確定的DEAの反復ゲームシミュレーション解法

02502560 日本大学 ＊ 畑澤 文祐 HATAZAWA Fumihiro
01205220 日本大学 篠原 正明 SHINOHARA Masaaki

1 はじめに

DEA (データ包絡分析法) は、多入力多出力システムの効率性に注目した相対評価法である。一般にDEAのCCRモデルは分数計画法、もしくは線形計画法を用いて解を求める方法が一般的であるが、我々の提案するDEA反復シミュレーションでは、それらを用いずに分数ゲーム理論にもとづく反復シミュレーション解法を用いて解を求める。なお、結果については、文献[4]のDEA求解ソフトと比較検討した。

2 ゲーム論的DEA

DEAにおいて、評価対象をDMU(Decision Making Unit) といい、n 個の評価対象があるときには $\mathrm{DMU}_j(j=1,2,\cdots,n)$ と書くことにする。さらに注目するUnitのことを DMU_o と書く。$\boldsymbol{x}_j$、$\boldsymbol{y}_j$ を DMU_j の入力データ列ベクトル、出力データ列ベクトル、$\boldsymbol{v}$、$\boldsymbol{u}$ を入力項目、出力項目について評価列ベクトルとするならば、入力指向CCRモデル(CCRI)の注目する DMU_o についてのLP定式化は次式(1)で与えられる。

< CCRI >

$$\left.\begin{array}{ll}\text{maximize} & \boldsymbol{u}^T\boldsymbol{y}_o\\ \text{subject to} & \boldsymbol{u}^T\boldsymbol{x}_o=1\\ & -\boldsymbol{v}^T\boldsymbol{X}+\boldsymbol{u}^T\boldsymbol{Y}\le 0\\ & \boldsymbol{u}\ge 0,\boldsymbol{v}\ge 0\end{array}\right\}\tag{1}$$

$X_{(m\times n)}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$: 入力データ行列
$Y_{(s\times n)}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n\}$: 出力データ行列

また、(1)は次のミニマクス・マクスミニ問題に等価変換できることが知られている。[1]

$$\left.\begin{array}{r}\max\limits_{(s_1,s_2)}\min\limits_{t} w(A,B,s_1,s_2,t)\\ =\min\limits_{t}\max\limits_{(s_1,s_2)} w(A,B,s_1,s_2,t)\\ s_1\in S_1=\{\boldsymbol{x}\in\boldsymbol{R}^m|\boldsymbol{x}^T\boldsymbol{1}=1,\boldsymbol{x}\ge 0\}\\ s_2\in S_2=\{\boldsymbol{x}\in\boldsymbol{R}^s|\boldsymbol{x}^T\boldsymbol{1}=1,\boldsymbol{x}\ge 0\}\\ \boldsymbol{t}\in T=\{\boldsymbol{x}\in\boldsymbol{R}^n|\boldsymbol{x}^T\boldsymbol{1}=1,\boldsymbol{x}\ge 0\}\end{array}\right\}\tag{2}$$

$$w(A,B,s_1,s_2,t)=\frac{{s_1}^TAt}{{s_2}^TBt}\tag{3}$$

なお、ここでA,Bは注目する DMU_o のデータをもとに以下に示すように正規化してある。

$$A=\{a_{ij}\},a_{ij}=\frac{x_{ij}}{x_{io}}\tag{4}$$

$$B=\{b_{kj}\},b_{kj}=\frac{y_{kj}}{y_{ko}}\tag{5}$$

ここで、s_1,s_2 はゲーム関数を最大化するプレーヤーグループ、t はゲーム関数を最小化するプレイヤーグループとした時の各々の混合戦略ベクトルと考えることができる。

