

数値計算のスクエア マトリクスの計算と応用

M.Shimura
JCD02773@nifty.ne.jp

2008 年 6 月 27 日
Ver 0.97 (未定稿)

目次

第 1 章	マトリクスの算法	7
1.1	鶴亀算	7
1.2	行列式	10
1.3	ガウスの消去法	12
1.4	行列あれこれ/正方形の行列	16
1.5	グラム・シュミットの直交化法	21
1.6	行列の内積	24
1.7	ルベリエ・ファデーエバ法	26
1.8	LF 法 3×3	29
1.9	ケーリー・ハミルトンの定理	31
1.10	固有ベクトル	34
1.11	標準化	37
1.12	分散共分散行列	40
第 2 章	マトリクスの分解と相似変換	43
2.1	LU 分解・LDU 分解	43
2.2	コレスキー分解	45
2.3	QR 分解	47
2.4	SVD 変換 (特異値分解)	49
2.5	マトリクスの相似変換	51
2.6	ジョルダン標準型	53
第 3 章	線形写像	59
3.1	写像	59
3.2	スペクトル分解	61
3.3	2 次形式	65
3.4	アフィン変換	66
第 4 章	マトリクスと回帰分析	69
4.1	縦長/横長の行列	69
4.2	キングの法則/多項式	74
4.3	時系列回帰	78
4.4	線形同時方程式	84

第 5 章	推移確率	95
5.1	Markov chain/マルコフの推移確率	95
5.2	人口問題とレスリー行列	101
第 6 章	マトリクスの活用	111
6.1	主成分分析	111
6.2	The input output model	113
6.3	マトリクスとグラフ理論	116
6.4	シンプレクス法	120
6.5	ゲーム理論	125
6.6	コンパニオン・マトリクス	128
6.7	混合配列とボックス	129
6.8	ネステッド・アレー	132
6.9	2 元分散分析	134

はじめに

印象派の時代の画家ロートレックは、多くの客を度々自宅に招いて自分で料理をして振る舞った。南フランスのその料理ノートは友人ジョワイアンにより整理され、出版されて残されている。手に取ると各ページには彼の絵のように、19世紀末の香りが残されている。

行列の特性多項式をルベリエ・ファディーエバ法で求め、艶やかな蝶の姿を眺めながら行列のもつ特性や諸機能を料理を楽しむように体験できるマトリクスレシピをしたためてみた。数多くの素材から何百かの最新のマイクロエンジンを作り、必要な部分を組み合わせ盛つけていく。数学的証明は少なくし、実際にサンプルで理論を動かし、理論の検証を経験できるようにした。

表記法

単位行列 単位行列は E と表す場合も多い。 E は *Einheitsmatrix* のヘッド。数字の 1 に似た I も用いられる。本稿は I で統一した。

行列除算 行列除算はマセマティシャンは XA^{-1} と表す。
 $\frac{X}{A}$ に比べて数値計算では確実に 1 ステップ余計にかかる。本稿では感覚的にわかりやすい $\frac{X}{A}$ も用いる。ちなみに、行列の乗算では $AB \neq BA$ である。

転置 $'A, A', A^T$ 等の書法がある。 A' を用いる。

分数 J には計算精度を上げるため π, e の数値定数化 1p1.1x1 や分数演算をサポートしている。分数は 1r3 は $\frac{1}{3}$ である。この見たままの分数表記を用いている箇所がある。

数値計算について

今や PC でもメモリ 8GB、マルチ CPU、高速クロックが手軽に手に入る時代である。学習用にできるだけシンプルで分かりやすいアルゴリズムを提示し、メモリや計算量は考慮していない。複雑なスクリプトの作成に費やす時間やコストの方が多大だからである。大規模計算の確立された手法は既に行き渡っており、本稿はハイグレーダー用ではない。

第 1 章

マトリクスの算法

1.1 鶴亀算

鶴亀算は中国で 1700 年前の六朝時代の**孫子算經**に雉と兎として表されている。我が国では江戸時代の**因起算歌（1640）**が最初であり、後になってめでたい鶴と亀が用いられた。

一次の連立方程式はギリシャでもインドでもなく、中国が起源である。

鶴と亀が合わせて 10 匹。足の数は 合わせて 34 本で ある。	$\begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 10 \\ 34 \end{pmatrix}$	$\text{JD110} = \begin{pmatrix} 1 & 1 & 10 \\ 2 & 4 & 34 \end{pmatrix}, \text{ : } \begin{pmatrix} 2 & 4 & 34 \\ 1 & 1 & 10 \end{pmatrix}$	$\text{cr D110} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \end{pmatrix}$
50 円切手と 80 円 切手を合わせて 14 枚買って 1000 円支払った。	$\begin{pmatrix} 1 & 1 \\ 50 & 80 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 14 \\ 1000 \end{pmatrix}$	$\text{JD111} = \begin{pmatrix} 1 & 1 & 14 \\ 50 & 80 & 1000 \end{pmatrix}, \text{ : } \begin{pmatrix} 50 & 80 & 1000 \\ 1 & 1 & 14 \end{pmatrix}$	$\text{cr D111} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 10 \end{pmatrix}$

1.1.1 連立方程式と線形代数

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

システム方程式をベクトルで表す

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

システム方程式をマトリクスで表す

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$Ax = b$$

$$x = \frac{b}{A} \quad (\text{または } x = A^{-1}b)$$

線形代数で表すとすっきりするが逆行列が必要である。

拡大係数行列

$\begin{bmatrix} 1 & 1 & 10 \\ 2 & 4 & 34 \end{bmatrix}$ のように $Ax = b$ の係数行列と定数ベクトルを並べてできる行列 (A, b) を拡大係数行列という。

1.1.2 クラメール法

Gabriel Cramer(1704 – 1752) Genova 生まれのスイス人数学者。

ガウスよりも早くコンピュータ時代の優雅な線形システムの解法が生まれていた。

最初にクラメールによる線形システムの解法を用いる。

$$\begin{cases} ax + by + cz = j \\ dx + ey + fz = k \\ gx + hy + iz = l \end{cases}$$

システム方程式をマトリクスで表す

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} j \\ k \\ l \end{bmatrix}$$

クラメールの解法

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{\begin{bmatrix} a & b & c & j \\ d & e & f & k \\ g & h & i & l \end{bmatrix}}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}$$

$$\begin{array}{ccc|cc} 1 & 1 & 10 & & \\ 2 & 4 & 34 & & \\ \hline & 1 & 1 & & \end{array} \Rightarrow \begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 7 \end{array}$$

J では俳句の発句より短く表現でき、次のような行列による除算を行う。次のようにタイプする。

```
cr=: %."1
```

巧く解けている時は左に単位行列が現れる。

```
cr D110
1 0 3
0 1 7
```


クラメル法は x, y, x を行列式の値を個別に求めて計算する方法が紹介されていることが多いが、このように逆行列を用いると行列演算で一度に計算できる。

*1

1.1.3 Grammar

リスト/ベクトル Jは数をまとめて入力できる。行単位リスト (数学のベクトル) で、スペースで区切る。一個づつの数はスペースを空ければよく、カンマで区切る必要はない。

定義 名詞の定義 (変数へ代入) は J は $=$ (local) と $:=$ (global) がある。多くの言語では変数への代入と言い、 $=, := <-$ などとも用いられる。J では $=$ は等しいか否かの判定に用いる。

層連結 *raminate* $,:$ 積層のマトリクスを作るには最後の行を $,:$ で連結する。

1 1 1, 1 1 10 $,:$ 2 4 34

matrix-divide $\%$. 逆行列を求める J の強力な原始関数 (primitive)。左右に引数を取ると即座に行列除算を行う。

列を落とす $\}$:"1 最終の列を落とす。

関数型定義 クラメル法を $cr0:= \%$. $\}$:"1 と定義している。これはなるべくパラメータを明示しないで関数記号のみを組み合わせるという J の新しい関数定義機能 (*tacit definition*) を使った。従来の計算順序は必ず左から右へ行いパラメータ (y) もはっきりと示す明示型 (*Explicit definition*) で表せば次のようになる。J では 2 つの型は併用できる。

$cr0:= 3 : 'y \% . \}$:"1 y'

行列除算 クラメル法で $\frac{B}{A}$ のマトリクスのサイズが異なる場合は J の *matrix_diviode*($\%$.) は次のように計算する。

$$Ax = B$$

$$x = A^{-1}B, (or \frac{B}{A})$$

$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 10 \\ 2 & 4 & 34 \end{bmatrix} = \begin{bmatrix} 2 & -0.5 \\ -1 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 10 \\ 2 & 4 & 34 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 7 \end{bmatrix}$$

*1 数値計算では 0 に限りなく近い数が現れることも多い。e.g. 9.997e_9 多くのツールでは下限を定めて取り除く関数を組み込んでいる。J にも関数は備わっているが探す間に簡単に自作できる

1.2 行列式

行列式は行列よりずっと早くから研究され、BC3 世紀の中国の算術書にあらわれている。その後の進展は乏しく時代は 16 世紀末に至り、高次方程式の解法との関連で再登場する。カルダノが 2×2 の行列式を考察し、多次元はライプニッツによる。関孝和(1642?-1708) もパイオニアの一人で二つ以上の方程式から未知数を消去するために行列式を研究している (1683)。2008 年は関孝和没後 300 年である。上州藤岡の生まれ。歴代の将軍は改暦のため優秀な数学者を召し抱えた。多くの世界先端の理論を欧州に輸出して知らしめられなかったのは残念である。最初の和暦貞享歴 (1685) を作成し幕府初代天文方に就いた渋川春海 (1635-1715) は囲碁の名門 2 代安井算哲でもあった。

1.2.1 行列式の値の算法

]D120=: 1 1 ,: 2 4

1 1

2 4

-/ . * a NB. determinant

2

サラスの方法

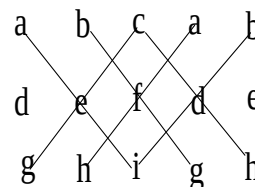
$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

$$= a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

$$= aei - afh + bfg - bdi + cdh - ceg$$

$$= aei + bfg + cdh - afh - bdi - ceg$$



1.2.2 行列式と逆行列

行列 A の逆行列 A^{-1} が存在するための条件は $\det A \neq 0$ で、0 でないことである。

逆行列が存在する行列を正則行列 (non-singular matrix) という。

行列式を用いた 2×2 の逆行列

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 4 & -1 \\ -2 & 1 \end{bmatrix}$ (右の行列は対角行列の上下を入れ替え、 残った要素はマイナスを付ける) 行列式の値が 0 になると解が得られない。	<pre> 1 1 2 4 %. D120 2 _0.5 _1 0.5 </pre> *2
--	---

1.2.3 Grammar

単項・両項 J の関数には右にのみ引数をとる場合 (単項) と両側に引数をとる場合 (両項) がある。^{*3}

determinant J の行列式のプリミティブ (原始関数) は単項の - / . * である

permanent 単項の +/ . * は全てを掛け合わせる。

```

+/ . * D120 NB. permanent
8          NB. 1*4*2*1

```

*2 学校の試験で 2×2 の逆行列を求めること以外に手計算を行うことも珍しくなった。

*3 3, 4 の引数はリストなどで与える

1.3 ガウスの消去法

ガウス Carl Frierich Gauss(1777 – 1853) は天分を愛され、*Duke of Braunschweig* の庇護のもとでゲッティンゲン大学で学ぶことが出来た。このゲッティンゲン大学はハノーバー選帝侯でイギリス国王にもなったジョージ 2 世がオックスフォードやケンブリッジをモデルに作った大学である。この国王の治世に建設された植民地がジョージア州であり、ヘンデルはロンドンで再会した国王に「水上の音楽」やメサイヤを演奏して興をさかせた。

高木貞治は 1900 年にゲッティンゲンでクラインとヒルベルトに学んでいる。

1.3.1 Gauss 法

この消去法はガウスが 1809 年の「天体運動論」で公表した手法で*4前進消去、後退代入でコツコツと解を求める確実な方法で、紙と鉛筆の時代の常用法であり、今でもよく用いられている。

太字はピボット

鶴亀算の例

前進消去

$$\left(\begin{array}{cc|c} \mathbf{1} & \mathbf{1} & 10 \\ 2 & 4 & 34 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} \mathbf{1} & \mathbf{1} & 10 \\ 0 & 2 & 14 \end{array} \right)$$

後退代入

$$2y = 14 \rightarrow y = 7 \rightarrow x + 7 = 10 \rightarrow x = 3$$

Example 3×3

$$\begin{cases} 2x_1 & +3x_2 & -x_3 & = 5 \\ 4x_1 & +4x_2 & -3x_3 & = 3 \\ -2x_1 & +3x_2 & -x_3 & = 1 \end{cases}$$

$$\left(\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 2 & 3 & 1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{array} \right)$$

$$x_3 = 3 \rightarrow -2x_2 = -4 \rightarrow x_2 = 2 \rightarrow 2x_1 = 2 \rightarrow x_1 = 1$$

*4 ラグランジュも同じアイデアを持っていた。中国で紀元前に類似のアルゴリズムの記述があるようだが、ヨーロッパ社会に影響を与えていない。

1.3.2 ピボット

ガウスの消去法系の解法はピボットに注意を払う必要がある。連立方程式は通常は行の順序にはこだわらないので交換しても差し支えない。数値計算では0で割る計算は避けるし、余りにも小さい数で割ると桁落ちにより計算誤差が増大する。ピボットにこのような数が入る場合には行を交換する。また大きい数を上に持ってきた方が計算が安定する。

$$\begin{cases} 2x_1 & +3x_2 & -x_3 & = 5 \\ 4x_1 & +4x_2 & -3x_3 & = 3 \\ -2x_1 & +3x_2 & -x_3 & = 1 \end{cases}$$

$$\left(\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & -3 & 3 \\ 2 & 3 & -1 & 5 \\ -2 & 3 & -1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & -3 & 3 \\ 0 & 2 & 1 & 7 \\ 0 & 10 & -5 & 5 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & -3 & 3 \\ 0 & 10 & -5 & 5 \\ 0 & 2 & 1 & 7 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & -3 & 3 \\ 0 & 10 & -5 & 5 \\ 0 & 0 & 10 & 30 \end{array} \right)$$

$$x_3 = 3 \rightarrow 10, x_2 = 20 \rightarrow x_2 = 2 \rightarrow 4x_1 = 4, \rightarrow x_1 = 1$$

*5

1.3.3 Gauss Jordan 法

Jordan による改良。元のマトリクスが単位行列になるまで掃き出しを繰り返すと解が得られる。

William Jordan(1842 – 1899) Deutch 測地学者

$$\begin{pmatrix} \hat{a}_{11} & & 0 \\ & \hat{a}_{12} & \\ & & \ddots \\ 0 & & & \hat{a}_{12} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{pmatrix}$$

鶴亀算の例

$$\left(\begin{array}{cc|c} 1 & 1 & 10 \\ 2 & 4 & 34 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 10 \\ 1 & 2 & 17 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 10 \\ 0 & -1 & -7 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 10 \\ 0 & 1 & 7 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} -1 & 0 & -3 \\ 0 & 1 & 7 \end{array} \right) \rightarrow \left(\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 7 \end{array} \right)$$

*5 列も交換する方法もある

Example 3×3

$$\begin{aligned}
& \left(\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 4 & 4 & -3 & 3 \\ 2 & 3 & -1 & 5 \\ -2 & 3 & -1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 1 & 1 & -0.75 & 0.75 \\ 2 & 3 & -1 & 5 \\ -2 & 3 & -1 & 1 \end{array} \right) \left[\begin{array}{cccc} -2 & -2 & 1.5 & -1.5 \\ 2 & 2 & -1.5 & 1.5 \end{array} \right] \rightarrow \\
& \left(\begin{array}{ccc|c} 1 & 1 & -0.75 & 0.75 \\ 0 & 1 & 0.5 & 3.5 \\ 0 & 5 & -2.5 & 2.5 \end{array} \right) \left[\begin{array}{cccc} 0 & -1 & -0.5 & -3.5 \\ 0 & -5 & -2.5 & -17.5 \end{array} \right] \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & -1.25 & -2.75 \\ 0 & 1 & 0.5 & 3.5 \\ 0 & 0 & -5 & -15 \end{array} \right) \rightarrow \\
& \left(\begin{array}{ccc|c} 1 & 0 & -1.25 & -2.75 \\ 0 & 1 & 0.5 & 3.5 \\ 0 & 0 & 1 & 3 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 0 & 0 & 1.25 & 3.75 \\ 0 & 0 & -0.5 & -1.5 \\ 0 & 0 & 1 & 3 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right)
\end{aligned}$$

1.3.4 応用

掃き出し法で逆行列を求める

行列の右に単位行列を付けて *gauss_jordan* 法で掃き出すと逆行列が得られる

```
D130=. 2 3 _1 , 4 4 _3
, : _2 3 _1
```

```
D130
2 3 _1
4 4 _3
_2 3 _1
```

単位行列を付加

```
D130,./~i.3
2 3 _1 1 0 0
4 4 _3 0 1 0
_2 3 _1 0 0 1
```

Gauss_jordan による掃き出し

```
require '~system/packages
/math/linear.ijs'

gauss_jordan D130,./~i.3
1 0 0 0.25 0 _0.25
0 1 0 0.5 _0.2 0.1
0 0 1 1 _0.6 _0.2
```

J のプリミティブの逆行列 %.

```
%.D130
0.25 0 _0.25
0.5 _0.2 0.1
1 _0.6 _0.2
```

■逆行列今昔物語 David Salsburg 竹内、熊谷訳 [統計学を拓いた異才たち] 日経新聞社
2006

当時は未だコンピュータが普及していなかったので、逆行列を求めるのは計算機を使って難しく退屈な作業をするしかなかった。私が大学院生だった頃、学生はみんな逆行列を計算しなければならなかった。私は 5×5 の逆行列を求めるのに数日かかったことを覚えているが、その大半は計算間違いを探すことと、その間違いのやり直しに費やしたのだった

1.4 行列あれこれ/正方形の行列

まずは正方形の行列と連立方程式の拡大係数行列を取り扱う。

1.4.1 rank

正方行列 X に含まれる線形独立な列ベクトルの最大数を行列の階数 (rank) という。

ガウスの消去法 ガウスの消去法で列が 0 にならない次数をランクと数えるのが簡易である。

$$\begin{pmatrix} 2 & 1 & 1 & 15 \\ 4 & 6 & 3 & 41 \\ 2 & 9 & 3 & 37 \end{pmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 15 \\ 4 & 6 & 3 & 41 \\ 2 & 9 & 3 & 37 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 15 \\ 0 & 4 & 1 & 11 \\ 0 & 8 & 2 & 22 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 15 \\ 0 & 4 & 1 & 11 \\ 0 & 8 & 2 & 22 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 15 \\ 0 & 4 & 1 & 11 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

0 が並ぶ場合には、不定と不能の場合がある。

$\begin{bmatrix} 0 & 0 & 0 & = 0 & \text{不定} \end{bmatrix}$ 解は無限にある。ベクトルの次数を調整する必要がある。
 $\begin{bmatrix} 0 & 0 & 0 & = 15 & \text{不能} \end{bmatrix}$ 解が求まらない。不能

小行列式 行列の階数は 0 でない小行列式の最大のサイズと一致する。

Example

```
D140
4  1 3 _1
_7  2 3  4
_1  1 2  1
11 _1 0 _5
```

```
small_matrix D140
+-----+-----+-----+
| 4  1 3 _1| 2 3  4  | 2  1| |
|_7  2 3  4| 1 2  1  | 0 _5|
|_1  1 2  1|_1 0 _5  |  |  |
|11 _1 0 _5|  |  |  |
+-----+-----+-----+
| 0  | 1.66533e_15|_10 |
+-----+-----+-----+
```

2×2 の小行列が 0 でないので、ランクは 2

行列式の値 D140 の行列式の値は 0 であるので、従属ベクトルが存在し、逆行列が取れない

1.4.2 品の良い行列

実対称行列	実数の逆行列が存在する 固有値は実数で重根は出ない。 異なる固有値の固有ベクトルは直交する。 直交行列によって対角化できる	コーシー 全ての対称行列は実数の固有値を持つ ブリキオ 直交行列の固有値は単位円上に分布する
正則行列（正規行列） non-singular matrix	逆行列が存在する行列 $A^{-1}A = AA^{-1} = I$	

1.4.3 単位行列の生成

単位行列の簡単な生成方法

$= \sqrt{1.3}$
1 0 0
0 1 0
0 0 1

1.4.4 複素数の行列

エルミート行列は直交行列の複素数版である。J は実数と複素数の間がシームレスになっているので、計算上特別なことをする必要はない。

Charles Hermite(1822 – 1901) France

エルミート行列 $A = A^*$ A	転置 A^t	複素共役 A^*
D141= $\begin{pmatrix} 1 & 3j_1 \\ 1 & 3j_1 \end{pmatrix}$ D141 $\begin{pmatrix} 1 & 3j_1 \\ 3j_1 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 \\ 3j_1 & 3j_1 \end{pmatrix}$: D141	$\begin{pmatrix} 1 & 3j_1 \\ 3j_1 & 2 \end{pmatrix}$ + D141 $\begin{pmatrix} 1 & 3j_1 \\ 3j_1 & 2 \end{pmatrix}$

複素数の行列の回転は *viewmat* で見ることができる。

```
require 'viewmat'
```

```
viewmat [z=: |. j.~/~ i:2j16
```

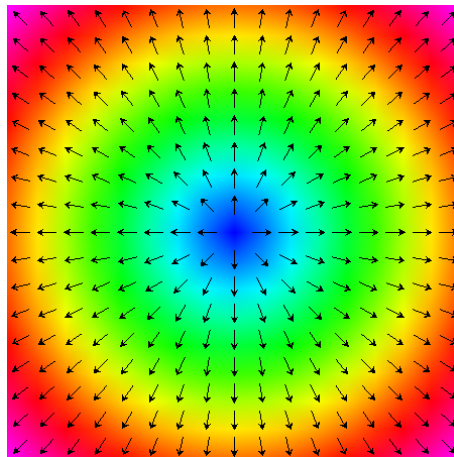


図 1.1 viewmat

正・負の行列

正定行列	$A > 0$	判定法（必要十分条件）は全ての首座小行列式が正 $\det D_i > 0, i = 1, \dots, n$
半正定行列	$A \geq 0$	半正定行列には $\det D_i \geq 0, i = 1, \dots, n$ は使えない
非負行列	プラスの数か 0 で構成された行列 $a_{ij} \geq 0, i, j = 1, \dots, n$ $A \geq 0$	経済などで用いる。

1.4.5 Grammar

複素数 j を用いる。 $3j2$

共役複素数 conjugate +

$3j2$

+ $3j2$

$3j_2$

1.4.6 応用

行列の摂動

本稿は非対称行列をベースにしているので、複素数が頻繁に出現する。

特性方程式は多項式であるので、複素数（2 元数である）も姿を現す。

Chatelin は *C.Moler* からの引用として次のような行列の摂動の例を挙げる。D143 の中央の 180 を僅かに動かすと特性方程式と固有値が摂動する。

a_{22} の周りの小さな摂動が結果として固有値の上に大きな変化を引き起こす場合がある。

D143 $\begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix}$	$-6 + 11\lambda - 6\lambda^2 + \lambda^3 = 0$ Eigenvalue: 3 2 1
D144 $\begin{bmatrix} -149 & -50 & -154 \\ 537 & 180.01 & 546 \\ -27 & -9 & -25 \end{bmatrix}$	$-1.67 + 9.26\lambda - 6.01\lambda^2 + \lambda^3 = 0$ Eigenvalue: 3.5019 2.30083 0.207266
D145 $\begin{bmatrix} -149 & -50 & -154 \\ 537 & 179.998 & 546 \\ -27 & -9 & -25 \end{bmatrix}$	$-6.96602 + 11.3882\lambda - 5.99777\lambda^2 + \lambda^3 = 0$ Eigenvalue: 2.89588 $1.55095 \pm j0.00799921$

グラムの行列式

グラムの行列式 $|G|$ は各縦ベクトルの組み合わせ（内積）を求め、そのマトリクスの行列式の値が 0 でなければ、1 次独立と判定する。

D146 3 _2 _1 _2 4 _1 1 _1 3 , . { : D146 +-----+ 3 _2 1 NB. X1 +-----+ _2 4 _1 NB. X2 +-----+ _1 _1 3 NB. X3 +-----+	グラム行列式 X_1X_1 X_1X_2 X_1X_3 X_2X_1 X_2X_2 X_2X_3 X_3X_1 X_3X_2 X_3X_3	gram_rank D146 +-----+-----+ 14 _15 2 625 _15 21 _5 2 _5 11 +-----+-----+
---	--	--

2つのベクトルの内積が0であればその2つのベクトルは直交している。

1.4.7 Script

```
small_matrix=: 3 : 0
NB. calc determinant of small matrix
NB. Usage: u matrix
SIZE=.|. { (;<: # y), 2) $ 2#}. >: - i. # y NB. make size 2*2 3*3 4*4 ..
MAT=. SIZE {. L:0 y NB. take matrix 2*2 3*3 4*4 ...
MAT,:(-/ . * L:0 MAT) NB. calc determinant of each small matrix
)
```

