数理手法 III　講義資料　　　　　　　　　　　　　　　　　　　　　　　　平成 30 (2018) 年度　寒野

凸計画による回帰分析とその CVXPY での実装

- この資料には CVXPY[*1] による実装例が含まれていますが，これは「最適化のプログラムを実際に動かしてみたい」という人向けです．講義の履修や単位の取得に，プログラミングは必要ではありません．
- プログラミングを除く部分は，理解することが期待されます．
- 実装例のファイルは，`https://www.or.mist.i.u-tokyo.ac.jp/kanno/lecture/` からダウンロードできます．

---

1. 最小 2 乗法

ある二つの量 a と b の観測値（データ）として，平面上の点 $(a_1, b_1), \dots, (a_r, b_r)$ が得られているとする．図 1(a) は，$r = 20$ の具体例である．回帰分析とは，変数 b を変数 a の簡単な関数として予測することである．ここで，b は目的変数とよばれ，a は説明変数とよばれる．この例のように説明変数が 1 つの場合は単回帰分析といい，2 つ以上の場合は重回帰分析という．

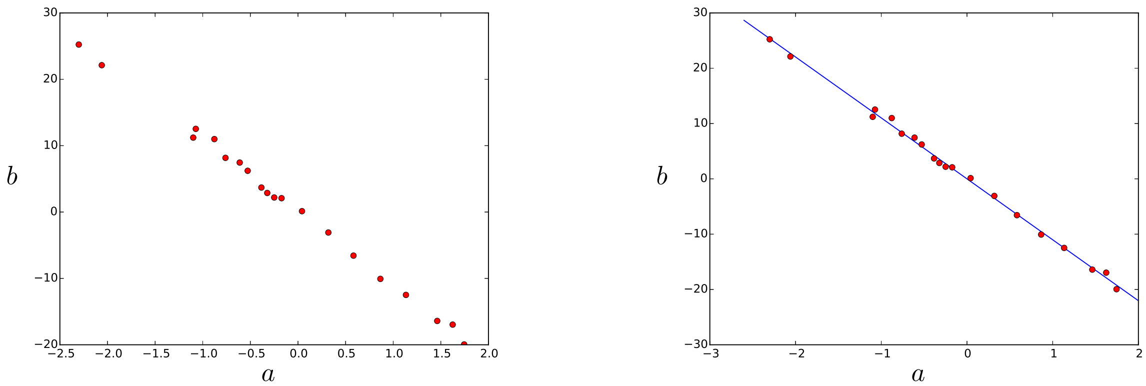
図 1(b) のように，b を a の 1 次関数

$$b = wa + v$$

として近似することを考える．予測値 $wa_l + v$ の観測値 b_l からの誤差の尺度として差の 2 乗和を用いることにすると，誤差を最小化する最適化問題は

$$\text{Minimize} \quad \sum_{l=1}^{r} \bigl[(wa_l + v) - b_l\bigr]^2 \tag{1}$$

と書ける．ただし，$w \in \mathbb{R}$ と $v \in \mathbb{R}$ が最適化の決定変数である．このように，予測値と観測値の差の 2 乗和が最小になる予測モデルを求める方法のことを，最小 2 乗法という．



(a)　　　　　　　　　　　　　　　　(b)

図 1: 最小 2 乗法による線形回帰の例．(a) 与えられたデータと (b) 得られる回帰直線

[*1] `http://www.cvxpy.org/`

問題 (1) の目的関数を整理すると [*2]

$$\left\| \begin{bmatrix} (a_1 w + v) - b_1 \\ \vdots \\ (a_r w + v) - b_r \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_r & 1 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} - \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix} \right\|_2^2$$

と書ける．表記の簡単のために

$$G = \begin{bmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_r & 1 \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} w \\ v \end{bmatrix}$$

とおくと，問題 (1) は

$$\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_2^2 \tag{2}$$

と整理できる．この目的関数をさらに展開すると

$$\boldsymbol{x}^\top (G^\top G)\boldsymbol{x} - 2\boldsymbol{h}^\top G\boldsymbol{x} + \|\boldsymbol{h}\|_2^2$$

