_0} \right) \\
 &= \text{NPV}_0 \left( 1 + (1-p)[(1+r_1^s) - (1+r_1^f)] \right)
 \end{aligned}$$

□

]

– 在下一个调仓日之前(包括调仓日)的任意  $t = n$  时刻, 净值为

$$\text{NPV}_n = \text{NPV}_0 \left( 1 + (1-p) \left[ \prod_{i=1}^n (1+r_i^s) - \prod_{i=1}^n (1+r_i^f) \right] \right) \quad (4)$$

– 在第一个调仓日  $t_1^*$  之后,  $t = t_1^* + 1$  时刻, 净值为

$$\text{NPV}_{t_1^*+1} = \text{NPV}_{t_1^*} \left( 1 + (1-p)[(1+r_{t_1^*+1}^s) - (1+r_{t_1^*+1}^f)] \right) \quad (5)$$

从公式(4)和公式(5)可以得出以下结论

(i) 任意两个调仓日  $t_i^*, t_{i+1}^*$  之间, 对冲组合的净值为

$$\text{NPV}_{t_1^*+n} = \text{NPV}_{t_1^*} \left( 1 + (1-p) \left[ \prod_{i=1}^n (1+r_{t_1^*+i}^s) - \prod_{i=1}^n (1+r_{t_1^*+i}^f) \right] \right) \quad (6)$$

(ii) 期货保证金账户的资金占用降低了组合净值的变化。假设期货保证金占用为零, 此时对冲组合的净值为

$$\text{NPV}_{t_1^*+n} = \text{NPV}_{t_1^*} \left( 1 + \left[ \prod_{i=1}^n (1+r_{t_1^*+i}^s) - \prod_{i=1}^n (1+r_{t_1^*+i}^f) \right] \right) \quad (7)$$

(iii) 有一种特殊的情况为每日调仓, 保证每日结算时多空敞口为零, 此种情况下

$$\text{NPV}_{n+1} = \text{NPV}_n \left( 1 + [(1+r_{n+1}^s) - (1+r_{n+1}^f)] \right) \quad (8)$$

## 2 量化选股策略

### 2.1 动态情景多因子策略

[朱建涛, 东方证券, 动态情景多因子Alpha模型, 2016]

### 2.1.1 概述

传统多因子模型大多是全市场范围对所有股票一视同仁打分评价，而忽视了个股的基本面情况差异以及选股因子对于不同风格股票组合的适用性。原文的动态情景多因子模型(Dynamic Contextual Alpha Model)将全市场的股票按照规模、估值、成长等因子进行了划分，在此基础上对不同分档的股票组合中对个股再赋予一定权重；通过这一方法大幅提升模型对于市场风格切换的适应能力。

### 2.1.2 算法

假设调仓频率为月度，调仓时点为每个月末。假设有 $N$ 个分层因子和 $M$ 个 $\alpha$ 因子。

- (i) 在调仓日根据风格因子(可称为分层因子)将股票池分组：方法为因子值从高到低排列，对半分组。每个分层因子对应一个股票池的分组

风格维度	因子名称和含义
规模	总市值
价值	最近财报的净资产/总市值
成长	净资产同比增长率
盈利	净资产收益率
流动性水平	季度日均换手率

表 1: 分层因子表

- (ii) 在不分层的情况下，在调仓日对每个 $\alpha$ 因子进行去极值化和标准化处理, 参见(1.1.2)和(1.1.3)
- (iii) 在不分层的情况下，在调仓日对每个 $\alpha$ 因子和标的阶段收益率 $R$ 进行行业 and 市值中性化处理（原文中称为风险调整），参见(1.1.1)。

$f_{i,t}$ 表示调仓日 $t$ 时刻的原始因子 $i$ 向量， $R_t$ 表示调仓日 $t-1$ 到 $t$ 之间的股票收益向量， $M_t$ 为调仓日 $t$ 时刻的市值向量， $X$ 为行业虚拟矩阵，参见公式(2)。调整后的因子和收益向量为

$$f_t^a = f_t^i - X\beta_1 - \beta_2 * \ln M_t$$

$$R_t^a = R_t - X\beta_1 - \beta_2 * \ln M_t$$

- (iv) 在每个分层空间内，对每个 $\alpha$ 因子计算风险调整IC, 用 $IC_{i,t}^a$ 表示调仓日 $t$ 时刻的基于因子 $i$ 的风险调整因子

