结合领域知识的因子分析: 在金融风险模型上的应用

冯 栩 [1] 喻文健 [1] 李 凌 [2]

摘 要 因子分析是一种在工业领域广泛使用的统计学方法. 在金融资产管理中, 因子分析通过对历史价格波动的极大似然估计推导自适应的统计学因子来生成风险模型. 与通过使用预先设定具有经济学含义的因子来生成风险模型的基本面因子模型相比, 通过因子分析生成的模型不仅更灵活, 还能发现在基本面模型中缺失的因子. 然而, 由于因子分析所生成模型中的统计学因子缺少可解释性, 因此当金融数据中存在显著噪音时容易过拟合. 针对中国股市数据的风险模型生成问题, 本文提出快速因子分析算法以及将基本面因子结合到因子分析中的挑选基本面因子的混合因子分析方法, 使风险模型同时在因子探索及模型可解释性上达到最优. 实验结果显示快速因子分析方法能够达到 31 倍以上的加速比, 且新混合因子分析方法能够增大人造数据集以及真实数据集上预测的对数似然估计值. 在真实数据集上, 新方法能最好够达到平均对数似然估计值 12.00, 比因子分析构建模型的 7.56 大 4.44, 并且两个算法均值差值的标准差为 1.58, 表现出新方法能构建更准确的风险模型.

关键词 因子分析, 基本面因子, 领域知识, 风险模型, 期望最大化过程

引用格式 冯栩, 喻文健, 李凌. 结合领域知识的因子分析: 在金融风险模型上的应用. 自动化学报, 2022, **48**(1): 121−132

DOI 10.16383/j.aas.c200342

Combining Domain Knowledge with Statistical Factor Analysis: An Application to Financial Risk Modeling

FENG Xu[1] YU Wen-Jian[1] LI Ling[2]

Abstract Factor analysis is a statistical method widely used in many industrial domains. In financial portfolio management, a statistical risk model can be constructed via factor analysis, decomposing the risk into self-adapting factors and maximizing log-likelihood of the historical price movement. Compared to a fundamental model that uses well defined factors with economical meanings, it is more dynamic and may discover factors that are missed in the fundamental ones. On the other hand, statistical factors lack the intuitive interpretation, and thus are less stable and more prone to overfitting especially for the financial data with remarkable noises. In this work, we propose a fast factor analysis and a hybrid method that incorporates the fundamental factors into the statistical process, achieving an optimized combination of model interpretability and data exploitation. Our experiment results show that the acceleration of fast factor analysis is up to more than 31 times, and the new hybrid method yields improved out-of-sample log-likelihood on both synthetic and real-world data. The best mean of log-likelihood of proposed algorithm on real-world data is 12.00, which is larger than 7.56 of factor analysis with difference 4.44, and the standard deviation of the difference is 1.58. All the results shows the proposed algorithms estimate risk model more accurately.

Key words Factor analysis, fundamental factor, domain knowledge, risk model, expectation-maximization process

Citation Feng Xu, Yu Wen-Jian, Li Ling. Combining domain knowledge with statistical factor analysis: An application to financial risk modeling. *Acta Automatica Sinica*, 2022, **48**(1): 121−132

金融分析中, 风险管理对于合理地保护资产十分重要. 通常, 资产价值的波动性被定义为风险, 而风险管理的目的则是合理的评估资产的波动性[1−7]. 风险模型 (Risk model) 是风险管理的重要方法, 而多因子模型 (Multiple-factor model) 是一种能够有效分析资产风险的风险模型[3, 8]. 多因子模型假设资产的回报被若干因子影响, 例如国家经济水平、工业领域周期以及公司财务指标等等具有经济学含义的基本面因子, 或者通过统计学方法计算出的统计学因子, 其一般形式为

收稿日期 2020-05-22 录用日期 2020-12-31
Manuscript received May 22, 2020; accepted December 31, 2020
国家自然科学基金 (61872206) 资助
Supported by National Natural Science Foundation of China (61872206)
本文责任编委 张军平
Recommended by Associate Editor ZHANG Jun-Ping
1. 清华大学计算机科学与技术系 北京信息科学与技术国家研究中心 北京 100084 2. 善流投资管理有限公司 上海 200000
1. Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084 2. Flow Assets Management, Shanghai 200000

$$y_i = \sum_{j=1}^{k} c_{ij} x_j + r_i \tag{1}$$

其中, y_i 是第 i 个资产的回报 ($i = 1, \cdots, m$); x_j 表示第 j 个因子的数值 ($j = 1, \cdots, k$); c_{ij} 表示第 j 个因子对于第 i 个资产影响程度, 被称为第 i 个资产在第 j 个因子上的暴露; 而 r_i 表示第 i 个资产的非因子回报, 通常被看做拟合残差. 式 (1) 显示所有资产回报都被 k 个相同因子驱动, 并且这些因子反映资产间的相关性以及内在的波动性. 每个因子 x_j 和残差 r_i 是不相关的, 且每个残差 r_i 之间也不相关. 通过式 (1), 可以推导出风险的表达式[3]:

$$\begin{aligned} Risk = {} & \mathrm{var}(\boldsymbol{y}) = \\ & \mathrm{var}(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{r}) = \\ & \mathrm{var}(\boldsymbol{C}\boldsymbol{x}) + \mathrm{var}(\boldsymbol{r}) = \\ & \boldsymbol{C}\boldsymbol{X}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R} \end{aligned} \tag{2}$$

其中, $\mathrm{var}(\boldsymbol{y})$ 代表 m 个资产回报 $\boldsymbol{y} = [y_1, \cdots, y_m]^{\mathrm{T}}$ 的协方差, $\boldsymbol{C} \in \mathbf{R}^{m \times k}$ 是 m 个资产对 k 个因子 $\boldsymbol{x} = [x_1, \cdots, x_k]^{\mathrm{T}}$ 的暴露矩阵, $\boldsymbol{X} \in \mathbf{R}^{k \times k}$ 是因子 $\boldsymbol{x}$ 的协方差矩阵, 而对角阵 $\boldsymbol{R} \in \mathbf{R}^{m \times m}$ 是残差矩阵. 风险模型的生成则是通过实际观测到的回报和一些分布假设在限制因子数目的情况下计算出式 (2) 中的 $\boldsymbol{C}$、$\boldsymbol{X}$ 和 $\boldsymbol{R}$ 得到风险矩阵, 再通过风险矩阵来进行资产的选择和配置来规避风险从而最大化回报. 通常情况下, 风险模型需要每隔一天或者更短的时间生成一次, 用于下一时段的资产选择和配置.

基本面因子模型 (Fundamental factor model) 和统计学因子模型 (Statistical factor model) 是两类经典的多因子模型[1−3, 8], 目前仍被广泛应用于金融分析领域. 基本面因子模型使用观测到的领域知识在资产上的暴露, 例如股息率、市盈率、市销率等等, 求出这些领域知识因子 (基本面因子) 的数值和残差来生成风险模型[2−3, 8−10]. 这些已知的基本面因子通常含有确切的经济学含义, 因此得到的模型具有很强的可解释性, 而模型的可解释性是对模型性能的重要保证[11−12]. 文献 [2, 9] 使用最小二乘法生成基本面因子模型进行风险资产评估, 并针对不同的真实数据进行了实验, 显示出基本面因子模型的优良性能. 然而由于能观测到的基本面因子数量有限, 且不是都对生成风险模型有价值, 因此文献 [2, 9] 中方法需要手动挑选合适的基本面因子来生成更好的风险模型. 统计学因子模型则使用因子分析 (Factor analysis) 生成模型[3], 其通常采用期望最大化过程 (Expectation-maximization process) 来计算统计学因子及其暴露[13−18]. 尽管统计学因子模型生成的因子没有确切的经济学含义, 容易在数据噪音较大时过拟合, 却能够捕捉到基本面因子模型中缺失的因子及其暴露. 由于基本面因子模型具有很强的可解释性, 而统计学因子模型可以捕捉到隐藏在回报中的因子, 因此需要构造一个结合领域知识且包含统计学因子的混合因子分析算法用来生成更准确的风险模型.

本文针对中国股市风险评估问题, 将基本面因子暴露整合到统计学因子分析中提出一种新的混合因子分析方法生成中国股票的风险模型. 首先, 本文提出一种快速因子分析算法. 其次, 修改因子分析的期望最大化过程使其包含基本面因子暴露及一个用来调整基本面因子的大小和相关性的方阵, 从而推导出新的混合因子分析算法. 最后, 基于混合因子分析算法提出了近似最优的基本面因子挑选算法, 并将其与混合因子分析算法结合得到挑选基本面因子的混合因子分析算法. 我们使用三个人造数据集和一个真实数据集来测试本文所提出算法的性能, 实验结果表明快速因子分析算法在第一个人造数据集上能够带来 31 倍以上的加速比, 并且本文提出的挑选基本面因子的混合因子分析算法能够有效地构建准确且稳定的风险模型. 在所有的人造数据集上, 使用我们的挑选基本面因子的混合因子分析算法得到的风险模型的对数似然估计值 (Log-likelihood) 的均值都要大于统计学因子分析生成的模型, 并且标准差基本相同甚至更小; 在真实数据集上, 该方法得到平均对数似然估计值为 12.00, 比因子分析构建模型的 7.56 大 4.44, 同时前者的平均对数似然估计值的标准差为 8.25, 小于因子分析的 9.06.

本文的剩余部分按照如下组织: 第 1 节为基于期望最大化过程的统计学因子分析的介绍; 第 2 节介绍我们的混合因子分析算法; 第 3 节为实验结果; 最后一节为全文的总结.

