

第6回基礎ゼミ資料

Practice NL&MXL from R

平成30年5月18日(金)

朝倉研究室 修士1年 小池卓武

使用データ

～横浜プローブパーソンデータ～

主なデータの中身

- ・トリップID
 - ・目的
 - ・出発, 到着時刻
 - ・総所要時間
 - ・移動距離
 - ・交通機関別の時間, 距離
 - ・アクセス, イグレス時間, 距離
 - ・費用
 - ・代表交通手段
 - ・代替手段生成可否
 - ・性別, 年齢等の個人属性
- etc..



NLモデルによる推定-NLの効用関数

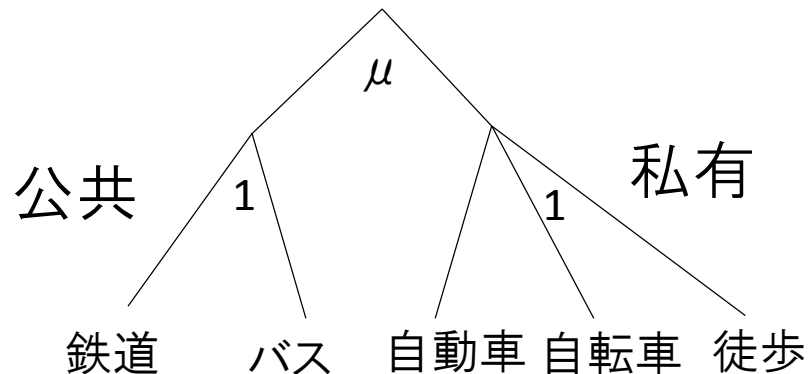
$$U_{train} = \beta_{time} \overset{\text{時間}}{time}_{train} + \beta_{cost} \overset{\text{費用}}{cost}_{train} + \overset{\text{誤差項}}{\varepsilon_{Public_train}} \quad (\text{選択肢間相関あり?})$$

公共
$$U_{bus} = \beta_{time} time_{bus} + \beta_{cost} cost_{bus} + \varepsilon_{Public_bus}$$

$$U_{car} = \beta_{time} time_{car} + \beta_{cost} cost_{car} + \varepsilon_{Private_car}$$

私有
$$U_{bike} = \beta_{time} time_{bike} + \beta_{cost} cost_{bike} + \varepsilon_{Private_bike}$$

$$U_{walk} = \beta_{time} time_{walk} + \beta_{cost} cost_{walk} + \varepsilon_{Private_walk}$$



下位ネストのスケールパラメータは1に固定し、上位ネストのスケールパラメータ μ を推定. その際、 $0 < \mu \leq 1$ が条件.

NLモデルによる推定-初期設定

～初期設定～

nl_maxLik_1.rを展開！

Session → Set Working Directory → To Source File Locationをしてから実行

```

3  ### データファイルの読み込み
4  Data <- read.csv("ensyu.csv", header=T)
5  ## データ数:Dataの行数を数える
6  hh <- nrow(Data)
7
8  ## パラメータの初期値の設定
9  b0 <- numeric(7)

```

読み込むデータの1行目が

- ・データでなく列名ならheader=T
- ・データから始まるならheader=F

nrow(行列): 引数行列の行数を返す

numeric(引数): 引数個の0を返す
→パラメータ推定値の初期値

(交通手段数-1)個の定数項

+

パラメータ β (時間, 料金) 2つ

+

スケールパラメータ(μ) × 1つ

↓

推定値7つ…初期値として0を与えておく

～対数尤度関数を定義～

```
11 - ##### Logit model の対数尤度関数の定義 #####
12
13 - fr <- function(x) {
14   ### パラメータの宣言
15   ## 定数
16   b1 <- x[1]
17   b2 <- x[2]
18   b3 <- x[3]
19   b4 <- x[4]
20
21   ## 目的地までの所要時間
22   d1 <- x[5]
23
24   ## 料金
25   f1 <- x[6]
26
27   ##スケールパラメータ
28   pa <- x[7]
29
30   ## 対数尤度のための変数を宣言
31   LL = 0
```

function(引数){算術式}を用いて対数尤度を計算するための関数をあらかじめ定義.

