統計解析Ⅱ 1〜20

①

$\mathbb{X} \in \mathbb{R}^{N \times P}$, $\mathbb{y} \in \mathbb{R}^{N}$, $\mathbb{\beta} \in \mathbb{R}^{P}$

中心化されている.

$$\frac{\partial}{\partial \beta_j}\left\{\sum_{i=1}^{N}\left(y_i - \sum_{k=1}^{P}\beta_k x_{ik}\right)^2\right\}$$ $(j=1,2,\cdots,N)$ について

$$= -2\sum_{i=1}^{N}\left\{\left(y_i - \sum_{k=1}^{P}\beta_k x_{ik}\right)x_{ij}\right\}$$

$$= -2\begin{pmatrix} x_{1j} & x_{2j} & \cdots & x_{Nj}\end{pmatrix}\begin{pmatrix} y_1 - \sum_{k=1}^{P}\beta_k x_{1k} \\ y_2 - \sum_{k=1}^{P}\beta_k x_{2k} \\ \vdots \\ y_N - \sum_{k=1}^{P}\beta_k x_{Nk}\end{pmatrix}$$

よって

$$\begin{pmatrix} \frac{\partial}{\partial \beta_1}\sum_{i=1}^{N}\left(y_i - \sum_{k=1}^{P}\beta_k x_{ik}\right)^2 \\ \vdots \\ \frac{\partial}{\partial \beta_P}\sum_{i=1}^{N}\left(y_i - \sum_{k=1}^{P}\beta_k x_{ik}\right)^2 \end{pmatrix} = -2\begin{pmatrix} x_{11} & \cdots & x_{N1} \\ \vdots & & \vdots \\ x_{1P} & \cdots & x_{NP}\end{pmatrix}\begin{pmatrix} y_1 - \sum_{k=1}^{P}\beta_k x_{1k} \\ \vdots \\ y_N - \sum_{k=1}^{P}\beta_k x_{Nk}\end{pmatrix}$$

▨

$$\begin{pmatrix} \sum_{k=1}^{p} \beta_k x_{1k} \\ \vdots \\ \sum_{k=1}^{p} \beta_k x_{Nk} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \\ x_{N1} & \cdots & x_{Np} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} = \mathbb{X}\mathbb{B}$$ と表せるので

$$-2 \begin{pmatrix} x_{11} & \cdots & x_{N1} \\ \vdots & & \\ x_{1p} & \cdots & x_{Np} \end{pmatrix} \begin{pmatrix} y_1 - \sum_{k=1}^{p} \beta_k x_{1k} \\ \\ y_N - \sum_{k=1}^{p} \beta_k x_{Nk} \end{pmatrix}$$

$$= -2 \mathbb{X}^T (\mathbb{y} - \mathbb{X}\mathbb{B})$$

目的関数 $L(\mathbb{B}) = \|\mathbb{y} - \mathbb{X}\mathbb{B}\|_2^2$ とした時に $L(\mathbb{B})$ は凸関数であるので微分して $\mathbb{0}$ となる $\mathbb{B}$ が最適解と言える。

$$\frac{\partial L}{\partial \mathbb{B}} = -2\mathbb{X}^T(\mathbb{y} - \mathbb{X}\mathbb{B}) = \mathbb{0}$$ の時

$$\mathbb{X}^T\mathbb{X}\mathbb{B} = \mathbb{X}^T\mathbb{y}$$

$$\hat{\mathbb{B}} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{y}$$

②
(a) $f(x) = x$

$$f(\alpha x + (1-\alpha)y) = \alpha x + (1-\alpha)y$$

$$\leq \alpha f(x) + (1-\alpha) f(y)$$

よって $f(x)$ は 凸

劣微分 $z = \partial f(0)$ について
z は任意の $x \in \mathbb{R}$ について上以下をみたすような集合.

$$x \geq 0 + z(x-0)$$

$$x \geq zx \longrightarrow \begin{cases} 1 \geq z & x > 0 \\ 1 \leq z & x < 0 \\ \text{不定} & x = 0 \end{cases}$$

従って $z = \partial f(0) = 1$ 四

(b) $f(x) = |x|$

$$f(\alpha x + (1-\alpha)y) = |\alpha x + (1-\alpha)y|$$

三角不等式より

$$\leq |\alpha x| + |(1-\alpha)y|$$

$\alpha \in (0,1)$ より

$$= \alpha|x| + (1-\alpha)|y|$$

$$= \alpha f(x) + (1-\alpha) f(y)$$ よって 凸

$z = \partial f(0)$ について

$$|x| \geq zx \longrightarrow \begin{cases} 1 \geq z & x > 0 \\ -1 \leq z & x < 0 \\ \text{不定} & \end{cases}$$

従って $-1 \leq z \leq 1$

よって $\partial f(0) = [-1, 1]$ 四

③

(a) $g(x), h(x)$ が凸の時 $\beta, \gamma \geq 0$ を用いて

$f(x) = \beta g(x) + \gamma h(x)$ とする. $\alpha \in (0,1)$ について

$f(\alpha x + (1-\alpha)y) = \beta g(\alpha x + (1-\alpha)y) + \gamma h(\alpha x + (1-\alpha)y)$

$g(x), h(x)$ は凸関数, 定義より

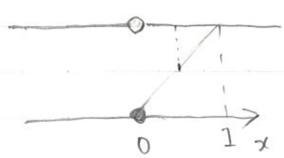
$\leq \beta\{\alpha g(x) + (1-\alpha)g(y)\} + \gamma\{\alpha h(x) + (1-\alpha)h(y)\}$

$= \alpha\{\beta g(x) + \gamma h(x)\} + (1-\alpha)\{\beta g(y) + \gamma h(y)\}$

$= \alpha f(x) + (1-\alpha) f(y)$

従って $\beta g(x) + \gamma h(x)$ は凸

(b) $f(x) = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases}$ が凸でないことを示す.



$\begin{cases} x = 0 \\ y = 1 \\ \alpha = \frac{1}{2} \end{cases}$ とした時に

$f(\alpha x + (1-\alpha)y) = f(\frac{1}{2}) = 1$

一方,

$\alpha f(x) + (1-\alpha) f(y) = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 = \frac{1}{2}$

これは $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ を

みたさないので凸でない

$\| \frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix} \|_0 = 1$ $\begin{pmatrix}1\\0\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix}$

$\frac{1}{2}\|\begin{pmatrix}1\\0\end{pmatrix}\|_0 + \frac{1}{2}\|\begin{pmatrix}0\\1\end{pmatrix}\|_0 = 1$

(C) (i) $f(\mathbb{B}) = \frac{1}{2N}\|y - X\mathbb{B}\|^2 + \lambda\|\mathbb{B}\|_0$

$g(\mathbb{B}) = \|\mathbb{B}\|_2^2$ とすると

$\forall \mathbb{B}_1, \mathbb{B}_2 \in \mathbb{R}^p, \forall \alpha \in (0,1)$ について

$$g(\alpha\mathbb{B}_1 + (1-\alpha)\mathbb{B}_2) - \left(\alpha g(\mathbb{B}_1) + (1-\alpha)g(\mathbb{B}_2)\right)$$

$$= \alpha(1-\alpha)\left\{\|\mathbb{B}_1\|^2 - 2\|\mathbb{B}_1\|\|\mathbb{B}_2\| + \|\mathbb{B}_2\|^2\right\}$$

$$= \alpha(1-\alpha)\|\mathbb{B}_1 - \mathbb{B}_2\|^2 \geq 0$$

よって 2ノルムは凸関数ということが言える。

$\mathbb{B}_1 = \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} \in \mathbb{R}^p$ $\quad f(\alpha\mathbb{B}_1 + (1-\alpha)\mathbb{B}_2)$

$= f(\frac{1}{2}\mathbb{1})$

$\mathbb{B}_2 = \begin{pmatrix}0\\1\\\vdots\\1\end{pmatrix} \in \mathbb{R}^p$ $\quad = \frac{1}{2N}\|y - \frac{1}{2}X\mathbb{1}\|^2 + \lambda p$

$\alpha = \frac{1}{2}$ とする.

