

〈研究ノート〉

動的離散選択モデルの構造推定（サーベイ論文）

— シングルエージェントの意思決定問題 —

楠田 康之*

要 旨

本稿は、シングルエージェントの意思決定問題に関する動的離散モデルの構造推定について、最近の研究を概観する。まず、動的離散選択の基本モデルについて説明し、そのモデルにもとづいて推定する手法について議論する。特に、Nested fixed point algorithm (NFXP) と Nested pseudo-likelihood algorithm (NPL) を解説し、それらを比較する。次に、実際にモンテカルロ法によってデータを生成し、そのデータを用いて簡単な構造推定を行なう。さらに、それ以外の推定法を解説し、最近の比較的新しい手法についても触れる。

キーワード：動的離散選択モデル、シングルエージェントの意思決定問題、推定アルゴリズム、モンテカルロ・シミュレーション

1 はじめに

本稿は、シングルエージェントの意思決定問題に関する動的離散モデルの構造推定について、最近の研究を概観する。近年、多くの経済学の分野において、構造推定が注目を集めている。経済分野における構造推定とは、経済理論を土台とした実証分析であり、個人の選好や信念、企業の技術や費用といった、“構造”を決定する構成要素を推定するものである。構造推定の分析では、分析の対象となる意思決定主体がわれわれが想定する経済理論モデル（構造モデル）の中で最適化行動をとっており、観察されたデータがそのような行動や均衡解を反映しているということが前提となる。したがって、推定を行なう場合、なんらかの構造モデルを適切に想定し、そのモデルを主体になりかわって解くことが必要となる。この節では、構造推定の特徴と利点、構造モデルの分類、構造推定を行なうときの問題点についてまとめ、その後で本稿の目的と本稿が扱う対象について述べる。

* 日本福祉大学経済学部 E-mail: kusuda@n-fukushi.ac.jp

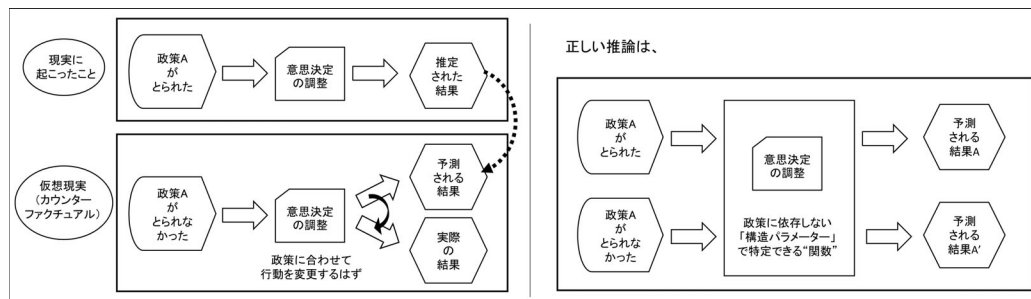


図1 構造推定の考え方

1.1 構造推定とは

なぜ構造推定を行なうのか？

構造推定の大きな特徴は、対象となる意思決定主体が構造モデルの中でどのような意思決定を行っているのか考察し、行動や選択に関するメカニズムを解明することである。この特徴により、もしそのような意思決定のメカニズムを解明できれば、実際に起きなかった状態やとられなかった政策にもとづいて意思決定主体がどのような行動をとっていたかを推測することができる。図1に、構造推定の考え方を示した。経済学のように過去の現実を再現して実験することができない場合、われわれは「現実に関わったこと」にもとづいて因果関係を推測する他ない。例えば、ある政策Aによって、人々の行動に関してどのような結果が起きるか推測したいとしよう。単純な推定では、図1（左）のように、「政策Aがとられた」という結果からなんらかの関係を推定する。では、その推定から、もし政策Aがとられていなければ起きていたであろう結果を予測することは可能であろうか？ 実際には、人々が選択する行動は彼らの意思決定によるので、政策Aがとられていなければ、それに合わせて人々は行動を変更しているはずであろう。したがって、政策Aがとられたという現実にもとづいた予測は間違った推測となるかもしれない。そこで、正しい推論は、図1（右）のように、まず、政策に依存しないような意思決定に関するパラメータを求めておき、それにもとづいた意思決定の結果を推測することである。そのためには、状態や時間によって変わることのない、その意思決定主体の持っている構造的なパラメータ（「構造パラメータ」）を推定する必要がある。つまり、構造推定の目的は、対象となる意思決定主体のインセンティブを正しく理解し、できるだけ正確にそのようなパラメータを推定することである。したがって、構造推定を行なうにあたって、それらが政策、制度、環境の変化に対してどのように反応するのか経済理論モデルによって解明する必要がある。

構造推定の分類

推定の土台となる経済理論モデルによって、構造推定を分類しておこう。まず、意思決定問題において時間の要素を考えるか考えないかによって、「動学モデル」と「静学モデル」に分けられる。本稿で扱うのは動学モデルである。特に、意思決定主体が選択する行動が離散的な場合、

これを動的離散選択モデルと呼ぶ¹。また、もし意思決定主体が行動を選択するときに観察する状態が離散的ならば、動的離散選択モデルは動的計画法（DP）を用いて分析できる。次に、この動学モデルを意思決定主体間の関係によって、(1) シングルエージェントの意思決定問題、(2) 動学ゲームの2つに分けよう²。(1)のシングルエージェントの意思決定問題では、個々の意思決定主体がそれぞれ独立に意思決定を行なう。つまり、個々の意思決定において他の意思決定主体の行動は考慮されないで、モデルの中に戦略的な関係は存在しない。したがって、このようなモデルでは、1人の代表的な「個人」を考え、その意思決定問題を分析すればよい。(2)の動学ゲームでは、モデルの中に意思決定主体間の戦略的な関係が存在する。したがって、ゲーム理論にもとづいて、各意思決定主体（プレイヤー）の他のプレイヤーの戦略に対する長期的な最適反応戦略を考えなければならない。一般的には、そのような戦略は「マルコフ戦略」として考えられる。動学ゲームにおいては、シングルエージェント問題にない問題点がある。まず、ゲームによっては均衡が複数存在する場合があります。また、それぞれのプレイヤーにとって他のプレイヤーの選択は内生的な意味を持つ。そのような意味で、動学ゲームはさらに複雑なものとなる。

構造推定の問題点

構造推定を行なうときのもっとも重要な分析上の問題点は、大量の計算を行わなければならないということである。これは、特に動的離散選択モデルの場合にはより深刻なものとなる。なぜなら、動的離散選択モデルを分析するには動的計画法の解を求めなければならないが、一般にその解はコンピューターによる反復計算により求められ、また、パラメーターの推定もしばしば反復計算が必要となるので、二重の意味で多くの反復計算を行わなければならないからである。図2に構造推定のイメージを示した。まず、ある固定された構造パラメーターの下で、経済理論モ

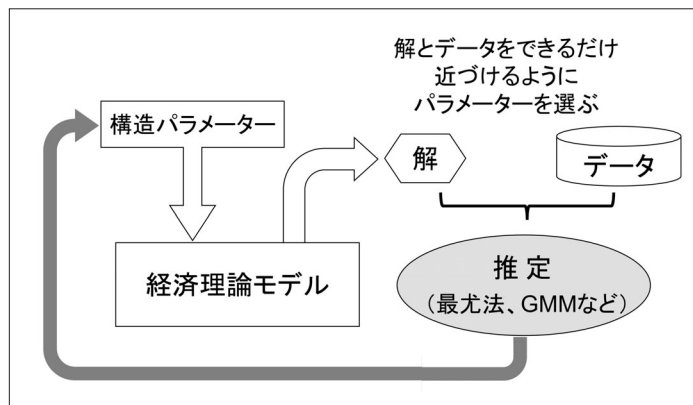


図2 構造推定のイメージ

1 「DDCM (Dynamic Discrete Chice Model)」と略して呼ばれることもある。
 2 この他に競争均衡モデルを考えることもある。

デルから数値計算により解が求められる。もし、その構造パラメーターが正しい値であれば、その解は観察されたデータに近い結果となっているはずである。しかし、その解がデータとかなり違うものであれば、あらためてその解とデータができるだけ近くなるような新しい構造パラメーターを求め、そのパラメーターの下で理論モデルの解を求めなければならない。単純に考えれば、このようなサイクルを繰り返し、理論モデルの解とデータをできるだけ近づけるような構造パラメーターが見つければ、それが推定すべきパラメーターとなる。しかし、一般的にはこのようなパラメーターが見つかるまで膨大な計算が必要となり、しばしばこのような単純な計算方法が不可能な場合もある。

1.2 本稿の目的と対象

上に述べたような構造推定、特に動的離散選択モデルの推定の特徴と問題点を考えると、構造推定に関する研究のもっとも重要な論点は、「どのような計算手法（アルゴリズム）を用いれば、推定結果の正確性を犠牲にせずに計算量や計算時間を小さくすることができるか？」という問題だと言っても過言ではないだろう。実際、この分野の研究の中で多くのアルゴリズムが提案されてきた。そこで、本稿の目的は、動的離散選択モデルの代表的なアルゴリズムである「ネステッド不動点アルゴリズム（Nested fixed point algorithm, NFXP）」と「ネステッド疑似尤度アルゴリズム（Nested pseudo-likelihood algorithm, NPL）」の2つを中心に、動的離散選択モデルを推定するためのいくつかのアルゴリズムについて平易に説明することである³。動的離散選択モデルを使って分析される応用分野は、労働経済学、産業組織論、経済人口学、医療経済学、開発経済学、政治経済学、マーケティングと多岐にわたるが、本稿では特定の分野を想定していない。

なお、本稿を書くにあたって、Aguirregabiria and Mira (2010) を参考にした。モデルの設定、表記法、応用例についてはほぼそれにしたがった。動的離散選択モデルの構造推定には、他にもすぐれたサーベイ論文がある。特に、Miller (1997), Keane, Todd, and Wolpin (2011), Arcidiacono and Ellickson (2011) なども参考にした。本稿では、多くの研究論文で用いられているアイデアをできるだけ平易な表現を用いて説明することにつとめた。したがって、この研究分野における日本語で書かれた初歩的なガイドとして読まれればさいわいである⁴。

本稿の構成は以下の通りである。第2節では、動的離散選択の基本モデルについて説明する。第3節では、そのモデルにもとづいて推定する手法について議論する。ここでは、NFXP と NPL を解説する。第4節では、実際にモンテカルロ法によってデータを生成し、そのデータより簡単な構造推定を行なう。第5節では、NFXP や NPL 以外の推定法を解説し、最近の比較的

3 ただし、このようなアルゴリズムの日本語名称は、筆者が本稿を書くにあたって便宜上名付けたものにすぎない。一般的には、本来の英語での名称またはその略称で通用していることに注意されたい。

4 ただし、言うまでもなくありうる誤解や誤植、計算の間違いなどはすべて筆者の責に帰すものである。

新しい手法についても触れる。最後に、経済分野の応用例を第6節で説明した後、第7節をまとめとする。

2 基本モデル

この節では、動的離散選択の基本モデルについて説明する。まず、モデルをDP問題として設定し、いくつかの仮定をおいた上でそれを推定しやすい形に変形する。モデルの解として、条件付き選択確率（CCP）を導出することがここでの目的となる。そして、そのモデルをロジット型に限定し、推定を行なう準備とする。