1 期望最大化过程和统计学因子分析

1.1 期望最大化过程

期望最大化过程是机器学习中一个重要的学习方法, 通过近似后验推断的过程学习包含隐藏变量的概率模型[13−16, 19−24]. 以式 (1) 代表的多因子模型为例, 其中 $\boldsymbol{y}$ 是已知的观测变量, 而 $\boldsymbol{x}$ 是隐藏变量, 在 $\boldsymbol{x}$ 属于特定分布的假设下, 目标是计算出式 (2) 中的 $\boldsymbol{C}$、$\boldsymbol{X}$ 和 $\boldsymbol{R}$ 使其能够最优拟合 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的联合分布. 文献 [14] 中首先提出了求解此类线性模型的期望最大化过程, 它通过计算 $\boldsymbol{x}$ 在 $\boldsymbol{y}$ 下分布的期望并最大化 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的联合分布的似然估计值均值, 迭代计算 $\boldsymbol{C}$、$\boldsymbol{X}$ 和 $\boldsymbol{R}$ 直到收敛.

对含有隐藏变量模型的随机梯度下降法可以看做是一种特殊的期望最大化过程, 其最大化步骤由单独的梯度下降步骤组成; 而其他一些期望最大化过程的变种还会包含更多额外的步骤[15]. 期望最大化过程有两个重要的特点: 1) 整个迭代过程是由对于变量分布的假设推导得到的, 因此对一个完整的数据集, 所有未知变量都可以通过期望最大化过程得到, 但这个特点并不是期望最大化过程独有的; 2) 当期望最大化过程得到中间结果时, 数据的分布假设可以被更改, 从而继续新的期望最大化过程, 而这个特点是其他机器学习算法中很罕见的[14]. 近年来, 期望最大化过程被应用在各种不同的场景当中: 例如使用期望最大化过程推测回声位置[19], 使用期望最大化过程将自回归和非自回归模型进行结合, 从而在保证精度的情况下降低模型延迟[20], 使用变种的期望最大化过程对噪音程度未知的图像进行去模糊化处理[21], 使用变种期望最大化过程进行径向基核函数网络自回归模型的参数估计[22], 使用期望最大化过程进行非刚性点集配准研究[23] 以及使用期望最大化过程与神经网络结合进行全重叠的手写数字识别与分离[24] 等, 这些应用都显示出期望最大化过程在当下仍旧是实用的算法.

1.2　统计学因子分析

因子分析是经典的统计学方法, 被广泛应用于机器学习和金融等领域[15−18, 25−29], 例如利用因子分析方法处理语音系统中的说话人识别问题[26] 和语音重建问题[27], 将因子分析方法运用于场景图像识别的神经网络中[28], 以及将变种的因子分析应用到脑神经数据的分析当中[29]. 而在金融领域中, 因子分析通常被用于生成风险模型来评估资产风险[3].

金融领域所需的因子分析需要优化的问题, 是通过对式 (1) 所示的多因子模型引入连续时间序列并对因子加上特定分布建立的模型. 首先, 在公式 (1) 的基础上假设所有因子均为隐藏变量并引入时间变量 t ($t = 1, \cdots, n$): 记 $\boldsymbol{y}_t \in \mathbf{R}^m$ 为 m 个资产的价值在 t 时刻的一条观测数据, $\boldsymbol{x}_t \in \mathbf{R}^s$ 为 t 时刻不能被观测到的 s 个服从 $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ 分布的隐藏因子, 其中, $\boldsymbol{I}$ 表示单位矩阵, 代表 s 个因子服从均值为 0 方差为 1 的正态分布, 同时 $\boldsymbol{r}_t \in \mathbf{R}^m$ 表示 m 个服从均值为 0 方差为 $\boldsymbol{R}$ 正态分布的残差. 根据上述的假设, 因子分析的基本模型为[15−16]:

$$\begin{cases} \boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \\ \boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{r}_t,\ \boldsymbol{r}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}) \end{cases} \tag{3}$$

其中, $\boldsymbol{C} \in \mathbf{R}^{m\times s}$ 是因子的暴露系数矩阵且 $\boldsymbol{R} \in \mathbf{R}^{m\times m}$ 是残差对角阵, 同时可知隐藏因子的协方差矩阵 $\boldsymbol{X} \in \mathbf{R}^{s\times s}$ 为单位阵. 基于式 (3), 可得[15−16]:

$$\boldsymbol{y}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}) \tag{4}$$

式 (4) 代表在式 (3) 的假设条件下, $\{\boldsymbol{y}_t\}$ 服从均值为 0, 方差为 $\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}$ 的正态分布, 因此评估相应资产的风险可通过因子分析方法计算出 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 并最终计算出风险矩阵 (2) 获得.

为了求解 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 使其拟合式 (3)、(4) 所代表的因子分析模型, 一个思路是求解 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 使其最优拟合 $\{\boldsymbol{x}_t\}$ 及 $\{\boldsymbol{y}_t\}$ 的联合分布. 文献 [15−16] 提出使用期望最大化过程求解 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 使其最优拟合 $\{\boldsymbol{x}_t\}$ 及 $\{\boldsymbol{y}_t\}$ 的联合分布, 即最大化 $\{\boldsymbol{x}_t\}$ 及 $\{\boldsymbol{y}_t\}$ 联合分布的对数似然估计值均值[13−16]. 首先, 根据期望最大化过程在期望计算步骤需要计算有关 $\boldsymbol{x}_t$ 在 $\boldsymbol{y}_t$ 下分布的期望及均值, 可通过式 (3) 推导出:

$$\begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{x}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R} & \boldsymbol{C} \\ \boldsymbol{C}^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix} \right) \tag{5}$$

借由式 (5), 可推导出 $\boldsymbol{x}_t$ 在 $\boldsymbol{y}_t$ 下的均值以及方差:

$$\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t] = \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{y}_t \tag{6}$$

$$\mathrm{var}[\boldsymbol{x}_t|\boldsymbol{y}_t] = \boldsymbol{I} - \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{C} \tag{7}$$

同时, $\boldsymbol{x}_t$ 在 $\boldsymbol{y}_t$ 下协方差矩阵均值为:

$$\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t] = \mathrm{var}[\boldsymbol{x}_t|\boldsymbol{y}_t] + \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]^{\mathrm{T}} \tag{8}$$

$$\begin{aligned} &\max_{\boldsymbol{C},\boldsymbol{R}}\ LL(\{\boldsymbol{y}_t\}, \{\boldsymbol{x}_t\}) = \ln(L(\{\boldsymbol{y}_t\}, \{\boldsymbol{x}_t\})) = \\ &\ln\left\{ \prod_{t=1}^{n} \left[(2\pi)^{-\frac{k}{2}} \exp\left(-\frac{\boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{x}_t}{2} \right) \right] \prod_{t=1}^{n} \left[(2\pi)^{-\frac{m}{2}} |\boldsymbol{R}|^{-\frac{1}{2}} \right. \right. \\ &\left. \left. \exp\left(-\frac{(\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t)}{2} \right) \right] \right\} = \\ &-\frac{1}{2}\left\{ \sum_{t=1}^{n} [((\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t)^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{x}_t)) + \boldsymbol{x}_t^{\mathrm{T}}\boldsymbol{x}_t] + \right. \\ &\left. n\ln|\boldsymbol{R}| + (m+k)n\ln(2\pi) \right\} \end{aligned} \tag{9}$$

$$\begin{aligned} &\max_{\boldsymbol{C},\boldsymbol{R}}\ \mathcal{Q} = \mathrm{E}[LL(\{\boldsymbol{y}_t\}, \{\boldsymbol{x}_t\})] = - \\ &\sum_{t=1}^{n} \left(\frac{1}{2}\boldsymbol{y}_t^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t] + \right. \\ &\left. \frac{1}{2}\mathrm{trace}[\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t]] \right) - \frac{n}{2}\ln|\boldsymbol{R}| + c \end{aligned} \tag{10}$$

由于因子分析需要最大化 $\{\boldsymbol{x}_t\}$ 以及 $\{\boldsymbol{y}_t\}$ 联合分布的似然估计值均值, 因此首先引入最大化对数似然估计值表达式 (9)[15−16], 同时最大化式 (9) 等价于最大化其均值, 因此推导出均值表达形式的式 (10)[15−16], 其中 $|\boldsymbol{R}|$ 表示 $\boldsymbol{R}$ 的行列式而 trace$(\cdot)$ 代表矩

阵的迹, 此时式 (10) 即为所求问题的表达式. 由于需要最大化 (10), 因此通过对式 (10) 求偏导可以得到 $\boldsymbol{C}$ 以及 $\boldsymbol{R}$ 的迭代式 (11) 和 (12), 从而进行迭代求解以达到收敛要求, 其中 diag{·} 表示抽取矩阵的对角元形成对角阵而 c 表示常量.

