

線形最小二乗法の諸問題 (I)¹⁾

関 本 年 彦

本論は、線形最小二乗法について、(1)数理的基礎、(2)アルゴリズム、(3)統計学的考察、の三つの側面から論ずる。

問題解決の場において、変数 $x=(\xi_1, \dots, \xi_n)$ の変動が変数 y の挙動を決定する方程式 (モデル) $y=f(x)=f(\xi_1, \dots, \xi_n)$ は²⁾、問題解決に当たる者が与えられたデータに基づいて知識、経験を駆使した総合的な判断から、新たに考案したり、あるいは問題の分野に応じてよく用いられるいくつかの候補の中から適当と思われるものを選択する。モデルは通常いくつかのパラメータを含むので、つぎには、与えられたデータをあてはめてみて「最適モデル」となるようにパラメータの値を決定する。このようなパラメータの決定作業を、データのモデルへのあてはめ、あるいは単にデータのあてはめ (data fitting) と呼んでいる。

線形最小二乗法は、

$$(*) \quad f(x; \beta) = \beta_0 + \beta_1 g_1(x) + \dots + \beta_n g_n(x)$$

のようにパラメータ $\beta=(\beta_0, \beta_1, \dots, \beta_n)$ について線形であるモデルを扱う。その「最適モデル」とは、与えられたデータ

$$(y_i, x_i) = (y_i, \xi_{i1}, \dots, \xi_{in}), \quad (i=1, \dots, m)$$

について

-
- 1) この論文は、成城大学教員特別助成による研究成果の一端をまとめたものである。
 - 2) x はベクトル変数、 ξ_i はスカラー変数、 f は多変数スカラー値関数であるが、一般論を述べるためにこうしたに過ぎない。

$$(**) \quad \Phi(\beta) = \sum_{i=1}^m (y_i - f(x_i, \beta))^2$$

を最小とするようにパラメータ β を定めたモデルのことである。線形最小二乗法は、 $f(x; \beta)$ が β について線形であれば、 x については必ずしも線形でなくても、すなわち (*) の $g_i(x)$ はどのような関数であっても適用できる。ただし、本論では特別な形をもった g_i についての議論には立ち入らないので、以下では

$$f(x; \beta) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

のように x についても線形（正確には、アフィン線形）であるモデルを対象にする。

(**) の $\Phi(\beta)$ を最小にするパラメータのモデルを最適とすることについては、数理的議論を展開する過程で、十分説得性をもつ選択であることが判明する筈であるが、統計学からもいくつかの支持が得られている。すなわち、計測誤差 $y_i - f(x_i; \beta)$ を確率変数 ε_i とし ($i=1, \dots, m$)、各 ε_i の期待値 $\mathbf{E}(\varepsilon_i) = 0$ 、と各 $\varepsilon_i, \varepsilon_j$ ($1 \leq i, j \leq m$) について $\mathbf{E}(\varepsilon_i \varepsilon_j) = \sigma^2 \delta_{ij}$ ³⁾ を仮定すると、 $\Phi(\beta)$ を最小にする $\beta = b = (b_0, b_1, \dots, b_n)$ の各 b_i は、 y の線形関数⁴⁾ であり、 β_i の不偏推定量となるが（線形不偏推定量）、 β_i の線形不偏推定量の中ではその分散が最小である（最小分散線形不偏推定量）。さらに、各 ε_i が正規分布にしたがうという仮定をつけ加えるならば、各 b_i は β_i の最尤推定量であり、また、 β_i のすべての不偏推定量の中で最小の分散をもつものであることが知られている。

本論の内容は以下の通りであるが、紙数の関係で5.3以降は続稿にゆずる。なお、以後線形最小二乗法を単に最小二乗法ということにする。

3) クロネッカーのデルタ。 $\delta_{ij} = 1 (i=j), = 0 (i \neq j)$ 。

4) 後述 (5.2) 参照。

§ 1 概要

§ 2 特異値分解

§ 3 擬逆写像

§ 4 行列の列ベクトルが生成する部分空間への射影

§ 5 最小二乗解のアルゴリズム

5.1 最小二乗解

5.2 Householder 変換による行列の2重対角化

5.3 QR 法

§ 6 モデル検証法

最小二乗法が基盤とする数理的内容は、(1)特異値分解、(2)擬逆写像、(3)射影、の3項目に大別される。以上のうち § 2～4 は、これらに関する議論である。筆者は、§ 5 をもとに C 言語によるプログラムを作成したが、これは現在東京大学共同利用大型機計算センター副システムのファイルにあって公開してある。

