産業聯關表 分析에 Partitioned Form의 應用

金　英　錄
(KIET 電算室)

◀ 目 次 ▶

Ⅰ. 序 論
Ⅱ. 問題點
Ⅲ. 計算過程
Ⅳ. Bordered Form
Ⅴ. 實驗 및 結論
〈附 錄〉

Ⅰ. 序 論

産業 聯關表(表Ⅰ－1)는 韓國銀行에서 우리나라 産業 構造와 聯關 관계를 表로 나타내서 産業 部門別 分析과 構造 分析 等을 쉽게 할 수 있도록 만들어 놓은 것이다.

이 表는 input과 output을 일치시켜 놓아 한눈에 構造와 흐름 等을 알 수 있다.

이 表를 利用하여 여러가지 分析을 하려고 할 때 필요한 計算 部分으로 逆行列係數(表Ⅰ－2)를 구하는 과정을 반드시 거치게 되는데, 이 과정을 가능한 效果的으로 처리하도록 하는 데 목적이 있으며 특히 전자계산기를 이용하여 처리함에 따른 問題點을 中心으로 알아 보겠다.

〈表Ⅰ－1〉　　1978年 産業聯關表

單位 : 10億원

		內生部門				外生部門				輸入(控除)	總産出額
		農林漁業	鑛工業	其他産業	中間需要計	消費	投資	輸出	最終需要計		
內生部門	農林漁業	711.3	1,914.4	136.5	2,762.2	4,510.0	162.0	325.5	4,997.5	1,108.6	6,651.1
	總工業	821.1	12,327.4	3,991.8	17,140.3	5,679.5	3,339.9	5,428.3	14,447.7	7,750.2	23,837.8
	其他産業	229.9	2,992.9	3,484.4	6,707.2	7,675.3	4,000.7	1,359.5	13,035.5	358.3	19,384.4
	中間投入計	1,762.3	17,234.7	7,612.7	26,609.7	17,864.8	7,502.6	7,113.3	32,480.7	9,217.1	49,873.3
外生部門	附加價値計	4,888.8	6,602.9	11,771.9	23,263.6						
總投入額		6,651.1	23,837.6	19,384.6	49,873.3						

〈表 I－2〉 逆行列係數의 意味

投入部門 / 産出部門	1 部 門	2 部 門	行 計
1 部門	r_{11} : 1 部門의 最終需要 1 單位를 充足하기 위하여 直間接으로 必要한 1 部門의 産出額	r_{12} : 2 部門의 最終需要 1 單位를 充足하기 위하여 間接으로 必要한 1 部門의 産出額	$r_{11}+r_{12}$: 各 部門에서 1 單位씩의 最終需要가 發生했을 때 이를 充足하기 위한 1 部門의 産出額
2 部門	r_{21} : 1 部門의 最終需要 1 單位를 充足하기 위하여 間接으로 必要한 2 部門의 産出額	r_{22} : 2 部門의 最終需要 1 單位를 充足하기 위하여 直間接으로 必要한 2 部門의 産出額	$r_{21}+_{22}$: 各 部門에서 1 單位씩의 最終需要가 發生했을 때 이를 充足하기 위한 2 部門의 産出額
列 計	$r_{11}+r_{21}$: 1 部門의 最終需要 1 單位를 充足하기 위하여 直間接으로 必要한 全部門의 産出額	$r_{12}+r_{22}$: 2 部門의 最終需要 1 單位를 充足하기 위하여 直間接으로 必要한 全部門의 産出額	

〈表 I－3〉 逆行列係數$(I-A)^{-1}$ 型

	農林漁業	鑛工業	其他産業	行 計
農 林 漁 業	1.1508	0.2075	0.0619	1.4202
鑛 工 業	0.3370	2.2762	0.5743	3.1875
其 他 産 業	0.1001	0.3573	1.3098	1.7672
列 計	1.5879	2.8410	1.9460	6.3749

II. 問題点

産業 聯關表를 利用하여 分析할 경우 各表의 Inverse Matrix를 구해야 모든 分析이 가능해짐은 周知의 事實이다.

이러한 Inverse Matrix를 求하기 위해 컴퓨터를 必須的으로 利用하는 데 다음과 같은 제약조건이 발생된다.

첫째, 分類를 細分化하여 처리할 경우, 즉, 392 分類로 처리할 경우 Matrix가 커져 계산 時間을 증가시킨다.

