

投资组合优化模型的一个序列凸近似算法

李卫国^{1,2}, 张宏伟^{*1}, 梁锡军³

(1. 大连理工大学 数学科学学院, 辽宁 大连 116024;

2. 辽宁地质工程职业学院, 辽宁 丹东 118303;

3. 中国石油大学(华东) 理学院, 山东 青岛 266580)

摘要:以 CVaR 为代表的凸优化投资组合模型近年来引起了广泛研究. 为克服传统投资组合模型中凸近似的不足, 提出了一个投资组合的 DC 规划模型. 该模型用一个 DC 函数替代了 CVaR 模型中的凸近似函数, 同时要求所有约束条件在概率意义下成立. 进一步地, 提出了一个序列凸近似(SCA)算法用于求解 DC 规划问题, 并运用 Monte-Carlo 方法来实现 SCA 算法. 初步的实验结果表明, 因子收益服从“尖峰厚尾”分布时, 模型的目标函数值优于采用 CVaR 近似的目标函数值.

关键词:投资组合; 序列凸近似; 凸优化; Monte-Carlo 方法

中图分类号: O224

文献标识码: A

doi: 10.7511/dllgxb201703016

0 引言

投资组合是指投资者根据其风险喜好在众多的有价证券中对风险投入进行最优投资分配. 投资组合选择问题最早由 Markowitz^[1] 作为一个最优化问题提出. 他用随机收益率的均值来衡量预期收益的好坏, 用随机收益率的方差衡量风险的大小. Markowitz 的均值-方差模型在现代投资理论中堪称经典, 并且始终在改进. 例如文献[2]对这些模型作了一个系统的总结. 只有当证券收益率服从正态分布或者投资者是风险厌恶型时, 均值-方差模型才是有效的. 但由于计算的复杂性, 在投资数额较大的优化问题中模型的使用率并不高. 本文试图改进这一模型, 研究不同的预期收益或风险衡量标准, 使得模型更符合实际, 更好地为投资者决策提供参考.

投资者经常会遇到投资项目的组合决策问题, 要考虑的因素有收益率、风险、增长潜力等条件, 希望该资产投资波动越小越好, 并进行权衡考虑获得一个最佳的投资方案. 在投资组合的优化研究过程中, 本文试图在传统的保守近似优化模

型的基础上, 克服指示函数 $I_{(0,+\infty)}(z)$ 在近似估计中的不足. 通过引入风险度量的 DC (difference of two convex functions) 近似, 给出一个投资组合的 DC 规划模型, 并提出一个序列凸近似 (sequential convex approximation, SCA) 算法来进行求解, 用一个严格凸二次函数来近似目标函数中的光滑函数, 用线性函数近似所有 DC 函数的第二个凸函数, 得到搜索方向的一系列的凸优化问题.

1 问题提出

考虑如下问题: 对 n 个资产投资以获取最大收益. 记 $\mathbf{x} = (x_1 \ x_2 \ \cdots \ x_n)^T$ 为 n 个资产的投资比例 ($x_i \geq 0, i = 1, 2, \dots, n, \sum_{i=1}^n x_i = 1$). 令 $\mathbf{x} \in \mathbf{X} := \mathbf{R}^n$. 各资产的收益 $\mathbf{r} = (r_1 \ r_2 \ \cdots \ r_n)^T$ 为随机向量, 假定其满足如下关系:

$$\mathbf{r} = \boldsymbol{\mu} + \mathbf{V}^T \boldsymbol{\xi} + \boldsymbol{\varepsilon}$$

式中: $\boldsymbol{\mu}$ 为资产收益的均值, $\boldsymbol{\mu} \in [\underline{\boldsymbol{\mu}}, \bar{\boldsymbol{\mu}}]$; $\mathbf{V}$ 为因子荷载矩阵, $\mathbf{V}^T = (V_1 \ V_2 \ \cdots \ V_n)$; $\boldsymbol{\xi}$ 为因子收益, $\boldsymbol{\xi} = (\xi_1 \ \xi_2 \ \cdots \ \xi_n)^T$ 服从 F 分布 (尖峰厚

尾); $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{D})$, $\mathbf{D}$ 为对称半正定矩阵, $\boldsymbol{\varepsilon}$ 为白噪声. 极大化所研究资产的平均收益:

$$\max E[\mathbf{r}^T \mathbf{x}] = \boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \mathbf{E}_F(\boldsymbol{\xi})$$

