計量経済
(3 行のスクリプトで始める)
未定稿

Masato Shimura
jcd02773@nifty.ne.jp

2005 年 5 月 31 日

目次

はじめに

ハーバード大学の Howard Aiken は 1944 年に Mark I, 1947 年に MarkII という真空管式コンピュータを完成させ、大学に Computer Labolatory を設けた。。IBM がパートナーであった。

1949 年の晩夏にハーバード大学教授 Wassily Leontief は注意深くパンチカードを MARK II にかけていた。

2 年間の調査を経て、合衆国の 25 万件の経済データを 500 部門に分割して、その関連を調査しようとしていた。しかし、当時最大のコンピューター MARKII では 500 変数の問題は処理できなかった。Leontief はセクタを 42 に圧縮した。プログラミングには数ヶ月を要した。

MARKII は 56 時間うなりを上げて、解を出した。ハーバードにおける Leontief の産業連関表の計算はコンピューターを活用した大規模な数学モデルの最初の事例の一つであった。Leontief の騒音と熱の中での産業連関表のモデルを考案した功績は 1973 年にノーベル経済学賞に輝いた。

[David C. Lay Linear Algebra and its Application]

Keneth E. Iverson (1920-2004) はカナダのアルバータ州に生まれ、第 2 次対戦にフライトエンジニアとして従軍した後、クイーンズ大学からハーバード大学へ進み、ハーバード大学で数学とコンピューターによる数学解析を教えた。1962 年には A Programming Language というタイトルの本を著した。このアイデアを IBM System360 という IBM の歴史を彩る機に搭載するため、IBM に移り、およそ 10 年の机上でのアイデアをアルゴリズムに練り上げる作業の後、パワーが追いついた大型機に APL が搭載された。大凡トランジスタの時代である。

コンピューターが出た少し後のコンピューター言語は FORTRAN と COBOL であって今も健在である。COBOL は事務処理専用、科学技術計算は FORTRAN が主に受け持った。FORTRAN は FORTRAN90/95 に進化し、スーパーコンピューターなどの科学技術計算分野で活躍している。

Iverson は数学者で、数学をベースに言語のコアの部分の設計を一人で行った。その一つがベクトルやマトリクスをスカラと同じように取り扱う一般多元配列環を装備し、線形数学などを自由に取り扱う事である。

当時の FORTRAN は少ないメモリと資源を活用するため、CPU の動作原理を優先させ、メモリーから CPU のレジスターへ順次転送にする手順を人手で逐次記述させ、人の思考

回路をコンピュータに合わせるように強いた。このルールは C/C++ や JAVA、PASCAL にも継承された。

一般多元配列環を装備し、コードを英語でなく数学記号に似せて新たに作成した記号で記述する言語として、APL が開発された。、

ムーアの法則により CPU は絶え間なく進化してきた。最近は少し変調のようであるが、私達は既に膨大な資源を手にしている。少し前の数値計算の大家の著書には逆行列 (ベクトルやマトリクスの除算) を用いることに腰が引けているものが見受けられる。100 年分の月次レベルの経済データ程度ならキーを押した瞬間に逆行列が表示される。

Iverson によってネット時代に合うように APL 文字をキーボード記号に移し替え、自由に各自の言語で再定義もできる J 言語が 1989 年に誕生し、改良を重ね進化している。(この J 言語のユーザーライセンスは現在は無料である。)

本稿は計量経済の主要理論を簡潔に記述し、即座に利用できるようにした。K.E.Iverson の遺した関数を活用すると多くの計量経済の関数は一行で記述できる。従って、数式も相応に簡素に記述した。

計量経済が難しいのではなく便利な道具に出会わなかったのである。

言語は専門家に任せアプリケーションのユーザーに回るのもよいし、言語を駆使して数学や科学技術計算の先端理論に頭脳をフル回転させて挑戦してみるのも良いことだ。

1　3 本のスクリプト

<table>
<tr><td>単回帰
重回帰
OLS</td><td><code>regx=: 3 : ’({:"1 y.) %. 1,.}:"1 y.’</code></td></tr>
<tr><td>多項式
Poli
-nominal</td><td><code>poly1=: 4 : ’y. %. (>:i. # y.) ^/ i. >: x.’</code></td></tr>
<tr><td>自己回帰
AR</td><td><code>ar0=: 4 : ’(x.}.tmp) %. |.("1)}: >x.<\ tmp=:y.-(+/%#)y.’</code></td></tr>
</table>

