最小 2 乗法とその応用
– ベクトル列の加速法 –

2000MM109 日置 昌吾
指導教員　鳥居 達生

1　はじめに

最小 2 乗法とベクトル列の加速法について研究しました. 過剰条件の連立方程式の解は一般的に解を持たないので、最小 2 乗法を利用します. そして, 最小 2 乗法の QR 分解の 1 つであるハウスホルダー変換の理論を学びました [1]. ベクトル列の加速に最小 2 乗法を利用します. ベクトル列の加速法は数列の加速法の拡張です [2].

2　最小 2 乗法

未知数 $x_1, x_2, \cdots, x_n$ の数 n が方程式の数 m より小さい連立 1 次方程式 $Ax = b$

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, m > n$$

を過剰条件の連立方程式といいます. この方程式は一般的には解を持ちません. そこで, 左辺 $y = Ax$ と右辺 b との距離の 2 乗 $F(x) \equiv ||b - y||_2^2 = ||b - Ax||_2^2$ が 0 にならないまでも最小となる $x = x^*$ を解として採用することが考えられます. この方法を最小 2 乗法、解 x^* を最小 2 乗解といいます.
方程式で 0 にすべき量：$r = b - Ax$ を残差といいます. 最小 2 乗法は残差を最小にする方法です. 係数行列 A を列ベクトルが n 個あるものとして考えると, $A = (a_1, a_2, \cdots, a_n)$ と表すことができます [1].

3　ハウスホルダー変換

3.1　ハウスホルダー行列

m 次元ベクトル u, $||u|| = \sqrt{2}$ で定義される行列

$$H(u) = I_m - uu^T \qquad (1)$$

をハウスホルダー行列, それによる変換をハウスホルダー変換といいます. これは一般の長方形行列を直交行列と上三角行列の積に分解するために使われます.

3.2　行列の第 1 列消去

$m \times n$ 行列 $A = (a_1, a_2, \cdots, a_n)$ に, 第 1 列ベクトル a_1 を第 1 軸化するハウスホルダー行列 $H = H(u)$ を左から掛けると

$$HA = (Ha_1, Ha_2, \cdots, Ha_n)$$
$$= \begin{pmatrix} d & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{m2} & \cdots & a'_{mn} \end{pmatrix}$$

となり, 第 1 列の第 2 成分以下が消去されます. $d = ||a_1||_2$ です.

3.3　ハウスホルダー QR 分解法

直交変換 Q はベクトルの長さを変えないので, $||Qb - QAx||_2^2 = ||Q(b - Ax)||_2^2 = ||b - Ax||_2^2$ です. $QAx - Qb$ も $b - Ax$ と同じ残差となります. ハウスホルダー QR 分解法は, ハウスホルダー変換 $H^{(1)}, H^{(2)}, \cdots, H^{(n)}$ により $(A^{(1)}, b^{(1)}) = H^{(1)}(A, b)$ を

$$(A^{(k+1)}, b^{(k+1)}) = (H^{(k)}A^{(k)}, H^{(k)}b^{(k)}) \qquad (k = 1, 2, \cdots, n) \qquad (2)$$

と逐次直交変換し, 最終的に係数行列を上三角行列

$$A^{(n+1)} = R = \begin{pmatrix} R_1 \\ O \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{nn} \\ & O & & \end{pmatrix}$$

にします. 中間結果 $A^{(k)}$ は第 $k-1$ 列まで上三角化された形で,

$$A^{(k)} = \begin{pmatrix} R^{(k)} & V^{(k)} \\ O & W^{(k)} \end{pmatrix}$$

と書けます. ここで、$R^{(k)}$ は $k-1$ 次上三角行列です. 変換

$$H^{(k)} = \begin{pmatrix} I_{k-1} & O \\ O & H(u^{(k)})W^{(k)} \end{pmatrix}$$

のハウスホルダー変換 $H(u^{(k)})$ は $W^{(k)}$ を第 1 列を消去するようにとるので,

$$A^{(k+1)} = H^{(k)}A^{(k)} = \begin{pmatrix} R^{(k)} & V^{(k)} \\ O & H(u^{(k)})W^{(k)} \end{pmatrix}$$

は第 k 列まで上三角化された行列になります.
ベクトル $b^{(n+1)}$ を

$$b^{(n+1)} = \begin{pmatrix} b_1^{(n+1)} \\ b_2^{(n+1)} \end{pmatrix}, b_1^{(n+1)} \in R^n, b_2^{(n+1)} \in R^{(m-n)}$$

とすると, 最終的な方程式 $Rx = b^{(n+1)}$ の正規方程式は
$R_1^T R_1 x = R^T Rx = R^T b^{(n+1)} = R_1^T b_1^{(n+1)}$
です. この両辺から R_1^{-T} を掛けた

$$R_1 x = b^{(n+1)} \qquad (3)$$

を後退代入法で解いて最小 2 乗解 $x = x^*$ が求まります.
ハウスホルダー変換の列と上三角行列は係数行列 A のみによって決まり, 右辺ベクトル b とは独立しています.
アルゴリズムを 2 段階に分けます.
1) 係数行列 A から $H^{(k)}{}_{k=1}^{n}$ と R を求め, 保存します.
2) 右辺ベクトル b から式 (2) より b_{n+1} を求め, 方程式