2.1 動的離散選択モデルの基本設定

この動的離散選択モデルでは、期間を t であらわし、それぞれ独立に意思決定を行なう個人（またはエージェント）を i であらわす。各個人 i は各期ごとに自分の利得にかかわる（payoff-relevant）ような状態（state）を観察した上で、1つの行動（action, decision）を選択しなければならない。この状態を s_{it} であらわし、行動を a_{it} であらわそう。行動 a_{it} は $A = \{a^1, \dots, a^J\}$, $J < \infty$ から選択されるものとする⁵。次に、個人が得ることができる効用は各期ごとに発生するとしよう。すると、個人 i が t 期に得られる効用は、状態 s_{it} と行動 a_{it} の関数として $U(s_{it}, a_{it})$ と書ける。この効用関数 U は期間を通じて不変で、すべての個人に共通であるとする。動的離散選択モデルでは、個人 i はその効用の割引総和の条件付き期待値

$$E\left(\sum_{\tau=0}^{\infty} \beta^\tau U(s_{i,t+\tau}, a_{i,t+\tau}) \mid s_{it}, a_{it}\right) \quad (1)$$

を最大化するように、行動の列 a_{i1}, a_{i2}, \dots を選択する。ただし、 $\beta \in (0, 1)$ は期間を通じて不変ですべての個人に共通な割引因子である。

ここで、個人は将来に実現する状態に関して完全な情報を持っていない。そこで、各個人の将来の状態に関する予測は、1次のマルコフ推移確率にしたがうとする。具体的には、個人 i が t 期において状態 s_{it} を観察して行動 a_{it} を選択したとき、次の $t+1$ 期に状態が $s_{i,t+1}$ となる確率（信念）を推移確率（transition probability）と呼び、 $p(s_{i,t+1} \mid s_{it}, a_{it})$ であらわす。この推移確率は期間を通じて不変で、すべての個人に共通であるとする。以上の $\{U, p, \beta\}$ を動的離散選択モデルの構成要素（primitive）と呼ぼう。構造推定（structural estimation）とは、この構成要素を決定する構造パラメーター（structural parameter）を理論モデルにもとづいて推定することである。

5 本稿では同じ記号を用いて、下付きの添字（ it ）で「個人」と「期」を、上付きの添字で「可能な値の候補」をあらわすことにする。例えば、“ $a_{it} = a^j$ ”は「 t 期に個人 i が選択した行動が $a^j \in A$ であること」を示す。

この個人の意思決定問題は、動的計画法（Dynamic Programming, DP）によって考えることができる⁶。(1) 式の効用を最大化する DP 問題は、ベルマンの最適性原理により、価値関数（value function）を V とする次のようなベルマン方程式（Bellman equation）であらわされる。

$$V(s_{it}) = \max_{a \in A} \left\{ U(s_{it}, a) + \beta \int V(s_{i,t+1}) dp(s_{i,t+1} | s_{it}, a) \right\}. \quad (2)$$

ベルマン方程式とは、(1) の効用を最大にする問題を、価値関数が再帰的になるような形であらわしたものである。ここで、(2) 式の右辺の最大化する部分 $\{\cdot\}$ を次のようにあらわそう。

$$v(s_{it}, a) = U(s_{it}, a) + \beta \int V(s_{i,t+1}) dp(s_{i,t+1} | s_{it}, a). \quad (3)$$

すると、個人の最適行動ルールは状態 s_{it} の関数として $\delta(s_{it}) = \arg \max_{a \in A} \{v(s_{it}, a)\}$ と書ける。

本稿では、この $v(s_{it}, a)$ を行動が a であるときの選択価値関数（choice-specific value function）と呼ぼう。

さて、われわれは、各個人がこのような DP 問題を解いて合理的に最適行動ルールを決定しているという前提の下で、このモデルの構成要素を決定する構造パラメーターを状態と個人の行動に関するデータを用いて推定したい。ところが、われわれは状態に関してあらゆるデータを観察できるわけではない。例えば、経済状況を示す指標は観察可能なデータであるが、各企業の財務・費用情報は私的情報であり観察不可能なデータであろう。あるいは、個人の結婚歴は観察可能であるが、個人の嗜好はわれわれには観察不可能である。したがって、通常このような分析では、状態を $s_{it} = (x_{it}, \varepsilon_{it})$ のように 2 つに分けて、 x_{it} は誰でも観察可能な状態、 ε_{it} は個人 i にしか観察されない状態とする⁷。したがって、われわれが観察できるデータは、「観察可能な状態」と「個人の行動」の 2 つである。これを $Data = \{(x_{it}, a_{it}) : i = 1, \dots, N; t = 1, \dots, T_i\}$ であらわすことにしよう⁸。一方、データに含まれない「観察不可能な状態」の ε_{it} は、個人の行動に関する変動と見なされる。つまり、後述するように、 ε_{it} は推定上の「誤差」として扱われる。

2.2 いくつかの仮定と推定の準備

このような動的離散選択モデルを構造推定するために、未知の構造パラメーターに関する尤度関数を設定し、それを最大化することを考えよう。そのような尤度関数を設定するために、個人の最適行動ルール δ を観察不可能な状態 ε_{it} で積分して、「ある観察可能な状態 x_{it} の下で、ある

6 経済分野の動的計画法の初歩としては、Stokey and Lucas (1989) を参照。また、Judd (1998) や Adda and Cooper (2003) はその数値計算による解法について詳しい。

7 一般的に、 x_{it} , ε_{it} はベクトルであるが、特に ε_{it} は $(J \times 1)$ 次元であるとする。

8 場合によっては、これに加えて利得変数（payoff variable）を考えることもある。利得変数は状態と行動に依存して決まる観察可能な変数である。後で説明する Rust and Phelan (1997) では、利得変数を用いたモデルが解説されている。

行動 a_{it} が選択される確率」を求めたい⁹。ところが、一般的にはこの積分を解くことは難しいので、それが計算可能となるように状態 ε_{it} に関するいくつかの仮定を置く必要がある。Rust (1987) 以降の研究分析においては、次のような3つの標準的な仮定が使われている¹⁰。

仮定 AS「加法分離性」(Additive Separability)

効用関数 U は次のように2つの部分に分けられる。

$$U(x_{it}, \varepsilon_{it}, a) = u(x_{it}, a) + \varepsilon_{it}(a). \quad (4)$$

ここで、 $\varepsilon_{it}(a)$ は平均0の確率変数で、行動 a に対応する ε_{it} の要素である。

仮定 CI「条件付き独立性」(Conditional Independence)

観察不可能な状態 ε_{it} は i, t に関して互いに独立で同一の、状態 x_{it} を条件とする確率分布 $g(\varepsilon_{it}|x_{it})$ にしたがう。さらに、推移確率は次のような積の形となる。

$$p(x_{i,t+1}, \varepsilon_{i,t+1} | x_{it}, \varepsilon_{it}, a_{it}) = f(x_{i,t+1} | x_{it}, a_{it}) \cdot g(\varepsilon_{i,t+1} | x_{i,t+1}). \quad (5)$$

ここで、 $f(x_{i,t+1} | x_{it}, a_{it})$ は ε_{it} に依存しない確率関数である。

仮定 DIS「離散型の状態」(Discrete Support)

状態 x_{it} のとる範囲(状態空間)は離散かつ有限である。つまり、ある $M < \infty$ が存在して、 $x_{it} \in X = \{x^1, \dots, x^M\}$ と書ける。

つまり、仮定 AS は、効用関数に関して観察可能な状態 x_{it} と観察不可能な状態 ε_{it} が加法的に分離されることを意味しており、仮定 CI は、推移確率に関してそれらが積の形で分離されることを意味している。そのような状態の推移プロセスを図3に示した(Rust, 1988, Fig. 1)。状態 x_{it} が次の期の状態 $x_{i,t+1}$ に影響するのに対して、状態 ε_{it} は互いに依存関係(系列相関)はなく、 ε_{it} は同じ期の状態 x_{it} にもみ依存する¹¹。 t 期の個人の最適行動ルール δ は状態 $s_{it} = (x_{it}, \varepsilon_{it})$ に依存するが、その δ によって選択された a_{it} は次の期の状態 $x_{i,t+1}$ にもみ影響を与える。また、仮定 DIS により状態 x_{it} は離散型であるので、その積分は総和の形となるが、状態 ε_{it} は一般的には連続な確率分布にしたがうと仮定される。

9 このように積分によってある変数を消去することは、積分消去(integrate out)または周辺化と呼ばれる。

10 Aguirregabiria and Mira (2002) では、観察不可能な状態 ε_{it} が独立同分布(independent and identically distributed, IID)であることを1つの仮定としているが、本稿ではこれを条件付き独立性の仮定の中にまとめた。

11 ただし、Aguirregabiria and Mira (2010) は、このように ε_{it} が x_{it} に依存すると仮定した分析はほとんどないと指摘している。実際、後で見る Rust モデルにおいても ε_{it} は x_{it} には依存していないと仮定している。

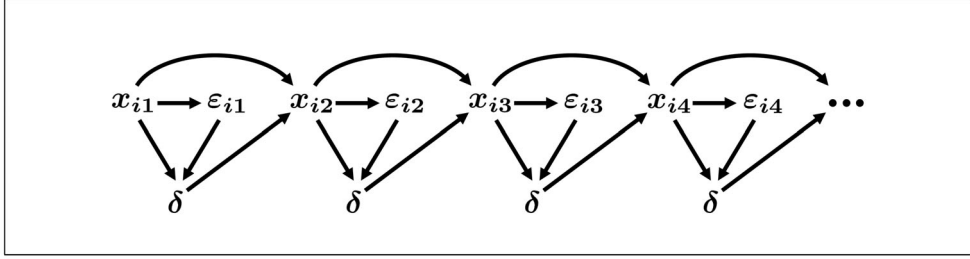


図3 状態の推移プロセス (Rust, 1988. Fig. 1)

このような仮定により，ベルマン方程式を書き換える．まず，(3) 式の選択価値関数は次のように書き換えることができる．

$$v(x_{it}, \varepsilon_{it}, a) = u(x_{it}, a) + \varepsilon_{it}(a) + \beta \sum_{m=1}^M f(x^m | x_{it}, a) \int V(x^m, \varepsilon_{i,t+1}) dg(\varepsilon_{i,t+1} | x_{i,t+1}). \quad (6)$$

次に，積分価値関数 (integrated value function) というものを考えよう¹²．これは， $\bar{V}(x) \equiv \int V(x, \varepsilon) dg(\varepsilon | x)$ として定義される．すると，(6) 式は

$$v(x_{it}, \varepsilon_{it}, a) = u(x_{it}, a) + \varepsilon_{it}(a) + \beta \sum_{m=1}^M f(x^m | x_{it}, a) \bar{V}(x^m) \quad (7)$$

となるので， $V(x_{it}, \varepsilon_{it}) = \max_{a \in A} \{v(x_{it}, \varepsilon_{it}, a)\}$ より，

$$\bar{V}(x_{it}) = \int \max_{a \in A} \left\{ u(x_{it}, a) + \varepsilon_{it}(a) + \beta \sum_{m=1}^M f(x^m | x_{it}, a) \bar{V}(x^m) \right\} dg(\varepsilon_{it} | x_{it}) \quad (8)$$

となる．この式は， \bar{V} に関して再帰的な関係を持っていることに注目しよう．つまり，上の仮定の下で，(2) 式のベルマン方程式は \bar{V} を用いて (8) 式のように書き換えることができた．

以降では， V の代わりにこの \bar{V} を用いることとし，そのベルマン方程式の一意的な解を求めることを考える．

2.3 条件付き選択確率と反復計算

(8) 式のベルマン方程式の解を，以下の手順で求めよう．

まず，上の定式化を行列・ベクトル表現であらわそう．つまり，

$$\mathbf{F}(a) = \begin{bmatrix} f(x^1 | x^1, a) & \dots & f(x^M | x^1, a) \\ f(x^1 | x^2, a) & \dots & f(x^M | x^2, a) \\ \vdots & \ddots & \vdots \\ f(x^1 | x^M, a) & \dots & f(x^M | x^M, a) \end{bmatrix},$$

12 あるいは，期待価値関数，Emax 関数とも呼ばれる．これは，McFadden (1981) によって社会余剰関数 (social surplus function) として導入された．

$\mathbf{u}(a) = (u(x^1, a), \dots, u(x^M, a))'$, $\boldsymbol{\varepsilon}(a) = (\varepsilon(a), \dots, \varepsilon(a))'$ とすると, 積分価値関数 $\bar{\mathbf{V}} = (\bar{V}(x^1), \dots, \bar{V}(x^M))'$ は,

$$\bar{\mathbf{V}} = \int \max_{a \in A} \{ \mathbf{u}(a) + \boldsymbol{\varepsilon}(a) + \beta \mathbf{F}(a) \bar{\mathbf{V}} \} d\mathbf{G}(\boldsymbol{\varepsilon}) \quad (9)$$

と簡潔にあらわすことができる¹³.

