

# 離散選択モデル入門

Introduction to Discrete Choice Models

林 由翔

Yuito Hayashi

2022 年 2 月 作成

2023 年 4 月 最終更新

# はじめに

この文書は、人間の選択行動を推定するときに用いられる「離散選択モデル」と、離散選択モデルの中で代表的なモデルである「多項ロジットモデル」(Multinomial Logit Model, MNL) について、理論および実装方法をまとめたものです。実装パートでは多項ロジットモデル (以下 MNL と書きます) を用いて交通手段選択モデルを実装します。このため理論パートも交通分野での利用を前提とした説明となっています。

# 目次

はじめに	i
<b>第 1 章 離散選択モデル</b>	<b>1</b>
1.1 離散選択モデルとは	1
1.2 効用最大化の仮定	1
1.3 確率計算とロジットモデル	2
1.3.1 プロビットモデル	2
1.3.2 ロジットモデル	3
1.4 仮定の整理	5
1.5 IIA 特性	5
<b>第 2 章 効用関数</b>	<b>7</b>
2.1 効用の立式	7
2.1.1 より複雑な立式	8
<b>第 3 章 パラメータ推定</b>	<b>9</b>
3.1 パラメータ推定に用いるデータ	9
3.2 尤度	10
3.2.1 対数尤度	11
3.2.2 未知パラメータが一意に定まるような効用式	11
<b>第 4 章 検定</b>	<b>12</b>
4.1 t 検定	12
4.1.1 統計量	12
4.1.2 推定値と推定量	12
4.1.3 区間推定	14
4.2 尤度比指標	15
<b>第 5 章 Python による MNL の実装</b>	<b>16</b>
5.1 ファイルの読み込み等	16
5.2 効用の計算	16
5.3 選択確率の計算	17
5.4 対数尤度の計算	17
5.5 最尤推定とその評価	19

目次	iii
5.6      コード全体 . . . . .	21
参考文献	23

## 第1章

# 離散選択モデル

### 1.1 離散選択モデルとは

現実世界では、日々様々な現象が起こっており、その現象の要因もまた複雑に入り組んでいます。こうした膨大な数の現象とその要因を、いくつかの仮定の下で整理し、現象を簡潔な形で表現したもののことをモデルといいます。

**離散選択モデル**とは、人間の選択行動に関するモデルであって、その選択結果が離散的であるものを指します。

例えば、いくつかある交通手段のうちから一つを選択する交通手段選択行動は、離散選択モデルの枠組みで表現することができます。また、観光客がどの目的地を選んで行くかといった目的地選択行動も、離散選択モデルで表現できます。なお、交通手段選択行動をモデルにしたものを**手段選択モデル**、目的地選択行動をモデルにしたものを**目的地選択モデル**等と呼んでいる資料もあります。手段選択モデルや目的地選択モデルは、多くの場合は離散選択モデルとして表現されます。

離散選択モデルを作る目的は、ひとことでいうと、**ある条件においてどの選択肢がどのくらいの確率で選ばれるのかを推定すること**です。各選択肢の選択確率が分かれば、現在の総トリップ数に選択確率を掛けることで、乗客数や来訪者数を推定できます。すると、運賃を下げたり所要時間を短縮したりすることが乗客数にどう影響するのか、広場の整備が来訪者数にどう影響するのか、といったことを数値的に示すことができます。

### 1.2 効用最大化の仮定

離散選択モデルにおいて一般的に用いられる仮定として、「**人は効用を最大にする選択肢を選ぶ**」という**仮定**があります。これはミクロ経済学の最も主要な仮定の一つです。

交通手段選択行動を例にとってみると、人は交通手段を選ぶときに「所要時間」「運賃・駐車場代」「(公共交通機関であれば)頻度」といった様々な要素を天秤にかけて、その中で最終的に一つの交通手段を選んでいきます。効用とは、これらの要素を総合的にふまえて決められる単一の数値であり、この効用の値が最も大きくなるような選択肢が常に選ばれる、というのが効用最大化の仮定です。

このように書くと、特定の選択肢に全員が集中してしまうようにも思えますが、そのようなことはありません。

まず、効用の値はその選択肢の性質だけではなく、選択行動をする個人の属性によっても変わるものと考えます。例えば東京から大阪へ行くのに、多くの人は新幹線を使うでしょうが、お金のない学生にとっ

ては運賃が高く感じられ、効用が大きく下がります。逆に夜行バスの効用は、体力や時間的自由がある学生にとってはあまり下がらないかもしれません。

次に、属性が同じであったとしても、最終的な効用は確率的に変動すると考えます。冒頭にも記した通り、モデル化に際しては様々な仮定を置き、現象を簡略化して表現しているので、モデル化に際して考慮されなかった要素を確率的に変化する誤差項としてまとめて考えます。

このような仮定をおくことで、「属性 P の人は選択肢 X を選ぶ傾向にある」といった現象や、「全員が X を選ぶのではなく中には選択肢 Y を選ぶ人もいる」といった現象を表現することができます。

## 1.3 確率計算とロジットモデル

### 1.3.1 プロビットモデル

1.2 節で述べた通り、効用は確率的に変動するとします。自然な仮定として、効用は正規分布に従うものとすることができます。なお、効用が正規分布に従うとするモデルを**プロビットモデル**と呼びます。

引き続き、新幹線と夜行バスの例を使うことにします。ある学生 A さんにとって、東京から大阪まで新幹線で移動する場合と夜行バスで移動する場合の効用は図 1.1 のようになりました。

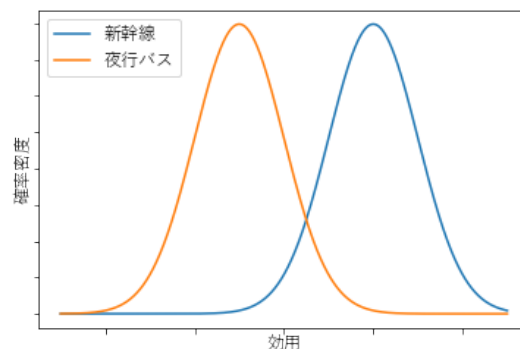


図 1.1 新幹線と夜行バスの効用の比較

A さんにとっては新幹線の方が効用が大きくなる可能性が高く、新幹線が選ばれる確率が高そうです。しかし、必ず新幹線が選ばれるとは限りません。

以下、A さんにとっての新幹線の効用を  $U_S$ 、夜行バスの効用を  $U_B$  と書くことにします。 $U_S, U_B$  は確率変数になります。ここでは  $U_S, U_B$  が**独立であることを仮定**します。

このとき、A さんが新幹線を選択する確率は、 $U_S$  が  $U_B$  より大きくなる確率であり、 $P(U_S - U_B \geq 0)$  と書くことができます。 $U_S$  と  $U_B$  が正規分布に従う場合、 $U_S - U_B$  も正規分布に従う<sup>\*1</sup>ため、 $P(U_S - U_B \geq 0)$  の値は正規分布の累積確率として求めることができます。

ここに第三の選択肢、飛行機を加えます。この時の効用は図 1.2 のようになります。

A さんにとっての飛行機の効用を  $U_P$  と書くことにします（当然これも確率変数です）。このとき、この 3 つの選択肢の中から A さんが新幹線を選択する確率は、 $U_S$  が  $U_B$  と  $U_P$  より大きくなる確率であり、 $P(U_S - U_B \geq 0 \wedge U_S - U_P \geq 0)$  と書くことができます。また、これを書き換えて  $P(U_S - \max(U_B, U_P) \geq 0)$  とすることもできます。

