10

第10章

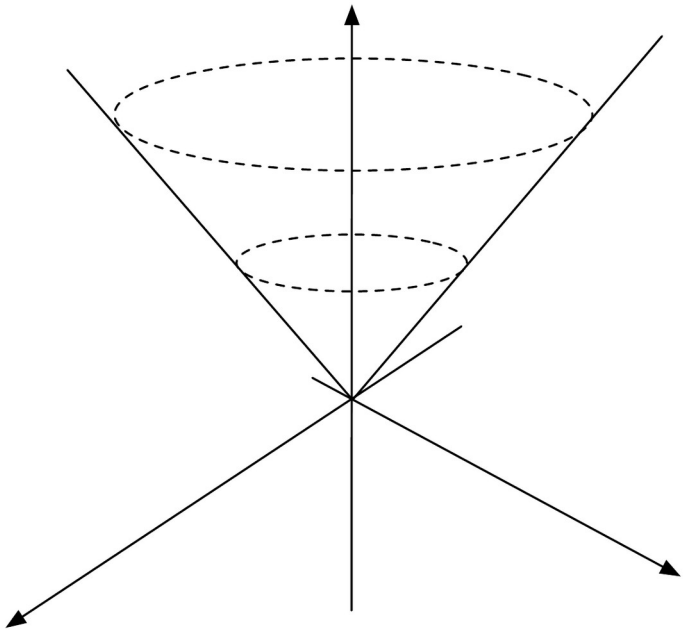
二次錐最適化問題

凸二次の非線形制約を二次錐制約として効率よく扱う

この章の内容

1. Weber問題
2. 二次錐制約と二次錐最適化問題
3. ロバスト最適化
4. ポートフォリオ最適化問題
5. 凹費用関数をもつ単一ソース制約付き施設配置問題

二次錐とは



3次元の二次錐 K_3

- $K_3 = \{(z, x, y) : z \geq \sqrt{(x^2+y^2)}\}$
- アイスクリームのコーンを無限に延ばした形
- 変数の一部が二次錐に入るという制約（二次錐制約）をもつ
- 内点法で効率よく解ける

10.1 Weber問題

砂漠の7組の家のために、どこに井戸を掘れば公平か？

家の位置と人数

距離×水の消費量（人数に比例）の総和が最小になる井戸の位置を求める

家	x座標	y座標	人数
1	24	54	2
2	60	63	1
3	1	84	2
4	23	100	3
5	84	48	4
6	15	64	5
7	52	74	4

最適解は $(32.8, 68.9)$ となった

Weber問題の定式化

家 i の位置 (x_i, y_i)、水の量 w_i、井戸の位置 (X, Y)

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in H} w_i z_i \\ \text{subject to} \quad & \sqrt{(x_i - X)^2 + (y_i - Y)^2} \leq z_i \quad \forall i \in H \end{aligned}$$

- 変数は (X, Y) と z_i
- 最適解では z_i は家 i と井戸の距離に一致する
- 制約は「$(z_i, x_i - X, y_i - Y)$ が K_3 に属する」二次錐制約

AMPLによるモデル

weber.mod

```
set I;                          # 顧客の集合

param x {I};                    # 顧客 i の x 座標
param y {I};                    # 顧客 i の y 座標
param w {I} >= 0;               # 顧客 i の重み

var X;                          # 施設の x 座標
var Y;                          # 施設の y 座標
var z {I} >= 0;                 # 施設から顧客 i までの距離

minimize Cost:
   sum {i in I} w[i] * z[i];

subject to Cone {i in I}:
   (x[i] - X)^2 + (y[i] - Y)^2 <= z[i]^2;
```

- 定式化の制約をそのまま書き下す
- X, Y は自由変数（負にもなりうる）
- z は非負変数
  - $z \geq 0$ で初めて二次錐制約と同値

受け皿の変数は（多くの場合）要らない

- ソルバーは左辺の行列 Q が半正定値と認識できる必要がある
- AMPLでは書き換えの多くを MP 層が引き受ける
  - Gurobi や COPT では定式化そのままで解ける
- Mosek や CPLEX では解けないことがある
  - u[i] = x[i]−X, v[i] = y[i]−Y を用意し u[i]^2 + v[i]^2 <= z[i]^2 と書く