次に, 個人 i が t 期の状態 $x_{it} = x$ を観察したときに行動 $a_{it} = a$ を選択する確率を考えよう. この確率は, 最適行動ルールを観察不可能な状態 ε_{it} で積分したものであるので,

$$\begin{aligned} P(a|x) &= \int I(\delta(x, \varepsilon_{it}) = a) dg(\varepsilon_{it}|x) \\ &= \int I\left(a = \arg \max_{a' \in A} \left\{ u(x, a') + \varepsilon_{it}(a') + \beta \sum_{m=1}^M f(x^m|x, a') \bar{V}(x^m) \right\}\right) dg(\varepsilon_{it}|x) \end{aligned} \quad (10)$$

と書ける. ここで $I(\cdot)$ は指示関数である. この P は条件付き選択確率 (Conditional Choice Probability, CCP) と呼ばれる. 以下の構造推定の方法において, この条件付き選択確率は重要な役割を果たすことになる. その方法の基本的なアイデアとは, 解が価値関数の形となる本来のベルマン方程式を, 条件付き選択確率の形で解くというものである. つまり, 条件付き選択確率から条件付き選択確率へ写すような写像 (オペレーター) をベルマン方程式より求め, その不動点を解とすればよい. 本稿では, 便宜的にこのオペレーターを「CCP オペレーター」と呼ぶことにしよう¹⁴. CCP オペレーターは以下のような手順で求められる.

まず, この $P(a|x)$, $x = x^1, \dots, x^M$; $a = a^1, \dots, a^J$ を要素とする ($M \times J$) の行列を \mathbf{P} としよう¹⁵. この \mathbf{P} が求めるべき解の候補である. (10) 式より P は \bar{V} の写像になっているので, \mathbf{P} を $\bar{\mathbf{V}}$ の写像と見なしたものを, $\boldsymbol{\Lambda}(\cdot)$ であらわすことにする.

次に, この条件付き選択確率の定義より, (8) 式は状態 x の関数として次のように書き換えられる.

$$\bar{V}(x) = \sum_{a \in A} P(a|x) \left\{ u(x, a) + E[\varepsilon(a)|x, \delta(x, \varepsilon) = a] + \beta \sum_{m=1}^M f(x^m|x, a) \bar{V}(x^m) \right\}. \quad (11)$$

$\mathbf{P}(a) = (P(a|x^1), \dots, P(a|x^M))'$, $\mathbf{E}(a) = (E[\varepsilon_{it}(a)|x^1, \delta(x^1, \varepsilon) = a], \dots, E[\varepsilon_{it}(a)|x^M, \delta(x^M, \varepsilon) = a])'$ とおくと, (11) 式は行列・ベクトル表現で

13 ここで, 積分記号 $\int \cdot d\mathbf{G}(\boldsymbol{\varepsilon})$ はベクトルの要素ごとの積分をまとめたものを意味している. また, $\max_{a \in A}$ もベクトルの要素ごとに最大化を行なうという意味である.
 14 Aguirregabiria and Mira (2002) その他では, policy iteration operator (PI operator) と呼んでいる.
 15 多くの研究分析では, $P(a|x)$ を要素とする ($MJ \times 1$) の列ベクトルとして \mathbf{P} を定義しているが, 行列として考えても以下の議論には影響しない.

$$\bar{V} = \sum_{a \in A} \mathbf{P}(a) * \{\mathbf{u}(a) + \mathbf{E}(a) + \beta \mathbf{F}(a) \bar{V}\}$$

と書ける。ここで*は要素ごとの積（アダマール積）をあらわす演算子である¹⁶。これより、

$$\bar{V} = \left[\mathbf{I}_M - \beta \sum_{a \in A} \mathbf{P}(a) * \mathbf{F}(a) \right]^{-1} \sum_{a \in A} \mathbf{P}(a) * \{\mathbf{u}(a) + \mathbf{E}(a)\} \quad (12)$$

となる。この \bar{V} は \mathbf{P} の写像となっていることに注目しよう。Hotz and Miller (1993) の Proposition 1 は、 $\mathbf{E}(a)$ も \mathbf{P} の写像となっていることを示している¹⁷。そこで、この \bar{V} を \mathbf{P} の写像と見なしたものを $\Phi(\cdot)$ であらわそう。

さて、われわれの目的は、DP問題の解 \mathbf{P} を求めることであった。(10)式より \bar{V} の写像として $\Lambda(\cdot)$ が得られ、(12)式より \mathbf{P} の写像として $\Phi(\cdot)$ が得られた。そこで、この2つの写像の合成写像を $\Psi(\cdot) \equiv \Lambda(\Phi(\cdot))$ とすると、この Ψ が目的のCCPオペレーターということになる。つまり、解のあらゆる候補を $\Psi(\cdot)$ に代入して

$$\mathbf{P} = \Psi(\mathbf{P}) \quad (13)$$

の関係が成り立てば、その不動点 \mathbf{P} こそが求めたい解となる。

以上の準備の上で、(13)式を解くアルゴリズムを考えよう。Aguirregabiria and Mira (2002) は、 Ψ は一意的な不動点を持ち、繰り返し代入により \mathbf{P} は収束し、同時に \bar{V} も収束することを示した。これを命題の形にまとめておこう。

命題 1 (Aguirregabiria and Mira (2002), Proposition 1) 仮定 AS (加法分離性), 仮定 CI (条件付き独立性), 仮定 DIS (離散型の状態) が満たされるとき、

- (a) Ψ は一意的な不動点 \mathbf{P} を持つ。
- (b) 任意の $\mathbf{P}^{(0)}$ に対して、列 $\mathbf{P}^{(K)} = \Psi(\mathbf{P}^{(K-1)})$, $K=1, 2, \dots$, は \mathbf{P} に収束する。
- (c) 任意の $\mathbf{P}^{(0)}$ に対して、 $\bar{V}^{(K)} = \Phi(\mathbf{P}^{(K)})$, $\mathbf{P}^{(K+1)} = \Lambda(\bar{V}^{(K)})$ となるような列 $\{\bar{V}^{(K)}, \mathbf{P}^{(K)}\}$ を考えると、列 $\{\bar{V}^{(K)}\}$ は一意的な解に収束して、(8)式のベルマン方程式の解となる。

この命題より、 \mathbf{P} を繰り返し Ψ に代入し、 $\mathbf{P}^{(0)}, \mathbf{P}^{(1)}, \mathbf{P}^{(2)} \dots$ のように逐次「更新」していくことで不動点を求めることが可能となる。一般的には、このようにCCPオペレータを用いて不動点を求める手法を政策反復法 (policy iteration, policy function iteration) と呼ぶ。本稿で

16 つまり、 $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ とすれば、 $A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$ 。

17 さらに、後で説明するロジット型モデルでは、 $E[\varepsilon_{it}(a) | x, \delta(x, \varepsilon) = a] = -\ln P(a | x)$ という関係が成り立っている。これは付録Aで導出する。

は、より直感的に、このアルゴリズムを「CCP 不動点アルゴリズム」と呼ぼう。

CCP 不動点アルゴリズム

ステップ 0 最初の $\mathbf{P}^{(0)}$ の各要素に適当な $[0, 1]$ 間の値を割り当てる ("guess").

以降, $K=0, 1, \dots$ に対して次の 2 つのステップを繰り返す.

ステップ 1 $\mathbf{P}^{(K)}$ を (12) 式の右辺に代入して $\bar{\mathbf{V}}^{(K)} = \Phi(\mathbf{P}^{(K)})$ とする.

ステップ 2 $\bar{\mathbf{V}}^{(K)}$ を (10) 式の右辺に代入して $\mathbf{P}^{(K+1)} = \Lambda(\bar{\mathbf{V}}^{(K)})$ とする.

適当なノルムと $\epsilon > 0$ を決めておいて, $\|\mathbf{P}^{(K+1)} - \mathbf{P}^{(K)}\| < \epsilon$ となって収束が確認されたら計算を終了して解とする.

このように、CCP オペレーターを用いたアルゴリズムを使うためには、(10) 式と (12) 式より、それぞれ $\Lambda(\bar{\mathbf{V}})$ と $\Phi(\mathbf{P})$ を計算すればよい。ところが、一般的には、写像 $\Lambda(\cdot)$ と $\Phi(\cdot)$ を導出するのは難しい。そこで、通常このような分析では、以下で説明する「ロジット型モデル」が使われることが多い。次節でロジット型モデルを定式化し、以降ではこのモデルに限定して議論を進めよう。

2.4 ロジット型モデル

Rust (1987) に続く多くの動的離散選択モデルにおいては、観察不可能な状態 ε_{it} に関して仮定をもうけてモデルを推定しやすくしている。本稿では、そのようなモデルを「ロジット型モデル」と呼ぼう。ロジット型モデルとは、上の 3 つの仮定に次の仮定を加えたモデルである。

仮定 CLOGIT「条件付きロジット型モデル」(Conditional Logit Model)

観察不可能な状態 $\{\varepsilon_{it}(a^j), j=1, \dots, J\}$ は互いに独立であり、それぞれタイプ I 型極値分布にしたがう。

具体的に、 ε_{it} は次の確率分布関数にしたがうとしよう¹⁸。

$$g(\varepsilon_{it}) = \exp[-\exp(-\varepsilon_{it} + \gamma)] \quad (14)$$

ここで、 $\gamma = 0.577216 \dots$ は、 $\gamma \equiv \int_0^\infty \ln z \exp(-z) dz$ で定義されるオイラー定数である。このとき、条件付き選択確率は、 $x^m \in \{x^1, \dots, x^M\}$, $a^j \in \{a^1, \dots, a^J\}$ に対して

18 一般的なタイプ I 型極値分布 (Gumbel 分布) は、 $F(\varepsilon; \mu, \sigma) = \exp[-\exp(\varepsilon - \mu)/\sigma]$ となる ($\sigma > 0$)。本稿では、Rust (1987) にならい、 $\mu = \gamma$ (オイラー定数)、 $\sigma = 1$ とする。ただし、他の研究分析では $\mu = 0$ としているものも多い。

$$P(a^j|x^m) = \frac{\exp\{u(x^m, a^j) + \beta \mathbf{F}(x^m, a^j) \tilde{\mathbf{V}}\}}{\sum_{a \in A} \exp\{u(x^m, a) + \beta \mathbf{F}(x^m, a) \tilde{\mathbf{V}}\}} \quad (15)$$