其约束条件被称为决策风险^[3], 用

$$\begin{aligned} \text{Prob}_{\boldsymbol{\xi} \sim F, \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{D})} \{ \mathbf{r}^T \mathbf{x} \leq a \} &\leq \beta \\ \mathbf{1}^T \mathbf{x} &= 1, \mathbf{x} \geq \mathbf{0} \end{aligned}$$

来描述, 其中 $\mathbf{1} = (1 \ 1 \ \cdots \ 1)^T \in \mathbf{R}^n$, $\beta \in (0, 1)$ 是一个较小的数, $a > 0$ 是保守收益的一个估计. 同时希望该资产的投资波动越小越好. 采用 $\text{var}(\mathbf{r}^T \mathbf{x}) = \mathbf{x}^T \mathbf{D} \mathbf{x}$, 作为投资波动的度量. 考虑如下模型:

$$\begin{aligned} \max \quad & E[\mathbf{r}^T \mathbf{x}] = \boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \mathbf{E}_F(\boldsymbol{\xi}) - \gamma \mathbf{x}^T \mathbf{D} \mathbf{x} \\ \text{s. t.} \quad & \text{Prob}\{-(\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi}) \leq -a\} \geq 1 - \beta \\ & \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (1)$$

其中 $\gamma > 0$ 是参数. 考虑模型(1)中的概率约束, 令

$$\begin{aligned} p(\mathbf{x}) &= 1 - \text{Prob}\{-(\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi}) + a \leq 0\} = \\ & \text{Prob}\{-(\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi}) + a \geq 0\} \end{aligned}$$

则约束条件

$$1 - p(\mathbf{x}) \geq 1 - \beta \Leftrightarrow p(\mathbf{x}) \leq \beta$$

$$p(\mathbf{x}) = \text{Prob}\{c(\mathbf{x}, \boldsymbol{\xi}) > 0\} = E[I_{(0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}))]$$

其中

$$c(\mathbf{x}, \boldsymbol{\xi}) := \boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V} \boldsymbol{\xi} - a$$

这里 $I_A(\cdot)$ 表示集合 A 上的指示函数:

$$I_A(z) = \begin{cases} 1; & z \in A \\ 0; & z \notin A \end{cases}$$

在 Rockafellar 等^[4]提出的 CVaR 近似中, 用

$$\psi(z, t) = [t + z]_+ / t$$

近似 $I_{(0, +\infty)}(z)$, 其中 $z > 0$, $[z]_+ = \max\{z, 0\}$.

Nemirovski 等^[5]曾经指出在所有的凸的保守近似中, CVaR 被公认为是最出色的. 但是, $\psi(z, t)$ 对于指示函数 $I_{(0, +\infty)}(z)$ 不是一个好的近似, 因为在 $z > 0$ 且 z 较大时, 两个函数差异较大. 为寻求一个更好的近似, 令

$$\phi(z, t) = \frac{1}{t} [z]_+ +$$

$$\pi(z, t) = \psi(z, t) - \phi(z, t) =$$

$$\frac{1}{t} [z + t]_+ - \frac{1}{t} [z]_+, t > 0$$

由于 $\psi(z, t)$ 和 $\phi(z, t)$ 都是 z 的凸函数, $\pi(z, t)$ 是一个关于 z 的 DC 函数. 函数 $\pi(z, t)$ 仅在区间 $(-t, 0)$ 与指示函数有差异, 在其余的区间与指示函数完全吻合. 可见, DC 函数 $\pi(z, t)$ 是指示函数

的一个更好的近似, 如图 1 所示.

令

$$g_1(\mathbf{x}, t) = E[\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi} - a + t]_+$$

$$g_2(\mathbf{x}) = g_1(\mathbf{x}, 0)$$

$$\hat{p}(\mathbf{x}, t) = \frac{1}{t} [g_1(\mathbf{x}, t) - g_2(\mathbf{x})] =$$

$$\begin{aligned} E \left[\frac{1}{t} (\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi} - a + t) - \right. \\ \left. \frac{1}{t} (\boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \boldsymbol{\xi} - a) \right] \end{aligned}$$

当 $t > 0$ 时, $\hat{p}(\mathbf{x}, t)$ 是 $p(\mathbf{x})$ 的一个保守 DC 近似.