また $\mathbb{1} = \begin{pmatrix}1\\\vdots\\1\end{pmatrix} \in \mathbb{R}^p$ と表記する

$$\mathbb{I}^T \mathbb{X}^T \mathbb{X} \mathbb{I} = (\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2)^T \underbrace{\mathbb{X}^T \mathbb{X}}_{\text{半正定値}} (\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2) \geq \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2$$

$$\alpha f(\boldsymbol{\beta}_1) + (1-\alpha) f(\boldsymbol{\beta}_2) = \frac{\alpha}{2N} \| \boldsymbol{y} - \mathbb{X}\boldsymbol{\beta}_1 \|^2 + \frac{1-\alpha}{2N} \| \boldsymbol{y} - \mathbb{X}\boldsymbol{\beta}_2 \|^2 + \frac{\lambda P}{2}$$

$$= \frac{1}{2} \cdot \frac{1}{2N} \left(\boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 \right.$$

$$\left. + \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T \mathbb{X} \boldsymbol{\beta}_2 + \boldsymbol{\beta}_2^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) + \frac{\lambda P}{2}$$

$$= \frac{1}{2} \cdot \frac{1}{2N} \left(2\boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T \mathbb{X} (\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2) + \boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) + \frac{\lambda P}{2}$$

$$= \frac{1}{2N} \left\{ \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \mathbb{X} (\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2) + \frac{1}{2} \left(\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) \right\} + \frac{\lambda P}{2}$$

$$= \frac{1}{2N} \left\{ \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \mathbb{X} \cdot \mathbb{I} + \frac{1}{4} \left(\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) \right\} + \lambda P$$

$$+ \frac{1}{8N} \left(\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) - \frac{\lambda P}{2}$$

$$\frac{1}{8N} \left(\boldsymbol{\beta}_1^T \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_1 + \boldsymbol{\beta}_2 \mathbb{X}^T \mathbb{X} \boldsymbol{\beta}_2 \right) - \frac{\lambda P}{2} \leq 0$$

をみたすように
λを選択すれば

$$\leq \frac{1}{2N} \left\| \boldsymbol{y} - \frac{1}{2} \mathbb{X} \mathbb{I} \right\|_2^2 + \lambda P$$

$$= f(\alpha \boldsymbol{\beta}_1 + (1-\alpha) \boldsymbol{\beta}_2)$$

となるので凸関数でなくなる。

従って(i)は必ず凸関数になるとは
限らない。

(ii) $f(\mathbb{B}) = \frac{1}{2N}\|y - \mathbb{X}\mathbb{B}\|^2 + \lambda\|\mathbb{B}\|_1$

ここで $g(\mathbb{B}) = \|\mathbb{B}\|_1$ とすれば

$\forall x, y \in \mathbb{R}^p$, $\forall \alpha \in (0,1)$ について

$$
\begin{aligned}
g(\alpha x + (1-\alpha)y) &= \|\alpha x + (1-\alpha)y\|_1 \\
&\leq \|\alpha x\|_1 + \|(1-\alpha)y\|_1 \\
&\qquad \because \text{ノルムの劣加法性より} \\
&= \alpha\|x\|_1 + (1-\alpha)\|y\|_1 \\
&= \alpha g(x) + (1-\alpha) g(y)
\end{aligned}
$$

より $g(\mathbb{B})$ は凸関数，前ページより第1項も凸関数より
$f(\mathbb{B})$ は凸関数

(iii) $f(\mathbb{B}) = \frac{1}{N}\|y - \mathbb{X}\mathbb{B}\|^2 + \lambda\|\mathbb{B}\|_2^2$

前ページより2ノルムは凸関数となるので

第1項，第2項共に凸関数となり，

凸関数の和も凸関数より

$f(\mathbb{B})$ は凸関数　$\blacksquare$

④

(a) $f: \mathbb{R} \to \mathbb{R}$ が凸で $x = x_0$ で微分可能

凸関数より $\alpha \in (0,1)$ を用いて

$$f(\alpha x + (1-\alpha)x_0) \leq \alpha f(x) + (1-\alpha) f(x_0)$$

$$f(\alpha(x - x_0) + x_0) \leq \alpha\{f(x) - f(x_0)\} + f(x_0)$$

$$f(x_0 + \alpha(x - x_0)) - f(x_0) \leq \alpha\{f(x) - f(x_0)\}$$

$\alpha > 0$ より

$$f(x) \geq f(x_0) + \frac{f(x_0 + \alpha(x - x_0))}{\alpha(x - x_0)}(x - x_0)$$

$\alpha \to 0$ とすると $\alpha(x - x_0) \to 0$ となるので

$$f(x) \geq f(x_0) + f'(x_0)(x - x_0)$$

$\boxed{\text{終}}$

(2) 劣微分 $Z = \partial f(x_0)$ とすれば

$\forall x \in \mathbb{R}$

$$f(x) \geq f(x_0) + Z(x - x_0)$$

$$(*) \cdots \begin{cases} \dfrac{f(x) - f(x_0)}{x - x_0} \geq Z & (x > x_0) \\ \dfrac{f(x) - f(x_0)}{x - x_0} \leq Z & (x < x_0) \end{cases}$$

任意の $x \in \mathbb{R}$ について $(*)$ が成り立つので

$$\lim_{x \to x_0 - 0} \frac{f(x) - f(x_0)}{x - x_0} \leq Z \leq \lim_{x \to x_0 + 0} \frac{f(x) - f(x_0)}{x - x_0}$$

ここで f は x_0 で微分可能より

$\displaystyle \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$ が存在して

$$\lim_{x \to x_0 + 0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{x \to x_0 - 0} \frac{f(x) - f(x_0)}{x - x_0} = f'(x_0)$$ となるので

$$Z = \{ f'(x_0) \}$$ となる

□

⑤

(a) $f(x) = x^2 - 3x + |x|$

$$= \begin{cases} x^2 - 3x + x & (x \geq 0) \\ x^2 - 3x - x & (x < 0) \end{cases}$$

$$f'(x) = \begin{cases} 2x - 2 & (x > 0) \\ 2x - 3 + [-1, 1] = [-4, -2] \not\ni 0 & (x = 0) \\ 2x - 4 & (x < 0) \end{cases}$$

よって　$x = 1$ で極小

(b) $f(x) = x^2 + x + 2|x|$

$$= \begin{cases} x^2 + x + 2x & (x \geq 0) \\ x^2 + x - 2x & (x < 0) \end{cases}$$

$$f'(x) = \begin{cases} 2x + 3 & (x > 0) \\ 1 + 2[-1, 1] = [-1, 2] \ni 0 & (x = 0) \\ 2x - 1 & (x < 0) \end{cases}$$

よって　$x = 0$ で極小

$\boxed{x_i}$

⑥

$\mathbb{X} \in \mathbb{R}^{N \times p}$ $\quad y \in \mathbb{R}^N$ $\quad \lambda \geq 0$ とする.

ここで $\frac{1}{N} \mathbb{X} \mathbb{X}^T = I \in \mathbb{R}^{N \times N}$

$L = \frac{1}{2N} \| y - \mathbb{X} \beta \|_2^2 + \lambda \| \beta \|_1$ の劣微分.

$S_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij} y_i$ とおく.

$$\frac{\partial L}{\partial \beta_j} = -\frac{1}{N} \left\{ \sum_{i=1}^{N} x_{ij} \left(y_i - \sum_{k=1}^{p} x_{ik} \beta_k \right) \right\} + \lambda \begin{cases} 1 & (\beta_j > 0) \\ [-1, 1] & (\beta_j = 0) \\ -1 & (\beta_j < 0) \end{cases}$$

$$\frac{\partial L}{\partial \beta_j} = -S_j + \beta_j + \begin{cases} \lambda & (\beta_j > 0) \\ [-\lambda, \lambda] & (\beta_j = 0) \\ -\lambda & (\beta_j < 0) \end{cases} \quad //$$

劣微分が 0 の時

$$0 \in \begin{cases} -S_j + \beta_j + \lambda & (\beta_j > 0) \\ -S_j + \beta_j + [-\lambda, \lambda] & (\beta_j = 0) \\ -S_j + \beta_j - \lambda & (\beta_j < 0) \end{cases}$$

β_j について書き直す.

$$\beta_j = \begin{cases} S_j - \lambda & (S_j > \lambda) \\ 0 & (-\lambda \leq S_j \leq \lambda) \\ S_j + \lambda & (S_j < -\lambda) \end{cases}$$

ソフトしきい値関数

$$S_\lambda(x) = \begin{cases} x - \lambda & x > \lambda \\ 0 & -\lambda \leq x \leq \lambda \\ x + \lambda & x < -\lambda \end{cases}$$ を用いて

$\frac{\partial L}{\partial \beta_j} = 0$ をみたす β_j は

$$\beta_j = S_\lambda(S_j)$$ となる.

⑦

$$S_\lambda(x) = \begin{cases} x-\lambda & x>\lambda \\ 0 & |x|\leq\lambda \\ x+\lambda & x<-\lambda \end{cases}$$

を sign 関数，max 関数用いて表す.

$f(x)=\mathrm{sign}(x)\,(|x|-\lambda)_+$ について

(i) $x>0$ の時

$$f(x)=\max(x-\lambda,0) \iff \begin{cases} x-\lambda & (x>\lambda) \\ 0 & (0<x\leq\lambda) \end{cases}$$

(ii) $x=0$ の時

$$f(x)=0$$

(iii) $x<0$ の時

$$f(x)=-\max(-x-\lambda,0) \iff \begin{cases} x+\lambda & (x<-\lambda) \\ 0 & (-\lambda\leq x<0) \end{cases}$$

以上より

$$f(x)=\begin{cases} x-\lambda & (x>\lambda) \\ 0 & (-\lambda\leq x\leq\lambda) \\ x+\lambda & (x<-\lambda) \end{cases}$$

となるので $f(x)=S_\lambda(x)$ となる　□

(10)

λが十分大きいと

$$L(\beta) = \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

の最適解は $\beta = 0$ となる.