1.1　OLS 単回帰・重回帰モデル

$$
\mathrm{y}=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_n \end{bmatrix} \quad , \quad \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & x_{13} & x_{23} & \cdots & x_{k3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}
$$

1.2　多項式モデル

変数 y が t に関する k 次の多項式によって

$y = C_{00} + C_{01}t + \cdots + c_{0k}t^k + \epsilon$

のように表されている場合に、パラメータの最尤推定は次の式を解くことによって得られる。

$$\begin{bmatrix} S_0 & S_1 & \cdots & S_k \\ S_1 & S_2 & \cdots & S_{k+1} \\ S_2 & S_3 & \cdots & S_{k+2} \\ \vdots & \vdots & \ddots & \vdots \\ S_K & S_{K+1} & \cdots & S_{2k} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdots \\ c_K \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \cdots \\ T_k \end{bmatrix}$$

1.3 AR モデル

M 次までの AR モデルの最小二乗解をすべて求めるには、次のようにすればよい。この行列を逆行列を用いれば一気に最小二乗解が得られる。

$$\begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1k} \\ S_{12} & S_{22} & \cdots & S_{2k} \\ S_{13} & S_{23} & \cdots & S_{3k} \\ \vdots & \vdots & \ddots & \vdots \\ S_{1K} & S_{2K} & \cdots & S_{kk} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_K \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \cdots \\ T_k \end{bmatrix}$$

1.4 数学ノート

除算の歴史

現在の人類がアフリカを出て大凡 4 万年を経て、突然エジプトの地にハイテク古代文明が出現した。エッフェル塔に抜かれるまで最高を誇った巨大ピラミッドや神殿、ナイル河の測量、メソポタミアの灌漑設備、フェニキア人の大型船、ローマのアーチ橋.. 石や日干し煉瓦と人力で作り上げる完成された技術があった。

BC4000 年には 30 日 ×12 月 +5 祝日のエジプト暦が作成され、1582 年のグレゴリオ歴までの間精度で優位に立った。ナイル河の測量では水位 0point は知られていた。

BC1600 年頃の「リンド · パピルス」と言われる数学書が発見されている。

BC2000 年頃のメソポタミアでは、60 進法で計算が行われ、棒で書かれ乾た赤褐色の粘土板には乗法と特殊な逆数を用いた乗法が書き残されている。

暦、交易、課税、戦争、 人の営みは絶え間なく繰り返され、抽象化の進んだ現代数学とは異なる実用の要請があった。

2　単回帰モデルと重回帰モデル

単回帰の正規方程式と重回帰モデルの正規方程式は次のようにあらわされる。

$(X'X)\hat{\beta} = X'y$

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

これらは、更に次のように簡約される。

$(X'X)^{-1}X'y = \dfrac{X'y}{X'X} = \dfrac{y}{X}$

線型数学で表現すれば単回帰も重回帰も同じである。配列計算言語では、単回帰も重回帰も同じスクリプトで記述でき、入力データにより、単回帰、重回帰のどちらでも簡単に計算できる。