初期値として対数尤度LL=0を設定しておく.

NLモデルによる推定-対数尤度関数の設定

～exp(V)を定義～

```

33   ### 今回用いる交通手段は以下5つ
34   ## 鉄道(train) - (公共)
35   ## バス(bus) - (公共)
36   ## 自動車(car) - (私有)
37   ## 自転車(bike) - (私有)
38   ## 徒歩(walk) - (私有)
39
40   ## 効用の計算:説明変数にした行列を入れる
41                                     # 時間                    # 料金                    # 定数項
42   train <- Data$代替手段生成可否train*exp(d1*Data$総所要時間train/100 +f1*Data$費用train/100 + b1*matrix(1,nrow =hh,ncol=1))
43   bus   <- Data$代替手段生成可否bus *exp(d1*Data$総所要時間bus/100 +f1*Data$費用bus/100 + b2*matrix(1,nrow =hh,ncol=1))
44   car   <- Data$代替手段生成可否car *exp(d1*Data$所要時間car/100 + b3*matrix(1,nrow =hh,ncol=1))
45   bike  <- Data$代替手段生成可否bike *exp(d1*Data$所要時間bike/100 + b4*matrix(1,nrow =hh,ncol=1))
46   walk  <- Data$代替手段生成可否walk *exp(d1*Data$所要時間walk/100 )

```

Data\$代替手段生成可否trainには、trainが代替交通手段として生成されるなら1，そうでなければ0が入る。

SOL変数を同程度のオーダーにして推定結果を安定させるためにそれぞれ100で除す。(100分, 100円単位にする)

NLモデルによる推定-対数尤度関数の設定

～上位ネストでの公共・私有選択確率～

```

56  ###上位ネスト構造：公共・私有の選択
57  ## ログサム変数の設定
58  logsum.public <- log( ( (train+bus)!=0) * (train + bus) + ((train+bus)==0))
59  logsum.private <- log( ( (car + bike + walk)!=0) * (car + bike + walk) + ((car + bike + walk)==0))
60
61  ## ログサム変数を用いて公共・私有の選択確率P(LV1)を表現
62  nume.public <- exp(mu*logsum.public)
63  nume.private <- exp(mu*logsum.private)
64
65  deno <- nume.public + nume.private
66  P.public <- nume.public / deno
67  P.private <- nume.private / deno
68

```

ログサム変数： $V'_d = \ln \sum_{m'} \exp(V_{m'})$

上位ネストでの(公共・私有)選択確率： $P(d) = \frac{\exp(\mu(V_d + V'_d))}{\sum_{d'} \exp(\mu(V_{d'} + V'_{d'}))}$

d : 上位ネスト, μ : 上位ネストのスケールパラメータ

NLモデルによる推定-対数尤度関数の設定

～下位ネストでの交通機関選択確率～

```

70   ###下位ネスト構造：交通機関の選択
71   ###公共・私有のなかでの選択：条件付き確率P(LV2|LV1)
72
73   ##公共ネストにおける交通機関の選択確率
74   deno.public    <- train + bus
75   P.train.public <- train / ((deno.public!=0)*deno.public + (deno.public==0))
76   P.bus.public   <- bus   / ((deno.public!=0)*deno.public + (deno.public==0))
77
78   ##私有ネストにおける交通機関の選択確率
79   deno.private   <- car + bike + walk
80   P.car.private  <- car   / ((deno.private!=0)*deno.private + (deno.private==0))
81   P.bike.private <- bike  / ((deno.private!=0)*deno.private + (deno.private==0))
82   P.walk.private <- walk  / ((deno.private!=0)*deno.private + (deno.private==0))
83
84   ### 最後に公共・私有それぞれの交通手段の同時確率を計算： P(LV1, LV2) = P(LV2|LV1)*P(LV1)
85   P.train <- P.train.public * P.public
86   P.bus   <- P.bus.public   * P.public
87   P.car   <- P.car.private  * P.private
88   P.bike  <- P.bike.private * P.private
89   P.walk  <- P.walk.private * P.private