デザイン行列の直交性が仮定できない時.

$$0 \in \frac{\partial L}{\partial \beta_j} = -\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \sum_{k=1}^{p} x_{ik}\beta_k\right) + \lambda \begin{cases} 1 & (\beta_j > 0) \\ [-1, 1] & (\beta_j = 0) \\ -1 & (\beta_j < 0) \end{cases}$$

$$\underbrace{y_i - \sum_{k \neq j}^{p} x_{ik}\beta_k}_{=\, r_{ij}\text{とする.}} - x_{ij}\beta_j$$

$$\frac{\partial L}{\partial \beta_j} = -\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(r_{ij} - x_{ij}\beta_j\right) + \lambda \begin{cases} 1 \\ [-1, 1] \\ -1 \end{cases}$$

β_k $(k \neq j)$を固定して → r_{ij}を計算できる.
β_jを更新する

→ 座標降下法

$$\beta_j = \frac{S_\lambda\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij} r_{ij}\right)}{\frac{1}{N}\sum_{i=1}^{N}(x_{ij})^2}$$

$j = 1, 2, \cdots, p$に対して
更新を行っていく

今回簡単のため

$$\sum_{i=1}^{N} (x_{ij})^2 = 1$$

十分大きな λ に対しては，$\beta = 0$ が最適解であり，

$$-\lambda \leq -\frac{1}{N} \sum_{i=1}^{N} x_{ij} (r_{ij} - x_{ij} \beta_j) \leq \lambda$$

が成立している.

よって $r_{ij} = y_i - \sum_{k \neq j}^{p} x_{ik} \beta_k = y_i \quad (i = 1, 2, \cdots, N)$

λを徐々に小さくしていくと

$$\frac{1}{N} \left| \sum_{i=1}^{N} x_{ij} y_i \right| \leq \lambda \quad (j = 1, \cdots, p)$$

について等式が成り立つような j が存在することになる.

この時の j について

$$\frac{1}{N} \left| \sum_{i=1}^{N} x_{ij} y_i \right| = \lambda$$

このような j は

$j = \underset{j=1,2,\cdots,p}{\operatorname{argmin}} \frac{1}{N} \left| \sum_{i=1}^{N} x_{ij} y_i \right|$ で与えられるので

λの最小値は

$$\lambda = \max_{1 \leq j \leq p} \left| \frac{1}{N} \sum_{i=1}^{N} x_{ij} y_i \right| \quad \text{■}$$

(11)

$\mathbb{X}^T\mathbb{X}$ は $\forall x \in \mathbb{R}^p$ を用いて

$$x^T \mathbb{X}^T \mathbb{X} x = \|\mathbb{X} x\|_2^2 \geq 0$$ より

$\mathbb{X}^T\mathbb{X}$ は半正定値行列なので

$\gamma_1, \cdots, \gamma_p \geq 0$ が成り立つ.

$\gamma_1 = 0$ とすれば γ_1 は固有値より固有方程式の解の1つであるので

$$\phi_{\mathbb{X}^T\mathbb{X}} = \det(0I - \mathbb{X}^T\mathbb{X}) = 0$$

$$\det(\mathbb{X}^T\mathbb{X}) = 0$$

よってこの時 $\mathbb{X}^T\mathbb{X}$ に逆行列が存在しない

従って求める条件は

<u>γ_i $(i=1,\cdots,p)$ の少なくとも1つが 0 の時</u> //

$X^T X$ の固有値は固有方程式

$\phi(\gamma) = \det(\gamma I - X^T X) = 0$ の解であり，それを $\gamma_1, \gamma_2, \cdots, \gamma_p$ と表す.

ここで $X^T X + N\lambda I$ についての固有方程式は，

$$\phi(\tau) = \det(\tau I - X^T X - N\lambda I)$$
$$= \det((\tau - N\lambda) I - X^T X)$$

よって

$X^T X + N\lambda I$ の固有値 $\tau_1, \tau_2, \cdots, \tau_p$ について

$$\tau_i - N\lambda = \gamma_i \quad (i = 1, 2, \cdots, p)$$

$$\tau_i = \gamma_i + N\lambda \quad (i = 1, 2, \cdots, p)$$ となる //

以上より $\lambda > 0$ と $\gamma_i \geq 0$ $(i = 1, 2 \cdots p)$ となるので

$$\tau_i > 0 \quad (i = 1, 2, \cdots, p)$$

となり全ての固有値が正なので $X^T X + N\lambda I$ には必ず逆行列が存在する.

(12)

$\lambda \geq 0$ として

$$\underbrace{\mathop{\arg\min}_{\mathbb{\beta}} \frac{1}{2N}\|\mathbb{y}-\mathbb{X}\mathbb{\beta}\|_2^2+\frac{\lambda}{2}\|\mathbb{\beta}\|_2^2}_{= L(\mathbb{\beta})}$$

$\mathbb{\beta}$で微分して

$$\frac{\partial L}{\partial \mathbb{\beta}} = -\frac{1}{N}\mathbb{X}^T(\mathbb{y}-\mathbb{X}\mathbb{\beta})+\lambda\mathbb{\beta}$$

$L(\mathbb{\beta})$は凸関数より微分して$\mathbb{0}$の点を求めて

$$\frac{\partial L}{\partial \mathbb{\beta}} = \mathbb{0}$$

$$\underbrace{(\mathbb{X}^T\mathbb{X}+N\lambda I)}_{\substack{\text{前題より} \\ \text{必ず逆行列が存在するので}}}\mathbb{\beta} = \mathbb{X}^T\mathbb{y}$$

$$\hat{\mathbb{\beta}} = (\mathbb{X}^T\mathbb{X}+N\lambda I)^{-1}\mathbb{X}^T\mathbb{y}$$

∎

15

(i) $$S(\mathbb{\beta}) = \sum_{i=1}^{N} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2$$

$$\mathbb{y} \in \mathbb{R}^N$$

$$\mathbb{X} = \begin{pmatrix} x_{11} & x_{12} \\ \vdots & \vdots \\ x_{N1} & x_{N2} \end{pmatrix} \in \mathbb{R}^{N \times 2}$$

$$\mathbb{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \in \mathbb{R}^2$$ として

$$S(\mathbb{\beta}) = \| \mathbb{y} - \mathbb{X}\mathbb{\beta} \|^2$$

Sは凸関数より
微分して0となる点を求めると

$$\frac{\partial S}{\partial \mathbb{\beta}} = (-2\mathbb{X}^T)(\mathbb{y} - \mathbb{X}\mathbb{\beta}) = 0$$

$$\hat{\mathbb{\beta}} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{y}$$

$$\hat{\mathbb{y}} = \mathbb{X}\hat{\mathbb{\beta}} = \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T \mathbb{y}$$

$$\begin{pmatrix} \sum_{i=1}^{N} x_{i1}(y_i - \hat{y}_i) \\ \sum_{i=1}^{N} x_{i2}(y_i - \hat{y}_i) \end{pmatrix} = \mathbb{X}^T(\mathbb{y} - \hat{\mathbb{y}}) = \mathbb{X}^T\mathbb{y} - \underbrace{\mathbb{X}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}}\mathbb{X}^T\mathbb{y}$$

$$= \mathbb{X}^T\mathbb{y} - \mathbb{X}^T\mathbb{y}$$

$$= 0$$

よって (i) は成り立つ.

(ii)

$\forall \boldsymbol{\beta} \in \mathbb{R}^2$ に対して

$$\hat{\boldsymbol{y}} = \mathbb{X}\hat{\boldsymbol{\beta}}$$

を用いると

$$\boldsymbol{y} - \mathbb{X}\boldsymbol{\beta} = \boldsymbol{y} - \mathbb{X}\boldsymbol{\beta} - \hat{\boldsymbol{y}} + \mathbb{X}\hat{\boldsymbol{\beta}}$$

$$= (\boldsymbol{y} - \hat{\boldsymbol{y}}) - \mathbb{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \quad \cdots (*)$$

よって各要素 i $(i=1,2,\cdots,N)$ について

$$y_i - \beta_1 x_{i1} - \beta_2 x_{i2} = y_i - \hat{y}_i - (\beta_1 - \hat{\beta}_1)x_{i1} - (\beta_2 - \hat{\beta}_2)x_{i2}$$

よって (ii) が成り立つ.