$$IC_{i,t}^a = \text{Corr}(f_{i,t-1}^a, R_t^a)$$

- (v) 用过去十二个月（不包括调仓的月末，因为需要预测的是下个月的收益率）的各个 $\alpha$ 因子在分层空间内的风险调整IC，计算因子加权矩阵。用 $\omega_{i,t}$ 表示在某分层中，因子 $i$ 在 $t$ 时刻的权重

$$\omega_{i,t} = \frac{\mu(IC_{i,t-12}^a, IC_{i,t-11}^a \dots IC_{i,t-1}^a)}{\sigma(IC_{i,t-12}^a, IC_{i,t-11}^a \dots IC_{i,t-1}^a)} \quad (9)$$

其中，分子为因子的均值，分母为因子的标准差。

- (vi) 使用月末的风险调整后的因子值，在分层空间内对股票进行排序打分。（当因子加权矩阵值为正时，表示因子值越大越好，因子值越大得分越高；当因子加权矩阵值为负时，表示因子值越小越好，因子值越小得分越高）。

- (vii) 在每个分层内, 计算每个股票的加权得分。用 $\omega_{i,c,t}$ 表示在 $t$ 时刻, 根据公式(9)计算的在分层因子 $c$ 空间内的因子 $i$ 的权重; 用 $s_{i,c,t}^j$ 表示在 $t$ 时刻, 股票 $j$ 在 $c$ 分层因子空间内的 $\alpha$ 因子 $i$ 的排序分数; 那么该时刻在该分层因子下股票 $j$ 的 $\alpha$ 因子加权得分可表示为

$$s_{\alpha,c,t}^j = \sum_{i=1}^M |s_{i,c,t}^j| \omega_{i,c,t}$$

- (viii) 另外还需要对股票在不同情景分层上的属性进行定量刻画(比如全市场估值最高的10%和45%的股票可能存在一些区别)。原文中根据在分层中的分位数定义了一个距离函数来描述股票的情景相似度, 此处本文将该函数其略微修正, 使得函数曲线更平滑。

用 $d_{c,t}^j$ 表示在 $t$ 时刻股票 $j$ 的分层因子 $c$ 的情景得分,  $p$ 为该股票在分层因子 $c$ 划分的分层空间内排序的百分比

$$d_{c,t}^j = \begin{cases} 9 & p \geq 85 \\ 9 \times \left(\frac{p}{35} - \frac{50}{35}\right)^{0.7} & 50 \leq p < 85 \\ -9 \times \left(\frac{50}{35} - \frac{p}{35}\right)^{0.7} & 15 \leq p < 50 \\ -9 & p < 15 \end{cases}$$

- (ix) 最后计算 $t$ 时刻股票 $j$ 的总得分为

$$\gamma_t^j = \sum_{c=1}^N s_{\alpha,c,t}^j \quad (10)$$

根据公式(10)计算的个股总得分进行由高到低的排序和筛选

## 3 附录

### 3.1 PCA推导

假设存在一组原始数据 $\{z_1, z_2, \dots, z_n\}$ , 其中 $z_i$ 为 $p \times 1$ 向量, 即每个数据有 $p$ 个特征。

首先, 将数据中心化 $\{x_1, x_2, \dots, x_n\}$ ,  $x_i = Z_i - \mu$ ,  $\mu = \frac{1}{n} \sum_{i=1}^n z_i$

用 $u_{p \times 1}$ 表示某投影方向上的单位向量, 那么 $x_i$ 在 $u$ 上的投影可以表示为

$$\langle x_i, u \rangle = (x_i^T u)$$

令 $X = (x_1, x_2, \dots, x_n)_{p \times n}$ , 那么所有数据在该方向上的投影的方差为 $\frac{1}{n} u^T X X^T u$ 。

[

*Proof.* 根据方差的定义, 可得投影的方差

$$\frac{1}{n} \sum_{i=1}^n (x_i^T u)^2 - \left[ \frac{1}{n} \sum_{i=1}^n (x_i^T u) \right]^2 \quad (11)$$