1.4.8 Reference

F. シャトラン/伊理正夫・由美訳「行列の固有値」Springer-Verlag Tokyo 1993/2003(新装版)

1.5 グラム・シュミットの直交化法

J.P.Gramm(1850 – 1916), *E.Schmidt*(1876 – 1959)

正規直交基底を取る互いが直交した長さが 1 のベクトルに変換する。

直交行列 (orthogonal matrix) とは $T^t = T^{-1}$ を満たすこと、即ち転置行列が逆行列となるような行列である。

$$A^t A = A A^t = I \text{ 直交行列}$$

$$A^t = A^{-1}$$

$U^* U = U U^* = I$ ユニタリ行列とは直交行列を複素数に拡張した行列のこと

$$U^* = U^{-1}$$

1.5.1 グラム・シュミット

$$\begin{aligned} v'_1 &= v_1 & u_1 &= \frac{v_1}{\|v_1\|} \\ v'_2 &= v_2 - (v_2, u_1)u_1 & u_2 &= \frac{v'_2}{\|v'_2\|} \\ v'_3 &= v_3 - (v_3, u_2)u_2 - (v_3, u_1)u_1 & u_3 &= \frac{v'_3}{\|v'_3\|} \\ v'_4 &= v_4 - (v_4, u_3)u_3 - (v_4, u_2)u_2 - (v_4, u_1)u_1 & u_4 &= \frac{v'_4}{\|v'_4\|} \\ &\vdots & &\vdots \end{aligned}$$

Example

v_1	v_2	v_3
1	1	2
1	3	-1
0	1	1

$$\begin{aligned} v'_1 &= v_1 & u_1 &= \frac{v_1}{\|v_1\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\ v'_2 &= v_2 - (v_2, u_1)u_1 & &= \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right) \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} & u_2 &= \frac{v'_2}{\|v'_2\|} = \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \\ v'_3 &= v_3 - (v_3, u_2)u_2 - (v_3, u_1)u_1 & &= \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \right) \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right) \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \frac{5}{6} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} & u_3 &= \frac{v'_3}{\|v'_3\|} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \end{aligned}$$

1.5.2 数値計算

シュミットの直交化は QR 分解で用いられている。Q,R の左がシュミットの直交化でもある。

D150	128!:0 D150
1 1 2	+-----+-----+
1 3 _1	0.707107 _0.57735 0.408248 1.41421 2.82843 0.707107
0 1 1	0.707107 0.57735 _0.408248 0 1.73205 _0.57735
	0 0.57735 0.816497 0 0 2.85774
	+-----+-----+
	NB. Q (shumit orthogonal) R
	'a0 a1'=. 128!:0 D150 NB. Q and R
	a0 +/ . * a1 NB. Q +/ . * R
	1 1 2
	1 3 _1
	0 1 2

1.5.3 Grammar

128 !:0 QR 分解の補助関数

*6

$A = Q$	$A^{-1} = Q^{-1}$
<pre> a0 0.707107 _0.57735 0.408248 0.707107 0.57735 _0.408248 0 0.57735 0.816497 </pre>	<pre> %. a0 0.707107 0.707107 _3.14995e_16 _0.57735 0.57735 0.57735 0.408248 _0.408248 0.816497 </pre>
$A^t = Q^t$	直交行列を A0 とすると $A0^t A0 = I \quad Q^t Q = I$
<pre> : a0 0.707107 0.707107 0 _0.57735 0.57735 0.57735 0.408248 _0.408248 0.816497 </pre>	<pre> (: a0) +/ . * a0 NB. 微少なゴミあり 1 3.88578e_16 1.11022e_16 3.88578e_16 1 _5.55112e_17 1.11022e_16 _5.55112e_17 1 </pre>

*6 J は暫定サポート関数を 128!:0 に入れている

1.6 行列の内積

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \times 3 + 0 \times 2 + 2 \times 1 & 1 \times 1 + 0 \times 1 + 2 \times 0 \\ -1 \times 3 + 3 \times 2 + 1 \times 1 & -1 \times 1 + 3 \times 1 + 1 \times 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

右の項は転置/*transpose* する。transpose した状態で2つのマトリクスは横のサイズが同一でなければならない。(縦は自由)

<pre> D160=. 1 0 2 ,: _1 3 1 D161=. 3 1 2 ,: 1 1 0 D160 1 0 2 _1 3 1 D161 3 1 2 1 1 0 </pre>	<pre> D160 +/ . * D161 NB. AB 5 1 4 2 D161 +/ . * D160 NB. BA 2 3 7 1 3 5 1 0 2 </pre>
--	---

1.6.1 Grammar

NB. 注釈/comment

transpose 転置 |:

```

      |: D161
      3 2 1
      1 1 0

```

内積 行列の内積 +/ . * (右引数の転置は自動的に行われる。)

$AB \neq BA$ 行列の乗算では $AB \neq BA$ である

ボックス ボックスは数や文字列を区切る特別の形式。

ボックス化 {@> スカラー毎にボックスに入れる方法。

次のような方法もある。

```
<"0 D162
```

1	2
3	1

1.6.2 応用

Hadamard products	$\begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 2 \cdot 3 \\ 3 \cdot 2 & 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 6 \\ 6 & 1 \end{bmatrix}$ 右項を転置しない	D162=: 1 2 ,: 3 1 D163=: 0 3 ,: 2 1 D162d 1 2 3 1 D163 0 3 2 1 D162 * D163 0 6 6 1
Kronecher products	$\begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 3 & 2 \cdot 0 & 2 \cdot 3 \\ 1 \cdot 2 & 1 \cdot 1 & 2 \cdot 2 & 2 \cdot 1 \\ 3 \cdot 0 & 3 \cdot 3 & 1 \cdot 0 & 1 \cdot 3 \\ 3 \cdot 2 & 3 \cdot 1 & 1 \cdot 2 & 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 0 & 6 \\ 2 & 1 & 4 & 2 \\ 0 & 9 & 0 & 3 \\ 6 & 3 & 2 & 1 \end{bmatrix}$ 総当たり	({> D162) * L:0 D163 +---+---+ 0 3 0 6 2 1 4 2 +---+---+ 0 9 0 3 6 3 2 1 +---+---+

例題の出典：http://en.wikipedia.org/wiki/Matrix_multiplication/

1.7 ルベリエ・ファディーエバ法

マトリクスの固有値を求める方法の一つにルベリエ・ファディーエバ法がある。非対称行列も取り扱うことができ、特性多項式を求めて、それを解くことで固有値を求める優雅で強力な手法であり学習用に最適である。

*7*8

機械的に解く数値解法よりもマトリクスを一旦特性多項式に変換してその全体像グラフに表すこともでき、マトリクスの全体像を理解しやすい。多項式を解くプロセスが入っているので余り大きなマトリクスは扱えない。

1.7.1 特性方程式を求める

$$A = \begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix} \quad \text{NB. D120}$$

の特性方程式と逆行列を LF 法で求める。マトリクス同士のかけ算は内積を用いる。tr は trace で対角行列の和

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}, B_1 = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}$$

$$P_1 = \text{tr}[B_1] = 1 + 4 = 5$$

$$B_2 = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \right) = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} -4 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

$$P_2 = \frac{1}{2} \text{tr}[B_2] = -2$$

$$P[\lambda] = \lambda^2 - \sum_{i=1}^2 P_i \lambda^{n-1}$$

$$P[\lambda] = 2 - 5\lambda + \lambda^2$$

(符号が逆になる)

特性方程式の解は J の多項式関数 p. に任せよう。特性多項式の解が固有値である。

p. 2 _5 1

++-----+

|1|4.56155 0.438447|

++-----+

plot (i:10); 2 _5 1 &p. i:10

*7 筆者は NJ 州ラトガース大学の Joha Randoll から教わった。

*8 Mathematica が採用している

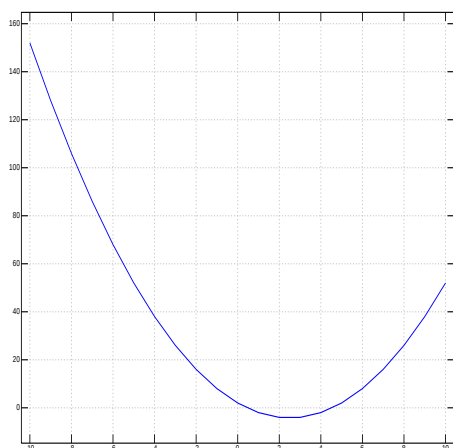


図 1.2 characteristic equation

1.7.2 逆行列を求める

$$A^{-1} = \frac{1}{P_2}(B_1 - P_2 I)$$

$$A^{-1} = -\frac{1}{2} \left(\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \right) = -\frac{1}{2} \begin{bmatrix} -4 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -0.5 \\ -1 & 0.5 \end{bmatrix}$$

1.7.3 Grammar

LF 法で特性方程式と固有値を求める用意した関数

```
char_lf 2 2 $ 1 1 2 4
```

```
+--+-----+-----+
|1|4.56155 0.438447|2 _5 1|
+--+-----+-----+
```

1.7.4 LF 法のアルゴリズム

$$\det(A - \lambda I_n) = 0$$

λ は多項式 $p(\lambda) = \det(A - \lambda I_n)$ の根である。

Leverrier-Faddeev 法は、多項式 $p(\lambda)$ の係数 c_k を求める優れた方法である。

$$p(\lambda) = \lambda^n + c_1 \lambda^{n-1} + c_2 \lambda^{n-2} + \cdots + c_{n-1} \lambda + c_n$$

マトリクスのトレース（対角行列の合計）を $\text{Tr}[A]$ とする

$$\text{Tr}[A] = a_{1,1} + a_{1,t} + \cdots + a_{n,n}$$

補助マトリクス $(B_k)_{k=1}^n$ を作成する。

$$\begin{aligned} B_1 &= A & \rightarrow & p_1 = \text{Tr}[B_1] \\ B_2 &= A(B_1 - p_1 I) & \rightarrow & p_2 = \frac{1}{2} \text{Tr}[B_1] \\ &\dots & & \dots \\ B_k &= A(B_{k-1} - p_{k-1} I) & \rightarrow & p_k = \frac{1}{k} \text{Tr}[B_k] \\ B_n &= A(B_{n-1} - p_{n-1} I) & \rightarrow & p_n = \frac{1}{n} \text{Tr}[B_n] \end{aligned}$$

次の多項式を得る。これをニュートン法などで解けば固有値がめられる。

$$p(\lambda) = \lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} - \dots - p_{n-1} \lambda - p_n$$

次の様に A の逆行列も求められる。

$$A^{-1} = \frac{1}{p_n} (B_{n-1} - p_{n-1} I)$$

1.7.5 Newton 法

ニュートン法は、

$$x - \frac{f(x)}{f'(x)}$$

の反復計算を行うものである。J は微分演算子 D. があるので APL より容易にスクリプトが作れる。

ニュートン法は微分演算子を用いてシンプルに定義できる。動詞を左パラメーターに取るので副詞で定義しなければならない。ランクはベクトルを引数に取ることができるように ("0) とする。 (^:_) は収束まで計算するようにしているが (^:100) 程度で打ち切っても良い。

```
new_1=: 1 : ' ] - x % x D.1' (^:_) ("0)
```

```
2 _5 1&p. new_1 i.5
0.438447 0.438447 0.438447 4.56155 4.56155
```

1.7.6 Grammar

polinomial p. J の多項式の値を計算するプリミティブ。最初に 1 などと反復数も示す
両項の p. 2 _5 1 &p.i:10 のように左に係数（逆順）を、右に区間をとれば推計値が
得られる。

i. i: i. n は 0 から始まる n 個の数値を、i: n はマイナス側にも n 個の数値を出す

単位行列 =/~ i.3 などと求められる

tr trace ,trA = $\sum_{i=1}^n a_{ii}$, 対角行列の和

1.8 LF 法 3×3

2×2 では全貌が掴みにくいので 3×3 を示す。(以降はサイズが大きくなっても同様の反復を行う。)

1.8.1 3×3 の数値例

$$\begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix}$$

特性多項式

特性多項式から, 固有値を求める

$$A = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix}, B_1 = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix}$$

$$p_1 = \text{Tr}[B_1] = 2$$

$$B_2 = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix} \times \left[\begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix} \times \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \right] = \begin{pmatrix} 1 & 7 & -5 \\ 2 & 0 & 1 \\ 2 & -8 & 9 \end{pmatrix}$$

$$p_2 = \frac{1}{2} \text{Tr}[B_2] = \frac{1}{2} \times 10 = 5$$

$$B_3 = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix} \times \left[\begin{pmatrix} 1 & 7 & -5 \\ 2 & 0 & 1 \\ 2 & -8 & 9 \end{pmatrix} \times \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix} \right] = \begin{pmatrix} -6 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -6 \end{pmatrix}$$

$$p_3 = \frac{1}{3} \text{Tr}[B_3] = \frac{1}{3} \times -18 = -6$$

$$P[\lambda] = \lambda^3 - \sum_{i=1}^3 p_i \lambda^{n-1}$$

$$p[\lambda] = 6 + 5\lambda - 2\lambda^2 + \lambda^3$$

λ の最高次数の係数は必ず 1, それ以外の項の係数は $\pm$ の符号が反転する

$p[\lambda] = 6 - 5\lambda - 2\lambda^2 + \lambda^3$ を解いて固有値を求める。J の p. に任せよう。

$$\mathbf{p} = \begin{pmatrix} 6 \\ -5 \\ -2 \\ 1 \end{pmatrix}$$

+-+-+-----+

|1|3 -2 1|

+-+-+-----+

固有値は 3 2 1

*9

逆行列

$$A^{-1} = \frac{1}{p_3}(B_{3-1} - p_{3-1}I)$$

$$A^{-1} = \frac{1}{-6} \times \left[\begin{pmatrix} 1 & 7 & -5 \\ 2 & 0 & 1 \\ 2 & -8 & 9 \end{pmatrix} - \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix} \right] = \frac{1}{-6} \times \begin{pmatrix} -4 & 7 & -5 \\ 2 & -5 & 1 \\ 2 & -8 & 4 \end{pmatrix} = \begin{pmatrix} 4/6 & -7/6 & 5/6 \\ -2/6 & 5/6 & -1/6 \\ -2/6 & 8/6 & -4/6 \end{pmatrix}$$

%. D180

```
0.666667 -1.16667 0.833333
-0.333333 0.833333 -0.166667
-0.333333 1.33333 -0.666667
```

1.8.2 Script

John Randall に啓発されて LF 法のスクリプトを作ってみた。

```
tr=: (<0 1)&|: NB. diag
NB. umatrix=: (=/~)@i.@#
char_lf=: 3 : 0
ANS=. TR_SUM=. +/ tr MAT=. y NB. sum of trace
UMAT=. =/~ i. # y
for_LF. i.<: # y do.
MAT=. y +/ . * MAT - UMAT * TR_SUM
TR_SUM=. (% 2+LF)* +/ tr MAT
ANS=. ANS,TR_SUM
end.
(p. POL), (<POL=. (-ANS),1)
)
```

ループは $n-1$ 回。LF がカウンターになっている。ここで $\frac{1}{2}$ からの p_k を $2+LF$ とする。係数は p. 用に次数の高い方を右にする。最高次の λ の係数 p^n は 1 である。

char_lf D180

```
+-----+-----+
|1|3 -2 1|6 -5 -2 1|
+-----+-----+
```

$$f(\lambda) = 6 - 5\lambda - 2\lambda^2 + \lambda^3$$

Eigenvalues 3 -2 1

最初の枠 (1) は J の反復回数で精度を示している。

次に p. で求めた固有値を先にし、最後の枠は特性多項式の係数を示す。

*9 最初のマスの 1 は p. が多項式を解いたときのループを表している。精度を見るととき以外は特に必要としない

1.9 ケーリー・ハミルトンの定理

ハミルトン (1805-1865) はアイルランドの神童。ケーリー (1821-1895) はイングランド人でケンブリッジで法律と数学を学んだが、父の希望で弁護士として 14 年を送ったのち純粋数学に戻りケンブリッジで教える。法律の勉強中にダブリンへハミルトンの講義を聴きに赴いたこともある。同じく法律家で数学者のシルベスターは友人。

$$\begin{aligned}\phi(\lambda) &= |A - \lambda I| \\ \phi(\lambda) = 0 &\rightarrow \text{特性方程式} \\ \phi(\lambda) &\lambda \leftarrow A \text{ とすると} \\ \phi(A) &= O_n\end{aligned}$$

1.9.1 Example

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 4 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1-\lambda & 1 & 1 \\ 2 & -\lambda & 1 \\ 4 & 1 & 2-\lambda \end{bmatrix}$$

$$|\lambda I - A| = -1 - 5\lambda - 3\lambda^2 + \lambda^3 = 0$$

$$-I - 5A - 3A^2 + A^3 = O_3$$

特性方程式の λ を A に変えた形で表すことが出来る。逆行列や A^n の計算にも用いられる。

1. LF 法で特性方程式を求める。
2. スクリプト `ch` で A^n を求め、ケーリー・ハミルトンの定理を確認する。

D190	char_lf D190	ch D190
1 1 1	+-----+	NB. 0 I A A^2 A^3
2 0 1	1 4.23607 _1 _0.236068 _1 _5 _3 1	+-----+
4 1 2	+-----+	0 0 0 1 0 0 1 1 1 7 2 4 27 11 17
	0 = -1 - 5λ - 3λ ² + λ ³	0 0 0 0 1 0 2 0 1 6 3 4 28 10 17
		0 0 0 0 0 1 4 1 2 14 6 9 62 23 38
		+-----+
		0 ₃ = -I - 5A - 3A ² + A ³
		左端が 0 になればよい

1.9.2 応用

逆行列

$$-I - 5A - 3A^2 + A^3 = O_3$$

両辺に A^{-1} を掛ける

$$-A^{-1} - 5I - 3A + A^2 = O_3$$

$$A^{-1} = 5I - 3A + A^2$$

D190	ch_inv D190	%. D190
1 1 1	_1 _1 1	_1 _1 1
2 0 1	0 _2 1	0 _2 1
4 1 2	2 3 _2	2 3 _2

1.9.3 Script

```
ch=: 3 : 0      NB. Cayley Hamilton theorem
'B EV CR'=: char_lf y      NB. CR is lambda
KH0=:(<=/~i. # y) , mp_matrix y      NB. I A A^2 A^3 ..
(<+/ > KH0 * L:0 {@> CR),KH0
NB. ans is 0//correct
)
```

```
mp_matrix=: 3 : 0      NB. matrix multiplication
TMP=. y +/ . * y
ANS=. y ; TMP
for_ctr. i.<:<: # y do.
TMP=. y +/ . * TMP
ANS=.ANS,<TMP
end.
)
```

```
ch_inv=:3 : 0
NB. modified 05/06/2008
NB. Cayley Hamilton inverse matrix
NB. e.g. A^(-1)= -5I-3A+A^2
'EV CR'=: }. char_lf y NB. CR is lambda
CH0=:(<=/~i. # y) , mp_matrix y NB. I A A^2 A^3 ..
CH1=:(<+/ > (}: CH0) * L:0 {@> }. CR)
```

CH1 % {.CR

NB. ans is 0//correct

)

1.10 固有ベクトル

1. 非対称行列では頻繁に複素数が出る (C^m)。複素数は素直に演算すれば困ることはない。
2. 対角化を行うには、固有ベクトルは各固有値に付き一本（重根の場合は重根の個数分の）異なる固有ベクトルが必要である
3. 固有値に重根がある場合には固有ベクトルが本数分取れる場合と取れない場合がある。取れない場合はジョルダン標準形を用いる。
4. 実計算では重根は希にしか出てこないの、見つけたらサンプリングしておこう

1.10.1 マセマティシャンはマジシャン

$(A - \lambda I)l = 0$ を解いて求める。

$$\begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix}$$

char_lf D180
+-----+
|1|3 _2 1|6 _5 _2 1|
+-----+
 $f(\lambda) = 6 - 5\lambda - 2\lambda^2 + \lambda^3$
Eigenvalues 3 -2 1

$$B = |A - \lambda I| = \begin{bmatrix} 2 - \lambda & -2 & 3 \\ 1 & 1 - \lambda & 1 \\ 1 & 3 & -1 - \lambda \end{bmatrix}$$

$\lambda = 3$ の場合

$$(A - \lambda_1 I)l = \begin{bmatrix} -1 & -2 & 3 \\ 1 & -2 & 1 \\ 1 & 3 & -4 \end{bmatrix} \begin{bmatrix} l \\ m \\ n \end{bmatrix} = 0$$

D180; (3 *=i.3);a1=.D180 - 3*=i.3

NB.D180 I D180-lamda I

```

+-----+-----+-----+
|2 _2 3|3 0 0|_1 _2 3|
|1 1 1|0 3 0| 1 _2 1|
|1 3 _1|0 0 3| 1 3 _4|
+-----+-----+-----+

```

この $A - \lambda I$ の行列は（どのような λ の場合でも）ランク落ちして逆行列も取れない厄介な行列である。ランク落ちは正方行列の場合は不定で拘束がなく任意の数を取ることができるとしてマジックのように固有ベクトルをつかみ取る教科書が多い。

この手法はジョルダン標準形のところで改めて説明する。

1.10.2 随伴マトリクスを用いる方法

Randoll のスクリプトは固有値の部分のみであった。随伴行列から固有ベクトルを求める方法が町田・駒崎・松浦「マトリクスの固有値と対角化」東海大学出版会 1990 に紹介されている。一見複雑そうだが、プログラムは難しくなくて確実に計算できる。

$B_{11} = (-1)^2 \begin{vmatrix} 1-\lambda & 1 \\ 3 & -1-\lambda \end{vmatrix} = \lambda^2 - 4$	$B_{12} = (-1)^3 \begin{vmatrix} -2 & 3 \\ 3 & -1-\lambda \end{vmatrix} = -2\lambda + 7$	$B_{13} = (-1)^4 \begin{vmatrix} -2 & 3 \\ 1-\lambda & 1 \end{vmatrix} = 3\lambda - 5$
$B_{21} = (-1)^3 \begin{vmatrix} 1 & 1 \\ 1 & -1-\lambda \end{vmatrix} = \lambda + 2$	$B_{22} = (-1)^4 \begin{vmatrix} 2-\lambda & 3 \\ 1 & -1-\lambda \end{vmatrix} = \lambda^2 - \lambda - 5$	$B_{23} = (-1)^5 \begin{vmatrix} 2-\lambda & 3 \\ 1 & 1 \end{vmatrix} = \lambda + 1$
$B_{31} = (-1)^4 \begin{vmatrix} 1 & -1\lambda \\ 1 & 3 \end{vmatrix} = \lambda + 2$	$B_{32} = (-1)^5 \begin{vmatrix} 2-\lambda & -2 \\ 1 & 3 \end{vmatrix} = 3\lambda - 8$	$B_{33} = (-1)^6 \begin{vmatrix} 2-\lambda & -2 \\ 1 & 1-\lambda \end{vmatrix} = \lambda^2 - 3\lambda + 4$

$$\begin{aligned}
 adj(B) &= \begin{bmatrix} \lambda^2 - 4 & -2\lambda + 7 & 3\lambda - 5 \\ \lambda + 2 & \lambda^2 - \lambda - 5 & \lambda + 1 \\ \lambda + 2 & 3\lambda - 8 & \lambda^2 - 3\lambda + 4 \end{bmatrix} \\
 &= \lambda^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & -2 & 3 \\ 1 & -1 & 1 \\ 1 & 3 & -3 \end{bmatrix} + \begin{bmatrix} -4 & 7 & -5 \\ 2 & -5 & 1 \\ 2 & -8 & 4 \end{bmatrix}
 \end{aligned}$$

この複雑な計算が `char_lf` を少しバックさせると出来ていた。2 項と 3 項は LF 法で生成する二つのマトリクスを引けば求められる。

$$\begin{aligned}
 &\begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix} - \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \\
 &\begin{pmatrix} 1 & 7 & -5 \\ 2 & 0 & 1 \\ 2 & -8 & 9 \end{pmatrix} - \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}
 \end{aligned}$$

ここから λ に固有値を入れると固有ベクトルが得られる。

各固有値に対応するベクトルを任意に一本ずつ取ればよい。(原型を見るために規格化していない)

```
char_lf_evec_sub=: 3 : 0
TR_SUM=. +/ tr MAT=. y NB. sum of trace
ANS=. <UMAT=. =/~ i. # y
for_LF. i.<: # y do.
MAT=. y +/ . * TMP=. MAT - UMAT * TR_SUM
TR_SUM=. (% 2+LF)* +/ tr MAT
ANS=. ANS,<TMP
end.
)
```

ここでの行列は $B_k = A(B_{k-1} - p_{k-1}I)$ の $(B_{k-1} - p_{k-1}I)$ の部分を用いればよい。

```
char_lf_evec_sub D180
+-----+-----+-----+
|1 0 0|0 _2 3|_4 7 _5|
|0 1 0|1 _1 1| 2 _5 1|
|0 0 1|1 3 _3| 2 _8 4|
+-----+-----+-----+
```