(*) のノルムを考えて

$$\|\boldsymbol{y} - \mathbb{X}\boldsymbol{\beta}\|_2^2 = \|(\boldsymbol{y} - \hat{\boldsymbol{y}}) - \mathbb{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\|_2^2$$

$$= \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 - 2(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \underbrace{\mathbb{X}^T(\boldsymbol{y} - \hat{\boldsymbol{y}})}_{=0} + \|\mathbb{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\|^2$$

$$= \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 + \underbrace{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \mathbb{X}^T\mathbb{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}_{2\text{次形式}}$$

$$= (\beta_1 - \hat{\beta}_1)^2 \sum_{i=1}^N x_{i1}^2 + 2(\beta_1 - \hat{\beta}_1)(\beta_2 - \hat{\beta}_2)\sum_{i=1}^N x_{i1}x_{i2}$$
$$+ (\beta_2 - \hat{\beta}_2)^2 \sum_{i=1}^N x_{i2}^2 + \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

が成り立つ ▥

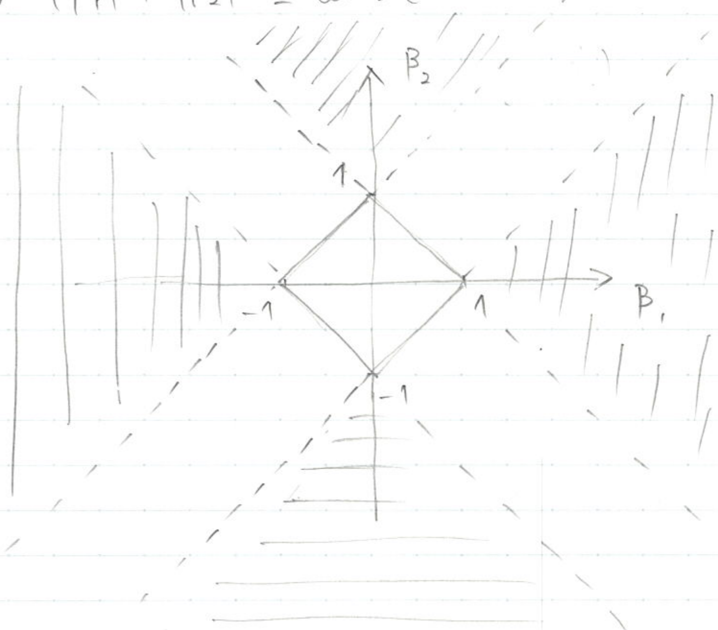
(h) $\sum_{i=1}^{N} x_{i1}^2 = \sum_{i=1}^{N} x_{i2}^2 = 1$, $\sum_{i=1}^{N} x_{i1} x_{i2} = 0$ の場合を考える.

目的関数

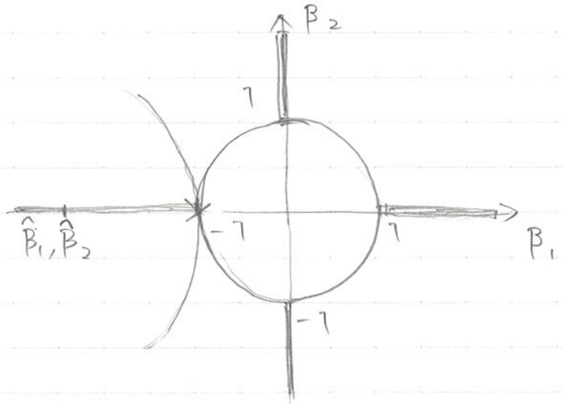
$$L(\beta) = \| y - X\beta \|^2$$

$$= (\beta_1 - \hat{\beta}_1)^2 + (\beta_2 - \hat{\beta}_2)^2 + \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

制約 $|\beta_1| + |\beta_2| \leq \text{const}$



(c) 制約式が円の場合



求めるべき $\hat{\beta}_1, \hat{\beta}_2$ は β_1, β_2 軸上かつ単位円外となる.

(16)

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

$$\frac{\partial L}{\partial \beta_k} = -\frac{2}{N}\sum_{i=1}^{N} x_{ik}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + 2\lambda\beta_k = 0$$

$$\frac{1}{N}\sum_{i=1}^{N} x_{ik}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 - \lambda\beta_k = 0$$

同様に

$$\frac{\partial L}{\partial \beta_l} = 0$$

$$\frac{1}{N}\sum_{i=1}^{N} x_{il}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 - \lambda\beta_l = 0$$

よって　$x_{il} = x_{ik}$ の時.

$$\beta_k = \frac{1}{N\lambda}\sum_{i=1}^{N} x_{ik}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$= \frac{1}{N\lambda}\sum_{i=1}^{N} x_{il}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 = \beta_l$$

Q.E.D.

$$\mathbb{X} = \underbrace{\left[\begin{matrix} \big| & \cdots & \big| \end{matrix} \right]}_{p} \Big\} N$$

⑱ elastic net

$$L(\boldsymbol{\beta}) = \frac{1}{2N} \|\boldsymbol{y} - \beta_0 - \mathbb{X}\boldsymbol{\beta}\|^2 + \lambda \left\{ \frac{1-\alpha}{2} \|\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_1 \right\}$$

$\mathbb{X}$, $\boldsymbol{y}$ の中心化をして $\beta_0 = 0$ とする.

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = -\frac{1}{N} \mathbb{X}^T (\boldsymbol{y} - \mathbb{X}\boldsymbol{\beta}) + \lambda(1-\alpha)\boldsymbol{\beta} + \lambda\alpha \frac{\partial}{\partial \boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1$$

j番目要素に対して

$$0 \in -\frac{1}{N} \underbrace{\left[\mathbb{X}^T (\boldsymbol{y} - \mathbb{X}\boldsymbol{\beta}) \right]_j} + \lambda(1-\alpha)\beta_j + \lambda\alpha \begin{cases} 1 & \beta_j > 0 \\ [-1, 1] & \beta_j = 0 \\ -1 & \beta_j < 0 \end{cases}$$

$$\downarrow$$

$$\sum_{i=1}^{N} x_{ij} \left(y_i - \sum_{k \neq j}^{p} x_{ik} \beta_k - x_{ij} \beta_j \right)$$

$$= \sum_{i=1}^{N} x_{ij} \left(r_{ij} - x_{ij} \beta_j \right)$$

$$0 \in -\frac{1}{N} \sum_{i=1}^{N} x_{ij} r_{ij} + \left\{ \frac{1}{N} \sum_{i=1}^{N} x_{ij}^2 + \lambda(1-\alpha) \right\} \beta_j + \lambda\alpha \begin{cases} 1 & \beta_j > 0 \\ [-1, 1] & \beta_j = 0 \\ -1 & \beta_j < 0 \end{cases}$$

ここでソフトしきい値関数の定義より

$$0 \in -S_j + t_j \beta_j + \mu \begin{cases} 1 & \beta_j > 0 \\ [-1, 1] & \beta_j = 0 \\ -1 & \beta_j < 0 \end{cases}$$

勾配を表している
右辺 = 0 を与えるような
β_j をみつけたい…
S_j はデータによって
変化するので
データによって極値を
与える β_j の値は変化する

(i) $\beta_j > 0$ の時

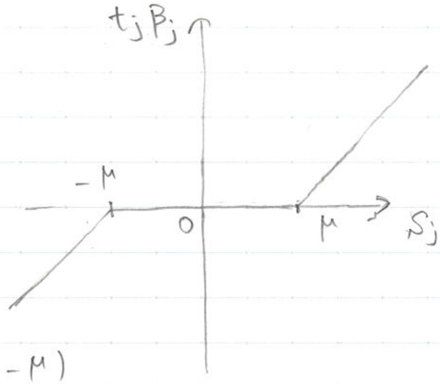
$$-S_j + t_j \beta_j + \mu = 0$$

$$t_j \beta_j = S_j - \mu \quad (S_j > \mu)$$

(ii) $\beta_j = 0$ の時

$$-S_j + \mu[-1, 1] \ni 0$$

$$-\mu \leq S_j \leq \mu$$



(iii) $\beta_j < 0$ の時

$$-S_j + t_j \beta_j - \mu = 0$$

$$t_j \beta_j = S_j + \mu \quad (S_j < -\mu)$$

$$t_j \beta_j = S_\mu(S_j)$$

ここで前ページで $S_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} r_{ij}$, $t_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 + \lambda(1-\alpha)$

$$t_j \beta_j = S_\mu(S_j)$$

$\mu = \lambda\alpha$ とすれば

$$\left\{ \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 + \lambda(1-\alpha) \right\} \beta_j = S_{\lambda\alpha}\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij} r_{ij} \right)$$

No.