<sup>\*1</sup> 正規分布の和・差の再生性として知られています。

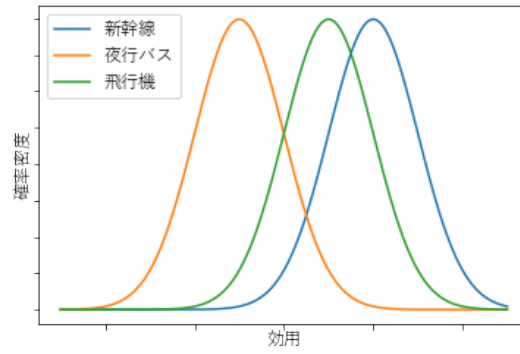


図 1.2 新幹線と夜行バスと飛行機の効用の比較

$U_S, U_B, U_P$  が正規分布に従う場合、この確率を計算するのは簡単ではありません。 $\max(U_B, U_P)$  が正規分布に従わないため、多重積分によって累積確率を求める必要があるからです。

このように、値を求めるのに積分計算が必要になるような式のことを**閉形式**といい、数値計算上処理が重くなることから嫌われます。先ほどの選択肢が二つしかないバージョンも、正規分布の累積確率を計算するには積分が要するため、閉形式です。

選択肢が二つ（二項）の場合であれば、単なる正規分布の累積確率なので数値計算ライブラリを使えば高速に値を求めることができますが、選択肢が三つ以上（多項）で多重積分が必要となると、計算時間が実用上問題になってくる恐れがあります。また数値積分なので誤差の懸念もあります。

そこで、**正規分布に似た形状の分布を正規分布の代わりに用いる**ことで、積分計算なしで選択確率を計算することが考えられました。

### 1.3.2 ロジットモデル

正規分布に形状が似ていて、離散選択モデルの計算上都合がいい性質を持っている分布として、**ガンベル分布**が用いられます。効用がガンベル分布に従うと仮定したモデルのことを**ロジットモデル**と呼びます。

ガンベル分布の確率密度関数は式 (1.1) の通りです。

$$f(x) = \mu \exp(-\mu(x - \eta)) \exp(-\exp(-\mu(x - \eta))) \quad (1.1)$$

ここで、 $\mu$  はガンベル分布の**スケールパラメータ**、 $\eta$  は**ロケーションパラメータ**と呼ばれます。正規分布において平均値  $\mu$  と分散  $\sigma^2$  が与えられれば分布の形状が分かるのと同様に、ガンベル分布においてスケールパラメータとロケーションパラメータが分かれば分布の形状が分かります。

スケールパラメータ  $\mu$  は分布のばらつきを意味する値であり、 $\mu > 0$  でなければいけません。 $\mu < 0$  だと確率密度関数が負値になってしまいます。

ロケーションパラメータ  $\eta$  は分布の最頻値に一致します。

ガンベル分布は、2つの都合の良い性質を示します。

**定理 1.** 互いに独立な確率変数  $X_1, X_2, \dots, X_i, \dots, X_N$  が、それぞれガンベル分布  $Gb(\eta_1, \mu), Gb(\eta_2, \mu), \dots, Gb(\eta_i, \mu), \dots, Gb(\eta_N, \mu)$  に従うとき、 $\max(X_1, X_2, \dots, X_i, \dots, X_N)$  は式 (1.2)

に示すパラメータを持つガンベル分布に従う。

$$\left( \frac{1}{\mu} \ln \sum_{i=1}^N \exp(\mu \eta_i), \mu \right) \quad (1.2)$$

**定理 2.** 独立な二つの確率変数  $X_1, X_2$  がそれぞれガンベル分布  $Gb(\eta_1, \mu), Gb(\eta_2, \mu)$  に従うとき、その差  $X_1 - X_2$  は、式 (1.3) に示す累積分布関数を持つ**ロジスティック分布**に従う。

$$F(x) = \frac{1}{1 + \exp(\mu(\eta_2 - \eta_1 - x))} \quad (1.3)$$

※ロケーションパラメータは変数ごとに異なってもよいですが、**スケールパラメータは同じである**必要があります。

この二つの性質を用いることで、積分計算なしに選択確率を計算することができます。

効用  $U_S, U_B, U_P$  が互いに独立でガンベル分布に従うとすると、新幹線の選択確率  $P(U_S - \max(U_B, U_P) \geq 0)$  を計算します。

まずここで、3つの交通手段の効用が**従うガンベル分布は、全てスケールパラメータが同一であること**を仮定します。以下、 $U_S \sim Gb(V_S, \mu), U_B \sim Gb(V_B, \mu), U_P \sim Gb(V_P, \mu)$  であるとします。

すると定理 1 により、

$$\max(U_B, U_P) \sim Gb\left(\frac{1}{\mu} \ln(\exp(\mu V_B) + \exp(\mu V_P)), \mu\right) \quad (1.4)$$

となります。

ここで  $V_S^* = \frac{1}{\mu} \ln(\exp(\mu V_B) + \exp(\mu V_P))$  とおき、 $U_S = V_S + \varepsilon_S, \max(U_B, U_P) = V_S^* + \varepsilon_S^*$  となるように  $\varepsilon_S, \varepsilon_S^*$  を定めると、 $\varepsilon_S, \varepsilon_S^*$  はいずれも  $Gb(0, \mu)$  に従う確率変数になります。

なお、 $V_S$  を**確定部分**、 $\varepsilon_S$  を**誤差項**と呼ぶことがあります。選択肢の効用  $U$  は確定部分と誤差項の和で表されます。

定理 2 を用いて、

$$\begin{aligned} P(U_S - \max(U_B, U_P) \geq 0) &= P(V_S + \varepsilon_S - V_S^* - \varepsilon_S^* \geq 0) \\ &= P(\varepsilon_S^* - \varepsilon_S \leq V_S - V_S^*) \\ &= \frac{1}{1 + \exp(\mu(V_S^* - V_S))} \\ &= \frac{\exp(\mu V_S)}{\exp(\mu V_S) + \exp(\mu V_S^*)} \\ &= \frac{\exp(\mu V_S)}{\exp(\mu V_S) + \exp(\mu \cdot \frac{1}{\mu} \ln(\exp(\mu V_B) + \exp(\mu V_P)))} \\ &= \frac{\exp(\mu V_S)}{\exp(\mu V_S) + \exp(\mu V_B) + \exp(\mu V_P)} \end{aligned} \quad (1.5)$$

より一般に、 $N$  個の選択肢があって  $i$  番目の選択肢の効用が  $Gb(V_i, \mu)$  に従うとき、 $j$  番目の選択肢が選択される確率は、以下の式 (1.6) で表されます。

$$\frac{\exp(\mu V_j)}{\sum_{i=1}^N \exp(\mu V_i)} \quad (1.6)$$



最後にもう一つ、 $\mu = 1$  であると仮定すれば、選択確率は以下の式 (1.7) で表されます。分子分母を  $\mu$  で割ったわけではないことに注意してください。

$$\frac{\exp(V_j)}{\sum_{i=1}^N \exp(V_i)} \quad (1.7)$$

ロジットモデル、特に項が 3 つ以上の**多項ロジットモデル (MNL)** では、各選択肢の効用の確定部分  $V_i$  さえわかれば、式 (1.6) や式 (1.7) のような平易な式で各選択肢の選択確率を求めることができます。積分が不要な**閉形式**になっているため、実用上も高速です。

## 1.4 仮定の整理

上の選択確率の式 (1.7) を導くために、様々な仮定をおいてきました。モデルを実際に適用する際には、これらの仮定がそもそも成り立っているとみなしてよいのか、十分に注意する必要があります。以下、この章でおいた仮定を列挙します。

