

# 大規模な CVaR 最小化に対する平滑化アプローチ Smoothing approaches to large-scale CVaR minimization

経営システム工学専攻 田邊 亮太

## 1 研究の背景と目的

金融分野において、株式などのリスク資産に投資するときの証券の組み合わせをポートフォリオと呼び、資産投資を行う際に投資家にとって最適なポートフォリオを見つける問題のことをポートフォリオ選択問題という。資産投資において株価の動きは不確実であり、予期することは難しい。その中で、投資者は損失を冒す可能性が少なくなるような投資をしたいと考える。そのための方法の1つとして複数資産への分散投資が基本であると言われている。たとえ、ある1つの資産が大きな損失を生んだ時に、他の資産でその損失分を賄える可能性があるからである。しかし、ただ資産を分散させることではこの問題は解決することは出来ない。それぞれの資産に対してリスクとリターンの2つの指標について考える必要がある。ただ、リスクを最小にしながらリターンが最大となるようなポートフォリオを得ることは難しい。リターンとリスクの関係性を考慮し、問題を解く必要がある。今回はその中でもリスクの減少に着目し、下方リスク尺度である Conditional-Value at Risk (CVaR) [?] の最小化問題について考える。

またこの問題を考える際に、信頼性のある投資比率を得るためには、多くの資産数、多くのシナリオ数の問題に対して解く必要がある。しかし、CVaR はシナリオ数が増加すると計算時間も増加する欠点がある。そこで本研究では大規模な CVaR 最小化問題に対して平滑化アプローチを行い、複数のアルゴリズムで解き、計算時間と得られる期待損失を比較することを目的とする。

## 2 CVaR 最小化問題

CVaR は VaR を上回る損失の期待値を求める指標で図??の網掛けの部分の期待値である。

このとき CVaR 最小化問題は次のように定式化される。

$$\begin{aligned} \min_{C, \mathbf{x}} \quad & C + \frac{1}{(1-\alpha)M} \sum_{j=1}^M \max\{-\mathbf{R}_j^T \mathbf{x} - C, 0\} \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \dots, n) \end{aligned}$$

- $M$ : シナリオ数 (期間数)
- $n$ : 資産数

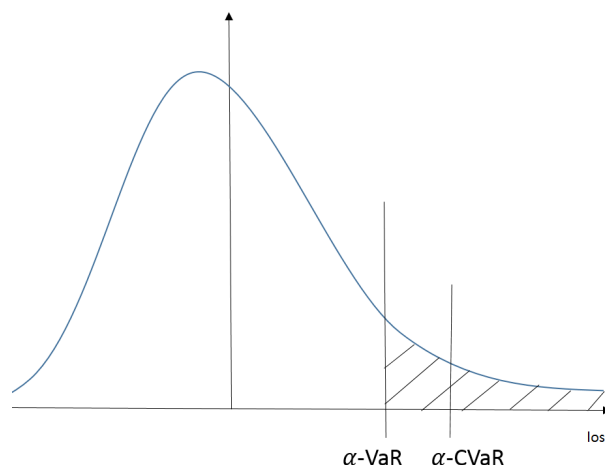


図 1: VaR と CVaR

- $\alpha$ : 信頼水準 ( $0 < \alpha < 1$ )

等式制約は予算制約、不等式制約は空売りを禁止する制約である。

また信頼水準  $\alpha$  は 0 に近い値を代入することが多い。個々のシナリオに対応して変数  $Z_j$  を導入すると定式化は次のように書き換えることができる。

$$\begin{aligned} \min_{C, Z_j} \quad & C + \frac{1}{(1-\alpha)M} \sum_{j=1}^M Z_j \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \dots, n) \\ & Z_j \geq -\mathbf{R}_j^T \mathbf{x} - C \quad (j = 1, \dots, M) \\ & Z_j \geq 0 \quad (j = 1, \dots, M) \end{aligned}$$