다시 말하면 컴퓨터가 순수하게 計算에 使用하는 時間보다 IO 時間이 더 크게 된다.

둘째, 컴퓨터가 IO 技能만 擴張되어 Multi－Job을 처리할 경우, 다른 作業에 미치는 영향이 커지게 된다.

실제 이러한 作業이 처리되는 동안, 다른 作業은 計算에 必要한 機會를 거의 갖지 못하게 되어 待期 狀態에 들어가게 된다.

세째, 이렇게 해서 얻어진 결과를 최종 분석에 利用할 때 그 값의 信賴度가 의심

스럽게 된다.

즉, 計算 時間과 計算 結果의 信頼性이 問題가 되므로 좀 더 나은 方法을 통해 改良하도록 하고자 한다.

여기서 使用한 컴퓨터는 IBM 4341-GⅡ로서 Main Memory가 4MB이다.

Ⅲ. 計算過程

産業 聯關表는 行과 列의 밸런스 表이므로 投入係數(表Ⅲ-1)를 매개로 한 産出과 投入의 관계를 다음의 一聯의 연립 방정식으로 表示할 수 있다.

즉, 競爭 輸入型의 경우

$$A_{11}X_1 + A_{12}X_2 + \cdots + A_{ij}X_j + \cdots + a_{in}X_n + Y_1 - M_1 = X_1$$

$$\vdots$$

$$a_{i1}X_1 + a_{i2}X_2 + \cdots + a_{ij}X_j + \cdots + a_{in}X_n + Y_i - M_i = X_i$$

$$\vdots$$

$$a_{n1}X_1 + a_{n2}X_2 + \cdots + a_{nj}X_j + \cdots + a_{nn}X_n + Y_n - M_n = X_n$$

$$\begin{cases} a_{ij} : \text{投入 係數} \\ X_i : i\text{部門의 産出額} \\ Y_i : i \quad // \quad \text{最終需要} \\ M_i : i \quad // \quad \text{輸入} \end{cases}$$

行列로 表記하면

$$AX - Y - M = X$$

단, $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \\ a_{ni} & \cdots & a_{nn} \end{pmatrix}$ $X = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$

$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$ $M = \begin{pmatrix} M_1 \\ \vdots \\ M_n \end{pmatrix}$ $I = \begin{pmatrix} 1 & \cdots 0 & 0 \\ 0 & \ddots & \\ 0 & 0 & \cdots 1 \end{pmatrix}$

〈表Ⅲ-1〉 投 入 係 數 表

	農林漁業	鑛 工 業	其他産業
農 林 漁 業	0.1069	0.0803	0.0070
鑛 工 業	0.1235	0.5171	0 2059
其 他 産 業	0.0346	0.1256	0.1798
中間投入計	0.2650	0.7230	0.3927
附加價値計	0.7350	0.2770	0.6073
總 投 入 計	1.0000	1.0000	1.0000

이 式을 X에 대해 풀면

$$X = (I - A)^{-1} (Y - M)$$

그러므로 Inverse Matrix $(I - A)^{-1}$ 을 求해야 한다.

이와 비슷한 계산과정을 각 表마다 반복하게 된다.

보통 Inverse Matrix를 구하는 데 使用되는 Subroutine MINV는 Gauss-Jordan Method를 利用한 것이며 IBM의 Scientific Subroutine Package 中에 들어 있다.

이 Subroutine이 Powerful한 기능을 갖고 있음은 周知의 사실이나 Raw Matrix의 크기가 앞에서 提示한 問題點을 갖고 있다.

그래서 Inverse Matrix를 구하는 과정에 Partitioned Form인 Bordered Form을 導入하여 적용하고자 한다.

Ⅳ. Bordered Form

Matrix A를 Partioned Form으로 表示할 경우,

$$A = \begin{bmatrix} PQ \\ RS \end{bmatrix}$$

로 Submatrix P,Q,R,S를 만들 수 있다.

여기서 A와 P가 Nonsingular 이라면

$$A^{-1} = \begin{bmatrix} X & -P^{-1}QW \\ -WRP^{-1} & W \end{bmatrix}$$

로 나타낼 수 있다.

단, $W = (S - RP^{-1}Q)^{-1}$

$X = P^{-1} + P^{-1}QWRP^{-1}$

마찬가지로

A와 S가 Nonsingular 라면

$$A^{-1} = \begin{bmatrix} X & -XQS^{-1} \\ -S^{-1}RX & W \end{bmatrix}$$ 로

나타낼 수 있다.