§ 1 概要

はじめに、本論で用いる数学的記号について述べておく。

R は実数の集合。 R^n は n 次元実数ユークリッド空間。 $R^{m \times n}$ は $m \times n$ 実数行列の集合。 I_n は n 次単位行列で、前後の関係から次数が明らかな場合は I とも記す。 A が行列のとき、 $'A$ は A の転置行列。 R^n の要素は n 次元列ベクトルで、 $'(\xi_1, \dots, \xi_n)$ (ただし、各 ξ_i は実数) のように表すことができる。 $x = '(\xi_1, \dots, \xi_n)$, $y = '(\eta_1, \dots, \eta_n) \in R^n$ に対して、内積 $\xi_1\eta_1 + \dots + \xi_n\eta_n = 'xy$ を $\langle x, y \rangle_n$ で表し、次数 n が明らかならば、 $\langle x, y \rangle$ とも記す。 $\sqrt{\langle x, y \rangle_n}$ をベクトル x の長さまたはノルムといい、 $\|x\|$ と記す。各 $1 \leq i \leq n$ に対して、 $e_i \in R^n$ を第 i 要素が 1 でそれ以外の要素は 0 であるベクトルとするとき、 e_1, \dots, e_n から成る R^n の基底を R^n の自然基底という。

線形最小二乗法の諸問題 (I)

最小二乗法を数学的にいえば、既知の行列⁵⁾ $X \in \mathbf{R}^{m \times n}$ とベクトル $y \in \mathbf{R}^m$ から

$$(1.1) \quad \|y - X\beta\|$$

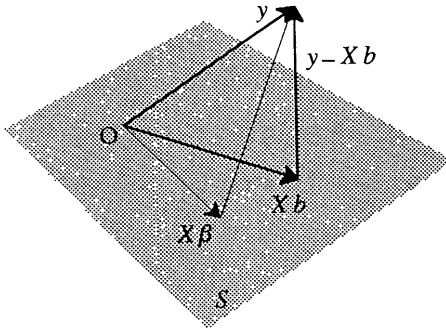
を最小にするようなベクトル $\beta \in \mathbf{R}^n$ (以下では最小二乗解ともいう) を求める手法である⁶⁾。行列 X の列ベクトルを $x_1, \dots, x_n \in \mathbf{R}^m$ とし、 S を x_1, \dots, x_n が生成する \mathbf{R}^m の部分空間とすると、

$$(1.2) \quad y - Xb \in S^\perp \quad (S^\perp \text{ は } S \text{ の直交補空間})$$

であるような $b \in \mathbf{R}^n$ は、つぎの不等式 (1.3) から (1.1) の最小二乗解であることがわかる。

$$(1.3) \quad \|y - X\beta\|^2 = \|y - Xb\|^2 + \|Xb - X\beta\|^2 \geq \|y - Xb\|^2.$$

また、この不等式から、(1.2) を満たす最小二乗解 b が存在するとして、他のいずれの最小二乗解 b' についても



$$(1.4) \quad Xb' = Xb$$

であることがわかる。(1.2) は、すべての $\beta \in \mathbf{R}^n$ に対して $\langle y - Xb, X\beta \rangle_m = \langle X(y - Xb), \beta \rangle_m = 0$, すなわち $\langle XXb, \beta \rangle_m = \langle Xy, \beta \rangle_m$ と書き直せるから、

5) 最小二乗法の応用のひとつである回帰分析法では、計画行列 (design matrix) ともいう。

6) $y = (y_1, \dots, y_m)$, $\beta = (\beta_1, \dots, \beta_n)$, $X = (\xi_{ij})$ において $y_i = (\xi_{i1}\beta_1 + \dots + \xi_{in}\beta_n)$, ($i=1, \dots, m$) を行列の記法で書いたものが $y - X\beta$ である。このとき、上述の $\Phi(\beta)$ は $\|y - X\beta\|^2$ である。

b は

$$(1.5) \quad 'XX\beta = 'Xy$$

を β について解いたものである。これを正規方程式 (normal equation) といい、結局最小二乗法は正規方程式を (β について) 解くことに帰着することがわかる。

ところで、正規方程式 (1.5) は連立1次方程式であるから、後は一見取り立てて問題はなさそうに見えるが、この方程式を数値的に解くにはまず行列 $'XX$ を計算しなければならず、この計算については X の要素の誤差が $'XX$ の要素の誤差に大きな影響を及ぼす場合があることが知られている。そこで近年、正規方程式 (1.5) を解く方法として行列の特異値分解のアルゴリズムによるものが開発され、これが現在もっとも優れた最小二乗法のアルゴリズムとされている。この方法の特徴は、正規方程式を解くのに $'XX$ のかわりに X を対角化することによって X の要素の誤差の影響を抑制しているところにある。

§ 2 特異値分解 (singular value decomposition—SVD)

この § では、 $m > n$ とする。

定理 2.1 (特異値分解) 任意の線形写像 $f: R^n \rightarrow R^m$ は、 R^m , R^n の適当な正規直交基底により対角要素は非負で対角要素以外は 0 である $m \times n$ 行列⁷⁾

$$(2.1) \quad D = \text{diag}(\sigma_1, \dots, \sigma_n) = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ & & & 0 \end{pmatrix}, \quad \sigma_1 \geq \dots \geq \sigma_n \geq 0$$

で表すことができる。

7) 対角要素以外は 0 である $m \times n$ 行列を、以後 $m \times n$ 対角行列ということにする。

線形最小二乗法の諸問題 (1)