これにより、目的関数の非線形項を削除し、CVaR 最小化問題は線形計画問題で表すことが出来た。

## 3 平滑化 CVaR 最小化問題

CVaR 最小化問題は目的関数の max 関数が非線形であることによって非平滑の確率的最適化問題である。この max 関数に平滑化近似を行うことで CVaR 最小化問題を平滑化近似した CVaR 最小化問題として書き換える。

$$\min_{\mathbf{x}, C \in \mathbb{R}^n \times \mathbb{R}} \quad C + \frac{1}{M(1-\alpha)} \sum_{j=1}^M \hat{p}_\epsilon(-\mathbf{R}_j^T \mathbf{x} - C)$$

- $\hat{p}_\epsilon(x)$ : max 関数の平滑化近似 ( $\mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}$ )

- $\epsilon$ : 平滑化パラメータ

また今回使用する平滑化 max 関数は以下のものである。

- Chen-Harker, Kanzow, Smale の平滑化 max 関数 [?]

$$\hat{p}_\epsilon(x) = \frac{x + \sqrt{x^2 + 4\epsilon^2}}{2} \quad (1)$$

この平滑化関数は次のような概形で表される。

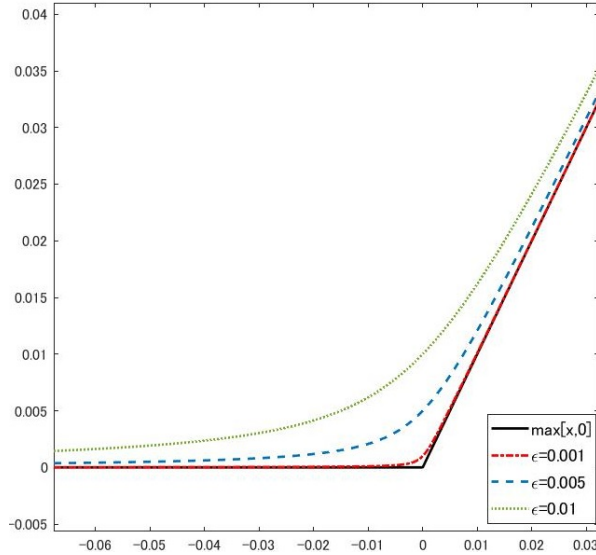


図 2:  $\epsilon$  の変化による平滑化関数の違い

図??より  $\epsilon$  がより 0 に近い値になると max 関数に近い概形になる。平滑化 CVaR 最小化問題は以下のように定式化出来る [?].

$$\begin{aligned} \min_{\mathbf{x}, C} \quad & C + \frac{1}{2M(1-\alpha)} \sum_{j=1}^M (-\mathbf{R}_j^T \mathbf{x} - C) \\ & + \sqrt{(-\mathbf{R}_j^T \mathbf{x} - C)^2 + 4\epsilon^2} \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \dots, n) \end{aligned}$$

これは非線形凸計画問題として解くことが出来る。この平滑化 CVaR 最小化問題を解くアルゴリズムとして近接勾配法を考える。

### 3.1 近接勾配法

近接勾配法は制約のない凸計画問題であり次の定式化によって表される。

$$\underset{\mathbf{x}, C}{\text{minimize}} \quad f(\mathbf{x}, C) = g(\mathbf{x}, C) + h(\mathbf{x}, C)$$

$$* f: \mathbb{R}^{n+1} \rightarrow \mathbb{R} \cup \{+\infty\}$$

$$* \text{dom} f = \{(\mathbf{x}, C) \in \mathbb{R}^{n+1}: f(\mathbf{x}, C) < +\infty\}$$

$$* g: \text{微分可能な凸関数} (\text{dom } g = \mathbb{R}^{n+1})$$

$$* h: \text{微分不可能でもよいが特殊な構造を持つ凸関数 (指示関数や L1 ノルムなど)} (\text{dom } h = \mathbb{R}^{n+1})$$

$g$  に対して勾配射影法,  $h$  に対して近接点法を適用したものが近接勾配法である。

平滑化 CVaR 最小化問題に近接勾配法を適用するためには  $g(\mathbf{x}, C)$  と  $h(\mathbf{x}, C)$  をそれぞれ定義する必要がある。

