

数理最適化 レポート課題3

最適化理論の応用

提出締切 2020年8月7日

情報学群情報科学類 3年

学籍番号：201811319

氏名：永崎遼太

目次

1. 実験の概要	3
2. 包絡的分析法(Data Envelopment Analysis, DEA)	3
2.1 DEAとは	3
2.2 効率性の評価	4
2.3 DMUの入出力項目の選び方	5
2.4 DEAの特徴	5
2.5 重み付け（ウェイト）	6
2.6 Charnes-Cooper-Rhodes モデル ; CCRモデル	7
3. 定式化・計算機実験	8
4. 店舗の経営効率性の評価	9
4.1 問題設定	9
4.2 DEAによる評価	9
4.3 Python による実装	10
4.4 実行結果	13
4.5 考察	14
5. サッカー選手・クラブの成績評価	17
5.1 スポーツにおけるデータ分析	17
5.2 サッカーにおけるデータ分析	17
5.3 Python による実装	18
5.4 実行結果	19
5.4.1 DEAを用いたプロサッカー選手の評価	19
5.4.2 DEAを用いたプロサッカークラブの評価	20
5.5 考察	22
6. 参考文献	22

1. 実験の概要

自分の興味がある数理最適化に関するトピックに関して、詳しく調べ、理論の理解、アルゴリズムの実装、計算機実験を行い結果をまとめる。

本実験では、包絡的分析法（Data Envelopment Analysis, DEA）について詳しく調べ、計算機実験を行った。

2. 包絡的分析法(Data Envelopment Analysis, DEA)

2.1 DEAとは

包絡的分析法(Data Envelopment Analysis, 以下DEA) とは、テキサス大学のA. チャーンズ、W.W. クーパーの両教授によって開発された、事業体の経営効率の評価・分析を行う手法のことである。もともとは、米国で公立学校の教育プログラムを評価しようとしたとき、従来の回帰分析では説明のつかない結果しか得られなかったことから、適切に評価するための手法として開発された。現在では経営効率分析法とも呼ばれており、その名称からもわかるように、経営効率や計画したことが効果的に行われたかどうかを評価するときに用いられている。

データ分析手法として、次のような企業の経営活動・公的機関の活動を評価する応用例がある。

- ・ 銀行経営予測
- ・ 州政府の予算配分評価
- ・ 自動の送迎バス運行システムの評価
- ・ スポーツ選手の選手の成績評価

DEAは、事業体を多入力多出力システムと捉え、その変換過程の効率性を評価する。企業の経営活動、公的機関の活動について、入力・出力は具体的には次のようにして捉えることができる。

・ 経営活動

[入力]：原材料・設備・労働力・資金・サービス・宣伝

[出力]：生産物・利益・信用・顧客

・ 公的機関

[入力/施策]：資金・労働力・施設

[出力/還元]：サービス・補助

このとき、DEAは単一の事業体についてのみ評価するのではなく、複数の意思決定主体 (Decision Making Unit; DMU)を比較し、評価を行う。DMUは、同じような入力・出力を行なって活動する事業体である。通常、少ない入力で、大きい出力を得ることを”効率的”と捉える。

2.2 効率性の評価

一般的には、事業体の経営効率性を評価する方法としては、収益率や資本利用率などの比率を取る方法や、費用便益分析など全ての効果を金額で表して算出する方法が考えられる。収益率や資本利益率はそれぞれの項目で評価対象を比較する場合は分かりやすいが、複数の項目をまとめて総合的に判断する場合はそれぞれの項目をどのように扱うかが難しくなる。費用便益分析はすべての項目を貨幣という同一の尺度で計測しているため、複数項目の相対比較が容易であるが、効果を金額に換算する方法が問題となる。

DEAは複数の項目を一度に扱うことができ、単位が異なっても取り扱うことができるため、これらの問題に対応することが可能である。

少ない入力から多くの出力を得ようとするDMUの変換効率性は、次の比率尺度で表されることが多い。

$$\text{変換効率} = (\text{出力}) / (\text{入力})$$

1入力 x_j と 1出力 y_j をもつ DMU_j があるとき、回帰分析における直線の当てはめでは平均的な挙動を表す回帰直線

$$y = a_0 + a_1x$$

が得られ、この直線の上にあるか下にあるかで評価する。回帰分析では平均を持ちに相対的に評価を行うが、DEAはそれぞれの対象(DMU)ごとに最も有利に評価した上で、相対