2.1　単回帰モデル

Working Example

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」）

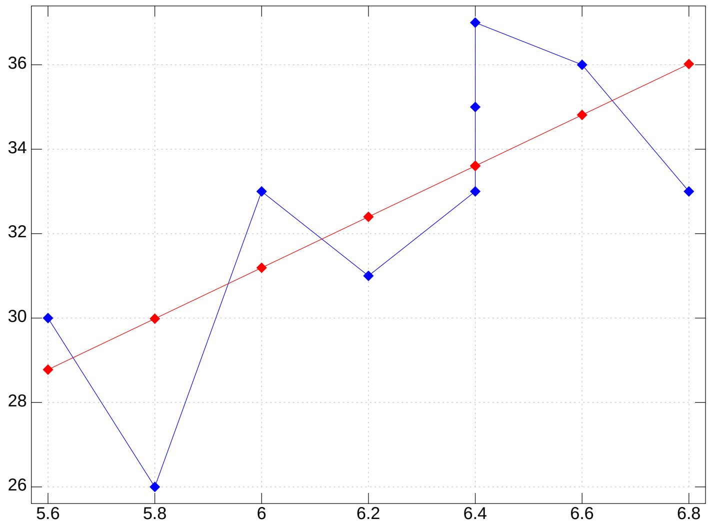

図 1　果実の直径と水分含有率

KY KX	1 KX	$(1KX)^{-1}$
	定数項を求めるために X に 1 を付加する	逆行列を求める。

`  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`
	`   1,.KX`
	`1 5.6`
	`1 5.8`
	`1   6`
	`1 6.2`
	`1 6.4`
	`1 6.4`
	`1 6.4`
	`1 6.6`
	`1 6.8`
		`  |: %. 1,.KX`
		`  3.51504 _0.545113`
		`  2.45865  _0.37594`
		`  1.40226 _0.206767`
		` 0.345865 _0.037594`
		`_0.710526  0.131579`
		`_0.710526  0.131579`
		`_0.710526  0.131579`
		` _1.76692  0.300752`
		` _2.82331  0.469925`

サンプルデータ KX の逆行列を手計算で行おうとすれば大変である。世の教科書には正方行列の数学ノートに記載した方法が紹介されているが、このような縦長の行列は扱っていない。ここでは、素直に $K.E.Iverson$ の $matrix - divide$(`%.`) に従おう。

$\frac{y}{X}$	$\frac{KY}{1,.KX}$ 両項の `%.` はマトリクスの除算を行う	`   KY %. 1,.KX` `_5.01128 6.03383`
yX^{-1}	$KY\ +/.*(1,.KX)^{-1}$ マトリクスの除算は逆行列の内積と同じである。	`KY +/ . *    |: %. 1,.KX` `_5.01128 6.03383`

	*reg*0 *y*.	`reg0 KD`
		`_5.01128 6.03383`
		
		$y = -5.01128 + 6.03383x$

2.2　重相関モデル

線形数学では単相関と重相関は同じ方程式で表すことが出来た。配列計算言語では、同じプログラムで計算できる。

Warking Example 発電量と経済指数 (1969-1994)

```
 reg0 1 2 3 6 pick_reg_n a
_3.10519 2.74842 0.860231 0.323955
```

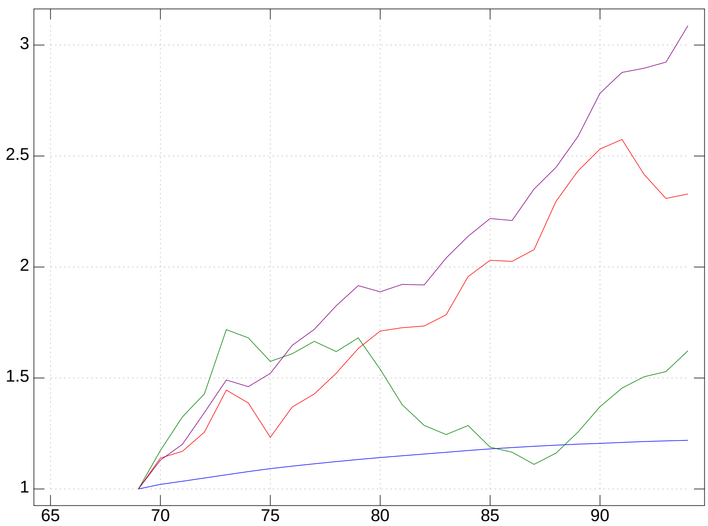

図 2　上から発電量 鉱工業生産指数 原油輸入量 人口

y 発電量

x_1 人口

x_2 鉱工業生産指数

x_3 原油輸入量

データは初年度 (1969) を 1 として指数化してある。(出典 杉原 敏夫「適応的モデルによる経済時系列分析」1996)