Date

In [1]:
```
import copy
import japanize_matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import scipy
from matplotlib.pyplot import imshow
from numpy.random import randn
from scipy import stats
import matplotlib.pyplot as plt
```

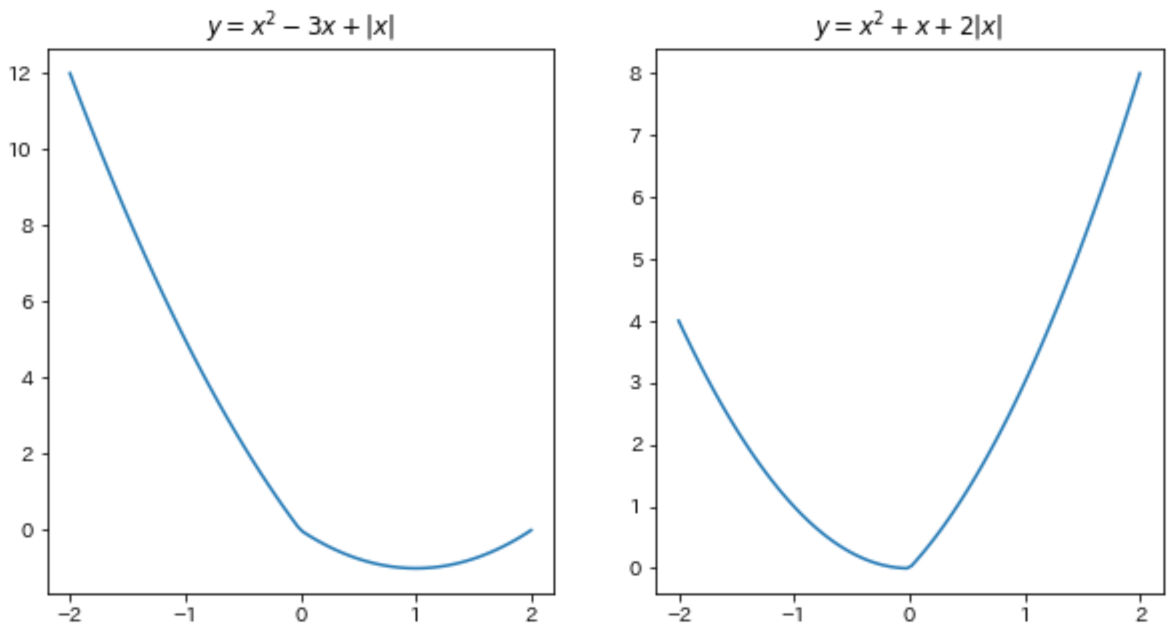
問題 1

In [2]:
```
def linear(X, y):
    p = X.shape[1]
    x_bar = np.zeros(p)
    for j in range(p):
        x_bar[j] = np.mean(X[:, j])
    for j in range(p):
        X[:, j] = X[:, j] - x_bar[j]     # Xの中心化
    y_bar = np.mean(y)
    y = y - y_bar                    # yの中心化
    beta = np.dot(                   #(1)
        np.linalg.inv(np.dot(X.T, X)),
        np.dot(X.T, y)
    )
    beta_0 = y_bar - np.dot(x_bar, beta)        #(2)
    return beta, beta_0
```

問題5

In [3]:
```
x = np.linspace(-2,2,100)

def f1(x):
    return x**2 - 3*x + np.abs(x)

def f2(x):
    return x**2 + x + np.abs(x)

fig, ax = plt.subplots(1,2, figsize = (10,5))
ax[0].plot(x,f1(x))
ax[0].set_title('$y = x^2 - 3x + |x|$')
ax[1].plot(x,f2(x))
ax[1].set_title('$y = x^2 + x + 2|x|$')
```

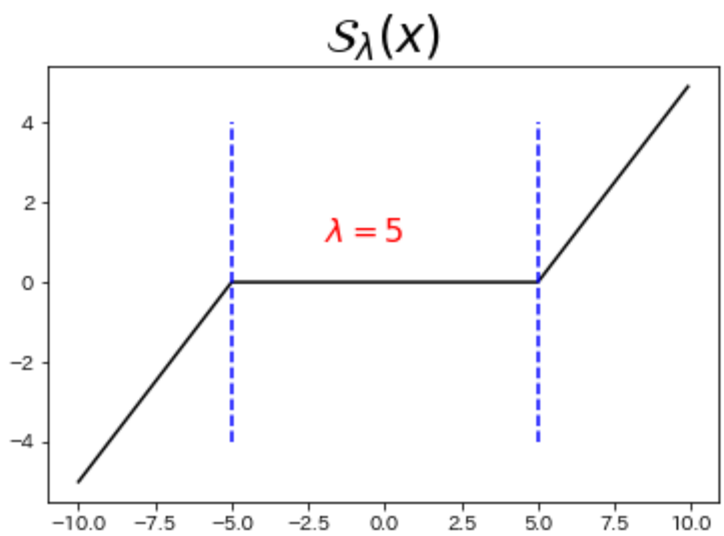
Out[3]:
```
Text(0.5, 1.0, '$y = x^2 + x + 2|x|$')
```



問題7

In [4]:
```
def soft_th(lam, x):
    return np.sign(x)*(np.maximum(np.abs(x)-lam, np.zeros_like(x)))


x = np.arange(-10, 10, 0.1)
y = soft_th(5, x)
plt.plot(x, y, c="black")
plt.title(r"${\cal S}_\lambda(x)$", size=24)
plt.plot([-5, -5], [-4, 4], c="blue", linestyle="dashed")
plt.plot([5, 5], [-4, 4], c="blue", linestyle="dashed")
plt.text(-2, 1, r"$\lambda=5$", c="red", size=16)
```

Out[4]: Text(-2, 1, '$\\lambda=5$')



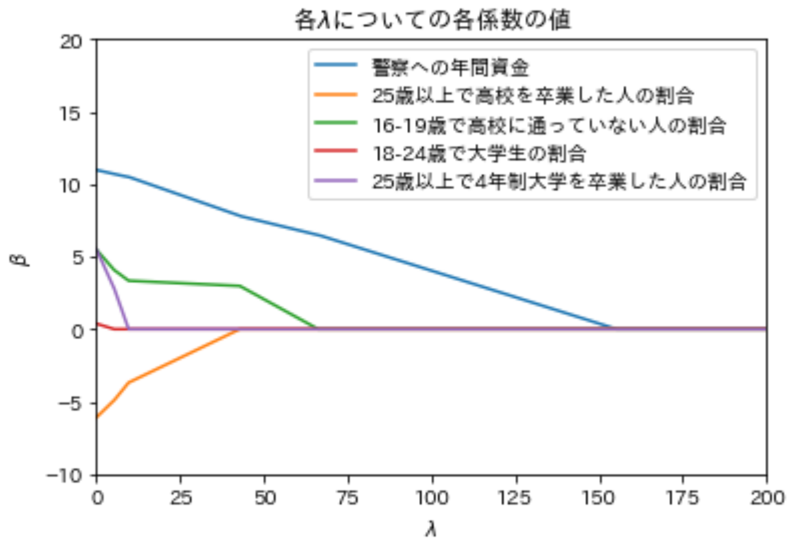
問題8

In [5]:
```
def linear_lasso(X, y, lam=0, beta=None):    #座標降下法によるlasso
    n, p = X.shape  #デザイン行列のサイズ
    if beta is None:
        beta = np.zeros(p)  #学習により求める重み
```

```
    X, y, X_bar, X_sd, y_bar = centralize(X, y)   # 中心化
    eps = 1
    beta_old = copy.copy(beta)   #同じポインタを参照しないための工夫
    while eps > 0.00001:   # このループの収束を待つ
        for j in range(p):
            r = y              #(1)
            for k in range(p):
                if j != k:
                    r = r - X[:, k] * beta[k]
            z = (np.dot(r, X[:, j]) / n) / (np.dot(X[:, j], X[:, j]) / n)
            beta[j] = soft_th(lam, z)
        eps = np.linalg.norm(beta - beta_old, 2) #更新前後での2ノルムの値で収束判定
        beta_old = copy.copy(beta)
    beta = beta / X_sd   # 各変数の係数を正規化前のものに戻す
    beta_0 = y_bar - np.dot(X_bar, beta)  #(2)
    return beta, beta_0

def centralize(X0, y0, standardize=True): #中心化
    X = copy.copy(X0)
    y = copy.copy(y0)
    n, p = X.shape
    X_bar = np.zeros(p)              # Xの各列の平均
    X_sd = np.zeros(p)               # Xの各列の標準偏差
    for j in range(p):
        X_bar[j] = np.mean(X[:, j])
        X[:, j] = X[:, j] - X_bar[j]     # Xの各列の中心化
        X_sd[j] = np.std(X[:, j])
        if standardize is True:
            X[:, j] = X[:, j] / X_sd[j]   # Xの各列の標準化
    if np.ndim(y) == 2:
        K = y.shape[1]
        y_bar = np.zeros(K)           # yの平均
        for k in range(K):
            y_bar[k] = np.mean(y[:, k])
            y[:, k] = y[:, k] - y_bar[k]  # yの中心化
    else:                          # yがベクトルの場合
        y_bar = np.mean(y)
        y = y - y_bar
    return X, y, X_bar, X_sd, y_bar


df = np.loadtxt("../data/crime.txt")
X = df[:,2:]
y = df[:,0]
p = X.shape[1]
y = df[:, 0]
lambda_seq = np.arange(0, 200, 0.1) #0 ~ 199.9の200個生成
plt.xlim(0, 200)
plt.ylim(-10, 20)
plt.xlabel(r"$\lambda$")
plt.ylabel(r"$\beta$")
plt.title(r"各$\lambda$についての各係数の値")
labels = ["警察への年間資金", "25歳以上で高校を卒業した人の割合",
          "16-19歳で高校に通っていない人の割合",
          "18-24歳で大学生の割合", "25歳以上で4年制大学を卒業した人の割合"]
r = len(lambda_seq)
coef_seq = np.zeros((r, p))
for i in range(r):
    coef_seq[i, :], _ = linear_lasso(X, y, lambda_seq[i])
for j in range(p):
    plt.plot(lambda_seq, coef_seq[:, j], label=labels[j])
plt.legend(loc="upper right")

print("lambda = 10の時の係数",coef_seq[100,:])
```

```
print("lambda = 50の時の係数",coef_seq[500,:])
print("lambda = 100の時の係数",coef_seq[1000,:])
```

```
lambda = 10の時の係数 [10.47248183 -3.64521939  3.33650908  0.          0.        ]
lambda = 50の時の係数 [ 7.40608466 -0.          2.0596716  -0.         -0.        ]
lambda = 100の時の係数 [ 4.03117071 -0.          0.         -0.         -0.        ]
```