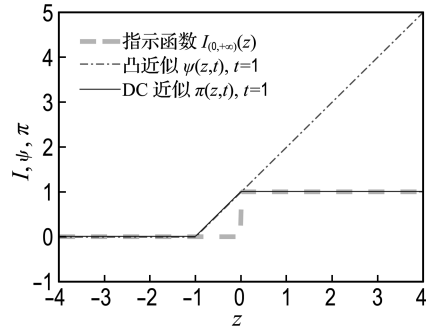


图 1 DC 近似与凸近似

Fig. 1 DC approximation and convex approximation

令 $\tilde{p}(\mathbf{x}) = \inf_{t > 0} \hat{p}(\mathbf{x}, t)$, 则 $\hat{p}(\mathbf{x}, t) \leq \beta \Rightarrow \tilde{p}(\mathbf{x})$

$\leq \beta$. 因此, $\tilde{p}(\mathbf{x})$ 是 $\hat{p}(\mathbf{x}, t)$ 所有近似中最好的保守近似.

$$\begin{aligned} \max \quad & \boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \mathbf{E}_F(\boldsymbol{\xi}) - \gamma \mathbf{x}^T \mathbf{D} \mathbf{x} \\ \text{s. t.} \quad & \tilde{p}(\mathbf{x}) \leq \beta \\ & \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (2)$$

称模型(2)为模型(1)的 DC 近似^[6].

约束条件可近似为 $\frac{g_1(\mathbf{x}, t) - g_2(\mathbf{x})}{t} \leq \beta$, 即用

下面的模型(3)近似模型(2).

$$\begin{aligned} \max \quad & \boldsymbol{\mu}^T \mathbf{x} + \mathbf{x}^T \mathbf{V}^T \mathbf{E}_F(\boldsymbol{\xi}) - \gamma \mathbf{x}^T \mathbf{D} \mathbf{x} \\ \text{s. t.} \quad & g_1(\mathbf{x}, t) - g_2(\mathbf{x}) \leq \beta t \\ & \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (3)$$

记 $\Omega(t)$ 为模型(3)的可行域:

$$\begin{aligned} \Omega(t) := \{ \mathbf{x} \in \mathbf{X}; g_1(\mathbf{x}, t) - g_2(\mathbf{x}) \leq \beta t, \\ \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \} \end{aligned}$$

算法 1 是求解模型(3)的序列凸近似算法.

算法 1 序列凸近似 (sequential convex approximation, SCA) 算法

SCA 算法具体步骤如下:

- Step 1 选取 $\mathbf{x}^0 \in \Omega(t), k=0$.
- Step 2 若 $\mathbf{x}^k$ 是最优解, 算法终止.
- Step 3 用 Monte-Carlo 方法^[6]求解下列凸规划问题:
$$\begin{aligned} \max \quad & \boldsymbol{\mu}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{V}^\top \mathbf{E}_F(\boldsymbol{\xi}) - \gamma \mathbf{x}^\top \mathbf{D} \mathbf{x} \\ \text{s. t.} \quad & g_1(\mathbf{x}, t) - [g_2(\mathbf{x}^k) + \langle \nabla g_2(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle] \leq \beta t \\ & \mathbf{1}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (4)$$
得最优解 $\mathbf{x}^{k+1}$.
- Step 4 $k := k+1$, 转 Step 2.

2 SCA 算法的实施

2.1 凸规划问题(4)的求解

依照前面的记号

$$\begin{aligned} c(\mathbf{x}, \boldsymbol{\xi}) &= \boldsymbol{\mu}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{V}^\top \boldsymbol{\xi} - a \\ g_1(\mathbf{x}, t) &= E[c(\mathbf{x}, \boldsymbol{\xi}) + t]_+ = \\ &E[I_{[0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}) + t)] \\ g_2(\mathbf{x}, t) &= g_1(\mathbf{x}, 0) \end{aligned}$$

另外, 记 $h(\mathbf{x})$ 为问题(4)的目标函数:

$$h(\mathbf{x}) := \boldsymbol{\mu}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{V}^\top \mathbf{E}_F(\boldsymbol{\xi}) - \gamma \mathbf{x}^\top \mathbf{D} \mathbf{x}$$

则有

$$\nabla_x g_1(\mathbf{x}, t) = E[\nabla_x c(\mathbf{x}, \boldsymbol{\xi}) \cdot I_{[0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}) + t)] = E[\nabla_x c(\mathbf{x}, \boldsymbol{\xi}) \cdot I_{[-t, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}))]$$

令

$$g(\mathbf{x}) = g_1(\mathbf{x}, t) - [g_2(\mathbf{x}^k) + \langle \nabla g_2(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle] - \beta t \quad (5)$$