固有ベクトルを求める (規格化されていない)

```
EIGEN2
3      _2      1
+-----+-----+-----+
|+--+--+|+--+--+|+--+--+| | | | | | | | | | | | |
||9|3|1|||4|_2|1|||1|1|1||
|+--+--+|+--+--+|+--+--+|
+-----+-----+-----+
```

```
char_evec D180
+-----+-----+-----+
|3      |_2      |1      |
+-----+-----+-----+
|5 1 4|0 11 _11|_3 5 _2|
|5 1 4|0 1 _1| 3 _5 2|
|5 1 4|0 _14 14| 3 _5 2|
+-----+-----+-----+
```

この $\lambda^2, \lambda, 1$ の値を固有値毎に char_lf_evec_sub0 に当てはめる

Script

```
char_evec=: 3 : 0
EIGEN=: {@> ; 1{ char_lf y
EIGEN2=: {@>L:0 EIGEN ^/L:0 |.i.# EIGEN
ADJMAT=: char_lf_evec_sub y
ANS=. <'
for_LF. i. # y do.
TMP=. +/> ( > LF{ EIGEN2) * L:0 ADJMAT
ANS=. ANS,<TMP
end.
EIGEN,:}. ANS
)
```

1.11 標準化

1.11.1 ノルム・距離・角度

	Example $x = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}, y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$	a1 <pre>-1 1 2</pre> a2 <pre>1 0 -1</pre>
<i>Euclidean vector norm</i> $\ x\$	$x'x = \sum_{i=1}^n x_i^2$ $\ x\ = \sqrt{x'x}$	<pre>%: a1 +/ . * a1 2.44949 = sqrt(6)</pre>
<i>distance between x and y</i>	$\ x - y\ = \sqrt{(x - y)'(x - y)}$	<pre>(a1 - a2) +/ . * a1 - a2 14 %: (a1 - a2) +/ . * a1 - a2 3.74166</pre>
<i>cosine of angle between x and y</i>	$\cos\theta = \frac{x'y}{\ x\ \ y\ }$	<pre>(a1 +/ . * a2) -3 %: a1 +/ . * a1 2.44949 %: a2 +/ . * a2 1.41421 -3 sqrt(6) sqrt(2) = -0.866025 = 150° 相関係数</pre>

<i>unit length</i>	$\ x\ =\ y\ =1$ $u_x = \frac{x}{\ x\ }$	<code>a1 % %: a1 +/ .* a1</code> <code>_0.408248 0.408248 0.816497</code>
--------------------	--	--

1.11.2 標準化

標準化とは、データを平均 0、分散 1 に変換することであり、単位が異なるデータの回帰や図示に用いる。

KX ある果実の直径 *cm*

KY 水分含有率 (%) のデータ。

(データの出典 金谷 「これならわかる応用数学教室」)

KX KY
5.6 30
5.8 26
6 33
6.2 31
6.4 33
6.4 35
6.4 37
6.6 36
6.8 33

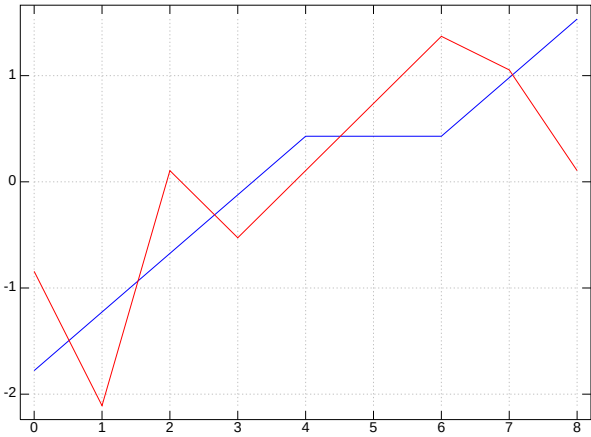


図 1.3 果実の直径 (直線) と水分含有率

標準化	$\frac{x - \bar{x}}{\sqrt{\frac{1}{n} \sum (x - \bar{x})^2}}$	データを平均 0、分散 1 に変換する stand KD <code>_1.7781 _0.843274</code> <code>_1.22628 _2.10819</code> <code>_0.674453 0.105409</code> <code>_0.122628 _0.527046</code> <code>0.429198 0.105409</code> <code>0.429198 0.737865</code> <code>0.429198 1.37032</code> <code>0.981023 1.05409</code> <code>1.53285 0.105409</code>
-----	---	---

平均を引く $x - \bar{x}$	<code>dev=:-(+/%#)</code> <code>dev2=: -"1(+/%#)NB. rank 1</code>	単位が異なるものはうまく一枚のグラフには表せないことが多いが標準化すると単位が無くなり、同一のグラフに表すことができる。 *10
------------------------	--	---

Script

```
stand=: dev % "1 sd
```

```
dev2=: -"1(+/% #)
```

```
var=: # %~ ([:+/[ : *: dev) NB. 分散
```

```
sd=. %:@var NB. 標準偏差
```

1.11.3 Reference

金谷健一 これなら分かる応用数学教室 共立出版 2000

1.12 分散共分散行列

数式に書くと厳めしいがスクリプトは単純である。データを予め標準化しておけば相関行列が求められる。

$$C = \frac{tXX}{N} = \frac{1}{N} \begin{pmatrix} x_1 - \bar{x} & x_2 - \bar{x} & \dots & x_n - \bar{x} \\ y_1 - \bar{y} & y_2 - \bar{y} & \dots & y_n - \bar{y} \end{pmatrix} \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ x_2 - \bar{x} & y_2 - \bar{y} \\ x_3 - \bar{x} & y_3 - \bar{y} \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

$$\frac{1}{N} \begin{pmatrix} (x_n - \bar{x})(x_n - \bar{x}) & (x_n - \bar{x})(y_n - \bar{y}) \\ (y_n - \bar{y})(x_n - \bar{x}) & (y_n - \bar{y})(y_n - \bar{y}) \end{pmatrix} = \begin{pmatrix} x \text{ の分散} & \text{共分散} \\ \text{共分散} & y \text{ の分散} \end{pmatrix}$$

項目/数式	Example
算術平均 $\frac{1}{n} \sum_{i=1}^n X_i$ am=: +/ % # am2=: # %~ +/	am STYLE 166.292 53.375 86.5417 58.7083 90.625
分散共分散行列	dev2=. -"1 +/%# (: dev2 STYLE) +/ . * dev2 STYLE vartable STYLE 5.87326 3.39063 1.88368 2.2934 0.0677083 3.39063 6.65104 1.92187 2.94271 1.51563 1.88368 1.92187 5.66493 0.366319 2.45313 2.2934 2.94271 0.366319 6.4566 0.401042 0.0677083 1.51563 2.45313 0.401042 5.90104

相関行列	cortable STYLE				
	1	0.542494	0.326565	0.372425	0.0115011
	0.542494	1	0.3131	0.449056	0.241926
	0.326565	0.3131	1	0.0605705	0.424285
	0.372425	0.449056	0.0605705	1	-0.0649716
	0.0115011	0.241926	0.424285	-0.0649716	1

■Grammar

*	:	<i>square</i>	2 乗 (^2 と同じ)
%	:	<i>square_root</i>	2 乗根 ($\sqrt{\quad}$)
- "1(+/%#)		$x - \bar{x}$	残差
[	:	<i>Cap</i>	関数を記述順に実行するように指定

第 2 章

マトリクスの分解と相似変換

2.1 LU 分解・LDU 分解

A が正則ならば
 $A = LU$
 $A = LDU$
の形に分解できる。

2.1.1 LU 分解

$A = \begin{bmatrix} 2 & 1 & 1 \\ -4 & 1 & 3 \\ -2 & 2 & 1 \end{bmatrix}$	$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow$ 単位行列の第 1 列に $-a_{21}/a_{11}, -a_{31}/a_{11}$ を入れる	$A^{(1)} = T_1 A = \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \rightarrow$ 内積計算
	$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \rightarrow$ 単位行列に $-a_{32}^{(1)}/a_{22}^{(1)}$	$A^{(2)} = T_2 A^{(1)} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$ 内積計算
	$T = T_2 T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} =$ $\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 1 & 1 \end{bmatrix} \rightarrow$ T_2, T_1 の内積	$L = T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$ 逆行列

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1r2 \end{bmatrix} = LU$$

(例題の出典 木村 P30)

数値計算では絶対値最大のピボットをはじめに持ってくるように列を交換する。

0 のピボットの除算を避けるように列交換を行う。

```
require '~system/packages/math/linear.ijs'
require '~system/packages/math/matfacto.ijs'
```

```
lud D210
```

```
+-----+-----+-----+
|  1 0 0|_4  1  3|0 1 0|
|_1r2 1 0| 0 3r2 5r2|1 0 0|
| 1r2 1 1| 0  0 _7r2|0 0 1|
+-----+-----+-----+
                                pivot
```

2.1.2 LDU 分解

先の U を DU に分解する。 D は U の対角行列。新しい U は U を D で割ったもの。

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1r2 \end{bmatrix}$$

$$\begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1r2 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1r2 \end{bmatrix} \begin{bmatrix} 1 & -1r2 & -1r2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1r2 \end{bmatrix} \begin{bmatrix} 1 & -1r2 & -1r2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} = LDU$$

```
a3 % diag a3  NB. a3 is U0
1 _1r2 _1r2
0  1  _1
0  0  1
```

2.2 コレスキー分解

André Louis Cholesky(1875-1918) ウクライナ系のフランス人で軍の測量技師。第 1 次大戦で死亡。

2.2.1 コレスキー分解

A が対称かつ正定置ならば下三角行列 L が存在し

$$A = LL^t$$

$$A = LDL^t$$

の形に分解できる。さらに分解は対角要素を正になるように定めることで一意に定まる。

対称性に着目し、ガウス–ジョルダン法のように LU 分解に右からの内積のステップを加える。

$A = \begin{bmatrix} 4 & -2 & -6 \\ -2 & 10 & 9 \\ -6 & 9 & 14 \end{bmatrix}$	$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & 0 & 1 \end{bmatrix} \rightarrow$ 単位行列の第 1 列に $-a_{21}/a_{11}, -a_{31}/a_{11}$ を入れる	$A^{(1)} = T_1 A T_1^t = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 6 \\ 0 & 6 & 5 \end{bmatrix} \rightarrow$ 2 重の内積計算
	$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2r3 & 1 \end{bmatrix} \rightarrow$ 単位行列に $-a_{32}^{(1)}/a_{22}^{(1)}$	$A^{(2)} = T_2 A^{(1)} T_2^t = \Sigma = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$ U^2 2 重の内積計算
$A = T^{-1} U U (T^{-1})^t$ $V = T^{-1} U$ とおくと $A = V V^t$	$T = T_2 T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2r3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \mathbf{1r2} & 1 & 0 \\ \mathbf{7r6} & -2r3 & 1 \end{bmatrix}$ $T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & \mathbf{2r3} & 1 \end{bmatrix}$ $(T_2 T_1)^{-1} U = \begin{bmatrix} 1 & 0 & 0 \\ -\mathbf{1r2} & 1 & 0 \\ \mathbf{3r2} & \mathbf{2r3} & 1 \end{bmatrix} \begin{bmatrix} \sqrt{4} & 0 & 0 \\ 0 & \sqrt{9} & 0 \\ 0 & 0 & \sqrt{1} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ -3 & 2 & 1 \end{bmatrix} =$ $V(\text{choleski} - \text{decomposition})$	

$A = VV^t$	$\begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -3 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & -2 & -6 \\ -2 & 10 & 9 \\ -6 & 9 & 14 \end{bmatrix}$
------------	---

$Ly = b$ を解く

2.2.2 修正コレスキ分解

LDL 分解のコレスキー版

<pre> D220 4 _2 _6 _2 10 9 _6 9 14 </pre>	<pre> v % 2 3 1 1 0 0 _1r3 1 0 _3 2 1 diag v </pre>	<pre> d0 2 0 0 0 3 0 0 0 1 diag v *=i.3 </pre>
<pre> v 2 0 0 _1 3 0 _3 2 1 </pre>	<pre> c=.:(: v % 2 3 1) +/. * d0 2 0 0 _1 3 0 _3 2 1 choleski D220 </pre>	<pre> c +/. * : c 4 _2 _6 _2 10 9 _6 9 14 c=v D220 </pre>

$DL^t x = y$ を解く

J のパッケージ

```
require '~system/packages/math/matfacto.ijs'
```

```

choleski D220 NB. A
2 0 0
_1 3 0
_3 2 1

```

2.2.3 Reference

木村英紀「線形代数」東京大学出版会 2003

2.3 QR 分解

(R) を求めるには元のデータを Gram-shumit のノルム化した直交行列 (Q) で割ればよい。1961 年に *Francis* と *Kublanovskaya* が独立にに考案した。^{*1}

シュミットの直交化は QR 分解で用いられている。Q,R の左がシュミットの直交化でもある。

D150	128!:0 D150
1 1 2	
1 3 _1	
0 1 1	
	<pre> +-----+-----+ 0.707107 _0.57735 0.408248 1.41421 2.82843 0.707107 0.707107 0.57735 _0.408248 0 1.73205 _0.57735 0 0.57735 0.816497 0 0 2.85774 +-----+-----+ NB. Q (shumit orthogonal) R 'a0 a1'=. 128!:0 D150 NB. Q and R a0 +/ . * a1 NB. Q +/ . * R 1 1 2 1 3 _1 0 1 2 </pre>

^{*1} householder matrix を用いる方法もある。

$$\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33} \end{bmatrix}$$

```
D150 %. a0    NB. A %. Q = R
      1.41421 2.82843 0.707107
1.11022e_16 1.73205 _0.57735
      0      0 2.85774
```

2.3.1 QR 法により固有値を求める

QR 分解の左右の内積を何回か反復すると固有値が求められる。(反復解法)

Script は次のように僅か 1,2 行と簡単である。

```
pwr=: ^:100 NB. repeat 100 times (change you like)
eval_t=: (>{: +/ . * >{:.)&(128 !:0)
```

eval_t pwr D180 NB. 100 回で完全に収束していない

```
      3      0.324443 _0.749269
_9.88131e_324      _2 _4.04145
      0 4.94066e_324      1
char_lf D180
++-----+-----+
|1|3 _2 1|6 _5 _2 1|
++-----+-----+
```

2.3.2 Grammar

power 関数型定義のループで慣れると便利である。^: 100 は 100 回リピート。~:_は収束するまでループ。(癖の悪い行列では永久反復することもあるので注意)

128!:0 QR 分解の補助関数

2.4 SVD 変換（特異値分解）

singular value decomposition

J の package/math に `svd.ijs` が入っている。^{*2}

ネットで拾った幾つかの声

- 学校では Jordan を教えるが、実務では圧倒的に SVD 法を用いる。
- SVD 法は面長や幅の広い非正方行列に適用できる。
- 金谷の本が唯一わかりやすい。買って置いて良かった。

早速「金谷」を取り出してみた。画像処理の著書を多く書いている関係か SVD は画像処理の項に入っていた。^{*3}

$$P = U \Sigma V^T$$

$$P = U \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix} V^T, \sigma_i = \sqrt{\lambda_i}, i = 1, \dots, r$$

P	$M = PP^T$	$N = P^T P$
D260 3 1 2 3 2 1	D260 +/ . * : D260 14 13 13 14	(: D260) +/ . * D260 18 9 9 9 5 4 9 4 5
M と N の 0 でない固 有値は一致 する。	char_lf D260 +/ . * : D260 +-+-----+ 1 27 1 27 _28 1 +-+-----+	char_lf (: D260)+/ . * D260 +-+-----+ 1 27 1 0 0 27 _28 1 +-+-----+
%: 27 1 5.19615 1	pick_evec D260 +/ . * : D260 0.707107 0.707107 0.707107 _0.707107	pick_evec (:D260) +/ . * D260 0.816497 0 _0.57735 0.408248 _0.707107 0.57735 0.408248 0.707107 0.57735

^{*2} LAPACK にも入っている

^{*3} $m^2 \times m^2, m = 512$ で 512^4 のサイズでも SVD 分解だと計算が実用の範囲に入るとのことである。

svd D260

```

+-----+-----+-----+
|0.707107 _0.707107|5.19615 1|0.816497 4.44089e_16|
|0.707107  0.707107|          |0.408248    0.707107|
|              |          |0.408248    _0.707107|
+-----+-----+-----+

```

U	P の出力の基底となる正規直交ベクトル
Σ	Σ は特異値を対角成分に持つ。特異値は増幅率を表し、入力成分がそれぞれ何倍されて出力されるかを表す
V^T	P の入力の基底となる正規直交ベクトル

2.5 マトリクスの相似変換

オードリー・ヘップバーンのドレスは若き日のジバンシーが担当した。ヘップバーンとジバンシーのドレスの双方を楽しむことができる。

行列 A に左右から正則な行列とその逆行列をかけることを A の相似変換という。

$$A' = TAT^{-1}$$

相似変換によっても、行列式、トレース、固有値は変わらない。

*4

2.5.1 対角化

各固有値に対応する固有ベクトルを 1 本ずつとる。固有ベクトルは大きさは無視してよい。

Example	<pre> D180 2 _2 3 1 1 1 1 3 _1 </pre>
特性方程式と 固有値	<pre> char_lf D180 +---+-----+-----+ 1 3 _2 1 6 _5 _2 1 +---+-----+-----+ </pre> $6 - 5\lambda - 2\lambda^2 + \lambda^3$
固有ベクトル	<pre> av=. pick_evec D180 0.57735 0.616849 _0.57735 0.57735 0.0560772 0.57735 0.57735 _0.785081 0.57735 </pre>
対角化	<pre> (%av)+/ . * D180 +/ . * av 3 4.44089e_16 5.55112e_17 8.88178e_16 _2 0 1.55431e_15 5.55112e_16 1 (微少なゴミが入っている) </pre>

*4 ローマの休日は担当していない

2.6 ジョルダン標準型

線形数学の高峰ジョルダン標準型 (*Jordan canonical form*) の登山口にたどり着けた。

1. 潰れた行列では固有値に重根がでる場合があること。
2. 重根の場合、固有ベクトルが重根と同じ個数取れる場合と、取れない場合があること。

この高峰の登頂に成功したのがジョルダンである。こちらはカミーユ・ジョルダン (Camille Jordan, 1838 年-1922) フランスの数学者。

父と旅をした若き日のモーツアルトはローマのシスティーナ礼拝堂で秘曲 9 声のグレゴリオ聖歌を聴き、宿で書き写し次の日再度聞いて数カ所のチェックで再現したと伝えられる。

相似変換で対角化したマトリクスは対角行列に固有値が並び見事なポリフォニーとなるが、潰れた重根を持つマトリクスでは些か不協和音が出る。これを見事なバランスで纏めるのがジョルダン標準形である。

2.6.1 一斉集合

潰れた行列のチェックにツール類に一斉に登場してもらおう

Example 3×3 笠原

```
D261
_1  2  0 _1
 1  2 _1 _2
_1 _1  0  1
 2  6 _2 _4
```

(例題の出典 笠原)

ルベリエ・ファディーエバ法 5 重根がある

```
+--+-----+-----+
|1|_1 _1 _1 _1|1 4 6 4 1|
+--+-----+-----+
```

行列式 0 ではない

```
-/ . * D261
1
```

J の逆行列 逆行列は取れる

```
%. D261
_1 _2  0  1
```

```
_1 _4  1  2
 1  1 _2 _1
_2 _6  2  3
```

小行列式 ランクを検出できない

```
small_matrix D261
+-----+-----+-----+
|_1  2  0 _1| 2 _1 _2| 0  1|
| 1  2 _1 _2|_1  0  1|_2 _5|
|_1 _1  0  1| 6 _2 _5|      |
| 2  6 _2 _5|      |      |
+-----+-----+-----+
|1      |_1      |2      |
+-----+-----+-----+
```

gauss_elimination/ガウスの掃き出し J のライブラリの *gauss_elimination*
一応通る

```
gauss_elimination D261
+-----+-----+-----+
|2 6  _2      _5|1 3 2 0|0 3|
|0 5  _1      _3.5|      |1 3|
|0 0 _0.6      _0.1|      |  |
|0 0  0 _0.166667|      |  |
+-----+-----+-----+
```

LAPACK LAPACK も難渋している。

```
require '~addons/math/lapack/lapack.ijs'
require '~addons/math/lapack/dgeev.ijs'
```

```
,. }. dgeev_jlapack_ D261
+-----+-----+-----+
|_1 _1 _1j4.38896e_8 _1j_4.38896e_8      |
+-----+-----+-----+
|      0 _9.06493e_17 0.227886j_4.34212e_10 0.227886j4.34212e_10|
|_0.408248      _0.408248  0.426309j5.00091e_9 0.426309j_5.00091e_9|
| 0.408248      0.408248 _0.198423j_4.14202e_9 _0.198423j4.14202e_9|
|_0.816497      _0.816497      0.852618      0.852618|
+-----+-----+-----+
```

固有値が違う
対角化出来ていない。

*5

```
(%. ev ) +/ . * D261 +/ . * ev
|domain error
| ( %.ev)+/ .*D261+/ .*ev
```

ガウスの消去法 フルランクと言っている

```
gausscp_sub D261
+-----+-----+-----+-----+
|1 0.333333 _0.333333 _0.833333|_0.833333 _0.4 0.166667 _0.166667|4|
|0      1      _0.4      _0.4|      | |
|0      0      1 0.166667|      | |
|0      0      0 _0.166667|      | |
+-----+-----+-----+-----+
```

多重根 2重根なら運が良ければベクトルが2本取れるが3重根なら絶望と思い知る。

2.6.2 Jordan

4重根 EXample は-1の4重根であった。次の何れかになる。

$$\begin{array}{c|c} \lambda_1 & 1 \\ \hline \lambda_1 & \end{array}, \quad \begin{array}{c|c} \lambda_1 & 1 \\ \hline \lambda_1 & 1 \\ \hline & \lambda_1 \end{array}, \quad \begin{array}{c|c|c} \lambda_1 & 1 & \\ \hline & \lambda_1 & \\ \hline & & \lambda_1 \end{array}$$

ランク 小行列式からランクを2と推測する

```
small_matrix a1
+-----+-----+-----+
| 0 2 0 _1| 3 _1 _2| 1 1|
```

*5 固有ベクトルがパワー不足で対角行列のみにならないで、固有値が欠け、他の箇所に成分が残る場合もある。

```
}. dgeev_jlapack_ D260
+-----+-----+
|2 2 2|1 _0.57735 0.57735|
| 0 0.57735 _0.57735|
| 0 _0.57735 0.57735|
+-----+
a1 is eigenvector

(%. a1) +/ . * D260 +/ . * a1
2 3.8002e_9 _3.8002e_9
0 2.13397 _0.133975
0 0.133975 1.86603
```

```

| 1 3 _1 _2|_1 1 1|_2 _4|
|_1 _1 1 1| 6 _2 _4|      |
| 2 6 _2 _4|      |      |
+-----+-----+-----+
|0          |0          |_2  |
+-----+-----+-----+

```

λ を入れる $A - (-1)I, (A - (-1)I)^2$ を求める

$A - (-1)I =$ <pre> 0 2 0 _1 1 3 _1 _2 _1 _1 1 1 2 6 _2 _4 a1=._1 small_pol D261 </pre>	$A - (-1)I^2 =$ <pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 a1 +/ . * a1 漂数は 2 </pre>
---	---

*6

マジシャン ランクを見ながら 2 本の固有ベクトルをつかみ取る。 $A - (-1)I$ にかける

```

a1 +/ . * 0 1 1 1
1 0 1 0
a1 +/ . * 0 1 0 1
1 1 0 2

```

固有ベクトル P

```

ev=. 1 0 1 0 ,. 0 1 1 1 ,. 1 1 0 2 ,. 0 1 0 1
ev
1 0 1 0
0 1 1 1
1 1 0 0
0 1 2 1

```

*6 グラム・シュミットの直交化も効かない

```

128!:=0 a
|domain error
|      128!:=0 a

```


対角化 $P^{-1}AP$

```

      (%. ev) +/ . * D261 +/ . * ev
_1  1  0  0
 0 _1  0  0
 0  0 _1  1
 0  0  0 _1

```

しかしながら任意の固有ベクトルをつかみ取る方法はなかなか難しい。 P を巧く取らないと重複固有値は対角に来るが 1 以外の数字が散乱することになる。

第 3 章

線形写像

3.1 写像

一面に虫除けのサッシのように小さい点を作り次の小さいマトリクスを作用させると点が少し動く。

$$\begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

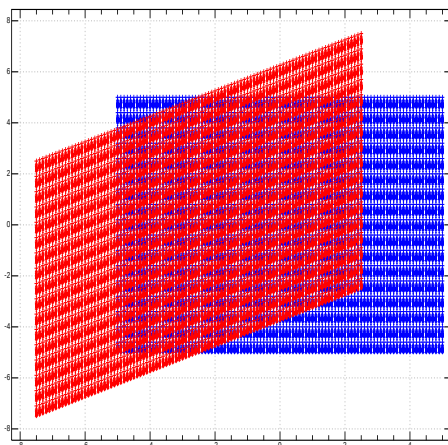


図 3.1 blue → red

次の計算例より点の数はかなり多い

A0

1 1r2

1r2 1

A0 plot_vfield 5

pd 'eps /temp/vfield2_0.eps'

