



QUANTAXIS

QUANTITATIVE FINANCIAL FRAMEWORK

QUANTAXIS FINTECH RESEARCH

Quantitative Finance with Leading Tech Methods

QUANTAXIS

QUANTAXIS DOCS

Author:
@yutiansut

Supervisor:
yutiansut

Strictly Private / Confidential Draft for Discussion Purpose Only

2022 年 5 月 18 日

QUANTAXIS

Leading Quantitative Framework

@yutiansut

2022 年 5 月 18 日

Contents

1 前言	8
1.1 QUANTAXIS 的设计思想	8
1.2 QUANTAXIS 适合谁去使用	8
1.3 QUANTAXIS 适用于哪些场景	8
2 环境准备	9
3 数据	10
4 分析	11
5 交易	12
6 可视化	13
7 基础知识	14
7.1 凸优化	14
7.1.1 KT 条件 Kuhn-Tucker condition	14
7.2 齐次函数的欧拉定理	14
7.3 矩阵	14
7.3.1 相似矩阵	14
7.3.2 对称矩阵	15
7.3.3 正定矩阵	16
7.4 矩阵分解	16
7.4.1 cholesky 分解	16
7.4.2 方阵分解 / EVD 特征值分解	16
7.4.3 非方阵分解 / SVD 奇异值分解	17
7.5 矩阵的二次型	18
7.6 随机矩阵理论 random matrix theory	19
7.6.1 Marcenko–Pastur 律	19
7.6.2 协方差的的特征值分布与 Marcenko–Pastur 分布	19
7.7 波动率, 方差与协方差	20
7.8 协方差矩阵	20
7.8.1 协方差矩阵的半正定性质	21
7.8.2 协方差矩阵的计算	21
7.8.3 协方差矩阵降噪	22
7.8.4 协方差矩阵降噪 Constant Residual Eigenvalue Method	22
7.8.5 协方差矩阵降噪 Targeted Shrinkage	22
7.8.6 协方差矩阵去调 Detoning	22
7.9 参数估计	22
7.9.1 矩估计	22
7.9.2 最大似然 MAX LIKEHOOD	22
7.9.3 GARCH model	22

7.9.4	factor models	22
7.9.5	resample	22
7.9.6	shrinking	22
7.10	常见的分布	23
7.10.1	伯努利分布	23
7.10.2	二项分布	23
7.10.3	多项分布	23
7.10.4	beta 分布	23
7.10.5	狄利克雷分布	23
7.10.6	正态分布	23
7.10.7	t 分布	23
7.10.8	F 分布	23
7.10.9	卡方分布	23
7.10.10	拉普拉斯分布	24
7.10.11	泊松分布	24
7.10.12	指数分布	24
7.10.13	伽马分布	24
7.11	多元线性回归	24
7.11.1	White 估计	24
7.11.2	Newey-West 估计	24
7.12	广义线性回归	24
7.13	聚类	25
7.13.1	KNN	25
7.13.2	K-Means	25
7.14	排序 Ranking	25
7.14.1	PointWise	25
7.14.2	PairWise	25
7.14.3	PairWise - RankingSVM	25
7.14.4	PairWise - RankNet	25
7.14.5	PairWise - LambdaRank	25
7.14.6	PairWise - LambdaMART	25
7.14.7	ListWise	25
7.14.8	ListWise - ListNet	25
7.14.9	ListWise - AdaRank	25
7.14.10	正则化 Regularization	25
7.14.11	LASSO L1	26
7.14.12	Ridge 岭回归 L2	26
7.14.13	Elastic Net L1L2	26
7.15	激活函数	27
7.15.1	Sigmoid	27
7.15.2	tanh	27
7.15.3	ReLU, P-ReLU, Leaky-ReLU	27

7.15.4	ELU	27
7.15.5	Maxout	27
7.15.6	softmax	27
7.16	序列化处理	27
7.16.1	embedding	27
7.16.2	seq-seq	27
7.16.3	BERT	27
7.16.4	LSTM	27
7.16.5	GRU	27
7.16.6	Transform	27
7.16.7	Attention	27
7.17	损失函数	27
7.17.1	MSE 均方差损失	27
7.17.2	MAE 平均绝对误差损失 L1 正则化	28
7.17.3	Huber Loss	28
7.17.4	Quantile Loss 分位数损失	29
7.17.5	Cross Entropy Loss 交叉熵损失	29
7.17.6	Hinge Loss 合页损失	29
7.18	集成学习	29
7.18.1	Bagging	29
7.18.2	Stacking	29
7.18.3	Boosting	29
7.18.4	Boosting - Id3	30
7.18.5	Boosting - C4.5	30
7.18.6	Boosting - CART	30
7.18.7	Boosting - Adaboost	30
7.18.8	Boosting - GBDT	31
7.18.9	Boosting - XGBoost	31
7.18.10	Boosting - LightGBM	31
8	现代资产管理理论	31
8.1	MVO 马科维茨的均值方差优化理论	31
8.1.1	概论	31
8.1.2	方差的二次型转化问题	31
8.1.3	最大收益下的最小组合波动率 Maximum Return –Minimum Volatility	31
8.1.4	风险厌恶系数- $\rightarrow \infty$ 的场景: 最小组合波动率 Minimum Volatility	32
8.1.5	加入卖空约束后的有效前沿	33
8.1.6	加入无风险资产后的 MVO – Capital Market Line	34
8.1.7	CAPM 模型市场均衡与 tangency portfolio	34
8.1.8	TR tracking error	35
8.2	APT 套利定价	35
8.3	SRM 结构化风险模型	35
8.3.1	对外部变化的响应	36

8.3.2	横截面比较因子	36
8.3.3	统计因子	36
8.3.4	因子组合	36
8.3.5	因子协方差矩阵	36
8.4	BARRA 多因子框架	36
8.4.1	因子的逻辑性	36
8.4.2	因子持续性	37
8.4.3	因子显著性	38
8.4.4	因子的信息增量性	38
8.4.5	因子稳健性	38
8.4.6	因子的可投资性	39
8.4.7	普适性	39
8.5	组合风险预算 Risk Budgeting	39
9	组合管理优化的一些必要注意	41
9.1	stability issues	41
9.2	Resampling techniques	41
9.3	Denoising and Detoning	41
9.3.1	Constant residual eigenvalue	41
9.3.2	target shrinkage	41
9.4	协方差收缩 Shrinking methodss	41
9.4.1	基本协方差收缩	41
9.4.2	Ledoit-Wolf 收缩	41
9.4.3	Oracle 近似收缩	41
10	主动组合管理	42
10.1	资产配置	42
10.2	逆方差优化 IVP Inverse Variance	42
10.3	最大多元化投资目标 Maximum Diversification	42
10.4	AAP Agnostic Allocation Portfolio	42
10.5	Black-Litterman 基础理论	43
10.6	TAA	44
10.7	SAA	44
10.8	HRP 分层风险平价	44
10.8.1	聚类分层	44
10.8.2	案例分析	44
11	从风险补偿到期权定价	45
11.1	风险补偿	45
11.2	期权定价	45
11.2.1	波动率计算的六种方法	45
11.2.2	Realized Volatility: Close/Close	45
11.2.3	Parkinson Volatility: High/Low	45
11.2.4	Garman-Klass Volatility: OHLC	45

11.2.5 Roger-Satchell Volatility: OHLC 45

12 过拟合问题 46

12.1 CSCV 46

12.2 GAN 46

12.3 PBO 46

1 前言

QUANTAXIS 实际上是本科毕业论文的一个附带部分的 matlab 程序, 第一个版本大约是在 2016 年夏天的时候, 基于 matlab 的 OOP 编程实现了基础的一些数据获取和账户撮合部分, 在 2017 年的时候, 学习了一阵子 python 的爬虫以后开始进行了重写, 最开始主要是把账户部分从 matlab 搬运过来并进行一个简单的回测过程, 之后又用 nodejs 优化了一下基础的界面部分, 这里就涉及到一些数据的 python-js 的获取, 于是产生了 quantaxiswebserver 项目去作为 nodejs 的数据部分.

大约是在 2018 年的时候, 社区的 @ 安东尼建议我采用 docker 进行一部分的封装于是 httpapi+ docker 逐步产生了一个微服务结构的雏形, 在一路发展过程中, 开源和社区始终是推动着 qa 不断发展的主要动力, 在 qa 的不断修改和重构过程中, 快期的 diff 交易协议也给了我很多的灵感.

2019 年的时候基于实盘的结构, 在 diff 的基础上我扩充了一个 qifi 协议作为 quantaxis 的账户部分的主要协议, 并在测试了 qifiaccount 项目以后逐步替换了老版本的 qaaaccount, 同时 diff 协议也作为后期引入多语言 (rust/go) 等埋下了基础.

2019 年底, 因为业务的需求, 在处理大量的实时账户计算和监控的过程中, python 逐步难以满足业务的需求, 在测试了 go, dart, scala, swift(google) 等语言后, 最终选择了 rust 作为下一代 quantaxis 的底层语言. 经过 2 年左右的不断实践和探索, rust 在 quantaxis 内部的应用越加广泛, 在闭源开发了一阵子以后, 2021 年我选择重新开源部分的 rust 项目, 以希望可以帮助更多的同行者在解决更大的数据分析量以及更快的数据分析需求场景下 python 表现难以为继的问题.

量化的场景看似虽小, 实际上在解决问题的时却需要同时兼顾多个方向和领域的知识, 这需要开发者在充分理解业务的同时发挥自己的想象力, 对于场景进行抽象, 以数据流和数据分析为核心, 构建适合策略场景的量化 infra 架构.

quantaxis 在设计之初就是高度解耦的微服务模式结构, 因此你可以将其作为一个业务结构的 infra 雏形来不断匹配和修改, 尽可能的适配实际的业务场景. 我也很欢迎你在对于业务有了一定的理解和理念的升级以后提交 PR 或者是 ISSUE

1.1 QUANTAXIS 的设计思想

1.2 QUANTAXIS 适合谁去使用

1.3 QUANTAXIS 适用于哪些场景

2 环境准备

3 数据

4 分析

5 交易

6 可视化

7 基础知识

鉴于之后的理论推导和数据分析需要一些高等数学, 线性代数和概率论相关的知识, 在此先温习一下基础知识的推导

7.1 凸优化

7.1.1 KT 条件 Kuhn-Tucker condition

KT / KKT(Karush Kuhn-Tucker) 条件主要是对于拉格朗日乘子法的进一步推广, 因为拉格朗日乘子法要求约束条件为等式, 而现实情况中会有一些非负 (如 long only portfolio 的权重) 的情况. KT 条件是确定约束极值点的必要条件, 对于凸优化而言也是一个充分条件. 我们通过对于原约束条件增加约束系数的形式对于原拉格朗日问题扩展. 首先增加 KT 乘子 β 使得原问题变成 $\operatorname{argmin} \mathcal{L} - \beta\omega$

$$\begin{cases} \beta_i \omega_i = 0 \\ \beta_i \geq 0 \end{cases} \quad (7.1)$$

7.2 齐次函数的欧拉定理

若 $f(x_1, x_2, x_3, \dots, x_s)$ 是 n 次齐次函数, 则有

$$\sum_{\alpha=1}^s \frac{\partial f}{\partial x_{\alpha}} = nf \quad (7.2)$$

证明:

若函数满足条件 $f(ax_1, ax_2, ax_3, \dots, ax_s) = a^n f(x_1, x_2, x_3, \dots, x_s)$, 则称其为 n 次齐次函数.