結果より

$\lambda = 10$の時は4,5番目の特徴量が0

$\lambda = 50$の時は2番目の特徴量も0

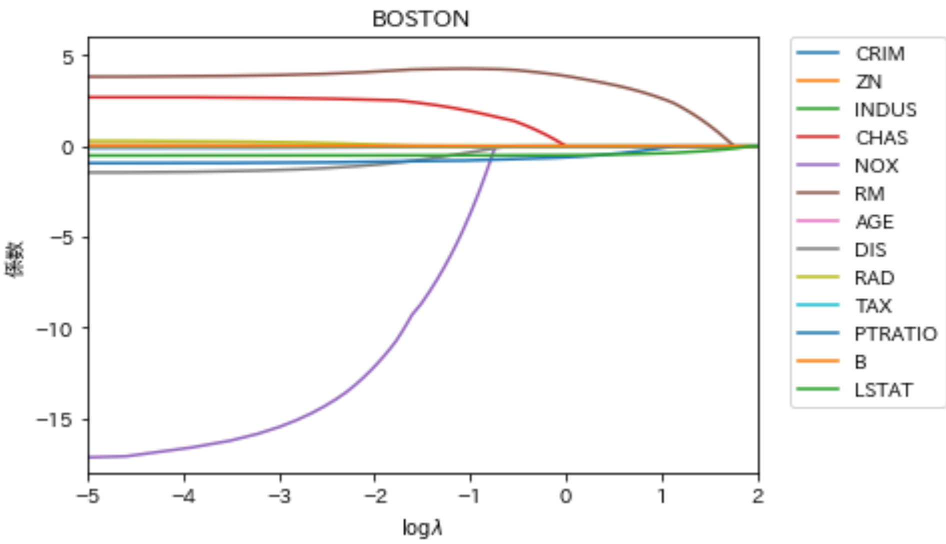
$\lambda = 50$の時は1番目の特徴量以外は全て0となる

問題8

In [6]:

```
#Boston datasetの読み込み
from sklearn.datasets import load_boston
boston = load_boston()
x = boston.data
y = boston.target
n, p = x.shape
lambda_seq = np.arange(0.0001, 20, 0.01)
r = len(lambda_seq)
plt.xlim(-5, 2)
plt.ylim(-18, 6)
plt.xlabel(r"$\log \lambda$")
plt.ylabel("係数")
plt.title("BOSTON")
labels = boston.feature_names
coef_seq = np.zeros((r, p))
for i in range(r):
    coef_seq[i, :], _ = linear_lasso(x, y, lam=lambda_seq[i])
for j in range(p):
    plt.plot(np.log(lambda_seq), coef_seq[:, j], label = labels[j])
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0, fontsize=10)
```

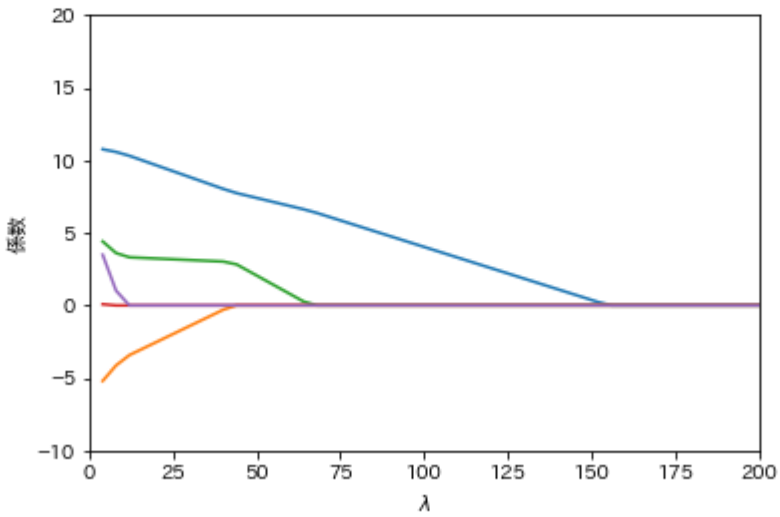
Out[6]: <matplotlib.legend.Legend at 0x7fd55f238ac0>



問題10

In [7]:

```
def warm_start(X, y, lambda_max=100):
    dec = np.round(lambda_max / 50)  #降り幅 1~maxを50等分する
    lambda_seq = np.arange(lambda_max, 1, -dec)
    r = len(lambda_seq)
    p = X.shape[1]
    beta = np.zeros(p)
    coef_seq = np.zeros((r, p))
    for k in range(r):
        beta, _ = linear_lasso(X, y, lambda_seq[k], beta)
        coef_seq[k, :] = beta   #k番目のlambdaの値における重み
    return coef_seq

df = np.loadtxt("../data/crime.txt", delimiter="\t")
X = df[:, [i for i in range(2, 7)]]
p = X.shape[1]
y = df[:, 0]
coef_seq = warm_start(X, y, 200)  #返り値が各ラムダにおける重み配列
lambda_max = 200
dec = round(lambda_max / 50)
lambda_seq = np.arange(lambda_max, 1, -dec)
plt.ylim(np.min(coef_seq), np.max(coef_seq))
plt.xlabel(r"$\lambda$")
plt.ylabel("係数")
plt.xlim(0, 200)
plt.ylim(-10, 20)
for j in range(p):
    plt.plot(lambda_seq, coef_seq[:, j])
```



問題13

In [8]:
```
def ridge(X, y, lam=0):
    n, p = X.shape
    X, y, X_bar, X_sd, y_bar = centralize(X, y)
    beta = np.dot(
        np.linalg.inv(np.dot(X.T, X) + n * lam * np.eye(p)),
        np.dot(X.T, y)
    )
    beta = beta / X_sd
    beta_0 = y_bar - np.dot(X_bar, beta)
    return beta, beta_0

df = np.loadtxt("../data/crime.txt", delimiter="\t")
X = df[:, [i for i in range(2, 7)]]
y = df[:, 0]
linear(X, y)
```

Out[8]:
```
(array([10.98067026, -6.08852939,  5.4803042 ,  0.37704431,  5.50047122]),
 489.6485969690334)
```

In [9]:
```
ridge(X, y)
```

Out[9]:
```
(array([10.98067026, -6.08852939,  5.4803042 ,  0.37704431,  5.50047122]),
 717.96)
```

In [10]:
```
ridge(X, y, 200)
```

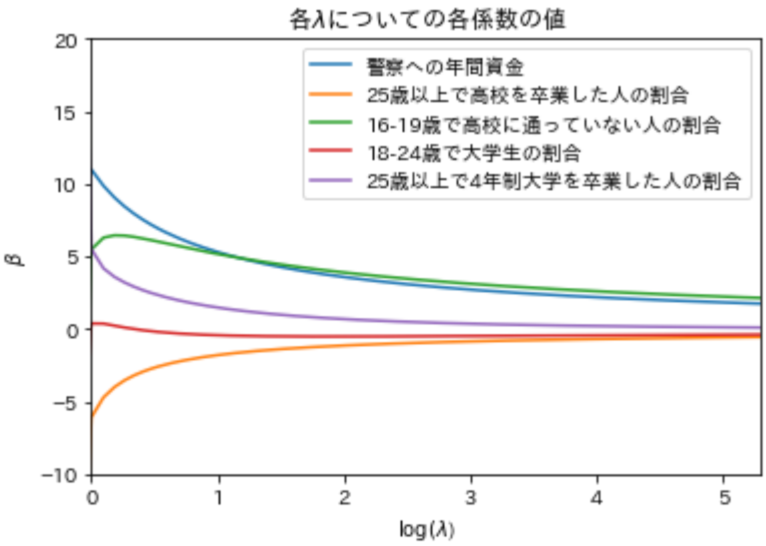
Out[10]:
```
(array([ 0.0563518 , -0.01976397,  0.07786309, -0.0171218 , -0.0070393 ]),
 717.96)
```