となることがわかる。ただし、 $\mathbf{F}(x^m, a)$ は、行列 $\mathbf{F}(a)$ の m 行目の要素ベクトル ($1 \times M$) である。すると、 $\Lambda(\tilde{\mathbf{V}})$ は、(15) 式の $P(a^j|x^m)$ を m 行目 j 列目の要素とする行列

$$\Lambda(\tilde{\mathbf{V}}) = \begin{bmatrix} P(a^1|x^1) & \dots & P(a^J|x^1) \\ P(a^1|x^2) & \dots & P(a^J|x^2) \\ \vdots & \ddots & \vdots \\ P(a^1|x^M) & \dots & P(a^J|x^M) \end{bmatrix}$$

となって計算可能となる。

さらに、ロジット型モデルの場合、観察不可能な状態 ε_{it} が (14) 式にしたがうならば、(12) 式の $\mathbf{E}(a)$ は $-\ln(\mathbf{P}(a))$ と等しくなる。したがって、仮定 CLOGIT より、(12) 式は

$$\Phi(\mathbf{P}) = \left[\mathbf{I}_M - \beta \sum_{a \in A} \mathbf{P}(a) * \mathbf{F}(a) \right]^{-1} \sum_{a \in A} \mathbf{P}(a) * \{\mathbf{u}(a) - \ln(\mathbf{P}(a))\} \quad (16)$$

と書くことができ計算可能となる。

ロジット型モデルの条件付き選択確率と $\mathbf{E}(a)$ の導出を付録 A に付した。後で試みるモンテカルロ・シミュレーションによる推定では、このロジット型モデルを仮定して、 $\Lambda(\tilde{\mathbf{V}})$ と $\Phi(\mathbf{P})$ を計算することにする。

なお、ロジット型モデルにおいて、積分価値関数は $x^m \in \{x^1, \dots, x^M\}$ に対して

$$\tilde{V}(x^m) = \ln \left(\sum_{a \in A} \exp\{u(x^m, a) + \beta \mathbf{F}(x^m, a) \tilde{\mathbf{V}}\} \right) \quad (17)$$

となる¹⁹。(17) 式を行列・ベクトル表現であらわすと、

$$\tilde{\mathbf{V}} = \ln \left(\sum_{a \in A} \exp\{\mathbf{u}(a) + \beta \mathbf{F}(x^m, a) \tilde{\mathbf{V}}\} \right) \quad (18)$$

となるが、この (18) 式は、積分価値関数をあらわす表現として (12) 式の代替的なものになっている。ここで、(12) 式には $\mathbf{P}(a)$ が含まれているのに対し、(18) 式には含まれていない。つまり、(18) 式の右辺を「価値オペレーター (Γ)」として、 $\tilde{\mathbf{V}} = \Gamma(\tilde{\mathbf{V}})$ の一意的な不動点 $\tilde{\mathbf{V}}$

19 この場合、(15) 式の条件付き選択確率と (17) 式の積分価値関数の間には次のような関係があることが容易に確認できる。

$$P(x^m|a^j) = \frac{\partial}{\partial u(x^m|a^j)} \tilde{V}(x^m).$$

Rust (1987) の Theorem 1 は、仮定 CI (条件付き独立性) の下で、この関係が一般的に成り立つことを示している。この関係より、条件付き選択確率は積分価値関数より求めることもできる。

が存在すれば、これは (9) 式の解になっている。このように、ロジット型モデルでは、CCP オペレーターを介さずに (18) 式より DP 問題の解を求めることもできる。このように、価値オペレーターを用いて不動点を求める手法は、**価値反復法 (value iteration, value function iteration)** と呼ばれる。研究分析によっては、政策反復法ではなく、この手法を用いて不動点を求める場合もある。(付録 A で、ロジット型モデルにおける積分価値関数の導出を行なった。)

3 推定アルゴリズム

前節では、個人が DP 問題を解いているという前提で、ある構造パラメーターの下でその DP 問題の解を導出した。ここからは、われわれの観察可能なデータから動的離散選択モデルの構造パラメーターを推定する手法について議論しよう。ここでは、推定の代表的なアルゴリズムである NFXP と NPL について解説する。

3.1 尤度関数の導出

まず、構造パラメーターを推定するために、推定上の目的関数である尤度関数を導出する。まず、われわれが観察できるデータは、 $Data = \{(x_{it}, a_{it}) : i=1, \dots, N; t=1, \dots, T_i\}$ であった。推定したい構造パラメーターベクトルを、効用関数 u に関する θ_u 、 g に関する θ_g 、 f に関する θ_f とする²⁰。条件付き選択確率 P はそれらすべてに依存し、推移確率 f は θ_f にのみ依存していることに注意しよう。したがって、これより、条件付き選択確率を $P(a_{it}|x_{it}; \theta_u, \theta_g, \theta_f)$ 、推移確率を $f(x_{i,t+1}|x_{it}, a_{it}; \theta_f)$ と書くことにする。これは、ある $\theta = (\theta'_u, \theta'_g, \theta'_f)'$ のもとで、それぞれ (x_{it}, a_{it}) と $(x_{it}, x_{i,t+1}, a_{it})$ が生起する確率であるが、見方を変えればこれは $Data$ を条件とする θ の尤度 (likelihood) と見なすことができる。したがって、仮定 CI (条件付き独立性) と仮定 DIS (離散型の状態) より、 $Data$ が与えられたときの尤度関数 $L(\theta)$ は、構造パラメーター θ に対して、

$$L(\theta) = \prod_{i=1}^N \prod_{t=1}^{T_i} P(a_{it}|x_{it}; \theta_u, \theta_g, \theta_f) \times \prod_{i=1}^N \prod_{t=1}^{T_i-1} f(x_{i,t+1}|x_{it}, a_{it}; \theta_f) \quad (19)$$

と書くことができる。この $L(\theta)$ の対数をとったものを $\ell(\theta)$ とすると、対数尤度関数は

$$\ell(\theta) = \sum_{i=1}^N \sum_{t=1}^{T_i} \ln P(a_{it}, x_{it}; \theta_u, \theta_g, \theta_f) + \sum_{i=1}^N \sum_{t=1}^{T_i-1} \ln f(x_{i,t+1}|x_{it}, a_{it}; \theta_f) \quad (20)$$

となる。最尤法により、構造パラメーター θ の推定は、ある与えられた $Data$ のもとで、この $\ell(\theta)$ を最大化するような θ を求めることになる。これ以降、関数 u, g, f は、いずれも $\theta_u, \theta_g, \theta_f$ の各要素のすべてに関して 2 回連続的の微分可能であると仮定する²¹。

20 ここでは、特に $\theta_u, \theta_g, \theta_f$ の次元は特定しない。ただし、それぞれ列ベクトルであるとする。

21 ある関数が 2 回連続的の微分可能とは、その関数が 2 回微分可能で、2 階の導関数が連続であることで

3.2 ネステッド不動点アルゴリズム (NFXP)

ネステッド不動点アルゴリズム (Nested Fixed Point Algorithm, NFXP) とは、構造パラメーターを推定する外部ループ (outer loop) と、ある構造パラメーターのもとで動的計画法を解く内部ループ (inner loop) の2つを交互に (入れ子的に) 行なうアルゴリズムである。まず、構造パラメーターが θ のときの CCP オペレーター Ψ を Ψ_θ であらわそう。そこで、このアルゴリズムでは次のような反復計算を行なう。

ネステッド不動点アルゴリズム (NFXP)

ステップ 0 最初の $\theta^{(0)}$ に適当な値を割り当てる。

以降、 $k=0, 1, \dots$ に対して次の2つのステップを繰り返す。

ステップ 1 $\theta^{(k)}$ を所与として、CCP 不動点アルゴリズムを用いて $\mathbf{P} = \Psi_{\theta^{(k)}}(\mathbf{P})$ を満たすような解 \mathbf{P} を求める。

ステップ 2 その解 \mathbf{P} を用いて (20) 式を計算し、次のような更新ルールを用いて、より対数尤度関数 $\ell(\theta)$ を大きくするような次の候補 $\theta^{(k+1)}$ を見つける。

$$\theta^{(k+1)} = \theta^{(k)} - \mathbf{H}^{(k)} \frac{\partial \ell(\theta^{(k)})}{\partial \theta}. \quad (21)$$

(ここで、 $\mathbf{H}^{(k)}$ 、 $\partial \ell(\theta^{(k)}) / \partial \theta$ は本稿の付録 B で定義している.)

適当なノルムと $\epsilon > 0$ を決めておいて、 $\|\theta^{(k+1)} - \theta^{(k)}\| < \epsilon$ となって収束が確認されたら計算を終了して解とする。

3.3 ネステッド疑似尤度アルゴリズム (NPL)

次に、Aguirregabiria and Mira (2002) が提案したネステッド疑似尤度アルゴリズム (Nested Pseudo-Likelihood Algorithm, NPL) を説明する。まず、構造パラメーターのうち、 θ_f だけは先に求めておく。ここでは、第2項を最大化する θ_f を $\hat{\theta}_f$ とする。そこで、あらためて $\theta = (\theta'_w, \theta'_g, \hat{\theta}'_f)'$ とし、次のように目的関数を最大化する θ を求めることにする。

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^N \sum_{t=1}^{T_i} \sum_{m=1}^M \sum_{j=1}^J \ln \Psi_{\theta}(\mathbf{P})(a_{it} = a^j | x_{it} = x^m). \quad (22)$$

ここで、 $\Psi_{\theta}(\mathbf{P})(a_{it} = a^j | x_{it} = x^m)$ は i, t に対する行列 $\Psi_{\theta}(\mathbf{P})$ の m 行 j 列目の要素である。Aguirregabiria and Mira (2002) は、この目的関数を疑似尤度関数 (pseudo-likelihood function) と呼んでいる。これを用いれば、アルゴリズムは次のようになる。

ある。

ネステッド疑似尤度アルゴリズム (NPL)

ステップ0 最初の $\mathbf{P}^{(0)}$ の各要素に適当な $[0, 1]$ 間の値を割り当てる。

以降, $k=1, 2, \dots$ に対して次の2つのステップを繰り返す。

ステップ1 次のようにして, θ の次の候補を見つける。

$$\theta^{(k)} = \arg \max_{\theta} \sum_{i=1}^N \sum_{t=1}^{T_i} \sum_{m=1}^M \sum_{j=1}^J \ln \Psi_{\theta}(\mathbf{P}^{(k-1)})(a_{it} = a^j | x_{it} = x^m). \quad (23)$$

ステップ2 次のようにして, \mathbf{P} の次の候補を見つける。

$$\mathbf{P}^{(k)} = \Psi_{\theta^{(k)}}(\mathbf{P}^{(k-1)}). \quad (24)$$

この計算を $\mathbf{P}^{(k)}$ と $\theta^{(k)}$ が収束するまで行なう。

3.4 NFXP と NPL の比較

図4に2つのアルゴリズムの違いを示した。NFXPでは、外部ループの中で θ の更新が行われ、各 θ に対する内部ループの中で \mathbf{P} の収束計算を行なう。つまり、独立した2つのループが“入れ子”の形になっている。内部ループでは、ある $\theta^{(k)}$ に対するCCPオペレーター $\Psi_{\theta^{(k)}}$ を固定した上で、CCP不動点アルゴリズムによって不動点 \mathbf{P} が求まるまで収束計算を行ない、外部ループでは(21)式によって θ を更新していくので、NFXPでは全体として「 \mathbf{P} の更新回数 $\times \theta$ の更新回数」分の計算を行わなければならないことになる。