閉凸集合  $D$  を

$$D = \{(\mathbf{x}, C) \mid \sum_{i=1}^n x_i = 1, x_i \geq 0 (i = 1, \dots, n)\}$$

と定義したとき,  $h(\mathbf{x}, C)$  を閉凸集合  $D$  の指示関数で与え, 次のように表される。

$$h(\mathbf{x}, C) = \begin{cases} 0 & (\mathbf{x}, C) \in D \\ +\infty & \text{otherwise} \end{cases}$$

また,  $g(\mathbf{x}, C) = C + \frac{1}{M(1-\alpha)} \sum_{j=1}^M \hat{p}_\epsilon(-\mathbf{R}_j^T \mathbf{x} - C)$  とすることで近接勾配法が使える形へ変形出来た。

これをもとに  $(\mathbf{x}, C)$  の更新式は次の子問題で表される。

$$\begin{aligned} & (\mathbf{x}^{(k+1)}, C^{(k+1)}) \\ & = \arg \min_{\mathbf{s}, v} (h(\mathbf{s}, v) + \nabla g(\mathbf{x}^{(k)}, C^{(k)})^T \begin{pmatrix} \mathbf{s} \\ v \end{pmatrix} \\ & \quad + \frac{1}{t} B_\psi((\mathbf{s}, v), (\mathbf{x}^{(k)}, C^{(k)}))) \end{aligned}$$

ここで  $t$  はステップサイズ ( $t > 0$ ),  $B_\psi$  は Bregman 距離 [?] であり次のように定義される。

Bregman 距離 [?]

$\psi$  を微分可能な (狭義) 凸関数とする。

$$B_\psi(x, y) = \psi(x) - \psi(y) - \nabla \psi(y)^T (x - y)$$

- 性質 1 :  $B_\psi(x, y) \geq 0$

- 性質 2 :  $B_\psi(x, x) = 0$

今回, Bregman 距離は  $\psi(\mathbf{x}, C) = \sum_{i=1}^n x_i \ln x_i + \frac{1}{2} C^2$  としたとき,

$$\begin{aligned} & B_\psi((\mathbf{s}, v), (\mathbf{x}^{(k)}, C^{(k)})) \\ & = \sum_{i=1}^n (s_i \ln \frac{s_i}{x_i^{(k)}} - s_i + x_i^{(k)}) + \frac{1}{2} (v - C^{(k)})^2 \end{aligned}$$

で与える。

これにより, この子問題を解くと  $(\mathbf{x}, C)$  の更新式は

$$\begin{aligned} x_i^{(k+1)} & = \frac{x_i^{(k)} \exp(-t \nabla_x g(x_i^{(k)}, C^{(k)}))}{\sum_{i=1}^n x_i^{(k)} \exp(-t \nabla_x g(x_i^{(k)}, C^{(k)}))} \\ C^{(k+1)} & = C^{(k)} - t \nabla_C g(x_i^{(k)}, C^{(k)}) \end{aligned}$$

となる.

ここで近接勾配法のアルゴリズムを次のように示す.

近接勾配法のアルゴリズム

1. 初期点  $\mathbf{x}^{(1)}, C^{(1)}$  を設定する
2. 更新式を用いて  $(\mathbf{x}, C)$  を更新する
3. 許容誤差を満たすなら break, 満たさないなら 2 へ戻る

### 3.2 加速付近接勾配法

近接勾配法の計算時間をさらに早くするために, 近接勾配法の更新式を 1 つ前の更新によって得られた  $(\mathbf{x}, C)$  だけでなく, それ以前の更新によって得られた  $(\mathbf{x}, C)$  を用いるアルゴリズムとして加速付近接勾配法 [?] がある.

加速付近接勾配法は以下の加速式を用いる.