1. 人は効用を最大にする選択肢を選ぶ。
2. 効用にはランダム性があるが、その分布はガンベル分布に従う。
3. 各選択肢の効用の分布は互いに独立である。
4. 効用が従うガンベル分布のスケールパラメータはどの選択肢でも同じである。すなわち、効用のばらつきはどの選択肢でも同じである。
  - これは効用の誤差項の分布が全て等しい ( $Gb(0, \mu)$  に従う) ことを意味する。
  - 3 番目の仮定と合わせて、各選択肢の効用の誤差項は互いに独立かつ同分布であり、このような条件を **IID** (Independent and Identically Distributed, 独立同分布) と呼ぶ。
5. スケールパラメータ  $\mu = 1$  である。

## 1.5 IIA 特性

最後に、ロジットモデルの **IIA 特性**に触れます。

IIA (Independence from Irrelevant Alternatives) 特性とは、二つの選択肢の間の選択確率の比は、その二つの選択肢の効用のみによって決まるという性質のことです。例えば MNL では二つの選択肢  $i, j$  の間の選択確率の比は  $\exp(V_i) : \exp(V_j)$  と表されるため、確かに IIA 特性を持っているといえます。ロジットモデルの IIA 特性は、各選択肢の効用の分布が互いに独立であるという仮定によって生じたものです。

効用の分布が相関するような（似ている）選択肢がある場合、MNL における選択確率の計算結果が期待されるものと大きく異なってしまいます。この問題は「**赤バス青バス問題**」と呼ばれています。

赤バス青バス問題は、次のような問題です。

## 赤バス青バス問題

「車」「赤バス」の二つの選択肢から選ぶロジットモデルにおいて、効用の確定部分が全く同じであれば、両者の選択確率は  $1/2, 1/2$  となる。ここに「青バス」という選択肢を加えたとき、これの効用の確定部分も他と同じであったとすれば、三つの選択肢の選択確率は  $1/3, 1/3, 1/3$  となる。色違いのバスが登場しただけでもともと車を利用していた人の三分の一もがバスに流れるのはおかしい。

赤バス青バス問題は、車と赤バスの選択確率の比が、青バスの有無によらず  $1:1$  であるという IIA 特性によって生じる問題です。

MNL で選択確率を適切に推定するには、選択肢の中で互いに似ているものがないことが必要です。赤バス青バス問題が起こりそうな場合には、**Nested Logit モデル**などのより応用的なモデルを用いることになります。

## 第2章

# 効用関数

1章では、各選択肢の効用の確定項  $V$  が分かれば、各選択肢の選択確率が計算できることを示しました。この章では、多項ロジットモデルにおいて効用の確定項  $V$  をどのように求めるかを示します。

### 2.1 効用の立式

まず、効用は連続変数であり、いくつかの説明変数の線形和であることを仮定します。

**説明変数**とは、各選択肢の定量可能な変数のことを指します。1.2節で、効用の値はその選択肢の性質や選択行動をする個人の属性によって変わると書きました。例えば手段選択モデルの場合、下記のようなものが説明変数になりえます。

- 所要時間
- 運賃
- 乗り換え回数
- 年齢
- 免許を持っているかどうか（持っていれば 1, 持っていなければ 0）
- ある交通手段の定期券を持っているかどうか

説明変数は大きく分けて、それぞれの交通手段に固有の特性と、選択をする個人の属性の2種類あるといえます。

これらの説明変数は、もちろん単位もバラバラですし、何よりどの指標がどれだけ効用に強く影響するかは分かりません。そこで、これらの説明変数をただ足し上げるのではなく、各説明変数に「**重み**」を掛けてから足し上げる必要があります。

例えば、東京から大阪へ行く選択肢が「新幹線」「夜行バス」「飛行機」の三択で、各選択肢の効用が「所要時間」「運賃」「年齢」で決まるとします。このときの新幹線の効用は、 $x_{S,time}, x_{S,cost}, x_{age}$ <sup>\*1</sup> がそれぞれ新幹線の所要時間・新幹線の運賃・個人の年齢であるとする、以下の式 (2.1) のようにおくことができます。

$$V_S = \beta_{time}x_{S,time} + \beta_{cost}x_{S,cost} + \beta_{S,age}x_{age} \quad (2.1)$$

式 (2.1) の  $\beta$  が「重み」に相当する値で、未知のパラメータです。

---

<sup>\*1</sup> 個人の属性は選択肢によって変わることはない、ので、 $x_{S,age}$  ではなく  $x_{age}$  です。

同様に、夜行バスと飛行機の効用も式にしてみます。

$$V_B = \beta_{time}x_{B,time} + \beta_{cost}x_{B,cost} + \beta_{B,age}x_{age} \quad (2.2)$$

$$V_P = \beta_{time}x_{P,time} + \beta_{cost}x_{P,cost} + \beta_{P,age}x_{age} \quad (2.3)$$

所要時間のパラメータ  $\beta_{time}$  と運賃のパラメータ  $\beta_{cost}$  は3つの式(2.1)(2.2)(2.3)で共通なのに対し、年齢のパラメータは選択肢ごとに  $\beta_{S,age}, \beta_{B,age}, \beta_{P,age}$  と別の変数になっている点に注目してください。

所要時間や運賃など交通手段に固有の特性に掛ける重みは、すべての選択肢で同じにすることで、その特性がどう選択に影響するかを表すことができます。例えば、 $\beta_{time}$  が負の値であれば、時間がかかるほど ( $x_{S,time}, x_{B,time}, x_{P,time}$  が大きくなるほど) 効用が小さくなるということを意味します。

一方、個人の属性に掛ける重みは、選択肢ごとに分けることで、その属性の持ち主が相対的にどの選択肢を選びやすくなるかを表すことができます。例えば、 $\beta_{B,age} < \beta_{P,age}$  ならば、年齢が高いほど ( $x_{age}$  が大きくなるほど) 相対的に飛行機の方が効用が大きくなるということを意味します。

また、これらの説明変数では説明できない、各選択肢に固有の好まれ具合を表現するために、最後に説明変数を掛けないパラメータを付けます。なおこのとき、「説明変数を掛けないパラメータ」を全ての選択肢につけてはいけません。理由はこの後の3.2.2項で説明します。

$$V_S = \beta_{time}x_{S,time} + \beta_{cost}x_{S,cost} + \beta_{S,age}x_{age} \quad (2.4)$$

$$V_B = \beta_{time}x_{B,time} + \beta_{cost}x_{B,cost} + \beta_{B,age}x_{age} + \beta_B \quad (2.5)$$

$$V_P = \beta_{time}x_{P,time} + \beta_{cost}x_{P,cost} + \beta_{P,age}x_{age} + \beta_P \quad (2.6)$$

### 2.1.1 より複雑な立式

- 「免許を持っていない人は絶対に車で移動しない」といったことを表現するには、 $V_{car} = -\infty$  とすればよいです。こうすることで選択確率  $\frac{\exp(V_{car})}{\sum \exp V}$  の分子が強制的に0になり、選択確率が0すなわち絶対に選択されない状態になります。
  - 説明変数の log などをとってから重みを掛けるという方法もあり得ます。「所要時間が20分から15分になった」場合と「所要時間が120分から115分になった」場合で、どちらも同じだけ効用が増加したと考えるよりは、20分→15分の方が効用の増加幅が大きいとする方が自然でしょう。log を取るなどすることでこのような非線形な効用の増減も表現できます。
- ただし、このような調整を恣意的に行うことは Data dredging や p-hacking 等と呼ばれ、研究不正にあたります。行動メカニズムとして説明できないような効用式を立ててはいけません。

## 第3章

# パラメータ推定

2章で立式した効用の値を求めるには、未知のパラメータである  $\beta_{time}, \beta_{cost}, \beta_{S,age}, \dots$  を求めなければいけません。これらの未知パラメータを、データを使って推定することを**パラメータ推定**といいます。

パラメータ推定では、パラメータの**尤度**（もっともらしさ）が最大になるようにパラメータを決定します。離散選択モデルの場合、**モデルから計算される各選択肢の選択確率が、実際の各選択肢の選択確率にできるだけ近くなるようにすればよいです。**