则

$$\begin{aligned} g(\mathbf{x}) &= E[c(\mathbf{x}, \boldsymbol{\xi}) + t]_+ - E[c(\mathbf{x}, \boldsymbol{\xi})]_+ + \\ &E(\nabla_x c(\mathbf{x}^k, \boldsymbol{\xi}) \cdot I_{[0, +\infty)}(c(\mathbf{x}^k, \boldsymbol{\xi})))^\top (\mathbf{x} - \mathbf{x}^k) - \beta t \end{aligned} \quad (6)$$

记 $\xi_1, \xi_2, \dots, \xi_n$ 为随机变量 $\boldsymbol{\xi}$ 的独立同分布的样本点,

$$\bar{g}_2(\mathbf{x}^k) = \frac{1}{n} \sum_{i=1}^n [c(\mathbf{x}^k, \xi_i)]_+ \quad (7)$$

$$\bar{\nabla} g_2(\mathbf{x}^k) = \frac{1}{n} \sum_{i=1}^n \nabla_x c(\mathbf{x}^k, \xi_i) \cdot I_{[0, +\infty)}(c(\mathbf{x}^k, \xi_i)) \quad (8)$$

$$\bar{\xi} = \frac{1}{n} \sum_{i=1}^n \xi_i \quad (9)$$

对 $g(\mathbf{x})$ 的一个自然的估计是

$$\begin{aligned} \bar{g}(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n [c(\mathbf{x}, \xi_i) + t]_+ - \\ &[\bar{g}_2(\mathbf{x}^k) + \bar{\nabla} g_2(\mathbf{x}^k)^\top (\mathbf{x} - \mathbf{x}^k)] - \beta t \end{aligned} \quad (10)$$

对 $E(\boldsymbol{\xi})$ 的估计是 $\bar{\xi}$. 建议求解

$$\begin{aligned} \min \quad & \boldsymbol{\mu}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{V}^\top \bar{\xi} - \gamma \mathbf{x}^\top \mathbf{D} \mathbf{x} \\ \text{s. t.} \quad & \bar{g}(\mathbf{x}) \leq 0 \\ & \mathbf{1}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{aligned}$$

2.2 $\nabla g(\mathbf{x})$ 的近似

为了在 n 较大时有效地求解问题(4), 给出下面的方法: 由式(5)、(6)及文献[7]引理 2 知,

$$\begin{aligned} \nabla g(\mathbf{x}) &= \nabla_x g_1(\mathbf{x}, t) - \nabla g_2(\mathbf{x}^k) = \\ &E[\nabla_x c(\mathbf{x}, \boldsymbol{\xi}) \cdot I_{[0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}) + t)] - \\ &E[\nabla_x c(\mathbf{x}^k, \boldsymbol{\xi}) \cdot I_{[0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}))] = \\ &E[\nabla_x c(\mathbf{x}, \boldsymbol{\xi}) \cdot I_{[-t, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}))] - \\ &E[\nabla_x c(\mathbf{x}^k, \boldsymbol{\xi}) \cdot I_{[0, +\infty)}(c(\mathbf{x}, \boldsymbol{\xi}))] \end{aligned}$$

则 $\nabla g(\mathbf{x})$ 可以近似为

$$\begin{aligned} \bar{\nabla} g(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n \nabla_x c(\mathbf{x}, \xi_i) \cdot I_{[-t, +\infty)}(c(\mathbf{x}, \xi_i)) - \\ &\frac{1}{n} \sum_{i=1}^n \nabla_x c(\mathbf{x}^k, \xi_i) \cdot I_{[0, +\infty)}(c(\mathbf{x}^k, \xi_i)) \end{aligned}$$

当 n 较大时, 用 $\bar{\nabla} g(\mathbf{x})$ 作为 $\nabla g(\mathbf{x})$ 的近似, 并直接用基于梯度^[8]的方法求解问题(4). 该方法可以视为用近似的 $g(\mathbf{x})$ 、 $\nabla g(\mathbf{x})$ 及 $E(\boldsymbol{\xi})$ 直接求解问题(4). 当采用基于梯度的方法求解问题(4)时, 样本点仅用于计算 $\bar{g}(\mathbf{x})$ 和 $\bar{\nabla} g(\mathbf{x})$ 的值. 所需的计算量是 $O(n)$. 这是求解问题(4)的一种较快的方法.