が得られるが，行列 $G^\top G$ は対称かつ半正定値 [*3] である．したがって，問題 (2) は凸 2 次関数の最小化問題であり，2 次計画の（制約をもたないという）特別な場合と言える．

図 1 の例に対する問題 (2) は，CVXPY を用いると次のようにして解くことができる．

```
#ファイル名：'least_square_ex.py'
import cvxpy as cp                  #まずはモジュールの読み込み
import numpy as np
import matplotlib.pyplot as plt     #これはグラフを描くためのモジュール
#
r = 20                              #データ点の数
np.random.seed(1)                   #乱数の種を指定（これは単に再現性のため）
G = np.hstack(( np.random.randn(r,1), np.ones([r,1]) ))
c = np.array( [G[:,0]] ).T
h = (10.0*np.random.randn() * c) + (0.8*np.random.randn(r,1))
#
x   = cp.Variable(2,1)              #最適化の決定変数の定義
obj = cp.Minimize( sum(cp.square(G*x-h)) )   #目的関数の定義
P = cp.Problem(obj)                 #最適化問題の定義
P.solve(verbose=True)               #求解
print(x.value)                      #最適解の出力
#
s = np.arange(c.min()-0.3, c.max()+0.3, 0.1) #ここからはグラフの描画
t = np.asscalar(x.value[0]) * s + np.asscalar(x.value[1])
plt.plot(s, t)
plt.plot(c, h, 'ro')
plt.show()
```

[*2] ベクトル $\boldsymbol{s} \in \mathbb{R}^n$ のユークリッド (Euclid) ノルムを $\|\boldsymbol{s}\|_2 = (\boldsymbol{s}^\top \boldsymbol{s})^{1/2}$ で表す．講義では，記号 $\|\boldsymbol{s}\|$ を主に用いている．
[*3] というのも，任意の $\boldsymbol{x} \in \mathbb{R}^2$ に対して，$\boldsymbol{x}^\top (G^\top G)\boldsymbol{x} = (G\boldsymbol{x})^\top (G\boldsymbol{x}) = \|G\boldsymbol{x}\|_2^2 \geq 0$ が成り立つ．

重回帰分析の場合，説明変数の数を d とすると，データ点は $\boldsymbol{a}_l \in \mathbb{R}^d$ と $b_l \in \mathbb{R}$ の組として与えられる．1 次関数での予測モデルは

$$b = \boldsymbol{w}^\top \boldsymbol{a} + v$$

と書けて，この $\boldsymbol{w} \in \mathbb{R}^d$ と $v \in \mathbb{R}$ を決めることが目標である．ここで，G, $\boldsymbol{h}$, $\boldsymbol{x}$ を

$$G = \begin{bmatrix} \boldsymbol{a}_1^\top & 1 \\ \vdots & \vdots \\ \boldsymbol{a}_r^\top & 1 \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{w} \\ v \end{bmatrix}$$

で定めると，最小 2 乗法はやはり問題 (2) の形で書ける．ただし，G は $r \times (d+1)$ 型行列，$\boldsymbol{h}$ は r 次元の列ベクトル，$\boldsymbol{x}$ は $d+1$ 次元の列ベクトルである．

予測モデルとして非線形の関数を用いる場合でも，最小 2 乗法は問題 (2) の形で書ける．例として，3 次関数の予測モデルを用いた単回帰分析を考える．つまり，予測モデルを

$$b = w_1 a^3 + w_2 a^2 + w_3 a + v$$

として，$w_1, w_2, w_3, v \in \mathbb{R}$ を決めることが目標である．この場合は，G, $\boldsymbol{h}$, $\boldsymbol{x}$ を

$$G = \begin{bmatrix} {a_1}^3 & {a_1}^2 & a_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ {a_r}^3 & {a_r}^2 & a_r & 1 \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} b_1 \\ \vdots \\ b_r \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ v \end{bmatrix}$$