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{C}} = 0 \Rightarrow \sum_{t=1}^{n} \boldsymbol{R}^{-1}\boldsymbol{y}_t \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]^{\mathrm{T}} - \sum_{t=1}^{n} \boldsymbol{R}^{-1}\boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t] = 0$$

$$\Rightarrow \boldsymbol{C} = \left(\sum_{t=1}^{n} \boldsymbol{y}_t \mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]^{\mathrm{T}}\right)\left(\sum_{t=1}^{n} \mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t]\right)^{-1} \tag{11}$$

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{R}^{-1}} = 0 \Rightarrow \frac{n}{2}\boldsymbol{R} - \sum_{t=1}^{n}\left(\frac{1}{2}\boldsymbol{y}_t\boldsymbol{y}_t^{\mathrm{T}} - \boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]\boldsymbol{y}_t^{\mathrm{T}} + \frac{1}{2}\boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t]\boldsymbol{C}^{\mathrm{T}}\right) = 0$$

$$\Rightarrow \boldsymbol{R} = \frac{1}{n}\mathrm{diag}\left\{\sum_{t=1}^{n}\left(\boldsymbol{y}_t\boldsymbol{y}_t^{\mathrm{T}} - \boldsymbol{C}\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]\boldsymbol{y}_t^{\mathrm{T}}\right)\right\} \tag{12}$$

$$LL(\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})) = -\frac{1}{2}[mn\ln(2\pi) + n\ln(|\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}|) + \mathrm{trace}(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1}\boldsymbol{Y})] \tag{13}$$

基于上述的所有推导, 因子分析的期望最大化过程为: 1) 根据已有的 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 计算期望 (6) ~ (8); 2) 通过已计算好的期望根据式 (11) 和 (12) 依次更新 $\boldsymbol{C}$ 和 $\boldsymbol{R}$; 3) 当迭代结果未收敛时重复前两步. 由于因子分析的分布假设下 $\{\boldsymbol{y}_t\}$ 数据服从式 (4) 的分布, 因此可由式 (4) 推导出迭代过程的判定收敛的对数似然估计值均值式 (13), 其中 $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n] \in \mathbf{R}^{m\times n}$ 为 m 个资产在 n 个时刻上的回报矩阵, 而收敛条件可写为两次迭代步之间 (13) 的变化小于预设的迭代收敛阈值 ϵ. 算法 1 中整理了基于期望最大化过程的因子分析算法:

算法 1. 基于期望最大化过程的因子分析 (FA)

输入. 回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times n}$, 统计学因子数 s, 迭代收敛阈值 ϵ

输出. $\boldsymbol{C}\in\mathbf{R}^{m\times s}$, $\boldsymbol{R}\in\mathbf{R}^{m\times m}$

1. 初始化 $\boldsymbol{C}$ 和 $\boldsymbol{R}$
2. WHILE (式 (13) 计算值变化大于 ϵ) DO
3. $\boldsymbol{B} = \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1}$
4. 根据式 (6) 计算 $\boldsymbol{D} = \sum_{t=1}^{n}\boldsymbol{y}_t\mathrm{E}[\boldsymbol{x}_t|\boldsymbol{y}_t]^{\mathrm{T}} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}$
5. 根据式 (6)~(8) 计算 $\boldsymbol{G} = \sum_{t=1}^{n}\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t] = n(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{C}) + \boldsymbol{B}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}$
6. 根据式 (11) 计算 $\boldsymbol{C} = \boldsymbol{D}\boldsymbol{G}^{-1}$
7. 根据式 (12) 计算 $\boldsymbol{R} = \mathrm{diag}\{\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} - \boldsymbol{C}\boldsymbol{D}^{\mathrm{T}}\}/n$
8. END WHILE
9. 返回矩阵 $\boldsymbol{C}$ 以及 $\boldsymbol{R}$

记算法 1 的总迭代步数为 $iter$, 由于每次迭代中时间复杂度最大的步骤为计算 $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$, 因此每次迭代的时间复杂度是 $\mathrm{O}(m^2n)$, 故算法 1 的时间复杂度为 $\mathrm{O}(iter\cdot m^2n)$.

2 结合基本面因子的快速因子分析

本节首先介绍利用采样协方差矩阵加速因子分析的快速因子分析, 并在快速因子分析的基础上推导引入基本面因子的混合因子分析, 最后介绍近似最优选择基本面因子的混合因子分析. 本文遵循 Matlab 语言的习惯来表示矩阵中的部分元素, 以及对矩阵的一些操作.

2.1 快速因子分析

实际情况中通常时刻数目 n 远大于统计学因子数目 s, 因此希望替换算法 1 中 $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ 的计算来减少算法运行时间. 记采样协方差矩阵 $\boldsymbol{S} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}/n$, 使用 $\boldsymbol{S}$ 代替 $\boldsymbol{Y}$ 进行迭代, 则可以推导出: $\boldsymbol{D} = n\boldsymbol{S}\boldsymbol{B}^{\mathrm{T}}$, $\boldsymbol{G} = n(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{C}) + n\boldsymbol{B}\boldsymbol{S}\boldsymbol{B}^{\mathrm{T}} = n(\boldsymbol{I} + \boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}))$, $\boldsymbol{R} = \mathrm{diag}\{\boldsymbol{S} - \boldsymbol{C}\boldsymbol{D}^{\mathrm{T}}/n\}$, 并且式 (13) 中的 $\mathrm{trace}(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathbf{T}} + \boldsymbol{R})^{-1}\boldsymbol{Y}) = \mathrm{trace}\ (\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathbf{T}} + \boldsymbol{R})^{-1}) = \mathrm{trace}(n\boldsymbol{S}(\boldsymbol{C}\boldsymbol{C}^{\mathbf{T}} + \boldsymbol{R})^{-1})$. 根据上述推导, 借助采样协方差矩阵的快速因子分析整理于算法 2 中:

算法 2. 快速因子分析 (FFA)

输入. 回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times n}$, 统计学因子数 s, 迭代收敛阈值 ϵ

输出. $\boldsymbol{C}\in\mathbf{R}^{m\times s}$, $\boldsymbol{R}\in\mathbf{R}^{m\times m}$

1. 初始化 $\boldsymbol{C}$ 和 $\boldsymbol{R}$
2. $\boldsymbol{S} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}/n$
3. WHILE (式 (13) 计算值变化大于 ϵ) DO
4. $\boldsymbol{B} = \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1}$
5. $\boldsymbol{D} = n\boldsymbol{S}\boldsymbol{B}^{\mathrm{T}}$
6. $\boldsymbol{G} = n(\boldsymbol{I} + \boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}))$
7. $\boldsymbol{C} = \boldsymbol{D}\boldsymbol{G}^{-1}$
8. $\boldsymbol{R} = \mathrm{diag}\{\boldsymbol{S} - \boldsymbol{C}\boldsymbol{D}^{\mathrm{T}}/n\}$
9. END WHILE
10. 返回矩阵 $\boldsymbol{C}$ 以及 $\boldsymbol{R}$

由于算法 2 中单个迭代步的时间复杂度为 $\mathrm{O}(m^2s)$, 而算法 2 只在第 2 步计算了一次 $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$, 因此算法 2 总体时间复杂度为 $\mathrm{O}(iter\cdot m^2s + m^2n)$, 小于算法 1 的 $\mathrm{O}(iter\cdot m^2n)$.

2.2 结合基本面因子的混合因子分析

基本面因子模型使用资产的历史回报以及已有的基本面因子暴露, 例如市值、市盈率、市销率和流动比率等, 生成风险模型用以评估风险[2−3, 9]. 基本面因子暴露在不同层面表征资产的金融特征, 例如: 市值表征一个公司的整体价值, 而流动比率代表公司偿还短期债务的能力, 因此基本面因子模型在金融风险管理中表现出很强的可解释性. 算法 3 是使用普通最小二乘法构建基本面因子模型的经典算法[2, 9]:

算法 3. 普通最小二乘法 (OLS)

输入. 回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times n}$, 基本面因子暴露矩阵 $\boldsymbol{C}_1\in\mathbf{R}^{m\times f}$

输出. 基本面因子协方差矩阵 $\boldsymbol{X}_1\in\mathbf{R}^{f\times f}$, $\boldsymbol{R}\in\mathbf{R}^{m\times m}$

1. 计算基本面因子 $\boldsymbol{F}_1=(\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{C}_1)^{-1}\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{Y}$
2. 计算残差矩阵 $\boldsymbol{R}_1=\boldsymbol{Y}-\boldsymbol{C}_1\boldsymbol{F}_1$
3. 计算残差协方差对角阵 $\boldsymbol{R}=\mathrm{diag}\{\boldsymbol{R}_1\boldsymbol{R}_1^{\mathrm{T}}\}/n$
4. 计算基本面因子协方差 $\boldsymbol{X}_1=\boldsymbol{F}_1\boldsymbol{F}_1^{\mathrm{T}}/n$
5. 返回 $\boldsymbol{X}_1$, $\boldsymbol{R}$

由于基本面因子暴露 $\boldsymbol{C}_1\in\mathbf{R}^{m\times f}$ 已知, 借由式 (1) 可以推出 $\boldsymbol{Y}=\boldsymbol{C}_1\boldsymbol{F}_1+\boldsymbol{R}_1$, 其中 $\boldsymbol{F}_1\in\mathbf{R}^{f\times n}$ 为基本面因子矩阵而 $\boldsymbol{R}_1\in\mathbf{R}^{m\times n}$ 为残差阵, 因此问题变为拟合 $\boldsymbol{Y}=\boldsymbol{C}_1\boldsymbol{F}_1$, 之后计算残差 $\boldsymbol{R}_1$, 最后计算构造风险矩阵 (2) 的矩阵 $\boldsymbol{X}$ 和 $\boldsymbol{R}$. 算法 3 中的第 1 步可由 $\boldsymbol{Y}=\boldsymbol{C}_1\boldsymbol{F}_1$ 等式两侧同乘 $\boldsymbol{C}_1^{\mathrm{T}}$ 得到 $\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{C}_1\boldsymbol{F}_1$ 构造; 第 2 步表示根据 $\boldsymbol{Y}-\boldsymbol{C}_1\boldsymbol{F}_1^{\mathrm{T}}$ 计算残差矩阵 $\boldsymbol{R}_1$; 第 3 步表示求解残差矩阵 $\boldsymbol{R}_1$ 的协方差以构造残差对角阵 $\boldsymbol{R}$; 第 4 步为计算基本面因子协方差矩阵 $\boldsymbol{X}_1\in\mathbf{R}^{f\times f}$; 最后返回 $\boldsymbol{X}_1$ 和 $\boldsymbol{R}$. 在算法 3 执行完成后, 使用 $\boldsymbol{C}_1\boldsymbol{X}_1\boldsymbol{C}_1^{\mathrm{T}}+\boldsymbol{R}$ 即可得到式 (2) 中的风险矩阵.