$y = -3.10519 + 2.74842x_1 + 0.860231x_2 + 0.323955x_3$

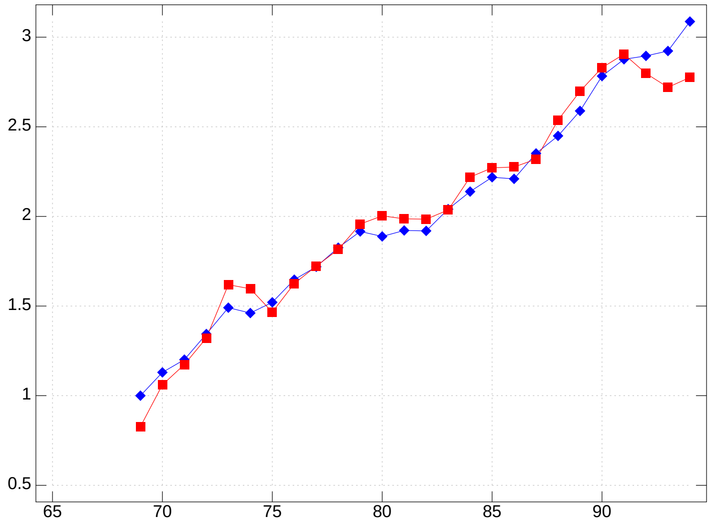

図 3　発電量の実数と推計 (赤) 推計値が失速している

2.3　モデルの選択

モデルは妥当性をアピールして、審判団の審査を受けなければならない。

判定法	paneler	
相関係数		
AIC	赤池 弘次	
t(student)	Cozett(Guiness 社)	
p		
DW	Durbin,Watson	

2.3.1　t

先の発電量に関する重相関モデルの採点結果である。

```
   reg0 reg_panel_ad  b0
+------+-----------------------------------+
|f=    |_3.10519 2.74842 0.860231 0.323955| 回帰係数
+------+-----------------------------------+
|corr=:|96.8057                            | 相関係数
+------+-----------------------------------+
|AIC:  |_109.527                           | AIC 値
+------+-----------------------------------+
|DW=   |0.582066                           | Durbin Watson 値
+------+-----------------------------------+
|t=:   |_3.25926 2.56923 5.78009 2.88985   | t 値
+------+-----------------------------------+
```

2.4　対数

スコットランド人 *John Napier*(1550 – 1617) が「驚くべき対数法則の記述」を著したのは 1614 年であった。

2.5　タグ付き変数

3　多項式

```
                  4 poly1 s2

_2.01542 5.65633 _0.889141 0.0485201 _0.000848678
```

$y = -2.01542 + 5.65633x - 0.889141x^2 + 0.0485201x^3 - 0.000848678x^4$

データが長いと紙面を取るので、説明のため、10 個の乱数データを用意した。

<table>
<tr>
<td>X, .a
X は単に順序数でオリジンを 1 とした。</td>
<td>X の 0 1 2 3 乗</td>
<td>左の逆行列</td>
</tr>
<tr>
<td>(>:i. # a) ,.a
1 6
2 3
3 19
4 15
5 10
6 14
7 0
8 7
9 12
10 17</td>
<td>(>:i.#a)^/ i.>: 3

1 1 1 1
1 2 4 8
1 3 9 27
1 4 16 64
1 5 25 125
1 6 36 216
1 7 49 343
1 8 64 512
1 9 81 729
1 10 100 1000</td>
<td>6.2 ": |: %. (>:i.#a)^/ i.>: 3
1.60 _0.93 0.16 _0.01
0.27 0.08 _0.04 0.00
_0.40 0.53 _0.12 0.01
_0.57 0.57 _0.11 0.01
_0.40 0.34 _0.05 0.00
_0.07 0.00 0.02 0.00
0.27 _0.32 0.09 _0.01
0.43 _0.45 0.11 _0.01
0.27 _0.25 0.05 0.00
_0.40 0.43 _0.11 0.01</td>
</tr>
</table>

```
   a %. (>:i.#a)^/ i.>: 3
_11.2 17.6678 _3.75583 0.227855
```