对上述定义等式两边关于 a 求导有

$$\sum_{\alpha=1}^s \frac{\partial f(ax_1, ax_2, ax_3, \dots, ax_s)}{\partial ax_{\alpha}} * x_{\alpha} = na^{n-1} f(ax_1, ax_2, ax_3, \dots, ax_s) \quad (7.3)$$

令 $a = 1$ 即可得

$$\sum_{\alpha=1}^s \frac{\partial f}{\partial x_{\alpha}} = nf \quad (7.4)$$

7.3 矩阵

7.3.1 相似矩阵

在线性代数中, 相似矩阵 (英语:similar matrix) 是指存在相似关系的矩阵. 相似关系是两个矩阵之间的一种等价关系. 两个 nn 矩阵 A 与 B 为相似矩阵当且仅当存在一个 nn 的可逆矩阵 P , 使得 P 被称为矩阵 A 与 B 之间的相似变换矩阵

设 A, B 都是 n 阶矩阵, 若存在可逆矩阵 P , 使 $P^{-1}AP = B$, 则称 B 是 A 的相似矩阵, 并称矩阵 A 与 B 相似, 记为 $A \sim B$. 对进行运算称为对进行相似变换, 称可逆矩阵为相似变换矩阵对于

设 A, B 和 C 是任意同阶方阵, 则有

- 1) 反身性: $A \sim A$
- 2) 对称性: 若 $A \sim B$, 则 $B \sim A$
- 3) 传递性: 若 $A \sim B, B \sim C$, 则 $A \sim C$
- 4) 若 $A \sim B$, 则 $r(A) = r(B), |A| = |B|, tr(A) = tr(B)$
- 5) 若 $A \sim B$, 且 A 可逆, 则 B 也可逆, 且 $B \sim A$
- 6) 若 $A \sim B$, 则 A 与 B
 - * 两者的秩相等;
 - * 两者的行列式值相等;
 - * 两者的迹数相等;
 - * 两者拥有同样的特征值, 尽管相应的特征向量一般不同;
 - * 两者拥有同样的特征多项式;
 - * 两者拥有同样的初等因子.
- 7) 若 A 与对角矩阵相似, 则称 A 为 **可对角化矩阵**, 若 n 阶方阵 A 有 n 个线性无关的特征向量, 则称 A 为单纯矩阵.
- 8) 相似矩阵具有相同的可逆性, 当它们可逆时, 则它们的逆矩阵也相似.

7.3.2 对称矩阵

对称矩阵 (Symmetric Matrices) 是指以主对角线为对称轴, 各元素对应相等的矩阵. 在线性代数中, 对称矩阵是一个方形矩阵, 其转置矩阵和自身相等 $A = A^T$.

- 1) 对称矩阵的特征值一定是实数
- 2) 对称矩阵的几何重数等于代数重数
- 3) 对称矩阵一定有 n 个线性无关的特征向量
- 4) 对称矩阵一定可以对角化
- 5) 对称矩阵可以正交对角化

每个对称矩阵都可以分解为 $A = Q\Lambda Q^{-1} = Q\Lambda Q^T$, Λ 中为实数的特征值, Q 为标准正交的特征向量. 对称矩阵的正交化证明: 假设 λ_1, λ_2 是对称矩阵 A 的两个特征值, p_1, p_2 是对应

的特征向量, 若 $\lambda_1 \neq \lambda_2$, 则 p_1 与 p_2 正交

$$\begin{aligned}
 \lambda_1 p_1 &= A p_1, \lambda_2 p_2 = A p_2, \lambda_1 \neq \lambda_2 \\
 &\because A, \therefore \\
 \lambda_1 p_1^T &= (\lambda_1 p_1)^T = (A p_1)^T = p_1^T A^T = p_1^T A \\
 &\therefore \\
 \lambda_1 p_1^T p_2 &= p_1^T A p_2 = p_1^T (\lambda_2 p_2) = \lambda_2 p_1^T p_2 \\
 (\lambda_1 - \lambda_2) p_1^T p_2 &= 0 \\
 &\because \lambda_1 \neq \lambda_2 \\
 p_1^T p_2 &= 0
 \end{aligned} \tag{7.5}$$

因此可知 p_1 与 p_2 正交

7.3.3 正定矩阵

给定一个大小为 $n \times n$ 的实对称矩阵 A , 若对于任意长度为 n 的非零向量 x , 有 $x^T A x > 0$ 恒成立, 则矩阵 A 是一个正定矩阵

7.4 矩阵分解

7.4.1 cholesky 分解

主要用于协方差矩阵为实半正定矩阵的情况, 将协方差矩阵分解为上三角阵和下三角阵的乘积

7.4.2 方阵分解 / EVD 特征值分解

假设 A 是线性空间 V 上的一个线性变换, 对于一个非零向量 $\alpha = (x_1, x_2, \dots, x_n)^T$ 使得

$$A\alpha = \lambda\alpha \tag{7.6}$$

则 λ 为 A 的一个特征值, α 为 A 的一个特征向量, 通过

$$\begin{aligned}
 A\alpha &= \lambda\alpha \\
 A\alpha - \lambda\alpha &= 0 \\
 (A - \lambda E)\alpha &= 0 \\
 (A - \lambda E)
 \end{aligned} \tag{7.7}$$

其中 $E = \text{diag}(1, 1, \dots, 1)$ 为单位对角阵, 即可求解其特征值, 进而求解特征向量. 若 A 是一个可逆矩阵, 则上式可以改写为:

$$A = Q \Sigma Q^{-1}$$

这样, 一个方阵 A 就被一组特征值和特征向量表示了. 例如, 对于如下矩阵进行特征值分解.

这里我们用一个简单的方阵来说明特征值分解的步骤. 我们的方阵 A 定义为

$$A = \begin{bmatrix} -1 & 1 & 0 \\ -4 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

首先, 由方阵 A 的特征方程, 求出特征值.

$$|A - \lambda E| = \begin{vmatrix} -1-\lambda & 1 & 0 \\ -4 & 3-\lambda & 0 \\ 1 & 0 & 2-\lambda \end{vmatrix} = (2-\lambda) \begin{vmatrix} -1-\lambda & 1 \\ -4 & 3-\lambda \end{vmatrix} = (2-\lambda)(\lambda-1)^2$$

则特征值为 2, 1

然后, 把每个特征值 λ 带入线性方程组 $(A - \lambda E)x = 0$, 求出特征向量.

当 $\lambda = 2$ 时, 解线性方程组 $(A - 2E)x = 0$.

$$(A - 2E) = \begin{vmatrix} -3 & 1 & 0 \\ -4 & 1 & 0 \\ 1 & 0 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

$$\text{解得 } x_1 = 0, x_2 = 0, \text{ 特征向量为: } p_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

当 $\lambda = 1$ 时, 解线性方程组 $(A - E)x = 0$.

$$(A - E) = \begin{vmatrix} -2 & 1 & 0 \\ -4 & 2 & 0 \\ 1 & -1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{vmatrix}$$

$$\text{解得 } x_1 + x_3 = 0, x_2 + 2x_3 = 0, \text{ 特征向量为: } p_2 = \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}$$

最后, 方阵 A 的特征值分解为:

$$A = Q\Sigma Q^{-1} = \begin{pmatrix} 0 & -1 & -1 \\ 0 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & -1 \\ 0 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix}^{-1}$$

7.4.3 非方阵分解 / SVD 奇异值分解

特征值分解矩阵的缺点: 只能对方阵, 但是我们要分解的大部分都不是方阵

我们前面讲了很多特征值, 特征向量和特征值分解, 而且基于我们以前学习的线性代数知识, 利用特征值分解提取特征矩阵是一个容易理解且便于实现的方法. 但是为什么还存在奇异值分解呢? 特征值分解最大的问题是只能对方阵, 即 $n \times n$ 的矩阵. 而在实际的应用中, 我们分解的大部分都不是方阵.

举个例子:

关系型数据库中的某一张表的数据存储结构就类似于一个二维矩阵, 假设这个表有 m 行, 有 n 个字段, 那么这个表数据矩阵规模就是 $m*n$. 很明显, 在绝大部分情况下, m 与 n 是不相等的. 如果这个时候要对这个矩阵进行特征提取, 特征值分解的方法明显就不行了. 此时, 就可以用 SVD 对非方阵矩阵进行分解.

奇异值分解奇异值分解是一个能适用于任意矩阵的一种分解的方法, 对于任意矩阵 A 总是存在一个奇异值分解:

$$A = U \Sigma V^T \quad (7.8)$$

$m \times n$ $m \times m$ $m \times n$ $n \times n$

假设 A 是一个 $m*n$ 的矩阵, 那么得到的 U 是一个 $m*m$ 的方阵, U 里面的正交向量被称为左奇异向量

Σ 是一个 $m*n$ 的矩阵, Σ 除了对角线其它元素都为 0, 对角线上的元素称为奇异值

V^T 是 V 的转置矩阵, 是一个 $n*n$ 的矩阵, 它里面的正交向量被称为右奇异值向量

一般来讲, 我们会将 Σ 上的值按从大到小的顺序排列.

首先, 我们将 A 和 A 的转置做矩阵的乘法, 得到一个方阵, 用这样的方阵进行特征分解, 得到的特征和特征向量满足下面的等式

$$(AA^T)u_i = \lambda_i u_i \quad (7.9)$$

这里的 u_i 就是左奇异向量

其次, 我们用矩阵 A 的转置乘以 A , 得到一个方阵, 用这样的方阵进行特征分解, 得到的特征值和特征向量满足下面的等式

$$(A^T A)v_i = \lambda_i v_i \quad (7.10)$$

这里的 v_i 就是右奇异向量

又因为:

定理 1: 实正交矩阵 U 满足 $UU^T = I$

定理 2: 对角矩阵 $\Sigma \Sigma^T = \Sigma^2$

有

$$A = U \Sigma V^T \Rightarrow A^T = V \Sigma^T U^T \Rightarrow A^T A = V \Sigma^T U^T U \Sigma V^T = V \Sigma^2 V^T \quad (7.11)$$

对比 $(A^T A)v_i = \lambda_i v_i$ 可以看出

$A^T A$ 的特征向量组成的就是 SVD 中的 V 矩阵

同理可知

$A^T A$ 的特征向量组成的就是 SVD 中的 U 矩阵

7.5 矩阵的二次型

7.6 随机矩阵理论 random matrix theory

起源于对量子力学中大量粒子能级的研究. 数学物理中的许多定律都是通过数值研究发现的. 在 20 世纪 50 年代后期, E. P. Wigner 根据随机矩阵的经验分布来阐述该问题 (Wigner 1955, 1958), 从而开始了对高斯矩阵半圆定律的研究. 自那以后, RMT 在现代概率论中便形成了一个活跃的分支.

设 A 是一个 $m \times n$ 矩阵, 其特征值为 $\lambda_1, \dots, \lambda_n$ 如果所有 λ_j 都是实值, 那么我们可以构造一个 1 维经验分布函数 (empirical distribution function)

$$F^A(x) = \frac{1}{n} \sum_{j=1}^n I(\lambda_j \leq x)$$

否则, 我们可以通过 λ_j 的实部和虚部构造一个 2 维经验分布函数, 即

$$F^A(x, y) = \frac{1}{n} \sum_{j=1}^n I(\Re(\lambda_j) \leq x; \Im(\lambda_j) \leq y)$$

然后, 我们称 F^A 为 A 的经验谱分布 (empirical spectral distribution, ESD). RMT 的主要任务是在 A 随机且阶数 n 趋于无穷的情况下研究 F^A 的极限性质. 如果存在极限分布 F , 那么我们称该极限为 A 序列的极限谱分布 (limiting spectral distribution, LSD). 存在很多有趣的问题, 包括找出 LSD (如果其存在) 的显式形式, 并研究其性质.

有两种方法用于确定 F^A 的极限性质 (Bai 1999). 一种是矩方法 (method of moments), 使用 F^A 的矩就是 A 的幂的缩放迹 (scaled traces) 这一事实. 另一种是使用 Stieltjes 变换, 对于任意分布函数 F , 定义如下

$$m(z) = \int dx \frac{1}{x - z} dF(x), z \in \mathbb{C}$$

与在大维随机矩阵的特征值上所取得的进展不同, 在本征矩阵 (eigenmatrix, 即 A 的标准化特征向量的矩阵) 的极限性质上已获得的结果非常少. 由于其在统计学及应用领域中的重要性, 对本征矩阵的研究现在变得更加活跃.