为了将金融的领域知识集成到统计学因子分析中, 需要推导新的期望最大化过程, 而首先则需要将已知的基本面因子暴露固定到第 1.2 节的原始模型中. 由于基本面因子的值在假设中是不可及的, 因此新优化问题的模型可修改为:

$$\begin{cases}[\boldsymbol{x}_{t,1};\boldsymbol{x}_{t,2}]=\boldsymbol{x}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \boldsymbol{y}_t=\boldsymbol{C}_1\boldsymbol{x}_{t,1}+\boldsymbol{C}_2\boldsymbol{x}_{t,2}+\boldsymbol{r}_t,\ \boldsymbol{r}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{R})\end{cases}\tag{14}$$

其中, $\boldsymbol{C}_1\in\mathbf{R}^{m\times f}$ 是观测到的基本面因子暴露矩阵而 $\boldsymbol{x}_{t,1}\in\mathbf{R}^f$ 是基本面因子, $\boldsymbol{C}_2\in\mathbf{R}^{m\times s}$ 为统计学因子暴露矩阵而 $\boldsymbol{x}_{t,2}\in\mathbf{R}^s$ 为统计学因子, $[\boldsymbol{x}_{t,1};\boldsymbol{x}_{t,2}]$ 表示将两个列向量拼接为一个列向量. 然而, 实际的基本面因子往往不是相互独立的, 所以式 (14) 的假设会导致不准确的结果. 因此, 我们引入方阵 $\boldsymbol{A}\in\mathbf{R}^{f\times f}$ 来调整基本面因子之间的数值关系和相关性:

$$\begin{cases}[\boldsymbol{x}_{t,1};\boldsymbol{x}_{t,2}]=\boldsymbol{x}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})\\ \boldsymbol{y}_t=\boldsymbol{C}_1\boldsymbol{A}\boldsymbol{x}_{t,1}+\boldsymbol{C}_2\boldsymbol{x}_{t,2}+\boldsymbol{r}_t,\ \boldsymbol{r}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{R})\end{cases}\tag{15}$$

其中, $\boldsymbol{A}$ 也需要在期望最大化过程中被迭代更新, 并且可由式 (15) 推导出 (2) 中的全因子的暴露矩阵 $\boldsymbol{C}=[\boldsymbol{C}_1\boldsymbol{A},\boldsymbol{C}_2]$ 用来进行期望最大化过程的迭代. 因此新的混合因子分析需要求解的问题变为计算 $\boldsymbol{A}$、$\boldsymbol{C}_2$ 和 $\boldsymbol{R}$ 使其最大化 $\{\boldsymbol{x}_t\}$ 及 $\{\boldsymbol{y}_t\}$ 联合分布.

根据式 (15) 以及第 1 节中推导的期望最大化过程, 首先推导 $\boldsymbol{x}_{t,1}$ 和 $\boldsymbol{x}_{t,2}$ 在 $\boldsymbol{y}_t$ 下的期望:

$$\mathrm{E}[\boldsymbol{x}_{t,1}|\boldsymbol{y}_t]=(\boldsymbol{C}_1\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{y_t}\tag{16}$$

$$\mathrm{E}[\boldsymbol{x}_{t,2}|\boldsymbol{y}_t]=\boldsymbol{C}_2^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R})^{-1}\boldsymbol{y_t}\tag{17}$$

而根据式 (8) 及式 (15) 可以推导出:

$$\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]=\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t](1:f,1:f)\tag{18}$$

$$\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]=\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t](1:f,f+1:f+s)\tag{19}$$

$$\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]=\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t](f+1:f+s,f+1:f+s)\tag{20}$$

其中,$\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t](1:f,f+1:f+s)$代表对矩阵 $\mathrm{E}[\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}|\boldsymbol{y}_t]$ 取 1 到 f 行与 $f+1$ 到 $f+s$ 列形成的新矩阵. 根据期望最大化过程中最大化步骤的需求, 需要最大化新的对数自然估计值的期望. 首先得到新的需要最大化的期望表达式 (21), 而通过对式 (21) 进行求偏导可以得到迭代计算 $\boldsymbol{A}$ 以及 $\boldsymbol{C}_2$ 的式 (22) 和 (23), 再将式 (12) 中对应的统计学因子暴露阵替换为本节中的全因子暴露矩阵后即可得到 $\boldsymbol{R}$ 的迭代公式.

$$\begin{aligned}\max_{\boldsymbol{A},\boldsymbol{C}_2,\boldsymbol{R}}\ \hat{\mathcal{Q}}\ =&\ \mathrm{E}[LL(\{\boldsymbol{y}_t\},\{[\boldsymbol{x}_{t,1};\boldsymbol{x}_{t,2}]\})]=-\\ &\sum_{t=1}^{n}\Big(\frac{1}{2}\boldsymbol{y}_t^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}_t-\boldsymbol{y}_t^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_1\boldsymbol{A}\mathrm{E}[\boldsymbol{x}_{t,1}|\boldsymbol{y}_t]-\\ &\boldsymbol{y}_t^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}|\boldsymbol{y}_t]+\\ &\frac{1}{2}\mathrm{trace}[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_1\boldsymbol{A}\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]]+\\ &\mathrm{trace}[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]]+\\ &\frac{1}{2}\mathrm{trace}[\boldsymbol{C}_2^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]]\Big)-\\ &\frac{n}{2}\ln|\boldsymbol{R}|+c\end{aligned}\tag{21}$$

$$\begin{aligned}\frac{\partial\hat{\mathcal{Q}}}{\partial\boldsymbol{A}}=0\Rightarrow&\sum_{t=1}^{n}\Big(-\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}_t\mathrm{E}[\boldsymbol{x}_{t,1}|\boldsymbol{y}_t]^{\mathrm{T}}+\\ &\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_1\boldsymbol{A}\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]+\\ &\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]\Big)=0\\ \Rightarrow\ &\boldsymbol{A}=(\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}_1)^{-1}\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}\times\end{aligned}$$

$$\left(\sum_{t=1}^{n}\boldsymbol{y}_t\mathrm{E}[\boldsymbol{x}_{t,1}|\boldsymbol{y}_t]^{\mathrm{T}}-\sum_{t=1}^{n}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]\right)\times$$

$$\left(\sum_{t=1}^{n}\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,1}^{\mathrm{T}}|\boldsymbol{y}_t]\right)^{-1} \tag{22}$$

$$\frac{\partial\hat{\mathcal{Q}}}{\partial\boldsymbol{C}_2}=0\Rightarrow\sum_{t=1}^{n}\Big(-\boldsymbol{R}^{-1}\boldsymbol{y}_t\mathrm{E}[\boldsymbol{x}_{t,2}|\boldsymbol{y}_t]^{\mathrm{T}}+$$

$$\boldsymbol{R}^{-1}\boldsymbol{C}_1\boldsymbol{A}\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]+$$

$$\boldsymbol{R}^{-1}\boldsymbol{C}_2\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]\Big)=0$$

$$\Rightarrow\boldsymbol{C}_2=\left(\sum_{t=1}^{n}\boldsymbol{y}_t\mathrm{E}[\boldsymbol{x}_{t,2}|\boldsymbol{y}_t]^{\mathrm{T}}-\sum_{t=1}^{n}\boldsymbol{C}_1\boldsymbol{A}\mathrm{E}[\boldsymbol{x}_{t,1}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]\right)$$

$$\left(\sum_{t=1}^{n}\mathrm{E}[\boldsymbol{x}_{t,2}\boldsymbol{x}_{t,2}^{\mathrm{T}}|\boldsymbol{y}_t]\right)^{-1} \tag{23}$$

根据上述的推导, 新的期望最大化过程可表述为: 第 1 步, 根据已有的 $\boldsymbol{A}$、$\boldsymbol{C}_2$ 以及 $\boldsymbol{R}$ 计算期望 (8)(16)(17); 第 2 步, 通过已计算好的期望以及式 (22)、(23)、(12) 依次更新 $\boldsymbol{A}$、$\boldsymbol{C}_2$ 及 $\boldsymbol{R}$; 第 3 步, 当迭代结果未收敛时重复前两步. 根据新的期望最大化过程和修改后的快速因子分析算法可整理出结合基本面因子的混合因子分析算法 4:

算法 4. 结合基本面因子的混合因子分析 (HFA)

输入. 回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times n}$, 基本面暴露阵 $\boldsymbol{C}_1\in\mathbf{R}^{m\times f}$, 统计学因子数 s, 迭代收敛阈值 ϵ.

输出. $\boldsymbol{C}\in\mathbf{R}^{m\times(f+s)}$, $\boldsymbol{R}\in\mathbf{R}^{m\times m}$

1. 初始化 $\boldsymbol{A}$、$\boldsymbol{C}_2$ 和 $\boldsymbol{R}$
2. $\boldsymbol{C}=[\boldsymbol{C}_1\boldsymbol{A},\boldsymbol{C}_2]$, $\boldsymbol{S}=\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}/n$, $k=f+s$
3. WHILE (对数似然估计值变化大于 ϵ) DO
4. $\quad\boldsymbol{B}=\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R})^{-1}$, $\boldsymbol{T}=\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}^{-1}$
5. $\quad\boldsymbol{D}=n\boldsymbol{S}\boldsymbol{B}^{\mathrm{T}}$
6. $\quad\boldsymbol{G}=n(\boldsymbol{I}+\boldsymbol{B}(\boldsymbol{D}-\boldsymbol{C}))$
7. $\quad$根据式 (22) 计算 $\boldsymbol{A}=(\boldsymbol{T}\boldsymbol{C}_1)^{-1}\boldsymbol{T}\{\boldsymbol{D}(1\!:\!m,1\!:\!f)-\boldsymbol{C}_2\boldsymbol{G}\ (f+1\!:\!k,1\!:\!f)\}\boldsymbol{G}(1\!:\!f,1\!:\!f)^{-1}$
8. $\quad$根据式 (23) 计算 $\boldsymbol{C}_2=\{\boldsymbol{D}(1\!:\!m,f+1\!:\!k)-\boldsymbol{C}_1\boldsymbol{A}\boldsymbol{G}(1\!:\!f,f+1\!:\!k)\}\boldsymbol{G}(f+1\!:\!k,f+1\!:\!k)^{-1}$
9. $\quad$计算完整暴露矩阵 $\boldsymbol{C}=[\boldsymbol{C}_1\boldsymbol{A},\boldsymbol{C}_2]$
10. $\quad\boldsymbol{R}=\mathrm{diag}\{\boldsymbol{S}^{\mathrm{T}}-\boldsymbol{C}\boldsymbol{D}^{\mathrm{T}}/n\}$
11. END WHILE
12. 返回矩阵 $\boldsymbol{C}$ 以及 $\boldsymbol{R}$

当因子总数目相同时 (设总因子数目为 k), 尽管算法 4 和算法 2 的时间复杂度相同, 但算法 4 的实际运行时间更短: 由于算法 4 中最大的矩阵求逆操作是对于 $(k-f)\times(k-f)$ 的矩阵进行求逆, 小于算法 2 中 $k\times k$ 的矩阵求逆操作; 同时算法 4 中需要迭代更新的矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{C}_2$ 大小为 $k\times k$ 以及 $m\times(k-f)$, 比算法 2 中的 $\boldsymbol{C}$ 的 $m\times k$ 小. 因此在总因子数目相同时, 算法 4 在 $f>0$ 时的运行时间要小于算法 2.