```

$$\text{下位ネストでの(交通機関}m\text{)選択確率: } P(m|d) = \frac{\exp(V_m)}{\sum_{m'} \exp(V_{m'})}$$

$$\text{交通機関選択確率: } P(m|d)P(d) = \frac{\exp(V_m)}{\sum_{m'} \exp(V_{m'})} \frac{\exp(\mu(V_d + V'_d))}{\sum_{d'} \exp(\mu(V_{d'} + V'_{d'}))}$$

NLモデルによる推定-対数尤度関数の設定

～対数尤度の計算式を定義～

```

91  ## 選択確率0になってしまった場合に起こる問題の回避：真数条件に反するエラーをケア
92  P.train <- (P.train!=0)*P.train + (P.train==0)
93  P.bus   <- (P.bus!=0) *P.bus   + (P.bus==0)
94  P.car   <- (P.car!=0) *P.car   + (P.car==0)
95  P.bike  <- (P.bike!=0) *P.bike + (P.bike==0)
96  P.walk  <- (P.walk!=0) *P.walk + (P.walk==0)
97
98  ## 選択結果
99  C.train <- Data$代表交通手段 == "鉄道"
100 C.bus   <- Data$代表交通手段 == "バス"
101 C.car   <- Data$代表交通手段 == "自動車"
102 C.bike  <- Data$代表交通手段 == "自転車"
103 C.walk  <- Data$代表交通手段 == "徒歩"
104
105  ### 対数尤度の計算
106  LL <- colSums(C.train*log(P.train) + C.bus*log(P.bus) +
107               C.car *log(P.car)   + C.bike *log(P.bike) +C.walk *log(P.walk))
108
109 }

```

log(P)の計算時に真数条件:log(P):P > 0
に反してエラーになるのを防止

代表交通手段がiならば
C.iには1が入る

対数尤度LL: (*LogLikelihood*): $\sum_n^N \sum_i^J d_i \ln P_n(i)$

N: 全サンプル数, n: 個人, J: 全交通機関選択肢数,

i: 交通機関選択肢, d_i : 交通機関選択ダミー(1 or 0), $P_n(i)$: 交通機関選択確率



未知パラメータに対して**最大化**することでパラメータ推定: maxLik関数

NLモデルによる推定-MaxLik関数

～対数尤度の最大化～

maxLik関数を用いて最尤推定(Optim関数の代替案！)

maxLik(対数尤度関数, start=パラメータ初期値, method="～")

ヘッセ行列(hess=), 勾配(grad=), 制約条件(constraint=)も引数として入れられるが, 今回は単純なケースとして上記の式形で計算.

Method...最大化手法にも様々ある...

- ・NM(Nelder-Mead)←計算が粗い..
- ・NR(Newton-Raphson)←勾配(grad=)が必要
- ・BFGS(Broyden-Fletcher-Goldfarb-Shanno)←gradientが無くても大丈夫

Etc..

→今回はBFGSを採用

今回はmethodについて考える機会を設けるためにあえて関数内にmethodを設定したが, 何も設定しなければ最適なmethodが自動適用される仕様になっている.

NLモデルによる推定-結果の出力

～maxLik関数でパラメータ推定～

```

113 ##パラメータ値の最適化
114 res <- maxLik(fr,start=b0,method="BFGS")
115 res
116 ## パラメータ推定値、ハッセ行列
117 b <- res$estimate
118 hhh <- res$hessian
119
120 ## t値の計算
121 tval <- b/sqrt(-diag(solve(hhh)))
122
123 ## 初期尤度
124 L0 <- fr(b0)
125 ## 最終尤度
126 LL <- res$maximum