それに対して、NPLでは、外部ループの中で \mathbf{P} が更新され、内部ループの中で θ が更新されている。つまり、NPLはNFXPで更新する θ と \mathbf{P} が交換(スワップ)されていると言える。ここで、NPLでは外部ループと内部ループが同時に更新されていることに注目しよう。内部ループの中で(24)式のように疑似尤度関数を最大化するように θ が更新され、外部ループの中で、この θ と1期前の \mathbf{P} によって \mathbf{P} が更新される。すなわち、 \mathbf{P} を求めるごとに動的計画問題を解く必要はない。このことにより、NPLはNFXPに比べて計算回数が少なくなっている。

また、Aguirregabiria and Mira (2002) の Proposition 3 では、NFXP と NPL の解が同一であることが示されている。次の命題にまとめよう。

命題2 (Aguirregabiria and Mira (2002), Proposition 3) 任意のデータと \mathbf{P} に対して (23) 式が一意的な内点解を持つとする。すると、もし NPL が収束するならば尤度方程式の根へ収束する。

NFXP は必ず尤度方程式の根へ収束するので、NPL が収束するならば、2つのアルゴリズムは同じ意味を持つことになる。さらに、Aguirregabiria and Mira (2002) は、ある条件の下で、NPL による推定量が一致性と漸近正規性を持ち、最尤推定量と漸近的に等しくなることも示し

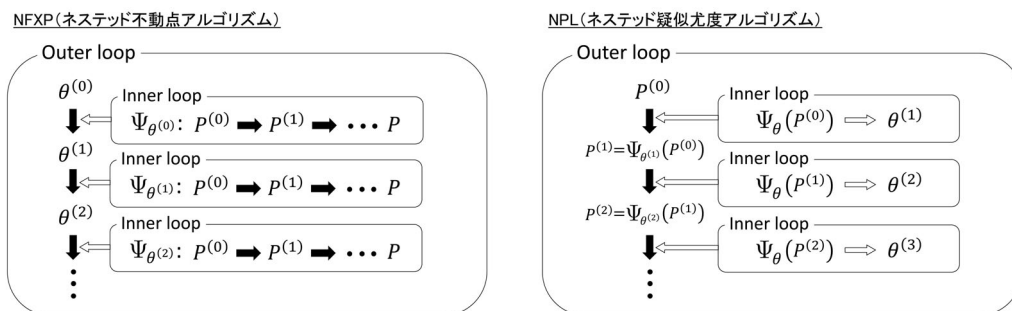


図4 NFXP と NPL

ている (Proposition 4).

次節では、モンテカルロ法を用いて動的離散選択モデルのデータを生成し、そのデータより NFXP と NPL の両アルゴリズムを使って構造推定を試みる。

4 モンテカルロ・シミュレーション

動的離散選択モデルの研究分析においては、アルゴリズムの効率性や正確性を確認するためにモンテカルロ法によるデータ生成を行ない、そのデータによる推定値を分析することがある。モンテカルロ法とは乱数を用いて数値計算を行なうことである。ここでは、モンテカルロ法を用いてある理論モデルからデータを生成して、そのデータより推定を行なうことを「モンテカルロ・シミュレーション」と呼ぼう。以下では、Rust (1987) が実際のデータを用いて構造推定を行なったのと同じ設定の下でモンテカルロ・シミュレーションを行なう。

4.1 Rust のバスエンジン交換モデル

この分野の代表的なモデルとして、Rust (1987) のバスエンジン交換モデルを考える。これは、マジソン（ウィスコンシン）メトロポリタン・バス会社 (Madison (Wisconsin) Metropolitan Bus Company) の修理担当者ハロルド・ザーチャー氏 (Harold Zurcher) が長期にわたってバスエンジンの交換に関する決定を行なったという動的計画問題である。ザーチャー氏はバスの状態 x (最後のエンジン交換時点からの走行距離) と彼にしか観察できない他の状態 ε を観察し、そのバスのエンジンを交換するか ($a=1$)、交換しないか ($a=0$) のいずれかを選択する。以下、このモデルにおける構造パラメーター $\theta = (\theta'_u, \theta'_g, \theta'_f)'$ を考えていこう。

まず、 t 期におけるバス i に関する効用関数は、状態、行動、構造パラメーターの関数として次のようにあらわされる。

$$U(x_{it}, \varepsilon_{it}, a; \theta_u) = u(x_{it}, a; \theta_u) + \varepsilon_{it}(a) \quad (25)$$

ただし、

$$u(x_{it}, a; \theta_u) = \begin{cases} -c(x_{it}; \theta_1) & \text{if } a_{it}=0, \\ -RC - c(\theta; \theta_1) & \text{if } a_{it}=1 \end{cases}$$

とする。ここで、 RC は交換に要する費用であり（ただし、スクラップを売却した利益は差し引かれている）、 $c(x_{it}; \theta_1)$ はバスの走行距離が x_{it} のときのメンテナンス費用である²²。したがって、効用関数に関するパラメーター（列ベクトル）は $\theta_u = (RC, \theta_1)'$ となる。

次に、 ε は $g(\varepsilon; \theta_g)$ にしたがう確率変数だが、これはロジット型モデルを仮定することによりパラメーターを推定する必要はない（つまり、 $\theta_g = \gamma = 0.577216 \dots$ ）。

さらに、状態 x_{it} は可能な全走行距離を等間隔の M 個のグリッドとして $x_{it} \in X = \{x^1, \dots, x^M\}$ としよう。すると、 x_{it} の推移確率は、 $d=0, \dots, M-1$ に対して、

$$f(x_{i,t+1} | x_{it} = x^j, a_{it}; \theta_f) = \begin{cases} \Pr \{x_{i,t+1} = x^{j+d}; \theta_3\} & \text{if } a_{it}=0, \\ \Pr \{x_{i,t+1} = x^{1+d}; \theta_3\} & \text{if } a_{it}=1 \end{cases}$$

となる。ここでは、現実的に可能な追加走行距離を限定して $d=0, \dots, 4$ とし、次のように考えよう²³。

$$\Pr \{x_{i,t+1} = x^{j+d}; \theta_3\} = \Pr \{x_{i,t+1} = x^{1+d}; \theta_3\} = \theta_{3d}, \quad d=0, \dots, 4; \quad \sum_{d=0}^4 \theta_{3d} = 1. \quad (26)$$

つまり、推移確率に関するパラメーター $\theta_3 = (\theta_{30}, \dots, \theta_{34})'$ は、追加走行距離が起きる確率そのものを考えればよい²⁴。

4.2 プログラムの説明

このRust (1987)のバスエンジン交換モデルを、NFXPとNPLの2つのアルゴリズムを使って推定し、推定結果を比較する²⁵。推定は、ワークステーションDell Precision T7610 (32GB RAM)上でMATLAB (R2015b)を用いて計算した。バスエンジンを交換しない場合を $a=a^0$ 、交換する場合を $a=a^1$ であらわす。費用関数は $c(x; \theta_u) = 0.001 \theta_{11}x$ とし、推移確率を $f(x^{m+d} | x^m, a^0; \theta_f) = \theta_{3d}$, $d=0, \dots, 4$ とする。Rust (1987)のTable Xを参考に、各パラメーターの真の値は、 $RC=11.7257$, $\theta_{11}=2.4569$, $\theta_{30}=0.0937$, $\theta_{31}=0.4475$,

22 Rust (1987)は8種類の費用関数 c を考えている。cubic: $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2 + \theta_{13}x^3$, quadratic: $c(x, \theta_1) = \theta_{11}x + \theta_{12}x^2$, linear: $c(x, \theta_1) = \theta_{11}x$, square root: $c(x, \theta_1) = \theta_{11}\sqrt{x}$, power: $c(x, \theta_1) = \theta_{11}x^{\theta_{12}}$, hyperbolic: $c(x, \theta_1) = \theta_{11}x/(91-x)$, mixed: $c(x, \theta_1) = \theta_{11}x/(91-x) + \theta_{11}\sqrt{x}$, および nonparametric である。

23 実際に推定するのは $\theta_{30}, \dots, \theta_{33}$ であり、 $\theta_{34} = 1 - \sum_{d=0}^3 \theta_{3d}$ となる。

24 Rust (1987)はバスのグループごとに推定値を求めており、例えば、全サンプルに関する結果は、 $\theta_{31}=0.475$, $\theta_{32}=0.517$, $\theta_{33}=0.007$ となっている (TABLE V, VI)

25 この推定は、Su and Judd (2012)の設定と数値にほぼしたがっている。

26 足して1になるように、 θ_{34} を付け加えた。

$\theta_{32}=0.4459$, $\theta_{33}=0.0127$, $\theta_{34}=0.0002$ とした²⁶.

ここでの目的は、モンテカルロ法により上の7つのパラメーターからデータを生成し、そのデータよりそれらのパラメーターを推定することである。データは、 $N=5$, $T=120$, $M=175$ とし、状態空間 X は、0 から 5000 までを M 個に分割したものである。

付録 C に MATLAB コードの一部 (説明用) を掲載した。プログラムの概要は以下のとおりである。

データの生成

まず、メインとなるプログラム *NFXP.m* より、サブプログラム *func_data.m* を呼び出してデータ生成を行なう。そのためには、真のパラメーターをもとに CCP 不動点アルゴリズムを用いて条件付き選択確率 (P_0) と積分価値関数 (v) を求める必要がある。そこで、まず $P_0(:, 1)$ に適当な初期値を与えてサブプログラム *func_Phi.m* より $v(:, 1)$ を求め、これをサブプログラム *func_Lambda* に代入して新しい $P_0(:, 2)$ を求める。このような過程を繰り返し、古い P_0 と新しい P_0 との差が $\text{ERROR}=1e-12$ ($=10^{-12}$) より小さくなったら反復計算を終了する。

次に、この $P_0, P_1 (=1-P_0)$ を用いてモンテカルロ法によってデータを生成する。データ生成プログラムは、一様乱数を発生させ、それが x の値に対応する P_1 より小さければ $a=1$, そうでなければ $a=0$ とする。 $a=0$ のとき、また乱数 r を発生させ、 $r < \theta_{30}$ ならば $+0$, $\theta_{30} < r < \theta_{31}$ ならば $+1$ というように次の期の x のレベルを更新する。このようにして、 $N * T=600$ 個の a と x を生成する。

内部ループ

このループでは、あるパラメーターベクトル θ を固定して、*func_Phi.m* と *func_Lambda.m* より v と P を計算する。反復計算は $\text{iter}=1, \dots, \text{iterMAX}$ 回行ない、収束した $v(:, \text{iter})$ を \bar{v} とする。

外部ループ

このループでは、パラメーターベクトル θ を更新する。この推定では、推移確率に関する θ_{33} のみはデータより推定しておき、効用関数に関する RC と θ_{11} のみを構造推定することにする。初期値は $RC=12$, $\theta_{11}=3$ とした。反復計算は $k=1, \dots, kMAX$ 回行ない、内部ループで得られた \bar{v} より勾配ベクトルとヘッセ行列を求めて、ニュートン法により θ を更新した。(ただし、0.02 をかけて収束の速度を調整している。) 対数尤度を計算して、古い θ のときより小さくなったら計算を終了した。(したがって、厳密には収束するまで計算していない。)