証明 ff は \mathbf{R}^n 上の対称変換で半正值⁸⁾ であるから $\sigma_1^2 \geq \dots \geq \sigma_r^2 > \sigma_{r+1}^2 = \dots = \sigma_n^2 = 0$ ($\sigma_i > 0, 1 \leq i \leq r \leq n$) であるような固有値⁹⁾ をもつ。 $v_1, \dots, v_n \in \mathbf{R}^n$ をたがいに直交する $\sigma_i^2 (1 \leq i \leq n)$ に対応する単位固有ベクトルとすると、 $1 \leq i, k \leq r$ ならば

$$\begin{aligned} \langle \sigma_i^{-1} f(v_i), \sigma_k^{-1} f(v_k) \rangle &= (\sigma_i \sigma_k)^{-1} \langle v_i, ff(v_k) \rangle \\ &= (\sigma_i \sigma_k)^{-1} \sigma_k^2 \langle v_i, v_k \rangle = \delta_{ik} \end{aligned}$$

(δ_{ik} はクロネッカーのデルタ) であるから、 $\sigma_1^{-1} f(v_1), \dots, \sigma_r^{-1} f(v_r) \in \mathbf{R}^m$ はたがいに直交する単位ベクトルである。そこで、これらを u_1, \dots, u_r とおき、さらに、 u_1, \dots, u_m がたがいに直交するように単位ベクトル u_{r+1}, \dots, u_m をつけ加える。以上の $u_1, \dots, u_m, v_1, \dots, v_n$ に対して

$$(2.2) \quad \langle u_i, f(v_k) \rangle = \begin{cases} \sigma_k \delta_{ik}, & 1 \leq i \leq n, \\ 0, & n+1 \leq i \leq m, \end{cases} \quad 1 \leq k \leq n$$

が成り立つ。□

系 任意の行列 $A \in \mathbf{R}^{m \times n}$ は、適当な m 次直交行列 U と n 次直交行列 V を用いて

$$(2.3) \quad A = UD^tV$$

と分解できる。 D は (2.1) で与えられる $m \times n$ 対角行列である。

証明 U, V として、上の定理の証明の $u_1, \dots, u_m, v_1, \dots, v_n$ を列ベクトル

- 8) \mathbf{R}^n 上の線形変換 g は、任意の $x \in \mathbf{R}^n$ に対して $\langle g(x), x \rangle \geq 0$ であるとき半正值であるという。
- 9) \mathbf{R}^n 上の対称変換 g の固有方程式の解は実数であることの略証。内積の定義を複素ユークリッド空間へ拡張して $x = (\xi_1, \dots, \xi_n), y = (\eta_1, \dots, \eta_n) \in \mathbf{C}^n$ に対して $\langle x, y \rangle = \xi_1 \bar{\eta}_1 + \dots + \xi_n \bar{\eta}_n$ とし、 λ を固有方程式の一つの解とすると、あるゼロでないベクトル $x \in \mathbf{C}^n$ に対して $\lambda \langle x, x \rangle = \langle g(x), x \rangle = \langle x, g(x) \rangle = \bar{\lambda} \langle x, x \rangle$ であるから λ は実数である。

\mathbf{R}^n 上の対称変換 g が n 個のたがいに直交する固有ベクトルをもつことの略証。 n に関する帰納法による。 u を g の固有ベクトル、 $\mathbf{R}u$ を u が生成する部分空間とすると、 $\mathbf{R}u$ の直交補空間 $(\mathbf{R}u)^\perp$ は g -不変 ($g((\mathbf{R}u)^\perp) \subseteq (\mathbf{R}u)^\perp$) である。 g を $n-1$ 次元部分空間 $(\mathbf{R}u)^\perp$ 上の線形変換と見れば、帰納法の仮定から $n-1$ 個のたがいに直交する固有ベクトルをもち、これらは u とも直交する。

とする行列を採ればよい。□

註2. 1 U としてかならずしも正方行列ではなく、 u_1, \dots, u_n を列ベクトルとする $m \times n$ 列を採ると、(2. 3) の D は $\sigma_1, \dots, \sigma_n$ を対角要素とする n 次正方対角行列となる。この U と D による分解の方を A の特異値分解ということもある。

$$(2. 4) \quad A = \begin{pmatrix} & U \\ & \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix} \begin{pmatrix} & V \\ & \end{pmatrix}$$

註2. 2 $\sigma_i (1 \leq i \leq n)$ を行列 A の特異値といい、行列 U の列ベクトル、すなわち $u_i (1 \leq i \leq n)$ を A の左特異ベクトル、また行列 V の列ベクトル、すなわち $v_i (1 \leq i \leq n)$ を A の右特異ベクトルという。 A の左特異ベクトルは、行列 $A'A$ の正規直交固有ベクトルであり、 A の右特異ベクトルは、行列 AA' の正規直交固有ベクトルである。

さらに、

$$AV = UD = (\sigma_1 u_1, \dots, \sigma_n u_n), \quad 'AU = VD = (\sigma_1 v_1, \dots, \sigma_n v_n)$$