問題14

In [11]:
```
df = np.loadtxt("../data/crime.txt", delimiter="\t")
X = df[:, [i for i in range(2, 7)]]
p = X.shape[1]
y = df[:, 0]
eps = 1e-10 #log(0)を防止するための微小項
lambda_seq = np.log(np.arange(0, 200, 0.1) + eps)
plt.xlim(0, np.log(200))
plt.ylim(-10, 20)
plt.xlabel(r"$\log(\lambda$)")  #横軸の表示を変更
plt.ylabel(r"$\beta$")
```

```
plt.title(r"各$\lambda$についての各係数の値")
labels = ["警察への年間資金", "25歳以上で高校を卒業した人の割合",
          "16-19歳で高校に通っていない人の割合",
          "18-24歳で大学生の割合", "25歳以上で4年制大学を卒業した人の割合"]
r = len(lambda_seq)
beta = np.zeros(p)
coef_seq = np.zeros((r, p))
for i in range(r):
    beta, beta_0 = ridge(X, y, lambda_seq[i])
    coef_seq[i, :] = beta
for j in range(p):
    plt.plot(lambda_seq, coef_seq[:, j], label=labels[j])
plt.legend(loc="upper right")
```

Out[11]: <matplotlib.legend.Legend at 0x7fd55f9640a0>



問題17

$z_1, z_2 \sim \mathcal{N}(0, 1)$

x_1, x_2, x_3はz_1にノイズが乗っている

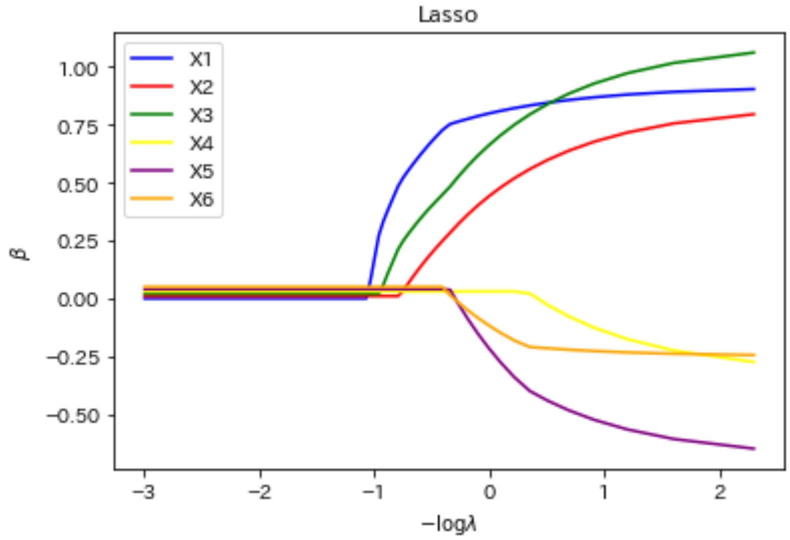
x_4, x_5, x_6はz_2にノイズが乗っている

In [12]:

```
np.random.seed(42)
n = 500
x = np.zeros((n, 6))  #特徴量次元 = 6
z = np.zeros((n, 2))  #これは2で十分
for k in range(2):
    z[:, k] = np.random.randn(n)
y = 3 * z[:, 0] - 1.5 * z[:, 1] + 2 * np.random.randn(n)  #(1)
for j in range(3):
    x[:, j] = z[:, 0] + np.random.randn(n) / 5
for j in range(3, 6):
    x[:, j] = z[:, 1] + np.random.randn(n) / 5
lambda_seq = np.arange(0.1, 20, 0.1)
p = 6
r = len(lambda_seq)
coef_seq = np.zeros((r, p))
cols = ["blue", "red", "green", "yellow", "purple", "orange"]
for i in range(r):
    coef_seq[i, :], _ = linear_lasso(x, y, lambda_seq[i])  #(2)
for j in range(p):
    plt.plot(-np.log(lambda_seq), coef_seq[:, j] + 0.01 * j,
             c=cols[j], label="X"+str(j+1))
plt.xlabel(r"$-\log \lambda$")
```

```
plt.ylabel(r"$\beta$")
plt.legend(loc="upper left")
plt.title("Lasso")
```

Out[12]: Text(0.5, 1.0, 'Lasso')



問題18

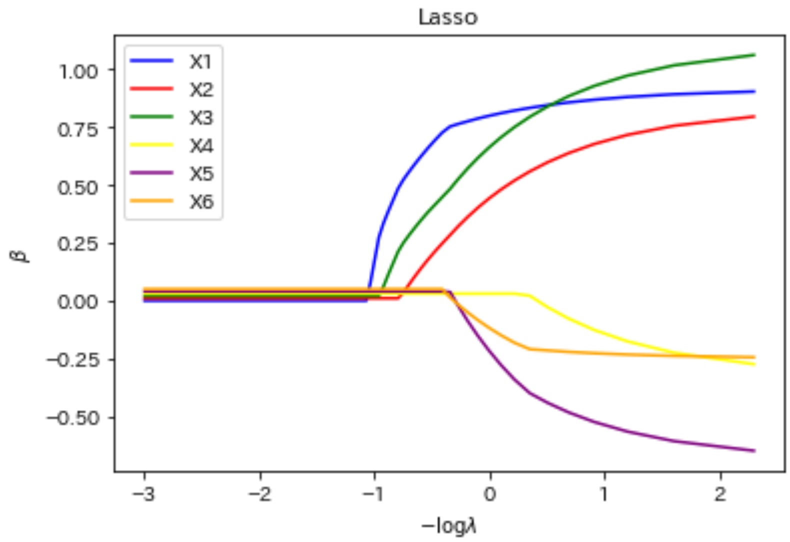
In [13]:
```
def linear_lasso_kai(X, y, lam=0, beta=None, alpha = 1):    #問題8のlinear_lassoの改良版
    n, p = X.shape  #デザイン行列のサイズ
    if beta is None:
        beta = np.zeros(p)  #学習により求める重み
    X, y, X_bar, X_sd, y_bar = centralize(X, y)  # 中心化
    eps = 1
    beta_old = copy.copy(beta)   #同じポインタを参照しないための工夫
    while eps > 0.00001:   # このループの収束を待つ
        for j in range(p):
            r = y             #(1)
            for k in range(p):
                if j != k:
                    r = r - X[:, k] * beta[k]
            z = (np.dot(r, X[:, j]) / n) / ((np.dot(X[:, j], X[:, j]) / n) + lam*(1-alpha)/np.sqrt((y-y_bar)
            beta[j] = soft_th(alpha*lam, z)
        eps = np.linalg.norm(beta - beta_old, 2) #更新前後での2ノルムの値で収束判定
        beta_old = copy.copy(beta)
    beta = beta / X_sd   # 各変数の係数を正規化前のものに戻す
    beta_0 = y_bar - np.dot(X_bar, beta)  #(2)
    return beta, beta_0
```

$\alpha = 1$の時に問題17と同じ挙動を示すかテストする

In [14]:
```
np.random.seed(42)
n = 500
x = np.zeros((n, 6))  #特徴量次元 = 6
z = np.zeros((n, 2))  #これは2で十分
for k in range(2):
    z[:, k] = np.random.randn(n)
y = 3 * z[:, 0] - 1.5 * z[:, 1] + 2 * np.random.randn(n)  #(1)
for j in range(3):
    x[:, j] = z[:, 0] + np.random.randn(n) / 5
for j in range(3, 6):
    x[:, j] = z[:, 1] + np.random.randn(n) / 5
lambda_seq = np.arange(0.1, 20, 0.1)
p = 6
r = len(lambda_seq)
```

```
coef_seq = np.zeros((r, p))
cols = ["blue", "red", "green", "yellow", "purple", "orange"]
for i in range(r):
    coef_seq[i, :], _ = linear_lasso_kai(x, y, lambda_seq[i],alpha = 1)  #(2)
for j in range(p):
    plt.plot(-np.log(lambda_seq), coef_seq[:, j] + 0.01 * j,
             c=cols[j], label="X"+str(j+1))
plt.xlabel(r"$-\log \lambda$")
plt.ylabel(r"$\beta$")
plt.legend(loc="upper left")
plt.title("Lasso")
```

Out[14]: Text(0.5, 1.0, 'Lasso')



問題19

In [15]:

```
def elastic_net(X, y, lam=0, alpha=1, beta=None):               #
    n, p = X.shape
    if beta is None:
        beta = np.zeros(p)
    X, y, X_bar, X_sd, y_bar = centralize(X, y)   # 中心化
    eps = 1
    beta_old = copy.copy(beta)
    while eps > 0.00001:    # このループの収束を待つ
        for j in range(p):
            r = y
            for k in range(p):
                if j != k:
                    r = r - X[:, k] * beta[k]
            z = (np.dot(r, X[:, j]) / n)
            beta[j] = (soft_th(lam * alpha, z)                  ##
                      / (np.dot(X[:, j], X[:, j]) / n + (1-alpha) * lam))  ## elastic_netでの変更点
        eps = np.linalg.norm(beta - beta_old, 2)
        beta_old = copy.copy(beta)
    beta = beta / X_sd   # 各変数の係数を正規化前のものに戻す
    beta_0 = y_bar - np.dot(X_bar, beta)
    return beta, beta_0
```