2.3 初始点的选取

为执行 SCA 算法, 需要选取初始点 $\mathbf{x}^0 \in \Omega(t)$, 给出两种选取办法.

第一种, 记

$$\Omega_0(t) = \{\mathbf{x} \in \mathbf{X} : g_1(\mathbf{x}, t) \leq \beta t, \mathbf{1}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$$

注意到, $g_2(\mathbf{x}) = E[\boldsymbol{\mu}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{V}^\top \boldsymbol{\xi} - a]_+ \geq 0, \forall \mathbf{x} \in \mathbf{X}$. 故 $\Omega_0(t) \subseteq \Omega(t)$. 另外, 由于 $g_1(\mathbf{x}, t)$ 是关于 $\mathbf{x}$ 的凸函数, 故 $\Omega_0(t)$ 是凸集合. 于是,

$$\begin{aligned} \min \quad & h(\mathbf{x}) \\ \text{s. t.} \quad & \mathbf{x} \in \Omega_0(t) \end{aligned}$$

是凸规划. 令 $\mathbf{x}_\epsilon = \arg \min \{h(\mathbf{x}) : \mathbf{x} \in \Omega_0(t)\}$, 则 $\mathbf{x}_\epsilon \in \Omega(t)$.

第二种, 令 $\Omega_{\text{CVaR}} = \{\mathbf{x} \in \mathbf{X} : \text{CVaR}_{1-\beta} c(\mathbf{x}, \boldsymbol{\xi}) \leq$

$0\}$. 依据前面的讨论, $\Omega_{\text{CVaR}} = \left\{ \mathbf{x} \in \mathbf{X} : \inf_{t \geq 0} \frac{1}{t} g_1(\mathbf{x}, t) \leq \beta \right\}$. 令 $\mathbf{x}_{\text{CVaR}} \in \arg \min \{h(\mathbf{x}) : \mathbf{x} \in \Omega_{\text{CVaR}}, \mathbf{1}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$, 即 Rockafellar 和 Uryasev 给出的 CVaR 近似的最优解.

令 $t^* = q_{1-\beta}(c(\mathbf{x}_{\text{CVaR}}, \boldsymbol{\xi}))$, 即 $c(\mathbf{x}_{\text{CVaR}}, \boldsymbol{\xi})$ 的 $1-\beta$ 分位数. 由文献[9], $t^* > 0$ 且

$$\inf_{t>0} \frac{1}{t} g_1(\mathbf{x}_{\text{CVaR}}, t) = \frac{1}{t^*} g_1(\mathbf{x}_{\text{CVaR}}, t^*)$$

于是, $\mathbf{x}_{\text{CVaR}} \in \{\mathbf{x} \in \mathbf{X}: g_1(\mathbf{x}, t^*) \leq t^* \beta\}$. 由于 $g_2(\mathbf{x}) \geq 0$, 故 $\mathbf{x}_{\text{CVaR}} \in \{\mathbf{x} \in \mathbf{X}: g_1(\mathbf{x}, t^*) - g_2(\mathbf{x}) \leq t^* \beta\} = \boldsymbol{\Omega}(t^*)$. 对于任意 $t \in (0, t^*]$, $\boldsymbol{\Omega}(t^*) \subseteq \boldsymbol{\Omega}(t)$. 因此, $\mathbf{x}_{\text{CVaR}} \in \boldsymbol{\Omega}(t)$, $0 < t \leq t^*$. 可以首先求解 CVaR 近似问题得到 $\mathbf{x}_{\text{CVaR}}$, 然后选取 $t \in (0, t^*]$, 则 $\mathbf{x}_{\text{CVaR}}$ 可以用作 SCA 算法的初始点. 若使用 $\mathbf{x}^0 = \mathbf{x}_{\text{CVaR}}$, 则 SCA 算法产生的迭代点列 $\{x_k\}$ 将改进或至少等价于 CVaR 近似解.