단, $\begin{cases} X = (P - QS^{-1}R)^{-1} \\ W = S^{-1} + S^{-1}R \times QS^{-1} \end{cases}$

특히 産業 聯關表의 投入 係數를 利用하여 Inverse Matrix를 구하게 되는 Raw Matrix에서 P와 S, 그리고 A가 모두 Nonsingular이기 때문에 더욱 計算이 알기 쉽다.

단지 그 계산 결과가 Non-Partitioned Form의 결과와 비교하여 타당성이 입증된다면 가능한 방법 중의 하나가 될것이다.

V. 実験 및 結論

이 같은 計算을 실제 行하면서 그 결과를 알아 보기 위해 다음과 같이 하였다.

Sample로 (8×8), (164×164) Nonsingular Matrix를 구한 것과 한번 Partitionize한 것과의 비교는 <表 Ⅳ> 와 같다.

<表 Ⅳ> 실 험 결 과

Type \ CPU time (sec)	Matrix (8×8)	Matrix (164×164)
Non-Partitionize	5	78
Partitionize	4	64

關聯 資料는 8×8의 경우 Nonsingular인 Sample 資料를 임의로 구성 실시하였고 164×164는 産業 聯關表의 164分類를 이용하였다.

計算 結果는 전체 element들의 合으로 比較하였는데 그 차이가 약 0.1689×10^{-5} 정도이므로 分析하는 데는 지장이 없을것이다.

1回 Partitionize한 것과 직접 求한 것과의 차이로 時間의 절약됨이 如實히 드러나고 있다.

더우기 Partitionize를 계속할 경우 Nonsingular인 Submatrix가 계속 存在한다면 더욱 效果가 있을 것이다.

상대적으로 他作業에도 有利한 方法으로 생각된다.

392分類의 경우는 計算의 正確性때문에 Double Precision을 使用하는 것이 바람직한데 이러한 경우에도 보조 기억장치를 이용하여 이와 같은 方法을 使用하면 상당한 效果가 있을 것이다.

參考로 Partitionize하는 Subprogram을 부록에 실었다.

〈附 錄〉

<附 錄> 1. Partitionize Subprogram

```
            SUBROUTINE PATINV(X,A,B,C,D,W,N1,N2)
            DIMENSION A(1), B(1), C(1), D(1), W(1), X(1)
      WRITE(6,101) N1,N2
101   FORMAT(/' PATINV N1 =', I4,' N2=', I4)
C
C           AA = (A-B.D(-1).C)(-1)
C           BB = -AA.B.D(-1)
C           CC = -D(-1) . C.AA
C           DD = D(-1) -D(-1) * C * BB
C     DECOMPOSE
      CALL  DECOMP(X,A,B,C,D,N1,N2,0)
C           GENR AA
      CALL  MINV(D,N2,W,W(2), W(N2 + 2)
      CALL  MPROD(B,B,D,N1,N2,N2)
      CALL  MPROD(W,B,C,N1,N2,N1)
      CALL  MASD(A,A,W, N1, N1, 1)
      CALL  MINV(A, N1,W,W(2), W(N1 + 2))
C           GENR BB
      CALL  MASD(W,B,B,N1,N2, 3)
      CALL  MPROD(B,A,W,N1, N1, N2)
C           GENR DD
      CALL  MPROD(W,D,C,N2,N2,N1)
      CALL  MASD (W,W,W, N2, N1, 3)
      CALL  MPROD(C,W,A, N2, N1, N1)
C           GENR DD
      CALL  MPROD(W,W,B,N2, N1, N2)
      CALL  MASD(D, D ,W,N2 ,N2, 0)
C           COMPOSE
      CALL  DECOMP(X,A,B,C,D, N1, N2,1)
            RETURN
            END
```

<附 錄> 2. 參考文獻

1. Ben Noble, James W. Daniel
   "APPLIED LINEAR ALGEBRA". Prentice-Hall (1977)
2. 金慶泰, 朴斗一 "應用 數値解析" 塔 出版社(1978)
3. 韓國銀行, "産業 聯關表 作成 報告" (1978)
4. IBM, "System/360 Scientific Subroutine Package" Version Ⅲ Programmer's Manual (1970)