比較を行う。そのため、模範的な対象だけでなく、個性的な対象についても評価を行うことができる。

前述のように、DEAでは、最も良好な挙動 $\max(y_j/x_j)$ を示すDMUを基準にして評価する。一般に、多入力・多出力システムは、比例尺度がいくつも存在することになるため、総合的な判断が難しい。そのような場合、最も優良なDMUは1つに定まらない。しかし、DEAでは、複数の入力と複数の出力を比較して、入力に対する出力の比を算出し、優れたDMUを選び出す。そして、そうでないDMUに対してどれだけ劣っているかを定量的に示し、評価を行う。

2.3 DMUの入出力項目の選び方

DMUの入出力項目は、次の規則に基づいて選ぶ。

- ・ 入力項目・出力項目ともに数値データが準備できる。原則として、前事業体についてその値は正である。
- ・ 自分が見たいと思う効率性の特徴をよく表す入力項目・出力項目を選ぶ。
- ・ 原則として、ある出力を得るための入力に関して、値が小さいほど好ましく、ある入力による出力に関しては大きいほど好ましい。
- ・ 入力・出力項目の数値の単位は任意にとってよい。

2.4 DEAの特徴

DEAの特徴として、主に次の3つが挙げられる。

1. 複数項目での総合評価

効率性を評価する方法として、収入と支出を比較した収益率や利益と資本を比較した資本利益率などが考えられるが、基本的には1対1の単純な比率を用いて評価されている。これらの項目をまとめて総合的に判断する場合、収益率や資本利益率などを見比べて判断するが、項目の数が多くなればなるほど項目間の比較が難しくなる。

DEAは、複数の入出力項目を1つの仮想入力と仮想出力にまとめ、相対的な総合判断を可能にする。

2. 個性的で多様性を生かした評価

項目を評価する場合、回忌分析が利用されることが多いが、回帰分析では、得られた回帰式が全てのDMUに当てはまると仮定している。そのため、回帰式から外れているが、特徴的な経営を行っている本来効率性の高いDMUが適正に評価されないという問題が生じる。

DEAは各評価項目のウェイトを各事業体ごとに最も有利になるように設定する。個性的で多様性を生かした考え方のDEAを取り入れることにより、新しい事実を見分けることができる。

3. 改善値の定量的な把握

事業体Aに有利なウェイトで、他の事業体の効率性を判断する場合、Aより効率的な事業体が存在した場合、その値を算出する。そして、Aにとって最も有利になるウェイトで評価した場合に最も効率値が高い事業体の値が、Aにとっての目標値にもなる。このように、DEAは各事業体の相対的な順位だけでなく、各項目の具体的な改善値を把握することができる。

以上のような特徴から、DEAは従来の評価手法とは異なった特徴を持つため、従来では見落とされていたような新しい結果を得ることができると期待される。

2.5 重み付け（ウェイト）

一般に、重み付けには次の2つの方法がある。

1. 固定ウェイト
2. 可変ウェイト

固定ウェイトは、各ウェイトの比率を固定したもので、個々の事業体の特色を無視したものとなる。仮に、固定ウェイトを単純に平均コストの比で決定する場合や、統計的な回帰分析で決定するような場合には、平均的な事業体像が前面に出てくるそのため、現実の事業体の持つ多様性を評価するのにはあまり適していないと言える。

可変ウェイトは、事業体ごとにウェイトを可変する方法である。入力ウェイト、出力ウェイトは評価の対象ごとに異なってよいものとする。それを決定する原則は、その対象にとって最も好都合となるようにということである。自分のもっとも得意とする項目に大きいウェイトを付け、苦手とする項目に小さいウェイトを付けてもよい。ただし、その同じウェイトで他の事業体も評価し、仮想的入力と仮想的出力を計算し、比率尺度によって、効率性を相対評価する。

可変ウェイトを用い、各事業体が有利になるようにウェイトを設定することは、入出力項目の選定が適正である限り、どの事業体からもクレームがつかないというメリットがある。また、ある事業体が非効率的であるという結果が現れた際には、他のどの事業体と比較してどの程度劣るのか、またどの点を改善すれば効率的になるのかと言ったことが可視化される。

DEAでは、各入力・出力項目ごとに可変ウェイトを割り当てる。

2.6 Charnes-Cooper-Rhodes モデル ; CCRモデル

DEAの基本モデルとして、Charnes-Cooper-Rhodesモデル（CCRモデル）がある。

CCRモデルは、比較の対象となる n 個の DMU_j (入力： x_j , 出力： y_j) に対して相対評価を実行する。この時、全てのDMUは同項目の入力・出力を扱い、個々のDMUは互いに独立に活動し、また互いに影響されない。