3 例題1

以下に文献[1]の例題について、比較実験した結果を示す。

表 1: 例題1の入出力データ

支店	1	2	3	4	5	6	7	8
入力 x_1	4.0	2.9	4.9	4.1	6.5	10.6	4.8	4.0
入力 x_2	2.1	1.5	2.6	2.3	4.1	5.6	3.3	2.4
出力 y_1	2.6	2.2	3.2	2.6	5.1	7.0	3.6	3.3
出力 y_2	4.1	3.5	5.1	5.7	7.4	11.8	6.1	5.0

表 2: 反復シミュレーション解1(計算回数10万回)

DMU	w	$\overline{s_1}$		$\overline{s_2}$	
1	0.8501	0.4521	0.5479	0.9997	0.0000
2	1	0	1	1	0
3	0.8541	0.6384	0.3616	0.3921	0.6079
4	1	1	0	0	1
5	0.9511	1	0	1	0
6	0.8956	0.6527	0.3473	0.3736	0.6264
7	0.9815	1	0	0.3028	0.0.6972
8	1	1	0	1	0

表 3: 求解ソフト解1

DMU	効率値	VX		UY	
1	0.8498	0.4494	0.5506	0.8498	0
2	1	0.5974	0.4026	0.5542	0.4458
3	0.8541	0.6456	0.3544	0.3306	0.5235
4	1	0.2273	0.7727	0.2183	0.7817
5	0.9510	1	0	0.9510	0
6	0.8956	0.6466	0.3534	0.3348	0.5608
7	0.9815	1	0	0.2969	0.6846
8	1	0.5613	0.4387	0.5662	0.4338

表 4: 反復シミュレーション解 2(計算回数 10 万回)

DMU	$\bar{t}$		
1	$t_2 = 0.9256$	$t_8 = 0.0744$	
2	$t_2 = 1$		
3	$t_2 = 0.8633$	$t_4 = 0.0213$	$t_8 = 0.1154$
4	$t_4 = 1$		
5	$t_8 = 1$		
6	$t_2 = 0.8301$	$t_4 = 0.1425$	$t_8 = 0.0274$
7	$t_4 = 0.3112$	$t_8 = 0.6888$	
8	$t_8 = 1$		

$*$ その他　$t_i = 0$

表 5: 求解ソフト解 2

DMU	Reference Set		
1	$\lambda(2) = 1.0623$	$\lambda(8) = 0.0797$	
2	$\lambda(2) = 1$		
3	$\lambda(2) = 1.1898$	$\lambda(4) = 0.0302$	$\lambda(8) = 0.1527$
4	$\lambda(4) = 1$		
5	$\lambda(8) = 1.5454$		
6	$\lambda(2) = 2.5445$	$\lambda(4) = 0.4373$	$\lambda(8) = 0.0803$
7	$\lambda(4) = 0.3667$	$\lambda(8) = 0.8021$	
8	$\lambda(8) = 1$		

以上の結果より、ゲームシミュレーション解法における w と求解ソフトによる DEA 効率値は、ほぼ一致し、$\bar{t}$ と、Reference Set も和を 1 に正規化することによって一致する。

しかしながら、$\overline{s_1},\overline{s_2}$ と VX,UY に関しては、効率的である DMU に関しては、正規化をしても一致しない。これは、すでにフロンティア上にある DMU は最大化プレーをする必要がないためであると考えられ、初期値から変化することがない。よって初期値に依存する。

また、DMU_1 の w の収束の様子は図 1 で示す。計算 1 回あたりの w の変化は 184 回目から 0.00001 以下になり、10000 回以降は非常に安定することが分かった。