```

Broyden-Fletcher-Golden-Shannoメソッドを最適化メソッドとして採用。

sqrt関数: 引数の平方根を計算

diag関数: 引数(行列)の対角成分を計算

solve関数: 引数(正方行列)の逆行列を計算

パラメータ推定値/標準偏差 = t値

```

140 ##### 結果の出力 #####
141 print(res)
142 ## 初期尤度
143 print(L0)
144 ## 最終尤度
145 print(LL)
146 ##p^2値
147 print((L0-LL)/L0)
148 ## 修正済p^2値
149 print((L0-(LL-length(b)))/L0)
150 ##パラメータ推定値
151 print(b)
152 ## t値
153 print(tval)

```

Printで出力



```

> ## 初期尤度
> print(L0)
[1] -582.4761
> ## 最終尤度
> print(LL)
[1] -379.8563
> ##p^2値
> print((L0-LL)/L0)
[1] 0.3478595
> ## 修正済p^2値
> print((L0-(LL-length(b)))/L0)
[1] 0.3358418
> ##パラメータ推定値
> print(b)
[1] 0.93445190 -0.15896202 -0.06760577 -0.48256927 -3.45664165 -0.04132932
> ## t値
> print(tval)
[1] 6.6422912 -0.7694445 -0.4375288 -2.6701749 -7.5745253 -2.4933597 6.9497208

```

スケールパラメータ μ が
 $0 < \mu \leq 1$ を満たしていない!!
 →ネスト構造の再検討が必要..

パラメータ推定値は左から
 定数項 $\times 4$, $\beta \times 2$, μ

2.98878973

MXLモデルによる推定-MXLの効用関数

～ランダム係数モデルの効用関数～

$$U_i = \beta_{time} time_i + \beta_{cost} cost_i + \varepsilon_i$$

$$= (\bar{\beta}_{time} + \sigma_{time} \eta_{time}) time_i + (\bar{\beta}_{cost} + \sigma_{cost} \eta_{cost}) cost_i + \varepsilon_i$$

$\bar{\beta}_x$: 平均, σ_x : 平均, η_x : 標準正規分布に従う乱数, ε_i : IIDガンベル分布

個人の嗜好性によってパラメータ β が確率的に変化
→ ランダム係数と呼ぶ

NLと同じ交通手段選択肢を用いてパラメータ推定！

MXLモデルによる推定-初期設定

～データの読み込み～

```

3   ### データファイルの読み込み
4   Data <- read.csv("ensyu.csv",header=T)
5   ## データ数:Dataの行数を数える
6   hh <- nrow(Data)
7
8   ## パラメータの初期値の設定
9   b0 <- numeric(8)
10
11  ##### 乱数の発生 #####
12
13  ## 乱数の発生回数
14  R <- 100

```

nrow(行列): 引数行列の行数を返す
→ サンプル数

numeric(引数): 引数個の0を返す
→ パラメータ推定値の格納配列

(交通手段数-1)個の定数項 + パラメータ β (時間, 料金)の平均, 標準偏差



推定値8つ…初期値として0を与えておく.

繰り返し計算によりパラメータ推定を行うので,
その繰り返し回数R=100を設定.

MXLモデルによる推定-乱数の生成

～ハルトン数列法で準乱数を生成～

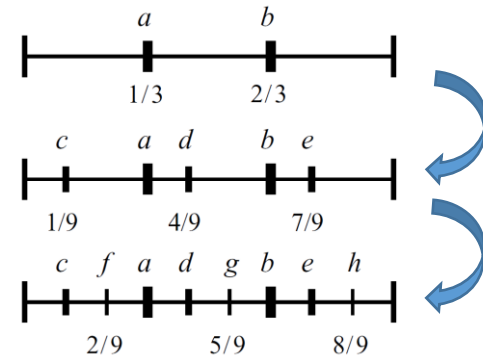
```

20 ## 乱数の発生-ハルトン数列関数による方法
21 ##### ハルトン数列関数
22 halton <- function(n,x) {
23   no <- 1:n
24   hal <- numeric(n)
25   i <- 1
26   while (n >= x^i) {
27     i <- i + 1
28   }
29   for (j in i:1) {
30     hal <- hal + ( no %% (x^(j-1)) ) * (1/ x^j)
31     no <- no %% x^(j-1)
32   }
33   hal
34 }
35 rand1 <- matrix(qnorm(halton(hh*R,2)),hh,R)
36 rand2 <- matrix(qnorm(halton(hh*R,3)),hh,R)