CCRモデルでは、個々の DMU_j の入力(x_j)、出力(y_j)を次のようにして表す。

入力： X

$$X = \begin{array}{ccccc} & DMU_1 & DMU_2 & \cdots & DMU_n \\ \left[\begin{array}{cccc} x_{11} & x_{12} & & x_{1n} \\ x_{21} & x_{22} & & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & & x_{mn} \end{array} \right] & \begin{array}{l} \text{入力 1} \\ \text{入力 2} \\ \vdots \\ \text{入力 } m \end{array} \end{array}$$

出力： Y

$$Y = \begin{array}{ccccc} & DMU_1 & DMU_2 & \cdots & DMU_n \\ \left[\begin{array}{cccc} y_{11} & y_{12} & & y_{1n} \\ y_{21} & y_{22} & & y_{2n} \\ \vdots & \vdots & & \vdots \\ y_{s1} & y_{s2} & & y_{sn} \end{array} \right] & \begin{array}{l} \text{出力 1} \\ \text{出力 2} \\ \vdots \\ \text{出力 } s \end{array} \end{array}$$

対象となる事業体を事業体を記号 o とし、 DMU_o と書く ($o = 1,2,3,\dots,n$)。入力への重みを v_i ($i = 1,2,3,\dots,m$)、出力への重みを u_r ($r = 1,2,3,\dots,s$)とする。

CCRモデルでは、次の分数計画問題を解き、 n 個の事業体の活動それぞれについて比例尺度で効率性を測定する。

$$< FP_o > \text{ 目的関数 } \quad \theta = \frac{u_1 y_{1o} + u_2 y_{2o} + \cdots + u_s y_{so}}{v_1 x_{1o} + v_2 x_{2o} + \cdots + v_m x_{mo}}$$

$$\text{制約式} \quad \frac{u_1 y_{1j} + \cdots + u_s y_{sj}}{v_1 x_{1j} + \cdots + v_m x_{mj}} \leq 1 \quad (j = 1, 2, \cdots, n)$$

$$v_1, v_2, \cdots, v_m \geq 0$$

$$u_1, u_2, \cdots, u_s \geq 0$$

上の分数計画問題に対して、次の等価な線形計画問題を解く。

$$< LP_o > \text{ 目的関数 } \quad \theta = u_1 y_{1o} + \cdots + u_s y_{so}$$

$$\text{制約式} \quad v_1 x_{1o} + \cdots + v_m x_{mo} = 1$$

$$u_1 y_{1j} + \cdots + u_s y_{sj} \leq v_1 x_{1j} + \cdots + v_m x_{mj} \quad (j = 1, 2, \cdots, n)$$

$$v_1, v_2, \cdots, v_m \geq 0$$

$$u_1, u_2, \cdots, u_s \geq 0$$

3. 定式化・計算機実験

次の2つの例についてDEAの定式化・計算機実験を行った。

1. 店舗の経営効率性の評価
2. サッカー選手・クラブの成績評価

それぞれについて説明をする。

4. 店舗の経営効率性の評価

4.1 問題設定

ある会社が、表1のような6つの店舗を抱えている。

店舗	店員数	稼働時間	売り上げ
Shop0	5	24	2
Shop1	10	12	6
Shop2	20	12	10
Shop3	20	24	12
Shop4	30	12	12
Shop5	50	12	20

店舗の経営実態

各店舗には、店員数、稼働時間、売り上げについてのデータが得られている。

このデータを元にして、各店舗が、全店舗に対して相対的に効率的に営業を行うことができているのかを判定する。

4.2 DEAによる評価

DEAでは、(出力)/(入力)という比例尺度で効率性を計算するため、多入力・多出力をそれぞれ1つの仮想的入力・仮想的出力に換算する。この問題について、入力は [店員数], [稼働時間] であり、出力は [売上] である。それぞれについての重み(ウェイト)を v_i , u_i ので表すと、仮想的入力・仮想的出力、効率性は次のように表される。

$$\text{仮想的入力} = v_1 \times \text{店員数} + v_2 \times \text{稼働時間}$$

$$\text{仮想的出力} = u_1 \times \text{売り上げ}$$

$$\text{効率性} = \text{仮想的出力} / \text{仮想的入力}$$

4.3 Python による実装

以下の方針で、Pythonにより実装する。

1. ライブラリのインポート
2. 入力の定義
3. 定式化・最適化
4. 最適化の実行
5. 結果の表示

Jupyter Notebook にて、dea_shop.ipynb ファイルを作成した。

上の方針に基づいて、機能ごとに作成したコードの説明をする。

ライブラリのインポート

```
%matplotlib inline
import numpy as np
from pulp import *
import pandas as pd
import pprint
import csv
MEPS=1.03-6
```

ライブラリをインポートしている。

入力の定義

```
df=pd.read_csv('shop.csv',header=0,usecols=[0])
df
```

...

```
DMUs=df.values
print(DMUs)
```

...

```
dfx=pd.read_csv('shop.csv',header=0,index_col=0,usecols=[0,1,2])
print(dfx)
print('')
datax=dfx.values
print(datax)
```

...

```
dfy=pd.read_csv('shop.csv',header=0,usecols=[3])
dfy
```

...

