多項ロジットモデル(MNL)の説明

ここでは, 配布した MNL モデルの推定コードを用いて, R のコードの内容と行っている計算の説明をします. コードの順を追って説明を行います.

1 データファイルの読み込み

```
1  ### Multinomial Logit model estimation
2
3  ### データファイルの読み込み
4  Data <- read.csv("C:/ensyu2011all.csv",header=T)
5  ## データ数:Data の行数を数える
6  hh <- nrow(Data)
7
```

3 行目で変数 Data に ensyu.csv のデータを入れます. この際にデータファイルの場所(ここでは "C:/ensyu.csv"を指定)が正しく指定されていなければエラーが出てしまいます. ディレクトリ(フォルダ)を表す記号が, ”¥”ではなく”/”であることに注意してください. header は1行目に列名が書かれているかどうかを示します. 書かれているなら T, 書かれていないなら F を選択してください.

nrow は行列の行数, ncol は列数を示します. ここでは Data の行数, すなわちサンプル数を数え, 変数 hh に代入しています. R では“<-“が変数の代入を表す記号で, 以降同様です.

また, R では, 行内の”#”以降はコメントとなり, 計算に反映されません.

2 パラメータの初期値の設定

```
8  ## パラメータの初期値の設定
9  b0 <- numeric(6)
10
```

数値計算では通常初期値を 0 と置きます. 今回の例ではパラメータは 6 個(後述)あるため, 0 を 6 個並べることになります. numeric(6)は 0 の要素が 6 つ並んだベクトルを生成する意味です. なお, 推定するパラメータの数と同じ数の要素を作成しておく必要があることに注意してください.

例)パラメータを 6 個から 10 個に変更する場合 : “b0 <- numeric(6)” → “b0 <- numeric(10)”

3 パラメータの宣言

```
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    ## 定数項
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    ## 対数尤度のための変数を宣言
28    LL = 0
29
```

推定するパラメータの数と同じ数の変数を宣言します. 配布コードでは交通手段選択の MNL モデルを作るにあたって,「定数項」,「目的地までの所要時間」,「料金」を利用しています.

また, LL は後で対数尤度の計算に使うための変数で, 初期値として 0 を代入します.

x[1]は変数 x の 1 番目の要素, x[2]は 2 番目の要素…を示しています. また, 変数が行列である場合は x[3, 4]というように[行, 列]の順で示します.

4 効用 V の計算

```
30    ### 今回用いる目的地は以下の 5 つ.
31    ## 鉄道(train)
32    ## バス(bus)
33    ## 自動車(car)
34    ## 自転車(bike)
35    ## 徒歩(walk)
36
37    ## 効用の計算:説明変数にしたい列を入れる.
38                                        # 時間                          # 料金
    # 定数項
39    train <- Data$代替手段生成可否 train * exp( d1*Data$総所要時間 train/100
    + f1*Data$費用 train/100 + b1*matrix(1,nrow =hh,ncol=1))
40    bus   <- Data$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))
41    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))
42    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))
43    walk  <- Data$代替手段生成可否 walk  * exp( d1*Data$所要時間 walk/100)
44
```

まずは各交通手段についての効用を計算します.

式は多項ロジットモデル

$$P_n(i) = \frac{\exp(\mu V_{in})}{\sum_{j=1}^{J} \exp(\mu V_{jn})}, i = 1,...,J$$

の　$\exp(\mu V_{in})$の部分を示します.

Data$代替手段生成可否 train は, 先ほどのデータファイルを読み込んだ変数"Data"の train の代替手段生成可否の行を示しています.

Data$総所要時間 train は"Data"の train の総所要時間(アクセス・イグレスを含め, 目的地までの所要時間)のすべての行を示しています.

同様に, Data$費用 train は"Data"の train の目的地までの費用のすべての行を示します.

なお, matrix(1,nrow=hh, ncol=1)は行数がサンプル数(hh), 列数が1の, 要素がすべて 1 である行列を作成せよという命令文です. ちなみに matrix(1:6, nrow=2, ncol=3)だと$\begin{pmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{pmatrix}$となります. ここでは,

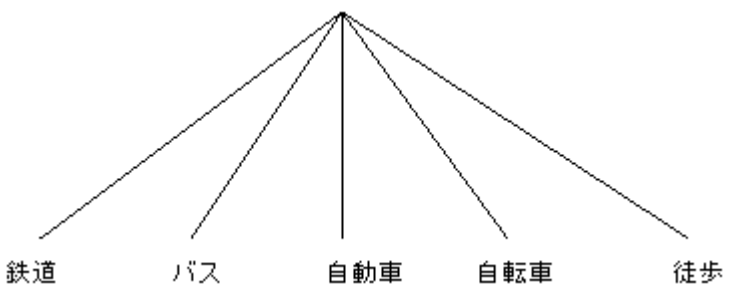
定数項の定義のために用いています. ちなみに, 定数項は全ての選択肢に含めると識別できなくなることに注意してください.