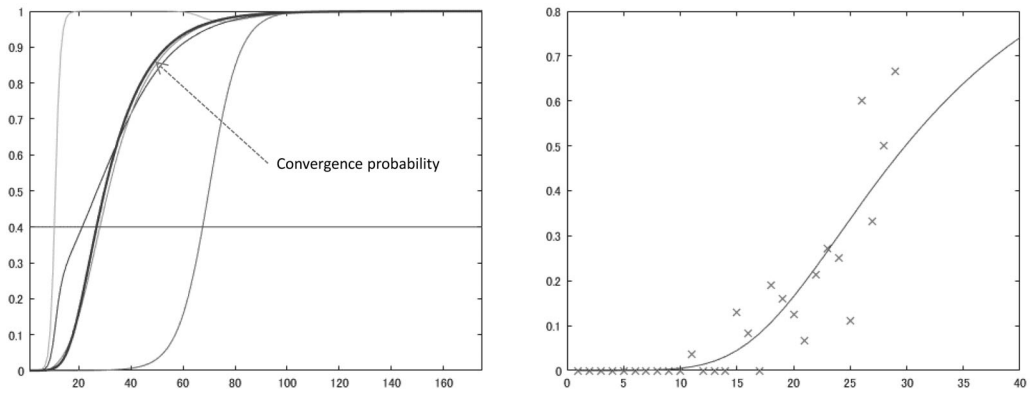


図5 交換確率とデータによる度数分布

4.3 推定結果

このようなプログラムを使って、以下のような結果が得られた。

まず、図5(左)に、パラメーターの真の値の下で内部ループの計算を行って得られた交換確率 P_1 の収束を示した。この場合は交換確率の初期値を状態にかかわらず0.4とし、反復計算を10回繰り返した。横軸に状態のレベル $1, \dots, 175$ を示している。この図から、交換確率は数回の反復計算で収束しているのが確認できる。収束した交換確率は、状態レベル30ぐらいで0.5に、レベル100ぐらいで1に達するのがわかる。一方、図5(右)は、この収束した交換確率のグラフ(実線)に、データより計算した、各状態で交換が発生した相対度数(x点)を重ねたものである。データでは、レベルが1から30までの状態が発生したが、状態レベル10に達するまで交換は一度も行われなかった。この図より、モンテカルロ法より得られたデータは、収束した交換確率 P_1 にほぼ一致するように生成されていることがわかる。

表1に推定結果を示した。この推定では、5つの割引因子 β (0.975, 0.980, 0.985, 0.990, 0.995) のそれぞれに対して、 $N=5$, $T=120$ のデータを100個生成し、それぞれのデータより RC と θ_{11} をNFXPとNPLを用いて構造推定した。MeanとStd. dev.はそれぞれ100個の推定値の平均と標準偏差、Mean Timeは1回の計算にかかった時間(秒)の平均、Mean of Iterationは外部ループの計算回数の平均をあらわしている。

その結果、NFXPとNPLでは、推定値にはそれほど差は見られなかった。一方、計算時間は明らかにNFXPよりもNPLにおいて短縮されている。また、計算回数もNFXPよりもNPLにおいて小さくなっていることがわかる。

この実験は RC と θ_{11} の2つのみ構造パラメーターを推定したので、計算時間、計算回数ともアルゴリズムによって大きな差は見られなかったが、パラメーターの数が増えるにつれて、この差は大きなものになっていくであろう。

5 その他のアルゴリズム

ここでは、NFXP、NPL 以外のアルゴリズムについて説明する。

5.1 CCP 推定法

Rust (1987) の NFXP アルゴリズムは概念的にはとてもわかりやすいものであるが、状態空間が大きくなると計算時間が膨大なものとなる。それに対して、Hotz and Miller (1993) は、DP 問題を解かないで構造パラメーターを推定する方法を提案した。これは、条件付き選択確率と推移確率をデータより推定し、その推定値を用いて（積分）価値関数を計算してパラメーターを推定する、という方法である。この方法は、しばしば「2 ステップアプローチ」とも呼ばれている。本稿で説明した NPL は、この CCP 推定法の再帰的な拡張と考えることもできる。（実際、Aguirregabiria and Mira (2010) は、彼らの NPL を "recursive CCP method" と呼んでいる。）

ここでは、CCP 推定法の基本的なアイデアを、ロジット型モデルによって簡単に説明する。まず、ロジット型では、条件付き確率は (15) 式で与えられる。それは、選択価値関数を用いると、あるパラメーター θ の下で $P(a^j|x) = \exp\{v(x, a^j)\} / \sum_{a \in A} \exp\{v(x, a^j)\}$ とあらわされる。

表 1 NFXP と NPL の計算結果^a

β	Algorithm	True values:	<i>RC</i> 11.726	θ_{11} 2.457	Mean Time (in sec.)	Mean of Iteration
0.975	NFXP	Mean	11.174	2.723	16.70	45.7
		Std. dev.	(0.498)	(0.356)		
	NPL	Mean	11.155	2.719	12.80	42.3
		Std. dev.	(0.454)	(0.364)		
0.980	NFXP	Mean	11.195	2.699	17.27	47.1
		Std. dev.	(0.554)	(0.387)		
	NPL	Mean	11.198	2.710	13.41	44.4
		Std. dev.	(0.541)	(0.394)		
0.985	NFXP	Mean	11.188	2.719	18.22	50.1
		Std. dev.	(0.451)	(0.367)		
	NPL	Mean	11.198	2.719	15.33	50.8
		Std. dev.	(0.448)	(0.373)		
0.990	NFXP	Mean	11.237	2.695	21.38	58.3
		Std. dev.	(0.513)	(0.365)		
	NPL	Mean	11.248	2.694	17.82	58.9
		Std. dev.	(0.510)	(0.370)		
0.995	NFXP	Mean	11.213	3.527	2.04	3.9
		Std. dev.	(0.029)	(0.074)		
	NPL	Mean	11.066	3.152	0.61	2.0
		Std. dev.	(0.025)	(0.058)		

a データを 100 個生成して、それぞれのデータより推定値を得た。Mean, Std. dev は、それぞれ推定値の平均と標準偏差、Mean Time は平均計算時間、Mean of Iteration は外部ループの平均回数。

ただし、 $v(x, a^j) \equiv u(x, a^j) + \beta \sum_{m=1}^M f(x^m | x_{it}, a^j) \bar{V}(x^m)$ とする²⁷。そこで、 a^1 を基準として次のような選択価値関数の「差」を考えよう。

$$\Delta v(x, a^j) \equiv v(x, a^j) - v(x, a^1), j=2, \dots, J. \quad (27)$$

すると、この $\Delta v(x, a^j)$ より $P(a^j | x)$ が求まることがわかる。(実際、 $\exp\{v(x, a^1)\} = 1$ と基準化すれば、 $P(a^j | x) = \exp\{\Delta v(x, a^j)\} / \sum_{a \in A} \exp\{\Delta v(x, a)\}$.) これを、 $\Delta v(x, a)$ から $P(a | x)$ への写像 Q とすると、Hotz and Miller (2010) の Proposition 1 より、任意の x に対して Q は逆写像を持つ。ロジット型モデルの場合は、 $Q^{-1}(P(a^j | x)) = \ln(P(a^j | x)) - \ln(P(a^1 | x))$ とすれば $\Delta v(x, a^j)$ が求まる。

そこで、CCP 推定法は、次の手順で行なう。(1) 観察されたデータを用いて各状態に対する選択された行動の相対度数を求める (図 5 の x 点)。(2) これを条件付き選択確率の推定値とし、 Q の逆写像により価値関数を求める。(3) この価値関数の推定値より、構造パラメーターを推定する。

Aguirregabiria and Mira (2010) では、 $v(x, a)$ を加法的に分解すること (「Hotz and Miller の表現」) により、より一般的なケースで CCP 推定法が使えることを示している。また、Aguirregabiria and Mira (2002) では、NPL の 1 回反復で得られた推定量が CCP 推定法の推定法と同じものであることを示している。

このように、CCP 推定法は DP 問題を解く必要がなく、特にロジット型モデルや、効用関数が線形である場合にはとても容易になるので多くの研究分析で使われてきた。しかし、この推定量は漸近性を持つとは限らない。そこで、この CCP 推定法を繰り返すことで、よい性質を持つ推定量を求めるために考えられた方法が NPL だと言える。

5.2 シミュレーションによる CCP 推定法

Hotz, Miller, Sanders and Smith (1997) は、モンテカルロ・シミュレーションによって、将来の効用を足し合わせることで価値関数を計算するというアイデアである。具体的には、ある条件付き選択確率の下で、「Hotz and Miller の表現」による 2 つの項をそれぞれ計算し、将来の行動と将来の状態の列を求める。そのような列を数多く求めて、その平均を将来の期待割引価値とし、それより推定量を求める。Hotz, Miller, Sanders and Smith (1997) は、シミュレーションの回数が多ければ一致性を持つような推定量を示した。

5.3 均衡制約つき最適化アルゴリズム (MPEC)

Su and Judd (2012) は、NFXP, NPL に代わる新しい推定アルゴリズムを提案している。

27 同じ記号を使うが、この $v(x, a^j)$ は (7) 式と違って $\varepsilon_{it}(a)$ は含まない。

このアルゴリズムは均衡制約つき最適化問題アルゴリズム（Mathematical Program with Equilibrium Constraints, MPEC）と呼ばれる。このアルゴリズムは、DP問題を解くことなく、制約付き最適化問題を1回だけ解くことにより構造パラメーターを推定することができる。つまり、構造パラメーターに対して得られる均衡式を制約として、構造パラメーターとモデルの内生変数によって目的関数を最適化する方法である。以下、Su and Judd (2012) にしたがって、ロジット型モデルに限定して説明しよう。

第2節の仮定の下で、 $EV(x, a) \equiv \sum_{m=1}^M f(x^m | x, a) \bar{V}(x^m)$ を定義しよう。ロジット型モデルの場合は、(17) 式より

$$\begin{aligned} EV(x, a) &= \sum_{m=1}^M \ln \left(\sum_{j=1}^J \exp \{ u(x^m, a^j) + \beta \mathbf{F}(x^m, a^j) \bar{V} \} \right) f(x^m | x, a) \\ &= \sum_{m=1}^M \ln \left(\sum_{j=1}^J \exp \{ u(x^m, a^j) + \beta EV(x^m, a^j) \} \right) f(x^m | x, a) \end{aligned} \quad (28)$$

となる。 $EV(x^m, a^j)$ を要素とする $(M \times J)$ の行列を \mathbf{EV} とすると、(28) 式は、パラメーター θ に対して次のような写像としてあらわすことができる。

$$\mathbf{EV} = T_{\theta}(\mathbf{EV}) \quad (29)$$

つまり、DP問題の解は(29)式の不動点として求められる。ロジット型モデルにおける(20)式の数値尤度関数を θ と \mathbf{EV} の関数として \mathcal{L} であらわそう。つまり、 \mathcal{L} は

$$\begin{aligned} \mathcal{L}(\theta; \mathbf{EV}) &= \sum_{i=1}^N \sum_{t=1}^{T_i} \ln \left(\frac{\exp \{ u(x_{it}, a_{it}; \theta_u) + \beta EV(x_{it}, a_{it}) \}}{\sum_{a \in A} \exp \{ u(x_{it}, a; \theta_u) + \beta EV(x_{it}, a) \}} \right) \\ &\quad + \sum_{i=1}^N \sum_{t=1}^{T_i-1} \ln f(x_{i,t+1} | x_{it}, a_{it}; \theta_f) \end{aligned} \quad (30)$$