データが格納されているshop.csvファイルからデータを読み込み、定義した変数に値を代入している。dfxは、入力データを格納し、dfyは出力データを格納している。

それぞれの内容は次のようになっている。

```
dfx=pd.read_csv('shop.csv',header=0,index_col=0,usecols=[0,1,2])
print(dfx)
print('')
datax=dfx.values
print(datax)
```

	店員数	稼働時間
店舗		
Shop0	5	24.0
Shop1	10	12.0
Shop2	20	12.0
Shop3	20	24.0
Shop4	30	12.2
Shop5	50	12.0

```
[[ 5. 24. ]
 [10. 12. ]
 [20. 12. ]
 [20. 24. ]
 [30. 12.2]
 [50. 12. ]]
```

```
dfy=pd.read_csv('shop.csv',header=0,usecols=[3])
dfy
```

売り上げ	
0	2
1	6
2	10
3	12
4	12
5	20

定式化・最適化

次のようにして、CCRモデルに基づき、制約式、目的関数を定義している。

for文の中で、各DMUに関して線形最適化問題を解いている。

```
n,m=x.shape
n,s=y.shape
```

```
res=[]
vs=[]
us=[]
```

```
for o in range(n):
    prob=LpProblem('DMU_'+str(o),LpMaximize)
    # 重み
    v=[LpVariable('v'+str(i),lowBound=0,
                  cat='Continuous') for i in range(m)]
    u=[LpVariable('u'+str(i),lowBound=0,
                  cat='Continuous') for i in range(s)]
    # 目的関数
    prob += lpDot(u,y.iloc[o,:])

    # 制約条件
    prob += lpDot(v,x.iloc[o,:])==1,'Normalize'+str(o)

    for j in range(n):
        prob += lpDot(u,y.iloc[j,:]) <= lpDot(v,x.iloc[j,:])

    prob.solve()
    vs=np.array([v[i].varValue for i in range(m)]) #v*
    vs_.append(vs)
    us=np.array([u[i].varValue for i in range(s)]) #u*
    us_.append(us)
    print('DMU_'+str(o),' : ',DMUs[o])
    print('vs_'+str(o),' : ',vs)
    print('us_'+str(o),' : ',us)
    print(' ')

    res.append([DMUs[o],value(prob.objective)])
```

結果の表示

```
df_f=pd.DataFrame(res,columns=['店舗番号','効率値'])
df_s=df_f.sort_values('効率値',ascending=False)
df_s
```

各店舗の効率値を出力する。

実際に、次のように表示される。

```
df_f=pd.DataFrame(res,columns=['店舗番号','効率値'])
df_s=df_f.sort_values('効率値',ascending=False)
df_s
```

	店舗番号	効率値
1	[Shop1]	1.000000
2	[Shop2]	1.000000
5	[Shop5]	1.000000
3	[Shop3]	1.000000
4	[Shop4]	0.896266
0	[Shop0]	0.666667

4.4 実行結果

3.1.1 節で定義した問題の入力データを用いて、DEAのCCRモデルにより、可変ウェイト、各店舗の経営効率値は次のようになった。

DMU_0 : ['Shop0']
vs_0 : [0.2 0.]
us_0 : [0.33333333]

DMU_1 : ['Shop1']
vs_1 : [0.1 0.]
us_1 : [0.16666667]

DMU_2 : ['Shop2']
vs_2 : [0.04 0.01666667]
us_2 : [0.1]

DMU_3 : ['Shop3']
vs_3 : [0.05 0.]
us_3 : [0.08333333]

DMU_4 : ['Shop4']
vs_4 : [0.025 0.02083333]
us_4 : [0.075]

DMU_5 : ['Shop5']
vs_5 : [0. 0.08333333]
us_5 : [0.05]

各店舗の可変ウェイト

	店舗番号	効率値
1	[Shop1]	1.000000
2	[Shop2]	1.000000
5	[Shop5]	1.000000
3	[Shop3]	1.000000
4	[Shop4]	0.900000
0	[Shop0]	0.666667

各店舗の効率値

以上の結果をまとめると、次の表のようになる。

		Shop0	Shop1	Shop2	Shop3	Shop4	Shop5
店舗	効率性	0.666667	1.0	1.0	1.0	0.900000	1.0
入力	重み(店員数)	0.2	0.1	0.04	0.05	0.025	0
	重み(稼働時間)	0	0	0.01666667	0	0.02083333	0.08333333
出力	重み(売り上げ)	0.33333333	0.16666667	0.1	0.08333333	0.075	0.05