(1) CVaR 近似解的计算

$$\mathbf{x}_{\text{CVaR}} = \arg \min \{h(\mathbf{x}): \text{CVaR}_{1-\beta} c(\mathbf{x}, \boldsymbol{\xi}) \leq 0, \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\} \tag{11}$$

(2) CVaR 分位数的计算

记 L 是一随机损失, $F_L(y) = \text{Prob}\{L \leq y\}$ 是 L 的累积分布函数. 对于任意 $\alpha \in (0, 1)$, 则损失 L 的 α -VaR 定义为 $v_\alpha = F_L^{-1}(\alpha)$, α -CVaR 定义为 $c_\alpha = 1/(1-\alpha) \cdot \int_\alpha^1 F_L^{-1}(\beta) d\beta$. 由文献[10]知道 $c_\alpha = v_\alpha + \frac{1}{1-\alpha} E[L - v_\alpha]_+ = E[L(\theta) | L(\theta) \geq v_\alpha(\theta)]$

令 $L_1, L_2, \dots, L_n$ 是损失 L 的 n 个独立同分布的观测, 则 L 的 α -VaR 可以估计为

$$\widehat{v}_\alpha^n = L_{[n\alpha; n]}$$

其中 $L_{[i; n]}$ 是排序后的 n 个观测中的第 i 个观测.

L 的 α -CVaR 可以估计为

$$\widehat{c}_\alpha^n = \widehat{v}_\alpha^n + \frac{1}{n(1-\alpha)} \sum_{i=1}^n [L_i - \widehat{v}_\alpha^n]_+ \tag{12}$$

(3) CVaR 分位数的梯度估计

假设随机损失 L 是参数 θ 的函数, 记为 $L(\theta)$. $L(\theta)$ 关于 θ 可微. 对任意的 θ , 设 $v_\alpha(\theta)$ 、 $c_\alpha(\theta)$ 是 $L(\theta)$ 的 α -VaR 和 α -CVaR, $0 < \alpha < 1$, 它们都是 θ 的函数.

依据文献[10]定理 3.1, 在适当的条件下, 对于任意的 θ ,

$$c'_\alpha(\theta) = E[L'(\theta) | L(\theta) \geq v_\alpha(\theta)] \tag{13}$$

记 $(L_1, D_1), (L_2, D_2), \dots, (L_n, D_n)$ 为 $(L(\theta), L'(\theta))$ 的 n 个独立同分布的观测. 提出 $c'_\alpha(\theta)$ 的估计:

$$\bar{\Gamma}_n = \frac{1}{n(1-\alpha)} \sum_{j=1}^n D_j \cdot \mathbf{1}_{L_j \geq \widehat{v}_\alpha^n}$$

3 实 验

采用下面的数据进行测试: 各资产的平均收益 $\boldsymbol{\mu} = (1.0 \ 1.2 \ 1.4 \ 1.6 \ 1.8)^T$, 期望收益 $\alpha = 1.5$, 因子荷载矩阵

$$\mathbf{V} = \begin{pmatrix} 0.400 & 0 & -0.800 & 0 & 0.400 & 0 \\ -0.400 & 0 & -0.400 & 0 & 0.800 & 0 \\ 0.714 & 0 & -0.714 & 0 & 0 & 0 \\ -0.714 & 0 & 0 & 0 & 0.714 & 0 \\ 0 & -0.714 & 0 & 0.714 & 0 & 0 \end{pmatrix}$$

方差-协方差矩阵

$$\mathbf{D} = 0.1 \cdot \begin{pmatrix} 2 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 2 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 1 & 0 \end{pmatrix}$$

实验测试 $\boldsymbol{\xi} = (\xi_1 \ \xi_2 \ \dots \ \xi_p)^T$ 服从如下两种分布的情形. (1) $\boldsymbol{\xi}$ 服从 F 分布: $\xi_i \sim F(i+2, i+4)$, $i=1, 2, \dots, p$; (2) $\boldsymbol{\xi}$ 服从正态分布: $\xi_i \sim N(\pi_i, \rho_i^2)$, $i=1, 2, \dots, p$, 为了比对两种情形, 取均值 π_i 、方差 ρ_i^2 与情形(1)中相应的 F 分布的均值和方差相同.

序列凸近似算法的参数设为 $\gamma = 1.0$, $\beta = 0.05$, $t = 0.01$. 算法在 Matlab 2012a 平台下编程实现, 在 Intel Core i7-4770 CPU 3.40 GHz, 8 GB RAM 计算机上执行.