と定めれば，最小 2 乗法は問題 (2) の形で書ける．

非線形の予測モデルを用いた重回帰分析の場合でも，最小 2 乗法はやはり問題 (2) の形で書ける．

2. 正則化

最小 2 乗法 (2) を解いて，予測モデルのパラメータ $\boldsymbol{x}$ が決まったとする [*4]．行列 $G \in \mathbb{R}^{m \times n}$ は観測されたデータであるから，通常は何らかの誤差を含んでいる．ここで，G の真の値がある行列 $\Delta \in \mathbb{R}^{m \times n}$ を用いて $G + \Delta$ と表せるものとする．すると，真のデータと予測値との差は，三角不等式より

$$\|(G + \Delta)\boldsymbol{x} - \boldsymbol{G}\|_2 \leq \|G\boldsymbol{x} - \boldsymbol{h}\|_2 + \|\Delta \boldsymbol{x}\|_2$$

と評価できる．ここで，最小 2 乗法は右辺第 1 項の最小化と等価であるが，右辺第 2 項の値が大きければ真のデータと予測値の差は大きくなり得ることがわかる．このことから，$\|\boldsymbol{x}\|_2$ が大きいモデルは，データに含まれる誤差の影響を大きく受けると言える．そこで，$\|G\boldsymbol{x} - \boldsymbol{h}\|_2$ と $\|\boldsymbol{x}\|_2$ の両方を小さくすることが望ましい．それぞれを 2 乗したものを（適当なバランスをとりながら）小さくする問題は，パラメータ $\gamma > 0$ を用いて

$$\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_2^2 + \gamma \|\boldsymbol{x}\|_2^2 \tag{3}$$

[*4] 以降では，一般論として，G は $m \times n$ 型行列 $(m > n)$ で階数（ランク）が n であるとする．

と定式化できる．この問題 (3) を解いて予測モデルのパラメータ $\boldsymbol{x}$ を決める手法を，ティコノフ (Tikhonov) 正則化付き最小 2 乗法（またはリッジ回帰）とよぶ．この問題も，凸 2 次関数の最小化問題である．

問題 (3) のように，目的関数に追加の項を導入して，誤差に対する予測モデルのロバスト性を高めたり予測モデルが複雑になり過ぎることを防ぐことは，正則化とよばれている．ティコノフ正則化以外にも，さまざまな正則化が知られていて，目的に応じて使いわけられている．たとえば，ℓ_1 ノルム正則化付き最小 2 乗法[*5]

$$\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_2^2 + \gamma\|\boldsymbol{x}\|_1 \tag{4}$$

は，$\boldsymbol{x}$ として 0 の成分を多く含むベクトル（そのようなベクトルを，疎なベクトルとよぶ）を求めたい場合に用いられる．問題 (4) を用いて予測モデルのパラメータを決める手法は，しばしば LASSO[*6] とよばれる．問題 (4) は，凸 2 次計画問題に帰着できる．つまり，補助的な変数 $\boldsymbol{z} \in \mathbb{R}^n$ を導入すると，問題 (4) は次のように変形できる：

$$\begin{aligned} \text{Minimize} \quad & \|G\boldsymbol{x} - \boldsymbol{h}\|_2^2 + \gamma\mathbf{1}^\top\boldsymbol{z} && \text{(5a)} \\ \text{subject to} \quad & \boldsymbol{z} \ge \boldsymbol{x}, && \text{(5b)} \\ & \boldsymbol{z} \ge -\boldsymbol{x}. && \text{(5c)} \end{aligned}$$

ただし，$\mathbf{1} = (1, 1, \dots, 1)^\top \in \mathbb{R}^n$ である．

CVXPY は，問題 (4) の形を入力すれば，自動的に問題 (5) の形に変換した上で最適解を出力する[*7]．

```python
# ファイル名：'lasso_ex.py'
import cvxpy as cp                  #まずはモジュールの読み込み
import numpy as np
gam = 100.0                         #gammaの値を入力
m, n = 10, 5                        #ここから問題のデータの入力
np.random.seed(2)
G, h = np.random.randn(m,n), 100 * np.random.randn(m,1)
x = cp.Variable(n,1)                #最適化の決定変数の定義
obj = cp.Minimize( sum(cp.square(G*x-h)) + (gam*cp.norm(x,1)) )  #目的関数
P = cp.Problem(obj)                 #最適化問題の定義
P.solve(verbose=True)               #求解
print(x.value)                      #最適解の出力
```