実行結果

目的関数値(効率性)は、 $\theta = 1$ ならば DMU_o はD効率的であり、 $\theta < 1$ ならば DMU_o はD非効率的である。D非効率的であるとは、各種制約式を満たすどのような重みを選択しても効率が1になり得ないということを表している。

4.5 考察

実行結果からわかるように、Shop1、Shop2、Shop5、Shop3はD効率的であるのに対し、Shop0、Shop4はD非効率であるとわかる。

D非効率である DMU_o の最適解が (u^*, v^*) では、不等式制約

$$u_1^* y_{1j} + u_2^* y_{2j} + \cdots + u_s^* y_{sj} \leq v_1^* x_{1j} + v_2^* x_{2j} + \cdots + v_m^* x_{mj}$$

がどれかの $j \neq o$ の集合を、

$$E_o = \{j : \sum_{r=1}^s u_r^* y_{rj} = \sum_{i=1}^m v_i^* x_{ij}, j = 1, 2, \dots, n, j \neq o\}$$

とする。 E_o に入っているDMUは、 DMU_o を非効率にしている原因となっていると判断できる。この E_o を DMU_o に対する参照集合であるという。D非効率は店舗に関しては、非効率となっている原因を形成している店舗集合を参照集合として導き出すことが可能である。参照集合や、入出力の重みをよく見ると、非効率な DMU_o の特徴がみえ、改善すべき点が見えてくる。

D非効率であったDMUに対して、より効率的な経営方針を提案することを考える。CCRモデルによって算出された重みを固定し、入力・出力値をどれだけ変化させれば相対的に効率的になるか、ということを計算することで、新たな経営方針を提案することができると考えた。この時、現状の営業実態から大幅に変更することは望ましくないため、入力・出力値の変化は小さい方が好ましいと考えられる。

入力・出力値の変化を変数 w_i を用いて表す。重みを固定し、入力・出力値の変化させ、D非効率的である店舗の効率性をD効率的になるようにする。入力・出力値の変化は小さい方が望ましいので、これを目的関数値とし、最小化する次の線形計画問題を解く。

$$\langle LP_o \rangle \text{ 目的関数 } -((w_1 - w_2) + (w_3 - w_4) + (w_5 - w_6))$$

制約式

$$\begin{aligned} v_1^* (x_{1o} + (w_1 - w_2)) + v_2^* (x_{2o} + (w_3 - w_4)) &\leq u_1^* (y_{1o} + (w_5 - w_6)) \\ -(v_1^* (x_{1o} + (w_1 - w_2)) + v_2^* (x_{2o} + (w_3 - w_4))) &\leq -u_1^* (y_{1o} + (w_5 - w_6)) \\ v_1, v_2, v_3, \dots, v_m &\geq 0 \\ u_1, u_2, u_3, \dots, u_s &\geq 0 \\ w_1, w_2, w_3, \dots, w_{(m+s)*2} &\geq 0 \end{aligned}$$

Shop4の効率値が0.9であったので、Shop4について線形計画問題を解くことを考える。

次のようなコードを追加し、線形最適化問題を解く。

```
x1=LpVariable('x1', lowBound=0.0)
x2=LpVariable('x2', lowBound=0.0)
x3=LpVariable('x3', lowBound=0.0)
x4=LpVariable('x4', lowBound=0.0)
x5=LpVariable('x5', lowBound=0.0)
x6=LpVariable('x6', lowBound=0.0)
prob=LpProblem(name='LP-Sample',sense=LpMinimize)

# 目的関数
prob += x1-x2+x3-x4+x5-x6
# 制約式
T1=vs_[k][0][0]*(datax[k][0]+x1-x2)
T2=vs_[k][0][1]*(datax[k][1]+x3-x4)
T3=us_[k][0][0]*(datay[k][0]+x5-x6)
T4=T1+T2
prob += T3==T4
# prob += (vs_[4][0][0]*(datax[4][0]+x1-x2))+vs_[4][0][1]*(datax[4][1]+x3-x4) = us_[4][0]*(datay[4][0]+x5-x6)
```

新たに変数 $x_1 \sim x_6$ を導入している。これは、定式化における $w_1 \sim w_6$ に対応する。

実際に計算をすると、次のような結果となった。

[入力・出力値の変化($w_i = x_i$)]

$x_1 = 0.0$
 $x_2 = 0.0$
 $x_3 = 0.0$
 $x_4 = 4.79999999$
 $x_5 = 0.0$
 $x_6 = 0.0$