また, 所要時間と費用については, 説明変数のおよそのケタを揃えて推定計算を安定させるため, 100 で割り, それぞれ 100 分, 100 円を単位として変数をとっています.

5 交通手段についての多項ロジットモデル

```
45  ### 選択確率の計算
46  ## 分母となる, 各々の exp(V) の和をつくる
47  deno <- (car + train + bus + bike + walk)
48
49  ## それぞれ計算する
50  Ptrain <- Data$代替手段生成可否 train*(train / deno)
51  Pbus   <- Data$代替手段生成可否 bus  *(bus   / deno)
52  Pcar   <- Data$代替手段生成可否 car  *(car   / deno)
53  Pbike  <- Data$代替手段生成可否 bike *(bike  / deno)
54  Pwalk  <- Data$代替手段生成可否 walk *(walk  / deno)
55
```

今回の例におけるモデルは以下の図で表されます.



下式をそのまま計算しています.

$$P_n(i) = \frac{\exp(\mu V_{in})}{\sum_{j=1}^{J} \exp(\mu V_{jn})}, i = 1,...,J$$

6 選択確率の補正

```
56   ## 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
57   Ptrain  <- (Ptrain!=0) * Ptrain + (Ptrain ==0)
58   Pbus    <- (Pbus!=0)   * Pbus   + (Pbus   ==0)
59   Pcar    <- (Pcar!=0)   * Pcar   + (Pcar   ==0)
60   Pbike   <- (Pbike!=0)  * Pbike  + (Pbike  ==0)
61   Pwalk   <- (Pwalk!=0)  * Pwalk  + (Pwalk  ==0)
62
```

5 では, 仮に鉄道を利用不可能な人については, 鉄道の選択確率が 0 になります. それを, 対数尤度を計算する 8 の部分に代入すると, ln (0)の値が計算できず, エラーとなってしまいます(y=ln(x)において x>0 でなくてはなりません).

このプログラム上の問題を回避するために, 5 で計算された Ptrain が 0 でなかったらそのまま Ptrain を, Ptrain が 0 であれば 1 を Ptrain に代入するような処理をしています.

※ "!="は not equal を意味します.

※ (PPtrain!=0)や(PPtrain==0)といった論理式は真のとき 1, 偽のとき 0 を返します.

7 選択結果

```
63   ## 選択結果
64   Ctrain  <- Data$代表交通手段 == "鉄道"
65   Cbus    <- Data$代表交通手段 == "バス"
66   Ccar    <- Data$代表交通手段 == "自動車"
67   Cbike   <- Data$代表交通手段 == "自転車"
68   Cwalk   <- Data$代表交通手段 == "徒歩"
69
```

Ctrain <- Data$代表交通手段=="鉄道"は"Data"の代表交通手段の列が「鉄道」である行には 1 を, そうでない行には 0 を出力し, Ctrain に代入することを示しています.

8 対数尤度の計算

```
70   ## 対数尤度の計算
71   LL <- colSums(Ctrain*log(Ptrain) + Cbus*log(Pbus) +
72                 Ccar  *log(Pcar)   + Cbike  *log(Pbike) +Cwalk *log(Pwalk))
73
74 }
75
```

式は$\ln L = \sum_{n=1}^{N}\sum_{i=1}^{J} d_{in} \ln P_n(i)$を計算しています.

※colSums()は列の総和, rowSums()は行の総和を表します.

74 行目の"}"は 13 行目で定義した対数尤度を定義する関数を締めくくる括弧です.

※最尤推定法について:

行動を表す理論モデルが正しいとの仮定のもとで, 観測されたデータが得られる尤もらしさ(尤度)が最大になるようにパラメータを定めます.

離散選択モデルのもとでの, データの尤度は選択確率を用いて次式で表されます.

$$L = \prod_{n=1}^{N}\prod_{j=1}^{J} P_n(i)^{d_m}$$

d_{in}:個人nが選択肢iを選択したとき1, そうでないとき 0. この式に対して両辺に対して対数をとることで簡易にしたものが上記の式です.

以上が対数尤度関数の定義となっています.

9 対数尤度関数の最大化

```
76  ##### 対数尤度関数 fr の最大化#####
77
78  ##パラメータ値の最適化
79  res <- optim(b0,fr, method = "Nelder-Mead", hessian = TRUE, control=list(fnscale=-1))
80
```

パラメータ"b0"に対して対数尤度関数"fr"を最大化し, その結果を"res"に出力します.