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{s}} (\nabla_{\mathbf{x}} g(\mathbf{y}^{(k)})^T \mathbf{s} + \frac{1}{t} B_{\psi}(\mathbf{s}, \mathbf{y}^{(k)}))$$

$$\mathbf{y}^{(k+1)} = \mathbf{x}^{(k+1)} + \beta^{(k)}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

$$\beta^{(k)} = \frac{k-2}{k+1}$$

## 4 数値実験

今回 2 つのデータを用いて CVaR 最小化と平滑化 CVaR 最小化の計算速度と精度の差を比較する.

使用したデータはそれぞれ

1. French の Data Library の 6\_Portfolios\_daily のデータ
2. French の Data Library の 25\_Portfolios\_daily のデータ

を使用した. それぞれ期間は 1926 年の 7 月 1 日~2017 年 7 月 31 日の 24034 期間, 資産数はそれぞれ 6 個と 25 個である.

このデータに対して,

1. CVaR 最小化問題のアルゴリズムによる計算時間と期待損失の比較
2. CVaR 最小化問題と平滑化 CVaR 最小化問題 (内点法, 近接勾配法, 加速付近接勾配法) の解の精度と計算速度の比較
3. CVaR 最小化問題における期間数の差における計算時間の変化

について比較を行う.

その他の条件を以下に示す.

- 信頼水準  $\alpha=0.95$
- ステップサイズ  $t = 0.8$
- 許容誤差  $\epsilon = 2.5$

平滑化パラメーター  $\epsilon$  はその都度表記する.

結果はそれぞれ次のようになった.

### 4.1 資産数 6, シナリオ数 24034 のデータについて

CVaR 最小化問題の 3 つのアルゴリズムについて比較する.

表 1: 3 つのアルゴリズムの比較

	双対単体法	レガシ内点法	内点法
$C$	1.5306	1.5306	1.5314
$x_1$	0	0	0
$x_2$	0.2237	0.2237	0.2232
$x_3$	0	0	0
$x_4$	0.5508	0.5508	0.5511
$x_5$	0.2255	0.2255	0.2256
$x_6$	0	0	0
最適値	2.5183	2.5183	2.5191
計算時間	17.20 秒	119.68 秒	14.68 秒

レガシ内点法とは主双対内点法である Mehrotra 予測子修正子アルゴリズムである.

表 2 より, 期待損失の良さは 双対単体法 = レガシ内点法 > 内点法 となった. しかし計算時間は 内点法 > 双対単体法 > レガシ内点法 となった.

次に, CVaR 最小化問題と平滑化 CVaR 最小化問題について比較を行う. 平滑化パラメーター  $\epsilon = 0.0005$

表 2: CVaR と平滑化 CVaR の比較

	双対単体法 (CVaR)	内点法 (平滑化 CVaR)	近接勾配法	加速付近接勾配法
$C$	1.5306	1.5302	1.5230	1.6000
$x_1$	0	0	0.0036	0.0965
$x_2$	0.2237	0.2235	0.2417	0.2136
$x_3$	0	0	0.0360	0.0737
$x_4$	0.5508	0.5502	0.3388	0.3183
$x_5$	0.2255	0.2263	0.3445	0.2739
$x_6$	0	0	0.0035	0.0240
最適値	2.5183	2.5184	2.5392	2.5726
計算時間	17.198 秒	0.1932 秒	0.2558 秒	0.1137 秒

である.

表 ?? より, 期待損失の良さは 内点法 > 近接勾配法 > 加速付近接勾配法 となった. 計算時間は 加速付近接勾配法 > 内点法 > 近接勾配法 となった.

## 4.2 資産数 25, シナリオ数 24034 のデータについて

資産数 25, シナリオ数 24034 のデータに対して CVaR 最小化問題の 3 つのアルゴリズムを比較する。

表 3: CVaR 最小化の比較

	双対単体法	レガシ内点法	内点法
最適値	2.4725	2.4725	2.4728
計算時間	25.820 秒	238.696 秒	22.225 秒

表??は資産数 25 の時の CVaR 最小化問題の 3 つのアルゴリズムの比較である。計算時間、期待損失を比較すると資産数 6 の問題と同じ結果になった。ただし資産数 6 の問題に比べて、内点法の期待損失の誤差はかなり小さくなった。