EXample

```

({@> ;("1),. 5# <i. 5), L:0 {@>|. i.5 a1 +/ . * L:0 a
+---+---+---+---+---+
|0 4|1 4|2 4|3 4|4 4|
+---+---+---+---+---+
|0 3|1 3|2 3|3 3|4 3|
+---+---+---+---+---+
|0 2|1 2|2 2|3 2|4 2|
+---+---+---+---+---+
|0 1|1 1|2 1|3 1|4 1|
+---+---+---+---+---+
|0 0|1 0|2 0|3 0|4 0|
+---+---+---+---+---+
+-----+-----+-----+-----+
|2 4 |3 9r2 |4 5 |5 11r2 |6 6 |
+-----+-----+-----+-----+
|3r2 3|5r2 7r2|7r2 4|9r2 9r2|11r2 5|
+-----+-----+-----+-----+
|1 2 |2 5r2 |3 3 |4 7r2 |5 4 |
+-----+-----+-----+-----+
|1r2 1|3r2 3r2|5r2 2|7r2 5r2|9r2 3 |
+-----+-----+-----+-----+
|0 0 |1 1r2 |2 1 |3 3r2 |4 2 |
+-----+-----+-----+-----+

```

3.2 スペクトル分解

行列を射影の一次結合で表すことをスペクトル分解という。

固有値が重根の場合には、部分分数展開が必要。

行列 A がスペクトル分解可能のとき、その射影行列は $P_1, \dots, P_r$ は次により求められる。

$$P_k = \frac{(A - \lambda_1 I) \cdots \underbrace{(A - \lambda_{k-1} I)}_k \cdots (A - \lambda_r I)}{(\lambda_k - \lambda_1) \cdots \underbrace{(\lambda_k - \lambda_{k-1})}_k \cdots (\lambda_k - \lambda_r)}$$

$$\begin{aligned} P_1 &= \frac{(A - \lambda_2 I)(A - \lambda_3 I)}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} \\ P_2 &= \frac{(A - \lambda_1 I)(A - \lambda_3 I)}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} \\ P_3 &= \frac{(A - \lambda_1 I)(A - \lambda_2 I)}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)} \end{aligned}$$

3.2.1 Example

```

char_lf a
a=.2 _2 3,1 1 1,:1 3 _1
2 _2 3
1 1 1
1 3 _1

+-----+-----+
|1|3 _2 1|6 _5 _2 1|
+-----+-----+

fx = 6 - 5λ - 2λ2 + λ3
Eigenvalue: 3 -2 1

P1 =  $\frac{(A - \lambda_2 I)(A - \lambda_3 I)}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} = \frac{1}{(3 - (-2))(3 - 1)} \begin{bmatrix} 4 & -2 & 3 \\ 1 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 3 \\ 1 & 0 & 1 \\ 1 & 3 & -2 \end{bmatrix} = \frac{1}{10} \begin{bmatrix} 5 & 1 & 4 \\ 5 & 1 & 4 \\ 5 & 1 & 4 \end{bmatrix}$ 

P2 =  $\frac{(A - \lambda_1 I)(A - \lambda_3 I)}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} = \frac{1}{(-2 - 3)(-2 - 1)} \begin{bmatrix} -1 & -2 & 3 \\ 1 & -2 & 1 \\ 1 & 3 & -4 \end{bmatrix} \begin{bmatrix} 1 & -2 & 3 \\ 1 & 0 & 1 \\ 1 & 3 & -2 \end{bmatrix} =$ 

 $\frac{1}{-15} \begin{bmatrix} 0 & 11 & -11 \\ 0 & 1 & -1 \\ 0 & -14 & 14 \end{bmatrix}$ 

P3 =  $\frac{(A - \lambda_1 I)(A - \lambda_2 I)}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)} = \frac{1}{(1 - 3)(1 - (-2))} \begin{bmatrix} -1 & -2 & 3 \\ 1 & -2 & 1 \\ 1 & 3 & -4 \end{bmatrix} \begin{bmatrix} 4 & -2 & 3 \\ 1 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} = \frac{1}{-6} \begin{bmatrix} -3 & 5 & -2 \\ 3 & -5 & 2 \\ 3 & -5 & 2 \end{bmatrix}$ 

A - λI の組み合わせ

MAT
+-----+-----+
|_1 _2 3|4 _2 3|1 _2 3|

```

```
| 1 _2 1|1 3 1|1 0 1|
| 1 3 _4|1 3 1|1 3 _2|
+-----+-----+-----+
```

λ の組み合わせ

LAMBDA2

```
+-----+-----+-----+
|3 _2 1|_2 3 1|1 3 _2|
+-----+-----+-----+
```

spectol a

```
+-----+-----+-----+
|1r2 1r10 2r5|0 11r15 _11r15| 1r2 _5r6 1r3|
|1r2 1r10 2r5|0 1r15 _1r15|_1r2 5r6 _1r3|
|1r2 1r10 2r5|0 _14r15 14r15|_1r2 5r6 _1r3|
+-----+-----+-----+
```

(基準化は行っていない。)

各固有値毎に異なる固有ベクトルを一本選べばよい。

スペクトル分解は

$$A = \lambda_1 P_1 + \lambda_2 P_2 + \lambda_3 P_{31}$$

となる。LF 法で求めた固有ベクトルと比較すると LF 法は分子の部分と同一である。

char_evec a

```
+-----+-----+-----+
|3      |_2      |1      |
+-----+-----+-----+
|5 1 4|0 11 _11|_3 5 _2|
|5 1 4|0 1 _1| 3 _5 2|
|5 1 4|0 _14 14| 3 _5 2|
+-----+-----+-----+
```

3.2.2 部分分数展開とスペクトル分解

重根がある場合は部分分数展開を行って後、スペクトル分解のジョルダン標準型版と言うべき形に持ち込む。

$$\frac{1}{\Phi(\lambda)} = \frac{g_1(\lambda)}{(\lambda - \lambda_1)^{m_1}} + \cdots + \frac{g_r(\lambda)}{(\lambda - \lambda_r)^{m_r}}$$

部分分数展開

Example issue: 笠原

D330

2	0	1
_1	3	1
1	_1	2

$$\begin{aligned} \text{部分分数展開} \quad & \frac{1}{(\lambda-2)^2(\lambda-3)} \\ = & \frac{1}{(\lambda-2)^2(\lambda-3)} \frac{A}{\lambda-2} + \frac{B}{(\lambda-2)^2} + \frac{C}{\lambda-3} \end{aligned}$$

分母を払う 分母を払う

$$\begin{aligned} 1 &= A(\lambda - 2)(\lambda - 3) + B(\lambda - 3) + C(\lambda - 2)^2 \\ 1 &= (A + C)\lambda^2 + (-5A + B - 4C)\lambda + 6A - 3B + 4C \\ \begin{cases} A & & +C & = 0 \\ -5A & +B & -4C & = 0 \\ 6A & -3B & +4C & = 1 \end{cases} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ -5 & 1 & -4 & 0 \\ 6 & -3 & 4 & 1 \end{bmatrix} \end{aligned}$$

連立方程式 連立方程式を解く

$$a_0 = .1 \ 0 \ 1, _5 \ 1 \ _4 \ , : \ 6 \ _3 \ 4$$

	a0	
1	0	1
_5	1	_4
6	_3	4

0 0 1 %. a0
_1 _1 1

$$\begin{aligned} \frac{1}{(\lambda-2)^2(\lambda-3)} &= \frac{-1}{\lambda-2} + \frac{-1}{(\lambda-2)^2} + \frac{1}{\lambda-3} \\ &= \frac{-\lambda+1}{(\lambda-2)^2} + \frac{1}{\lambda-3} \end{aligned}$$

スペクトルを求める

固有値	<pre> char_lf D330 +-+-----+-----+ 1 3 2 2 _12 16 _7 1 +-+-----+-----+ 3 と 2 の重根 </pre>
$A - 2I$	<pre> 2 small_pol D330 0 0 1 _1 1 1 1 _1 0 </pre>
$P_1 = (-A + I)(A - 3I)$ $P_2 = (A - 2I)^2$	$P_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix}$
$S = 2P_1 + 3P_2$	$2P_1 + 3P_2 = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 2 & 0 \\ 1 & -1 & 2 \end{bmatrix}$
$N = (A - 2I)P_1$ $A = S + N$	$(A - 2I)P_1 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

$$A = S + N$$

$$\begin{bmatrix} 2 & 0 & 1 \\ -1 & 3 & 1 \\ 1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 2 & 0 \\ 1 & -1 & 2 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

3.2.3 Reference

笠原 皓司「行列の構造」日本評論社 1994

3.3 2 次形式

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

■Example $x_1^2 + x_2^2 + x_3^2 + 4x_1x_2 + 4x_2x_3 + 4x_1x_3$

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

<pre> b1 1 2 2 2 1 2 2 2 1 </pre>	固有値が良くないので LAPACK で固有ベクトルを求める <pre> require '~addons/math/lapack/lapack.ijs' require '~addons/math/lapack/dgeev.ijs'] ev=. ;{: dgeev_jlapack_ b1 0.816497 _0.57735 0 _0.408248 _0.57735 _0.707107 _0.408248 _0.57735 0.707107 </pre>
<pre> char_lf b1 ++-----+-----+ 1 5 _1 _1 _5 _9 _3 1 ++-----+-----+ f(x) = -5 - 9λ - 3λ² + λ³ = 0 固有値 5, -1(重根) </pre>	対角化 <pre> (: ev) +/ . * b1 +/ . * ev _1 8.88178e_16 1.66533e_16 5.55112e_16 5 1.66533e_16 1.11022e_16 4.44089e_16 _1 </pre>

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -1 \end{bmatrix} \text{ 次の標準形が得られる。}$$

$$x^t A x = -y_1^2 + 5y_2^2 - 1y_3^2$$

3.4 アフィン変換

Affin transform アフィン変換

平行移動、スケーリング（拡大・縮小）、回転、シェアリングを同時に行う。

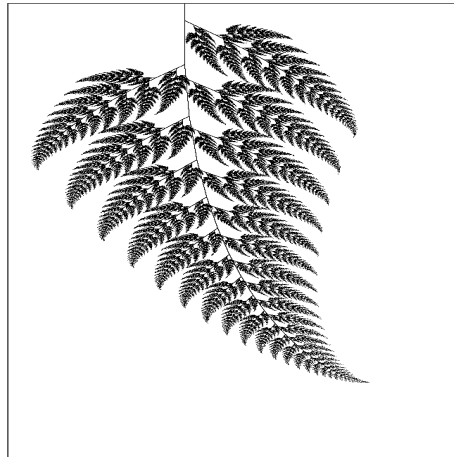


図 3.2 fern

3.4.1 バーンズレーの羊歯/2 次元

次の行列（アフィン変換行列）と並進ベクトルで表現できる。

$$w(x) = w \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Ax + t$$

θ_n は反時計回りの極座標での回転

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 \cos \theta_1 & -r_2 \sin \theta_2 \\ r_1 \sin \theta_1 & r_2 \cos \theta_2 \end{pmatrix}$$

Burnsley のパラメータ

バーンスレーの IFS のパラメータは次のような表に記載されている。

Burnsley's fern

parameter of fern by Burnsray

w	a	b	c	d	e	f	p
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	-0.04	0.85	0	1.6	0.85
3	0.2	-0.26	0.23	0.22	0	1.6	0.07
4	-0.15	0.28	0.26	0.24	0	0.44	0.075

p は点を打つ確率（出現頻度）

<i>parameter</i>	<i>calc</i>
FERN	10 calc_ifs0 FERN
0 0 0 0.16 0 0 0.01	0 1.6
0.85 0.04 _0.04 0.85 0 1.6 0.85	_0.416 1.952
0.2 _0.26 0.23 0.22 0 1.6 0.07	_0.27552 3.27584
_0.15 0.28 0.26 0.24 0 0.44 0.07	_0.103158 4.39548
	0.0881348 5.34029
	0.288526 6.13572
	0.490676 6.80382
	0.689227 7.36362
	0.880388 7.83151
	1.06159 8.22157
	2.1428 2.68919

50000 の点で描画する。

```
view_pixel 50000 calc_ifs0 FERN
```

3.4.2 Script

```
NB. -----section 3.4-----
  calc_ifs0=:4 : 0
NB. Usage: 100 calc_ifs0 FERN
X0=. x
Y0=. (\:({"1 y)){ y NB. sort by prob
AD=. 2 2 $ L:0 { 4{"1 Y0 NB. a b c d/box
EF=. {4 5{"1 Y0 NB. e,f/box
RND=. ( ? 100)% 100 NB. rand
P0=.0.01 - ~ +/ \{"1 Y0 NB. probability/0.01 adjustment origin
ANS=.< TMP=.(<{EF) +(>{AD) +/ . *"(1) 0 0 NB. first
for_ctr. i. X0 do.
  IND=.{.((?100)%100) < P0) # i. # y
  TMP=. (<IND{EF)+ (>IND{AD) +/ . * "1 TMP
ANS=. ANS,<TMP
end.
;"1),. ANS
)
```

```
view_pixel=:3 : 0
gopen ''
```

```
gclear''
glpixel adj_pixel_sub y
)
    adj_pixel_sub=:3 : 0
TMP=. y
MINMAX=. ;("1),.(<./y); >./y
TMP=. TMP + -<./ { . MINMAX
MINMAX=. ;("1),.(<./TMP); >./TMP
TMP * TIMES=. <.<./1000 % {: MINMAX
)
```

3.4.3 Reference

Michael F. Barnsley [Fractals Everywhere 2nd Edition] 1993 Morgan-Kaufmann

第 4 章

マトリクスと回帰分析

4.1 縦長/横長の行列

4.1.1 縦長の行列

未知数の数より条件数が長い縦長の係数行列の場合は誤差ノルムを最小限にする、すなわち最少自乗法の考え方で計算できる。

$Ax = b$	
$(A^t A)x = A^t b$	左から A^t をかける
$x = (A^t A)^{-1} A^t b$	A のベクトルが線形独立ならば $A^t A$ は正則な正方行列
$A^\dagger = (A^t A)^{-1} A^t$	

J は % y で縦長の逆行列をサポートしている。実際は $A^\dagger = (A^t A)^{-1} A^t$ よりも相当複雑なプロセスを経ているようだ。

*1

4.1.2 横長の行列

横長の係数行列の場合は未知数 < 条件数であり、SVD 分解からムーア・ペンローズの一般逆行列を生成する。(SVD 分解の章へ)

4.1.3 応用

最少自乗法/OLS

縦長の行列の逆行列がとれれば、OLS(ordinary reast aquare equation) は正規方程式を求めなくとも次の方法で簡単に求められる。マトリクスを用いれば単相関も重相関も同じ一行のスクリプトで計算できる。

$$\frac{y}{X}$$

定数項を付けるには説明変数のマトリクスXの左に 1 のみの列を付ける。

$$\frac{y}{1 \ X}$$

*1 K.E.Iverson が IBM を離れた後、IBM は Iverson に逆行列のアルゴリズムの書き直 v しを求めた。
(E.Mcdanell のメール)

マトリクスの成分で書けば次のようになる。

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$y_i = \alpha + \beta x_i$$

4.1.4 重回帰

単回帰モデル $Y = \alpha + \beta X_i + u_i$

重回帰モデル $Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + u$

単回帰も重回帰も縦長の逆行列が計算できるので $\frac{y}{X}$ で回帰係数が求められる。

定数項のために前に 1 を付加する場合は $\frac{y}{1, X}$ となる。

スクリプトも次の一行で表すことができる。

```
regx=:3 : '({:"1 y ) %. 1,.}:"1 y ' NB. y %. 1,X
```

データの型は

単回帰、時系列 1 変数の場合は横。2 変数以上は縦型

重回帰 縦型

x, y 説明変数 $x_1, x_2, \dots, x_n$, 従属変数 y の順で y の行を最後に置く

Example

同時方程式の項でも用いる例題を用意する。

D410 リンゴの需給モデル (issue : shirasago)

年	数量	市場価格	購入者の所得	一日平均日照時間
t	Q	P	I	T
	$\times 1000$	yen	man - en	h
0	57	78	28	7
1	55	96	29	4.1
2	66	87	32	7.2
3	65	98	33	5.4
4	71	104	35	5.8
5	74	105	36	6.7
6	71	110	36	5
7	71	113	38	6.3

最初に相関行列を取ってみる。

```
cortable D410
```

```
1 0.782326 0.977432 0.271114
0.782326      1 0.890509 _0.371396
0.977432 0.890509      1 0.0674221
0.271114 _0.371396 0.0674221      1
```

需要	供給
<pre>1 2 0{"1 D410 P I Q=Y 78 28 57 96 29 55 87 32 66 98 33 65 104 35 71 105 36 74 110 36 71 113 38 77</pre>	<pre>1 3 0{"1 D410 P T Q=y 78 7 57 96 4.1 55 87 7.2 66 98 5.4 65 104 5.8 71 105 6.7 74 110 5 71 113 6.3 77</pre>
<pre>reg0 1 2 0{"1 D410 _5.16659 _0.282913 3.00044 y = -5.16659 - 0.282913P + 3.00044I</pre>	<pre>reg0 1 3 0{"1 D410 _28.6958 0.68098 4.77707 y = -28.6958 + 0.68098P + 4.77707Y</pre>

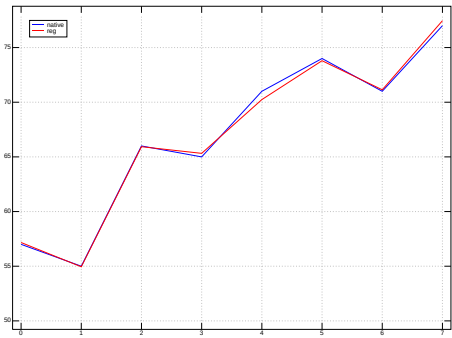


図 4.1 mreg

推定	X 1 は定数項用	X に回帰係数をかけると推計値が求められる (供給モデル)
	1, . 1 3{"1 D410	
	X1 X2	
	1 78 7	Q estim^Q
	1 96 4.1	57 56.7785
	1 87 7.2	55 54.6865
	1 98 5.4	66 66.234
	1 104 5.8	65 66.1224
	1 105 6.7	71 70.4258
	1 110 5	74 73.1433
	1 113 6.3	71 71.7287
		77 76.8809

({"1 D410),. estim_reg0 1 2 0{"1 D410

AIC による変数選択

重回帰の場合には変数の数を増やしていくと（重）相関係数が上がっていくという厄介な問題を抱えている。また、データが派閥を作り、多重共線性を生じることがある。このような場合に AIC の値が小さい方のモデルを選択すれば良い。

AIC（情報量規準/赤池情報量規準）は統計数理研究所の所長であった赤池弘治により開発された方程式の最尤選択を示すクライテリアであり、制御工学から経済まで広く使われている。幾つかのモデルの AIC の値を比較し、数字が一番小さいモデルが尤も優れていると判定する。クライテリアなので一つのモデルの数値自体の大きさには特段の意味はない。

*2

重回帰モデルの AIC は次により求める。

$$MLL = -\frac{1}{2} \log \frac{Q}{n}$$

$$AIC = n \log \frac{Q}{n} + 2 \times (k + 1)$$

Q は残差平方和

k + 1 は推定したパラメータの個数

需要関数

供給関数

```

reg0 reg_exam_ad 1 2 0 {"1 D410      reg0 reg_exam_ad 1 3 0 {"1 D410
+-----+-----+-----+ +-----+-----+-----+
|f=      |_5.16659 _0.282913 3.00044| |f=      |_28.6958 0.68098 4.77707|
+-----+-----+-----+ +-----+-----+-----+
|corr=:|99.2859                      | |corr=:|97.798                      |
+-----+-----+-----+ +-----+-----+-----+
|AIC:   |_1.66005                      | |AIC:   |7.34814                      |
+-----+-----+-----+ +-----+-----+-----+
|DW=    |2.3734                      | |DW=    |1.8483                      |
+-----+-----+-----+ +-----+-----+-----+
|t=:    |_1.84266 _5.12296 16.3289 | |t=:    |_4.38366 14.3311 9.1157 |
+-----+-----+-----+ +-----+-----+-----+

```

もう一本 3 変数のモデルも作ってみよう。

```

reg0 reg_exam_ad 1 2 3 0 {"1 D410
+-----+-----+-----+
|f=      |_13.8241 0.0284083 2.04448 1.64728|
+-----+-----+-----+
|corr=:|99.7715                      |
+-----+-----+-----+
|AIC:   |_8.77759                      |
+-----+-----+-----+
|DW=    |1.93168                      |
+-----+-----+-----+
|t=:    |_3.99759 0.252904 5.8781 2.91603 |
+-----+-----+-----+

```

AIC は最後 3 の説明変数のモデルが一番小さく最良のモデルであるが、変数 P の t 値が小さいので 1 のモデルも選択の対称とする。

4.2 キングの法則/多項式

穀物の欠乏と市況に関して、18世紀の経済学者グレゴリー・キング (1648-1712 イングランド) のキングの法則が知られている。

George King's Law, is

An estimate of by how much a deficiency in the supply of corn will raise the price of corn.

減収率	高騰率
Δ10%	30%
Δ20%	80%
Δ30%	160%
Δ40%	280%
Δ50%	450%

```

KING
10 30
20 80
30 160
40 280
50 450

'line,marker' plot {}: KING
pd 'eps temp/king0.eps'
```

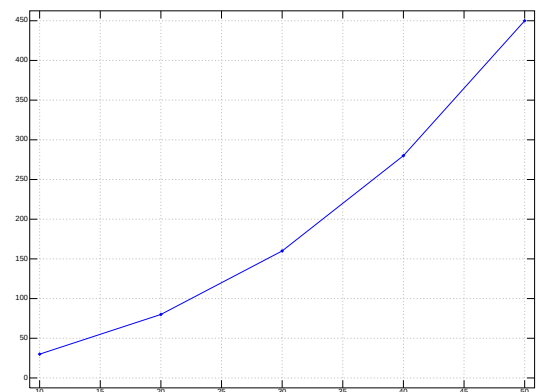


図 4.2 kings rule

穀物や油、鉱物資源など需給が市況に左右される商品の価格変動の大きさはキングの時代と余り変わっていない。

4.2.1 多項式回帰

OLS をより一般的な曲線

$$y_i = c_0 + c_1 x_1 + c_2 x_2^2 + \dots$$

にフィットさせる。以下の行列方程式を用いる。

*3

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots \\ 1 & x_2 & x_2^2 & \cdots \\ \vdots & \vdots & \vdots & \cdots \\ 1 & x_n & x_n^2 & \cdots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

多項式回帰は難しく考えなければ重回帰の少し変形したものである。

*3 左辺の行列は Vandermonde Matrix である。

4.2.2 Kings rule を計算する

```
2 estim_poly KING
30 32
80 76
160 160
280 284
450 448
```

```
3 estim_poly KING
30 30
80 80
160 160
280 280
450 450
```

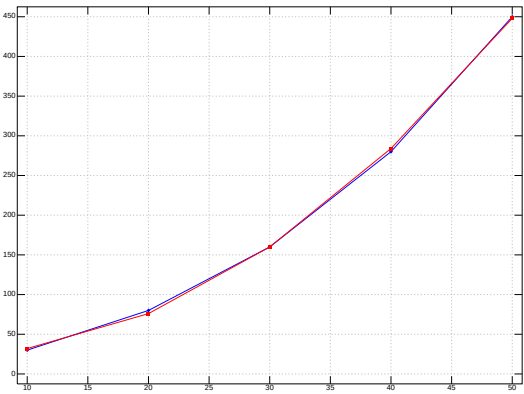


図 4.3 speed and breaki

この範囲では 3 項の多項式でぴたりと一致する

$$f = 2.33333x + 0.05x^2 + 0.00166667x^3$$

KING		10 20 30 40 50 ^/ 0 1 2
10	30	1 10 100
20	80	1 20 400
30	160	1 30 900
40	280	1 40 1600
50	450	1 50 2500
		30 80 160 280 450 %.
		10 20 30 40 50 ^/ 0 1 2
		28 _1.6 0.2

<pre>2 poly2 KING 28 _1.6 0.2 3 poly2 KING 3.95062e_11 2.33333 0.05 0.00166667</pre>	$f = 28 - 1.6x + 0.2x^2$ $f = 2.33333x + 0.05x^2 + 0.00166667x^3$
---	---

4.2.3 マイナス側へ拡大

マイナス側にも拡大してみよう。減産と増産と仮定する。^{*4}

```
3 poly2 a1
7.10543e_15 3.16861 0 0.00234654
```

$$y = 3.16861x + 0.00234654x_3$$

```
({"1 a1"),. 3 estim_poly a1
_50 _450 _451.748
_40 _280 _276.923
_30 _160 _158.415
_20 _80 _82.1445
_10 _30 _34.0326
0 0 7.10543e_15
10 30 34.0326
20 80 82.1445
30 160 158.415
40 280 276.923
50 450 451.748
```

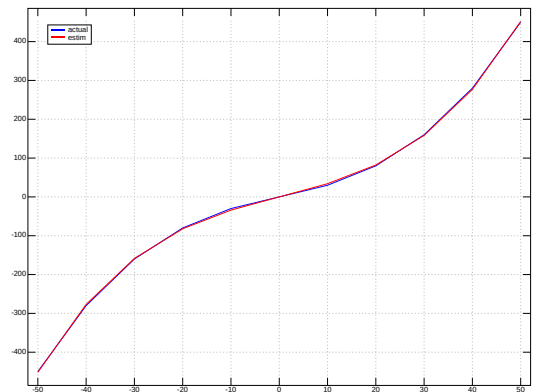


図 4.4 kings law extend to minus

比較のために単回帰を行ってみる。多項式では3次の項がなめらかに追従している。

```
({"1 a1")%. 1, {"1 a1
7.10543e_15 7.3445

y = 7.3445x
```