から、各 $1 \leq i \leq n$ に対して

$$Av_i = \sigma_i u_i, \quad 'Au_i = \sigma_i v_i$$

が得られる。

§ 3 擬逆写像

$f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ を線形写像とする。 f は、標準線形写像¹⁰⁾ $\varphi: \mathbf{R}^n \rightarrow \mathbf{R}^n / \text{Ker}(f)$ と標準単射¹¹⁾ $j: \mathbf{R}^n / \text{Ker}(f) \rightarrow \mathbf{R}^m$ により

$$(3. 1) \quad \mathbf{R}^n \xrightarrow{\varphi} \mathbf{R}^n / \text{Ker}(f) \xrightarrow{\bar{f}} f(\mathbf{R}^n) \xrightarrow{j} \mathbf{R}^m$$

と分解でき、 $x + \text{Ker}(f) \rightarrow f(x)$ で定義される線形写像 \bar{f} は正則である。し

10) 各 $x \in \mathbf{R}^n$ に対して $\varphi(x) = x + \text{Ker}(f)$ 。なお、 $\text{Ker}(f) = \{x \in \mathbf{R}^n: f(x) = 0\}$ で、 f の零空間または核空間と呼ばれる。

11) 各 $y \in f(\mathbf{R}^n)$ に対して $j(y) = y$ 。

たがって、 $p: \mathbf{R}^m \rightarrow f(\mathbf{R}^n)$ を正射影とすると、線形写像 $\bar{f}^{-1} \cdot p: \mathbf{R}^m \rightarrow \mathbf{R}^n / \text{Ker}(f)$ が定義できる。さらに、任意のベクトル $x \in \mathbf{R}^n$ を $x = x_1 + x_2$ ($x_1 \in \text{Ker}(f)^\perp$, $x_2 \in \text{Ker}(f)$) と分解したとき、 $x + \text{Ker}(f) = x_1 + \text{Ker}(f) \rightarrow x_1$ で定義される $\mathbf{R}^n / \text{Ker}(f)$ から $\text{Ker}(f)^\perp$ への線形写像は全単射である。これによって $\mathbf{R}^n / \text{Ker}(f)$ と $\text{Ker}(f)^\perp$ を同一視し、 $\bar{j}: \text{Ker}(f)^\perp \rightarrow \mathbf{R}^n$ を標準単射とすると、 $\bar{j} \cdot \bar{f}^{-1} \cdot p$ は \mathbf{R}^m から \mathbf{R}^n への線形写像になる。これを f の擬逆写像 (pseudoinverse) または一般逆写像 (generalized inverse) といい、 f^+ で表す。

$$(3. 2) \quad f^+ : \mathbf{R}^m \xrightarrow{p} f(\mathbf{R}^n) \xrightarrow{\bar{f}^{-1}} \text{Ker}(f)^\perp \xrightarrow{\bar{j}} \mathbf{R}^n$$

である。 f がもともと正則であるときは、 $f^+ = f^{-1}$ 、すなわち擬逆写像と逆写像は一致する。

以下で擬逆写像 f^+ の \mathbf{R}^m と \mathbf{R}^n に自然基底を採ったときの表現行列を求めておく。 f の自然基底による表現行列 A が、 m 次直交行列 U と n 次直交行列 V および (2. 1) の $m \times n$ 対角行列 D によって (2. 3) のように特異値分解されるならば、 r を $\sigma_r > \sigma_{r+1} = 0$ であるような整数とし、 $u_1, \dots, u_m, v_1, \dots, v_n$ を U, V の列ベクトルとすると、各 $1 \leq k \leq n$ に対し $v_k = (0, \dots, 0, \overset{k}{1}, 0, \dots, 0)$ であるから $f(v_k) = \sigma_k u_k$ である。したがって、 v_1, \dots, v_r が生成する部分空間を $[v_1, \dots, v_r]$ 、 v_{r+1}, \dots, v_n が生成する部分空間を $[v_{r+1}, \dots, v_n]$ と書くと

$$(3. 3) \quad \text{Ker}(f)^\perp = [v_1, \dots, v_r], \quad \text{Ker}(f) = [v_{r+1}, \dots, v_n]$$

であり

$$\langle v_k, f^+(u_i) \rangle = \begin{cases} \langle v_k, \sigma_i^{-1} u_i \rangle = \sigma_i^{-1} \delta_{ik}, & 1 \leq i \leq r, \\ 0, & r+1 \leq i \leq m, \end{cases} \quad 1 \leq k \leq n$$

となるから、 $u_1, \dots, u_m, v_1, \dots, v_n$ を $\mathbf{R}^m, \mathbf{R}^n$ の基底にとった場合の f^+ の表現行列は $n \times m$ 対角行列

$$\begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{n+m}, \quad \Sigma^{-1} = \text{diag}(\sigma_1^{-1}, \dots, \sigma_r^{-1})$$

である。これから f^+ の自然基底による表現行列を A^+ と記すことにすると

$$(3.4) \quad A^+ = V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} U$$

となることがわかる。この行列 A^+ を A の擬逆行列という。もちろん、 A が正則のときは、擬逆行列と逆行列は一致する。

§ 4 行列の列ベクトルが生成する部分空間への正射影

この § でも $m > n$ とする。

\mathbf{R}^n 上の線形変換 f は、 $f^2 = f$ であるとき射影と呼ばれ、さらに ' $f = f$ ならば正射影 (または直交射影) と呼ばれる。