2.3 近似最优的基本面因子选取算法

在实际情况中, 被观测到的基本面因子数量众多, 但只有部分对于生成风险模型更有价值, 因此本节着眼于通过历史数据找到合适的基本面因子, 并使用挑选出的因子联合算法 4 生成更准确的风险模型. 尽管第 1 节、第 2.1 节、第 2.2 节均旨在最大化该时刻的对数似然估计来推导期望最大化过程, 本节提出的基本面因子选择算法却旨在通过对未来时刻预测的对数似然估计值来选择最优的基本面因子组合.

记所有观测到的基本面因子总数目为 F 而风险模型的总因子数目为 k, 若需要得到 t 时刻最优的基本面因子组合, 则要综合考察之前时刻所有基本面因子组合生成风险模型的预测结果. 以 $t-1$ 时刻为例, 当需要得到基本面因子的所有组合在时刻 $t-1$ 生成的风险模型对于时刻 t 回报预测的对数似然估计值结果时, 需要执行 $\sum_{i=1}^{\min(F,k)}\binom{F}{i}$ 次算法 4 以得到完整的结果, $\binom{F}{i}$ 代表从 F 个元素中选择 i 个元素的组合数, 这意味着当 F 与 k 数量级相同时需要执行 $\mathrm{O}(2^F)$ 次算法 4. 当 t 时刻之前的所有预测的对数似然估计值被累积好后, 便选择使对数似然估计值最大的基本面因子组合来生成 t 时刻的风险模型. 然而 $\mathrm{O}(2^F)$ 次算法 4 的执行需要花费大量时间, 因此需要寻找一个近似最优的算法, 能在更快的运行时间下得到合适的基本面因子组合.

为了快速找到合适的基本面因子组合, 需要一个贪婪算法来搜索可能的解空间. 首先, 通过将每一个的基本面因子暴露在 $t-1$ 时刻的采样协方差矩阵上的拟合结果对基本面因子排序: 记 $\hat{\boldsymbol{C}}_1\in\mathbf{R}^{m\times F}$ 为所有基本面因子暴露矩阵, 即将 $\hat{\boldsymbol{C}}_1$ 的每一列、$t-1$ 时刻的采样协方差矩阵和统计学因子数目 $k-1$ 作为输入传入算法 4, 按照程序结束时风险矩阵对于输入回报矩阵的在式 (13) 上计算出的对数似然估计值对所有基本面因子进行降序排序, 并记排序后的索引向量为 $\boldsymbol{d}$. 之后使用 $t-1$ 时刻的采样协方差矩阵、统计学因子数目 $k-i$ 以及基本面因子暴露矩阵 $\hat{\boldsymbol{C}}_1[1\!:\!m,1\!:\!\boldsymbol{d}[1\!:\!i]],(i=1,\cdots,\min(F,k))$ 通过算法 4 计算风险矩阵, 并使用风险矩阵计算对于 t 时刻的回报 $\boldsymbol{y}_t$ 预测对数似然估计值 (该预测的对数似然估计值在实验部分进行介绍), 并将对数似然估计

值累积到向量 $\boldsymbol{v}\in\mathbf{R}^{\min(F,k)}$ 中, 而算法 5 中的衰减率 r 被用来以一定的衰减率累积对数似然估计值. 最后, 通过找到 $\boldsymbol{v}$ 中最大值的下标 d 来确定 t 时刻风险模型所需的基本面因子组合 $\boldsymbol{d}[1:d]$ 和基本面因子暴露矩阵 $\hat{\boldsymbol{C}}_1[1:m,\boldsymbol{d}[1:d]]$. 算法 5 描述了基本面因子挑选流程 (zeros(F) 为生成长度为 F 的零向量操作):

算法 5. 部分基本面因子挑选算法 (FS)

输入. t–1 时刻回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times(t-1)}$, t 时刻回报 $\boldsymbol{y}_t\in\mathbf{R}^m$, 基本面因子暴露阵 $\hat{\boldsymbol{C}}_1\in\mathbf{R}^{m\times F}$, 似然估计累积向量 $\boldsymbol{v}$, 总因子数 k, 衰减率 r, 迭代收敛阈值 ϵ

输出. 选择后的基本面暴露矩阵 $\boldsymbol{C}_1$, $\boldsymbol{v}$

$\boldsymbol{p}=\text{zeros}(F)$, $\hat{\boldsymbol{v}}=\text{zeros}(\min(F,k))$

FOR $i=1,2,\cdots,F$ DO

$[\boldsymbol{C},\boldsymbol{R}]=\text{HFA}(\boldsymbol{Y},\hat{\boldsymbol{C}}_1[1:m,i],k-1,\epsilon)$

记 $\boldsymbol{p}[i]$ 为风险矩阵 $\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R}$ 与 $\boldsymbol{Y}$ 根据 (13) 计算的对数似然估计值

END FOR

对 $\boldsymbol{p}$ 按照降序排序得到索引 $\boldsymbol{d}$

FOR $i=1,2,\cdots,\min(F,k)$ DO

$[\boldsymbol{C},\boldsymbol{R}]=\text{HFA}(\boldsymbol{Y},\hat{\boldsymbol{C}}_1[1:m,\boldsymbol{d}[1:i]],k-i,\epsilon)$

记 $\hat{\boldsymbol{v}}[i]$ 为风险矩阵 $\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R}$ 对于 $\boldsymbol{y}_t$ 预测的对数似然估计结果

END FOR

$\boldsymbol{v}=r\boldsymbol{v}+\hat{\boldsymbol{v}}$

记 d 为 $\boldsymbol{v}$ 中最大值的索引

返回 $\boldsymbol{C}_1=\hat{\boldsymbol{C}}_1[1:m,\boldsymbol{d}[1:d]]$, $\boldsymbol{v}$

算法 5 在每个时刻使用了 O(F) 次算法 4 来得到近似最优的因子组合, 远小于搜索整个空间的 O(2^F).

先使用算法 5 找到合适的基本面因子, 再通过算法 4 计算风险矩阵, 即可得到实际应用中用以计算风险的算法 6:

算法 6. 挑选基本面因子的混合因子分析 (HFA+)

输入. t–1 时刻回报矩阵 $\boldsymbol{Y}\in\mathbf{R}^{m\times(t-1)}$, t 时刻回报 $\boldsymbol{y}_t\in\mathbf{R}^m$, 基本面因子暴露阵 $\hat{\boldsymbol{C}}_1\in\mathbf{R}^{m\times F}$, 似然估计累积向量 $\boldsymbol{v}$, 总因子数 k, 衰减率 r, 迭代收敛阈值 ϵ

输出. $\boldsymbol{C}\in\mathbf{R}^{m\times k}$, $\boldsymbol{R}\in\mathbf{R}^{m\times m}$, $\boldsymbol{v}$

$[\boldsymbol{C}_1,\boldsymbol{v}]=\text{FS}(\boldsymbol{Y},\boldsymbol{y}_t,\hat{\boldsymbol{C}}_1,\boldsymbol{v},k,r,\epsilon)$

使用 t 时刻的回报 $\boldsymbol{y}_t$ 更新 $\boldsymbol{Y}$, 记 $\boldsymbol{C}_1$ 列数为 f

$[\boldsymbol{C},\boldsymbol{R}]=\text{HFA}(\boldsymbol{Y},\boldsymbol{C}_1,k-f,\epsilon)$

返回 $\boldsymbol{C}$、$\boldsymbol{R}$ 以及 $\boldsymbol{v}$

算法 6 首先使用算法 5 挑选出合适的基本面因子并更新回报矩阵到下一时刻, 随后使用挑选出的基本面因子通过算法 4 生成风险矩阵. 算法 6 中保持累积向量 $\boldsymbol{v}$ 的更新, 使得算法 6 能即时通过 $\boldsymbol{v}$ 挑选合适的基本面因子来生成更准确的风险模型, 也因此算法 6 中每一时刻选择的基本面因子组合及数目均有可能不同.

3　实验

我们在模拟实际情况的三个人造数据集和一个真实数据集上对本文算法进行了对比实验. 实验平台为拥有 Intel Xeon E5-2 680 CPU (2.50 GHz) 的 Cent OS 服务器. 我们使用 Python 3.6 实现了本文算法并记录它们运行所需的 CPU 时间, 单位为秒. 实验中所有算法的迭代收敛阈值 ϵ 为 10^{-8}.