ちなみに、DMU8 つ全ての解を出すのに、計算回数 1 万回では約 9 秒、10 万回では約 1 分 15 秒だった。

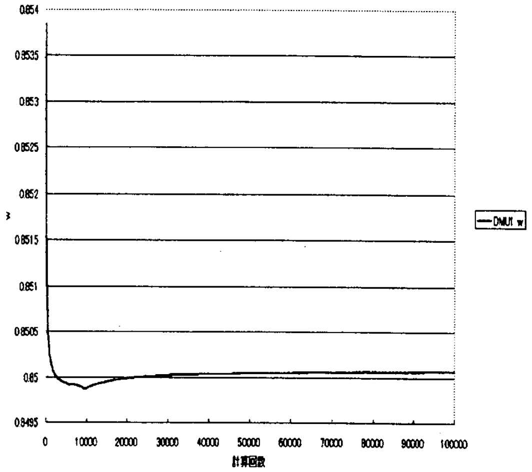

図 1: DMU_1 の w 収束過程

4 例題 2

文献 [3](p.92) の DMU の数が比較的多い例題を解く。

表 6: 例題 2 の入出力データ

$No.$	x_1	x_2	x_3	x_4	y_1	y_2
1	2.249	163.523	26	49.196	5.561	105.321
2	4.617	338.671	30	78.599	18.106	314.682
3	3.873	281.655	51	176.381	16.498	542.349
4	5.541	400.993	78	189.397	30.810	847.872
5	11.381	363.116	69	192.235	57.279	758.704
6	10.086	541.658	114	194.091	66.137	1438.746
7	5.434	508.141	61	228.535	35.295	839.597
8	7.524	338.804	74	238.691	33.188	540.821
9	5.077	511.467	84	267.385	65.391	1562.274
10	7.029	393.815	68	277.402	41.197	978.117
11	11.121	509.682	96	330.609	47.032	930.437
12	7.072	527.457	92	332.609	56.064	1345.185
13	9.348	601.594	127	356.504	69.536	1164.801
14	7.781	528.799	96	365.844	37.467	1348.588
15	6.235	394.158	77	389.894	57.727	1100.779
16	10.593	515.624	101	417.513	46.160	1070.488
17	10.866	566.708	118	503.914	102.967	1707.645
18	6.500	467.617	74	517.318	47.236	1223.026
19	11.469	768.484	103	537.746	84.510	2299.694
20	10.868	669.996	107	590.601	69.576	1901.465
21	10.717	844.949	120	622.550	89.401	1909.698
22	19.716	1258.981	242	660.164	97.941	3055.193
23	10.888	1148.863	202	808.369	191.166	4096.300

表 7: 効率値比較

DMU	w	求解ソフト
1	0.3500	0.3500
2	0.7920	0.7918
3	0.5733	0.5733
4	0.7187	0.7187
5	1	1
6	1	1
7	0.6969	0.6967
8	0.5803	0.5803
9	1	1
10	0.7051	0.7051
11	0.5689	0.5689
12	0.7584	0.7584

DMU	w	求解ソフト
13	0.7476	0.7475
14	0.7215	0.7215
15	0.8441	0.8441
16	0.5822	0.5823
17	1	1
18	0.7868	0.7867
19	1	1
20	0.8486	0.8486
21	0.7872	0.7872
22	0.7849	0.7849
23	1	1

5 おわりに

入出力データ行列が確定的な確定的 DEA は、従来の線形計画法による解法だけでなく、ゲームシミュレーションとして解く事も可能であることを確かめた。またその際に、効率値や参照集合は一致したが、効率的な DMU に対して $\overline{s}_1$, $\overline{s}_2$ といった、"戦略ベクトル" は一意に決まらなかった。今後は、その点に関してさらに考察していきたい。

参考文献

[1] 篠原正明:「DEA とゲーム理論」オペレーションズリサーチ　vol.46 pp290-295

[2] 篠原正明:「CCR モデル-LP 定式化のゲーム理論ならびに多変量解析的解釈」, 第 9 回 RAMP シンポジウム論文集, pp.113-134　(1997)

[3] 刀根薫:「経営効率性の測定と改善」日科技連 (1993)

[4] William W. Cooper, Lawrence M. Seiford and Kaoru Tone:
Data Envelopment Analysis
Kluwer Academic Publishers(2000)