射影については、以下が基本的である。

(1) 射影の固有値は 1 または 0 である。

(2) f が射影のとき $I - f$ も射影で、 \mathbf{R}^n は $\text{Im}(f)$ ¹²⁾ と $\text{Im}(I - f)$ の直和である。とくに、 f が正射影ならば、 $\text{Im}(f)$ と $\text{Im}(I - f)$ は直交する、すなわちたがいに直交補空間になる。

(1) から、射影の階数 (rank) はトレースに等しい、ことがわかり、この事実は統計学においてカイ二乗分布あるいはその派生分布の自由度を求めるのに利用される。

$u \in \mathbf{R}^n$ が単位ベクトルであるとき、 $u'u$ は u が生成する \mathbf{R}^n の部分空間への正射影を表す m 次正方行列である。つぎに、 u_1, \dots, u_r ($r < m$) をたがいに直交する単位ベクトルとすると、これらを列ベクトルとする行列を U と書けば $U'U$ は u_1, \dots, u_r が生成する \mathbf{R}^n の部分空間への正射影を表す m 次正方行列である。つぎの定理は、以上の事実を一般化したものである。

12) $\text{Im}(f) = f(\mathbf{R}^n)$ 。

定理 4.1 X を $m \times n$ 行列 ($n < m$) とすると, $X'(XX)^+X$ は X の列ベクトルが生成する R^m の部分空間への正射影を表す m 次正方行列である。

証明 X の特異値分解が $UD'V$ で D の対角要素は $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$ であるとする。各 $1 \leq i \leq n$ に対し, $e_i = (0, \dots, 0, \overset{i}{1}, 0, \dots, 0) \in R^n$ とおくと $x_i = Xe_i$ は X の第 i 列ベクトルであり, 一方 $D'Ve_i$ は第 $1 \sim r$ 番目以外の要素は 0 である n 次元ベクトルであるから, $UD'Ve_i$ は U の第 $1 \sim r$ 列ベクトル u_1, \dots, u_r の線形結合である。したがって, u_1, \dots, u_r が生成する R^m の部分空間を S とすると $x_i \in S$ であり, 行列 D の階数は r で U, V は直交行列であるから X の階数も r で, S は X の列ベクトルが生成する部分空間と一致することがわかる。 $X = UD'V$ により

$$X'(XX)^+X = U \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} U' \quad (I_r \text{ は } r \text{ 次単位行列})$$

が得られるから, u_1, \dots, u_r を列ベクトルとする行列を \bar{U} とすると,

$$X'(XX)^+X = \bar{U}'\bar{U}$$

であり定理が証明された。□

系 $\varepsilon_1, \dots, \varepsilon_m$ をたがいに独立な平均 0 分散 1 の正規分布にしたがう確率変数とするとき, 行列 X が最大階数 (full rank) をもつ, すなわち $\text{rank } X = n$ ならば, $X'(XX)^{-1}X\varepsilon$, $(I - X'(XX)^{-1}X)\varepsilon$ は, 密度関数 $(2\pi)^{-n/2} \prod_{i=1}^n e^{-x_i^2/2}$ および $(2\pi)^{-(m-n)/2} \prod_{i=1}^{m-n} e^{-x_i^2/2}$ をもつ n 次元および $m-n$ 次元正規分布にしたがう確率変数である。ただし, $\varepsilon = (\varepsilon_1, \dots, \varepsilon_m)$ とする。

§ 5 最小二乗解のアルゴリズム

最小二乗解のアルゴリズムは, 固有値を求めるそれから派生したものである。固有値の数値解法では, 対角化可能な正方行列 A に対して適当な方法で直交相似変換を施して対角行列を求めることにより行われる。すなわち, 適当な直交行列 U を用いて

$$UAU = \text{diag}(\lambda_1, \dots, \lambda_n)$$

を求める。これに対して、最小二乗解のアルゴリズムにおける特異値分解は、

- (1) すべての行列に適用することができ、
- (2) 直交行列を用いる点は変わらないが、 ${}^tUAV=D$ (式 (2. 3) 参照) に見られるように相似変換である必要はない。

したがって、最小二乗解のアルゴリズムにおいては自由度がいくぶん大きくなり工夫を折り込む余地があるわけである。

この § では、 $m > n$ 、 X は $m \times n$ 行列でその列ベクトルを $x_1, \dots, x_n \in \mathbf{R}^m$ とし、 $y \in \mathbf{R}^m$ とする。 X の特異値分解 (定理 2. 1 の系参照) は

$$(5. 1) \quad X = U \operatorname{diag}(\sigma_1, \dots, \sigma_n) {}^tV$$

で、この場合 V は $v_1, \dots, v_n \in \mathbf{R}^n$ を列ベクトルとする n 次直交行列であるが U はたがい直交する単位ベクトル $u_1, \dots, u_n \in \mathbf{R}^m$ を列ベクトルにもつ $m \times n$ 行列とする。また、 n 次対角行列の対角要素は $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$ であるとする。