实验分别测试了 ϵ -近似解和 CVaR 近似得到的解作为 SCA 算法的初始点. 随机变量 $\boldsymbol{\xi}$ 的抽样次数为 1.0×10^5 . 图 2 中给出了目标函数值 f 随迭代次数 k 的变化, $\boldsymbol{\xi}$ 分别服从 F 分布和正态分布. 可以看出, 无论 $\boldsymbol{\xi}$ 服从 F 分布还是正态分布, CVaR 初始点处的目标函数值大于 ϵ -近似解处的目标函数值, 并且在两种不同的初始点下, 目标函数值随迭代次数的增加而增加, 最终近乎收敛于共同的函数值. $\boldsymbol{\xi}$ 服从 F 分布时(图 2(a)), 算法改进了 ϵ -近似的目标函数值(1.605 6)和 CVaR 近似的目标函数值(1.608 0); 算法收敛时的目标函数值为 1.621 2. 两种初始点计算出的最优解皆为 $\mathbf{x}^* = (0 \ 0 \ 0.171 \ 2 \ 0 \ 0.828 \ 8)^T$. 可见模型选择了第 3 个和第 5 个资产进行资产配置. $\boldsymbol{\xi}$ 服从正态分布时(图 2(b)), 算法改进了 ϵ -近似的

目标函数值(1.618 2)和 CVaR 近似的目标函数值(1.618 3);算法收敛时的目标函数值为1.618 7.两种初始点计算出的最优解皆为 $\mathbf{x}^* = (0 \ 0 \ 0.203 \ 8 \ 0 \ 0.796 \ 2)^T$.可见模型仍选择了第3个和第5个资产进行资产配置,但资产配置比例与 ξ 服从 F 分布时的情形有差别.

为测试抽样规模的影响,实验比较了不同抽样规模下的目标函数值以及计算时间(单位:s),

如图3所示(ξ 服从 F 分布).在每个抽样规模下,进行了100次重复实验,目标函数值和计算时间取其平均值,图中画出了其标准差(每个柱线段的长度是相应标准差的2倍).由图3(a)可以看出,目标函数值的标准差在抽样规模 N 大于 0.5×10^5 时趋于稳定(标准差小于 1.0×10^{-3});图3(b)显示,计算时间随抽样规模的增长而增加. ξ 服从正态分布时有类似的规律,见图4.

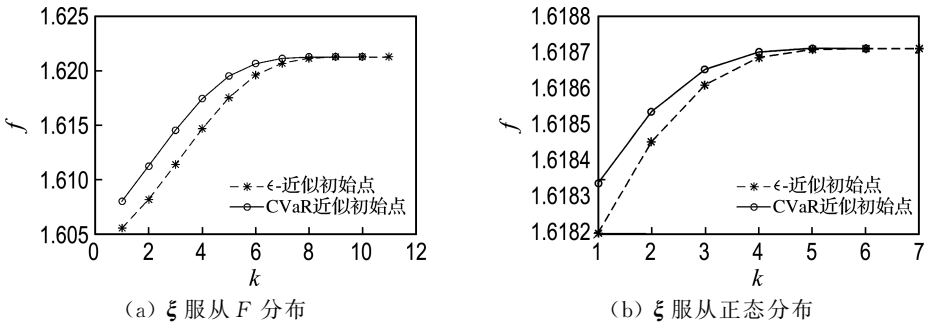


图2 目标函数值随迭代次数 k 的变化

Fig. 2 Variations of objective function values with iteration number k

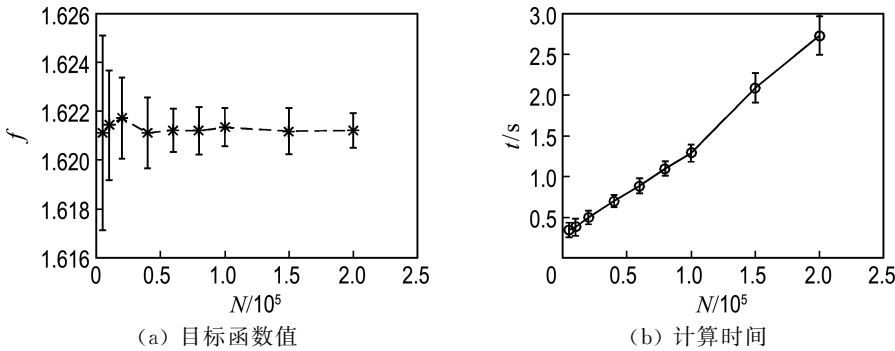


图3 不同抽样规模下的目标函数值以及计算时间(ξ 服从 F 分布)

Fig. 3 Objective function values and computing time with various sampling sizes (ξ satisfies F -distribution)