(3) を解きます.
このように分ければ, 右辺ベクトルが複数個ある場合でも ${H^{(k)}}_{k=1}^{n}$ と R は共通に使えます.
行列 $U = H^{(n)}H^{(n-1)}\cdots H(1)$ は直交行列 (直交行列の積は直交行列) で, 式 (2) より $R = A^{(n+1)} = UA$ となるので,

$$A = QR, Q = U^T = U^{(-1)} \tag{4}$$

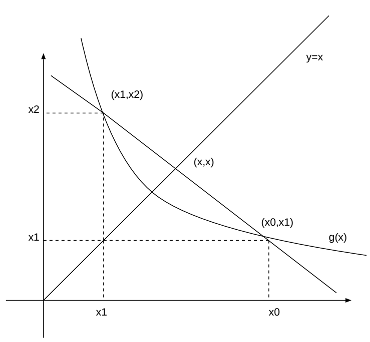
このような変換を行列の QR 分解といいます [1].

4 ベクトル列の加速法

反復法で収束する点列またはベクトル列が分かっているとき, その結果に手を加えることにより, 収束を速めることができます. その操作を収束の加速と呼んでいます.

4.1 数列の加速

数列は, あらかじめ与えられるかあるいは何らかの方法で生成されるとします. $x_0, x_1, x_2, \cdots$ または x_0 を与えて $x_{k+1} = g(x_k)$(x_k を与えると新しい x_{k+1} が生成される) を考えます. xy 平面上に 2 点 $(x_0, x_1), (x_1, x_2)$ をとり, 直線で結びます. ($g(x)$ の割線近似)



これと直線 $y = x$ の交点を $g(x)$ の不動点の近似として考えます.
$\Delta x_k = x_{k+1} - x_k, \Delta^2 x_k = \Delta x_{k+1} - \Delta x_k$ とすると,

$$x = x_0 - \frac{\Delta x_0^2}{\Delta^2 x_0} \tag{5}$$

となります.
x_1 の補正として変形すれば,

$$x = x_1 - \frac{\Delta x_0 \Delta x_1}{\Delta^2 x_0} \tag{6}$$

となります. この式をエイトケン ($Aitken$) の Δ^2 加速法といいます.

4.2 ベクトル列の加速

ベクトル列 x_k は, あるアフィン変換に従うとします. x_0 を初期値として $x_i = Bx_{i-1} + c$, (B は行列, c はベクトル) によってベクトル列 x_i が生成されます.
$\Delta x_i = B\Delta x_{i-1}\ i = 1, 2, \cdots, k$
$x = Bx + c$ の右辺の不動点を
$x = x_k + \sum_{i=1}^{k} \alpha_i \Delta x_{i-1}$ と模型を仮定します.
残差を $r = (Bx + c) - x$ とし, $||r||$ を $\alpha_1, \alpha_2, \cdots, \alpha_k$ の関数と考え, $||r||$ を最小にするように $\alpha_1, \alpha_2, \cdots, \alpha_k$ を求めます.

$$r = B(x_k + \sum_{i=1}^{k} \alpha_i \Delta x_{i-1}) + c - (x_k + \sum_{i=1}^{k} \alpha_i \Delta x_{i-1})$$
$$= (Bx_k + c) + \sum_{i=1}^{k} \alpha_i B \Delta x_{i-1} - x_k - \sum_{i=1}^{k} \alpha_i \Delta x_{i-1}$$
$$= x_{k+1} - x_k + \sum_{i=1}^{k} \alpha_i (\Delta x_i - \Delta x_{i-1})$$
$$= \Delta x_k + \sum_{i=1}^{k} \alpha_i \Delta^2 x_{i-1}$$

$(\Delta^2 x_{i-1} = \Delta x_i - \Delta x_{i-1})$
$a = (\alpha_1, \alpha_2, \cdots, \alpha_k)^t$, とおけば, a は過剰条件連立 1 次方程式

$$\Delta^2 X a = -\Delta x_k \tag{7}$$

の最小 2 乗解です.
$\Delta X = (\Delta x_0, \Delta x_1, \cdots, \Delta x_{k-1})$ とおけば,

$$x = x_k + \Delta X a \tag{8}$$

これが新しい近似解です. x を x_0 として収束するまで反復します. その意味で最小 2 乗法の応用になります [2].

4.3 ベクトル列の加速法のプログラム

ベクトルを作るのに反復法の 1 つであるヤコビ法を使います. 次に ΔX と $\Delta^2 X$ を作ります. ΔX と $\Delta^2 X$ が求まったら, 係数行列を $\Delta^2 X$, 右辺ベクトルを ΔX としてハウスホルダー変換を使い最小 2 乗解を求めます. x に関する模型より x を求めます. それが収束していなければ, そのベクトルを初期値として用いヤコビ法から同じ計算を繰り返します [3].

5 おわりに

ベクトル列の加速は正解が分かっている問題でチェックしました. ハウスホルダー変換もベクトル列の加速のプログラムも何度もアルゴリズムをチェックし改良を重ねました. 加速の効果は問題に依存するので, 今後さらに研究する必要があります.

参考文献

[1] 鳥居達生：数列, ベクトル列の加速 (拡張 $Aitken$ 法) 研究会資料, (2003.8.28).
[2] 杉浦 洋 著: 数値計算の基礎と応用, サイエンス社 (1997).
[3] 岡田 稔: C によるプログラミング演習 (1993.11)