```

ハルトン数列法

素数xの累乗を分母においた分数を0～1区間で複数生成し、qnorm関数を用いて標準正規分布に従うような順乱数にする手法。



$$S_{t+1} = \left\{ S_t, S_t + \frac{1}{x^t}, S_t + \frac{2}{x^t}, \dots, S_t + \frac{x-1}{x^t} \right\}$$



Qnorm関数で標準正規分布に従わせる←ハルトン数列

～対数尤度関数の設定～

```
40 - fr <- function(x) {  
41   ### パラメータの宣言：  
42   ## 定数項  
43   b1 <- x[1]  
44   b2 <- x[2]  
45   b3 <- x[3]  
46   b4 <- x[4]  
47  
48   ## 目的地までの所要時間  
49   dmu1 <- x[5] #平均  
50   dsigma1 <- x[6] #標準偏差  
51  
52   ## 料金  
53   fmu1 <- x[7] #平均  
54   fsigma1 <- x[8] #標準偏差  
55  
56   ## シミュレーション尤度のための変数を宣言  
57   SimLL = 0
```

パラメータ β に個人嗜好性を持たせたランダム係数を表現するために、平均、標準偏差を推定.

対数尤度をR回計算し、その和をRで除すことで対数尤度の平均を出す
→シミュレーション尤度.
その和の受け皿としてSimLLを定義.

MXLモデルによる推定-MXLの効用関数

～ランダム係数を定義～

平均

標準偏差

ハルトン数列法で生成した乱数

```

66  ##R回シミュレーションを行うループ
67  for (i in 1:R) {
68
69    ## 対数尤度のための変数を宣言
70    LL = 0
71
72    ##乱数からパラメータをシミュレーションで発生
73    d1 <- dmu1 + dsigma1 * rand1[,i]
74    f1 <- fmu1 + fsigma1 * rand2[,i]
75
76    ## 効用の計算:説明変数にしたい列を入れる。
77    train <- Data$代替手段生成可否train*exp(d1*Data$総所要時間train/100 +f1*Data$費用train/100 + b1*matrix(1,nrow =hh,ncol=1))
78    bus <- Data$代替手段生成可否bus *exp(d1*Data$総所要時間bus/100 +f1*Data$費用bus/100 + b2*matrix(1,nrow =hh,ncol=1))
79    car <- Data$代替手段生成可否car *exp(d1*Data$所要時間car/100 + b3*matrix(1,nrow =hh,ncol=1))
80    bike <- Data$代替手段生成可否bike *exp(d1*Data$所要時間bike/100 + b4*matrix(1,nrow =hh,ncol=1))
81    walk <- Data$代替手段生成可否walk *exp(d1*Data$所要時間walk/100 )
82
83    ### 選択確率の計算
84    ## 分母となる, 各々のexp(V)の和をつくる
85    deno <- (car + train + bus + bike + walk)