以上では，予測値の観測値からのずれの尺度として，それらの差の 2 乗和を用いている．その他の尺度も，目的に応じて使われる．たとえば，ℓ_∞ ノルム[*8] を用いた問題

$$\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_\infty \tag{6}$$

はチェビシェフ (Chebyshev) 近似問題とよばれている．この問題は，補助的な変数 $s \in \mathbb{R}$ を導入する

[*5] ベクトル $\boldsymbol{s} \in \mathbb{R}^n$ の ℓ_1 ノルムは $\|\boldsymbol{s}\|_1 = |s_1| + |s_2| + \cdots + |s_n|$ で定義される．
[*6] LASSO は，least absolute shrinkage and selection operator の略である．
[*7] もちろん，問題 (5) の形を入力してもよい．
[*8] ベクトル $\boldsymbol{s} \in \mathbb{R}^n$ の ℓ_∞ ノルムは $\|\boldsymbol{s}\|_\infty = \max\{|s_1|, |s_2|, \dots, |s_n|\}$ で定義される．

ことで次の線形計画問題に帰着できる：

$$
\begin{aligned}
\text{Minimize} \quad & s && \text{(7a)}\\
\text{subject to} \quad & s\mathbf{1} \geq G\boldsymbol{x} - \boldsymbol{h}, && \text{(7b)}\\
& s\mathbf{1} \geq -G\boldsymbol{x} + \boldsymbol{h}. && \text{(7c)}
\end{aligned}
$$

CVXPY では，問題 (6) の形を直接，入力することができる（この場合には，線形計画問題への帰着は CVXPY が行う）．CVXPY で問題 (6) を解くには，'lasso_ex.py' の「目的関数」の行を次のように変更すればよい．

```
obj = cp.Minimize( cp.norm(G*x-h,"inf") )
```

また，問題 (4) に比べてデータに含まれる外れ値（観測や記録の失敗などにより異常に大きな誤差をもつデータ点）の影響を受けにくい問題として，絶対値の総和を用いた問題

$$
\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_1 + \gamma\|\boldsymbol{x}\|_1 \tag{8}
$$

が考えられる．というのも，z^2 と $|z|$ を比べると，$|z|$ が大きい値をとるにつれて z^2 はより速く増加する．このため，誤差の 2 乗和を最小化すると外れ値での誤差を小さくする予測モデルが選択される傾向があるが，絶対値の和の最小化ではこの傾向が緩和されるからである．問題 (8) も，線形計画問題に帰着できる（問題 (4) から問題 (5) への変形と同様に考えればよい）．

CVXPY では，問題 (8) の形を入力することができる．実際，'lasso_ex.py' の「目的関数」の行を次のように変更すればよい．

```
obj = cp.Minimize( cp.norm(G*x-h,1) + (gam*cp.norm(x,1)) )
```

3. 演習問題

問 1　問題 (8) を，線形計画問題に変形せよ．さらに，それを線形計画問題の等式標準形に直せ．

問 2　問題 (3) の最適解を，G, $\boldsymbol{h}$, γ および単位行列 I を用いて表せ．

問 3　γ および ρ を正の定数とするとき，最適化問題

$$
\text{Minimize} \quad \|G\boldsymbol{x} - \boldsymbol{h}\|_2^2 + \gamma\|\boldsymbol{x}\|_2^2 + \rho\|\boldsymbol{x}\|_1
$$

はエラスティックネット正則化付き最小 2 乗法とよばれる問題である．この問題を，2 次計画問題

$$
\begin{aligned}
\text{Minimize} \quad & \frac{1}{2}\boldsymbol{y}^\top Q\boldsymbol{y} + \boldsymbol{c}^\top\boldsymbol{y}\\
\text{subject to} \quad & A\boldsymbol{y} = \boldsymbol{b},\\
& \boldsymbol{y} \geq \mathbf{0}
\end{aligned}
$$

の形に直せ．ただし，Q は半正定値対称行列とする．

（以上）