AIC

多項式の AIC は

$$MLL = -\frac{n}{2} \log \frac{Q}{n}$$

$$AIC = -2 \times MLL + 2 \times (k + 1)$$

4.2.4 Script

NB. polynomial

^{*4} キングの法則は対称性については触れていない。

```
({"1 a1),. +/"1 (({"1 a1)% 1,. {"1 a1)&p.1,. {"1 a1
```

```
_50 _450 _359.927
_40 _280 _286.473
_30 _160 _213.018
_20 _80 _139.564
_10 _30 _66.1091
 0 0 7.34545
10 30 80.8
20 80 154.255
30 160 227.709
40 280 301.164
50 450 374.618
```

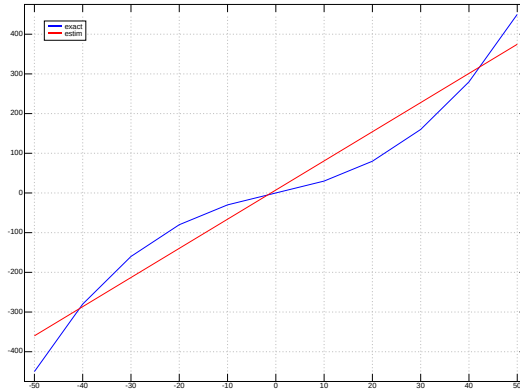


図 4.5 kings law extend to minus

```
({"1 a1)% 1,. {"1 a1
7.10543e_15 7.34545
```

NB. Usage: e.g. 3 reg_poly y

```
reg_poly=: 4 : 0
```

```
if. 1= +/ * $ y. do. x. poly1 y.
```

```
else. x. poly2 y. end.
```

```
)
```

NB. -----polynomial regression-----

```
poly1=: 4 : 'y. %. (>i. # y.) ^/ i.>: x.'
```

```
poly2=: 4 : '({"1 y.) %. ({."1 y.) ^/ ("1 i.>: x.'
```

```
estim_poly=: 4 : 0
```

NB. single type/ double type is same

NB. 2 estim_poly KING

```
if. 1=+/$ y do.Y0=. y. ,. ((>i. # y.) ^/ i.>: x.) +/ . * x. poly1 y.
```

```
else. Y0=. ({."1 y.),. ((;"1 y.) ^/ i.>: x.) +/ . * x. poly2 y.
```

```
end.
```

```
Y0
```

```
)
```

NB. modified estim_poly 5/Dec./2007

4.2.5 Reference

Meyberk/Vachenauer 及川正行訳 工科系の数学 (4) 多変数の微積分サイエンス社
1996

4.3 時系列回帰

4.3.1 自己相関

時系列のアウトラインを掴むために次のような小さなデータを用意する。時間の流れは → であり、後ろの方がより新しい。

D430=:3 6 8 4 4 8

ある確率過程が定常であるならば、時間の経過での変化の関係をとらえることによって、時間領域での性質を要約することができる。

時系列 y_n と時刻を k だけシフトした y_{n-k} を時系列 y_n の自己共分散関数 (autocovariance function) といい、時間が y_{n-k} だけ経過したとき、確率変数がどのように経過するかを示している。

自己相関係数 (autocorrelation) は、標準偏差で規格化された自己共分散であり、自己共分散、自己相関関数は時系列の過去の変動との関連の強さを表現したものである。

ここでは自己共分散関数と自己相関関数を求める。

$$ACF(k) = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

定常過程の場合、 $X(t)$ の期待値は、時刻 t に依存しない。自己共分散 $Cov[X(t_1), X(t_2)]$ も時刻 t_1, t_2 には依存しないで、その時間差 $\tau = t_1 - t_2$ のみに依存する。

順を追って書き下してみよう。

経過と説明

prefix, suffix 最初に例題 D430 のすべての組み合わせを作成する。J の *prefix, suffix* の機能を存分に活用している

```
(<\. D430), . | . (<\D430
```

```
<\D430 NB. prefix
```

```
+--+---+-----+-----+-----+-----+
|3|3 6|3 6 8|3 6 8 4|3 6 8 4 4|3 6 8 4 4 8|
+--+---+-----+-----+-----+-----+
```

```
<\.D430 NB. suffix
```

```
+-----+-----+-----+-----+-----+
|3 6 8 4 4 8|6 8 4 4 8|8 4 4 8|4 4 8|4 8|8|
+-----+-----+-----+-----+-----+
```

prefix を逆さにして双方を縦に連結すると次のテーブルができる

時間 t は新しいほうが右にくる。右の欄は上から 原型、 $Y_t, Y_{t-1}, Y_{t-2}, Y_{t-3}...$ と新しい順に落としている。左の欄は頭の方から古い時間を順に落し、右の欄とサイズを合わせた組み合わせを作成する。

```
+-----+-----+
|3 6 8 4 4 8|3 6 8 4 4 8|
```

```

+-----+-----+
| 6 8 4 4 8 | 3 6 8 4 4 |
+-----+-----+
| 8 4 4 8   | 3 6 8 4   |
+-----+-----+
| 4 4 8     | 3 6 8     |
+-----+-----+
| 4 8       | 3 6       |
+-----+-----+
| 8         | 3         |
+-----+-----+

```

平均を引く 各要素から平均を引く。ACF では $\bar{Y}$ に原系列の平均を用いる。

```

tmp=: ((<\.D430),.|. @(<\ ) D430)- L:0 (+/%#) D430
+-----+-----+
|_2.5 0.5 2.5 _1.5 _1.5 2.5|_2.5 0.5 2.5 _1.5 _1.5 2.5|
+-----+-----+
|0.5 2.5 _1.5 _1.5 2.5   |_2.5 0.5 2.5 _1.5 _1.5   |
+-----+-----+
|2.5 _1.5 _1.5 2.5       |_2.5 0.5 2.5 _1.5       |
+-----+-----+
|_1.5 _1.5 2.5           |_2.5 0.5 2.5           |
+-----+-----+
|_1.5 2.5                |_2.5 0.5                |
+-----+-----+
|2.5                     |_2.5                     |
+-----+-----+

```

左右の要素を掛け合わせる 各行毎に左右の要素を掛け合わせる。

```

*/ L:0 ,. ({."1 tmp) ,: L:0 {:"1 tmp
+-----+
|6.25 0.25 6.25 2.25 2.25 6.25|
+-----+
|_1.25 1.25 _3.75 2.25 _3.75   |
+-----+
|_6.25 _0.75 _3.75 _3.75      |
+-----+
|3.75 _0.75 6.25              |
+-----+
|3.75 1.25                    |

```

```

+-----+
|_6.25      |
+-----+

```

行の合計=自己共分散 各行 (次数) ごとに合計すると自己共分散が求まる

```

;+ / L:0 TMP1
23.5 _5.25 _14.5 9.25 5 _6.25

```

分散で割る=自己相関係数 分散で除算すると自己相関係数が得られる

```

TMP1 % (+/)@:( ^&2)@ dev y'
NB. 自己相関
1 _0.223404 _0.617021 0.393617 0.212766 _0.265957

```

ACF

```

acf D430
NB. Cov % Var
1 _0.223404 _0.617021 0.393617 0.212766 _0.265957

```

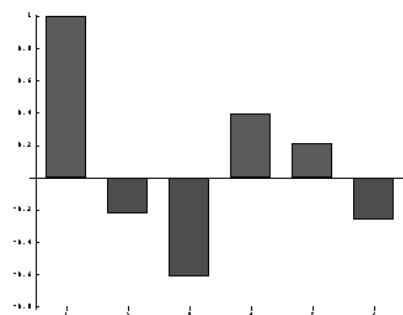
$$\begin{bmatrix} A_t A_t & A_t A_{t-1} & A_t A_{t-2} & A_t A_{t-3} & A_t A_{t-4} & A_t A_{t-5} \\ 1 & -0.223404 & -0.617021 & 0.393617 & 0.212766 & -0.265957 \end{bmatrix}$$


図 4.6 ACF

Script

次のスクリプトは, 自己相関関数を一度に求めてしまうものである。

$$ACF(k) = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

SCRIPT

```

acf=:3 : 0
NB. refine ACF2 code
NB. ACF autocorellation coefficients (many times at once)
C1=. ( <\. y ) ,. |.@(<\)y
C3=. C1 - L:0 C2=. mean y NB. dev
C4=.> + / L:0 */ L:0 <"2 >C3 NB. change box style

```



```
NB. sum (Y_t-Y')*(Y_(t-k) - Y')
C4 % +/ (y - mean y )^2      NB. C4 % VAR y
)
```

4.3.2 Autoregression 自己回帰

時系列モデルを用いると時系列の特徴を簡潔に表現できる。ここでは自己回帰モデル (autoregressive model, AR) を当てはめる問題を取り扱う。

AR の解法は Yule-Walker 法, バーク法, ハウスホルダー法などがある。

ここでは、Yule – Walker 法を紹介する。

最もポピュラーな Yule-Walker 法は ACF から正規方程式を求める方法である。

Yull-Walker 方程式

$$C_M a_M = c_M$$

ACF

$$c(k) = \frac{1}{N} \sum_{s=1}^{N-k} x(s)x(s+k), (k = 0, 1, 2, \dots, M)$$

Yull-Walker 行列

$$C_M = \begin{bmatrix} c(0) & c(1) & \dots & c(M-2) & c(M-1) \\ c(1) & c(0) & c(1) & \dots & c(M-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c(M-2) & \dots & c(1) & c(0) & c(1) \\ c(M-1) & c(M-2) & \dots & c(1) & c(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_p \end{bmatrix}$$

$$c_M = (c(1), c(2), \dots, c(M))^T$$

自己相関 ACF(k) を用いて C_M を作れば、この方程式はクラメール法で簡単に解くことができ Yull-Walker 法はこの yull-walker 方程式を直接解くことにより AR 係数の推定値を求める。

経過と解説

ACF 最初に ACF を求める。

```
acf a=: 3 6 8 4 4 8
```

```
1 _0.223404 _0.617021 0.393617 0.212766 _0.265957
```

Yull-Walker 行列 指定次数の Yull-Walker 行列を作成する

Yull-walker 方程式 acf の結果を 3 次の C_M に、右に c_M を連結したもの

$$\begin{bmatrix} c(0) & c(1) & c(2) \\ c(1) & c(0) & c(1) \\ c(2) & c(1) & c(0) \end{bmatrix} \begin{bmatrix} c(1) \\ c(2) \\ c(3) \end{bmatrix}$$

3 yw a NB. YW 行列			
	1	_0.223404	_0.617021
_0.223404		1	_0.223404
_0.617021	_0.223404		1
			0.393617

クラメル法で解く 上の YW 行列をクラメル法で解いたもの。右端の列が解

```
cr=: %."1      NB. Cramer Method
```

```
3 yw a
      1 _8.88178e_16 6.66134e_16 _0.37704
_2.22045e_16      1 4.44089e_16 _0.70024
_6.66134e_16 _8.32667e_16      1 0.00453847
```

自己回帰式 $y(t) = -0.37704x_{t-1} - 0.70024x_{t-2} + 0.00453847x_{t-3}$

Script

```
yw_sub0=:4 : 0
NB. Yull- Waker sub
NB. make C(m) Matrix
NB. for univariate AR
T1=: (|.}. M) , M=: y.      NB. reverce and connect
T2=:> x. # < T1           NB. copy x. vertical
T3=: ({. $ T2), <: {: $ T2 NB. decrease $ to (0 _1)
T4=:(-x.) {. ("1) T3 $ , T2 NB. take backward (_x.) column
)
```

```
yw=:4 : 0
NB. Yule Walker Method(Univariate)
NB. Data type is Yoko list
NB. X(t)=X1(t-1)+ X2(t-2) + X3(t-3).....
NB. Usage: x. jisuu // y. Data
Y1=: acf y.
Y3=:}. Y2=: (>: x.) {. Y1
Y4=: (x. yw_sub0 Y2),. Y3
Y5=: cr Y4      NB. cramer decomposition
)
```

NB. 上の部分解を求める

```
y4=: 4 : '(x. yw_sub0 Y2),. }. Y2=:(>: x.) {. acf y.'
```

Yull-Walker 法で当てはめたモデルは, 定常性をもつ。

時間を逆転させた, 過去に向かっての推計が可能なことも, 時系列の特色である。

4.3.3 Reference

Stephen A. DeLurgio [Forecasting Principles and Applications] McGRAW-HILL 1998

4.4 線形同時方程式

同時方程式は多入力、多出力の連立方程式システムが特色である。1950年にコールズ委員会でのクライン作成のモデル1が計量経済学の船出となったと言われる。

4.4.1 小さなアナログモデルから始めよう

■Working Example リンゴのモデル

リンゴの数量、市場価格、購買者の所得、日照時間から需要関数、供給関数を求める。些か無謀なモデルであるが、モデルを求める手順の好例であるので検討してみよう。

構造型方程式	$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$ $Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$	
「内生変数」	Q_t = リンゴの数量 P_t = リンゴの市場価格	連立方程式の体系の中で相互依存により決定される変数
「外生変数」	Y_t = 購買者の所得 T_t = 日照時間	

最初の Example のデータは重相関で用いた D410

(データの出典:白砂堤津耶「例題で学ぶ・初歩からの計量経済学」1998/2007 日本評論社)

構造型方程式から誘導型方程式を導く

内生変数を含む式は、Haavelmo^{*5} バイアスと呼ばれる歪みを持ち、不偏性と一致性を持たない。そこで、外生変数と先決変数とで構成される**誘導型方程式**を導き、次いで、**構造型方程式**に戻すと、一致性は確保できる。誘導型を求めるため例題の数式を展開すると次のようになる。

$$\alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t} = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$$

$$\alpha_1 P_t - \beta_1 P_t = -\alpha_0 - \alpha_2 Y_t + u_{1t} + \beta_0 + \beta_2 T_t + u_{2t}$$

$$P_t(\alpha_1 - \beta_1) = -\alpha_0 + \beta_0 - \alpha_2 Y_t + \beta_2 T_t - u_{1t} + u_{2t}$$

$$P_t = \frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 Y_t}{\alpha_1 - \beta_1} + \frac{\beta_2 T_t}{\alpha_1 - \beta_1} + \frac{-u_{1t} + u_{2t}}{\alpha_1 - \beta_1}$$

P_t を次式に代入する。

$$\begin{aligned} Q_t &= \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t} \\ &= \alpha_0 + \alpha_1 \left(\frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} + \frac{-\alpha_2 Y_t}{\alpha_1 - \beta_1} + \frac{\beta_2 T_t}{\alpha_1 - \beta_1} + \frac{-u_{1t} + u_{2t}}{\alpha_1 - \beta_1} \right) + \alpha_2 Y_t + u_{1t} \\ &= \alpha_0 + \frac{\alpha_1(-\alpha_0 + \beta_0)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-\alpha_2 Y_t)}{\alpha_1 - \beta_1} + \frac{\alpha_1(\beta_2 T_t)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-u_{1t} + u_{2t})}{\alpha_1 - \beta_1} + \alpha_2 Y_t + u_{1t} \end{aligned}$$

^{*5} Trigve M.Haavelmo 1911-1999 Norway 1989 ノーベル記念賞

$$\begin{aligned}
&= \frac{\alpha_0(\alpha_1 - \beta_1) + \alpha_1(-\alpha_0 + \beta_0)}{\alpha_1 - \beta_1} + \frac{\alpha_1(-\alpha_2 Y_t) + \alpha_2 Y_t(\alpha_1 - \beta_1)}{\alpha_1 - \beta_1} \\
&+ \frac{\alpha_1(\beta_2 T_t)}{\alpha_1 - \beta_1} + \frac{(-\alpha_1 u_{1t} + \alpha_1 u_{2t}) + (\alpha_1 u_{1t} - \beta_1 u_{1t})}{\alpha_1 - \beta_1} \\
&= \frac{\alpha_1 \beta_0 + \alpha_0 \beta_1}{\alpha_1 - \beta_1} + \frac{-\alpha_2 \beta_1 Y_t}{\alpha_1 - \beta_1} + \frac{\alpha_1 \beta_2 T_t}{\alpha_1 - \beta_1} + \frac{\alpha_1 u_{2t} - \beta_1 u_{1t}}{\alpha_1 - \beta_1}
\end{aligned}$$

誘導型パラメータを次のように置くと下表右欄の誘導方程式を導くことができる

$$\begin{aligned}
\pi_{10} &= \frac{\alpha_1 \beta_0 - \alpha_0 \beta_1}{\alpha_1 - \beta_1} & \pi_{11} &= \frac{-\alpha_2 \beta_1}{\alpha_1 - \beta_1} & \pi_{12} &= \frac{\alpha_1 \beta_2}{\alpha_1 - \beta_1} \\
\pi_{20} &= \frac{-\alpha_0 + \beta_0}{\alpha_1 - \beta_1} & \pi_{21} &= \frac{-\alpha_2}{\alpha_1 - \beta_1} & \pi_{22} &= \frac{\beta_2}{\alpha_1 - \beta_1} \\
u_{1t} &= \frac{-u_{t1} + u_{2t}}{\alpha_1 - \beta_1} & u_{2t} &= \frac{\alpha_1 u_{2t} - \beta_1 u_{1t}}{\alpha_1 - \beta_1}
\end{aligned}$$

構造型方程式	誘導型方程式
$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$	$Q_t = \pi_{10} + \pi_{11} Y_t + \pi_{12} T_t + u_{1t}$
$Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$	$P_t = \pi_{20} + \pi_{21} Y_t + \pi_{22} T_t + u_{2t}$

内生変数と外生変数の区分は微妙であって判然とした区分は存在しない。モデルの中の変数が「自立的に、そしてモデル内の他の変数とは独立に変動すると期待できるかどうか」*6を一応の目安として、行列表現で右の項に記述する外生変数と先決内生変数を先決変数とする。内生変数は左欄の推計対象の変数である。

4.4.2 線形同時方程式の行列表現

構造型方程式

$$Q_t = \alpha_0 + \alpha_1 P_t + \alpha_2 Y_t + u_{1t}$$

$$Q_t = \beta_0 + \beta_1 P_t + \beta_2 T_t + u_{2t}$$

「内生変数」

Q_t =リンゴの数量

P_t =リンゴの市場価格

構造型方程式の行列型への整理

$$Q_t - \alpha_1 P_t = \alpha_0 + \alpha_2 Y_t + u_{1t} \quad (\text{第1式})$$

$$Q_t - \beta_1 P_t = \beta_0 + \beta_2 T_t + u_{2t} \quad (\text{第2式})$$

「外生変数」

Y_t = 購買者の所得

T_t =日照時間

構造型の行列表現

内生変数を左に集め、右の項は外生変数と先決内生変数 (X_{t-1} 型など) で記述する。

この区分は、大きな水槽の中を浸透膜や浸透壁で分けするようなもので、変数自体は相互に行き来できなくなるが、パラメーターは透過して、行き来できる。*7

*6 W.H.Green 計量経済分析 II(2000)p835

*7 モデルによっては、元気な金魚が上を飛び越して往来している場合も見受けられる。

第1式

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 \\ -\alpha_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_2 \\ 0 \end{bmatrix} + u_{1t}$$

第2式

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 \\ -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \beta_0 \\ 0 \\ \beta_2 \end{bmatrix} + u_{2t}$$

全体を行列で表す

$$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix} + (u_1 + u_2)_t$$

行列の成分を表すと

$$\begin{bmatrix} Q_1 & P_1 \\ Q_2 & P_2 \\ \vdots & \vdots \\ Q_n & P_n \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix} = \begin{bmatrix} 1 & Y_1 & T_1 \\ 1 & Y_2 & T_2 \\ \vdots & \vdots & \vdots \\ 1 & Y_n & T_n \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix} + (u_1 + u_2)_t$$

構造型を記号で表現すると

$$YB = Z\Gamma + E$$

誘導型の行列表現

行列表現では、誘導型は逆行列を用いて簡潔に表せる。

$$Y = Z\Gamma B^{-1} + EB^{-1}$$

これは、方程式体系の同時決定部分を解いたものである。誤差項を除くと次のようになる。

$$Y = Z\Gamma B^{-1}$$

$$Y = \frac{Z\Gamma}{B}$$

Jの高度な配列計算機能を用いると簡単に計算できる。

Bのパラメータは、BとΓの作成の時に定義式や明示的に移項した項の場合は数式通りとする。Γの定義式は数式通りとする。それ以外のOLSのパラメータは成り行きとする。

$\begin{bmatrix} Q_t & P_t \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\alpha_1 & -\beta_1 \end{bmatrix}$	$\begin{bmatrix} 1 & Y_t & T_t \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix}$
<pre>b=.shira_sub20 D410 1 0.295458 1 _0.569307</pre>	<pre>g=.shira_sub30 D410 4.62269 2.72182 0 _14.0737 0 4.04775</pre>

■経過と説明 B と Γ を作成する。この B と Γ の定義でモデルの構成が決まる。モデルの性格や精度を検討し、識別テストをクリアするモデルを、重相関に時系列や A/B 型などを組み合わせて設計する。

内生変数が B 、説明変数（外生変数、先決内生変数）が Γ になる。

$$\begin{bmatrix} 1 & -\alpha_1 \\ 1 & -\beta_1 \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ \beta_0 & 0 & \beta_2 \end{bmatrix}$$

最初に OLS で構造型のパラメータを求め、行列に組み込む。

OLS で構造型のパラメータを求める (重回帰で求めたものと同じ) $Q = \alpha_0 + \alpha_1 P + \alpha_2 Y$ $Q = \beta_0 + \beta_1 P + \beta_2 T$	<pre>shira_sub0 D410 _5.16659 _0.282913 3.00044 _28.6958 0.68098 4.77707 Q = -5.16659 - 0.282913P + 3.00044Y Q = -28.6958 + 0.68098P4.77707T</pre>	各変数の単独の OLS
B を作成 $\begin{bmatrix} 1 & -\alpha_1 \\ 1 & -\beta_1 \end{bmatrix}$	<pre>b=.shira_sub2 D410 1 0.282913 1 _0.68098</pre>	
B^{-1} B の逆行列 $\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} -\beta_1 & \alpha_1 \\ -1 & 1 \end{bmatrix}$	<pre>%. shira_sub2 D410 0.706489 0.293511 1.03746 _1.03746</pre>	1 の部分にもウェイトがかかる。

Γ を作成 $\begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ \beta_0 & 0 & \beta_2 \end{bmatrix}$	<pre>g=.shira_sub3 D410 shira_sub3 D410 _5.16659 3.00044 0 _28.6958 0 4.77707</pre>	外生変数と先決 内生変数 Z のパ ラメーター部分で ある。 行の数は Z の変 数の数で、列の数 はモデルの式の数 で決まる。
$B^{-1}\Gamma$ システムの回帰式 (誘導型)	<pre>(%. b) +/ . * g _12.0727 2.11978 1.40212 24.4106 3.11283 _4.95602 -12.0727 + 2.11978Y + 1.40212T 24.4106 + 3.11283Y - 4.95602T</pre>	
$\hat{Q}$ 推計値	先決変数に回帰式をかける <pre>(1,. 2 3 {"1 D410) +/ . * : (% . b) +/ . * g 57.096 76.8778 55.1496 94.363 65.8555 88.3379 65.4514 100.372 70.2518 104.615 73.6335 103.267 71.2499 111.692 77.3122 111.475</pre>	一行でモデルの計 算ができる。これ は全体テストであ る

構造型 $YB = Z\Gamma$ から誘導型 $Y = \frac{Z\Gamma}{B}$ に変換する。このとき、 B の逆行列 B^{-1} 即ち内生変数に移った OLS のパラメーターの逆行列が先決変数に残った OLS のパラメーターに作用して同時方程式システムのパラメーターを決定する。

この白砂モデルは適正識別である。 B と Γ のサイズの差は 1 でこれは Γ の OLS の定数項部分である。

$$YB = Z\Gamma \rightarrow Y = \frac{Z\Gamma}{B}$$

この一行で同時方程式は計算できる。

shirasago_model D410

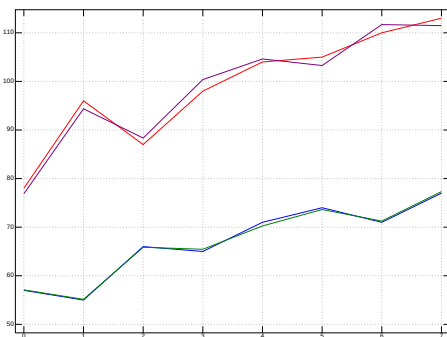


図 4.7 Q P

shirasago_model D410

shirasago_model D410			
+-----+-----+-----+			
57	78	57.096	76.8778
55	96	55.1496	94.363
66	87	65.8555	88.3379
65	98	65.4514	100.372
71	104	70.2518	104.615
74	105	73.6335	103.267
71	110	71.2499	111.692
77	113	77.3122	111.475
+-----+-----+-----+			