### 3.1 パラメータ推定に用いるデータ

実際の各選択肢の選択確率は、交通分野の場合、以下のようなデータを組み合わせて用いることで得ることができます。

**LOS データ** 交通サービス水準。所要時間や運賃など交通手段に固有の特性を表すデータ。

**RP 調査によるデータ** 人々がある地点から別の地点へ実際に移動する際に、いつ・どこへ・どんな交通手段で・何のために移動したのかの情報を収集したデータ。個人の属性の情報もセットで記録する。

**PT（パーソントリップ）データ** RP 調査のうち、平均的な一日の行動についてアンケートによって収集したデータ。

**PP（プローブパーソン）データ** RP 調査のうち、GPS 等を用いて実際の行動記録を収集して整理したデータ。

**SP 調査によるアンケートデータ** 様々な条件を仮定したときにどんな移動をすることを選ぶかの希望を、アンケートによって収集したデータ。実際の行動に基づかないためやや信頼性が落ちる代わりに、未来の交通手段などを想定したアンケートをすることができる。個人の属性の情報もセットで記録する。LOS に相当する、どの選択肢ならどんな条件で使えるかという情報がアンケート内に含まれている。

これらのデータから、交通手段に固有の特性や個人の属性といった説明変数になりえる値と、実際にどの選択肢が選ばれたかという結果がわかります。パラメータ推定によってモデルを実際の選択結果に近づけ、できたモデルをもとに新たな条件ではどの選択肢がどれだけの確率で選択できるかを推定する、というのがモデルの活用の流れです。

## 3.2 尤度

MNL における尤度の求め方を示します。

$M$  人が  $N$  個の選択肢の中から独立に選択行動をします。 $i$  番目の人にとっての説明変数が  $\mathbf{x}_i$  である<sup>\*1</sup>とき、選択肢  $y_i$  が選ばれる確率を  $P(y_i; \mathbf{x}_i)$  と書くことにします。

このとき  $1, 2, \dots, M$  番目の人がそれぞれ  $y_1, y_2, \dots, y_M (1 \leq y_i \leq N)$  の選択肢を選択する同時確率は、各人の選択行動が独立であることから、式 (3.1) のように表されます。

$$P(y_1; \mathbf{x}_1, y_2; \mathbf{x}_2, \dots, y_M; \mathbf{x}_M) = P(y_1; \mathbf{x}_1)P(y_2; \mathbf{x}_2) \cdots P(y_M; \mathbf{x}_M) = \prod_{i=1}^M P(y_i; \mathbf{x}_i) \quad (3.1)$$

なお、MNL では式 (3.1) の値は次の式 (3.2) のように書けます。

$$\prod_{i=1}^M \frac{\exp(V_{y_i})}{\sum_{j=1}^N \exp(V_j)} \quad (3.2)$$

効用の確定項  $V$  は、2 章で立式したように、説明変数  $\mathbf{x}$  と未知パラメータ  $\beta$  <sup>\*2</sup>によって決まる値でした。

ところで、 $i$  番目の人が  $y_i$  を選ぶという事象はすでに確定していますし、各人・各選択肢の説明変数も分かっています。分かっている値は未知パラメータです。そこで**尤度関数**  $L(\beta)$  を、式の形は変えずに次のように定義します。

$$L(\beta) := \prod_{i=1}^M P(y_i; \mathbf{x}_i) = \prod_{i=1}^M \frac{\exp(V_{y_i})}{\sum_{j=1}^N \exp(V_j)} \quad (3.3)$$

すると、この尤度関数を最大化することは、 $\beta$  を変化させることによって  $y_1, y_2, \dots, y_M$  の選択肢が選択される同時確率を最大化することにほかなりません。したがって、尤度関数を最大化するような  $\beta$  と、 $\beta$  によって計算される効用の確定項  $V$  は、実際の現象をもっともうまく表現しているといえることができます。

このように、尤度を最大化するようなパラメータを真のパラメータであるとする手法のことを**最尤推定法**といいます。

ある条件においてどの選択肢がどのくらいの確率で選ばれるのかを推定する、という問題は、最終的に関数を最大化する問題に帰着されました。関数の最大化問題は、古くから数理最適化の分野で研究が進められ、強力なソルバーが多数提供されています。よって、この関数の最大化問題には深く立ち入らず、実装上は R や Python の SciPy で提供されている**最適化ライブラリ**を使用して解きます。

ライブラリに投げる前に、2 点注意すべきことがあります。1 つは尤度を直接最大化する代わりに次項で述べる**対数尤度**を最大化すること、もう 1 つは最尤推定法によって未知パラメータが一意に定まるように効用を立式することです。

<sup>\*1</sup> 説明変数は複数個ありうるのでベクトルで表記します。ここでは、交通手段の特性由来の説明変数も、個人の属性由来の説明変数も一つにまとめています。

<sup>\*2</sup> 未知パラメータも複数あるのでベクトルで表記します。

### 3.2.1 対数尤度

対数尤度とは、尤度の対数を取った値です。尤度  $L(\beta) = \prod_{i=1}^M P(y_i; \mathbf{x}_i)$  は、1 より小さい値を大量に掛け合わせて得られるため、非常に小さな値をとります。このため、プログラムで尤度を直接求めると、誤差によって 0 に丸められてしまう恐れがあります。

そこで、尤度を最大化する代わりに尤度の対数を取った値を最大化することを考えます。対数尤度  $LL(\beta)$  は次の式 (3.4) で定義されます。

$$LL(\beta) := \log L(\beta) = \sum_{i=1}^M \log P(y_i; \mathbf{x}_i) \quad (3.4)$$

対数尤度の値は、選択確率が非常に小さくなるような選択肢が選ばれていない限りは問題なく計算できます。

### 3.2.2 未知パラメータが一意に定まるような効用式

尤度が最大となるようなパラメータの組が無数通りある、という状況が生じてしまうと、尤度関数の最大化が収束しなくなりパラメータ推定に失敗してしまいます。

例えば、効用の確定項を下の式 (3.5) のように立式すると、パラメータ推定に失敗します。

$$\begin{aligned} V_S &= \beta_{time} x_{S,time} + \beta_{cost} x_{S,cost} + \beta_{S,age} x_{age} + \beta_S \\ V_B &= \beta_{time} x_{B,time} + \beta_{cost} x_{B,cost} + \beta_{B,age} x_{age} + \beta_B \\ V_P &= \beta_{time} x_{P,time} + \beta_{cost} x_{P,cost} + \beta_{P,age} x_{age} + \beta_P \end{aligned} \quad (3.5)$$

「説明変数を掛けないパラメータ」を全ての選択肢につけてはいけません。次の式 (3.6) のように、「説明変数を掛けないパラメータ」の数は、選択肢の数よりも 1 少なくなるようにする必要があります。

$$\begin{aligned} V_S &= \beta_{time} x_{S,time} + \beta_{cost} x_{S,cost} + \beta_{S,age} x_{age} \\ V_B &= \beta_{time} x_{B,time} + \beta_{cost} x_{B,cost} + \beta_{B,age} x_{age} + \beta_B \\ V_P &= \beta_{time} x_{P,time} + \beta_{cost} x_{P,cost} + \beta_{P,age} x_{age} + \beta_P \end{aligned} \quad (3.6)$$

選択確率の式 (1.6) を見ればわかるように、選択確率は  $V$  の値そのものではなく、 $\exp(V)$  の値の比によって決まります。「説明変数を掛けないパラメータ」をすべての選択肢の効用式につけると、各選択肢の選択確率が同じになるような  $(\beta_S, \beta_B, \beta_P)$  の値の組が無数通り存在することになってしまいます。説明変数を掛けないパラメータを一つ減らすことで自由度が下がり、この問題が解消されます。