7.6.1 Marcenko-Pastur 律

令 $X = (x_{ij})_{p \times n}$, 其元素是均值为 0, 方差为 1 的独立随机变量. 如果 $p/n \rightarrow y \in (0, \infty)$ 且对于任意的 $\delta > 0$,

$$\frac{1}{np} \sum_{ij} E|x_{ij}|^2 I(|x_{ij}| > \delta \sqrt{n}) \rightarrow 0$$

那么 $S_n = \frac{1}{n} X X^*$ (即所谓的采样协方差矩阵 (sample covariance matrix)) 的 ESD 趋向于密度为

$$\rho = \frac{1}{2\pi x y \sigma^2} \sqrt{(\lambda_+ - x)(x - \lambda_-)} I(\lambda_- < x < \lambda_+)$$

的 Marcenko-Pastur 律, 这里 $\lambda_- = \sigma^2(1 - \sqrt{y})^2$, $\lambda_+ = \sigma^2(1 + \sqrt{y})^2$. 此外, 如果 $y > 1$, 那么 LSD 在原点处的点质量为 $1 - 1/y$.

7.6.2 协方差的的特征值分布与 Marcenko-Pastur 分布

在 <Noise Dressing of Financial Correlation Matrices> 文章中, 主要对于金融标的相关系数矩阵的特征值分布与 RMT 的预测有一定的吻合, 基于现实金融数据的计算表现出

有少量特征值超过随机矩阵预测的估计的最大值, 其次在去除这些超过估计的较大的特征值之后, 与随机矩阵理论预测的分布吻合的很好. 在文中, 作者们把最大的特征值对应到市场本身, 认为是市场成分. 前面所示的现象在金融数据处理中反复得到验证, 所以随机矩阵一直是一个基本研究工具.

$$C_{ij} = \frac{1}{T} \sum_{t=1}^T \delta x_i(t) \delta x_j(t) = \frac{1}{T} M M^T$$

则协方差矩阵 C 的特征值的密度函数为

$$\rho_C(\lambda) = \frac{1}{N} \frac{dn(\lambda)}{d\lambda}$$

, 我们令 M 为 $T \times N$ 的随机矩阵, 在 $N \rightarrow \infty, T \rightarrow \infty, Q = T/N \geq 1$ 的时候可以得知 $\rho_C(\lambda)$ 的近似约为

$$\rho_C(\lambda) = \frac{Q}{2\pi\sigma^2} \frac{\sqrt{(\lambda_{\max} - \lambda)(\lambda - \lambda_{\min})}}{\lambda}$$

我们可以推算出 λ 的上下限

$$\lambda_{\min}^{\max} = \sigma^2(1 + 1/Q \pm 2\sqrt{1/Q})$$

7.7 波动率, 方差与协方差

首先, 我们先讨论下金融学中的风险. 在金融中, 我们通常会使用波动率来衡量一个资产的风险程度, 而波动率则可以使用方差/标准差来代替

$$\sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (7.12)$$

当我们把风险的定义扩展到多个资产的时候, 首先我们考虑 x, y 两种资产的情况

$$\sigma_{portfolio} = \sqrt{(w_x \sigma_x + w_y \sigma_y)^2} = \sqrt{w_x^2 \sigma_x^2 + w_y^2 \sigma_y^2 + 2w_x w_y \sigma_{x,y} \sigma_x \sigma_y} \quad (7.13)$$

而对于其中的协方差 $\sigma(x, y)$, 有

$$\sigma(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (7.14)$$

由协方差的公式也不难看出, 方差只是协方差的一个特殊形式 ($x = y$)

7.8 协方差矩阵

当我们的组合中的资产数量超过 2 个的时候, 我们需要推广到一个更加一般化的结果中去:

首先, 根据方差的定义, 我们可以推导出一个更加一般化的结论:

给定 d 个随机变量 $x_k, k = 1, 2, 3, \dots, d$, 则这些随机变量的方差为

$$\sigma(x_k, x_k) = \frac{1}{n-1} \sum_{i=1}^n (x_{ki} - \bar{x}_k)^2, k = 1, 2, 3, \dots, d \quad (7.15)$$

对于这些随机变量, 我们还可以根据协方差的定义, 求出两两之间的协方差, 即

$$\sigma(x_m, x_k) = \frac{1}{n-1} \sum_{i=1}^n (x_{mi} - \bar{x}_m)(x_{ki} - \bar{x}_k) \quad (7.16)$$

因此, 协方差矩阵为

$$\Sigma = \begin{bmatrix} \sigma(x_1, x_1) & \cdots & \sigma(x_1, x_d) \\ \vdots & \ddots & \vdots \\ \sigma(x_d, x_1) & \cdots & \sigma(x_d, x_d) \end{bmatrix} \quad (7.17)$$

其中, 对角线上的元素为各个随机变量的方差, 非对角线上的元素为两两随机变量之间的协方差, 根据协方差的定义, 我们可以认定: 矩阵 Σ 为对称矩阵 (symmetric matrix), 其大小为 $d \times d$.

回到我们已经学过的线性代数内容, 对于任意对称矩阵 Σ , 存在一个特征值分解 (eigenvalue decomposition, EVD)

$$\Sigma = U \Lambda U^T \quad (7.18)$$

其中, U 的每一列都是相互正交的特征向量, 且是单位向量, 满足 $UU^T = I$, Λ 对角线上的元素是从大到小排列的特征值, 非对角线上的元素均为 0.

7.8.1 协方差矩阵的半正定性

协方差矩阵是一个半正定矩阵

证明: 现给定任意一个向量 x , 对于任意多元变量 t , 协方差矩阵为 $C = E[(t - \bar{t})(t - \bar{t})^T]$

$$\begin{aligned} x^T C x &= x^T E[(t - \bar{t})(t - \bar{t})^T] x = E[(t - \bar{t})(t - \bar{t})^T x] = E(s^2) = \sigma_s^2 \\ \sigma_s &= x^T (t - \bar{t}) = (t - \bar{t})^T x \\ \sigma_s^2 &\geq 0 \Rightarrow x^T C x \geq 0 \end{aligned} \quad (7.19)$$

作为一个半正定矩阵, 我们可以快速推导出两个性质

- 1) 设 A 为 n 阶对称矩阵, 则必有正交矩阵 P , 使得 $P^{-1}AP = P^TAP = B$, 其中 B 是以 A 的特征值为对角线元素的对角矩阵.
- 2) 作为半正定矩阵, 我们可以对协方差矩阵进行 Cholesky 分解: 半正定矩阵 Σ , 可以分解为 $\Sigma = U^T \Lambda U$, 其中 U 是上三角阵, Λ 是对角线元素都非负的对角矩阵. 所以

$$\Sigma = U^T \Lambda U = [U^T \Lambda^{1/2}][\Lambda^{1/2} U] = [\Lambda^{1/2} U]^T [\Lambda^{1/2} U] \quad (7.20)$$

这样一来, 矩阵 $\Sigma = C^T C$, 其中 $C = \Lambda^{1/2} U$

7.8.2 协方差矩阵的计算

<https://zhuanlan.zhihu.com/p/36252293>

- * 先让样本矩阵中心化, 即每一维度减去该维度的均值, 使每一维度上的均值为 0
- * 然后直接用新的到的样本矩阵乘上它的转置
- * 然后除以 (N-1) 即可

7.8.3 协方差矩阵降噪

7.8.4 协方差矩阵降噪 Constant Residual Eigenvalue Method

7.8.5 协方差矩阵降噪 Targeted Shrinkage

7.8.6 协方差矩阵去调 Detoning

7.9 参数估计

现代金融体系 MPT-APT 都是建立在资产的收益率是正态分布上的, 而实际实务中的资产/组合过多, 同时还有大量的黑天鹅事件的产生, 因此我们对于正态分布下的两个重要参数均值/方差的估计就非常必要. 而在此中, 最核心的估计还是对于资产的协方差矩阵的估计.

7.9.1 矩估计

7.9.2 最大似然 MAX LIKEHOOD

最大似然估计是高斯分布下最常用的对于协方差的估计方法, 我们在假设了资产的独立同分布之后, 我们利用一部分观察到的资产情况对于整体的协方差矩阵进行快速的估计. 我们写出最大似然函数

$$\mathcal{L}(\mu, \Sigma) = -\frac{nT}{2} \ln(2\pi) - \frac{T}{2} \ln|\Sigma| - \frac{1}{2} \sum_{t=1}^T (R_t - \mu)^T \Sigma^{-1} (R_t - \mu)$$

7.9.3 GARCH model

7.9.4 factor models

7.9.5 resample

7.9.6 shrinking

在矩阵的求逆过程中, 最大似然估计不是协方差矩阵的特征值的一个很好的估计, 所以从反演得到的精度矩阵是不准确的. 有时, 甚至出现因矩阵元素地特性, 经验协方差矩阵不能求逆. 为了避免这样的反演问题, 入了经验协方差矩阵的一种变换方式, 收缩协方差.

7.10 常见的分布

7.10.1 伯努利分布

7.10.2 二项分布

7.10.3 多项分布

7.10.4 beta 分布

贝塔分布 (Beta Distribution) 是一个作为伯努利分布和二项式分布的共轭先验分布的密度函数，在机器学习和数理统计学中有重要应用。在概率论中，贝塔分布，也称 B 分布，是指一组定义在 $(0,1)$ 区间的连续概率分布。

7.10.5 狄利克雷分布

狄利克雷分布 (Dirichlet distribution) 是多项分布的共轭分布，也就是它与多项分布具有相同形式的分布函数。

7.10.6 正态分布

正态分布 (Normal distribution) 又名高斯分布 (Gaussiandistribution)，若随机变量 X 服从一个数学期望为 μ 、方差为 σ^2 的高斯分布，记为 $N(\mu, \sigma^2)$ 。其概率密度函数为正态分布的期望值 μ 决定了其位置，其标准差 σ 决定了分布的幅度。我们通常所说的标准正态分布是 $\mu = 0, \sigma = 1$ 的正态分布

7.10.7 t 分布

T 分布是类似于正态分布的一种对称分布，他通常要比正态分布平坦和分散。一个特定的 t 分布依赖于称之为自由度的参数，随着自由度的增大，t 分布也逐渐趋于正太分布。用于根据小样本来估计呈正态分布且方差未知的总体的均值。

7.10.8 F 分布

设 X 、 Y 为两个独立的随机变量， X 服从自由度为 n 的分布， Y 服从自由度为 m 的分布，这两个独立的卡方分布除以各自的自由度以后的比率服从 F 分布

7.10.9 卡方分布

卡方检验是一种用途很广的基于卡方分布的假设检验方法，其根本思想就是在于比较理论频数和实际频数的吻合程度或拟合优度问题。其主要应用于分类变量，根据样本数据推断总体分布与期望分布是否有显著差异或推断两个分类变量是否相关或相互独立。卡方检验可以参照一般假设检验步骤：设置原假设与备择假设设置显著性水平根据问题选择具体的假设检验方式计算统计量，并通过统计量获取 P 值根据 P 值与显著性水平，决定接受

原假设还是备择假设一般可以设原假设为：观察频数与期望频数没有差异，或者两个变量相互独立不相关。卡方检验的计算公式为：

$$\chi^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

从公式也可以看出它是利用类别变量的观测值频数与期望值频数进行构建的

```
crit = stats.chi2.ppf(q=0.95, df=5)
print(crit)
P_value = 1 - stats.chi2.cdf(x=chi_squared_stat, df=5)
print('P_value')
print(P_value)
stats.chisquare(f_obs=observed, #Array of observed counts
               f_exp=expected) #Array of expected counts
```