```

=確率的に変化するパラメータ:ランダム係数

$$\text{ランダム係数: } \beta_i = \bar{\beta}_i + \sigma \eta_i$$

$\bar{\beta}_i$: 平均, σ : 標準偏差, η_i : 乱数

～選択確率，対数尤度の定義～

```
109  ## それぞれ計算する
110 Ptrain <- Data$代替手段生成可否train*(train / deno)
111  Pbus   <- Data$代替手段生成可否bus  *(bus   / deno)
112  Pcar   <- Data$代替手段生成可否car  *(car   / deno)
113  Pbike  <- Data$代替手段生成可否bike *(bike  / deno)
114  Pwalk  <- Data$代替手段生成可否walk *(walk  / deno)
115
116  ## 選択確率が0になってしまった場合に起こる問題の回避
117  #
118  Ptrain <- (Ptrain!=0)*Ptrain + (Ptrain==0)
119  Pbus   <- (Pbus!=0)*Pbus   + (Pbus==0)
120  Pcar   <- (Pcar !=0)*Pcar   + (Pcar ==0)
121  Pbike  <- (Pbike !=0)*Pbike + (Pbike ==0)
122  Pwalk  <- (Pwalk!=0)*Pwalk + (Pwalk ==0)
123
124  ## 選択結果
125  #代表交通手段が「鉄道」なら，Ctrainに1が入る
126  Ctrain <- Data$代表交通手段 == "鉄道"
127  Cbus   <- Data$代表交通手段 == "バス"
128  Ccar   <- Data$代表交通手段 == "自動車"
129  Cbike  <- Data$代表交通手段 == "自転車"
130  Cwalk  <- Data$代表交通手段 == "徒歩"
131
132  ## 対数尤度の計算
133  #simllが最終的な対数尤度
134  LL <- colSums(Ctrain*log(Ptrain) + Cbus*log(Pbus) +
135               Ccar *log(Pcar)  + Cbike *log(Pbike) +Cwalk *log(Pwalk))
136
137  SimLL <- SimLL + LL
138
139  ##ループここまで
140  }
```

対数尤度の計算，パラメータ推定をR回(100回)行うため，待ち時間が長い。暇なので結果を出力。確認せずともよい人はprintの前に#を付けてコメント化してもOK。

```
142  ##計算の途中経過を表示
143  print(x)
144  print(SimLL/R)
145
146  ##計算反復回数Rで割る
147  SimLL <- SimLL / R
148
149  }
```

MXLモデルによる推定-結果の出力

～結果の出力～

```

164 - ##### 対数尤度関数frの最大化#####
165 res <- maxLik(fr,start=b0,method="BFGS")
166
167 ## パラメータ推定値、ハッセ行列
168 b <- res$estimate
169 hhh <- res$hessian
170
171 ## t値の計算
172 tval <- b/sqrt(-diag(solve(hhh)))
173
174 ## 初期尤度
175 L0 <- fr(b0)
176 ## 最終尤度
177 LL <- res$maximum
178
179 - ##### 結果の出力 #####
180 print(res)
181 ## 初期尤度
182 print(L0)
183 ## 最終尤度
184 print(LL)
185 ## p^2値
186 print((L0-LL)/L0)
187 ## 修正済p^2値
188 print((L0-(LL-length(b)))/L0)
189 ##パラメータ推定値
190 print(b)
191 ## t値
192 print(tval)

```

NL同様，対数尤度の最大化に
maxLik関数を使用。

Printで各種推定結果を出力！

```

> ## 初期尤度
> print(L0)
[1] -564.1796
> ## 最終尤度
> print(LL)
[1] -360.8475
> ## p^2値
> print((L0-LL)/L0)
[1] 0.3604032
> ## 修正済p^2値
> print((L0-(LL-length(b)))/L0)
[1] 0.3462233
> ##パラメータ推定値
> print(b)
[1] 0.5991486996 -1.3684323442 -1.2949274960 -1.2438858492 -9.0696974385 1.1313555021 -0.0473542711 -0.0006618126
> ## t値
> print(tval)
[1] 2.41674077 -4.08352565 -6.39496797 -6.23883735 -11.57108920 1.86000167 -1.11506572 -0.02403964

```

パラメータ推定値は左から
定数項×4，時間のランダム係数の平均および標準偏差，
料金のランダム係数の平均および標準偏差

参考文献:

- 1)北村隆一・森川高行・佐々木邦明・藤井聡・山本俊行,
交通行動の分析とモデリング
- 2)東京大学羽藤研究室, <http://bin.t.u-tkyo.ac.jp/kaken/>
- 3)Rdocumentation maxLik
<https://www.rdocumentation.org/packages/maxLik/versions/1.3-4/topics/maxLik>