変数の最適値

線形計画問題を解いて得られた解の適応前と適応後の結果を比較すると次のようになる。

店舗		Shop4	Shop4 revision
効率性		0.900000	1.0
入力	重み(店員数)	0.025	0.025
	重み(稼働時間)	0.02083333	0.02083333
出力	店員数	30	30
	稼働時間	12	7.2
	重み(売り上げ)	0.075	0.75
	売り上げ	12	12

改善前と改善後の比較

これは、売り上げをそのままに、稼働時間を減らせば良いということを表している。

最終的に店舗の経営効率はこのようになる。

	店舗番号	効率値
1	[Shop1]	1.000000
2	[Shop2]	1.000000
5	[Shop5]	1.000000
3	[Shop3]	1.000000
4	[Shop4]	1.000000
0	[Shop0]	0.666667

改善後の効率値

Shop4の効率値が1.0となったことから、新たな経営方針は、Shop4の経営を改善していると言える。

今回のShop4の経営方針に対する提案内容は、他の店舗の経営実態と比較してみると、実現可能であることがわかる。しかし、元々の経営実態によっては、同様の方法で新たな経営方針を提案しても、実現困難な場合が生じてしまうことが考えられる。そのような問題に対しては、店員数、稼働時間、売り上げのそれぞれに対する値の取り得る範囲を設定することで、解決されることが考えられる。

非効率的なDMUを効率的に改善するための目標値を設定するとき、環境変数のようにそれぞれのDMUの改善努力によって変更できない変数については、現状のままにしておき、変更が可能な変数のみの改善による改善目標値の設定方法を示すことができる。この場合は、定式化における制約条件において、変更が不可能な入出力項目に対する不等式制約を等式製薬に変えることで実現できる。これは、効率値を測る距離の取り方を変えることに対応する。

5. サッカー選手・クラブの成績評価

5.1 スポーツにおけるデータ分析

近年のラグビーやバレーボールの試合を観戦していると、監督の周辺でパソコンを操作して何らかの情報を提供するスタッフがいたり、タブレットを持った監督やコーチが送られてくる情報を画面で確認して選手に指示を送ったりするシーンが散見される。これらは、試合中における選手の動きを分析し、その情報をもとに、効果的な戦略や、選手の調子などを評価している。

このように、近年のIT技術の発展のおかげもあり、多くのスポーツ現場でデータ収集やデータ分析がなされている。それに伴い、選手やクラブを統計データから評価する研究も増えてきている。スポーツを統計学の観点から研究している論文はスポーツ工学やORの分野で数多く見られる。

5.2 サッカーにおけるデータ分析

スポーツにおけるデータ分析が近年盛んに行われているが、中でも、ヨーロッパを中心として、サッカーに関するデータ分析が活発に行われている。サッカーにおけるデータ分析が注目されるきっかけとなったのは、2010年の南アフリカW杯で、選手の走行距離、トップスピード、ポゼッションエリア等、今まででは見られなかったデータを元にした戦略を練ったチームが好成績を収めたことである。データを整理・分析し、スコアシートレベルではわからなかった情報を得ることで、スポーツの戦略的な面白さは格段に増していると言える。それに伴い、スポーツのデータを分析する会社が設立すると言った新たな動きが見られている。

トップレベルの現代サッカー界においては、アナリストの資質が勝負を分ける要因になりうるとされ、その重要性は年々高まっている。

サッカー以外にも、例えばバスケットボールでは、コート上の選手とボールの動き、フォーメーション傾向、シュート角度等のデータを分析すると言ったことが盛んに行われ始めている。

サッカーにおけるデータ分析は、一般に難しいとされてきた。それは次のような理由からである。

1. 一人一人が動き回る。

野球の打率・防御率などとは違う。

2. 数値のみではチーム・個人のパフォーマンスを図り切るのは難しい。

雑誌等の各媒体ごとに選手の採点結果は異なる。

3. ポジションごとに評価の難しさが異なる。

アタッカーの評価はしやすい一方で、ディフェンダーの評価は難しい。

このような、扱いが難しいサッカーのデータも、DEAを用いれば総合的な相対評価を行うことができる。

本実験では、選手データ、クラブデータの両方のデータを利用して、選手のシーズンを通しての成績評価、クラブの経営評価の2つの分析を行う。

5.3 Python による実装

3.2.2節と同様に、以下の方針で、Pythonにより実装した。

1. ライブラリのインポート
2. 入力の定義
3. 定式化・最適化
4. 最適化の実行
5. 結果の表示

5.4 実行結果

5.4.1 DEAを用いたプロサッカー選手の評価

実際のデータを元に、DEAを用いてプロサッカー選手の投資効率評価を行う。実験データは、現在世界で最もレベルが高いと言われているイングランドにおけるサッカーリーグ "Premiere League"での選手データを用いる。具体的には、2019-2020シーズンについて、年俸が高い10選手について、その成績データを用いて評価する。入力・出力データは次のデータを用いた。