## 第4章

# 検定

3章までで、多項ロジットモデルを作って各選択肢の選択確率を求めることができました。

しかし、最尤推定法によって尤もらしいパラメータを推定することができたからといって、そのパラメータが統計的に信頼できるとは限りません。また、そのパラメータと説明変数を持つモデルが実際のデータにどのくらいうまく適合しているかも知る必要があります。

例えば、もし所要時間の重みパラメータ  $\beta_{time}$  の推定値が統計的に信頼できないとみなすならば、そのモデルを使って所要時間の変化に伴う乗客数の変化を推定することはできません。また、新しい説明変数とパラメータを追加することでモデルの適合度が有意に向上するのであれば、新しいモデルの方を採用すべきでしょう。

求めたパラメータが統計的に信頼できるかどうかを判定するための方法の一つとして、**t 検定**がしばしば用いられます。また、モデルと実際のデータの適合度合いを見るために**尤度比検定**が用いられます。

### 4.1 t 検定

ここでは、紛らわしい統計用語を順を追って整理し、最終的に t 検定について説明します。

#### 4.1.1 統計量

統計的分析の対象となる人や物事の集まりを**母集団**、母集団から抽出してきた人や物事の集まりを**標本**といいます。標本に対して、標本に関する何らかの値（選択行動の結果など）を取り出してくると、その値は標本の選び方によって変わる**確率変数**になります。このようにして得られる確率変数の集まりのことを**標本変数**といいます。

**統計量**とは、標本変数に対して何らかの操作を行うことによって得ることができる、その標本変数の特徴を簡潔に示す量のことです。例えば算術平均は「標本変数の総和を標本数で割る」という操作によって得ることができる統計量です。また分散、中央値、最頻値なども同様に統計量です。

重要な事実として、標本変数は確率変数の集まりなので、それに対して操作を加えて得られる**統計量もまた確率変数**です。

#### 4.1.2 推定値と推定量

3.2 節では最尤推定法によって未知パラメータ  $\beta$  の値を推定しました。このときは実際に観測されたデータを使って推定を行ったので  $\beta$  の具体的な値を求めることができました。これを  $\beta$  の**推定値**と呼び



ます。特にこの場合は最尤推定法によって得られた値なので、**最尤推定値**と呼びます。しかし、最尤推定値の値は標本の選び方によって変わるはずです。

最尤推定の結果は、標本変量に対して尤度を最大化するという操作を行うことで得られる量なので、統計量の定義を満たしています。このように、何らかの推定のために得る統計量のことを**推定量**と呼び、 $\hat{\beta}$ と記します。特にこの場合は最尤推定法によって得られた推定量なので、**最尤推定量**と呼びます。最尤推定量は統計量なので確率変数であり、したがって確率分布を持ちます。

t 検定では、「各パラメータの最尤推定量がとる値が 0 になる確率」を計算します。 $i$  番目のパラメータ  $\beta_i$  の真の値が 0 であるとき、そのパラメータに対応する説明変数  $x_i$  に何を代入しても効用の値が変化しないので、その説明変数は選択行動を説明するのに全く役立たないということになります。逆にこのようなことになる確率が十分に低ければ、そのパラメータに対応する説明変数  $x_i$  は選択行動を説明しているということが出来ます。

なお、最尤推定量は以下の性質を満たします。

**一致性** 無限個の標本を用いて推定した最尤推定値は、パラメータの真値に一致する。

**漸近正規性** 標本が多く<sup>\*1</sup>なると、最尤推定量の分布は正規分布に近づく。

**漸近有効性** 標本が多くなると、最尤推定量の分散は**クラメール・ラオの不等式**の下限に近づく。

### 推定量の有効性

推定量の期待値がパラメータの真値に一致するとき、その推定量は**不偏推定量**であるといいます（一致性とは異なる概念です）。さらに、その不偏推定量が他のどの不偏推定量よりも分散が大きいとき、その推定量は**有効推定量**であるといいます。最尤推定量は不偏推定量ではないので、当然有効推定量でもありませんが、標本が多くなると漸近的に有効推定量のようなふるまいを見せる、というのが漸近有効性です。

有効推定量が定義可能であることから分かるように、不偏推定量の分散には下限が存在します。この不偏推定量の分散の下限を示すのが**クラメール・ラオの不等式**です。

**定理 3** (クラメール・ラオの不等式)。

$$\text{cov}(\hat{\beta}) \geq -E \left[ \nabla \nabla^\top LL(\hat{\beta}) \right]^{-1} \quad (4.1)$$

ただし  $\hat{\beta}$  は不偏推定量とし、 $\text{cov}(\hat{\beta})$  は  $\hat{\beta}$  の分散共分散行列、 $LL$  は対数尤度関数です。また、 $\nabla \nabla^\top LL(\hat{\beta})$  は書き出すと下の式 (4.2) のようになり、特に行列部分には**ヘッセ行列**という名前がつけられています。

$$\nabla \nabla^\top LL(\hat{\beta}) = \begin{bmatrix} \frac{\partial^2 LL}{\partial \beta_1^2} & \frac{\partial^2 LL}{\partial \beta_1 \partial \beta_2} & \cdots & \frac{\partial^2 LL}{\partial \beta_1 \partial \beta_n} \\ \frac{\partial^2 LL}{\partial \beta_2 \partial \beta_1} & \frac{\partial^2 LL}{\partial \beta_2^2} & \cdots & \frac{\partial^2 LL}{\partial \beta_2 \partial \beta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 LL}{\partial \beta_n \partial \beta_1} & \frac{\partial^2 LL}{\partial \beta_n \partial \beta_2} & \cdots & \frac{\partial^2 LL}{\partial \beta_n^2} \end{bmatrix} \hat{\beta} \quad (4.2)$$

等号が成立する時、 $\hat{\beta}$  は分散の下限を達成しているので、 $\hat{\beta}$  は有効推定量となります。

最尤推定量の場合は、漸近有効性により、標本が十分多ければ等号成立、すなわち

<sup>\*1</sup> 50 件とか 100 件以上などと言われています。

$$\text{cov}(\hat{\beta}) = -E \left[ \nabla \nabla^\top LL(\hat{\beta}) \right]^{-1} \quad (4.3)$$

としてよいです。

### 4.1.3 区間推定

t 検定では、各パラメータについて、次のような帰無仮説を立てます。

t 検定での帰無仮説

パラメータ  $\beta_i$  の最尤推定量  $\hat{\beta}_i$  がとる値は 0 である。

この帰無仮説が一定の**有意水準**で**棄却**されるならば、パラメータ  $\beta_i$  の最尤推定量が 0 とは有意に異なるとみなすことができます。

帰無仮説が棄却されるかどうかを調べるためには、 $\beta_i$  の最尤推定量がとりうる区間を推定し、0 が**信頼区間**内に入っていないことを確認すればよいです。

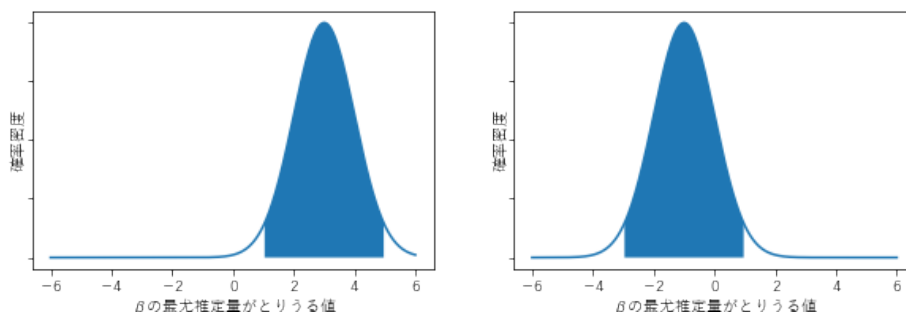


図 4.1 信頼区間

図 4.1 で青く塗っている区間が信頼区間です。例えば「95% 信頼区間」であれば、パラメータ  $\beta_i$  の最尤推定量  $\hat{\beta}_i$  がとる値が 95% の確率でその区間内に収まることを意味します。