とあらわすことができる。MPECアルゴリズムでは、次の制約付き最大化問題を解くことにより $\hat{\theta}$ を推定することができる。

$$\begin{aligned} \max_{\theta, \mathbf{EV}} \quad & \frac{1}{N} \mathcal{L}(\theta; \mathbf{EV}), \\ \text{subject to:} \quad & \mathbf{EV} = T_{\theta}(\mathbf{EV}). \end{aligned} \quad (31)$$

つまり、DP問題の解の条件を制約として、尤度関数（の標本平均）を最大化する問題を解けばよい。近年の数値計算ソフトはこのような制約付き最適化問題にも対応してきており、Su and Judd (2012) は MATLAB と AMPL のソルバーを用いてこのアルゴリズムにより Rust モデルを推定している。

5.4 標準的な仮定を満たさない場合

現実の研究分析では、観察不可能な状態が個人によって異なることがありえる。例えば、バス

エンジン交換の場合、バスが走る路線によって交換に関する選択確率が異なるかもしれない。一般に、観察可能な状態や行動が個人によって異なる観察不可能な状態に依存している場合は大きな問題となる。これを、**観察不可能な異質性 (unobserved heterogeneity)** の問題と呼ぼう。このような問題に対応するために、有限混合分布を仮定することがある。有限混合分布とは、異なるパラメーターにもとづく有限個の分布をウェイト付けして和をとった確率分布である。例えば、確率変数 x が異なる密度関数 $f_q(x)$, $q=1, \dots, Q$ にしたがう可能性がある場合、有限混合分布 p は π_q をウェイトとして

$$p(x) = \sum_{q=1}^Q \pi_q f_q(x) \quad (32)$$

となる。一般に、有限混合分布を使ったモデルは**有限混合モデル (finite mixture model)** と呼ばれる。有限混合モデルでは、確率分布のパラメーターに加えてウェイト π_q を推定する。Arcidiacono and Jones (2003) は、このモデルを**EM アルゴリズム**で推定する方法を提案している²⁸。さらに、Arcidiacono and Miller (2011) は、この EM アルゴリズムを観察不可能な状態の条件付き期待値と条件付き選択確率を求めるアルゴリズムに拡張している。本稿の第 6.2 節では、Aguirregabiria and Mira (2010) にしたがって、Keane and Wolpin (1997) のモデルを EM アルゴリズムで推定する方法を簡単に解説する。

その他の推定研究として、Arcidiacono, Bayer, Bugni, and James (2013) は、Sieve Value Function Iteration (SVFI) を用いて価値関数を近似する推定法を、Kasahara and Shimotsu (2008) はブートストラップ法を用いる手法を提案している。

さらに、Kasahara and Shimotsu (2011) は、(1) 仮定 AS が満たされないか、仮定 CLOGIT が満たされない、(2) 観察不可能な状態 ε_{it} が系列相関を持っている、(3) 選択される行動が ε_{it} に依存している、という状況での推定法を提案している。

6 経済分野の応用例

最初に述べたように、動的離散選択モデルを使って分析される応用分野は、労働経済学、産業組織論、経済人口学、医療経済学、開発経済学、政治経済学、マーケティングと多岐にわたる。

28 一般に、EM アルゴリズムとは、不完全なデータに基づいてパラメーターを推定する手法である。このアルゴリズムは、**E ステップ** (Expectation ステップ：観察されたデータが与えられた条件のもとで欠測しているデータに関する条件付き期待値を求めて、それを擬似的な観察データとする) と、**M ステップ** (Maximization ステップ：擬似的に完全化したデータにもとづいて、尤度を最大にするパラメーターを求める) の 2 つのステップを収束条件を満たすまで交互に繰り返す。動的離散選択モデルの場合は、観察された状態と行動のデータを条件として観察不可能な状態の条件付き期待値を求め (**E ステップ**)、観察不可能な状態を観察可能な状態と見なして最尤法によりパラメーターを求める (**M ステップ**) のを交互に繰り返せばよい。詳しくは、Greene (2011)、小西他 (2008)などを参照。

この節では、労働経済学の分野から Rust and Phelan (1997) の退職モデルと Keane and Wolpin (1997) の若年層の職業選択モデルをとりあげ、Aguirregabiria and Mira (2010) にしたがってそれらのモデルを簡単に説明する。まず、注8で述べた利得変数に関して仮定をもうけておこう。

仮定 CI-Y 「利得変数の条件付き独立性」

任意の現在の状態と行動に対して、利得変数 y は観察不可能な状態 ε_{it} とは独立である。

また、利得変数 y_{it} はその期の状態 x_{it} と行動 a_{it} によって決定されるとしよう。

6.1 Rust and Phelan (1997) の退職モデル

まず、NFXP アルゴリズムを使った例として Rust and Phelan (1997) の退職モデルがある。このモデルは、各期に個人が働き続けるか ($a_{it}=1$)、退職して年金生活を始めるか ($a_{it}=0$) のどちらかを選択する有限期間モデルである。個人 i の t 期の効用は消費支出 c_{it} の関数として

$$U(x_{it}, a_{it}, \varepsilon_{it}) = E(c_{it}^{\theta_{u1}} | x_{it}, a_{it}) \times \exp \left\{ \theta_{u2} + \theta_{u3} h_{it} + \theta_{u4} m_{it} + \theta_{u5} \frac{t_{it}}{1+t_{it}} \right\} - \theta_{u6} a_{it} + \varepsilon_{it}(a_{it}), \quad (33)$$

消費支出と収入 y_{it} の関係は

$$c_{it} = y_{it} - h c_{it}, \quad y_{it} = a_{it} w_{it} + (1 - a_{it}) b_{it}, \quad (34)$$

賃金 w_{it} は

$$w_{it} = \exp \left\{ \theta_{w1} + \theta_{w2} h_{it} + \theta_{w3} m_{it} + \theta_{w4} \frac{t_{it}}{1+t_{it}} + \theta_{w5} pp_{it} + \xi_{it} \right\} \quad (35)$$

となる。ただし、 h_{it} は健康状態、 m_{it} は結婚歴、 t_{it} は年齢、 $h c_{it}$ は医療支出（控除除く）である。 $b_{it} = b(ra_{it}, pp_{it})$ は（観察可能な）年金システムのルールで決まる年金で、 ra_{it} は退職時の年齢、 pp_{it} は年金点数とする。効用関数は、消費から得られる効用から労働による不効用を引いたものである。このモデルで、観察可能な状態は、 $x_{it} = \{h_{it}, hc_{it}, m_{it}, t_{it}, ra_{it}, pp_{it}\}$ となる。また、 y_{it} は観察可能な利得変数と考えることができる。さて、この状態のうち、 h_{it} と m_{it} はそれぞれ推移確率 $F_h(h_{i,t+1} | h_{it})$ と $F_m(m_{i,t+1} | m_{it})$ にしたがう、 t_{it} と ra_{it} は確定的なルール ($t_{i,t+1} = t_{it} + 1$, $ra_{i,t+1} = ra_{it} + 1$) にしたがう。 pp_{it} に関しては、年金システムにより $F_{pp}(pp_{i,t+1} | pp_{it}, w_{it})$ を想定している。さらに、 hc_{it} は h_{it} と m_{it} を条件とするパレート分布にしたがう。

以上の設定で、このモデルが仮定 AS、仮定 CI、仮定 DIS、仮定 CI-Y を満たしていることを確認しよう。観察不可能な状態 $\varepsilon_{it}(a)$ は、(33) 式の効用関数の中に加法分離的に含まれているので明らかに仮定 AS は満たされている。また、 ε_{it} には系列相関はなく、それとは独立に各状

態が推移している。さらに、 h_{it} と m_{it} は賃金 w_{it} とは無関係に推移すると仮定されているので、 ξ_{it} とも無関係である。したがって、仮定 CI も満たされている。そして、すべての状態は（医療支出と年金点数も含めて）離散型としているので、仮定 DIS も満たされることになる。さらに、 y_{it} は ε_{it} と無関係なので、仮定 CI-Y も満たされることになる。すると、状態、行動、利得の推移確率は、

$$\Pr(x_{i,t+1}, a_{i,t+1}, y_{i,t+1} | x_{it}, a_{it}, y_{it}) = f_y(y_{i,t+1} | x_{i,t+1}, a_{i,t+1}) \cdot P(a_{i,t+1} | x_{i,t+1}) \cdot f(x_{i,t+1} | x_{it}, a_{it}) \quad (36)$$

と積の形で書くことができる。 ξ_{it} が平均 0、分散 σ_ξ^2 の正規分布にしたがうとすれば、 f_y は $I\{a_{it}=1\} \cdot \phi\left(\left[\ln y_{it} - \theta_{w1} - \theta_{w2}h_{it} - \theta_{w3}m_{it} - \theta_{w4}\frac{t_{it}}{1+t_{it}} - \theta_{w5}pp_{it} - \xi_{it}\right] / \sigma_\xi\right)$ のようにあらわすことができるので、(36) 式より対数尤度関数を導出することができる。 ε_{it} に関して仮定 CLOGIT を仮定すれば、このモデルはロジット型の NFXP アルゴリズムで推定することができる。

Rust and Phelan (1997) は、このモデルを NFXP で推定し、米国で退職年齢のピークが 65 歳になるのは、社会保険便益が 65 歳以降の退職では不利なものとなり、また、65 歳までメディケアの受給資格を待つからであると結論づけている。

6.2 Keane and Wolpin (1997) の職業選択モデル

一方、Keane and Wolpin (1997) は、若年層の個人がどのような職業（または無職）を選択するかを動的離散選択モデルとして分析した。これも、Aguirregabiria and Mira (2010) にしたがって簡単にまとめる。このモデルでは、16 歳以上の個人 i が職業選択に関して t 期に行動 a_{it} を選択する。ただし、 $a_{it}=0$ （無職）、 $a_{it}=1$ （ホワイトカラー）、 $a_{it}=2$ （ブルーカラー）、 $a_{it}=3$ （軍職）、 $a_{it}=4$ （就学）とする。効用関数は、

$$U(x_{it}, a_{it}) = \begin{cases} \omega_i(0) + \varepsilon_{it}(0) & \text{if } a_{it}=0 \\ W_{it}(1) & \text{if } a_{it}=1 \\ W_{it}(2) & \text{if } a_{it}=2 \\ W_{it}(3) & \text{if } a_{it}=3 \\ \omega_{it}(4) - \theta_{tc1} I(h_{it} \geq 12) - \theta_{tc2} I(h_{it} \geq 16) + \varepsilon_{it}(4) & \text{if } a_{it}=4, \end{cases} \quad (37)$$

$$\text{ただし、 } W_{it}(a) = r_a \exp\left\{\omega_i(a) + \theta_{a1}h_{it} + \theta_{a2}k_{it}(a) - \theta_{a3}(k_{it}(a))^2 + \varepsilon_{it}(a)\right\}$$