[入力データ]

- ・ 年俸

[出力データ]

- ・ 出場試合数、ゴール数、アシスト数、枠内シュート数、一試合あたりのパス数、一試合あたりのタックル数

年俸が高い10選手は、全てアタッカーの選手であったため、アタッカーの選手を評価するのに適した出力データを選ぶ必要があった。

実際のデータをまとめると、次のようになる。

Name	Salary (€)	Appearance	Goals	Assists	Shots on Target	Passes per game	Tackle per game
Kevin De Bruyne	16683333	33	11	19	28	54	3
Raheem Sterling	15600000	31	17	2	32	36	2
Paul Pogba	15080000	14	1	4	8	65	2
Mesut Ozil	13975000	18	1	3	3	47	2
Anthony Martial	13000000	30	17	6	38	23	1
Sergio Aguero	11967000	24	16	4	27	15	1
Mohamed Salah	10400000	32	19	10	58	29	1
Marcus Rashford	10400000	29	17	8	42	31	1
Harry Kane	10400000	28	17	3	36	20	1

選手データ

DEAを用いたプロサッカー選手の評価について、各DMU（選手）についての可変ウェイト、各選手の投資効率評価は次のようになった。

```

DMU_0 : ['Kevin De Bruyne']
vs: [5.9940061e-08 None None None None]
us: [0.01948052]

DMU_1 : ['Raheem Sterling']
vs: [6.4102564e-08 None None None None]
us: [0.02083333]

DMU_2 : ['Paul Pogba']
vs: [6.6312997e-08 None None None None]
us: [0.02155172]

DMU_3 : ['Mesut Ozil']
vs: [7.1556351e-08 None None None None]
us: [0.02325581]

DMU_4 : ['Anthony Martial']
vs: [7.6923077e-08 None None None None]
us: [0.025]

DMU_5 : ['Sergio Aguero']
vs: [8.3563132e-08 None None None None]
us: [0.02715802]

DMU_6 : ['Mohamed Salah']
vs: [9.6153846e-08 None None None None]
us: [0.03125]

DMU_7 : ['Marcus Rashford']
vs: [9.6153846e-08 None None None None]
us: [0.03125]

DMU_8 : ['Harry Kane']
vs: [9.6153846e-08 None None None None]
us: [0.03125]

DMU_9 : ['Tanguy Ndombele']
vs: [9.6153846e-08 None None None None]
us: [0.03125]

```

	Name	Efficiency
6	[Mohamed Salah]	1.000000
7	[Marcus Rashford]	0.906250
8	[Harry Kane]	0.875000
4	[Anthony Martial]	0.750000
9	[Tanguy Ndombele]	0.656250
5	[Sergio Aguero]	0.651792
1	[Raheem Sterling]	0.645833
0	[Kevin De Bruyne]	0.642857
3	[Mesut Ozil]	0.418605
2	[Paul Pogba]	0.301724

各選手の効率値

各選手の可変ウェイト

5.4.2 DEAを用いたプロサッカークラブの評価

実際のデータを元に、DEAを用いてプロサッカークラブの経営効率評価を行う。実験データは、"Premiere League"でのクラブ運営データを用いる。具体的には、2019-2020シーズンについて、年棒が高い10選手について、その成績データを用いて評価する。入力・出力データは次のデータを用いた。

[入力データ]

- ・監督の給与、移籍金支出、クラブの歴史、クラブレーティング(FIFA20)

[出力データ]

- ・クラブの利益、勝ち点

クラブの歴史は、イングランドの一部リーグ"Premiere League"でのシーズン数を表している。クラブレーティングは、サッカーゲーム"FIFA20"で、各クラブの評価値を利用している。このクラブレーティングは、オフENS、ミッドフィルダー、ディフェンスの各ポジションの在籍選手をもとにして総合的なチーム力として算出された値である。

実際のデータをまとめると、次のようになる。

Club	Profit(€)	Points	Manager Salary(€M)	Expenditure(€M)	Seasons in Premiere League	Club Rating
Liverpool	152425146	93	24	10.40	28	85
Manchester City	150986355	75	27	166.82	23	85
Chelsea	146030216	63	7.5	45.00	28	81
Leicester City	123328078	62	9.1	104.30	14	79
Manchester United	142193180	62	8.3	214.00	28	81
Wolverhampton Wanderers	127165114	59	8	121.80	6	78
Tottenham Hotspur	145230801	58	23	148.50	28	82
Sheffield United	96628865	54	3.0	69.00	4	75
Burnley	107340558	54	3.6	19.40	6	76
Arsenal	142193180	53	6.5	160.40	28	80