7.10.10 拉普拉斯分布

7.10.11 泊松分布

7.10.12 指数分布

7.10.13 伽马分布

7.11 多元线性回归

7.11.1 White 估计

white 估计是一种对于异方差的检验方法，其优点是不需要对于异方差的性质做出任何的假定

7.11.2 Newey-West 估计

7.12 广义线性回归

在处理较为复杂的数据的回归问题时，普通的线性回归算法通常会出现预测精度不够，如果模型中的特征之间有相关关系，就会增加模型的复杂程度。当数据集中的特征之间有较强的线性相关性时，即特征之间出现严重的多重共线性时，用普通最小二乘法估计模型参数，往往参数估计的方差太大，此时，求解出来的模型就很不稳定。在具体取值上与真值有较大的偏差，有时会出现与实际意义不符的正负号。

岭回归与 Lasso 回归的出现是为了解决线性回归出现的过拟合以及在通过正规方程方法求解的过程中出现的 $(X^T X)$ 不可逆这两类问题的，这两种回归均通过在损失函数中引入正则化项来达到目的。在日常机器学习任务中，如果数据集的特征比样本点还多， $(X^T X)^{-1}$ 的时候会出错。岭回归最先用来处理特征数多于样本数的情况，现在也用于在估计中加入偏差，从而得到更好的估计。这里通过引入 λ 限制了所有 β 之和，通过引入该惩罚项，能够减少不重要的参数，这个技术在统计学上也叫作缩减 (shrinkage)。和岭回归类似，另一个缩减 LASSO 也加入了正则项对回归系数做了限定。为了防止过拟合 (θ 过大)，在目标函数 $J(\theta)$

后添加复杂度惩罚因子, 即正则项来防止过拟合. 正则项可以使用 $L1 - norm(Lasso)$, $L2 - norm(Ridge)$, 或结合 $L1 - norm, L2 - norm(ElasticNet)$.

7.13 聚类

7.13.1 KNN

7.13.2 K-Means

7.14 排序 Ranking

7.14.1 PointWise

7.14.2 PairWise

7.14.3 PairWise - RankingSVM

7.14.4 PairWise - RankNet

7.14.5 PairWise - LambdaRank

7.14.6 PairWise - LambdaMART

7.14.7 ListWise

7.14.8 ListWise - ListNet

7.14.9 ListWise - AdaRank

7.14.10 正则化 Regularization

正则化 (Regularization) 是机器学习中一种常用的技术, 其主要目的是控制模型复杂度, 减小过拟合. 最基本的正则化方法是在原目标 (代价) 函数中添加惩罚项, 对复杂度高的模型进行“惩罚”。其数学表达形式为:

$$\hat{J}(\omega; X, y) = J(\omega; X, y) + \alpha \Omega(\omega)$$

, 式中 X, y 为训练样本和相应标签, ω 为权重系数向量; $J()$ 为目标函数, $\Omega(\omega)$ 即为惩罚项, 可理解为模型“规模”的某种度量; 参数 α 控制控制正则化强弱. 不同的 Ω 函数对权重 ω 的最优解有不同的偏好, 因而会产生不同的正则化效果. 最常用的 Ω 函数有两种, 即 l_1 范数和 l_2 范数, 相应称之为 l_1 正则化和 l_2 正则化. 此时有:

$$l_1 : \Omega(\omega) = \|\omega\| = \sum |\omega|$$

$$l_2 : \Omega(\omega) = (\|\omega\|)^2 = \sum \omega^2$$

l_1 正则化是指权值向量 w 中各个元素的绝对值之和, l_2 正则化是指权值向量 w 中各个元素的平方和然后再求平方根 (可以看到 Ridge 回归的 l_2 正则化项有平方符号)

7.14.11 LASSO L1

L1 正则化可使解更加靠近某些轴, 而其它的轴则为 0, 所以 L1L1 正则化能使得到的参数稀疏化, 若用 VCVC 维衡量模型的复杂度, 其参数个数得到了压缩, 因此控制了模型的复杂度。

$$(1/(2 * nsamples)) * \|y - Xw\|_2^2 + alpha * \|w\|_1$$

```
>>> from sklearn import linear_model
>>> clf = linear_model.Lasso(alpha=0.1)
>>> clf.fit([[0,0], [1, 1], [2, 2]], [0, 1, 2])
Lasso(alpha=0.1)
>>> print(clf.coef_)
[0.85 0. ]
>>> print(clf.intercept_)
0.15...
```

7.14.12 Ridge 岭回归 L2

$$\|y - Xw\|_2^2 + alpha * \|w\|_2^2$$

```
>>> from sklearn.linear_model import Ridge
>>> import numpy as np
>>> n_samples, n_features = 10, 5
>>> rng = np.random.RandomState(0)
>>> y = rng.randn(n_samples)
>>> X = rng.randn(n_samples, n_features)
>>> clf = Ridge(alpha=1.0)
>>> clf.fit(X, y)
Ridge()
```

7.14.13 Elastic Net L1L2

$$1/(2 * nsamples) * \|y - Xw\|_2^2 + alpha * l_1ratio * \|w\|_1 + 0.5 * alpha * (1 - l_1ratio) * \|w\|_2^2$$

```
from sklearn.linear_model import ElasticNet
from sklearn.datasets import make_regression

>>> X, y = make_regression(n_features=2, random_state=0)
>>> regr = ElasticNet(random_state=0)
>>> regr.fit(X, y)
ElasticNet(random_state=0)
>>> print(regr.coef_)
[18.83816048 64.55968825]
>>> print(regr.intercept_)
1.451...
>>> print(regr.predict([[0, 0]]))
[1.451...]
```

7.15 激活函数

<https://www.cnblogs.com/missidiot/p/9378079.html>

7.15.1 Sigmoid

7.15.2 tanh

7.15.3 ReLU, P-ReLU, Leaky-ReLU

7.15.4 ELU

7.15.5 Maxout

7.15.6 softmax

7.16 序列化处理

7.16.1 embedding

7.16.2 seq-seq

7.16.3 BERT

7.16.4 LSTM

7.16.5 GRU

7.16.6 Transform

7.16.7 Attention

7.17 损失函数

7.17.1 MSE 均方差损失

$$\mathcal{J}_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

MSE 背后的逻辑是模型的输出值和真实值的误差满足高斯分布下的最大似然估计, 我们假设模型的输出和真实的误差值服从标准的高斯分布 ($\mu = 0, \sigma = 1$), 则给定 x_i 输出真实的 y_i 的概率为

$$P(y_i|x_i) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2}\right)$$

进一步我们假设数据集中 N 个样本点之间相互独立, 则给定所有 x 输出所有真实 y 的概率 (似然 LikeHood), 为所有 $P(y_i|x_i)$ 的累乘.

$$L(x, y) = \prod_i \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2}\right)$$

为了计算方便, 我们使用最大对数化似然函数 MAX LOG LIKEHOOD

$$LL(x, y) = \log(L(x, y)) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \sum_i (y_i - \hat{y}_i)^2$$

去掉与 y_i 无关的第一项, 并转化为 Negative MLLikehood

$$NLL = \frac{1}{2} \sum_i (y_i - \hat{y}_i)^2$$

可以看出这个最大似然估计就是 MSE 的表现形式

7.17.2 MAE 平均绝对误差损失 L1 正则化

$$\mathcal{J}_{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

MAE 背后的逻辑是模型的输出值和真实值的误差满足拉普拉斯分布下的最大似然估计, 我们假设模型的输出和真实的误差值服从标准的拉普拉斯分布 ($\mu = 0, b = 1$), 则给定 x_i 输出真实的 y_i 的概率为

$$P(y_i|x_i) = \frac{1}{2} \exp(-|y_i - \hat{y}_i|)$$

进一步我们假设数据集中 N 个样本点之间相互独立, 则给定所有 x 输出所有真实 y 的概率 (似然 LikeHood), 为所有 $P(y_i|x_i)$ 的累乘.

$$L(x, y) = \prod_i \frac{1}{2} \exp(-|y_i - \hat{y}_i|)$$

为了计算方便, 我们使用最大对数化似然函数 MAX LOG LIKEHOOD

$$LL(x, y) = \log(L(x, y)) = -\frac{N}{2} - \sum_i |y_i - \hat{y}_i|$$

去掉与 y_i 无关的第一项, 并转化为 Negative MLLikehood

$$NLL = \sum_i |y_i - \hat{y}_i|$$

7.17.3 Huber Loss

Huber Loss 是一种将 MAE 与 MSE 合并的损失函数, 通过设定一个超参数 δ , 在误差接近 0 的时候使用 MAE, 而在误差较大的时候使用 MSE

$$\mathcal{J}_{HuberLoss} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_{|y_i - \hat{y}_i| \leq \delta} \frac{(y_i - \hat{y}_i)^2}{2} + \mathbb{I}_{|y_i - \hat{y}_i| \geq \delta} (\delta |y_i - \hat{y}_i| - \frac{1}{2} \delta^2)$$

```
def huber(true, pred, delta):  
    loss = np.where(np.abs(true-pred) < delta, 0.5*((true-pred)**2), delta*np.abs(true  
        - pred) - 0.5*(delta**2))  
  
    return np.sum(loss)
```

7.17.4 Quantile Loss 分位数损失

7.17.5 Cross Entropy Loss 交叉熵损失

7.17.6 Hinge Loss 合页损失

7.18 集成学习

<https://github.com/yutiansut/DataScience>

7.18.1 Bagging

bagging 模型会把 (相同的) 若干基础模型简单的“装起来”——基础模型独立训练, 然后将它们的输出用特定的规则综合 (比如求平均值) 起来, 形成最后的预测值。最常见的 bagging 模型是随机森林 (Random Forest)。在回归类任务中, bagging 模型假设各个基础模型的预测值“错落有致”, 分布在真实值的周围——把这些预测值平均一下, 就可以稳定地得到一个比较准确的预测值; 而在分类任务中, bagging 模型认为“每一个基础模型都判断错误”发生的概率是比较低的, 基础模型中的相当一部分会做出正确的判断, 因此可以基于大家的投票结果来选择最终类别。

7.18.2 Stacking

Stacking 模型比 bagging 模型更进 2 步: (a) 允许使用不同类型的模型作为 base model; (b) 使用一个机器学习模型把所有 base model 的输出汇总起来, 形成最终的输出。(b) 所述的模型被称为“元模型”。在训练的时候, base model 们直接基于训练数据独立训练, 而元模型会以它们的输出为输入数据、以训练数据的输出为输出数据进行训练。Stacking 模型认为, 各个基础模型的能力不一, 投票的时候不能给以相同的权重, 而需要用一个“元模型”对各个基础模型的预测值进行加权。

7.18.3 Boosting

Boosting 模型采用另一种形式, 把基础模型组合起来——串联。这类模型的思想是, 既然一个基础模型可以做出不完美的预测, 那么我们可以用第二的基础模型, 把“不完美的部分”补上。我们可以使用很多的基础模型, 不断地对“不完美的部分”进行完善, 以得到效果足够好的集成模型。Boosting 的策略非常多, 以 GBDT 为例, 它会用第 K 个 CART 拟合前 k-1 个 CART 留下的残差, 从而不断的缩小整个模型的误差

7.18.4 Boosting - Id3

ID3 算法是基于信息增益来度量划分数据集前后不确定性减少的程度来分裂节点

7.18.5 Boosting - C4.5

C4.5 算法是基于信息增益比来度量划分数据集前后不确定性减少的程度来分裂节点

7.18.6 Boosting - CART

CART 算法分为两种, 对于分类问题, 采用基尼指数最小化进行特征选择, 而回归树则选择平方误差最小化 MIN MSE

$$Gini(D) = 1 - \sum_{k=1}^N p_k^2$$

样本分布越集中, 基尼系数越小, 当只有 1 个样本时, $gini = 1 - 1 = 0$, 而反之样本分布越均匀, Gini 越大

$$MSE(D) = \sum_{x_i \in R^n} (y_i - f(x_i))^2$$

```
def MSE(prediction, real_value):
    r = 0
    for i in range(len(prediction)):
        r += (prediction[i] - real_value[i])**2
    return r
```