公式(11)中的第二项为零, 因为

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n x_i^T u &= \frac{1}{n} \sum_{i=1}^n x_i^T u = \frac{1}{n} \sum_{i=1}^n (z_i^T - \mu^T) u \\ &= \frac{1}{n} \left( \sum_{i=1}^n z_i^T - n \mu^T \right) u \\ &= \frac{1}{n} \left( \sum_{i=1}^n z_i^T - \sum_{i=1}^n z_i^T \right) u \\ &= 0 \end{aligned} \quad (12)$$

第一项可以简化为

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n (x_i^T u)^2 &= \frac{1}{n} \sum_{i=1}^n (x_i^T u)^T (x_i^T u) \quad (x_i^T u \text{ 是一个数}) \\ &= \frac{1}{n} \sum_{i=1}^n u^T x_i x_i^T u \quad (u \text{ 与 } i \text{ 无关}) \\ &= \frac{1}{n} u^T \left( \sum_{i=1}^n x_i x_i^T \right) u \end{aligned} \quad (13)$$

因为 $X = (x_1, x_2, \dots, x_n)_{p \times n}$ , 则 $X^T = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}_{n \times p}$

那么

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n (x_i^T u)^2 - \left[ \frac{1}{n} \sum_{i=1}^n (x_i^T u) \right]^2 &= \frac{1}{n} u^T \left( \sum_{i=1}^n x_i x_i^T \right) u \quad (\text{根据公式 (12) 和公式 (13)}) \\ &= \frac{1}{n} u^T X X^T u \quad (\text{根据 } X \text{ 的定义}) \end{aligned}$$

实际上方差矩阵省略 $\frac{1}{n}$ 对特征向量没有影响, 对特征值大小关系也没有影响。

□



]

此处 $XX^T$ 是半正定对称矩阵。

[

*Proof.* (i)  $(XX^T)^T = XX^T$ ,  $XX^T$ 是对称阵

(ii) 假设 $XX^T$ 某特征值为 $\lambda$ , 对应特征向量为 $\xi$

由特征向量的定义可知 $(XX^T)\xi = \lambda\xi$

两边转置后同乘以 $\xi$ 有

$$(XX^T\xi)^T\xi = (\lambda\xi)^T\xi = \lambda\xi^T\xi \quad (14)$$

又因为

$$(XX^T\xi)^T\xi = \xi^T XX^T\xi = (X^T\xi)^T(X^T\xi) = \|X^T\xi\|^2 \quad (15)$$

由公式(14)和公式(15)可知

$$\begin{aligned} \|X^T\xi\|^2 &= \lambda\xi^T\xi \\ &= \lambda\|\xi\|^2 \geq 0 \\ \implies \lambda &\geq 0 \\ \implies XX^T &\text{半正定} \end{aligned}$$

□

]

作为对称矩阵,  $XX^T$ 的特征向量相互正交, 构成了空间的正交基。(证明略)

假设对称矩阵 $XX^T \in C^{p \times p}$ 的 $p$ 个特征值为

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$$

对应的单位特征向量为 $\xi_1, \xi_2, \dots, \xi_p$ 。这些特征向量表示投影方向的单位向量 $u$ , 假设 $u = \sum_{i=1}^p \alpha_i \xi_i$

$$\begin{aligned} u^T XX^T u &= \langle u, XX^T u \rangle = \left\langle \sum_{i=1}^p \alpha_i \xi_i, \sum_{i=1}^p \alpha_i XX^T \xi_i \right\rangle \\ &= \left\langle \sum_{i=1}^p \alpha_i \xi_i, \sum_{i=1}^p \alpha_i \lambda_i \xi_i \right\rangle \\ &= \lambda_1 \alpha_1^2 + \dots + \lambda_n \alpha_n^2 \quad (\text{单位特征向量}\xi_i, \xi_j \text{正交}, i \neq j) \\ &\leq \lambda_1 (\alpha_1^2 + \dots + \alpha_n^2) \\ &= \lambda_1 \|u\| \\ &= \lambda_1 \quad (u \text{为单位向量}, \|u\| = 1) \end{aligned}$$

故 $u^T XX^T u \leq \lambda_1$

综上所述, 数据集 $X = (x_1, x_2, \dots, x_n)_{p \times n}$ 的投影方差最大为 $\frac{\lambda_1}{n}$ , 该投影方向为 $\lambda_1$ 对应的特征向量的方向。