Mixed Logit モデル(MXL)の説明

MNL との相違点のみ述べます.

1 乱数の発生

```
##### 乱数の発生 #####

## 乱数の発生回数
R <- 100

## 乱数の発生-ランダムドロー法
#rand1 <- matrix(rnorm(hh*R),hh,R)
#rand2 <- matrix(rnorm(hh*R),hh,R)

## 乱数の発生-ハルトン数列関数による方法
#### ハルトン数列関数
halton <- function(n,x) {
  no <- 1:n
  hal <- numeric(n)
  i <- 1
  while (n >= x^i) {
    i <- i + 1
  }
  for (j in i:1 ) {
    hal <- hal + ( no %/% (x^(j-1)) ) * (1/ x^j)
    no <- no %% x^(j-1)
  }
  hal
}
rand1 <- matrix(qnorm(halton(hh*R,2)),hh,R)
rand2 <- matrix(qnorm(halton(hh*R,3)),hh,R)

```

MXL では積分計算を数値計算で行うため，シミュレーションによるアプローチが必要となります．計算ではシミュレーションを乱数の発生により行います．ここでは，正規分布から乱数を抽出します．ここではランダムドロー法とハルトン数列法の２種類を選択できるようにしています．配布コードでは，ランダムドロー法による抽出は#でクロスアウトして無効として，ハルトン数列法を採用しています．

※ランダムドロー法

`rnorm(n)`は平均 0，標準偏差 1 の標準正規分布から n 個の乱数をベクトルとして生成する関数です．ここでは行方向にサンプル数，列方向に乱数発生回数を取った行列の各要素にそれぞれ独立な乱数を入れています．

※ハルトン数列法

ハルトン数列法は乱数の代わりに，分布範囲を均等に網羅し，かつ相互に相関などを持たない数列を準乱数として用いる方法です．ランダムドロー法よりも効率よく分布範囲をカバーできる利点があります．

数列は，ベースとなる素数を選び，(0,1)区間をその数で分割することを繰り返すことにより得られます．素数を 3 とした場合，{1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, 8/9, 1/27, 10/27, 19/27, 4/27,…}となります．`halton(n,x)` でここでは素数 `x` をベースとした準乱数を `n` 個発生させる関数として定義しています．`qnorm` は標準正規分布の確率密度関数の逆関数であり，(0,1)区間をとるハルトン数列を，標準正規分布の準乱数の数列にしています．

ハルトン数列法について詳しくは Train(1999)を参照してください．

2 パラメータの宣言

```
## 目的地までの所要時間
dmu1 <- x[5] #平均
dsigma1 <- x[6] #標準偏差

## 料金
fmu1 <- x[7] #平均
fsigma1 <- x[8] #標準偏差

```

MXL では個人 n が選択肢 i を選ぶ効用関数は以下のように表されます．

$$U_{in} = V_{in} + \eta_{in} + \varepsilon_{in}$$
$$V_{in} = \beta_i X_{in}$$

Vin はパラメータ β，説明変数ベクトル X で表される部分，η は平均 0，標準偏差 σ の正規分布に従う部分です．ε は IID ガンベル分布です．

ここではパラメータを，パラメータ自体が確率的に変動するランダム係数として，パラメトリックに表します．このランダム係数が，サンプル間の異質性を考慮した部分になります．配布コードでは，「目的地までの所要時間」と「料金」のパラメータの平均と標準偏差をそれぞれ変数として定義しています．ここでは，変数間での誤差相関は考慮していません．

3 シミュレーション計算パラメータの宣言

```
##R 回シミュレーションを行うループ
for (i in 1:R) {

（略）

  ##乱数からパラメータをシミュレーションで発生
  d1 <- dmu1 + dsigma1 * rand1[,i]
  f1 <- fmu1 + fsigma1 * rand2[,i]

（略）

  ## 対数尤度の計算
  LL <- colSums(Ctrain*log(Ptrain) + Cbus*log(Pbus) +
          Ccar  *log(Pcar)   + Cbike  *log(Pbike) +Cwalk *log(Pwalk))

  SimLL <- SimLL + LL

##ループここまで
}

```

73/74 行目は,「目的地までの所要時間」と「料金」のパラメータを平均と標準偏差, また標準正規分布から抽出した乱数によって表しています. `[,i]`は行列の i 列目を取り出す意味です. MNL と同様に対数尤度は計算され, 関数の中で対数尤度の計算は R 回繰り返されます.

4 途中経過の表示

```
117 ##計算の途中経過を表示
118 print(x)
119 print(SimLL/R)
120
```

この後で行うパラメータ推定の尤度最大化の最適計算では, シミュレーションの計算を繰り返すため時間がかかります. そこで, ここでは最適化計算のパラメータ値と尤度についての途中経過を, 計算中に表示させるようにしています.

5 シミュレーション尤度の計算

```
121 ##計算反復回数 R で割る
122 SimLL <- SimLL / R
123
```

MXL での, データのシミュレーション対数尤度は選択確率を用いて次式で表されます.

$$\ln SL = \frac{1}{R}\sum_{r=1}^{R}\sum_{n=1}^{N}\sum_{i=1}^{J} d_{in} \ln P_n(i)$$

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.

`SimLL` はこのシミュレーション対数尤度を表しており, これを最大化するパラメータを求めます.

以降の計算と結果の表示は MNL と同様です.