7.18.7 Boosting - Adaboost

- 先通过对 N 个训练样本的学习得到第一个弱分类器
- 将分错的样本和其他的新数据一起构成一个新的 N 个的训练样本, 通过对这个样本的学习得到第二个弱分类器
- 将 1 和 2 都分错了的样本加上其他的新样本构成另一个新的 N 个的训练样本, 通过对这个样本的学习得到第三个弱分类器
- 最终经过提升的强分类器即某个数据被分为哪一类要由各分类器权值决定

7.18.8 Boosting - GBDT

7.18.9 Boosting - XGBoost

7.18.10 Boosting - LightGBM

8 现代资产管理理论

8.1 MVO 马科维茨的均值方差优化理论

8.1.1 概论

我们首先对于一般性的组合的收益进行规定 $R_p = \omega^T R_i$, 因此组合的方差 $Var(R) = \sigma^2 = Var(\omega^T R) = \omega_1^2 \sigma_1^2 + \omega_2^2 \sigma_2^2 + \omega_3^2 \sigma_3^2 + \dots + \omega_n^2 \sigma_n^2 + 2\omega_1 \omega_2 \sigma_1 \sigma_2 \sigma_{12} + 2\omega_2 \omega_3 \sigma_2 \sigma_3 \sigma_{23} + \dots + 2\omega_{n-1} \omega_n \sigma_{n-1} \sigma_n \sigma_{n-1,n}$, 通过之前的线性代数基础知识, 我们可以快速的把这个式子转化为一个二次型问题, 及

$$Var(R_p) = Var(\omega^T R) = \omega^T Cov \omega = \omega' \Sigma \omega$$

, 我们再重新考虑 MVO 的问题, 我们可以对于不同的约束目标进行限定, 从而获得不同的约束下的最优组合

8.1.2 方差的二次型转化问题

我们首先把组合的收益率表达为一个矩阵的形式, $R_p = \mathbb{E}(R(x)) = \mathbb{E}(\omega^T R(x)) = \omega^T \mathbb{E}(R) = \omega^T \mu$, 因此, 我们对于这个收益率矩阵求解方差

$$\sigma^2 = \mathbb{E}[(R(x) - \mu(x))(R(x) - \mu(x))^T] = \mathbb{E}[(\omega^T R - \omega^T \mu)(\omega^T R - \omega^T \mu)^T]$$

又因为 $(\omega^T)^T = \omega$ 因此我们可以推出

$$\sigma^2 = \omega^T \mathbb{E}[(R - \mu)(R - \mu)^T] \omega = \omega^T \Sigma \omega$$

, 因此之后的 MVO 问题可以从一般的均值/方差问题进行进一步的扩展, 基于不同的市场情况我们可以对于 μ 以及 ω 进行不同的约束, 进而转化为一个凸优化的 QP 求解问题

8.1.3 最大收益下的最小组合波动率 Maximum Return -Minimum Volatility

我们需要获得在最大收益下的最小组合波动率, 及

$$\max_{\omega} \omega' R - \frac{\delta}{2} \omega' \Sigma \omega$$

其中 ω 是资产权重向量, R 和 Σ 分别为预期收益率向量以及协方差矩阵, δ 是风险厌恶系数. 我们首先讨论 δ , 当 $\delta > 0$ 的时候, 原式子转化为对于最大收益率的考量, 而当 $\delta \rightarrow \infty$ 时, 原式子转化为对于风险的无限厌恶

当不考虑任何约束是, 上述 MVO 问题的最优解是

$$\omega_{mvo} = (\delta \Sigma)^{-1} R$$

如果考虑所有资产权重加起来占资金量 100, 即 $\sum \omega = 1$, 则可以将上述最优解中的等号换成‘正比于’符号, 它和最终的权重只差一个 scaling factor 而已

$$\omega_{mvo} \propto (\delta \Sigma)^{-1} R$$

对于权重为 ω 的投资组合, 其夏普率为

$$SharpeRatio = \frac{\omega^{-1} R}{\sqrt{\omega' \Sigma \omega}}$$

数学推导可以证明, 最大化投资组合夏普率的 ω 也正比于 $\Sigma^{-1} R$, 因此 MVO 的最优解 ω_{mvo} 可以最大化投资组合的夏普率. 这就是它吸引人的原因. 当然, 由于对输入 (R 和 Σ) 非常敏感, 且事前预测未来的 R 和 Σ 异常困难 (Σ 中包括资产自己的波动率 σ_i 以及资产间的相关系数 ρ_{ij}), MVO 在投资实务中也没少被人诟病. 为了避免猜 R 和 Σ 的问题, 人们又相继提出了很多其他的资产配置方法, 诸如: equal weight, minimum variance, maximum diversification 以及 risk parity (又称 equal risk contribution). 在实际资产配置中, 由于 R_i , σ_i 和 ρ_{ij} 中的一个或多个难以预测, 我们舍弃 MVO 并退而求其次选择上述这些方法之一. 无论采取哪种方法, 我们都希望最大化投资组合的夏普率. 事实上, 当 R_i , σ_i 和 ρ_{ij} 这些参数满足特定条件时, 所有上述方法均可以等价于 MVO. 在资产配置时, 我们可以对资产参数 R_i , σ_i 和 ρ_{ij} 所满足的条件做适当的假设, 从而选择最合适的配置方法.

8.1.4 风险厌恶系数 $\rightarrow \infty$ 的场景: 最小组合波动率 Minimum Volatility

上世纪 50 年代, Markowitz 基于数理方法提出了基于均值-方差的证券投资组合模型, 其假设资产收益率服从多元正态分布, 并且投资者厌恶风险、遵循投资分散化原则使用效用函数进行投资. 投资目标是在给定权重、预期收益 (μ) 或风险 (σ) 下, 实现预期收益最大、或证券风险最小的最优规划, 其代数形式表现如下:

目标函数

$$\begin{aligned} \min & \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j \text{Cov}(r_i, r_j) \\ \max & \sum_{i=1}^n \omega_i r_i \end{aligned}$$

我们约定持仓权重的加总为 1 (满仓) 及

$$\sum_{i=1}^n \omega_i = 1$$

可以构建拉格朗日乘子法来确定最优的解析解, 有

$$\mathcal{L}(\omega, \lambda_1, \lambda_2) = \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j \text{Cov}(r_i, r_j) - \lambda_1 \left(\sum_{i=1}^n \omega_i \bar{r}_i - E(R) \right) - \lambda_2 \left(\sum_{i=1}^n \omega_i - 1 \right)$$

, 对上式各部分分别求导, 我们有

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \omega_i} &= \sum_{j=1}^n \omega_j \text{Cov}(r_i, r_j) - \lambda_1 \bar{r}_i - \lambda_2 1_n = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_1} &= \sum_{i=1}^n \omega_i \bar{r}_i - E(R) = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_2} &= \sum_{i=1}^n \omega_i - 1 = 0 \end{aligned} \tag{8.1}$$

记 $\omega_j = \omega, \bar{r}_i = R, Cov(r_i, r_j) = \Sigma$, 有

$$\begin{aligned} 2\Sigma\omega - \lambda_1 R - \lambda_2 1_n &= 0 \\ \omega^T R - E(R) &= 0 \\ \omega^T 1_n - 1 &= 0 \end{aligned} \quad (8.2)$$

我们假定投资组合的各个资产不存在完全的线性相关, 及收益率的协方差矩阵 $Cov(r_i, r_j)$ 可逆, 继续带入上式推导可得

$$\begin{aligned} \omega &= 1/2\Sigma^{-1}(\lambda_1 R + \lambda_2 1_n) \\ E(R) &= \omega^T R \\ \omega^T 1_n &= 1 \end{aligned} \quad (8.3)$$

化简可得

$$\begin{cases} E(R) = (1/2\Sigma^{-1}(\lambda_1 R + \lambda_2 1_n))^T R = 1/2(\lambda_1 R^T \Sigma^{-1} R + \lambda_2 1_n^T \Sigma^{-1} R) \\ 1 = (1/2\Sigma^{-1}(\lambda_1 R + \lambda_2 1_n))^T 1_n = 1/2(\lambda_1 R^T \Sigma^{-1} 1_n + \lambda_2 1_n^T \Sigma^{-1} 1_n) \end{cases} \quad (8.4)$$

$$\therefore \begin{vmatrix} E(R) \\ 1 \end{vmatrix} = 1/2 \begin{vmatrix} R^T \Sigma^{-1} R & 1_n^T \Sigma^{-1} R \\ R^T \Sigma^{-1} 1_n & 1_n^T \Sigma^{-1} 1_n \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \end{vmatrix}$$

8.1.5 加入卖空约束后的有效前沿

加入卖空约束后, 我们加入对于权重向量 $\omega_i \geq 0$ 的约束, 则新的问题可以表达为:

$$\begin{aligned} \min \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j Cov(r_i, r_j) \\ \begin{cases} \sum_{i=1}^n \omega_i \bar{r}_i = E(R) \\ \sum_{i=1}^n \omega_i = 1 \\ \omega_i \geq 0 \end{cases} \end{aligned} \quad (8.5)$$

构造拉格朗日函数, 可得

$$\mathcal{L}(\omega, \lambda_1, \lambda_2, \beta) = \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j Cov(r_i, r_j) - \lambda_1 \left(\sum_{i=1}^n \omega_i \bar{r}_i - E(R) \right) - \lambda_2 \left(\sum_{i=1}^n \omega_i - 1 \right) - \beta^T \omega$$

其中, β 为 KT 乘子, 对此求解可得

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \omega_i} = \sum_{j=1}^n \omega_j Cov(r_i, r_j) - \lambda_1 \bar{r}_i - \lambda_2 1_n - \beta = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_1} = \sum_{i=1}^n \omega_i \bar{r}_i - E(R) = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_2} = \sum_{i=1}^n \omega_i - 1 = 0 \\ \beta_i \omega_i = 0 \\ \beta_i \geq 0 \end{cases} \quad (8.6)$$

8.1.6 加入无风险资产后的 MVO – Capital Market Line

我们发现最优的资产组合是基于一个约束条件下的最优解的集合, 但是当我们加入了无风险资产以后就会出现一个最优解

$$R_f = (1 - \alpha)R_{riskfree} + (\alpha)R_{port}$$

因此,

$$\mathbb{E}(R_f) = (1 - \alpha)r + \alpha\mu(\omega) = r + \alpha(\mu(\omega) - r)$$

, 求得方差为

$$\sigma^2 = \alpha^2 \sigma_\omega^2$$

我们可以推出 $\alpha = \frac{\sigma_f}{\sigma_\omega}$, 带回之前的公式可知,

$$\mathbb{E}(R_f) = r + \frac{\mu(\omega) - r}{\sigma_\omega} \sigma_f$$

我们进一步定义夏普比例 $SR(\omega|r) = \frac{\mu(\omega) - r}{\sigma_\omega}$, 因此我们可以画出 \mathbb{E}/σ 的二维几何解, 其中, r 为该图形的截距项, 而 SR 比率则为斜率, 我们称这个二维图像为 CML(Capital Market Line)

8.1.7 CAPM 模型市场均衡与 tangency portfolio

我们定义 CML 与有效前沿的切点为 tangency portfolio, 在这个点我们认为是最优化下的组合配置. 1964 年威廉夏普 (William Sharpe) 发表了资本资产定价模型 (capital asset pricing model(CAPM)). 我们进行进一步的阐述. 首先我们令 ω^* 为 tangency portfolio, 在有效前沿上, 我们可以重新表述这个点的意义.