5. 1. 最小二乗解

x_1, \dots, x_n が生成する \mathbf{R}^m の部分空間を S としその直交補空間を S^\perp とすると、不等式 (1. 3) からわかるように $y - Xb \in S^\perp$ を満たす $\beta = b \in \mathbf{R}^n$ があればそれは $\|y - Xb\|$ を最小にする。定理 4. 1 によれば、 $X({}^tXX)^+X$ は S への正射影であるから $I - X({}^tXX)^+X$ は S^\perp への正射影である。しかるに、

$$(5. 2) \quad b = ({}^tXX)^+Xy = v_1 \frac{\langle u_1, y \rangle_m}{\sigma_1} + \dots + v_r \frac{\langle u_r, y \rangle_m}{\sigma_r}$$

であると $y - Xb = (I - X({}^tXX)^+X)y \in S^\perp$ となり、 b は最小二乗解であることがわかる。条件 $y - X\beta \in S^\perp$ と同値な正規方程式

$${}^tXX\beta = {}^tXy$$

から明らかに、 tXX が正則ならば $({}^tXX)^+ = ({}^tXX)^{-1}$ となって $b = ({}^tXX)^{-1}$

' Xy が正規方程式の唯一の解であるが、' XX が正則でないときは

$$(5.3) \quad Xc = u_1\sigma_1 < v_1, c >_n + \cdots + u_r\sigma_r < v_r, c >_n = 0$$

となるような任意のベクトル $c \in \mathbf{R}^n$ について $b' = b + c$ も正規方程式の解である。ベクトル c に関する条件 (5.3) は c が v_1, \dots, v_r と直交すること、すなわち v_{r+1}, \dots, v_n の線形結合であることと同値である¹³⁾。 b は、(5.2) から v_1, \dots, v_r の線形結合であるから、最小二乗解の中で長さが最小であることがわかる。

b の数値解は、(5.1) が示すように X の特異値分解を求める、すなわち行列 U, V と X の特異値を求めることによって得られるが、 X の特異値分解は 5.3 で詳説する QR 法を X に適用することによって得られる。ただし、 QR 法を X に直接適用するのではなく、はじめに Householder 変換によって X を 2 重対角行列に変換しておくが、この意義はつぎの 2 点にある。

(1) X に直接 QR 法を適用したのでは、計算時間が膨大になるおそれがあり QR 法が実用性を欠く結果になる。

(2) QR 法は、本来正方形行列の固有値を求めるアルゴリズムであり、行列 ' XX に対して QR 法のステップを繰り返すと行列 V の列ベクトルが ' XX の固有ベクトルに収束する筈のものである。しかるに、行列 ' XX に対してではなく、行列 X に対して QR 法のステップ適用してもなお行列 V の列ベクトルに求める固有ベクトルが得られることを保証するのは、後述の Q インプリシット定理に基づく原理である。ただし、この原理は Hessenberg 行列¹⁴⁾ にだけ適用が可能である。対称な Hessenberg 行列は 3 重対角行列であり、とくに、 X が 2 重対角行列ならば ' XX は 3 重対角行列で

13) 行列 X が表す線形写像を f とすると、(3.3) から (5.3) の条件は $c \in \text{Ker}(f)$ と同値である。

14) $i > k+1$ のとき $a_{ik} = 0$ であるような正方形行列 (a_{ik}) を (上方) **Hessenberg** 行列という。たとえば、 $\begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}$ は 4×4 Hessenberg 行列である。

ある。

5. 2 Householder 変換による行列の 2 重対角化

\mathbf{R}^p のベクトル $(1, 0, \dots, 0)$ を e_1 と記すことにする。Householder 変換はベクトル $x \in \mathbf{R}^p$ を e_1 方向のベクトルへうつす直交変換で、これを繰り返して適用することにより X を 2 重対角化する。

部分空間 S に関する対称移動の変換 ϕ_S : この変換は、 p_S を S への正射影とすると、各 $y \in \mathbf{R}^p$ について $\frac{\phi_S(y) + y}{2} = p_S(y)$ で定義される。すなわち、

$$(5. 4) \quad \phi_S = 2p_S - I$$

である。また、 p_S は正射影であるから $p_S = p_S$, $p_S^2 = p_S$ であり、したがって $\phi_S = \phi_S$ (対称変換) で、かつ $\phi_S^2 = 4p_S^2 - 4p_S + I = I$ (べき等変換) である。したがって、 ϕ_S は直交変換である。