$$y = -11.2 + 17.6678x - 3.75583x_2 + 0.227855x_3$$

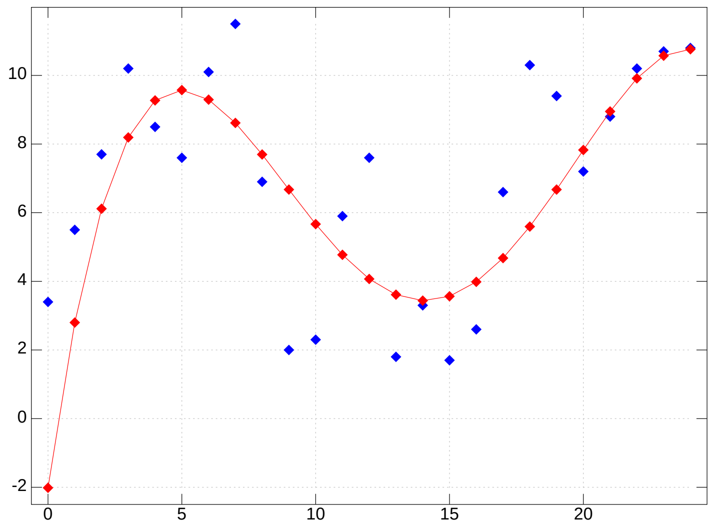

図 4　東京の冬期の最低気温

4　自己回帰

自己回帰のアルゴリズムには $Yull - Walker$ 法、$Burk$ 法、$Householder$ 法があるが、ここでは、$Householder$ のデータ構成を用いるが $Householder$ 変換を経由しないで、逆行列を用いて、一度に計算してしまうシンプルな方法を用いてみよう。

なお、3 方式のアルゴリズムによる回帰係数には若干の差があり、$Burk$ 法はピークに強く、$Yull - Walker$ 法は谷に強い。$householder$ 法は中立的であると言われている。

最初の 12 個のデータで自己回帰の仕組みを再現してみよう。

$y, x_{t-0}, x_{t-1}, x_{t-2}, x_{t-3}$ という型のデータの組み合わせ (X) を作る。

次数相当分のデータ (y) が組み合わせの結果減少するので、先頭から落として個数を合わせる。

$\dfrac{y}{X}$ を計算する。X の逆行列が求まるので、簡単なマトリクスの除算で係数が求まる。

<table>
<tr><td>データの最初から12個
NR, y</td><td>$y, x_{t-0}, x_{t-1}, x_{t-2}, x_{t-3}$</td><td></td></tr>
<tr><td>(>:i. $ a),.
1 3.4
2 5.5
3 7.7
4 10.2
5 8.5
6 7.6
7 10.1
8 11.5
9 6.9
10 2
11 2.3
12 5.9</td><td>a(2}. a),. |.("1) 3>\ a
7.7 7.7 5.5 3.4
10.2 10.2 7.7 5.5
8.5 8.5 10.2 7.7
7.6 7.6 8.5 10.2
10.1 10.1 7.6 8.5
11.5 11.5 10.1 7.6
6.9 6.9 11.5 10.1
2 2 6.9 11.5
2.3 2.3 2 6.9
5.9 5.9 2.3 2</td><td></td></tr>
</table>

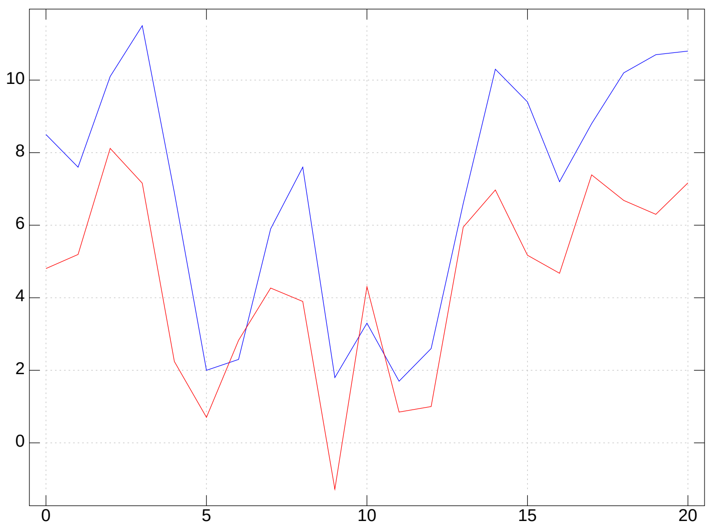

図 5　東京の冬期の最低気温