$$\mathbb{E}(R_f) = r + \frac{\mu(\omega^*) - r}{\sigma_{\omega^*}} \sigma_f$$

我们可以基于 tangency portfolio 的极值点推广这个 capm 模型到空间上的任意一点 i , 我们假定一个新的投资组合 Z_i , 其中, 他的投资范围是 w_i 投资与其他资产 i , 剩下的投资于 $1 - w_i$ tangency portfolio ω^* . 我们可以直接写出他的收益期望以及 portfolio z 的方差表达式:

$$\mathbb{E}(Z_i) = w_i \mu_i + (1 - w_i) \mu(\omega^*)$$

方差为

$$\sigma_i^2 = w_i^2 \sigma_i^2 + (1 - w_i)^2 \sigma_{\omega^*}^2 + 2w_i(1 - w_i) \text{cov}(i, \omega^*) \sigma_i \sigma_{\omega^*}$$

我们对于上下两个式子分别求偏导, 并合并起来则有

$$\frac{\frac{\partial \mathbb{E}}{\partial \omega}}{\frac{\partial \sigma}{\partial \omega}} = \frac{\mu_i - \mu(\omega^*)}{(\omega \sigma_i^2 + (\omega - 1) \sigma^2(\omega^*) + (1 - 2\omega) \rho(e_i, \omega^*) \sigma_i \sigma_{\omega^*}) \sigma_z^{-1}}$$

非常容易知道当 $w=0$ 的时候, 此处的表达式为 tangency portfolio 的点的值

$$\sigma^* \frac{\mu_i - \mu(\omega^*)}{(\omega \sigma_i^2 + (\omega - 1) \sigma^2(\omega^*) + (1 - 2\omega) \rho(e_i, \omega^*) \sigma_i \sigma_{\omega^*}) \sigma_z^{-1}} = \frac{(\mu_i - \mu(\omega^*)) \sigma}{0 * \sigma_i^2 + (0 - 1) \sigma^2(\omega^*) + (1 - 2 * 0) \rho(e_i, \omega^*) \sigma_i \sigma_{\omega^*}}$$

整理可知

$$\frac{(\mu_i - \mu(\omega^*)) \sigma}{\rho(e_i, \omega^*) \sigma_i \sigma_{\omega^*} - \sigma^2(\omega^*)} = \frac{\mu(\omega^*) - r}{\sigma(\omega^*)} = \beta_i = \frac{\text{cov}(R_i, R_{\omega^*})}{\text{var}(R(\omega^*))}$$

因此我们就推导出了 CAPM 公式的一般表达式

$$R_i = \alpha_i + \beta_i R_t(x) + \epsilon_{i,t}$$

在得到了一般化的表达式后, 我们也可以从数据出发反向基于 OLS 对于 CAPM 的定理进行拟合, 此处的 ϵ 是一个高斯白噪音的残差项 (在本 CAPM 的推导过程中是残差项) 我们对于 β 进行进一步的讨论, 我们开始研究当引入 $E(R)/\sigma$ 曲线上的任意一点 b 后, b 的 beta 于 tangency portfolio 的 beta 的相关性, 我们可以进行如下的推导:

$$\beta_{x|b} = \frac{\text{cov}(R(\omega^*), R(b))}{\text{var}(R(b))} = \frac{\text{cov}(\omega^T R, b^T R)}{\text{var}(b^T R)}$$

基于二次型知识的拆解, 我们可以快速的把以上的式子变成

$$\beta_{x|b} = \frac{\omega^T \Sigma b}{b^T \Sigma b} = \omega^T \frac{\Sigma b}{b^T \Sigma b} = \omega^T \beta_b = \sum_i \omega_i \beta_i$$

我们可以知道 beta 的相关性可以被表达为带权 beta 的线性叠加. 同时这个任意一点的 b 点我们也可以理解为现实投资世界的一个实际的后验组合的位置, 我们可以计算他的跟踪误差 TR

8.1.8 TR tracking error

关于跟踪误差的问题, 我们还是以上一节的讨论继续, 我们可以知道 b 点的投资组合与 tangency portfolio 点的偏离期望为: $e = E(\omega^*) - E(b) = (\omega - b)^T R$, 我们基于以往的讨论, 可以迅速的写出跟踪误差的期望和方差

$$\sigma(\omega|b) = \sqrt{(\omega - b)^T \Sigma (\omega - b)}$$

8.2 APT 套利定价

8.3 SRM 结构化风险模型

我们首先发现, 对于大样本而言, 协方差矩阵的估计是一个非常困难的问题, 同时, 基于 APT, 我们也可以发现, 如果我们可以抽离出底层因子的风险, 我们可以使用因子的收益率来一定程度的估计和替换资产的收益. 因此, 我们首先改写收益率的表达模型 $r = Xb + u$, 其中 r 为超额收益率, X 为因子暴露度矩阵 (factor exposure/ factor loading), b 为因子的收益率矩阵, 而 u 为特异风险矩阵 (残差收益率矩阵), 我们有以下两个假设

1. 特异收益率 u 与因子收益率 b 不相关 (残差收益)
2. 残差收益内部两两不相关,

由此我们可以推出, 对整个式子做协方差估计可知:

$$\Sigma = X * F * X^T + \Delta$$

其中, F 为因子收益率矩阵的协方差, 而 Δ 为特异因子收益的协方差矩阵, 根据假设 2 我们可以知道 Δ 是一个对角矩阵.

我们用 (X, F, Δ) 来表示一个结构化的因子模型, 如果超额收益率 f 可以被这个因子模型的因子暴露 X 线性表达, 我们称改因子模型解释了超额预期收益率, 及

$$f = X * m$$

在结构化模型中,所有的因子必须是先验的(及可以在考察期初就可以确定),我们将因子分为三个大类:对外部响应的因子,资产属性的横截面比较因子,以及纯粹的内在统计因子我们分别进行讨论

8.3.1 对外部变化的响应

债券市场收益率(债券 beta),通胀异动,油价变动,汇率变动,工业产能 -> 宏观因子

8.3.2 横截面比较因子

横截面比较因子比较股票之间的各种属性,于股票之外的经济没有任何直接的联系,横截面属性的因子主要可以分为两类:基本面类和市场类

基本面类因子一般会比较各种比率如:分红比率,盈利率,以及分析师对于各个股票的预测
市场类因子主要是包括对于市场过去一段时间的波动率,收益率,换手率等等

8.3.3 统计因子

统计性质的因子可以基于股票的基本 k 线或者 tick 级别的数据进行统计学性质的分析和生成

8.3.4 因子组合

8.3.5 因子协方差矩阵

8.4 BARRA 多因子框架

BARRA 框架的本质是通过对于收益率因子的估计来降低因子协方差矩阵的大小,通常我们研究因子,我们需要从以下 6 个大的方面对于因子的收益进行评估和测试,及逻辑性,持续性,信息增量性,稳健性,可投资性和普适性

通过回归法研究因子风险敞口,以之表明风险因子对投资组合业绩的贡献程度,其间接体现了'利用风险因子进行投资'的可行性.本处假设以当期单因子为解释变量、下期个股收益率为被解释变量,构建一元线性回归方程如下:

$$r^{T+1} = a + by_j^T + \varepsilon = \hat{r}^{T+1}$$

其中系数 b 代表了 T 期 j 因子 y_j^T 的风险敞口,含义是因子收益率对于投资收益率的斜率,风险因子敞口 b 越大,则说明下期投资组合从该因子中取得的收益率越大

8.4.1 因子的逻辑性

有时候,基于数理统计和暴力穷举,我们也能发现一些基于历史数据来说看上去能够预测收益率的变量,但是他们可能是因为特定历史的时间窗口对于正确的风险因素进行了凑巧的正暴露而已,并不能保证在未来依旧持续的获得超额收益的能力,我们称其为在样本外的过拟合.因此在对于因子的处理中,我们首先要对于逻辑性进行分析,主要我们可以将其分

解为风险补偿类和错误定价类。风险补偿类因子主要是通过主动暴露并承担一些尾部风险来获得平常时间的风险溢价，而错误定价一般则是通过对于行为金融学的研究和分析，基于市场上的错误定价和交易的行为修正的过程进行捕捉而获利。

8.4.2 因子持续性

在通过理论上的逻辑合理判定之后，我们还需要对于因子本身的实证结果进行分析，这是因为在不同的市场，不同的时期，不同的宏观政策下，市场中的实际的投资者构成不同导致一些在 A 市场行之有效的因子在 b 市场出现水土不服的情况。目前我们主要是对于 IC 进行分析和研究

$$\begin{aligned}
 IC = \text{corr}(r^{T+1}, y^T) &= \text{corr}(r^{T+1}, \hat{r}^{T+1}) = \frac{\text{cov}(r^{T+1}, \hat{r}^{T+1})}{\sigma_{r^{T+1}} \sigma_{\hat{r}^{T+1}}} \\
 &= \frac{\sum (r^{T+1} - \bar{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})}{\sqrt{\sum (r^{T+1} - \bar{r}^{T+1})^2 \sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2}} \\
 &= \frac{\sum (r^{T+1} - \hat{r}^{T+1} + \hat{r}^{T+1} - \bar{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})}{\sqrt{\sum (r^{T+1} - \bar{r}^{T+1})^2 \sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2}} \\
 &= \frac{\sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2 + 0}{\sqrt{\sum (r^{T+1} - \bar{r}^{T+1})^2 \sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2}} \\
 &= \sqrt{\frac{\sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2}{\sum (r^{T+1} - \bar{r}^{T+1})^2}} = \sqrt{1 - \frac{RSS}{TSS}} = \sqrt{R^2}
 \end{aligned} \tag{8.7}$$

关于 IC 和 R^2 的推导中

$$\begin{aligned}
 &\sum (r^{T+1} - \hat{r}^{T+1} + \hat{r}^{T+1} - \bar{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1}) \\
 &= \sum (r^{T+1} - \hat{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1}) + (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1}) \\
 &= \sum (\hat{r}^{T+1} - \bar{\hat{r}}^{T+1})^2 + 0 \\
 &\sum (r^{T+1} - \hat{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1}) = \sum (r^{T+1} - \beta_0 - \beta_1 x_i)(\beta_0 + \beta_1 x_i - \bar{r}^{T+1}) \\
 &= (\beta_0 - \bar{r}^{T+1}) \sum (r^{T+1} - \beta_0 - \beta_1 x_i) + \beta_1 \sum (r^{T+1} - \beta_0 - \beta_1 x_i) x_i
 \end{aligned} \tag{8.8}$$

因为在 R^2 的回归过程中，SSE 为最小二乘法 OLS 最小残差平方和，我们希望 SSE 最小，因此

$$SSE = \sum (e_i)^2 = \sum (r^{T+1} - \hat{r}^{T+1})^2 = \sum (r^{T+1} - \beta_0 - \beta_1 x_i)^2$$

，我们希望 SSE 最小，因此对 SSE 求偏导，有

$$\frac{\partial SSE}{\partial \beta_0} = \sum -2(r^{T+1} - \beta_0 - \beta_1 x_i) = 0$$

以及

$$\frac{\partial SSE}{\partial \beta_1} = \sum -2x_i(r^{T+1} - \beta_0 - \beta_1 x_i) = 0$$

因此，

$$\sum (r^{T+1} - \hat{r}^{T+1})(\hat{r}^{T+1} - \bar{\hat{r}}^{T+1}) = \sum (r^{T+1} - \beta_0 - \beta_1 x_i)(\beta_0 + \beta_1 x_i - \bar{r}^{T+1}) = 0$$

研究当期因子暴露与下期个股收益率的相关性，即因子信息系数 (IC)，因此 IC 在一定程度上体现了当期个股因子对下期个股收益率的显著程度，因子 IC 绝对值越高，表明投资