Householder 変換 φ_x の定義 : つぎにこの ϕ_S を用いてベクトル $x \in \mathbf{R}^p$ を e_1 方向のベクトルへうつす Householder 変換 φ_x を定義する。 φ_x が直交変換であることへの要請から、

$$\varphi_x(x) = \pm \|x\| e_1$$

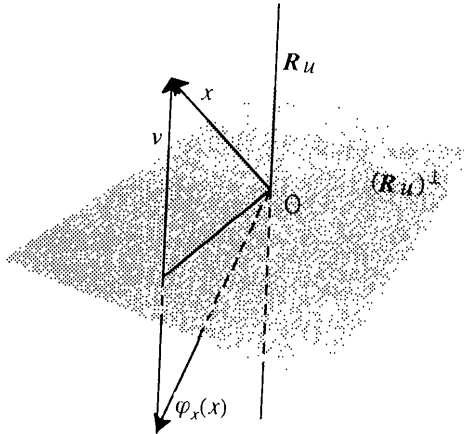
でなければならない。つぎに、 u を $x - \varphi_x(x) = x - \varepsilon \|x\| e_1$ 方向の単位ベクトル (ただし $\varepsilon = \pm 1$)、すなわち $v = x - \varepsilon \|x\| e_1$, $u = v / \|v\|$ とすると、 u が生成する部分空間 $\mathbf{R}u$ の直交補空間 $(\mathbf{R}u)^\perp$ は u を法線ベクトルとする \mathbf{R}^p の超平面であり; φ_x としては部分空間 $(\mathbf{R}u)^\perp$ に関する対称移動の変換を採ればよい。 $\mathbf{R}u$ への正射影は $u'u$ であることから、 $(\mathbf{R}u)^\perp$ への正射影は $I - u'u$ である。そこで実際 (5. 4) において $p_S = I - u'u$ として計算してみると $\phi_S(x) = \varepsilon \|x\| e_1$ が得られる。結局、

$$(5. 5) \quad \varphi_x = I - 2u'u$$

であることがわかる。なお、 ε は計算途上の桁落ちを防ぐために、その符

号がベクトル x の第 1 要素の符号と反対であるように定める。また、こうすることにより、 $x \neq 0$ ならば、 $v \neq 0$ であることが保証される。

(5. 5) の φ_x を x で定まる p 次元 **Householder** 変換という。上で見たように、Householder 変換は部分空間に関する対称移動の変換であるから、直交変換であり、同時に対称性とべき等性をもっている。



行列 X の 2 重対角化：以下では、 p 次正方形行列 P から $\begin{pmatrix} I_{m-p} & 0 \\ 0 & P \end{pmatrix}$ のような m 次正方形行列を作ることが多いが、これを $I_{m-p} \oplus P$ と記すことにする。 P から同じようにつくる n 次正方形行列は $I_{n-p} \oplus P$ と記す。ただし、 $I_0 \oplus P = P$ であるとする。 $m \times n$ 行列 X を

$$X = \begin{pmatrix} Y \\ Z \end{pmatrix}_{\substack{m-p \\ p}} = (Y' \ Z')_{\substack{n-p \\ p}}$$

と分割すると、

$$(5. 6) \quad \begin{pmatrix} I_{m-p} & 0 \\ 0 & P \end{pmatrix} \begin{pmatrix} Y \\ Z \end{pmatrix} = \begin{pmatrix} Y \\ PZ \end{pmatrix}, \quad (Y' \ Z') \begin{pmatrix} I_{n-p} & 0 \\ 0 & P \end{pmatrix} = (Y' \ Z'P)$$

であり、 X に左から $I_{m-p} \oplus P$ を乗じた場合第 $1 \sim m-p$ 行は不変、右か

ら $I_{n-p} \oplus P$ を乗じた場合第 1~ $n-p$ 列は不変であることに注意する。

Householder 変換により行列 X を (上) 2 重対角化するには、まず、以下の様に、 m 次元直交行列 U_1, \dots, U_n と n 次元直交行列 V_1, \dots, V_{n-2} をつくる。

U_1 : X の第 1 列ベクトルが定める m 次元 Householder 変換行列を U_1 とする。

V_1 : $U_1 X$ の第 1 行ベクトルの 2 番目以降の要素が定める $n-1$ 次元 Householder 変換行列を H_1 として $V_1 = I_1 \oplus H_1$ とする。

$U_k (2 \leq k \leq n-1)$: $X_{k-1} = U_{k-1} X_{k-2} V_{k-1}$ (ただし、 $X_0 = X$) とし、 X_{k-1} の第 k 列ベクトルの k 番目以降の要素が定める $m-k+1$ 次元 Householder 変換行列を G_k として $U_k = I_{k-1} \oplus G_k$ とする。

$V_k (2 \leq k \leq n-2)$: $U_k X_{k-1}$ の第 k 行ベクトルの $k+1$ 番目以降の要素が定める $n-k$ 次元 Householder 変換行列を H_k として $V_k = I_k \oplus H_k$ とする。

U_n : $U_{n-1} X_{n-2}$ の第 n 行ベクトルの n 番目以降の要素が定める $m-n+1$ 次元 Householder 変換行列を G_n として $U_n = I_{n-1} \oplus G_n$ とする。

この結果、 $U_n U_{n-1} \dots U_1 X V_1 \dots V_{n-2}$ が 2 重対角行列となる。下図は X が 6×4 行列の場合である。

$$\begin{array}{c}
 \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{pmatrix} \xrightarrow{U_1} \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{pmatrix} \xrightarrow{V_1} \begin{pmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{pmatrix} \xrightarrow{U_2} \begin{pmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix} \\
 X \qquad \qquad \qquad X_1 \\
 \xrightarrow{V_2} \begin{pmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix} \xrightarrow{U_3} \begin{pmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \end{pmatrix} \xrightarrow{U_4} \begin{pmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
 X_2
 \end{array}$$