クラブデータ

DEAを用いたプロサッカークラブの強化について、各DMU（クラブ）についての可変ウェイト、各クラブの経営効率評価は次のようになった。

```

DMU_0 : ['Liverpool']
vs: [0. 0.02133541 0.02778971]
us: [0. 0.01075269]

DMU_1 : ['Manchester City']
vs: [0. 0.00098743 0.03631637]
us: [2.2084327e-09 0.0000000e+00]

DMU_2 : ['Chelsea']
vs: [0.12969803 0.00060588 0. ]
us: [4.4593308e-09 0.0000000e+00]

DMU_3 : ['Leicester City']
vs: [0. 0.00161499 0.05939692]
us: [3.6119828e-09 0.0000000e+00]

DMU_4 : ['Manchester United']
vs: [0.12048193 0. 0. ]
us: [3.7405571e-09 0.0000000e+00]

DMU_5 : ['Wolverhampton Wanderers']
vs: [0. 0. 0.16666667]
us: [6.8992497e-09 0.0000000e+00]

DMU_6 : ['Tottenham Hotspur']
vs: [0. 0.00084868 0.03121325]
us: [1.8981072e-09 0.0000000e+00]

DMU_7 : ['Sheffield United']
vs: [0.30099329 0.00140609 0. ]
us: [1.0348875e-08 0.0000000e+00]

DMU_8 : ['Burnley']
vs: [0. 0.00416543 0.15319845]
us: [9.3161431e-09 0.0000000e+00]

DMU_9 : ['Arsenal']
vs: [0.15384615 0. 0. ]
us: [4.7764036e-09 0.0000000e+00]

```

	Team	Efficiency
7	[Sheffield United]	1.000000
8	[Burnley]	1.000000
0	[Liverpool]	1.000000
5	[Wolverhampton Wanderers]	0.877344
9	[Arsenal]	0.679172
2	[Chelsea]	0.651197
4	[Manchester United]	0.531882
3	[Leicester City]	0.445459
1	[Manchester City]	0.333443
6	[Tottenham Hotspur]	0.275664

各クラブの効率値

各クラブの可変ウェイト

5.5 考察

投資効率は、低い年棒でいかに成績を残すかということを表してる。この数値は、2019-2020シーズンの成績をもとに算出されているため、次シーズン以降の年棒の調整に利用することができ、具体的には効率が低い選手の年棒を下げたりするといったことの方針を考えることができる。

クラブの経営効率は、低いクラブの”レベル”でいかに高い順位、収入を得るかということを表している。この数値を用いることで、次シーズン以降の経営を効率化するための方針を考えることができる。

ビックデータ時代に突入し、多くの企業データが得られるようになっている。これに伴い、DEAの適応事例として、企業経営や生産現場における活動の効率化のため、無駄な活動をなくすことも求められることから、その適応分野はますます広がっていくと考えられる。

DEAは、近年は大規模問題へ適用するための計算手法やユーザーインターフェースの開発、入出力項目の選択あるいは統合の方法、多変量解析や実験計画法などの統計的手法との融合、様々なモデルに対する感度分析、確率的DEAの確率などが注目を集めている。

データを扱うときには平均値などの確定値のみで議論するだけではなく、その不確実性を考慮に入れた解析も必要になる。不確実性に対するロバスト性や効率値の一致性や最尤性などの統計的性質、データの再サンプリングによる効率性の推定、不確実性を取り込んだ効率性の尺度など、その研究対象は様々である。DEAは、確率的計画法との関連から、いくつかの確率的DEAモデルがある。

また、ばらつきのあるデータに対するDEAの効率性の信頼度を定義する方法がある。ばらつきのあるデータはその平均値などを用いて解析することもあるが、その場合には得られた効率性がばらつきに対してどの程度信頼できるものかが注目される。効率的と評価されるDMUは全て効率値は1となるが、効率的なDMU同士には優劣をつけられない。しかし、データのばらつきに対する影響の大きさには違いがあるので、これを信頼度として定義し、効率的と評価されたDMUのばらつきに対するロバスト性を考慮することができる。

6. 参考文献

1. 包絡分析法 (DEA) について
2. 包絡分析法 (DEA) モデル
3. DEA (包絡分析法) の理論と応用
4. infogol
5. spotrac

6. Premier League

7. FIFA Index