行列表現の構造

このような小さいモデルでも誘導型を導いてパラメータを構造型で推計するのは大変な作業である。行列を用いるからには 2 段階最少自乗法などのプロセスも省略してしまいたい。そこで、

$$YB = Z\Gamma \text{ から } Y = \frac{Z\Gamma}{B} \text{ であるので}$$

$Z\Gamma$ を B で割れば Y の推計値 ($\hat{Y}$) を求められるが、マトリクスのサイズが異なる場合には $Z\Gamma \cdot B^{-1}$ は直接計算できないので先に逆行列 B^{-1} を求めて、左から Γ と内積演算を行う。

このプロセスと誘導型のパラメータの比較を行ってみる。

B $\begin{bmatrix} 1 & -\alpha_1 \\ 1 & -\beta_1 \end{bmatrix}$ b $1 \ 0.282913$ $1 \ -0.68098$	逆行列 B^{-1} $\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} -\beta_1 & \alpha_1 \\ -1 & 1 \end{bmatrix}$ $\%.$ b $0.706489 \ 0.293511$ $1.03746 \ -1.03746$
内積計算では右パラメーターは (自動で) 転置され、列が (同じサイズの) 左パラメーターに掛け合わされる。	$\frac{1}{\alpha_1 - \beta_1} \begin{bmatrix} \beta_1 & -\alpha_1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha_0 & \alpha_2 & 0 \\ \beta_0 & 0 & \beta_2 \end{bmatrix}$ 最終的にはこの形で Γ と行 $\times$ 行で掛け合わされる。 $\begin{aligned} (-\beta_1, +\alpha_1) \times (\alpha_0, \beta_0) &= -\alpha_0\beta_1 + \alpha_1\beta_0 = \pi_{10} \\ (-\beta_1, +\alpha_1) \times (\alpha_2, 0) &= -\alpha_2\beta_1 = \pi_{11} \\ (-\beta_1, +\alpha_1) \times (0, \beta_2) &= \alpha_1\beta_2 = \pi_{12} \\ (-1, +1) \times (\alpha_0, \beta_0) &= -\alpha_0 + \beta_0 = \pi_{20} \\ (-1, +1) \times (\alpha_2, 0) &= -\alpha_2 = \pi_{21} \\ (-1, +1) \times (0, \beta_2) &= \beta_2 = \pi_{22} \end{aligned}$ ($\frac{1}{\alpha_1 - \beta_1}$ の記述をを省略した。) 誘導型のパラメーターを推計している。

4.4.3 Klein Model

Klein(1950) の有名なマクロ経済のモデルである。

このクラインのモデル 1 は僅か 8 本の方程式で構成され、1921 年から 40 年にかけての
大不況の経緯をものの見事にトレースし、これを持って、経済学は一人前の科学として
船出を遂げたと言われる。

*8

Lawrence Robert Klein, (1920-) USA

1920 年ネブラスカ州オマハに生まれる。UC Berkley で経済学と数学を学び、MIT でサ
ミュエルソンの下で博士号を得る。シカゴ大学におかれていたコールズ委員会で計量モデ
ルを作成。マッカーシー旋風を避けてヨーロッパに渡って Oxford でイギリスの経済モデ
ルの作成に従事。1958 年に帰国しペンシルバニア大学教授に就任。1980 年にノーベル記
念経済学賞を受賞

(消費)	$C_t = \alpha_0 + \alpha_1 P_t + \alpha_2 P_{t-1} + \alpha_3 (WP_t + WG_t) + e_{1t}$
(投資)	$I_t = \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \beta_3 K_{t-1} + e_{2t}$
(民間賃金)	$WP_t = \gamma_0 + \gamma_1 X_t + \gamma_2 X_{t-1} + \gamma_3 A_t + e_{3t}$
(均衡需要)	$X_t = C_t + I_t + G_t$
(民間利潤)	$P_t = X_t - T_t - WP_t$

*8 佐和 隆光 計量経済学への誘い 大蔵省財政金融研究所 「ファイナンシャルレビュー」 June4-1992

(資本ストック) $K_t = K_{t-1} + I_t$

外生変数	G	政府非賃金支出
	T	間接法人税 + 純輸出
	WG	政府賃金支出
先決変数	A	1930 年から測ったトレンド
	K	ラグ付き資本ストック
	P	民間利潤
	X	総需要
内生変数		各式の左に置いてある

内生変数を左に送って整理する。

$$\begin{array}{lcl}
 1 & C_t - \alpha_1 P_t - \alpha_3 WP_t & = \alpha_0 + \alpha_2 P_{t-1} + \alpha_3 WG_t + e_{1t} \\
 2 & I_t - \beta_1 P_t & = \beta_0 + \beta_2 P_{t-1} + \beta_3 K_{t-1} + e_{2t} \\
 3 & WP_t - \gamma_1 X_t & = \gamma_0 + \gamma_2 X_{t-1} + \gamma_3 A_t + e_{3t} \\
 4 & X_t - C_t - I_t & = G_t \\
 5 & P_t - X_t + WP_t & = -T_t \\
 6 & K_t - I_t & = K_{t-1}
 \end{array}$$

行列型で表現する。(上のナンバーは式の番号)

$$B = \begin{array}{c|cccccc}
 & C_t & I_t & WP_t & X_t & P_t & K_t \\
 \hline
 C_t & 1 & 0 & -\alpha_3 & 0 & -\alpha_1 & 0 \\
 I_t & 0 & 1 & 0 & 0 & -\beta_1 & 0 \\
 WP_t & 0 & 0 & 1 & -\gamma_1 & 0 & 0 \\
 X_t & -1 & -1 & 0 & 1 & 0 & 0 \\
 P_t & 0 & 0 & 1 & -1 & 1 & 0 \\
 K_t & 0 & -1 & 0 & 0 & 0 & 1
 \end{array}$$

$$\Gamma = \begin{array}{c|cccccccc}
 & 1 & P_{t-1} & WG_t & K_{t-1} & X_{t-1} & A_t & G_t & T_t \\
 \hline
 C_t & \alpha_0 & \alpha_2 & \alpha_3 & 0 & 0 & 0 & 0 & 0 \\
 I_t & \beta_0 & \beta_2 & 0 & \beta_3 & 0 & 0 & 0 & 0 \\
 WP_t & \gamma_0 & 0 & 0 & 0 & \gamma_2 & \gamma_3 & 0 & 0 \\
 X_t & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 P_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
 K_t & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
 \end{array}$$

$$U = +(u_1 + u_2 + u_3)i$$

OLS	<pre> ja=. klein_sub0 KLEIN (1) 16.2366 0.192934 0.0898849 0.796219 (2) 8.36456 0.514518 0.225604 _0.0977402 (3) 0.194592 0.439477 0.14609 0.130245 </pre>
B	<pre> klein_sub2 KLEIN 1 0 _0.796219 0 _0.192934 0 0 1 0 0 _0.514518 0 0 0 1 _0.439477 0 0 _1 _1 0 1 0 0 0 0 1 _1 1 0 0 _1 0 0 0 1 </pre>
Γ	<pre> klein_sub3 KLEIN 16.2366 0.0898849 0.796219 0 0 0 0 0 8.36456 0.225604 0 _0.0977402 0 0 0 0 0.194592 0 0 0 0.14609 0.130245 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 _1 0 0 0 1 0 0 0 0 0 </pre>
REG	<pre> : (%. klein_sub2 KLEIN) +/ . * klein_sub3 KLEIN (1) (2) (3) (4) (5) (6) 60.832 36.268 42.8678 97.1 54.2322 36.268 0.659878 0.584474 0.546864 1.24435 0.697488 0.584474 2.23474 0.905702 1.38015 3.14045 1.76029 0.905702 _0.176587 _0.20892 _0.169421 _0.385507 _0.216086 0.79108 0.111563 _0.0604149 0.168568 0.0511479 _0.11742 _0.0604149 0.0994628 _0.0538624 0.150286 0.0456005 _0.104685 _0.0538624 1.8067 1.1375 1.73338 3.9442 2.21081 1.1375 _1.47109 _1.31925 _1.22629 _2.79033 _2.56405 _1.31925 </pre> スペースの関係で <i>transpose</i> :した

RMSE

```
,. rmse klein_model KLEIN
```

```
+-----+
|5.51403 4.28168 4.1196 9.60386 5.8179 8.05186|
+-----+
|0.0027136 0.872179 0.0035421 0.00640492 0.0275251 0.000443801|
+-----+
```

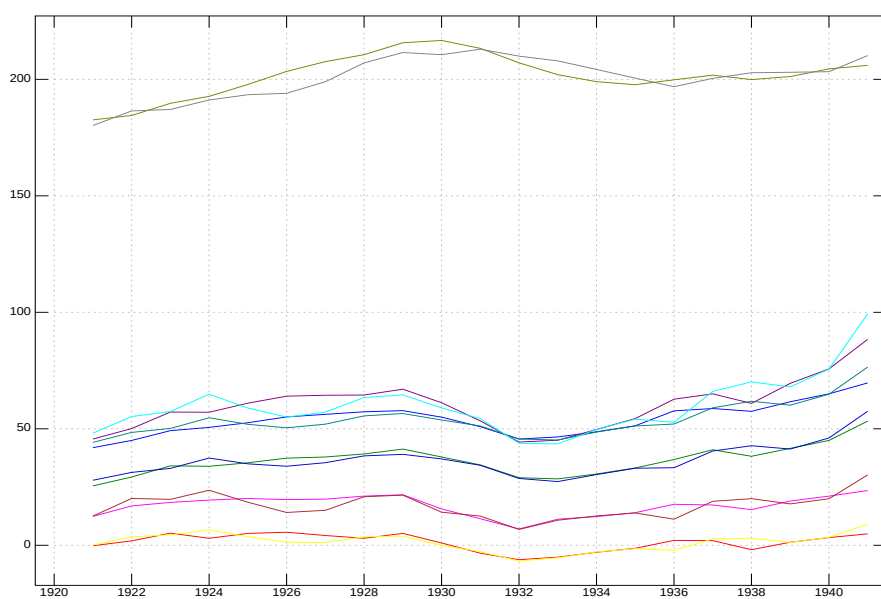


図 4.8 L.Klein

4.4.4 Reference

白砂堤津耶「例題で学ぶ・初歩からの計量経済学」1998/2007 日本評論社

第 5 章

推移確率

5.1 Markov chain/マルコフの推移確率

Andrei A.Markov(1856-1922) がプーシキンの「スペードの女王」を読んでいたといわれる。(1906)

The Markov property means that,given present state,future states are independant of past state

過去、現在、未来の連続性を認めず、未来は過去とは独立であるとする考え方である。
哲学はさておき数学ではマルコフ性とは未来の挙動は現在の値のみで決定される系列で、過去の挙動とは無関係であるというものである。

古いデータをあまり気にしなくとも良いという性質を前向きに考えると利用の範囲は広がる。

推移確率

$$P = \begin{bmatrix} P(1,1) & P(1,2) & P(1,3) & \cdots \\ P(2,1) & P(2,2) & P(3,3) & \cdots \\ P(3,1) & P(2,2) & P(3,3) & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Example	<pre>] M0=. 1r2 1r4 1r4,1r2 0 1r2 , :1r4 1r4 1r2</pre><table><tr><th></th><th><i>Rain</i></th><th><i>Nice</i></th><th><i>Snow</i></th></tr><tr><th><i>Rain</i></th><td>1r2</td><td>1r4</td><td>1r4</td></tr><tr><th><i>Nice</i></th><td>1r2</td><td>0</td><td>1r2</td></tr><tr><th><i>Snow</i></th><td>1r4</td><td>1r4</td><td>1r2</td></tr></table>		<i>Rain</i>	<i>Nice</i>	<i>Snow</i>	<i>Rain</i>	1r2	1r4	1r4	<i>Nice</i>	1r2	0	1r2	<i>Snow</i>	1r4	1r4	1r2	<pre>_1 x: M0 ([+/ . *])^:10 M0 0.4 0.2 0.4 0.4 0.2 0.4 0.4 0.2 0.4 10 回の反復で収束する</pre>
	<i>Rain</i>	<i>Nice</i>	<i>Snow</i>															
<i>Rain</i>	1r2	1r4	1r4															
<i>Nice</i>	1r2	0	1r2															
<i>Snow</i>	1r4	1r4	1r2															

issue:Kemeny,Snell,Thompson[Introduction to Finite Mathematics 3rd.ed.] Prentishall
1974

5.1.1 極限分布と定常分布

極限分布が存在するかどうかは $P^{n+1} = P^n \times P$ をループで回して計算するのが早道である。

$n \rightarrow \infty$ 実際は少ない回数で収束する。

$\pi = \pi P$ に収束した場合の分布を定常分布と言う。

```

10 markov_loop M0
+---+-----+
|0 | 0.5 0.25 0.25 |
| | 0.5 0 0.5 |
| | 0.25 0.25 0.5 |
+---+-----+
|1 |0.4375 0.1875 0.375 |
| | 0.375 0.25 0.375 |
| | 0.375 0.1875 0.4375 |
+---+-----+
|2 | 0.40625 0.203125 0.390625 |
| | 0.40625 0.1875 0.40625 |
| | 0.390625 0.203125 0.40625 |
+---+-----+
|3 |0.402344 0.199219 0.398438 |
| | 0.398438 0.203125 0.398438 |
| | 0.398438 0.199219 0.402344 |
+---+-----+
|4 |0.400391 0.200195 0.399414 |
| | 0.400391 0.199219 0.400391 |
| | 0.399414 0.200195 0.400391 |
+---+-----+
+---+-----+
|5 |0.400146 0.199951 0.399902 |
| | 0.399902 0.200195 0.399902 |
| | 0.399902 0.199951 0.400146 |
+---+-----+
|6 |0.400024 0.200012 0.399963 |
| | 0.400024 0.199951 0.400024 |
| | 0.399963 0.200012 0.400024 |
+---+-----+
|7 |0.400009 0.199997 0.399994 |
| | 0.399994 0.200012 0.399994 |
| | 0.399994 0.199997 0.400009 |
+---+-----+
|8 |0.400002 0.200001 0.399998 |
| | 0.400002 0.199997 0.400002 |
| | 0.399998 0.200001 0.400002 |
+---+-----+
|9 |0.400001 0.2 0.4 |
| | 0.4 0.200001 0.4 |
| | 0.4 0.2 0.400001 |
+---+-----+
|10|0.4 0.2 0.4 |
| | 0.4 0.2 0.4 |
| | 0.4 0.2 0.4 |
+---+-----+

```

10 回の反復で収束した。この値が極限分布であり、定常分布である。

5.1.2 固有ベクトルから定常分布を求める

既約、再帰的かつ非周期的な場合に、推移確率行列 P^n は次の性質をもつ。

既約、再帰的かつ非周期的とは推移確率をグラフで描くとはっきりするが、非常に友好

的で誰とでも、いつでも、何回でも訪問できるという関係を示す。

最大固有値が 1 最大固有値を求めると 1 で、他は絶対値が 1 より小さい。

固有値 1 の固有ベクトルは この固有値 1 の固有ベクトルは定常分布となる。

```
char_lf M0
+--+-----+-----+
|1|1 1r4 _1r4|1r16 _1r16 _1 1|
+--+-----+-----+

cahr_evec M0
+-----+-----+-----+
|1          |1r4          |_1r4          |
+-----+-----+-----+
|3r8 3r16 3r8|_3r16 0 3r16|1r16 _1r8 1r16|
|3r8 3r16 3r8| 0 0 0|_1r4 1r2 _1r4|
|3r8 3r16 3r8| 3r16 0 _3r16|1r16 _1r8 1r16|
+-----+-----+-----+

f(λ) = 1/16 - 1/16λ - λ² + λ³
Eigenvalue is 1 0.25 -0.25

a1=. {.;{.f: char_evec D510
3r8 3r16 3r8

_1 x: a1 % +/- a1
0.4 0.2 0.4
```

これはフロベニウスの定理の応用である。^{*1}

このように友好的な推移確率ばかりではなく周期的反復を繰り返し極限分布を持たないものもある。

5.1.3 マルコフチェーンの応用/銘柄選択

ビールの銘柄 (A,B,C,D) 選択。直前に買った銘柄と次に買う銘柄。

推移確率行列 M3

^{*1} フロベニウスの定理はレオンテフ逆行列でも現れる

$M3 = \begin{bmatrix} \text{Beer} & A & B & C & D \\ A & 0.9 & 0.05 & 0.03 & 0.02 \\ B & 0.1 & 0.8 & 0.05 & 0.05 \\ C & 0.08 & 0.1 & 0.8 & 0.02 \\ D & 0.1 & 0.1 & 0.1 & 0.7 \end{bmatrix}$					
<pre>{: 100 markov_loop M3</pre>					
<pre> A B C D</pre>					
<pre>+-----+</pre>					
<pre> 0.482133 0.252978 0.178673 0.0862167 </pre>					
<pre> 0.482133 0.252978 0.178673 0.0862167 </pre>					
<pre> 0.482133 0.252978 0.178673 0.0862167 </pre>					
<pre> 0.482133 0.252978 0.178673 0.0862167 </pre>					
<pre>+-----+</pre>					

(issue: 伏見 例 7.1)

時が過ぎると、A は 48%、B は 25%、C は 18%、D は 9% 程度に落ち着くと言うのがマルコフ連鎖の主張である。

既約、再帰、非周期な推移行列は反復している内に友好関係から現在どの状態にいるかよりも、どの状態にもいけるチャンスが等しく、コミュートした状態で収束する。

現在のシェアではなく、推移確率が未来を決定する。脇の甘い銘柄はシェアをとれない。固定客、常連こそが肝要である。脇の甘い銘柄に客を奪われるのはなお悪い。

5.1.4 電球の取り替え問題

[例題] ある工場の 1000 個ある電球取り替え率。切れた電球は月末にまとめて取り替える。5ヶ月で全部が取り変わる。

経過月数 取り替え率

1	0.05
2	0.20
3	0.60
4	0.15
5	0.02

M4

0.03 0.2 0.6 0.15 0.02

推移確率

$$P = \begin{bmatrix} p_1 & q_1 & 0 & 0 & 0 \\ q_2 & 0 & q_2 & 0 & 0 \\ q_3 & 0 & 0 & q_3 & 0 \\ q_4 & 0 & 0 & 0 & q_4 \\ q_5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$p_i + q_i = 1$$

$$p_5 = 1$$

$p_1 = 0.03$
 $q_1 p_2 = 0.20$
 p_1, q_i と取り替え率の関係 $q_1 q_2 p_3 = 0.60$
 $q_1 q_2 q_3 p_4 = 0.15$
 $q_1 q_2 q_3 q_4 p_5 = 0.15$
 p_1, q_i と取り替え率の関係から p_1, q_i が求められる。

```

r_markov M4
0.03 0.97      0      0      0
0.206186      0 0.793814      0      0
0.779221      0      0 0.220779      0
0.882353      0      0      0 0.117647
      1      0      0      0      0

;{: 100 markov_loop r_markov M4
NB. 1      2      3      4      5
0.341297 0.331058 0.262799 0.0580205 0.00682594
0.341297 0.331058 0.262799 0.0580205 0.00682594
0.341297 0.331058 0.262799 0.0580205 0.00682594
0.341297 0.331058 0.262799 0.0580205 0.00682594
0.341297 0.331058 0.262799 0.0580205 0.00682594
  
```

5.1.5 Reference

伏見正則 「確率と確率過程」 朝倉書店 2004

5.1.6 Script

```

] M0=: 1r2 1r4 1r4,1r2 0 1r2,:1r4 1r4 1r2

mmp=: [ +/ . * ]
NB. M0 +/ . * M0 +/ . * M0
NB. M0 mmp (^:10) M0

markov_loop=: 4 : 0
NB. markov chain
NB. x is time of loop
NB. y is markov matrix
TMP=. y +/ . * y
ANS=. y ; TMP
for_ctr. i. x do.
TMP=. TMP +/ . * y
  
```

```

ANS=.ANS,<TMP
end.
ANS=._1 x: L:0 ANS      NB. rational to natural number
TMP1=.reform_sub L:0 ANS NB. pick-off dust
,.(;$ L:0 {. ANS)$ L:0 TMP1 NB. reform
)

reform_sub=: 3 : ' (0) (I. ( ,y) < 1e_6)},y'

```

5.2 人口問題とレスリー行列

5.2.1 平均寿命

小さなマルコフ推移確率行列

最初に出生を除いた現在の状態からのマルコフ推移確率行列を扱う。

吸収型 (推移確率のいずれかが 1 のとき)

Q の固有値が 1 より小さいので

$$Q^n \rightarrow 0 (n \rightarrow \infty)$$

$$I + Q + Q^2 + \cdots = (I - Q)^{-1}$$

*2

小さなサンプル

issue: Hirashita p49

D524

age 死亡率

0 0.1

1 0.05

2 0.2

3 1

age.mat.sub D524

$$\begin{pmatrix} & death & age0 & age1 & age2 & age3 \\ death & 1 & 0 & 0 & 0 & 0 \\ age0 & 0.1 & 0 & 0.9 & 0 & 0 \\ age1 & 0.05 & 0 & 0 & 0.95 & 0 \\ age2 & 0.2 & 0 & 0 & 0 & 0.8 \\ age3 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ R & Q \end{pmatrix}$$

$$Q = \begin{pmatrix} 0 & r_0 & 0 & 0 \\ 0 & 0 & r_1 & 0 \\ 0 & 0 & 0 & r_2 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q^2 = \begin{pmatrix} 0 & 0 & r_0 r_1 & 0 \\ 0 & 0 & 0 & r_1 r_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q^3 = \begin{pmatrix} 0 & 0 & 0 & r_0 r_1 r_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

0 才の平均余命 $1 + r_0 + r_0 r_1 + r_0 r_1 r_2$ $1 + 0.9 + 0.855 + 0.684$

1 才の平均余命 $1 + r_1 + r_1 r_2$ $1 + 0.95 + 0.76$

2 才の平均余命 $1 + r_2$ $1 + 0.8$

3 才の平均余命 1

0 才の平均余命が平均寿命である。

*2 レオンテフ逆行列と同一の構造である。

回転の力業

2 行目に 0 才の行列があるのが特徴である

```

2 markov_loop a0
+--+-----+
|0|  1 0  0  0  0 |
| | 0.1 0 0.9  0  0 |
| |0.05 0  0 0.95  0 |
| | 0.2 0  0  0 0.8 |
| |  1 0  0  0  0 |
+--+-----+
|1|  1 0 0  0  0 |
| |0.145 0 0 0.855  0 |
| | 0.24 0 0  0 0.76 |
| |  1 0 0  0  0 |
| |  1 0 0  0  0 |
+--+-----+
|2|  1 0 0 0  0 |
| |0.316 0 0 0 0.684 |
| |  1 0 0 0  0 |
| |  1 0 0 0  0 |
| |  1 0 0 0  0 |
+--+-----+

```

定常分布が存在するのでループ回数は関係なくなる。

最初の行を取り去り、横に足せば平均寿命となる

```

+ /> 2 markov_loop a0

      4 0  0  0  0
1.561 0 0.9 0.855 0.684
2.29 0  0 0.95 0.76
3.2 0  0  0 0.8
4 0  0  0  0

alive D524
0 3.439
1 2.71
2 1.8
3 1

```

$I - Q$ の逆行列

$(I - Q)^{-1}$ NB. $I - Q$ の逆行列

matrix Q

```

0 0.9  0  0
0  0 0.95  0
0  0  0 0.8
0  0  0  0

```

逆行列による解

$(I - Q)^{-1}$

定常分布になる

```

alive0 D524 N
1 0.9 0.855 0.684
0 1 0.95 0.76
0 0 1 0.8
0 0 0 1

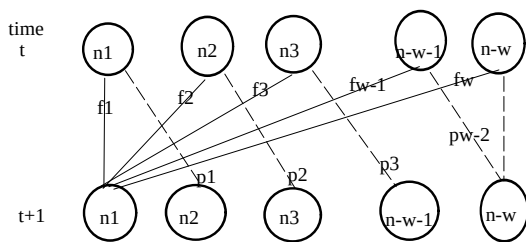
```

5.2.2 レスリー行列

P.H.Leslie(1900 – 1974) Oxford に学び、Oxford の *Bureau of Animal Population* で働く。

1940 年代のはじめに *Bxernardelli*(1941), *Lewis*(1942), *Leslie*(1945) がマトリクスで人口や動物の生殖のモデル化に成功した。

1959 Leslie の改良版 1966 J.H. Pollard stochastic version



$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_k \end{bmatrix} (t+1) = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_{k-1} & f_k \\ p_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & p_2 & & & & \\ 0 & 0 & p_3 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ 0 & 0 & 0 & \cdots & p_{k-1} & 0 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_k \end{bmatrix} (t)$$

RACCOON の繁殖モデル

先のモデルを少し改良して *Leslie* 行列を作成する。

狸 狸の繁殖モデル（次のように仮定する）

子がつがいで一匹生む。年一回繁殖

雌雄の出生率に差はない。

寿命は 5 才で天寿を終える

入力データ RACCOON ; ベクトルを; で連結する

初期個体数	出生率	生存率
100 100 100 100 100	0 0.4 0.3 0.2 0.1	0.9 0.95 0.8 0.7

Leslie Matrix `leslie_mat` }. RACCOON

```

      0  0.4 0.3 0.2 0.1
0.9    0    0    0    0
      0 0.95    0    0    0
      0    0 0.8    0    0
      0    0    0 0.7    0