3.1　实验准备

记 $\boldsymbol{Y}=[\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n]\in\mathbf{R}^{m\times n}$ 为 m 个资产在 n 个时刻的回报数据, F 是基本面因子的数目而 S 是统计学因子的数目, 通过以下 4 个步骤可生成人造数据集 (所有的随机阵均为标准高斯分布随机阵):

生成 $F\times n$ 的基本面因子随机阵 $\hat{\boldsymbol{F}}_1$ 以及 $m\times F$ 的基本面因子暴露随机阵 $\hat{\boldsymbol{C}}_1$

生成 $S\times n$ 的统计学因子随机阵 $\hat{\boldsymbol{F}}_2$ 以及 $m\times S$ 的统计学因子暴露随机阵 $\hat{\boldsymbol{C}}_2$

生成 $m\times n$ 的随机噪音阵 $\boldsymbol{E}$

计算回报矩阵 $\boldsymbol{Y}=\hat{\boldsymbol{C}}_1\boldsymbol{\Sigma}_1\hat{\boldsymbol{F}}_1+\hat{\boldsymbol{C}}_2\boldsymbol{\Sigma}_2\hat{\boldsymbol{F}}_2+\delta\boldsymbol{E}$, 其中 $\boldsymbol{\Sigma}_1$ 以及 $\boldsymbol{\Sigma}_2$ 为对角阵, 且 $\boldsymbol{\Sigma}_1$、$\boldsymbol{\Sigma}_2$ 和 δ 分别代表不同部分的权重

使用滑动窗口来生成风险模型并评估风险, 这样使得每次都使用最新的一部分数据来计算采样协方差矩阵, 需要对回报矩阵做如下的操作以不同重要程度保留退出滑动窗口数据的历史信息:

$$\tilde{\boldsymbol{Y}}_t=[\lambda^{t-1}\boldsymbol{y}_1,\cdots,\lambda\boldsymbol{y}_{t-1},\boldsymbol{y}_t] \tag{24}$$

其中, $\lambda=0.5^{\frac{1}{h}}$ 且 h 是表示半衰期的常量, 代表经过 h 个时刻数据的重要性变为原来的一半. 通过式 (24), 所有数据的信息都被保留, 并且越新的数据重要性越高. 记 $n_t=\sum_{i=0}^{t-1}\lambda^{2i}$, 采样协方差矩阵表达式被修正为 $\boldsymbol{S}_t=\tilde{\boldsymbol{Y}}_t\tilde{\boldsymbol{Y}}_t^{\mathrm{T}}/n_t$, 同时所有算法步骤中的 n 也需要相应的修改为 n_t.

真实数据来源于中国的股票市场. 记 $\boldsymbol{p}_t$ 为 m 支股票在第 t 天的价格数据, 可以使用对数回报率来作为回报向量使用, 即:

$$\boldsymbol{y}_t=\ln\left(\frac{\boldsymbol{p}_{t+1}}{\boldsymbol{p}_t}\right)=\ln\boldsymbol{p}_{t+1}-\ln\boldsymbol{p}_t \tag{25}$$

通常对数回报率被看做服从高斯独立分布. 对数回报率矩阵同样进行了式 (25) 的操作. 所有基本面因子暴露 $\hat{\boldsymbol{C}}_f$ 来自 Tushare 金融大数据社区 (ht-

tps://tushare.pro/), 同时按列进行了标准化处理. 所有的人造数据矩阵、真实数据的信息和可运行的示例代码都被整理在 https://github.com/ExpDataForRiskModel/ExpData 中.

实验结果的评价也很重要. 记 $\boldsymbol{\Omega}_t = \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}$ 为算法 1、2、4、6 在时刻 t 生成的风险矩阵, 而 $\boldsymbol{\Omega}_t = \boldsymbol{C}_1\boldsymbol{X}_1\boldsymbol{C}_1^{\mathrm{T}} + \boldsymbol{R}$ 为算法 3 在时刻 t 生成的风险矩阵, 且 $\boldsymbol{y}_{t+1}$ 为 $t+1$ 时刻的回报数据, 则 t 时刻的风险模型对于 $t+1$ 时刻回报预测的对数似然估计值为:

$$LL(\boldsymbol{y}_{t+1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}_t)) = -\frac{1}{2}[m\ln(2\pi) + \ln(|\boldsymbol{\Omega}_t|) + \boldsymbol{y}_{t+1}^{\mathrm{T}}\boldsymbol{\Omega}_t^{-1}\boldsymbol{y}_{t+1}] \quad (26)$$

式中, 因为 $m\ln(2\pi)$ 是正常数, 预测的对数似然估计值只与 $\boldsymbol{\Omega}_t$ 相关. 而 $\boldsymbol{\Omega}_t$ 是借助算法以及 $\boldsymbol{S}_t$ 计算得到, 代表 $\boldsymbol{\Omega}_t$ 与 $\boldsymbol{S}_t$ 非常接近, 当 $\boldsymbol{S}_t$ 的行列式小于 1 时, $\ln|\boldsymbol{\Omega}_t|$ 会变得非常小而此时式 (26) 可能产出正数. 由于对不同数据集式 (26) 计算的值的绝对大小会有所不同, 因此单个实验内的相对大小的比较是关注的重点. 在实验过程中统计单个算法在实验时间段内的平均对数似然估计值 (E(LL)) 以及平均对数似然估计的标准差 σ(E), 其中前者代表了所生成风险模型的准确程度而后者代表了生成风险模型的稳定性. 同时标准差也是评价新算法是否显著的好于其他算法的重要指标.

由于实际情况下会以一天、半天甚至一个小时计算风险模型用以评估下一时段的风险, 同时股票的实际数目有数千只, 因此本文人造数据的实验着眼于 m 较大的情况来对比不同算法的运行时间.

3.2 人造数据集实验

本节在 3 个人造数据集上进行了不同算法的对比实验: 第一个人造数据集上的实验首先测试在相同的数据集下算法 2 是否相对于算法 1 在保持结果基本不变的同时获得了较高的加速; 第二个人造数据集上的实验测试当基本面因子重要性更高时, 算法 4 的结果是否要优于算法 1、2、3; 第三个人造数据集上的实验则是测试提出的算法 6 是否能在模拟真实情况的基本面因子中挑选出合适的基本面因子组合来生成比算法 1、2、3、4 更好的风险模型.

第一个人造数据集的设置为: $m = 500$, $n = 50\,100$, $F = 10$, $S = 10$, $\boldsymbol{\Sigma}_1 = 5\boldsymbol{I}$, $\boldsymbol{\Sigma}_2 = 3\boldsymbol{I}$, $\delta = 5, 10$, $h = 60$. $\boldsymbol{\Sigma}_1 = 5\boldsymbol{I}$ 和 $\boldsymbol{\Sigma}_2 = 3\boldsymbol{I}$ 表示基本面因子的重要性高于统计学因子, 而 $\delta = 5, 10$ 表示不同程度的噪音. 我们分别测试算法 1 和算法 2 在总因子数目为 10 和 13 时, 从时刻 50 000 到时刻 50 100 的预测结果, 实验结果列于表 1 中. 除了对数似然估计值均值、标准差和时间外还统计了算法的平均迭代步数 (E($iter$)) 和算法 2 对算法 1 的加速比.

由于噪音程度 δ 不同, 对于回报矩阵的元素大小有影响, 同时也会影响到最终求出的对数似然估计值的均值, 而更大的 δ 会导致更大的回报矩阵数值, 因此通过式 (26) 所求得的对数似然估计值就会更小. 表 1 的数据表现出算法 1 和算法 2 在总因子数目相同时对数似然估计值均值和其标准差基本相同, 同时迭代步数也基本相同 (结果的波动则是由于数值精度误差所导致的), 但由于时刻数 n 远大于总因子数目, 因此算法 2 在所有条件下都要比算法 1 快 31 倍以上, 显示出算法 2 良好的运行效率.

第二个人造数据集数据集设置 $m = 1\,000$, $n = 4\,200$, $\delta = 3, 5, 10$, 其余设置与第一个数据集相同. $\delta = 3, 5, 10$ 用于分别模拟由低到高不同噪音程度的人造数据, 而 $m = 1\,000$, $n = 4\,200$ 用于模拟更加真实的数据集的情况. 第二个人造数据集用于测试在更重要的基本面因子被挑选出后算法 3 和算法 4 的性能. 时刻 4 000 到时刻 4 100 被设置为实验区间来统计预测结果. 由于基本面因子数量为 10, 首先分别测试了算法 1 和算法 2 有 10 个统计学因子和算法 3 有 10 个基本面因子的结果, 再测试了算法 1、算法 2 与算法 4 在总因子数目为 13 的实验结果, 其中算法 1 和算法 2 使用 13 个统计学因子而算法 4 使用 10 个基本面因子和 3 个统计学因子. 算法 3 和

表 1 算法 1 和算法 2 在第一个人造数据集上的实验结果

Table 1 Results on first synthetic dataset of Alg.1 and Alg.2

因子数	算法	$\delta = 5$					$\delta = 10$				
		时间 (s)	E (LL)	σ (E)	E ($iter$)	加速比	时间 (s)	E (LL)	σ (E)	E ($iter$)	加速比
$s = 10$	算法1 (FA)	444.21	−1 927.63	10.01	709.02	—	252.07	−2 059.10	6.30	405.51	—
$s = 10$	算法2 (FFA)	14.05	−1 927.63	10.01	709.01	31.6	7.80	−2 059.10	6.30	405.55	32.3
$s = 13$	算法1 (FA)	653.60	−1 861.02	11.48	1 019.66	—	313.86	−2 027.42	6.29	492.91	—
$s = 13$	算法2 (FFA)	20.38	−1 863.73	11.39	1 020.77	32.1	9.58	−2 027.08	6.24	492.79	32.8

算法 4 都使用整个基本面因子暴露矩阵作为输入.