プログラムについての注釈：C言語による，2重対角化の関数 `bidiag` を定義するプログラムの先頭行は

```
void bidiag(double *a, double* v, double* d, double* e, int m, int n)
```

となる。a は $m \times n$ 行列のポインタで，関数 `bidiag` はこの行列を2重対角化する。関数 `bidiag` は2重対角化された行列の対角要素を $d[0] \sim d[n-1]$ に，また副対角要素を $e[0] \sim e[n-2]$ に与え，さらに，ポインタ a が指定する行列の記憶場所には，変換行列 $U_1 \cdots U_n$ を，ポインタ v が指定する記憶場所には n 次正方行列である変換行列 $V_1 \cdots V_{n-1}$ を戻す。したがって，もとの行列は保存されない。

冒頭で，ポインタ v が指定する記憶場所を， n 次単位行列化しておく。

関数 `bidiag` のメインループは，以下の4個の関数を使用している。

```
double colu(double* a, int m, int n, int i);
```

```
void chouse(double* a, int m, int n, int i, int j);
```

```
double rowu(double* a, int n, int i);
```

```
void rhouse(double* a, double* u, int n, int i, int j);
```

関数 `colu` は U_k を定義する Householder 変換 $\varphi_x = I - 2u'u$ の単位ベクトル u を計算する。関数値（関数が返す値）は，求める2重対角行列の第 k 対角要素となる。ポインタ a は，2重対角化しようとする行列の記憶場所を指示し，Householder 変換 φ_x を定義するベクトル x として，列ベクトル $a[n*i+i] \sim a[n*(m-1)+i]$ を採り，計算されたベクトル u は同じ $a[n*i+i] \sim a[n*(m-1)+i]$ に置く。

関数 `chouse` は，関数 `colu` が計算した $a[n*i+i] \sim a[n*(m-1)+i]$ にある単位ベクトル u を用いて，Householder 変換 $\varphi_x = I - 2u'u$ を施す。

実際の変換は，列ベクトル $a[n*i+j] \sim a[n*(m-1)+j]$ に対して施す。

関数 `rowu` は V_k を定義する Householder 変換の単位ベクトル u を計算する。関数値（関数が返す値）は，求める2重対角行列の第 k 副対角要素となる。ポインタ a は，2重対角化しようとする行列の記憶場所

線形最小二乗法の諸問題 (I)

を指示し, Householder 変換 φ_x を定義するベクトル x として, 行ベクトル $a[n*i+i+1] \sim a[n*i+n-1]$ を採り, 計算されたベクトル u は同じ $a[n*i+i+1] \sim a[n*i+n-1]$ に置く。

関数 `rhouse` は, 関数 `rowu` が計算した $a[n*i+i+1] \sim a[n*i+n-1]$ にある単位ベクトル u を用いて, Householder 変換 $\varphi_x = I - 2u'u$ を施す。

実際の変換は, 行ベクトル $a[n*j+i+1] \sim a[n*j+n-1]$ に対して施す。

関数 `bidiag` のメインループは次のようになる。

```

for(i=0;i<n;i++){
    /* column */
    d[i]=colu(a, m, n, i);
    if(d[i])
        for(j=i+1;j<n;j++)
            chouse(a, m, n, i, j);
    /* row */
    e[i]=rowu(a, n, i);
    if(e[i])
        for(j=i+1;j<m;j++)
            rhouse(a, a+n*i, n, i, j);
    /* accumulation of V */
    if(e[i])
        for(j=0;j<n;j++)
            rhouse(v, a+n*i, n, i, j);
}

```

関数 `chouse` は, 本来 $j=0 \sim n-1$ について実行すべきであるが ((5. 6) 参照), $j=0 \sim i-1$ の変換を施すべき行要素はすでにゼロに変換にされており, 実際にはその場所に Householder 変換の単位ベクトル u が存在している。また, $j=i$ についても $\varepsilon \|x\| e_i$ となることが分かっている。

線形最小二乗法の諸問題 (1)

るから, $j=i+1 \sim n-1$ についてだけ実行している。

関数 rhouse の実行についても, ほぼ同様である。

行列 $U=U_1 \cdots U_n$ は, メインループ終了後もとの行列の下三角部分に残しておいた Householder 変換の単位ベクトル u を用い, もとの行列の場所に生成する。この部分のプログラムは次のようになる。

```

/* accumulation of U */
for(i=n-1;i>=0;i--){
    /* process j-th col. */
    for(j=i+1;j<n;j++){
        a[n*i+j]=ip=0.0;
        for(k=i+1;k<m;k++) /* ip=<j-th col., u */
            ip+=a[n*k+j]*a[n*k+i];
        ip*=-2.0;
        for(k=i;k<m;k++)
            a[n*k+j]+=ip*a[n*k+i];
    }
    ip=-2.0*a[n*i+i];
    a[n*i+i]=1.0+ip*a[n*i+i];
    for(j=i+1;j<m;j++)
        a[n*j+i]*=ip;
}

```