左の図では、95% 信頼区間内に 0 が含まれていないので、 $\hat{\beta}_i$  が取る値が 0 になる確率は 5% 以下となります。すると「パラメータ  $\beta_i$  の最尤推定量  $\hat{\beta}_i$  がとる値は 0 である」という帰無仮説は 5% 有意水準で棄却されることになり、そのパラメータに対応する説明変数は選択行動を説明しているといえます。

いっぽう右の図では、95% 信頼区間内に 0 が含まれているので、帰無仮説は棄却されません。すると、そのパラメータに対応する説明変数が選択行動を説明しているかどうかは分かりません（説明変数が「効かない」ことを意味しているわけではなく、「効くとは言い切れない」ということを意味します）。

ここで、信頼区間の幅を求めるには、最尤推定量の分散が必要になります。そこで、先ほどの  $\text{cov}(\hat{\beta}) = -E \left[ \nabla \nabla^\top LL(\hat{\beta}) \right]^{-1}$  を用いると、各パラメータの分散を得ることができます。とくに今回は推定値が一つしかないなので、その推定値（ここでは  $\beta^*$  とおきます）をそのまま代入して  $\text{cov}(\hat{\beta}) = -(\nabla \nabla^\top LL(\beta^*))^{-1}$  としてよいです。L のヘッセ行列は数値微分によって得ることができます。

各パラメータの最尤推定量の分散  $s^2$  が分かれば、最尤推定量の漸近正規性により、正規分布の場合と同様に信頼区間を求めることができます。

各パラメータの最尤推定量を  $t_i := \frac{\hat{\beta}_i}{s}$  と標準化します。このとき出現する値  $t_i$  は **t 値** と呼ばれ、t 値が**標準正規分布**の 95% 信頼区間内に入っていないならば、帰無仮説は 5% 有意水準で棄却されることになります。このような t 値の閾値は  $|t| > 1.96$  です（標準正規分布の両側 5% 点は 1.96）。また、 $|t| > 2.58$  となれば帰無仮説は 1% 有意水準で棄却されることになります。

## 4.2 尤度比指標

尤度比検定は、先ほどの t 値とは異なり、あるモデルが全体として別のモデルより適合度が高いかどうかを見る検定です。

モデル A を、モデル B にいくつかの説明変数とパラメータを加えたモデルとします。モデル A とモデル B の尤度比は、文字通り尤度の比  $\frac{L_A(\hat{\beta}_A)}{L_B(\hat{\beta}_B)}$  です。尤度比の対数は  $LL_A(\hat{\beta}_A) - LL_B(\hat{\beta}_B)$  となり、対数を取った方の値を単に**尤度比**と呼ぶことがあります。尤度比は確率変数なので検定可能です。

尤度比検定では、次のような帰無仮説を立てます。

尤度比検定での帰無仮説

帰無仮説：モデル A とモデル B の尤度比は 0 である。

尤度が大きい方がより「もっともらしい」モデルなので、説明変数が多いモデル A の方が尤度が大きくなってほしいところです。この帰無仮説が棄却されればモデル A の方が有意に尤度が大きいということができ、二つのモデルの適合度の差が有意なものであるということが出来ます。

ところで、標本が十分多いとき、尤度比を 2 倍した値は**カイ二乗分布**と呼ばれる分布に従うことが知られています。特に、 $\beta_A$  のパラメータ数が  $N_A$  個、 $\beta_B$  のパラメータ数が  $N_B$  個であるとき、 $2(LL_A(\hat{\beta}_A) - LL_B(\hat{\beta}_B))$  は**自由度**  $N_A - N_B$  のカイ二乗分布に従います。

したがって、帰無仮説「モデル A とモデル B の尤度比は 0 である」を棄却するには、 $2(LL_A(\hat{\beta}_A) - LL_B(\hat{\beta}_B))$  の信頼区間内に 0 が入ってこないことを示せばよいです。

たとえば  $N_A = 7, N_B = 5$  のとき、自由度 2 のカイ二乗分布の上側 5% 点は 5.99 なので、 $2(LL_A(\hat{\beta}_A) - LL_B(\hat{\beta}_B)) > 5.99$  ならば 5% 有意水準で帰無仮説が棄却されることになります。

特に、すべての選択枝の効用を 0 としたモデル（説明変数もパラメータもないモデル）をモデル B として尤度比検定することで、作ったモデルの適合度が有意であることを確認することができます。すべての選択枝の効用を 0 としたモデルのことを **Null model** と呼び、その対数尤度は  $LL(0)$  で求めることができます。

注意すべき点として、モデル A はモデル B を含んでいる、上位互換のモデルである必要があります。厳密に述べると、モデル A に含まれない説明変数・パラメータであって、モデル B に含まれるような説明変数・パラメータがあってははいけません。

## 第5章

# Python による MNL の実装

ここでは、[1] に付録されている MNL の R による実装について、Python で書き換えたものを示します。

### 5.1 ファイルの読み込み等

```
1 import numpy as np
2 import pandas as pd
3 from scipy.optimize import minimize
4 df = pd.read_csv('ensyu.csv', encoding='shift-jis')
```

パラメータ推定に用いる PT データは、たいていの場合 CSV で提供され、一行に一人分のデータが入っている、というようなフォーマットになっています。Python でこのようなデータを扱うには Pandas が便利です。

### 5.2 効用の計算

```
1 def fr(x: np.array) -> float:
2     b1, b2, b3, b4, d1, f1 = x # b1~b4: 定数項, d1: 所要時間, f1: 運賃
3
4     train: pd.Series = df["代替手段生成可否train"] * np.exp(
5     d1 * df["総所要時間train"] / 100 + f1 * df["費用train"] / 100 + b1
6     )
7     bus: pd.Series = df["代替手段生成可否bus"] * np.exp(
8     d1 * df["総所要時間bus"] / 100 + f1 * df["費用bus"] / 100 + b2
9     )
10    car: pd.Series = df["代替手段生成可否car"] * np.exp(
11    d1 * df["総所要時間car"] / 100 + b3
12    )
13    bike: pd.Series = df["代替手段生成可否bike"] * np.exp(
14    d1 * df["総所要時間bike"] / 100 + b4
15    )
16    walk: pd.Series = df["代替手段生成可否walk"] * np.exp(
17    d1 * df["総所要時間walk"] / 100
18    )
19
20    # 続きます
```

「電車」「バス」「車」「自転車」「徒歩」から選択する手段選択モデルをつくります。効用関数は式 (5.1) の通りとしています。

$$\begin{aligned}
 V_{train} &= \frac{\beta_{d1}x_{train,time}}{100} + \frac{\beta_{f1}x_{train,cost}}{100} + \beta_1 \\
 V_{bus} &= \frac{\beta_{d1}x_{bus,time}}{100} + \frac{\beta_{f1}x_{bus,cost}}{100} + \beta_2 \\
 V_{car} &= \frac{\beta_{d1}x_{car,time}}{100} + \beta_3 \\
 V_{bike} &= \frac{\beta_{d1}x_{bike,time}}{100} + \beta_4 \\
 V_{walk} &= \frac{\beta_{d1}x_{walk,time}}{100}
 \end{aligned} \tag{5.1}$$

$\beta_{d1}$  が所要時間に関するパラメータ、 $\beta_{f1}$  が運賃に関するパラメータ、それ以外の4つが定数項（説明変数に掛けないパラメータ）です。3.2.2 項で見た通り、説明変数に掛けないパラメータの数は、選択肢の数より1少なくしないといけません。