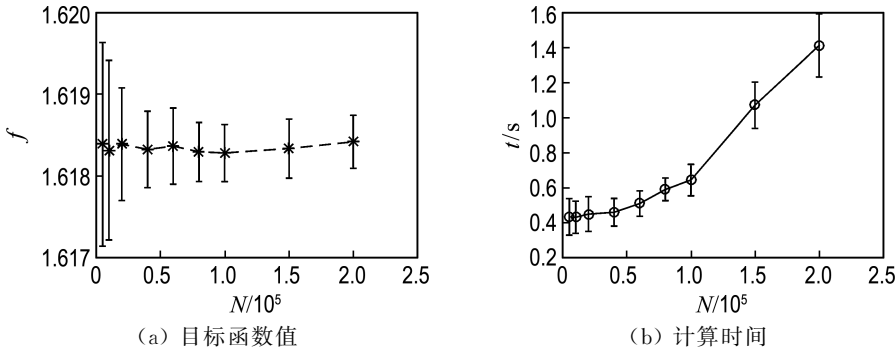


图4 不同抽样规模下的目标函数值以及计算时间(ξ 服从正态分布)

Fig. 4 Objective function values and computing time with various sampling sizes (ξ satisfies normal distribution)

4 结 语

本文考虑因子收益 ξ 服从“尖峰厚尾”分布的情形,例如 F 分布,用 DC 函数来近似概率约束中的指示函数,将资产投资组合问题建模为一个 DC 规划,并提出了求解该 DC 规划的序列凸近似(SCA)算法.该算法通过迭代求解一系列凸二次规划问题,来计算原问题最优解.通过所提出的梯度近似计算方法,减少了算法的计算量.初步的数值实验结果表明,所提出的模型及 SCA 算法是有效的.因子收益 ξ 服从 F 分布时,目标函数值优于采用 CVaR 近似的目标函数值.

参考文献:

- [1] MARKOWITZ H M. Portfolio selection [J]. *Journal of Finance*, 1952, **7**(1):77-91.
- [2] MITRA G, KYRIAKIS T, LUCAS C, *et al.* A review of portfolio planning: models and systems [J]. *Advances in Portfolio Construction and Implementation*, 2003, **26**(1):1-39.
- [3] GOLDFARB D, IYENGAR G. Robust portfolio selection problems [J]. *Mathematics of Operations Research*, 2003, **28**(1):1-38.
- [4] ROCKAFELLAR R T, URYASEV S. Optimization

- of conditional value-at-risk [J]. *Journal of Risk*, 2000, **2**(3):21-40.
- [5] NEMIROVSKI A, SHAPIRO A. Convex approximations of chance constrained programmes [J]. *SIAM Journal on Optimization*, 2006, **17**(4):969-996.
- [6] HONG L J, YANG Yi, ZHANG Liwei. Sequential convex approximations to joint chance constrained programmes: a Monte Carlo approach [J]. *Operations Research*, 2011, **59**(3):617-630.
- [7] GLASSERMAN P. *Monte Carlo Methods in Financial Engineering* [M]. New York: Springer, 2004.
- [8] FREUND R M. Subgradient optimization, generalized programming, and nonconvex duality [R]. Cambridge: Massachusetts Institute of Technology, 2004.
- [9] PFLUG G C. Some remarks on the value-at-risk and the conditional value-at-risk [M] // *Probabilistic Constrained Optimization: Methodology and Applications*. New York: Springer US, 2000:272-281.
- [10] HONG L J, LIU Guangwu. Simulating sensitivities of conditional value-at-risk [J]. *Management Science*, 2009, **55**(2):281-293.

A sequential convex approximation algorithm for portfolio optimization model

LI Weiguo^{1,2}, ZHANG Hongwei^{*1}, LIANG Xijun³

(1. School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China;

2. Liaoning Geology Engineering Vocational College, Dandong 118303, China;

3. College of Science, China University of Petroleum, Qingdao 266580, China)

Abstract: CVaR has drawn extensive attentions as a representative convex optimization portfolio model in recent years. To overcome the limits of convex approximations in traditional portfolio models, a DC programming model for portfolio is proposed. In the proposed programming model, a DC function is used as a surrogate for the convex approximation function in the CVaR model. All the constraints are satisfied in the probabilistic sense in the DC programming problem. Moreover, a sequential convex approximation (SCA) algorithm is designed to solve the DC programming problem. The SCA algorithm is implemented by employing Monte-Carlo method. Preliminary experimental results have shown that the objective function values of the DC programming are better than those with CVaR approximation when the income factors satisfy "high peak and fat tail" distributions.

Key words: portfolio; sequential convex approximation; convex optimization; Monte-Carlo method