! HiGHS に注意

HiGHS は二次錐制約を扱えず、区分線形近似の近似解を "solved" として返す。二次錐対応ソルバーを指定すること。

10.2

二次錐制約と二次錐最適化問題

どのような非線形制約が二次錐制約に帰着できるか？

n+1次元の二次錐

$$\mathcal{K}_{n+1} = \left\{ (z, x_1, x_2, \ldots, x_n) \ : \ z \geq \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \right\}$$

- 変数（または式）の一部が二次錐に入るという制約が二次錐制約
- 二次錐制約を含む最適化問題が二次錐最適化問題
- 内点法で線形最適化とほとんど同じように効率よく解ける

AMPLで書ける二次錐制約

種類	形	条件
二次錐制約	$\Sigma\, x_i^2 \le y^2$	y は非負変数
凸二次制約	$\Sigma\, Q_{ij}\, x_i\, x_j + \Sigma\, c_i\, x_i \le d$	Q は対称半正定値
回転つき二次錐制約	$\Sigma\, x_i^2 \le y\, z$	y, z は非負変数
分数制約	$1/y \le z$	$y > 0$（$1 \le yz$ と等価）
調和平均型	$xy/(x+y) \ge \beta$	$x, y > 0$

いずれも $\sqrt{\ }$ をとらずにそのまま書き下せる

経済発注量問題の二次錐最適化

目的関数の分数項 F_i / T_i を上から押さえる変数 c_i を導入する

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} c_i + \frac{h_i d_i T_i}{2} \\
\text{subject to} \quad & \sum_{i \in I} w_i d_i T_i \leq W \\
& F_i \leq c_i T_i \quad \forall i \in I \\
& T_i > 0
\end{aligned}
$$

- 最適解では $F_i / T_i = c_i$ が成り立つ
- $F_i / T_i \leq c_i$ の両辺を T_i 倍すると回転つき二次錐制約

経済発注量問題のモデル

eoq_socp.mod

```
set I;

param F {I} > 0;            # 品目 i の発注費用
param h {I} > 0;            # 品目 i の在庫保管費用
param d {I} > 0;            # 品目 i の需要量
param w {I} > 0;            # 品目 i の大きさ
param W > 0;                # 倉庫の容量

var T {I} >= 0.001;         # 品目 i の発注間隔
var c {I} >= 0;             # 品目 i の発注費用の項

minimize Cost:
   sum {i in I} (c[i] + h[i]*d[i]*T[i]/2);

subject to Cone {i in I}:
   F[i] <= c[i] * T[i];

subject to Capacity:
   sum {i in I} w[i]*d[i]*T[i] <= W;
```

- F[i] <= c[i] * T[i] が回転つき二次錐制約
- c と T が非負変数であることが効いている
- 正の下限 0.001 は T > 0 の明示