このコードでは `train bus` などの変数に、効用の確定項そのものではなくその `exp` をとったものを代入しています。

`exp` をとったあとに `df代替手段生成可否["train"]` などを掛けていますが、この列は「その選択肢が選択可能かどうか」を表しています（選択可能なら 1, 不可なら 0）。\*\*[3 章 効用の立式]\*\* の最後で少し触れていますが、選択確率を 0 にしたいという場合には  $V = -\infty$  とすればよく、このコードではその代わりに  $\exp(V) = 0$  としています。

なお、これらの変数は `pd.Series` という型になっています。これらはそれぞれ人数分の長さを持つ一次元配列のようなものだと考えてください。

### 5.3 選択確率の計算

```

1 def fr(x: np.array) -> float:
2     # 前略()
3     # 続き
4
5     deno: pd.Series = car + train + bus + bike + walk
6
7    Ptrain: pd.Series = df["代替手段生成可否train"] * (train / deno)
8     Pbus: pd.Series = df["代替手段生成可否bus"] * (bus / deno)
9     Pcar: pd.Series = df["代替手段生成可否car"] * (car / deno)
10    Pbike: pd.Series = df["代替手段生成可否bike"] * (bike / deno)
11    Pwalk: pd.Series = df["代替手段生成可否walk"] * (walk / deno)
12
13    # 続きます

```

`deno` には選択確率の分母となる  $\sum_{i=1}^N \exp(V_i)$  の値を代入し、それを使って各選択肢の選択確率  $P_j = \frac{\exp(V_j)}{\sum_{i=1}^N \exp(V_i)}$  を計算しています。これで、各参加者がそれぞれどれだけの選択確率で選択肢を選ぶかが分かります。

### 5.4 対数尤度の計算

```

1 def fr(x: np.array) -> float:
2     # 前略()
3     # 続き
4
5    Ptrain = Ptrain.where(Ptrain != 0, 1)
6     Pbus = Pbus.where(Pbus != 0, 1)
7     Pcar = Pcar.where(Pcar != 0, 1)
8     Pbike = Pbike.where(Pbike != 0, 1)
9     Pwalk = Pwalk.where(Pwalk != 0, 1)
10
11     Ctrain: pd.Series = df["代表交通手段"] == "鉄道"
12     Cbus: pd.Series = df["代表交通手段"] == "バス"
13     Ccar: pd.Series = df["代表交通手段"] == "自動車"
14     Cbike: pd.Series = df["代表交通手段"] == "自転車"
15     Cwalk: pd.Series = df["代表交通手段"] == "徒歩"
16
17     LL: float = np.sum(
18         Ctrain * np.log(Ptrain)
19         + Cbus * np.log(Pbus)
20         + Ccar * np.log(Pcar)
21         + Cbike * np.log(Pbike)
22         + Cwalk * np.log(Pwalk)
23     )
24     return LL

```

選択肢を計算した後に、`Ptrain = Ptrain.where(Ptrain != 0, 1)` のような処理が入っています。これは「選択確率が 0 のときに 1 に直す」という処理で、単に選択確率を出すだけならいらない（というよりやってはいけない）処理です。しかし、最後に対数尤度を計算する際に選択確率の  $\log$  を取る必要があるため、選択確率に 0 が含まれていると困ります。そこで  $P = 1$  にすることで、 $\log(1) = 0$  を利用して  $LL$  の値に影響を与えないようにすることができます。

その次の `Ctrain` などの変数には、「選択されたのが鉄道なら 1、そうでないなら 0」のような値が入ります。もちろんこれらも人によって選択結果が違うので、人数分の長さの配列になっています。

最後の  $LL$  の式が対数尤度を求める式ですが、3.2.1 項で出た式とは少し違う形になっています。3.2.1 項の式は、 $i$  番目の人が選択した選択肢が  $y_i$  とすると、

$$LL(\beta) = \sum_{i=1}^M \log P(y_i; \mathbf{x}_i) \quad (5.2)$$

という式でした。しかしここで実装しているのは、あえて数式にするならば、

$$LL(\beta) = \sum_{i=1}^M \sum_{j=1}^N C_{i,j} \log P(j, \mathbf{x}_i) \quad (5.3)$$

(ただし  $C_{i,j}$  は  $i$  番目の人が  $j$  番目の選択肢を選択したなら 1、そうでないなら 0) のような式です。しかしよく見ると、両者は同値であることが分かります。

## 5.5 最尤推定とその評価

ここから先のコードは Python ならではの問題があり、R と同じように簡潔に書くことができないため、R による実装は無視して書いています。

```
1 def mf(x: np.array) -> float:
2     return -fr(x)
3
4 x0 = np.zeros(6)
5 res = minimize(mf, x0, method="Nelder-Mead")
6 print(res)
```

`minimize` が関数最小化のライブラリです。

`minimize` 関数は最小化したい関数、初期値、使うアルゴリズムを引数にとるのですが、ここで渡す関数が先ほど実装した `fr` ではなく `mf` であることに注意してください。やりたいのは `fr` つまり対数尤度  $LL$  の\*\*最大化\*\*ですが、R と違って Python の `scipy.optimize` には関数最大化のライブラリがないので、`fr(x)` の符号を逆にしたもの すなわち  $-LL$  を返すような関数 `mf` を新たに定義し、これを\*\*最小化\*\*するというアプローチをとります。

なお、`scipy.optimize.minimize` で指定する `method` はここでは Nelder-Mead を用いていますが、他の `method` を指定してもよいです。`method = 'SLSQP'` が高速かつより発展的なモデルのための制約もつけやすく、良いようです。`scipy.optimize.minimize` の公式マニュアルを見ると、使用できる `method` の一覧があります。

ここまでのコードを実行するとこんな感じの結果が返ってきます。

```
1 final_simplex: (array([[ 1.72219562,  1.12706031,  0.59742075, -0.12335375,
2     -2.35299203,
3     -0.14858004],
4 [ 1.72223374,  1.12713838,  0.59742214, -0.12333357, -2.3530793 ,
5     -0.14858499],
6 [ 1.72227063,  1.1271141 ,  0.59740231, -0.12335428, -2.35305542,
7     -0.14859453],
8 [ 1.72225107,  1.12711746,  0.5974423 , -0.1233492 , -2.35308902,
9     -0.14858328],
10 [ 1.72224146,  1.12709266,  0.59741655, -0.12335483, -2.35303904,
11     -0.14858497],
12 [ 1.72219851,  1.12702764,  0.59740098, -0.12336702, -2.35294512,
13     -0.14858476],
14 [ 1.72224552,  1.12706853,  0.59744533, -0.12336778, -2.35304103,
15     -0.14858999])), array([437.76456954, 437.76456967, 437.76456967, 437.76456968,
16 437.76456971, 437.76456975, 437.76456976]))
17 fun: 437.7645695352565
18 message: 'Optimization terminated successfully.'
19 nfev: 666
20 nit: 420
21 status: 0
22 success: True
23 x: array([ 1.72219562,  1.12706031,  0.59742075, -0.12335375, -2.35299203,
-0.14858004])
```

見るべきは `fun` と `x` です。`fun` は `mf` のとりうる最小値、`x` はそのときのパラメータです。つまり  $-\text{fun}$

が  $LL$  の最大値となります。これを使えば 4.2 節の尤度比検定\*\*をすることができます。

最後に  $t$  値を求めてみます。R であればヘッセ行列を求める関数がデフォルトで使えるのですが、Python には調べた限りなかったので、自分で数値微分を実装してヘッセ行列を求める必要があります。