组合下期的收益越能被当期因子所解释。

相关系数要求样本严格服从正态分布,但是在投资实务中 IC 受因子极端值影响导致误差较大,效果往往不太理想。相比之下,斯皮尔曼系数 (RankIC) 研究的是变量的序列相关关系,强调随机变量之间的‘单调相关性’。本文通过研究当期因子暴露与下期个股收益率序列的 RankIC 进行有效因子识别,具体计算方法如下

$$\text{RankIC}_i = \text{corr}(\text{rank}(y_i^{T-1}), \text{rank}(r^T)) \quad (8.9)$$

其中, $\text{rank}(y_i^{T-1})$ 为 t-1 期因子暴露 i 的排序, $\text{rank}(r^T)$ 为 t 期个股收益率的排序。一般来说,基于日频的计算,ic 值在 0.02 以上即为优秀的预测,除了对于 ic 的研究,我们主要使用 IC, IR(预测收益能力的稳定性), IC 值大于 0.02 的比例, t 值来对于 ic 进行测试。在投资组合绩效评价中,信息比率 (IR) 用来衡量投资组合在单位风险下的超额收益能力。对于因子的稳定性,这里采用 ICIR 来替代策略 IR,即 RankIC 的均值与标准差的比值

$$\text{ICIR} = \frac{\text{RANKIC}}{\text{stdRANKIC}}$$

因子协方差矩阵估计: Newey-West 自相关调整、特征值调整、波动率偏误调整
 特异风险矩阵估计: Newey-West 自相关调整、结构化模型调整、贝叶斯收缩调整、波动率偏误调整

8.4.3 因子显著性

因子敞口值的大小并不能体现因子对下期投资组合收益率的显著关系,即有效但不一定显著。一般采用 t 值方法来对因子风险敞口进行显著性检验, t 值越大说明因子对收益率解释更有效。对单因子模型来说, t 值与拟合优度 R² 均能反映因子的显著性,而相关系数 corr 与拟合优度又存在数量关系,因此可以用相关系数来检验因子的显著性

8.4.4 因子的信息增量性

当我们进行因子的分析和挖掘以后,我们必然面临大量已知因子的管理和新挖的因子和处理问题,这时候我们需要对于最新的因子进行收益的分析和归因 (VIF 测试),来判断这个新的因子是不是能够被已知的因子所解释,同时,因子的正交化也是一个较为优秀的方法

8.4.5 因子稳健性

因子的稳健性主要是构造因子的方法是否对于参数过于敏感,以及在不同的实证区间内是否一致,在不改变预测变量逻辑的情况下,如果重新定义构造的方法或者改变区间,是否依然能够显著的预测收益率。

8.4.6 因子的可投资性

8.4.7 普适性

8.5 组合风险预算 Risk Budgeting

我们假定有如下的两个资产组成的资产组合

$$R_p = x_1 R_1 + x_2 R_2$$

$$\sigma_p^2 = x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}$$

$$\sigma_p = \sqrt{x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}}$$

我们首先分析最简单的情况 $\sigma_{12} = 0$, 及两个资产完全不相关

$$\begin{aligned} \sigma_p^2 &= x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 \rightarrow \\ x_1^2 \sigma_1^2 &\rightarrow 1 \\ x_2^2 \sigma_2^2 &\rightarrow 2 \\ \frac{x_1^2 \sigma_1^2}{\sigma_p^2} &\rightarrow 1 \\ \frac{x_2^2 \sigma_2^2}{\sigma_p^2} &\rightarrow 2 \\ \sigma_p &= \frac{\sigma_p^2}{\sigma_p} = \frac{x_1^2 \sigma_1^2}{\sigma_p} + \frac{x_2^2 \sigma_2^2}{\sigma_p} \\ \frac{x_1^2 \sigma_1^2}{\sigma_p} &\rightarrow 1 \\ \frac{x_2^2 \sigma_2^2}{\sigma_p} &\rightarrow 2 \end{aligned} \quad (8.10)$$

我们再分析 $\sigma_{12} \neq 0$, 及两个资产部分相关

$$\begin{aligned} \sigma_p^2 &= (x_1^2 \sigma_1^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}) + (x_2^2 \sigma_2^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}) \rightarrow \text{组合风险} \\ x_1^2 \sigma_1^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12} &\rightarrow \text{资产 1 的方差贡献} \\ x_2^2 \sigma_2^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12} &\rightarrow \text{资产 2 的方差贡献} \\ \frac{x_1^2 \sigma_1^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}}{\sigma_p} &\rightarrow \text{资产 1 的单位风险贡献} \\ \frac{x_2^2 \sigma_2^2 + x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}}{\sigma_p} &\rightarrow \text{资产 2 的单位风险贡献} \end{aligned} \quad (8.11)$$

在基础知识中我们已经知道, 齐次函数具备欧拉定理, 可以有效的将函数分解成各个因素的影响之和

$$\sigma_p = f(x_1, x_2) = \sqrt{x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}}$$

$$f(cx_1, cx_2) = \sqrt{c^2 x_1^2 \sigma_1^2 + c^2 x_2^2 \sigma_2^2 + c^2 2x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}} = c \sqrt{x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_1 \sigma_2 \sigma_{12}} = cf(x_1, x_2)$$

我们推广到一般性质上, 对于 n 个资产的 x

$$\begin{aligned}
 x &= (x_1, x_2, \dots, x_n)^T \\
 R &= (r_1, r_2, \dots, r_n)^T \\
 E(R) &= \mu, \text{cov}(R) = \Sigma \\
 R_p &= R_p(x) = x^T R \\
 \mu_p &= \mu(x) = x^T \mu \\
 \sigma_p^2 &= \sigma_p^2(x) = x^T \Sigma x \\
 \sigma_p &= \sqrt{\sigma_p^2(x)} = \sqrt{x^T \Sigma x}
 \end{aligned} \tag{8.12}$$

$$\frac{\partial f(x)}{\partial x_1} = \frac{\partial \sqrt{x^T \Sigma x}}{\partial x} = 1/2 \sqrt{x^T \Sigma x} \frac{\partial \sqrt{x^T \Sigma x}}{\partial x} = 1/2 \frac{2 \Sigma x}{\sqrt{x^T \Sigma x}} = \frac{\Sigma x}{\sqrt{x^T \Sigma x}} = \frac{\Sigma x}{\sigma_p(x)} \tag{8.13}$$

由欧拉公式我们可知, $\sum_{\alpha=1}^s \frac{\partial f(x)}{\partial x_\alpha} = n f(x)$

$$\begin{aligned}
 f(x) &= x^T \frac{\partial f(x)}{\partial x} = x^T \sqrt{x^T \Sigma x} = \sqrt{x^T \Sigma x} x^T x = \sqrt{x^T \Sigma x} x^T x \\
 RM_p(x) &= \sigma_p(x) = \sqrt{x^T \Sigma x} \\
 RM_p(x) &= x_1 \frac{\partial RM_p(x)}{\partial x_1} + x_2 \frac{\partial RM_p(x)}{\partial x_2} + \dots + x_n \frac{\partial RM_p(x)}{\partial x_n} = \sum_{i=1}^n x_i \frac{\partial RM_p(x)}{\partial x_i} = x^T \frac{\partial RM_p(x)}{\partial x} \\
 \frac{\partial RM_p(x)}{\partial x} &= \frac{\Sigma x}{\sigma_p(x)} = MCR_i^\sigma
 \end{aligned} \tag{8.14}$$

又

$$\begin{aligned}
 MCR_i^{RM} &= \frac{\partial RM_p(x)}{\partial x_i} \rightarrow \text{Marginal Contribution to Risk} \\
 CR_i^{RM} &= x_i * MCR_i^{RM} \rightarrow \text{Contribution to Risk of asset } i \\
 RM_p(x) &= x_1 * MCR_1^{RM} + x_2 * MCR_2^{RM} + \dots + x_n * MCR_n^{RM} = CR_1^{RM} + CR_2^{RM} + \dots + CR_n^{RM} \\
 P CR_i^{RM} &= \frac{CR_i^{RM}}{RM_p(x)} \rightarrow \text{Percent Contribution to Risk of asset } i
 \end{aligned} \tag{8.15}$$

我们试图将资产组合的风险贡献关系看成连续函数, 则:

$$MCR = \frac{\partial \sigma_p(x)}{\partial x} = \frac{\Delta \sigma_p(x)}{\Delta x}$$

同时又有

$$\sum_{i=1}^n x_i = 1$$

的要求, 因此对于组合内的 x_i, x_j , 有 $\Delta x_i = -\Delta x_j$

故, 我们可以推导出

$$\begin{aligned}
 \Delta \sigma_p &\approx \sum MCR_i^\sigma \Delta x_i \\
 \Delta \sigma_p &\approx MCR_i^\sigma \Delta x_i + MCR_j^\sigma \Delta x_j \\
 &= MCR_i^\sigma \Delta x_i - MCR_j^\sigma \Delta x_i = (MCR_i^\sigma - MCR_j^\sigma) \Delta x_i
 \end{aligned} \tag{8.16}$$

9 组合管理优化的一些必要注意

9.1 stability issues

9.2 Resampling techniques

9.3 Denoising and Detoning

9.3.1 Constant residual eigenvalue

9.3.2 target shrinkage

9.4 协方差收缩 Shrinking methodss

9.4.1 基本协方差收缩

9.4.2 Ledoit-Wolf 收缩

```
from sklearn.covariance import LedoitWolf
>>> import numpy as np
>>> from sklearn.covariance import LedoitWolf
>>> real_cov = np.array([[.4, .2],
...                      [.2, .8]])
>>> np.random.seed(0)
>>> X = np.random.multivariate_normal(mean=[0, 0],
...                                   cov=real_cov,
...                                   size=50)
>>> cov = LedoitWolf().fit(X)
>>> cov.covariance_
array([[0.4406..., 0.1616...],
       [0.1616..., 0.8022...]])
>>> cov.location_
array([ 0.0595..., -0.0075...])
```

9.4.3 Oracle 近似收缩

10 主动组合管理

10.1 资产配置

整理以上公式, 我们发现, 当风险的关系满足齐次等式时, 基于欧拉定理我们可以将组合的风险分解到组合内的每一个策略单元 i 上, 我们定义 b_i 为第 i 个策略单元的风险贡献在组合风险 $R(x)$ 中所占的百分比, 则我们可以对于组合管理中的权重分配进行线性规划的划分:

$$RC_i = b_i R(x) s.t = \begin{cases} b_i \geq 0 \\ 1 \geq x_i \geq 0 \\ \sum_{i=1}^n b_i = 1 \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

基于拉格朗日乘数法, 我们可以定义目标函数为各资产的风险贡献的误差平方和最小化

$$\min_x f(x; b) = \sum_{i=1}^n (x_i \frac{\partial R(x)}{\partial x_i} - b_i R(x))^2 s.t = \begin{cases} 1 \geq x_i \geq 0 \\ \sum_{i=1}^n x_i = 1 \end{cases} \quad (10.1)$$

10.2 逆方差优化 IVP Inverse Variance

在统计学上, 逆波动率加权是一种对随机变量测量值进行加权平均的方法, 每个随机变量被其方差的倒数加权. 该方法可使平均值的方差最小.

若随机变量的一系列独立测量值为 y_i , 其方差为 σ_i^2 , 则这些测量值的逆方差加权平均为

$$\hat{y} = \frac{\sum_i \frac{y_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

在所有加权平均方法中, 逆方差加权平均的方差最小, 为

$$D^2(\hat{y}) = \frac{1}{\sum_i \frac{1}{\sigma_i^2}}$$

若各测量值的方差相等, 则逆方差加权平均与简单平均相同.

逆方差加权通常在元分析中用来整合独立测量的结果.