10.3 ロバスト最適化

原料の成分比率に誤差があっても、最悪の場合に制約を満たすには？

製品混合問題

問題設定

- 4種類の原料を混ぜて製品1トンを作る
- 単価は 5, 6, 8, 20 万円/トン
- 成分1, 2, 3 を 20%, 30%, 20% 以上

成分比率（%）

- 原料1：25, 15, 20
- 原料2：30, 30, 10
- 原料3：15, 65, 5
- 原料4：10, 5, 80

誤差を考慮した制約

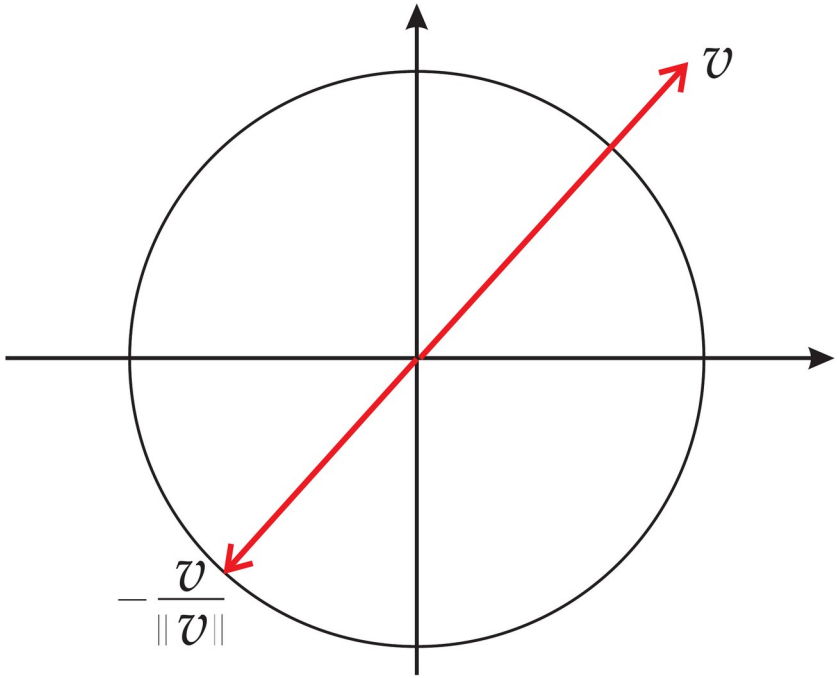
成分比率 a_{ik} は誤差 e_{ik} をもつ。誤差は a_{ik} を中心とし、大きさを ε で抑える

$$\sum_{i=1}^{4} e_{ik}^2 \leq \epsilon^2$$

$$\min \left\{ \sum_{i=1}^{4} (a_{ik} + e_{ik}) x_i \ : \ \sum_{i=1}^{4} e_{ik}^2 \leq \epsilon^2 \right\} \geq LB_k$$

- 最悪の誤差でも制約を満たす＝ロバスト最適化
- 左辺の min は半径 ε の球上での線形関数の最小化

球上での線形関数の最小化



円上での $e_1 v_1 + e_2 v_2$ の最小化

- 円上の最小点は $(e_1, e_2) = -(v_1, v_2) / \|v\|$
- 半径 ε の球では $e_{ik} = -\varepsilon x_i / \|x\|$
- min の値は $-\varepsilon \|x\|$ になる

ロバスト混合問題の定式化

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{4} p_i x_i \\
\text{subject to} \quad & \sum_{i=1}^{4} x_i = 1 \\
& \sqrt{\epsilon^2 \sum_{i=1}^{4} x_i^2} \leq -LB_k + \sum_{i=1}^{4} a_{ik} x_i \quad k = 1, 2, 3 \\
& x_i \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad i = 1, 2, 3, 4
\end{aligned}
$$

- 二次錐最適化問題になる
- $\varepsilon = 0$ とすれば元の線形最適化の混合問題に戻る

ロバスト最適化のモデル

prodmix.mod

```
set I;                      # 原料の集合
set K;                      # 成分の集合
param a {I,K};              # 成分の比率
param p {I} >= 0;           # 原料 i の単価
param epsilon > 0;          # 成分比率の誤差
param LB {K};               # 成分 k の比率の下限

var x {I} >= 0;             # 原料 i の混合比率
var rhs {K} >= 0;           # 二次錐制約の右辺

minimize Cost:
   sum {i in I} p[i] * x[i];
subject to Budget:
   sum {i in I} x[i] = 1;
subject to RhsDef {k in K}:
   rhs[k] = -LB[k] + sum {i in I} a[i,k] * x[i];
subject to Cone {k in K}:
   sum {i in I} epsilon^2 * x[i]^2 <= rhs[k]^2;
```

- 複雑な右辺を変数 rhs に代入して単純化
- rhs を非負変数として宣言するのが重要

ロバスト最適化の効果

比率は%単位、$\varepsilon = 1$（誤差の大きさが 1%）で解いた結果

モデル	最適値	x1	x2	x3	x4
ロバスト最適化	約 7.48	0.53	0.05	0.32	0.10
通常の線形最適化	7.13	0.61	0	0.32	0.08

安全を見込む分、ロバスト最適化の方が費用が大きい

10.4 ポートフォリオ最適化問題

100万円を5つの株に「うまく」分散投資するには？

各株式の1年後の価格

現在の価格はすべて1株1円

株式	1	2	3	4	5
期待値 r_i	1.01	1.05	1.08	1.10	1.20
標準偏差 σ_i	0.07	0.09	0.1	0.2	0.3

各株の価格は互いに独立と仮定する（共分散を考える場合も少しの変更で二次錐最適化に帰着できる）

Markowitzモデル

期待資産価値 α M 以上の条件で、リスク（分散）を最小化する

$$
\begin{aligned}
\text{minimize} \quad & \rho \\
\text{subject to} \quad & \sum_{i=1}^{n} \sigma_i^2 x_i^2 \le \rho \\
& \sum_{i=1}^{n} r_i x_i \ge \alpha \\
& \sum_{i=1}^{n} x_i = 1 \\
& x_i \ge 0 \qquad \forall i
\end{aligned}
$$