表 2 中的结果表现出当总因子数目为 10 时算法 3 在所有噪音程度下都比算法 1 拥有更大的似然估计值均值和更小的标准差, 并且算法 3 的运行时间最少, 表现出基本面模型的优良性能; 但算法 3 的结果却比算法 2 在总因子数目为 13 的结果差, 表现出统计学因子分析良好的扩展性. 由于时刻 n 与总因子数目比例变小, 算法 2 相对算法 1 的加速比只有 2 倍以上, 符合第 2.1 节中的推导. 表 2 中算法 4 的结果表现出当总因子数目为 13 时, 算法 4 的运行结果优于算法 2, 因为算法 4 相对于算法 1、2 拥有更大的对数似然估计的均值以及更小的标准差, 同时算法 4 的运行时间更短; 算法 4 与算法 3 相比, 算法 4 拥有更大的对数似然估计值, 表现出混合因子分析算法优良的性能. 当 $\delta = 3$ 且总因子数目为 13 时, 算法 4 的对数似然估计值为−3 564.72, 大于算法 1 的−3 617.34 且拥有表中最大的差值 52.62; 此外, 算法 4 的标准差为 25.94 小于算法 1 的 33.82, 此时算法 4 相对于算法 1 拥有超过 6.3 倍的最大加速比. 所有的结果都显示出算法 4 在合适的基本面因子已知时能够比算法 1、2 和算法 3 生成更准确的风险模型.

第三个人造数据集的 m、n、F、S、h 和 $\boldsymbol{\Sigma}_2$ 都和第二个人造数据集相同, $\boldsymbol{\Sigma}_1$ 的对角元被设置为 $[5, 5, 5, 3, 3, 3, 3, 1, 1, 1]$, 被用来模拟真实数据中基本面因子重要性高低不同的情况, 代表部分基本面因子的重要性高于统计学因子, 而另一部分的重要性则弱于统计学因子. 第三个人造数据集用于测试算法 6 能否挑选出重要性更高的基本面因子来生成更准确的风险模型矩阵. 算法 6 设置衰减率 $r = 0.7$, 从时刻 4 000 到时刻 4 100 累积对数似然估计向量 $\boldsymbol{v}$, 并记录从时刻 4 100 到时刻 4 200 的预测结果. 其余算法设置时刻 4 100 到时刻 4 200 为测试时段. 算法 1、算法 2、算法 3 和算法 6 在总因子数目为 10 的结果以及算法 1、算法 2、算法 4 和算法 6 在总因子数目为 13 的结果都被记录在表 3 中, 其中算法 3、算法 4 和算法 6 都使用整个基本面因子暴露矩阵作为输入.

表 3 中因子数 $s+f = 10, 13$ 表示限定算法 6 的总因子数目为 10 和 13, 但具体选择基本面因子与统计学因子数目在每个时刻是不固定的, 以 $\delta = 10$ 为例, 算法 6 在总因子数为 10 时平均使用约 5.39 个

表 2　算法 1、算法 2、算法 3 和算法 4 在第二个人造数据集上的实验结果
Table 2　Results on second synthetic dataset of Alg.1, Alg.2, Alg.3 and Alg.4

因子数	算法	$\delta = 3$			$\delta = 5$			$\delta = 10$		
		时间 (s)	E (LL)	σ (E)	时间 (s)	E (LL)	σ (E)	时间 (s)	E (LL)	σ (E)
$s=10, f=0$	算法1 (FA)	484.94	−3 789.03	28.85	389.27	−3 866.59	24.26	227.39	−4 116.68	14.46
$s=10, f=0$	算法2 (FFA)	207.17	−3 789.02	28.85	171.58	−3 866.60	24.65	99.19	−4 116.68	14.46
$s=0, f=10$	算法3 (OLS)	0.23	−3 734.41	22.60	0.23	−3 815.23	19.05	0.23	−4 072.60	11.39
$s=13, f=0$	算法1 (FA)	779.07	−3 617.34	33.82	562.08	−3 732.61	25.96	279.78	−4 046.25	13.46
$s=13, f=0$	算法2 (FFA)	331.09	−3 616.64	33.72	247.17	−3 731.49	26.16	121.48	−4 045.92	13.46
$s=3, f=10$	算法4 (HFA)	123.49	**−3 564.72**	25.94	92.44	**−3 678.27**	20.23	48.63	**−4 002.81**	10.52

表 3　算法 1、算法 2、算法 3、算法 4 及算法 6 在第三个人造数据集上的实验结果
Table 3　Results on third synthetic dataset of Alg.1, Alg.2, Alg.3, Alg.4 and Alg.6

因子数	算法	$\delta = 3$			$\delta = 5$			$\delta = 10$		
		时间 (s)	E (LL)	σ (E)	时间 (s)	E (LL)	σ (E)	时间 (s)	E (LL)	σ (E)
$s=10, f=0$	算法1 (FA)	693.57	−3 594.51	26.64	507.47	−3 709.94	20.81	263.41	−4 019.67	10.86
$s=10, f=0$	算法2 (FFA)	290.06	−3 593.84	26.97	215.58	−3 703.39	21.36	105.50	−4 019.93	10.88
$s=0, f=10$	算法3 (OLS)	0.24	−3 712.75	22.24	0.25	−3 796.19	18.78	0.24	−4 059.71	11.19
$s+f=10$	算法6 (HFA$^+$)	1 726.79	**−3 561.04**	25.26	1 307.14	**−3 683.00**	20.75	596.52	**−4 005.34**	11.34
$s=13, f=0$	算法1 (FA)	845.15	−3 389.00	30.54	721.67	−3 550.65	21.55	312.41	−3 955.47	10.07
$s=13, f=0$	算法2 (FFA)	351.61	−3 389.82	30.89	304.29	−3 549.53	21.45	123.93	−3 955.47	10.07
$s=3, f=10$	算法4 (HFA)	111.30	−3 536.52	28.39	81.30	−3 661.63	22.11	48.15	−3 993.70	11.43
$s+f=13$	算法6 (HFA$^+$)	2 314.11	**−3 378.86**	32.36	1 826.26	**−3 522.23**	21.72	796.98	**−3 933.08**	10.42

表 4 算法 1、算法 2、算法 3 和算法 6 在真实数据集上的实验结果比较
Table 4 Results on real-world dataset of Alg.1, Alg.2, Alg.3 and Alg.6

因子数	算法	r	时间 (s)	E (LL)	σ (E)	E $(LL_{Alg.6}-LL_{Alg.2})$	σ(E $(LL_{Alg.6}-LL_{Alg.2})$)
$s=15, f=0$	算法1 (FA)	—	1.61	7.56	9.06	—	—
$s=15, f=0$	算法2 (FFA)	—	0.55	7.54	9.06	—	—
$s=0, f=22$	算法3 (OLS)	—	0.01	−108.0	15.50	—	—
$s+f=15$	算法6 (HFA$^+$)	0.6	10.82	11.96	8.18	4.40	1.64
		0.7	10.90	11.86	8.20	4.29	1.61
		0.8	10.40	11.77	8.24	4.21	1.58
		0.9	10.51	**12.00**	**8.25**	**4.44**	**1.58**

基本面因子和 4.61 个统计学因子, 而当总因子数目为 13 时平均使用约 5.79 个基本面因子和 7.21 个统计学因子. 表 3 的结果表现出无论在总因子数目为 10 还是 13 的情况下, 算法 6 对数似然估计值均值都大于其他算法, 同时由于 n 与总因子数目的比值与第二个人造数据集接近, 算法 2 相对算法 1 的加速比与第二个人造数据集实验结果相似. 从表 3 中可以看出, 当基本面因子未经过挑选时, 即直接使用算法 3 和算法 4 生成风险模型进行预测, 其表现在相同总因子数目的算法中最差, 甚至不如算法 1、2 代表的统计学因子分析, 这意味着基本面因子需要经过适当的挑选才能生成更好的风险模型. 当总因子数目为 10 时, 算法 6 的表现在 $\delta=3$ 上最好, 其对数自然估计均值达到−3 561.04, 比算法 1 的 −3 594.51 大 33.47, 同时算法 6 与算法 1 的标准差基本相同; 当总因子数目为 13 时, 算法 6 在 $\delta=5$ 时表现最好, 其对数自然估计值达到−3 522.23, 比算法 1 的−3 550.65 大 28.42, 同时算法 6 与算法 1 的标准差基本相同. 这些结果都显示出算法 6 要显著地比算法 1 表现好. 尽管算法 6 的运行时间比算法 1 略长, 增长的运行时间仍旧在可接受的范围之内, 表示算法 6 能够和算法 1 在相同的时间间隔上生成更优的风险模型.

3.3 真实数据集实验

真实数据集挑选了中国股票作为资产. 由于股市的历史数据影响深远, 因此设置半衰期 $h=200$. 在实验中挑选了 22 个包含了市值、市盈率、流动比率等的基本面因子, 设置 2010 年 1 月 3 日为第一天, 且随机挑选了 194 支股票价格信息及基本面信息相对完整的股票. 算法 6 从第 900 天到第 1 300 天累积对数似然估计向量 $\boldsymbol{v}$, 同时算法 1、算法 2 和算法 6 第 1 300 天到 1 500 天总因子数目为 15 的风险模型预测结果被记录在表 4 中, 同时加入了使用全部 22 个基本面因子的算法 3 作为基本面模型预测结果对比, 此时算法 3 和算法 6 都使用所有 22 个基本面因子的暴露矩阵作为输入. 同时表 4 中记录了算法 6 在不同衰减系数 r 下的实验结果. 图 1 为算法 1、算法 3 与算法 6 在 $r=0.9$ 时前 30 天的对数似然估计值.