※optim 関数について

optim 関数は以下のような引数をとります.

```
optim(par, fn, gr = NULL, method = "Nelder-Mead", lower = -Inf, upper = Inf,
control = list(), hessian = FALSE)
```

par	初期値(必須)
fn	最適化する目的関数(必須)
gr	一階偏微分関数の指定(NULL でデフォルト関数使用)

method	最適化手法の指定(5 種類から選ぶ, Nelder-Mead(デフォルト), BFGS, CG, L-BFGS-B, SANN)
lower	L-BFGS-B 法での変数の下限(デフォルトは-Inf)
upper	L-BFGS-B 法での変数の上限(デフォルトは Inf)
control	制御パラメータ fnscale 関数に与える比例定数, optim 関数は最小化を行うので, 最大化のときは, control=list(fnscale=-1)と指定します. ここでは尤度を最大化しています. hessian 最適解のヘッセ行列を返すかどうか, パラメータ推定では, t 値計算にヘッセ行列が必要なので, hessian = TRUE とします.

10 パラメータ, ヘッセ行列の抜き出し

```
81  ## パラメータ推定値, ヘッセ行列
82  b   <- res$par
83  hhh <- res$hessian
84
```

最大化された結果である"res"から, パラメータ(par)およびヘッセ行列(hessian)を抜き出しています.

11 t 値の計算

```
85  ## t 値の計算
86  tval <- b/sqrt(-diag(solve(hhh)))
87
```

sqrt は平方根の計算, diag は行列の作成を行います(その際の設定に hhh の逆行列を指定するための関数 solve を使っています).

12 初期尤度, 最終尤度の算出

```
88  ## 初期尤度
89  L0 <- fr(b0)
90  ## 最終尤度
91  LL <- res$value
92
```

初期尤度は, パラメータを全て 0 としたときの尤度です.

最終尤度は, パラメータ推定値を求めたときに最大化された尤度です.

13 結果の表示

```
 93  ##### 結果の出力 #####
 94  print(res)
 95  ## 初期尤度
 96  print(L0)
 97  ## 最終尤度
 98  print(LL)
 99  ##ρ^2 値
100  print((L0-LL)/L0)
101  ## 修正済ρ^2 値
102  print((L0-(LL-length(b)))/L0)
103  ##パラメータ推定値
104  print(b)
105  ## t 値
106  print(tval)
107
```

初期尤度，最終尤度，尤度比，修正済み尤度比，パラメータ推定値，t 値を表示させます．

14 結果の表示と解釈

以上のコードを実行すると，計算が行なわれ，その結果が表示されます．入力が赤字で，出力が青字となっています．前半の入力の部分は省略し，"## 結果の出力"以降の部分を以下に示します．

```
> print(res)
$par
[1]  0.5166185024 -1.7122114083 -1.5555058043 -1.3701750852 -0.1111461187  0.0001876913

$value
[1] -1265.759

$counts
function gradient
     104       22

$convergence
[1] 0

$message
NULL

$hessian
             [,1]         [,2]         [,3]       [,4]          [,5]         [,6]
[1,]   -173.841516    7.556166   119.703557   27.11163 -2.452542e+03   -62600.711
[2,]      7.556166  -34.187062     9.432767    7.43628  4.993405e+01    -5191.759
[3,]    119.703557    9.432767  -225.822769   53.23063  2.626840e+03    54186.559
[4,]     27.111629    7.436279    53.230633 -146.45477  6.721227e+02     7664.358
[5,]  -2452.541643   49.934052  2626.840222  672.12266 -8.258336e+04  -1094200.328
[6,] -62600.710523 -5191.758620 54186.559083 7664.35787 -1.094200e+06 -36048315.893
```

res は optim 関数の出力内容です．それぞれの意味は以下の通りです．

par	パラメータ値の最適解の値
value	最適化関数の目的値の最適値（ここでは最大尤度）
counts	function は最適化の過程での計算の繰り返し回数，gradient は計算の中で 1 階偏微分計算が行われた回数です．

convergence	収束判定です. 0 ならば収束しています. 0 以外の値の場合は, 効用関数や尤度関数の設定に何らかの誤りがある可能性があります.
messege	エラーが出た場合にエラーメッセージを表示します. NULL ならばエラーはありません.
hessian	ヘッセ行列です. このヘッセ行列を用いて t 値の計算を行っています.

```
> ## 初期尤度
> print(L0)
[1] -2109.710
> ## 最終尤度
> print(LL)
[1] -1265.759
> ##ρ^2 値
> print((L0-LL)/L0)
[1] 0.4000319
> ## 修正済ρ^2 値
> print((L0-(LL-length(b)))/L0)
[1] 0.3971879
> ##パラメータ推定値
> print(b)
[1]  0.5166185024 -1.7122114083 -1.5555058043 -1.3701750852 -0.1111461187  0.0001876913
> ## t 値
> print(tval)
[1]   3.4565899  -8.6617975 -13.2299589 -12.4937770 -20.4124961   0.6119783
>
>
```

ここでは, 初期対数尤度, 最終対数尤度, 尤度比, 修正済み尤度比(パラメータの数を考慮した尤度比), パラメータ推定値, t 値を表示しています. パラメータ推定値と t 値の表示順序は, 尤度関数の中での x[1], x[2],…の順番と対応しています.