- x_i：株 i に投資する割合
- 分散は $M^2 \Sigma \sigma_i^2 x_i^2$（$M$ は省略）
- 補助変数 ρ で目的関数を線形に
- 最初の制約は二次錐制約

Markowitzモデルの記述

markowitz.mod

```
set I;                          # 株式の集合

param sigma {I} >= 0;           # 株式 i の標準偏差
param r {I};                    # 株式 i の期待値
param alpha;                    # 期待資産価値の下限

var x {I} >= 0;                 # 株式 i を購入する割合

minimize Risk:
   sum {i in I} sigma[i]^2 * x[i]^2;

subject to Return:
   sum {i in I} r[i] * x[i] >= alpha;

subject to Budget:
   sum {i in I} x[i] = 1;
```

- $\alpha = 1.05$（5%以上の利益）で解く
- 最適な購入割合（株式1〜5）：0.392, 0.270, 0.239, 0.063, 0.036

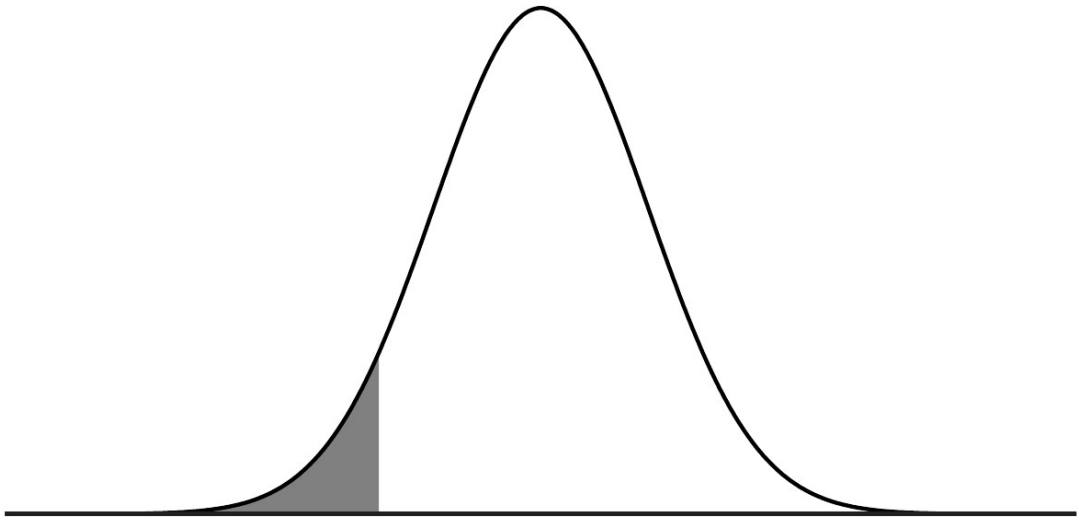
損をする確率を抑えるモデル

資産価値が αM 以下になる確率を β 以下に抑えて、期待資産価値を最大化

$$\mathrm{Prob}\left\{\sum_{i=1}^{n} R_i x_i \le \alpha\right\} \le \beta \;\Leftrightarrow\; -\Phi^{-1}(\beta)\sqrt{\sum_{i=1}^{n} \sigma_i^2 x_i^2} \le -\alpha + \sum_{i=1}^{n} r_i x_i$$

- 分散は得する方向のずれも避けてしまう
- R_i は独立な正規分布 → $\Sigma\, R_i\, x_i$ も正規分布
- $\beta < 1/2$ なら $\Phi^{-1}(\beta) < 0$ で二次錐制約になる

正規分布と $\Phi(\gamma)$



正規分布の確率密度関数

- 灰色部分の面積 $\Phi(\gamma)$：γ 以下となる確率
- Φ とその逆関数 Φ^{-1} は数学ライブラリで計算できる
- $\Phi^{-1}(\beta)$ は Python 側で計算しパラメータで渡す