10.3 最大多元化投资目标 Maximum Diversification

10.4 AAP Agnostic Allocation Portfolio

在 AAP 中, 作者提出一个较为统一的研究框架,

$$w_{a,b,c}^t = \omega \Sigma^{-a} \sigma^b M^c \mathbf{1}_n$$

其中, M 为 marketcap 及市值矩阵, 对于目标组合的最优权重则划归一个统一的结构:

$$a = b = 0, c = 1 > MCportfolio$$

$$a = b = c = 0 > equalweightportfolio$$

$$a = c = 0, b = -1 > equalvolatilityportfolio$$

$$a = 1, b = c = 0 > mean - varianceportfolio$$

$$c = 0, a = b = 1 > mostdiversificationportfolio$$

- a: 协方差敏感参数 $a > 0$ 组合不关心协方差 $a > 1$ 组合更加关注协方差组合
- b: 波动率敏感参数 b 正相关于波动率的回报 (b 越大越考虑波动率的暴露)
- c: 市值敏感参数 $c > 0$ 不考虑市值的暴露, $c > 1$ 等市值组合

我们之前讨论过协方差矩阵可逆且正定, 因此我们可以对于协方差矩阵进行 EVD 分解, 通过拆分协方差矩阵的特征根和特征向量, 我们可以得到

$$\Sigma = \sum_k \lambda_k u_k u_k^T$$

, 其中 λ 为特征根, u 为特征性向量. 我们再代入之前的式子, 并令 $p = \sigma^b M_i^c$, 则最优权重可以重新表达为

$$w = \omega \sum_k \lambda_k^{-a} (u_k p) (u_k^T)$$

, 则根据风险和二次型的定义, 我们知道该组合权重下的方差 R^2 为

$$R^2 = w^T \Sigma w = \omega^2 \sum_k \lambda_k^{1-2a} (u_k p)^2$$

我们可以发现, 当 $a < 0.5$ 时, 方差与特征根成正向关系, 而 $a > 0.5$ 时, 方差与特征根呈负向关系, 因此我们认为当 $a = 0.5$ 的时候, 此时组合对于风险是无偏估计, 此处则推导出 AAP 的约束: $a = 0.5, b = c = 0$,

$$w_{ERP} = \omega \Sigma^{-0.5} Q^{-0.5} p$$

, 其中 Q 为 p 的协方差矩阵

10.5 Black-Litterman 基础理论

Black-Litterman 模型是基于 MPT 基础上的资产配置理论 BL 模型在隐含市场收益率和分析师主观预测信息的基础上, 成功解决了 MPT 模型中假设条件不成立, 参数敏感等问题. 根据资产配置理论的标准假设, 如 Markowitz(1952) 在《Portfolio Selection》一文所述, 假设市场中资产种类为 n , 则投资者的效用函数为:

$$U = \omega' \mu - 1/2 \lambda \omega' \Sigma \omega$$

其中:

ω : n 类资产的权重矩阵, $n \times 1$;

μ : n 类资产的市场回报率矩阵, $n \times 1$;

λ : 投资者风险厌恶因子, 采用 Idzorek(2002) 给出的公式计算

$$\lambda = E(r_{mkt} - r_f) / \sigma_{mkt}^2$$

r_{mkt}, σ_{mkt}^2 为市场收益率及市场收益方差, r_f 为无风险利率

Σ : n 类资产收益的协方差矩阵, 一般用历史收益求得, $n \times n$

10.6 TAA

10.7 SAA

10.8 HRP 分层风险平价

风险平价是构建多样化和均衡投资组合十分流行选择。众所周知,大多数资产类别的未来表现很难预测。通过仅使用资产的风险特征和相关矩阵构建投资组合,风险平价方法克服了这一缺点。Lohre, Rother 和 Schafer 三位作者在经典风险平价基础上,提出了分层风险平价。他们的方法是:利用图论和机器学习来构建投资领域的层次结构。这种结构可以更好地将资产/因子划分为具有相似特征的集群,无需依赖经典的相关性分析。

10.8.1 聚类分层

给定一组资产类别和风格因子收益,相应的算法将根据一定的距离度量对这些资产类别和风格因子进行聚类,然后沿着这些聚类分配相等的风险预算。这种集群可能被认为是比聚合风险因子更自然的组成部分,因为它们自动选择依赖结构,并形成有意义的成分,以帮助投资组合的差异化。论文的贡献在于深入研究了分层聚类在多资产多因子投资环境中的应用和优点。特别是,它将把此方法与其他方法进行对比,比如 1=N, 最小方差, 标准风险平价和多样化风险平价。一个主要的创新是研究基于尾部相关聚类的 HRP 策略,而不是标准的基于相关性聚类。考虑到某些样式因子的尾部风险升高,这种方法可能特别相关。分层风险平价策略通常基于两个步骤:首先,分层聚类算法揭示了所考虑的投资领域的层次结构,从而得到基于树的表示。其次,投资组合权重是通过沿着层次结构应用分配策略得出的。

10.8.2 案例分析

分层风险平价案例

我们再一次回顾一下分层风险平价:它是一种执行资产配置的方法,不需要反协方差矩阵。它本质上是采用标准的层次聚类算法,计算出一个层次树,然后对不同的聚类进行多样化处理。我们将按如下方式进行

- 1, 建立了一个人工相关矩阵 C 与几个层次集群
- 2, 把 C 转换成协方差矩阵
- 3, 从正态分布 $N(0, \text{Sigma})$ 中抽样时间序列
- 4, 对这些时间序列应用分层风险平价

11 从风险补偿到期权定价

11.1 风险补偿

11.2 期权定价

11.2.1 波动率计算的六种方法

- 1 Realized
- 2 Parkinson
- 3 Garman-Klass
- 4 Roger-Satchell
- 5 Garman-Klass -Yang-Zhang
- 6 Yang-Zhang

11.2.2 Realized Volatility: Close/Close

$$\sigma_{realized} = \sqrt{\frac{N}{n-2} \sum_{i=1}^{n-1} (r_t - \bar{r})^2}$$

$r_t = \log \frac{C_t}{C_{t-1}}$: 收益率

$\bar{r} = \frac{1}{n} \sum_{t=1}^n r_t$: 平均收益率

11.2.3 Parkinson Volatility: High/Low

$$\sigma_{parkinson} = \sqrt{\frac{1}{4 * \ln 2} \times \frac{252}{n} \times \sum_{i=1}^{n-1} \ln\left(\frac{H_t}{L_t}\right)^2}$$

11.2.4 Garman-Klass Volatility: OHLC

$$\sigma_{garman-klass} = \sqrt{\frac{N}{n} \times \sum_{i=1}^N \left[\frac{1}{2} \times \left(\log \frac{H_i}{L_i} \right)^2 - (2 \times \log 2 - 1) \times \left(\log \frac{C_i}{O_i} \right)^2 \right]}$$

11.2.5 Roger-Satchell Volatility: OHLC

$$\sigma_{roger-satchell} = \sqrt{\frac{N}{n} \times \sum_{i=1}^N \left[\log \frac{H_i}{L_i} \times \log \frac{H_i}{O_i} + \log \frac{HL_i}{L_i} \right]}$$

12 过拟合问题

12.1 CSCV

12.2 GAN

12.3 PBO

PBO (Probability of Backtest Overfitting) 是定量衡量回测过拟合风险的指标, 计算方式基于 Bailey、Borwein、López de Prado 和 Zhu 在 2017 年提出的组合对称交叉验证 (Combinatorially-Symmetric Cross-Validation, 简记为 CSCV) 框架。假设以夏普比率 (Sharpe Ratio, 简记为 SR) 作为框架中的策略评价指标, 那么 PBO 可按如下方式定义:

$$PBO = P[SR_{n^*} < ME(SR)]$$

* 其中, SR 表示“测试集”各组策略的夏普比率, n^* 表示“训练集”表现最好 (夏普比率最高) 的那组策略, ME 表示中位数。注意到这里的“交叉验证”、“训练集”和“测试集”并不完全等价于机器学习传统意义上的相关概念, 但是有异曲同工之处 * 上述定义的含义是: “训练集”夏普比率最高的策略 n^* , 在“测试集”的夏普比率也应该较高, 表现至少应优于一半的策略。如果策略 n^* 的测试集夏普比率排名在后 50, 那么很有可能属于回测过拟合。回测过拟合的概率, 即为最优策略 n^* 的测试集夏普比率排名位于后 50 的概率。PBO 的定义引申出新的问题: 对于量化策略, 尤其是非机器学习策略, 通常不存在“训练集”和“测试集”的概念。PBO 是如何根据回测结果划分“训练集”和“测试集”呢? 下面我们展示 PBO 的计算步骤:

- 1 构建矩阵; 每列分别表示第 N 组策略下 T 期的收益率序列。
- 2 按行切割矩阵, 得到子矩阵, $= 1, 2, \dots, S$; 需要注意的是, 这里 S 必须为偶数, 此时每个子矩阵维度相同, 均为 $\frac{T}{S} * N$
- 3 从 S 个子矩阵中任意选出 $\frac{S}{2}$ 个为一组, 用 C_s 表示所有可能的组合; 根据组合原理, 这样的组合共有 $C_S^{\frac{S}{2}}$ 种。
- 4 对于 C_s 中的任意一组 c, 进行如下操作:
 - a. 构建训练集: 将 c 中的 $\frac{S}{2}$ 个子矩阵 M 按行拼接起来。
 - b. 构建测试集 \bar{J} : 即 J 的补集, 将不包含在 c 中的子矩阵按行拼接起来。
 - c. 对于训练集, 计算每列的夏普比率, 得到夏普比率最高的策略策略。
 - d. 对于测试集 \bar{J} , 得到策略 n^* 在测试集夏普比率 SR_{n^*} 的绝对排名 $Rank(n^*)$ 和相对排名 ω (均为降序排列); 通常取 $\omega = \frac{Rank(n^*)}{N+1}$, $\omega \in (0, 1)$
 - e. 定义对数几率 $\lambda = \log(\frac{\omega}{1-\omega})$: λ 随 ω 增大而增大, 当 $\omega = 0.5$ 时, $\lambda = 0$; 当 ω 接近 1 时, λ 取无穷大。
5. 根据第 4 步, 对于 C_s 中的任意一组 c, 可计算 λ 。对全部 c 进行遍历, 最终 $\lambda_m, m = 1, 2, 3, \dots, C_S^{\frac{S}{2}}$ 。进一步可得到 λ 的经验分布 $f(\lambda)$, PBO 是 $f(\lambda)$ 在区间 $(-\infty, 0]$ 上的定积分:

$$PBO = \int_{-\infty}^0 f(\lambda) d\lambda$$

需要说明的是, 上述定积分的前提假设是 S 取无穷大, 此时相对排名 ω 和对数几率 λ 为离散变量。根据离散变量经验分布的定义:

$$f(\lambda) = \sum_{i=1}^n I_{\lambda_i < \lambda}$$

有如下推导:

$$PB = \int_{-\infty}^0 f(\lambda) d\lambda = \frac{\#\{\lambda_m < 0\}}{C_S^{S/2}} = \frac{\#\{\omega_m < 0.5\}}{C_S^{S/2}}$$

其中 $\#\{\lambda_m > 0\}$ 表示 $C_S^{S/2}$ 中大于 0 的个数, $\#\{\lambda_m > 0.5\}$ 则表示 $C_S^{S/2}$ 各 ω_m 中大于 0.5 的个数。当划分完“训练集”和“测试集”后,** 只需要计算相对排名 ω , 随后统计相对排名 ω 中大于 0.5 的个数即可。**

PBO 的计算框架包含以下优点: 1. 保证“训练集”和“测试集”样本量相同, 使得夏普比率具有可比性; 2. 各组策略关系对等, 排除其它影响夏普比率因素的干扰; 3. 划分数据时将回测时间 T 划分为 S 个子集, 每个子集内部保留原始时序。4. 该框架为非参模型, 无需过多假设。5. 具备灵活性, 可以根据实际情况将夏普比率换成其它策略评价指标。