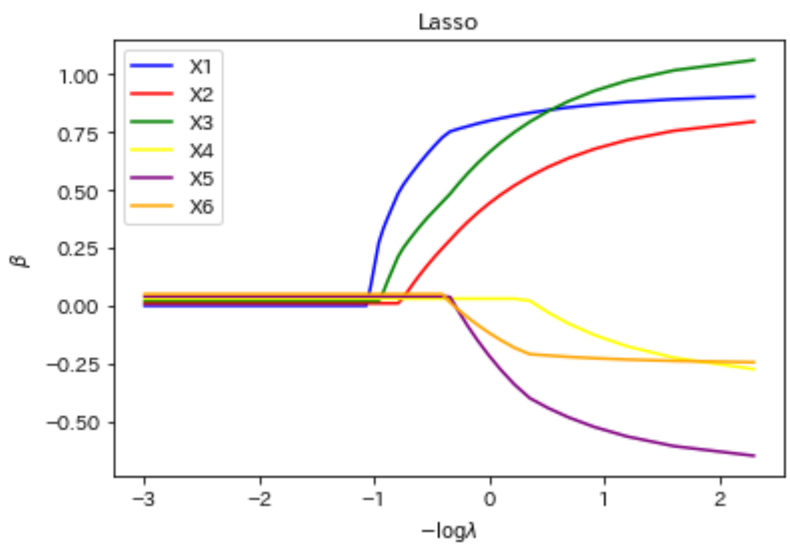
$\alpha = 1$(lasso回帰)として問題17と同じ挙動を示すかテスト

In [16]:

```
np.random.seed(42)
n = 500
x = np.zeros((n, 6))  #特徴量次元 = 6
z = np.zeros((n, 2))  #これは2で十分
for k in range(2):
```

```
    z[:, k] = np.random.randn(n)
y = 3 * z[:, 0] - 1.5 * z[:, 1] + 2 * np.random.randn(n)  #(1)
for j in range(3):
    x[:, j] = z[:, 0] + np.random.randn(n) / 5
for j in range(3, 6):
    x[:, j] = z[:, 1] + np.random.randn(n) / 5
lambda_seq = np.arange(0.1, 20, 0.1)
p = 6
r = len(lambda_seq)
coef_seq = np.zeros((r, p))
cols = ["blue", "red", "green", "yellow", "purple", "orange"]
for i in range(r):
    coef_seq[i, :], _ = elastic_net(x, y, lambda_seq[i],alpha = 1)  #(2)
for j in range(p):
    plt.plot(-np.log(lambda_seq), coef_seq[:, j] + 0.01 * j,
             c=cols[j], label="X"+str(j+1))
plt.xlabel(r"$-\log \lambda$")
plt.ylabel(r"$\beta$")
plt.legend(loc="upper left")
plt.title("Lasso")
```

Out[16]: Text(0.5, 1.0, 'Lasso')



問題20

In [17]:
```
def cv_linear_lasso(x, y, alpha=1, k=10, retall = False):  # k-fold
    print(np.dot(x.T,x).shape)
    lam_max = np.max(np.dot(x.T, y) / np.dot(x.T, x))  #最初の非ゼロが現れるlambdaから始める
    lam_seq = np.arange(0, 1, 0.01)*lam_max
    n = len(y)
    m = int(n/k) #一つのバッチサイズ
    r = n%k
    S_min = np.inf
    obj_his = [] #目的関数の更新履歴

    for lam in lam_seq:
        S = 0 #暫定損失関数
        for i in range(k):
            index = range(i*m, i*m + m)             #validation用のデータindex
            _index = list(set(range(n)) - set(index))  #学習用のデータindex
            beta, beta0 = elastic_net(x[_index], y[_index], lam, alpha)
            z = np.linalg.norm((y[index] - beta0 - np.dot(x[index], beta)), 2)
            S = S + z
        obj_his.append(S)
        if S_min > S:
```

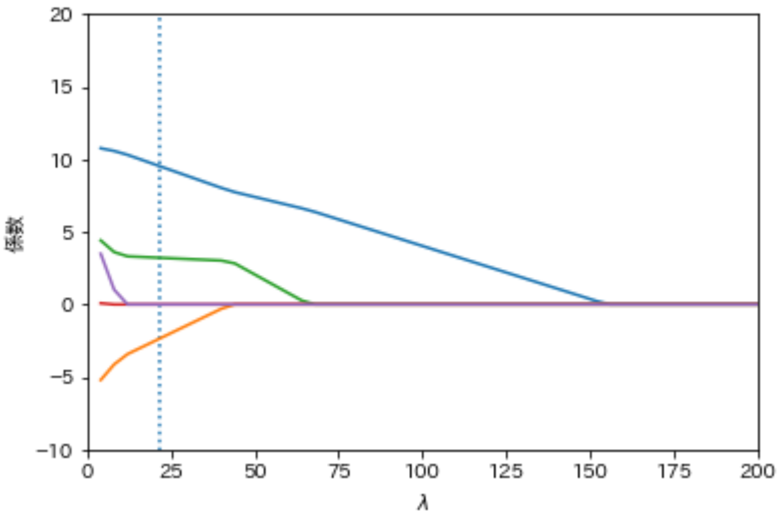
```
            S_min = S.copy()
            lam_best = lam.copy()
            beta0_best = beta0.copy()
            beta_best = beta.copy()

    if retall == True:  #履歴を返す用
        return lam_best, beta0_best, beta_best, S_min ,lam_seq, obj_his

    return lam_best, beta0_best, beta_best, S_min
```

cross_validationのテスト

In [18]:

```
df = np.loadtxt("../data/crime.txt", delimiter="\t")
X = df[:, [i for i in range(2, 7)]]
p = X.shape[1]
y = df[:, 0]
lam, beta0, beta, S = cv_linear_lasso(X, y)
print(lam)
print(beta0)
print(beta)
coef_seq = warm_start(X, y, 200)
lambda_max = 200
dec = round(lambda_max / 50)
lambda_seq = np.arange(lambda_max, 1, -dec)
plt.ylim(np.min(coef_seq), np.max(coef_seq))
plt.xlabel(r"$\lambda$")
plt.ylabel("係数")
plt.xlim(0, 200)
plt.ylim(-10, 20)
plt.axvline(x=lam, ymin=-10, ymax=10, linestyle="dotted")
for j in range(p):
    plt.plot(lambda_seq, coef_seq[:, j])
```

```
(5, 5)
21.797349629779895
527.046543093802
[ 9.51570956 -3.81437924  2.65954288 -0.         0.        ]
```



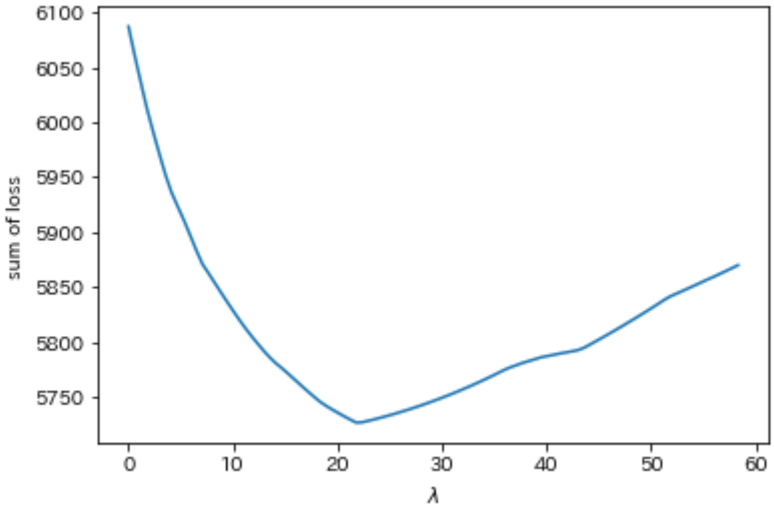
In [19]:

```
df = np.loadtxt("../data/crime.txt", delimiter="\t")
X = df[:, [i for i in range(2, 7)]]
p = X.shape[1]
y = df[:, 0]
lam, beta0, beta, S, lam_seq, objval = cv_linear_lasso(X, y, retall = True)
coef_seq = warm_start(X, y, 200)
lambda_max = 200
dec = round(lambda_max / 50)
```

```
plt.plot(lam_seq, objval)
plt.xlabel(r"$\lambda$")
plt.ylabel("sum of loss")
```

(5, 5)

Out[19]: Text(0, 0.5, 'sum of loss')



実際に$\lambda = 21$くらいで最小値を取っていることが確認できる