損をする確率を抑えるモデルの記述

p_portfolio.mod

```
param phi_inv_beta;        # Phi^{-1}(beta) の値

var x {I} >= 0;
var rho;                   # 期待資産価値
var rhoaux >= 0;           # 二次錐制約の右辺

maximize Return:
   rho;

subject to RhoDef:
   rho = sum {i in I} r[i] * x[i];

subject to Budget:
   sum {i in I} x[i] = 1;

subject to AuxDef:
   rhoaux = (alpha - rho) / phi_inv_beta;

subject to Cone:
   sum {i in I} sigma[i]^2 * x[i]^2 <= rhoaux^2;
```

- Python 側で ampl.param["phi_inv_beta"] = phi_inv(beta)
- モデルで表しにくい計算は Python で行う
- 右辺の変数 rhoaux は非負

β を変化させたときの最適解

$\alpha = 95\%$ に固定。最適値は期待利益

β	株式1, 2	株式3	株式4, 5	最適値
0.1	0, 0	0.310	0.235, 0.455	1.14
0.05	0, 0.057	0.492	0.198, 0.253	1.11
0.02	0.052, 0.299	0.404	0.128, 0.117	1.08
0.01	許容解なし	—	—	—

β を小さくすると安全な銘柄にシフトし期待利益が減る。*Markowitz* モデルより大胆にリスクのある銘柄へ投資する

10.5

凹費用関数をもつ単一ソース制約付き施設配置問題

整数変数をもつ二次錐最適化（混合整数二次錐最適化）

単一ソース制約付き施設配置問題

施設 j の輸送量 X_j に凹費用 $f_j\sqrt{X_j}$ がかかる。x_{ij} は顧客 i を施設 j に割り当てるとき 1

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{m} f_j \sqrt{\sum_{i=1}^{n} d_i x_{ij}} + \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} d_i x_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} = 1 && \forall i \\
& \sum_{i=1}^{n} d_i x_{ij} \le M_j && \forall j \\
& x_{ij} \in \{0,1\} && \forall i, j
\end{aligned}
$$

- d_i：需要量、c_{ij}：単位輸送費用、M_j：容量

二次錐制約への変換

補助変数 u_j を導入し、0-1 変数では $x_{ij} = x_{ij}^2$ であることを使う

$$f_j \sqrt{\sum_{i=1}^{n} d_i x_{ij}} \leq u_j \quad \Longrightarrow \quad \sqrt{\sum_{i=1}^{n} f_j^2 d_i x_{ij}^2} \leq u_j \quad \forall j$$

- 目的関数は $\Sigma\, u_j + \Sigma\, c_{ij}\, d_i\, x_{ij}$
- 整数変数を含む二次錐最適化（混合整数二次錐最適化問題）

AMPLによるモデル

flp_socp.mod

```
param d {I} >= 0;           # 顧客 i の需要量
param M {J} >= 0;           # 施設 j の容量
param f {J} >= 0;           # 施設 j の費用係数
param c {I,J} >= 0;         # 単位あたりの輸送費用

var x {I,J} binary;         # 割り当てるとき 1
var X {j in J} >= 0, <= M[j];
var u {J} >= 0;

minimize Cost:
   sum {j in J} u[j]
   + sum {i in I, j in J} c[i,j] * d[i] * x[i,j];
subject to Assign {i in I}:
   sum {j in J} x[i,j] = 1;
subject to Total {j in J}:
   sum {i in I} d[i] * x[i,j] = X[j];
subject to Cone {j in J}:
   sum {i in I} f[j]^2 * d[i] * x[i,j]^2 <= u[j]^2;
```

- 二次錐制約も通常の制約も subject to で書く
- 扱いは AMPL が判断する
- MISOCP 対応ソルバーが必要

まとめ

- ✓ 二次錐制約は凸二次の非線形制約を表し、内点法で効率よく解ける
- ✓ AMPLでは $\sqrt{\ }$ をとらずに `x^2 <= z^2` の形でそのまま書ける
- ✓ 右辺の変数は非負に宣言することが本質的に重要
- ✓ ロバスト最適化・ポートフォリオ最適化が二次錐最適化に帰着する
- ✓ 0-1 変数と組み合わせると混合整数二次錐最適化になる