次に CVaR 最小化問題の内点法アルゴリズムと平滑化 CVaR 最小化問題の 3 つのアルゴリズム（内点法、近接勾配法、加速付近接勾配法）の計算速度と期待損失を比較する。

表 4: CVaR と平滑化 CVaR の比較

	内点法 (CVaR)	内点法 (平滑化 CVaR)	近接勾配法	加速付近接勾配法
最適値	2.4728	2.4726	2.5122	2.5665
計算時間	22.225 秒	6.024 秒	3.506 秒	1.557 秒

平滑化パラメータ  $\epsilon = 0.0005$  である。

表??は、計算時間は資産数 6 のときと変わり、近接勾配法のほうが内点法と比較して早く解けた。より大規模な資産数の問題を解く際にさらに計算時間の差は顕著になると考えられる。

## 4.3 シナリオ数の変化による比較

次にシナリオ数の変化による CVaR 最小化問題の計算時間についてそれぞれ資産数 6 と 25 の時を比較する。平滑化 CVaR 最小化の平滑化パラメータは  $\epsilon = 0.0005$

表 5: 資産数 6 のときの計算時間の比較

	シナリオ数 3000	シナリオ数 24034
CVaR 最小化	0.2387 秒	17.198 秒
平滑化 CVaR 最小化	0.0698 秒	0.1932 秒

である。また用いたアルゴリズムはそれぞれ CVaR 最小化が双対単体法、平滑化 CVaR 最小化が内点法である。

次に資産数 25 の時を比較する。平滑化 CVaR 最小化の平滑化パラメータは  $\epsilon = 0.0005$  である。また用いたアルゴリズムはそれぞれ CVaR 最小化が双対単体法、平滑化 CVaR 最小化が内点法である。

表??, 表??より、資産数 6 の時と同様でシナリオ数が増えると CVaR 最小化問題の計算時間は急激に増加

表 6: 資産数 25 のときの計算時間の比較

	シナリオ数 3000	シナリオ数 24034
CVaR 最小化	0.3989 秒	22.225 秒
平滑化 CVaR 最小化	0.5672 秒	6.024 秒

する。また、資産数 6 と 25 のときの計算時間を比較すると CVaR 最小化問題は資産数の増減の影響を受けにくい、平滑化 CVaR 最小化は影響を受けやすく、シナリオ数が 3000 のときを見ると、CVaR 最小化より計算時間がかかることが分かる。

## 5 まとめと今後

本研究では大規模スケールの CVaR 最小化問題に対して、平滑化アプローチを行い、計算時間と期待損失を比較した。CVaR 最小化問題はシナリオ数が増加することにより、制約の本数も増加することから計算時間が著しく増加した。しかし、資産数の増加に対して計算時間が大幅に増加することはなかった。CVaR 最小化問題に対し、ある程度の期待損失を保ちつつ、計算時間を減少させることを目的として平滑化アプローチを行った。その結果、大規模な問題に対し計算時間の大幅な減少を見ることが出来た。期待損失は平滑化近似の影響で元の CVaR 最小化問題には劣るが比較的良好な値が得られた。このことから大規模スケールの CVaR 最小化問題に対して、平滑化アプローチは有用であることが言えた。

また CVaR 最小化問題は  $\mu$ -SVM と見なすことが出来ることから金融分野以外にも応用が利くと考えられる。

## 参考文献

- [1] R. T. Rockafellar, and S. Uryasev, Optimization of Conditional value at Risk, Journal of Risk, 2(3), 21–41,2000.
- [2] B. Chen, and P. T. Harker, A non-interior-point continuation method for linear complementarity problems, SIAM Journal on Matrix Analysis and Applications, 1168–1190, 1993.
- [3] L. M. Bregman, The relaxation method of finding the common point convex sets and its application to the solution of problems in convex programming, USSR Computational Mathematics and Mathematical Physics,200–217,1967 .
- [4] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization.SIAM Journal on Optimization, 697–725, 2006.