```

アルゴリズムの確認 最初の乗算は次のようになる。獣の個体数の初期値として客齢 100 匹とすると

1 行目は出生数となる。2 行目以下は各齢の生存数である。

```

a4 * P0
0 40 30 20 10 NB. birth 100
90 0 0 0 0 NB. die 10
0 95 0 0 0 NB. die 5
0 0 80 0 0
0 0 0 70 0

```

内積計算は上のマトリクスを横方向に足し合わせる。次年の各齢の生息数になる。初年度は各齢 100 なので出生数は 100 になり再生できた。しかし、次の年からは 100 を切っているので個体数は再生できない。

```

a4 +/ . * P0      NB. 合計は横に取る
100 90 95 80 70

```

計算結果 客年の個体数の推移。1 このモデルでは 10 年で 43

```
10 leslie_loop RACCOON
```

t	1	2	3	4	5-old	total(n)
0	100	100	100	100	100	500
1	100	90	95	80	70	435
2	87.5	90	85.5	76	56	395
3	82.45	78.75	85.5	68.4	53.2	368.3
4	76.15	74.205	74.8125	68.4	47.88	341.447
5	70.5938	68.535	70.4947	59.85	47.88	317.354
6	65.3204	63.5344	65.1083	56.3958	41.895	292.254
7	60.4149	58.7884	60.3577	52.0866	39.4771	271.125
8	55.9877	54.3734	55.849	48.2861	36.4606	250.957
9	51.8073	50.3889	51.6547	44.6792	33.8003	232.33
10	47.9678	46.6266	47.8695	41.3238	31.2754	215.063

5.2.3 雌雄の差のあるモデル

男女、雌雄の出生率、生存率に差がある場合は、レスリー行列を F M の生存率のマトリクスと出生率のベクトルの3つに分解して、シンクロナイズしなければならない。

ループの中の出生率は一定にしてあるが工夫すれば出生率や生存率を変化させることもできる。

RACCOON2

100 100 100 100 100 NB. Alive F

100 100 100 100 100 NB. Alive M

0 0.4 0.3 0.2 0.1 NB. fx

0.9 0.95 0.8 0.7 0 NB. Pf

0.8 0.9 0.7 0.6 0 NB. Pm

0.9 0 0 0 0 NB. Pf matrix

0 0.95 0 0 0

0 0 0.8 0 0

0 0 0 0.7 0

0 0.4 0.3 0.2 0.1 NB. fx

5 leslie_human_loop RACCOON2

```
+-----+
|0      200      200      200      200      200|NB. Total
|1      100      170      185      150      130|
|2      87.5    84.878    157.5    139      98|
|3    64.011    74.2683    78.5854    118.8    91|
|4    46.878    54.3313    68.7622    59.1805    78.12|
|5    33.6503    39.7892    50.3033    51.7829    38.8449|
+-----+
|0      100      100      100      100      100|NB. F
|1    48.7805      90      95      80      70|
|2    42.6829    43.9024    85.5      76      56|
|3    31.2249    38.4146    41.7073    68.4    53.2|
|4    22.8673    28.1024    36.4939    33.3659    47.88|
|5    16.4148    20.5806    26.6973    29.1951    23.3561|
+-----+
|0      100      100      100      100      100|NB. M
|1    51.2195      80      90      70      60|
|2    44.8171    40.9756      72      63      42|
|3    32.7861    35.8537    36.878    50.4    37.8|
```

```
| 4 24.0107 26.2289 32.2683 25.8146 30.24|
| 5 17.2355 19.2086 23.606 22.5878 15.4888|
+-----+
```

5.2.4 人口計算のサンプル

10 才構成のモデル (三土による。団塊の世代のサンプルらしい)
 レスリーモデルとは明記していないが計算確認のサンプルの好例として用いる。
 次のような構成になる。(10 才階層)
 f は出生率、 P は生存率を入れる。

```
D522
_0.02 0.13 0.63 0.16 0.02 0 0 0 0 0
0.98 0 0 0 0 0 0 0 0 0
0 0.99 0 0 0 0 0 0 0 0
0 0 0.98 0 0 0 0 0 0 0
0 0 0 0.97 0 0 0 0 0 0
0 0 0 0 0.95 0 0 0 0 0
0 0 0 0 0 0.9 0 0 0 0
0 0 0 0 0 0 0.8 0 0 0
0 0 0 0 0 0 0 0.63 0 0
0 0 0 0 0 0 0 0 0.15 0
```

issue: 三土

5.2.5 レスリー行列/階層モデル

マルコフは先ず力業で回してみ、安定したら他の数学方式で検証する。
 人口を掛け合わせるプロセスが必要なので *markov.loop* をレスリー行列用に改良する。
 やはり数行で足りる。

```
leslie_loop=:4 : 0
NB. markov chain
NB. Usage: e.g. leslie_loop RACCOON/ POP;Fx;Px
NB. x is times to loop
NB. y is 3 factors
NB. Population; f(birth-rate);p(alive-rate)
'POP F0 P0'=. y
P0=. } P0
MAT=. F0, (P0 *=i. # P0),. 0 NB. make Leslie matrix
ANS=. < POP
for_ctr. i. x do.
POP=. MAT +/ . * POP
```

```

ANS=.ANS,<POP
end.
TMP=;("1),. ANS
(i.>: x),. TMP,. +/"1 TMP
)

leslie_mat0=: 3 : 0
'F0 P0'= y NB. Fx;Px
F0, (P0 *=i. # P0),. 0 NB. make Leslie matrix
)

leslie_human_loop=:4 : 0
NB. like sincronized swimming
NB. Usage: e.g. leslie_loop RACCOON2
NB. x is times to loop
NB. y is 3 factors
NB. Population; f(birth-rate);p(alive-rate)
'POPF POPM F0 PF PM'= y NB. if table(file) {|: file // same length
PF=. }: PF NB. discard lsat added 0
PM=. }: PM
NB. fixed parameter matrix /vector
MATF=: (PF *=i. # PF),.0 NB. make alive famele matrix
MATM=: (PM *=i. # PM),.0 NB. make alive man matrix
VECFX=: F0 NB. birth vector
NB. -----
ANSF=. < POPF
ANSM=. < POPM
for_ctr. i. x do.
'BIRTHF BIRTHM'= VECFX leslie_birth_sub POPF
ANSF=. ANSF,<POPF=. BIRTHF, MATF +/ . * POPF
ANSM=. ANSM,<POPM=. BIRTHM, MATM +/ . * POPM
end.
TMP=. (ANSF + L:0 ANSM),ANSF,ANSM
TMP=.;("1)( ;3# <{@> i.>: x),. TMP
IND=. ;3 #< 1, x # 0
,.IND<:.1 TMP
)

leslie_birth_sub=: 4 : 0
BRATE=.tmp % +/ tmp=. 100 105 NB. birth rate/famele man

```

```

BIRTH=. x +/ . * y      NB. total birth
BIRTH * BRATE            NB.
)

```

```

\begin{verbatim}
    leslie_mat D521;D520
0.02 0.13 0.63 0.16 0.02  0  0  0  0 0
0.98  0  0  0  0  0  0  0  0 0
  0 0.99  0  0  0  0  0  0  0 0
  0  0 0.98  0  0  0  0  0  0 0
  0  0  0 0.97  0  0  0  0  0 0
  0  0  0  0 0.95  0  0  0  0 0
  0  0  0  0  0 0.9  0  0  0 0
  0  0  0  0  0  0 0.8  0  0 0
  0  0  0  0  0  0  0 0.63  0 0
  0  0  0  0  0  0  0  0 0.15 0

```

三土の *Example* で 20 年分計算してみる。(10 才刻みで団塊の世代の女性版のようだ) 人口動態モデルになる。

```

5j0 ": 20 leslie_loop D523;D521;D520
t 0->10 ->20 ->30 ->40 ->50 ->60 ->70 ->80 ->90 ->10x total
0  98  55  35  28  23  22  16  9  6  1 293
1  36  96  54  34  27  22  20  13  6  1 309
2  54  35  95  53  33  26  20  16  8  1 341
3  75  52  35  93  52  32  23  16  10  1 389
4  46  73  52  34  90  49  28  19  10  1 404
5  50  45  73  51  33  86  44  23  12  1 419
6  61  49  45  71  49  32  77  35  14  2 437
7  48  60  49  44  69  47  28  62  22  2 432
8  48  47  60  48  43  66  42  23  39  3 419
9  53  47  47  58  47  41  59  34  14  6 406
10 47  52  47  46  57  44  37  47  21  2 400
11 46  46  52  46  45  54  40  29  30  3 389
12 48  45  46  51  44  42  48  32  18  4 378
13 44  47  44  45  49  42  38  39  20  3 371
14 43  44  46  43  43  47  38  30  24  3 362
15 43  42  43  45  42  41  42  30  19  4 352
16 42  43  42  42  44  40  37  34  19  3 345
17 40  41  42  41  41  42  36  30  21  3 336
18 40  39  40  41  40  39  38  29  19  3 328

```

19	39	39	39	40	40	38	35	30	18	3	320
20	38	38	39	38	38	38	34	28	19	3	313

5.2.6 Reference

David Sherwell , LondiweMasigna [Leslie Matrix Model in Population Dynamics]

三土修平 「初歩からの経済数学」日本評論社 1991

平下幸男「数理科学のレッスン」産業図書 1992

第 6 章

マトリクスの活用

6.1 主成分分析

相関行列 分散共分散行列から計算する場合と相関行列からの場合がある。ここでは相関行列から行う。

固有値 2 段目の上の行が固有値である。(大きい順に並べてる。) 2、3 行は寄与率、累積寄与率で算数で求められる。主成分を検討するとき、相関行列なら固有値が 1 以上、分散共分散行列なら累積寄与度が 80% が一つの目安のようだ。

固有ベクトル 3 段目が各固有値に関連する固有ベクトルで縦に表記されている。第一主成分 (最大固有値) や第二主成分の固有ベクトルがどの変数と関連が強いかを見て主成分を推測する

次のデータ (STYLE) は第 2 回から第 25 回のミスユニバース日本代表の身長、体重と 3 サイズである。(出典 鈴木)

*1

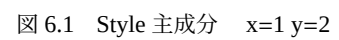
```
}: calc_once STYLE
```

+-----+-----+-----+-----+-----+-----+									
COR		1	0.542494	0.326565	0.372425	0.0115011			
		0.542494		1	0.3131	0.449056	0.241926		
		0.326565	0.3131		1	0.0605705	0.424285		
		0.372425	0.449056	0.0605705		1	_0.0649716		
		0.0115011	0.241926	0.424285	_0.0649716			1	
+-----+-----+-----+-----+-----+-----+									
EVAL	2.14898	1.30735	0.675427	0.509665	0.358586				
		0.429795	0.261469	0.135085	0.101933	0.0717172			
		0.429795	0.691264	0.82635	0.928283		1		
+-----+-----+-----+-----+-----+-----+									
EVEC	0.521033	_0.236935	_0.534739	0.27597	0.557029				
		0.570928	_0.104455	0.198549	0.481168	_0.626247			
		0.419977	0.483214	_0.401878	_0.591852	_0.279874			

*1 Appendix 参照

6.1.1 主成分スコア

plot_principal STYLE
右上の飛び抜けたのが *Akiko Kojima*



6.2 The input output model

Wassili Leontief (1905 – 1999) レニングラードで生まれ、レニングラード大学で哲学、社会学、経済学を学ぶ。癌を理由にソ連を出国でき、ドイツを経て 1932 年からハーバード大学で研究。

IBM の巨大な真空管のリレー式コンピューター *MarkII* はペンタゴンの次にハーバードに導入された。産業連関の計算の要請がレオンテフから寄せられ、これをサポートしたのがハーバードで数学とコンピュータの研究を始めたばかりの若き日の *K.E.Iverson* であった。50×50 に縮小したマトリクスの逆行列の計算に *MarkII* はうなりをあげて 48 時間を要したと記録されている。

6.2.1 石炭、電気、鉄道の 3 部門モデル

Consider an economy with three industries:

	coal	electricity	railway
coal	0	0.65	0.55
electricity	0.25	0.05	0.10
railway	0.25	0.05	0

To produce 1 \$ of coal , requires \$ 0.25 worth of electricity and \$ 0.25 rail costs for transportation

To produce 1 \$ of electricity , requires \$ 0.65 worth of coal for fuel and \$ 0.05 of electricity for the auxulialy equipment ,and \$ 0.05 for transportation

To provide 1 \$ of transport , the railway requires \$ 0.55 of coal for fuel and \$ 0.10 electricity

Each week the external demand for coal is \$ 25000

external demand for electricity is \$ 50000

and no external demand for railway

$$x = \begin{bmatrix} 50000 \\ 25000 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.65 & 0.55 \\ 0.25 & 0.05 & 0.10 \\ 0.25 & 0.05 & 0 \end{bmatrix} x = d + Ax$$

issue: Martin and Anthony Example 19.3

$$(-A)x = d$$

$$x = (I - A)^{-1}d$$

	A5 0 0.65 0.55 0.25 0.05 0.1 0.25 0.05 0 D5 50000 25000 0
I	(= /~ i.3) 1 0 0 0 1 0 0 0 1
$I - A$	(= /~ i.3) - A5 1 _0.65 _0.55 _0.25 0.95 _0.1 _0.25 _0.05 1
$(I - A)^{-1}$	D5 %. %. (= /~ i.3) - A5 1.50298 1.07753 0.934394 0.437376 1.37177 0.377734 0.397614 0.337972 1.25249
$x = (I - A)^{-1}d$ weekly production schedule	D5 %. (= /~ i.3) - A5 102087 56163 28330 NB. coal electricity railway

char_lf A5

```

+-----+
|1|0.601741 _0.497454 _0.0542863|_0.01625 _0.305 _0.05 1|
+-----+

```

Characteristic equation: $-0.01625 - 0.305\lambda_1 - 0.05\lambda_2^2 + \lambda_3^3$

$$(I - A)^{-1} = I + A + A^2 + A^3 + A^4 + \dots$$

$$(I - A)^{-1} = \lim_{n \rightarrow \infty} (I + A + A^2 + A^3 + A^4 + \dots + A^{n-1} + A^n)$$

$$(I - A)^{-1} = \lim_{n \rightarrow \infty} \sum_{k=0}^n A^k = \lim S_n$$

6.2.2 非負正方行列

投入係数行列は非負正方行列である。この行列は最大固有値が正の実数で、この最大固有値に属する固有ベクトルは非負ベクトルである。(フロベニウスの定理)

投入係数行列の(絶対値)最大の固有値から斉一成長の成長率が求められる。

$$g = \frac{1}{\lambda} - 1$$

6.2.3 Reference

Martin Anthony Norman Biggs [Mathematics for Economics and Finance] Cambridge University Press 1996

6.3 マトリクスとグラフ理論

6.3.1 マトリクスとグラフ理論

先の石炭、電気、鉄道の3部門の産業連関をグラフで表したものの。

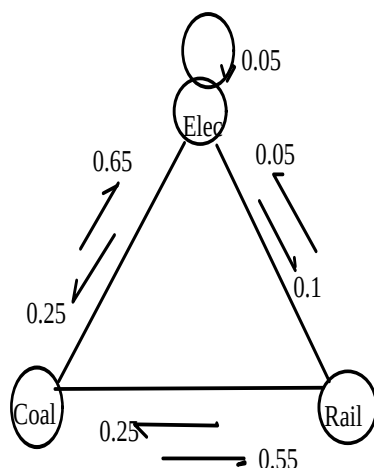


図 6.2 characteristic equation

強連結

対角優位 各行の非対角成分の絶対値の総和がその行の対角成分の絶対値を超えず、少なくとも一つの行については厳密に小さいこと

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, i = 1, \dots, n$$

6.3.2 ワーシャル・フロイト法

Warshall-Floyd 法は列と行の外積（和）を求め、行列の比較を行い、 $\min$ を採っていく方法である。組み合わせによるフロイド法よりもアルゴリズムが簡潔であり、堅牢である。

6.3.3 Matrix sample

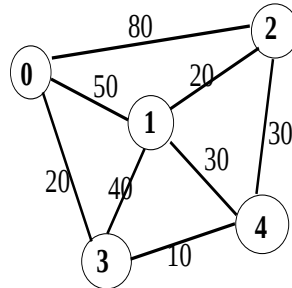
伊理ー古林に紹介されているマトリクス

6.3.4 Algolism

0列と0行の**加算の外積**（和）をもとめる。

当初の行列と比較し $\min$ があれば書き換える。（更新）無限大から100に2カ所更新

	0	1	2	3	4
0	0	50	80	20	-
1	50	0	20	40	30
2	80	20	0	-	30
3	20	40	-	0	10
4	-	30	30	10	0



される。

$$d_{ij}^1 = \min(d_{ij}^0, d_{i1}^0 + d_{1j}^0)$$

(0{"1 wfd} +/ 0{wfd NB. 0行と0列との加算の外積

0	50	80	20	-	0	50	80	20	-
50	100	130	70	-	50	0	20	40	30
80	130	160	100	-	80	20	0	100	30
20	70	100	40	-	20	40	100	0	10
-	-	-	-	-	-	30	30	10	0

1列と1行の加算の外積（和）をもとめる。

先の更新された行列と比較し $\min$ があれば書き換える。（更新）される。

$$d_{ij}^2 = \min(d_{ij}^1, d_{i1}^1 + d_{1j}^1)$$

100	50	70	90	80	0	50	70	20	80
50	0	20	40	30	50	0	20	40	30
70	20	40	60	50	70	20	0	60	30
90	40	60	80	70	20	40	60	0	10
80	30	50	70	60	80	30	30	10	0

以後同様に n 列 n 行の外積（和）を計算し、 $\min$ を求めて、更新を最終行列まで繰り返す。

返す。

最終結果と軌跡 (TRACE)

wf wfd

```
0 50 60 20 30
50 0 20 40 30
60 20 0 40 30
20 40 40 0 10
30 30 30 10 0
```

TRACE

```

0 0 4 0 3
1 1 1 1 1
4 2 2 4 2
3 3 4 3 3
3 4 4 4 4

```

6.3.5 高速道路の最短距離のアルゴリズム

東名高速道路 東京ー御殿場間のノードと距離

```

NODE
0 1 7.6 7.6 NB. 東京ー東名川崎
1 2 5.7 5.7
2 3 6.4 6.4
3 4 15.3 15.3
4 5 15.1 15.1
5 6 7.8 7.8
6 7 25.8 25.8 NB. 大井松田ー御殿場

```

計算の基礎となるマトリクス（両側）

```

make_mat_1 NODE
0 7.6 - - - - -
7.6 0 5.7 - - - - -
- 5.7 0 6.4 - - - -
- - 6.4 0 15.3 - - -
- - - 15.3 0 15.1 - -
- - - - 15.1 0 7.8 -
- - - - - 7.8 0 25.8
- - - - - - 25.8 0

```

距離テーブル

```

wf_non make_mat_1 NODE
NB.
東京 川崎 青葉 横浜 厚木 秦野 大井 御殿場
0 7.6 13.3 19.7 35 50.1 57.9 83.7
7.6 0 5.7 12.1 27.4 42.5 50.3 76.1
13.3 5.7 0 6.4 21.7 36.8 44.6 70.4
19.7 12.1 6.4 0 15.3 30.4 38.2 64
35 27.4 21.7 15.3 0 15.1 22.9 48.7
50.1 42.5 36.8 30.4 15.1 0 7.8 33.6

```

57.9	50.3	44.6	38.2	22.9	7.8	0	25.8
83.7	76.1	70.4	64	48.7	33.6	25.8	0

6.4 シンプレックス法

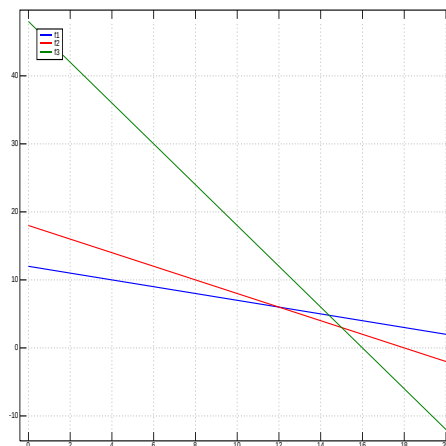
線形計画は、変数が多数の1次不等式によって制約されている場合に、与えられた1次式を最適化する。この効率的な解法がシンプレックス法である。シンプレックス法はG.B.Dantzig(1914-2005)が1947年に提唱した。

6.4.1 図解

Example

	材料 A(kg)	材料 B(kg)	材料 C(kg)	売上単価 (万円)
製品 A	1	1	3	2
製品 B	2	1	1	3
提供できる材料	14	8	18	

$$\begin{array}{ll}
 \text{maximize :} & 2x_1 + 3x_2 \\
 \text{subject to :} & \begin{array}{l} x_1 + 2x_2 \leq 14 \quad f(1) \\ x_1 + x_2 \leq 8 \quad f(2) \\ 3x_1 + x_2 \leq 18 \quad f(3) \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array}
 \end{array}$$



```

f1=. 3 : '7- 0.5*y'
f2=. 3 : '8-y'
f3=. 3 : '18 - 3 * y'

'key f1 f2 f3' plot ;("1") ,.
(f1;f2;f3) L:0 i:10

pd 'eps /temp/simplex_p0.eps'

```

図 6.3 simplex

$f(1)$ と $f(2)$ の交点 A と $f(2)$, $f(3)$ の交点 B が浮かび出る。

A は

$$\begin{cases} x_1 + 2x_2 = 14 \\ x_1 + x_2 = 8 \end{cases}$$

$x_1 = 2, x_2 = 6$ である

$$\text{maximize : } 2x_1 + 3x_2 = 22$$

A を 2 個、 B を 6 個作って 22 万円の売上

B は

$$\begin{cases} x_1 + x_2 = 8 \\ 3x_1 + x_2 = 18 \end{cases}$$

$x_1 = 5, x_2 = 3$ である

$maximize : 2x_1 + 3x_2 = 19$

最適解ではない

6.4.2 simplexnr

`simplexnr` を用いて、シンプレックス法を解いてみよう。

*2

`system/package/math` に `simplexnr.ijs` のスクリプトが用意されている。

EXample

先の例題を検証してみよう。

入力フォーマットは box を連結する。最後のボックスは $\leq$ is _1, $\geq$ is 1, $=$ is 0

```
mat=. 2 3;( 1 2, 1 1,: 3 1);14 8 18;_1 _1 _1
mat
+---+---+-----+-----+
|2 3|1 2|14 8 18|_1 _1 _1|
|   |1 1|       |       |
|   |3 1|       |       |
+---+---+-----+-----+

simplexnr mat
+-+---+---+
|0|2 6|22|
+-+---+---+
```

J の simplexnr の sample

$Maximize x_1 + x_2 + 3x_3 - 0.5x_4$

subject to

$$\begin{array}{rrrrrcl} x_1 & & +2x_3 & & & \leq & 740 \\ & 2x_2 & & -7x_4 & & \leq & 0 \\ & x_2 & -x_3 & +2x_4 & & \geq & 0.5 \\ x_1 & +x_2 & +x_3 & +x_4 & & = & 9 \end{array}$$

```
mat3
+-----+-----+-----+-----+
|1 1 3 _0.5|1 0 2 0|740 0 0.5 9|_1 _1 1 0|
```

*2 simplexnr は H.Ritch が^s Numeric Receipt より作成したもので精度がよい。

```

|          |0 2  0 _7|          |          |
|          |0 1 _1  2|          |          |
|          |1 1  1  1|          |          |
+-----+-----+-----+-----+
simplexnr mat3
+-----+-----+
|0|0 3.325 4.725 0.95|17.025|
+-----+-----+

```

スラック変数

不等式の小さい方の辺に非負の数を加えると等式になる。

この加える変数をスラック変数という。式は = になる。

スラック変数は 1 個 (λ) の場合と、単位行列に相応する x_n を付加する場合がある。

Example

$$f = 29x_1 + 45x_2 \rightarrow \max$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

$$2x_1 + 8x_2 + x_3 = 60$$

$$4x_1 + 4x_2 + x_4 = 60$$

```
ax7=. 29 45 0 0 ;(2 4 $ 2 8 1 0 4 4 0 1);60 60;0 0
```

```

+-----+-----+-----+-----+
|29 45 0 0|2 8 1 0|60 60|0 0|
|          |4 4 0 1|          |          |
+-----+-----+-----+-----+
simplexnr ax7
+-----+-----+
|0|10 5 0 0|515|
+-----+-----+

```

$$x_1 = 10, x_2 = 5, f = 515$$

■Working example $f = 150x_1 + 300x_2 \rightarrow \max$

$$2x_1 + x_2 + x_3 = 16$$

$$x_1 + x_2 + x_4 = 8$$

$$x_2 + x_5 = 3.5$$

```
ax3=.150 300 0 0 0 0;(3 5 $ 2 1 1 0 0 1 1 0 1 0 0 1 0 0 1);16 8 3.5 ;0 0 0
```

```

+-----+-----+-----+-----+
|150 300 0 0 0 0|2 1 1 0 0 0|16 8 3.5|0 0 0|
|          |1 1 0 1 0|          |          |
|          |0 1 0 0 1|          |          |
+-----+-----+-----+-----+