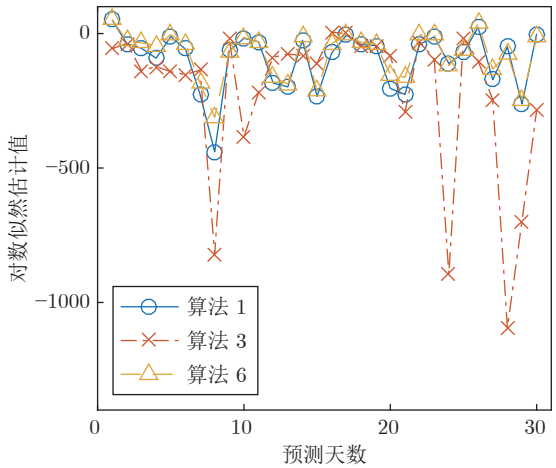

图 1 算法 1、算法 3 和算法 6 $(r=0.9)$ 前 30 天风险模型在真实数据集上预测的对数似然估计值的结果
Fig. 1 The predicted log-likelihood of the risk models estimated by Alg.1, Alg.3 and Alg.6 $(r=0.9)$ on first 30 days

算法 6 在 $r=0.6, 0.7, 0.8, 0.9$ 时分别平均使用了约 7.54、7.45、7.47、7.40 个基本面因子, 剩余的为统计学因子. 图 1 中表现出算法 6 在大部分时间的预测值是要好于算法 1 的, 并且在表现如不算法 1 的时间点相差也不大; 同时图 1 中算法 3 的效果说明使用所有的基本面因子尽管在少数时间的效果要好于算法 1 和算法 6, 但由于基本面因子没有经过挑选, 因此会出现波动巨大的情况. 表 4 中的结果显示出尽管算法 3 总因子数目最多, 但其对数似然估计值均值却远小于其他算法, 并且均值标准差大于其他算法, 显示出算法 3 生成了更坏而且更不稳定的风险模型; 同时算法 2 仍旧能比算法 1 有将近

三倍的加速比, 且对数似然估计值均值和标准差相同. 表 4 中 $\mathrm{E}(LL_{Alg.6} - LL_{Alg.2})$ 表示算法 6 与算法 2 对数似然估计值差的均值, 值越大代表算法 6 的性能越比算法 1 好, 而 $\sigma(\mathrm{E}(LL_{Alg.6} - LL_{Alg.2}))$ 表示该均值的标准差, 用来显示算法 6 优于算法 2 的显著程度. 由于真实数据的对数回报数值较小, 因此最后计算出的对数似然估计值较大. 表 4 中的结果显示出在不同 r 的取值下算法 6 的结果都要优于算法 1. 不同 r 取值的结果表现出算法 6 可通过调节 r 来改善预测结果. 当 $r = 0.9$ 时, 算法 6 拥有最大的 $\mathrm{E}(LL_{Alg.6} - LL_{Alg.2}) = 4.44$ 且 $\sigma(\mathrm{E}(LL_{Alg.6} - LL_{Alg.2}) = 1.58$, 并且算法 6 的 $\sigma(\mathrm{E})$ 为 8.25 小于算法 2 的 9.06, 显示出算法 6 显著好于算法 1 并且更稳定. 尽管算法 6 运行时间更长, 增长的时间仍在可接受范围内.

4 结论

针对金融数据中的风险模型生成问题, 本文提出了快速因子分析算法, 同时提出了结合领域知识的混合因子分析算法以及在实际应用中使用的挑选基本面因子的混合因子分析算法. 实验结果显示快速因子分析算法在人造数据集上能够达到最多 31 倍的加速比, 同时挑选基本面因子的混合因子分析算法在人造数据集和真实数据集上均有更好的表现, 并且运行时间的增长也在可接受范围内.

References

1 Alexander C. *Market Models: A guide to financial data analysis*. John Wiley & Sons, 2001

2 Alexander C. *Market Risk Analysis, Practical Financial Econometrics*. John Wiley & Sons, 2008

3 MSCI Barra. *Barra Risk Model Handbook*. MSCI Barra Applied Research, 2007, 43

4 Christoffersen P, Goncalves S. *Estimation Risk in Financial Risk Management*. CIRANO, 2004.

5 Christoffersen P, Diebold F. How relevant is volatility forecasting for financial risk management? *Review of Economic and Statistics*, 2000, **82**(1): 12−22

6 Higgins R C, Reimers M. *Analysis for Financial Management*. Number 53. Irwin Chicago, 1995

7 Smith C W, Smithson C W, Wilford D S. *Managing Financial Risk*. Irein Burr Ridge, 1995

8 Connor G. The three types of factor models: A comparison of their explanatory power. *Finanacial Analysis Journal*, 1995, **51**(3): 42−46

9 Boyer M M, Filion D. Common and fundamental factors in stock returns of Canadian oil and gas companies. *Energy Economics*, 2007, **29**(3): 428−453

10 Dechow P M, Hutton A P, Meubroek L, Sloan R G. Short-sellers, fundamental analysis, and stock returns. *Journal of financial Economics*, 2001, **61**(1): 71−106

11 Doshi-Velez F, Kim B. Towards a Rigorous science of interpretable machine learning. *arXiv preprint arXiv: 1702.08608*, 2017

12 Molnar C. *Interpretable Machine Learning: A guide for making black box models explainable* [Online]. available: https://christophm.github.io/interpretable-ml-book, 2020

13 Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT press, 2016.

14 Dempster A P. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 1977, **39**(1): 1−38

15 Rubin D and Thayer D. EM algorithms for ML factor analysis. *Psychometrika*, 1982, **47**(1): 69−76

16 Ghahramani Z, Hinton G. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Canada, 1996

17 Kaiser H. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 1960, **20**(1): 141−151

18 Roweis S, Ghahramani Z. A unifying review of linear Gaussian models. *Nueral Computation*, 1999, **11**(2): 305−345

19 Saqib U, Gannot S, Jensen J R. Estimation of Acoustic Echoes Using Expectation-Maximization Methods. *Eurasip Journal on Audio, Speech, and Music Processing*, 2020, **2020**(1): 1−15

20 Sun Z, Yang Y. An EM Approach to Non-autoregressive Conditional Sequence Generation. In: Proceedings of the 37th International Conference on Machine Learning. arXiv: 2006.16378, 2020

21 Nan Y, Quan Y, Jim H. Variational-EM-Based Deep Learning for Noise-Blind Image Deblurring. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 3626−3635

22 Xi Yan-Hui, Peng Hui, Mo Hong. Parameter Estimation of RBF-AR Model Based on the EM-EKF Algorithm. *Acta Automatica Sinica*, 2017, **43**(9): 1636−1643
(席燕辉, 彭辉, 莫红. 基于EM-EKF算法的RBF-AR模型参数估计. 自动化学报, 2017, **43**(9): 1636−1643)

23 Ma Xin-Ke, Yang Yang, Yang Kun, Luo Yi. Registration Algorithm Based on Fuzzy Shape Context and Local Vector Similarity Constraint. *Acta Automatica Sinica*, 2020, **46**(2): 342−357
(马新科, 杨扬, 杨昆, 罗毅. 基于模糊形状上下文与局部向量相似性约束的配准算法. 自动化学报, 2020, **46**(2): 342−357)

24 Yao Hong-Ge, Dong Ze-Hao, Yu Jun, Bai Xiao-Jun. Fully overlapped handwritten number recognition and separation based on deep EM capsule network. *Acta Automatica Sinica*, DOI: 10.16383/j.aas.c190849
(姚红革, 董泽浩, 喻钧, 白小军. 深度EM胶囊网络全重叠手写数字识别与分离. 自动化学报, DOI: 10.16383/j.aas.c190849)

25 Thomposon B. *Exploratory and confirmatory factor analysis: Under concepts and applications*. American Psycholohical Association, 2004

26 Guo Wu, Li Yi-Jie, Dai Li-Rong, Wang Ren-Hua. Factor Analysis and Space Assembling in Speaker Recognition. *Acta Automatica Sinica*, 2009, **35**(9): 1193−1198

(郭武, 李轶杰, 戴礼荣, 王仁华. 说话人识别中的因子分析以及空间拼接. 自动化学报, 2009, **35**(9): 1193−1198)

27 Gonzalez J A, et al. A silent speech system based on permanent magnet articulography and direct synthesis. *Computer Speech & Language*, 2016, **39**: 67−87

28 Li Y, Dixit M, Vasconcelos N. Deep scene image classification with the MFAFVNet. In: Proceedings of the 2017 IEEE International Conference on Computer Vision, 2017: 5746−5754

29 Kesteren E van, Kievit R A. Exploratory Factor Analysis with Structured Residuals for Brain Network Data. *Network Neuroscience*, 2020: 1−45

冯 栩 清华大学计算机科学与技术系博士研究生. 2017 年获得清华大学计算机科学与技术系学士学位. 主要研究方向为数值线性代数算法, 机器学习, 大数据分析.
E-mail: fx17@mails.tsinghua.edu.cn
(**FENG Xu** Ph. D. candidate in the Department of Computer Science and Technology, Tsinghua University. He received his bachelor degree from Tsinghua University in 2017. His research interest covers numerical linear algebra algorithms, machine learning, and big-data analytics.)

喻文健 清华大学计算机科学与技术系长聘教授. 2003 年获得清华大学计算机科学与技术系博士学位, 随后留校任教. 主要研究方向为集成电路计算机辅助设计算法, 机器学习, 大数据分析算法、数值计算及其应用. 本文通信作者.
E-mail: yu-wj@tsinghua.edu.cn
(**YU Wen-Jian** Professor in the Department of Computer Science and Technology, Tsinghua University. He received his Ph. D. degree in Computer Science from Tsinghua University in 2003. His research interest covers EDA algorithm and software, machine learning, big-data analytics, and numerical algorithms and applications. Corresponding author of this paper.)

李 凌 加州理工学院计算机科学博士 (辅修电子工程). 主要研究方向为机器学习, 量化投资, 自动化交易.
E-mail: liling@flowam.com
(**LI Ling** Ph. D. in computer science (minor in electrical engineering) from California Institute of Technology. His research interest covers machine learning, quantitative investing, and automated trading.)