コードを示すと次のようにできます。

```

1 def hessian(x: np.array) -> np.array:
2     h = 10 ** -4 # 数値微分用の微小量
3     n = len(x)
4     res = np.zeros((n, n))
5     for i in range(n):
6         for j in range(n):
7             e_i, e_j = np.zeros(n), np.zeros(n)
8             e_i[i] = 1
9             e_j[j] = 1
10
11             res[i][j] = (fr(x + h * e_i + h * e_j)
12                          - fr(x + h * e_i - h * e_j)
13                          - fr(x - h * e_i + h * e_j)
14                          + fr(x - h * e_i - h * e_j)) / (4 * h * h)
15     return res
16
17
18 def tval(x: np.array) -> np.array:
19     return x / np.sqrt(-np.diag(np.linalg.inv(hessian(x))))
20
21 print(f"tval: {tval(res.x)}")

```

まず 1 階微分の\*\*中心差分近似\*\*を考えると、微小な値  $h$  を用いて、

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \frac{f(x_1, x_2, \dots, x_i + h, \dots, x_n) - f(x_1, x_2, \dots, x_i - h, \dots, x_n)}{2h} \quad (5.4)$$

です。 $i$  番目の要素だけが 1 でそれ以外が 0 の単位ベクトル  $\mathbf{e}_i = [0, 0, \dots, 0, 1, 0, \dots, 0]$  を用いると、

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h} \quad (5.5)$$

と書けます。

ヘッセ行列の各要素はこれをもう一度微分すると得られます。

$$\begin{aligned}
 \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} &= \frac{\frac{\partial f(\mathbf{x} + h\mathbf{e}_j)}{\partial x_i} - \frac{\partial f(\mathbf{x} - h\mathbf{e}_j)}{\partial x_i}}{2h} \\
 &= \frac{\frac{f(\mathbf{x} + h\mathbf{e}_i + h\mathbf{e}_j) - f(\mathbf{x} - h\mathbf{e}_i + h\mathbf{e}_j)}{2h} - \frac{f(\mathbf{x} + h\mathbf{e}_i - h\mathbf{e}_j) - f(\mathbf{x} - h\mathbf{e}_i - h\mathbf{e}_j)}{2h}}{2h} \\
 &= \frac{f(\mathbf{x} + h\mathbf{e}_i + h\mathbf{e}_j) - f(\mathbf{x} - h\mathbf{e}_i + h\mathbf{e}_j) - f(\mathbf{x} + h\mathbf{e}_i - h\mathbf{e}_j) + f(\mathbf{x} - h\mathbf{e}_i - h\mathbf{e}_j)}{4h^2}
 \end{aligned} \quad (5.6)$$

ヘッセ行列が求まれば、あとは 4.1 節で見た通りに実装できます。

最尤推定値にヘッセ行列をかけて、逆行列をとって  $-1$  倍すれば、最尤推定量の分散共分散行列が得られます。`np.diag` で行列の対角成分（つまり  $n$  個のパラメータの分散）が得られるので、そのルートを



取った値は各パラメータの最尤推定量の標準偏差となり、最尤推定値を標準誤差で割れば  $t$  値が得られます。

実行してみると  $t$  値は下のようになります。

```
1 [ 6.31137194  3.71837913  2.59395594 -0.5058321 -6.08917584 -3.02836851]
```

これで、4 番目のパラメータ以外は 1% 有意であることがわかりました。

たとえば、5 番目のパラメータは  $d1$  すなわち所要時間のパラメータだったので、このモデルにおいて所要時間は効用に 1% 有意で影響しているといえます。また、そのパラメータの最尤推定値は  $-2.35299203$  で負の値なので、時間がかかるほど効用が負になるということを意味しています。これがもし正の値だったら何かおかしいと考えた方が良いでしょう。

## 5.6 コード全体

```
1 import numpy as np
2 import pandas as pd
3 from scipy.optimize import minimize
4
5 df = pd.read_csv(r'..\ensyu_mnl\ensyu.csv', encoding='shift-jis')
6
7
8 def fr(x: np.array) -> float:
9     b1, b2, b3, b4, d1, f1 = x # b1~b4: 定数項, d1: 所要時間, f1: 運賃
10
11     train: pd.Series = df["代替手段生成可否train"] * np.exp(
12         d1 * df["総所要時間train"] / 100 + f1 * df["費用train"] / 100 + b1
13     )
14     bus: pd.Series = df["代替手段生成可否bus"] * np.exp(
15         d1 * df["総所要時間bus"] / 100 + f1 * df["費用bus"] / 100 + b2
16     )
17     car: pd.Series = df["代替手段生成可否car"] * np.exp(
18         d1 * df["所要時間car"] / 100 + b3
19     )
20     bike: pd.Series = df["代替手段生成可否bike"] * np.exp(
21         d1 * df["所要時間bike"] / 100 + b4
22     )
23     walk: pd.Series = df["代替手段生成可否walk"] * np.exp(
24         d1 * df["所要時間walk"] / 100
25     )
26
27     deno: pd.Series = car + train + bus + bike + walk
28
29     Ptrain: pd.Series = df["代替手段生成可否train"] * (train / deno)
30     Pbus: pd.Series = df["代替手段生成可否bus"] * (bus / deno)
31     Pcar: pd.Series = df["代替手段生成可否car"] * (car / deno)
32     Pbike: pd.Series = df["代替手段生成可否bike"] * (bike / deno)
33     Pwalk: pd.Series = df["代替手段生成可否walk"] * (walk / deno)
34
35     Ptrain = Ptrain.where(Ptrain != 0, 1)
36     Pbus = Pbus.where(Pbus != 0, 1)
37     Pcar = Pcar.where(Pcar != 0, 1)
38     Pbike = Pbike.where(Pbike != 0, 1)
```

```

39     Pwalk = Pwalk.where(Pwalk != 0, 1)
40
41     Ctrain: pd.Series = df["代表交通手段"] == "鉄道"
42     Cbus: pd.Series = df["代表交通手段"] == "バス"
43     Ccar: pd.Series = df["代表交通手段"] == "自動車"
44     Cbike: pd.Series = df["代表交通手段"] == "自転車"
45     Cwalk: pd.Series = df["代表交通手段"] == "徒歩"
46
47     LL: float = np.sum(
48         Ctrain * np.log(Ptrain)
49         + Cbus * np.log(Pbus)
50         + Ccar * np.log(Pcar)
51         + Cbike * np.log(Pbike)
52         + Cwalk * np.log(Pwalk)
53     )
54     return LL
55
56
57 def mf(x: np.array) -> float:
58     return -fr(x)
59
60
61 def hessian(x: np.array) -> np.array:
62     h = 10 ** -4 # 数値微分用の微小量
63     n = len(x)
64     res = np.zeros((n, n))
65     for i in range(n):
66         for j in range(n):
67             e_i, e_j = np.zeros(n), np.zeros(n)
68             e_i[i] = 1
69             e_j[j] = 1
70
71             res[i][j] = (fr(x + h * e_i + h * e_j)
72                 - fr(x + h * e_i - h * e_j)
73                 - fr(x - h * e_i + h * e_j)
74                 + fr(x - h * e_i - h * e_j)) / (4 * h * h)
75     return res
76
77
78 def tval(x: np.array) -> np.array:
79     return x / np.sqrt(-np.diag(np.linalg.inv(hessian(x))))
80
81
82 x0 = np.zeros(6)
83 res = minimize(mf, x0, method="Nelder-Mead")
84 print(res)
85
86 print(f"tval: {tval(res.x)}")
87 L0 = fr(x0)
88 print(f"L0: {L0:.4f}")
89 LL = -res.fun
90 print(f"LL: {LL:.4f}")

```

## 参考文献

- [1] BinN studies シリーズネットワーク行動学 -都市と移動-. 最終更新日 2014 年 9 月 26 日.  
<http://bin.t.u-tokyo.ac.jp/kaken/>