```

```
simplexnr ax3
```

```

+---+-----+
|0|4.5 3.5 3.5 0 0|1725|
+---+-----+

```

$$x_1 = 4.5, x_2 = 3.5, x_3 = 3.5, f = 1725$$

■Working Example 双対問題

$$f = x_1 + 2x_2 \rightarrow \max$$

$$x_1 \geq 0, x_2 \geq 0$$

$$2x_1 - x_2 \leq 7$$

$$3x_1 + x_2 \leq 10$$

$$-x_1 + 2x_2 \leq 18$$

```
ax5=: 1 2 ; (3 2 $ 2 _1 3 1 _1 2); 7 10 18; _1 _1 _1
```

```
simplexnr ax5
```

```

+---+-----+-----+-----+
|1 2| 2 _1|7 10 18|_1 _1 _1|
|   | 3 1|       |       |
|   |_1 2|       |       |
+---+-----+-----+

```

```
simplexnr ax5
```

```

+---+-----+-----+
|0|0.285714 9.14286|18.5714|
+---+-----+-----+

```

$$x_1 = \frac{2}{7}, x_2 = \frac{64}{7}, f = \frac{130}{7}$$

スラック変数を加えた場合

```
ax4=. 1 2 0 0 0 ; (3 5 $ 2 _1 1 0 0 3 1 0 1 0 _1 2 0 0 1); 7 10 18 ; 0 0 0
```

```
simplexnr ax4
```

```

+---+-----+-----+-----+
|1 2 0 0 0| 2 _1 1 0 0|7 10 18|0 0 0|
|           | 3 1 0 1 0|       |       |
|           |_1 2 0 0 1|       |       |
+---+-----+-----+

```

```
simplexnr ax4
```

```

+---+-----+-----+
|0|0.285714 9.14286 15.5714 0 0|18.5714|
+---+-----+-----+

```

双対問題は転置した形である。

$$g = 7y_1 + 10y_2 + 18y_3 \rightarrow \min$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

$$2y_1 + 3y_2 + y_3 \geq 1$$

$$-y_1 + y_2 + 2y_3 \geq 2$$

標準形に直す

$$g' = -7y_1 - 10y_2 - 18y_3 \rightarrow \max$$

$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

$$-2y_1 - 3y_2 + y_3 \leq 1$$

$$y_1 + -2y_2 - 2y_3 \leq 2$$

$$\text{ax6} = [-7 \ -10 \ -18; (2 \ 3 \ 1 \ 1 \ -1 \ -2); -1 \ -2 \ -1 \ -1]$$

$$\begin{array}{cccccc|cccc} + & + & + & + & + & + & + & + & + & + & + & + \\ | & - & 7 & & - & 10 & & - & 18 & | & - & 2 & & - & 3 & & 1 & | & - & 1 & & - & 2 & | & - & 1 & & - & 1 & | \\ | & & & & & & & & & | & 1 & & - & 1 & & - & 2 & | & & & & | & & & & & & & & | \end{array}$$

$$\begin{array}{cccccc|cccc} + & + & + & + & + & + & + & + & + & + & + & + \\ | & 0 & | & 0 & & 0.571429 & & 0.714286 & | & - & 18.5714 & | \\ + & + & + & + & + & + & + & + & + & + & + & + \end{array}$$

$$y_1 = 0, x_2 = \frac{4}{7}, y_3 = \frac{5}{7}, f = \frac{130}{7}$$

■Reference 金谷健一 これなら分かる最適化数学 共立出版 2005

6.5 ゲーム理論

ゲーム理論はプリンストンでハンガリー人のフォン・ノイマンとオーストリーのモルゲンシュタインの二人の亡命科学者の討論と共同研究から生まれた。(Theory of Games and Economics behavior 1944)

予めボードで提示されたゲームスコアを前提にどのような戦略を採用かを考えるが、スコアを動かして戦略の変化を考える楽しみもある。

6.5.1 ジレンマゲーム

競争ゲームである。ほんの少し前まで国道筋のガソリンスタンドでは安売りの看板が並んでいた。ブランドを確立できない商品は安売りに走りやすく疲弊が続く。

ガソリンスタンドのジレンマ (囚人のジレンマのガソリンスタンド・バージョン)

gmatrix GN1

現状維持 値下げ

+-----+

現状維持 | 3 3 | 0 4 |

+-----+

値下げ | 4 0 | 1 1 |

+-----+

ゲーム理論に登場するのは合理的思考を持った冷徹な人であり、競争ゲームでは相談して仲良く (3,3) を取りに行くことは想定外である。

一回きりのゲームでは従って支配戦略が存在し、それは裏切りで、双方が裏切り利得は少なくなる。(値下げ・値下げ) がナッシュ均衡である。

繰り返しゲームでは、最終回が明示されて予定されている場合と、定められていない場合では結論が異なる。また、永久懲罰 (裏切った場合は永久に許さない) か、オウム返し (裏切ったら懲罰を与えるが適当な次期に許す) かの戦略で結果は異なる。

6.5.2 3 択の混合戦略

混合戦略は支配戦略が存在せず、確率でハンドを選択できる。A,B の選択肢が 3 個になった場合たとえばジャンケン。

A\B	y	v	$1-y-v$
x	xy	xv	$x(1-y-v)$
u	uy	uv	$u(1-y-v)$
$1-x-u$	$y(1-x-u)$	$v(1-x-u)$	$(1-x-u)(1-y-v)$

(展開すると)

$A \setminus B$	y	v	$1 - y - v$
x	xy	xv	$x - xy - xv$
u	uy	uv	$u - uy - uv$
$1 - x - u$	$y - xy - uy$	$y - vx - uv$	$1 - x - y - u - v + xy + xv + uy + uv$

係数の $\pm$ は次のような表を作る。ここに $+-$ を付けた箇所から得点を取得する。

0	1	2
3	4	5
6	7	8

そして x, y, u, v と定数項を含めての連立方程式を作成する。連立方程式の解が x, y, u, v の値となる。

	x	y	u	v	定数項																															
x	0	<table><tr><td>+</td><td>-</td></tr><tr><td></td><td></td></tr><tr><td>-</td><td>+</td></tr></table>	+	-			-	+	0	<table><tr><td>+</td><td>-</td></tr><tr><td></td><td></td></tr><tr><td>-</td><td>+</td></tr></table>	+	-			-	+	<table><tr><td></td><td>+</td><td>-</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td>-</td><td>+</td></tr></table>		+	-					-	+	<table><tr><td></td><td></td><td>+</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td>-</td></tr></table>			+						-
+	-																																			
-	+																																			
+	-																																			
-	+																																			
	+	-																																		
	-	+																																		
		+																																		
		-																																		
y	<table><tr><td>+</td><td>-</td></tr><tr><td></td><td></td></tr><tr><td>-</td><td>+</td></tr></table>	+	-			-	+	0	<table><tr><td></td><td></td><td></td></tr><tr><td>+</td><td>-</td></tr><tr><td>-</td><td>+</td></tr></table>				+	-	-	+	0	<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td>+</td><td></td><td>+</td></tr></table>							+		+									
+	-																																			
-	+																																			
+	-																																			
-	+																																			
+		+																																		
u	0	<table><tr><td></td><td></td><td></td></tr><tr><td>+</td><td>-</td></tr><tr><td>-</td><td>+</td></tr></table>				+	-	-	+	0	<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>+</td><td>-</td></tr><tr><td></td><td>-</td><td>+</td></tr></table>					+	-		-	+	<table><tr><td></td><td></td><td>+</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td>-</td></tr></table>			+						-						
+	-																																			
-	+																																			
	+	-																																		
	-	+																																		
		+																																		
		-																																		
v	<table><tr><td></td><td>+</td><td>-</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td>-</td><td>+</td></tr></table>		+	-					-	+	0	<table><tr><td></td><td></td><td></td></tr><tr><td></td><td>+</td><td>-</td></tr><tr><td></td><td>-</td><td>+</td></tr></table>					+	-		-	+	0	<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td>+</td><td>-</td></tr></table>								+	-				
	+	-																																		
	-	+																																		
	+	-																																		
	-	+																																		
	+	-																																		

Working Example (3次・単行列)

次のデータを入れてみる。

$A \setminus B$	y	v	$1 - y - v$	
x	0	15	-5	3 3 \$ GH1
u	5	0	24	0 15 _5
$1 - x - u$	13	8	0	5 0 24
				13 8 0

(出典 平下 文献参照)

$$E = \begin{matrix} & 0xy & +15xv & -5x(1-y-v) \\ & 5uv & +0uv & +24u(1-y-v) \\ +13(1-x-u)y & +8(1-x-u)v & +0(1-x-u)(1-y-v) \end{matrix}$$

$$= -5x + 13y + 24u + 8v - 8xy + 12xv - 32uy - 32uv$$

連立方程式に展開すると次のとおりで、行列方程式をクラメル法で解いて x, y, u, v を求める。

ハンドは x, y, u, v から、プレイヤー A は $x, u, 1 - x - u$, Y は $y, v, 1 - y - v$ が求められる。

$$\begin{cases} x & -8y & +12v & = & 5 \\ y & -8x & -32u & & = & -13 \\ u & -32y & & -32v & = & -24 \\ v & 12x & -32u & & = & -8 \end{cases}$$

mix_3s GH1

+-----+-----+-----+-----+																
	0	15	_5		0	_8	0	12	5		1	0	0	0	0.25	
	5	0	24		_8	0	_32	0	_13		0	1	0	0	0.2	
	13	8	0		0	_32	0	_32	_24		0	0	1	0	0.34375	
					12	0	_32	0	_8		0	0	0	1	0.55	
+-----+-----+-----+-----+																
	A				0.25	0.34375	0.40625									
+-----+-----+-----+-----+																
	B				0.2	0.55	0.25									
+-----+-----+-----+-----+																

6.5.3 Reference

平下幸男「数理科学のレッスン」1992 産業図書

6.6 コンパニオン・マトリクス

6.6.1 多項式を解く

$$p(x) = x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

$$\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

このコンパニオン・マトリクスの固有値は多項式の解である。

Example

$$x^4 - 10x^3 + 36x^2 - 58x + 35 = 0$$

```
a=[ -10 _36 58 _35, (=~/i.3),.0
```

```
10 _36 58 _35
```

```
1 0 0 0
```

```
0 1 0 0
```

```
0 0 1 0
```

```
char_lf a
```

```
++-----+
```

```
|1|4.41421 2j1 2j_1 1.58579|35 _58 36 _10 1|
```

```
++-----+
```

```
p. 35 _58 36 _10 1
```

```
++-----+
```

```
|1|4.41421 2j1 2j_1 1.58579|
```

```
++-----+
```

6.6.2 Reference

小国 力 「新数値計算法」サイエンス社 1997

6.7 混合配列とボックス

APL と J を開発した K.E.IVERSON は数学者でコンピューター科学者であった。IBM に在籍して APL を考案していたときも数学のルールを優先し煩瑣なコンピュータのルールを省力した。従って、APL や J には配列宣言や配列の型の指定はない。数値を入れれば数列であるし文字を入れれば文字列として取り扱う。APL2 では数列と文字列の混合配列をサポートした。J はボックスと言う第三の型を用意した。

6.7.1 ボックス

ボックスを利用すると多次元配列が平面に並ぶ。次元（ランク）が簡単に見えるようになり ($L:0$) でボックス内の並列演算ができる。

メンデルの法則

母の状態と父の状態はそれぞれ
W W /W R/ R R
(beans like ingen) White and Red
○○, ○●, ●●

```
] a=. 0 0;0 1; 1 1
+---+---+---+
|0 0|0 1|1 1|
+---+---+---+
```

9 組の組み合わせができる。

1. 演算は 4 方向に足し合わせる。
2. 0/1/2 の数を数える

```
mendel_sub0 ''
+---+---+---+
|0 0|0 0|0 0|
|0 0|0 1|1 1|
+---+---+---+
|0 1|0 1|0 1|
|0 0|0 1|1 1|
+---+---+---+
|1 1|1 1|1 1|
|0 0|0 1|1 1|
+---+---+---+
```

```

a          b
↓  ↘  ↙  ↓
c          d

mendel_sub1 ''
+-----+-----+-----+
|0 0 0 0|0 1 0 1|1 1 1 1|
+-----+-----+-----+
|1 0 0 1|1 1 0 2|2 1 1 2|
+-----+-----+-----+
|1 1 1 1|1 2 1 2|2 2 2 2|
+-----+-----+-----+
```

子の状態の確率を求める。MM FF はビット、子の状態は確率

MM FF | ○○, ○●, ●●, ●●

```
mendel ''
+-----+-----+-----+
|0 0 0 0|1 0 0      |
+-----+-----+-----+
|0 0 0 1|0.5 0.5 0  |
```

```

+-----+-----+
|0 0 1 1|0 1 0      |
+-----+-----+
|0 1 0 0|0.5 0.5 0    |
+-----+-----+
|0 1 0 1|0.25 0.5 0.25|
+-----+-----+
|0 1 1 1|0 0.5 0.5    |
+-----+-----+
|1 1 0 0|0 1 0      |
+-----+-----+
|1 1 0 1|0 0.5 0.5    |
+-----+-----+
|1 1 1 1|0 0 1      |
+-----+-----+
M      F  |White Medium Red

```

Script

```
diag=: (<0 1)&|: NB. trace
```

```
mendel=: 3 : 0
```

```
NB. find mendel color rate
```

```
NB. Usage: mendel ''
```

```
MAT=:./:~ L:0 mendel_sub1 ''
```

```
TMP=: {4 %~ |: 3 9 $ ;; +/ L:0( 3#<(.,MAT))e. L:0 {@> 0 1 2
```

```
NB. sum-up 0/1/2
```

```
(,;L:0 mendel_sub0''),. TMP NB. add index for view
```

```
)
```

```
mendel_sub0=: 3 : 0
```

```
NB. make mat
```

```
NB. usage: mendel_sub0 ''
```

```
TMP=: 0 0;0 1; 1 1
```

```
TMP,: L:0 ;("1),.3#<TMP NB. copy and raminare
```

```
)
```

```
mendel_sub1=: 3 : 0
```

```
NB. usage: mendel_sub1 ''
```

```
MAT=: mendel_sub0 ''
```

```
((+/& diag&|. ),(+/&diag ), +/ ) L:0 MAT NB. calc 4 direction
```

)

6.7.2 関数をボックスに入れて並列計算

鶴亀算

$$\begin{cases} x + y = 10 & \rightarrow y = 10 - x \\ 2x + 4y = 34 & \rightarrow y = 8.5 - 0.5x \end{cases}$$

多項式の関数 p . を基底に
用いて並列でグラフ用の計算を行う。

```
f0=: 10 _1 &p.
```

```
f1=: 8.5 _0.5&p.
```

```
.,(f0;f1) L:0 i.10
```

```
+-----+
|10 9 8 7 6 5 4 3 2 1|
+-----+
|8.5 8 7.5 7 6.5 6 5.5 5 4.5 4|
+-----+
```

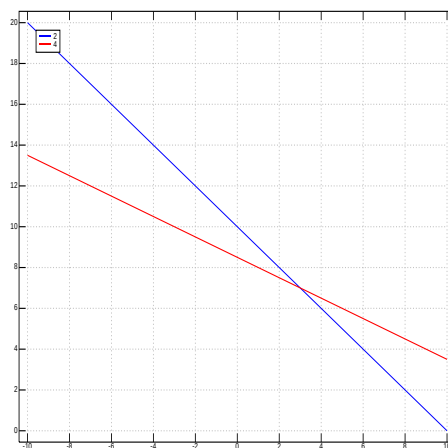


図 6.4 鶴亀

```
require 'plot numeric trig'
'key 2 4' plot tmp ; ;("1),.(f0;f1) L:0 tmp =.i:10
pd 'eps /temp/cr_tt0.eps'
```

6.8 ネステッド・アレー

nest は巢である。*APL2* のサポートするネステッド・アレーを *J* は未だサポートしない。欲しいのはネストされた逆行列であるがこれは *APL2* もサポートしていない。ネストから開く関数を作成して見よう。

6.8.1 EXPAND

次の左の行列は次で入力できる。

それを右のように拡張したい。そして *Box* を開いて逆行列を求めてしまう。

*3

```
NM2=:12; 4; 3; 0; 0; 4;(4*=/~i. 3); 1; 1; 0; 3; 1;(3*=/~i. 4)
NM2=:5 5 $ NM2,0; 1; 0; 1; 0; 0; 0; 0; 0; 1; 0; 0
```

```

      NM2
+---+-----+-----+---+
|12|4      |3      |0|0|
+---+-----+-----+---+
|4 |4 0 0|1      |1|0|
|  |0 4 0|      | | |
|  |0 0 4|      | | |
+---+-----+-----+---+
|3 |1      |3 0 0 0|0|1|
|  |      |0 3 0 0| | |
|  |      |0 0 3 0| | |
|  |      |0 0 0 3| | |
+---+-----+-----+---+
|0 |1      |0      |0|0|
+---+-----+-----+---+
|0 |0      |1      |0|0|
+---+-----+-----+---+

```

```

      exp_nest NM2
+---+-----+-----+---+
|12|4 4 4|3 3 3 3|0|0|
+---+-----+-----+---+
|4 |4 0 0|1 1 1 1|1|0|
|4 |0 4 0|1 1 1 1|1|0|
|4 |0 0 4|1 1 1 1|1|0|
+---+-----+-----+---+
|3 |1 1 1|3 0 0 0|0|1|
|3 |1 1 1|0 3 0 0|0|1|
|3 |1 1 1|0 0 3 0|0|1|
|3 |1 1 1|0 0 0 3|0|1|
+---+-----+-----+---+
|0 |1 1 1|0 0 0 0|0|0|
+---+-----+-----+---+
|0 |0 0 0|1 1 1 1|0|0|
+---+-----+-----+---+

```

exp_nest と *nut_crack* は正方形の行列用である。(縦長用に拡張は容易である。)

ループで丁重にボックスを開く *nut_crack* を用意した。

*4

*3 ネストの状態が無理をしても逆行列の元が取れているかどうか不安がつきまとう。IBM もやっていないことは即座に解決できないでしょう。

*4 配列が複雑で;"2) で巧く開けない

nut_crack A exp_nest NM2

```
12 4 4 4 3 3 3 3 0 0
  4 4 0 0 1 1 1 1 1 0
  4 0 4 0 1 1 1 1 1 0
  4 0 0 4 1 1 1 1 1 0
  3 1 1 1 3 0 0 0 0 1
  3 1 1 1 0 3 0 0 0 1
  3 1 1 1 0 0 3 0 0 1
  3 1 1 1 0 0 0 3 0 1
  0 1 1 1 0 0 0 0 0 0
  0 0 0 0 1 1 1 1 0 0
```

6.9 2 元分散分析

肥料と収穫の2元分散分析の例を先の *exp.mat* と *nut.crack* を用いてシステム方程式に展開して解いてみよう。

D680		地 域				収 穫
		I	II	III	IV	
肥料	A	30	23	26	29	α_1
	B	22	18	21	19	α_2
	C	29	25	31	27	α_3
収穫		β_0	β_1	β_3	β_4	

出典 鈴木

モデルは次の4ケースを比較する

M1, M2, M3 は一元分散分析である。

$$M1 \quad \alpha_0 = \alpha_1 = \alpha_2 \quad = 0$$

$$\beta_0 = \beta_1 = \beta_2 = \beta_4 \quad = 0$$

$$M2 \quad \beta_0 = \beta_1 = \beta_2 = \beta_4 \quad = 0$$

$$M3 \quad \alpha_0 = \alpha_1 = \alpha_2 \quad = 0$$

$$M4 \quad \text{少なくとも一つの}\alpha \text{も } \beta \text{ も } 0 \text{ でない}$$

6.9.1 方程式

最尤推定は次の方程式を解いて得られる。

$$\begin{bmatrix}
 12 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 0 & 0 \\
 4 & 4 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
 4 & 0 & 4 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
 4 & 0 & 0 & 4 & 1 & 1 & 1 & 1 & 1 & 0 \\
 3 & 1 & 1 & 1 & 3 & 0 & 0 & 0 & 0 & 1 \\
 3 & 1 & 1 & 1 & 0 & 3 & 0 & 0 & 0 & 1 \\
 3 & 1 & 1 & 1 & 0 & 0 & 3 & 0 & 0 & 1 \\
 3 & 1 & 1 & 1 & 0 & 0 & 0 & 3 & 0 & 1 \\
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0
 \end{bmatrix}
 \times
 \begin{bmatrix}
 \mu \\
 \alpha_1 \\
 \alpha_2 \\
 \alpha_3 \\
 \beta_1 \\
 \beta_2 \\
 \beta_3 \\
 \beta_4 \\
 \lambda \\
 \nu
 \end{bmatrix}
 =
 \begin{bmatrix}
 \sum x_i & + \sum y_i & + \sum z_i \\
 \sum x_i & & & & & & & & & \\
 & \sum y_i & & & & & & & & \\
 & & \sum z_i & & & & & & & \\
 x_1 & +y_1 & +z_1 & & & & & & & \\
 x_2 & +y_2 & +z_2 & & & & & & & \\
 x_3 & +y_3 & +z_3 & & & & & & & \\
 x_4 & +y_4 & +z_4 & & & & & & & \\
 0 & & & & & & & & & \\
 0 & & & & & & & & &
 \end{bmatrix}$$

AIC

AIC は次により求める

$$MLL = -\frac{12}{2}(1 + \log 2\pi\sigma^2)$$

$$AIC(M4) = -2 \times MLL + 2 \times 7$$

計算

$$\begin{bmatrix} \sum x_i & + \sum y_i & + \sum z_i \\ \sum x_i & & \\ & \sum y_i & \\ & & \sum z_i \\ x_1 & +y_1 & +z_1 \\ x_2 & +y_2 & +z_2 \\ x_3 & +y_3 & +z_3 \\ x_4 & +y_4 & +z_4 \\ & 0 & \\ & 0 & \end{bmatrix} \text{を求める}$$

mk_Y D680

300 108 80 112 81 66 78 75 0 0

 $\mu, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \beta_4, \lambda, \nu$ を求める

25 2 _5 3 2 _3 1 0 0 0

2元分析のモデル M4

D680 ava_m2 NM2

```

+-----+-----+-----+
|52.9201|1.5|25 2 _5 3 2 _3 1 0|
+-----+-----+-----+

```

D680 ava_mx NM2

```

+-----+-----+-----+
|72.5147|17.6667|25          | NB. M1
+-----+-----+-----+
|61.3678|5      |25 2 _5 3      | NB. M2
+-----+-----+-----+
|75.8652|14.1667|25 2 _3 1 0    | NB. M3
+-----+-----+-----+
|52.9201|1.5    |25 2 _5 3 2 _3 1 0| NB. M4
+-----+-----+-----+

```

M4 のモデルが AIC が一番小さいので選択する。収穫量は肥料と地域の 2 要因が効いている。

.0.2 同時方程式/Klein の数値

NB.	YEAR	C*	P	W	I	(K-1)	X	W'	G	T
	1920	39.8	12.7	28.8	2.7	180.1	44.9	2.2	2.4	3.4
	1921	41.9	12.4	25.5	-0.2	182.8	45.6	2.7	3.9	7.7
	1922	45.0	16.9	29.3	1.9	182.6	50.1	2.9	3.2	3.9
	1923	49.2	18.4	34.1	5.2	184.5	57.2	2.9	2.8	4.7
	1924	50.6	19.4	33.9	3.0	189.7	57.1	3.1	3.5	3.8
	1925	52.6	20.1	35.4	5.1	192.7	61.0	3.2	3.3	5.5
	1926	55.1	19.6	37.4	5.6	197.8	64.0	3.3	3.3	7.0
	1927	56.2	19.8	37.9	4.2	203.4	64.4	3.6	4.0	6.7
	1928	57.3	21.1	39.2	3.0	207.6	64.5	3.7	4.2	4.2
	1929	57.8	21.7	41.3	5.1	210.6	67.0	4.0	4.1	4.0
	1930	55.0	15.6	37.9	1.0	215.7	61.2	4.2	5.2	7.7
	1931	50.9	11.4	34.5	-3.4	216.7	53.4	4.8	5.9	7.5
	1932	45.6	7.0	29.0	-6.2	213.3	44.3	5.3	4.9	8.3
	1933	46.5	11.2	28.5	-5.1	207.1	45.1	5.6	3.7	5.4
	1934	48.7	12.3	30.6	-3.0	202.0	49.7	6.0	4.0	6.8
	1935	51.3	14.0	33.2	-1.3	199.0	54.4	6.1	4.4	7.2
	1936	57.7	17.6	36.8	2.1	197.7	62.7	7.4	2.9	8.3
	1937	58.7	17.3	41.0	2.0	199.8	65.0	6.7	4.3	6.7
	1938	57.5	15.3	38.2	-1.9	201.8	60.9	7.7	5.3	7.4
	1939	61.6	19.0	41.6	1.3	199.9	69.5	7.8	6.6	8.9
	1940	65.0	21.1	45.0	3.3	201.2	75.7	8.0	7.4	9.6
	1941	69.7	23.5	53.3	4.9	204.5	88.4	8.5	13.8	11.6

Source D.N.Gjarati,Basic Econometrics 4th Edition Table 20.5