である。ここで、 h_{it} は就学年数、 $k_{it}(a)$ は勤労年数、 ω_i は、 ε_{it} は観察不可能な状態とする。年齢を t_{it} とすると、観察可能な状態は $x_{it} = \{h_{it}, t_{it}, k_{it}(a)\}$ 、観察不可能な状態は $\varepsilon_{it}(a)$ と $\omega_{it}(a)$ となる ($a=1, 2, 3$)。 $W_{it}(a)$ は $a=1, 2, 3$ のときの賃金なので、このモデルでは観察可能な利得

変数と見なすことができる。（ $a=0, a=4$ のときは 0 となる。）すると、 $\varepsilon_{it}(a)$ が $W_{it}(a)$ に含まれているため、Rust and Phelan (1997) の例とは違い、仮定 CI-Y は満たされない。

もう一つ、このモデルでは、各個人が選択する行動がその個人に特有の観察不可能な状態 $\omega_i(a)$ に依存して、セレクション・バイアスが生じている可能性がある。なぜなら、職業選択は、観察可能な状態 x_{it} だけでなく、観察不可能な個人の能力や経済状態などにも依存するからである。これは、上で述べた観察不可能な異質性の問題ということになる。この問題を扱うために、次のような有限混合モデルを考える。

まず、Keane and Wolpin (1997) は 6 歳からの状態を含む個人のデータを入手している。つまり、第 1 期の状態 x_{i1} はすべての個人においてそれ以前の状態には依存せず、 ω_i にのみ依存しているはずである。個人のタイプが $Q < \infty$ あるとして、 $\omega_i \in \Omega = \{\omega^1, \dots, \omega^Q\}$ とし、個人 i がタイプ q であるときの尤度関数を

$$L_i(\theta, \omega^q) = \prod_{t=1}^{T_i} f_y(y_{it} | x_{it}, a_{it}; \theta, \omega^q) \cdot P(a_{it} | x_{it}; \theta, \omega^q) \prod_{t=1}^{T_i-1} f(x_{i,t+1} | x_{it}, a_{it}; \theta, \omega^q) \quad (38)$$

としよう。すると、初期の状態が x_{i1} のとき、 $\omega_i = \omega^q$ となる条件付き確率を $\pi(\omega^q | x_{i1})$ とすれば、個人 i の対数尤度関数は、 $\pi(\omega^q | x_{i1})$ をウェイトとして

$$\ell_i(\theta, \Omega, \pi) = \ln \left(\sum_{q=1}^Q \pi(\omega^q | x_{i1}) L_i(\theta, \omega^q) \right) \quad (39)$$

と書くことができる。ただし、 $\pi = (\pi(\omega^1 | x_{i1}), \dots, \pi(\omega^Q | x_{i1}))$ 。これを有限混合尤度関数と呼ぶ。そこで、Arcidiacono and Jones (2003) の提案した逐次的 EM アルゴリズムを使って、次のようにパラメーター $\{\hat{\theta}, \hat{\Omega}, \hat{\pi}\}$ を更新する。

逐次的 EM アルゴリズム (ESM)

次の 2 つのステップを、パラメーターが収束するまで交互に繰り返す。

E ステップ あるパラメーター $\{\hat{\theta}, \hat{\Omega}, \hat{\pi}\}$ の下で、状態、行動、利得の履歴 (history) が $(\mathbf{x}, \mathbf{a}, \mathbf{y})$ であるときの ω^q の条件付き確率 $\Pr(\omega^q | \mathbf{x}, \mathbf{a}, \mathbf{y}; \hat{\theta}, \hat{\Omega}, \hat{\pi}) = \hat{\pi}(\omega^q | x_{i1}) L_i(\hat{\theta}, \hat{\omega}^q) / \exp\{\ell_i(\hat{\theta}, \hat{\Omega}, \hat{\pi})\}$ を計算する。

(Sequential) M ステップ $\Pr(\omega^q | \mathbf{x}, \mathbf{a}, \mathbf{y}; \hat{\theta}, \hat{\Omega}, \hat{\pi})$ を最大化するように、 $\{\hat{\theta}, \hat{\Omega}^q, \hat{\pi}\}$ を推定する。ただし、 $\hat{\pi}(\omega^q | x_{i1}) = (1/Q) \sum_{i=1}^Q \Pr(\omega^q | \mathbf{x}, \mathbf{a}, \mathbf{y}; \hat{\theta}, \hat{\Omega}, \hat{\pi})$ とする。

Arcidiacono and Jones (2003) によると、このアルゴリズムが収束すれば、それは一致漸近的正規推定量となるが、それは漸近的有効性は持たない。彼らは学校選択モデルを推定して、ESM アルゴリズムが完全情報最尤法 (FIML) よりも計算速度が速いことを示している。

その他、NFXP を使って原子力発電所の稼働問題を推定した Rust and Phelan (1997)、出生

率をライフサイクルの最適化問題と考えて子供の価値を推定した Ahn (1995), 有限期間で医療支出を分析した Gilleskie (1998), また, 最近の研究では, 移民の問題の Keane and Walker (2011), ミンサー型賃金関数を推定した Ge (2013), 自動車購入問題の Cirillo, Xu, and Baskin (2015), 居住地選択問題の Bayer, McMillan, Murphy, and Timmins (2016), などがある. さらに, Keane and Wolpin (2009) は動的離散選択モデルを静的モデルと比較する形で解説し, Lee and Wolpin (2010), Keane and Wolpin (2000), Todd and Wolpin (2006), Lucarelli (2006), Diermeier, Keane, and Merle (2005), Collusi (2006) の6つの論文のそれぞれに関して問題を設定して概観した²⁹.

7 おわりに

筆者は, 2001年から2006年までニューヨーク市立大学大学院に在籍し, そこで Michael Grossman 教授の健康資本モデルに興味を持った³⁰. そのモデル自体は動学モデルをいわば静学的に解くというものであったが, それをきっかけに, その当時より急速に研究が進んでいた動的離散選択モデルに関心を抱くようになった. 動的離散選択モデルの研究はコンピューターの計算能力の発達とともに, 多くの経済学の分野において, 今なお新しい成果を生み出している. 本稿は, 動的離散選択モデルに貢献してきた数多の研究に心からの敬意を払いつつ, 十数年の間, 関心を持ち続けてきたこの分野にささやかな寄与をなしたいという動機によって作成したものである. ここで扱ったトピックはごく基本的かつ限定的なものにすぎないが, この分野のさらなる発展に資することができればさいわいである.

なお, 本稿ではシングルエージェントの意思決定問題のみを扱い, 動学ゲームについては言及しなかった. もし機会があれば, 動学ゲームに関するトピックについてもまとまった形で概説したい.

参考文献

Adda, J., & Cooper, R. (2003). *Dynamic Economics*. Cambridge, MA: MIT Press.

29 その6つの問題とは, 問題1: 1960年代の米国における (i) 賃金不平等の拡大, (ii) 大卒者プレミアムの上昇, (iii) 男女間の賃金格差の縮小, (iv) 女性の労働参加率の上昇, (v) サービス部門の成長, の要因となった経済変化とは何か? 問題2: 金銭的報酬は, 学歴と賃金に関する人種間格差をどの程度低下させるか? 問題3: 発展途上国における学歴の上昇に対して, もっとも効果的な政策は何か? 問題4: 処方薬の適用を含むメディケアの拡大は, 高齢者の健康と平均余命にどのような影響を与えるか? 問題5: 任期制限, 議員報酬の改定, 退職後の職業変更などの政策によって政治家の職業選択はどのように影響を受けるか? 問題6: 雇用者制裁と国境警備の増強がメキシコから米国への不法移民に対してどれぐらい影響を与えるか?である.

30 Grossman (1972).

- Aguirregabiria, V., & Mira, P. (2002). Swapping the nested fixed point algorithm: A class of estimators for discrete Markov decision models. *Econometrica*, 70, 1519-1543.
- Aguirregabiria, V., & Mira, P. (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156, 38-67.
- Ahn, N. (1995). Measuring the value of children by sex and age using a dynamic programming model. *Review of Economic Studies*, 62, 361-379.
- Arcidiacono, P., Bayer, P., Bugni, F. A., & James, J. (2013). Approximating high-dimensional dynamic models: Sieve value function iteration. In E. Choo & M. Shum (Eds.) *Structural econometric models (Advances in econometrics, Vol. 31)* (pp. 45-95). West Yorkshire. UK: Emerald Group Publishing.
- Arcidiacono, P., & Ellickson, P. B. (2011). Practical methods for estimation of dynamic discrete choice models. *Annual Review of Economics*, 3, 363-394.
- Arcidiacono, P., & Jones, J. (2003). Finite mixture distributions, sequential likelihood and the EM algorithm. *Econometrica*, 71, 933-946.
- Arcidiacono, P., & Miller, R. A. (2011). Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79, 1823-1867.
- Bayer, P., McMillan, R., Murphy, A., & Timmins, C. (2016). A dynamic model of demand for houses and neighborhoods. *Econometrica*, 84, 893-942.
- Cirillo, C., Xu, R., & Bastin, F. (2015). A dynamic formulation for car ownership modeling. *Transportation Science*, 50, 322-335.
- Colussi, A. (2006). Migrants' networks: An estimable model of illegal mexican immigration. Mimeo. University of Western Ontario.
- Diermeier, D., Keane, M. P., & Merlo, A. (2005). A political economy model of congressional careers. *American Economic Review*, 95, 347-373.
- 福島雅夫 (1996). 『数理計画入門』朝倉書店.
- Ge, S. (2013). Estimating the returns to schooling: Implications from a dynamic discrete choice model. *Labour Economics*, 20, 92-105.
- Gilleskie, D. B. (1998). A dynamic stochastic model of medical care use and work absence. *Econometrica*, 66, 1-45.
- Greene, W. H. (2011). *Econometric Analysis (8th Edition)*. London. UK: Pearson Education.
- Grossman, M. (1972). On the concept of health capital and the demand for health. *Journal of Political Economy*, 80, 223-255.
- Hotz, V. J., Miller, R. A., (1993). Conditional choice probabilities and estimation of dynamic models. *Review of Economic Studies*, 60, 497-529.
- Hotz, V. J., Miller, R. A., Sanders, S., & Smith, J. (1994). A simulation estimator for dynamic models of discrete choice. *Review of Economic Studies*, 61, 265-289.
- Judd, K. L. (1998). *Numerical Methods in Economics*. Cambridge. MA: MIT Press.
- Kasahara, H., & Shimotsu, K. (2008). Pseudo-likelihood estimation and bootstrap inference for structural discrete Markov decision models. *Journal of Econometrics*, 146, 92-106.
- Kasahara, H., & Shimotsu, K. (2011). Sequential estimation of dynamic programming models. Manuscript.
- Kennan, J., & Walker, J. R. (2011). The effect of expected income on individual migration decisions. *Econometrica*, 79, 211-251.
- Keane, M. P., Todd, P. E., & Wolpin, K. I. (2011). The structural estimation of behavioral models: Discrete choice dynamic programming methods and applications. In O. Ashenfelter, & D. Card (Eds.), *Handbook of Labor Economics, Vol. 4a* (pp. 331-461). Amsterdam: North-Holland.
- Keane, M. P., & Wolpin, K. I. (1997). The career decisions of young men. *Journal of Political Economy*, 105, 473-522.
- Keane, M. P., & Wolpin, K. I. (2000). Eliminating race differences in school attainment and labor

- market success. *Journal of Labor Economics*, 18, 614-652.
- Keane, M. P., & Wolpin, K. I. (2009). Empirical applications of discrete choice dynamic programming models. *Review of Economic Dynamics*, 12, 1-22.
- 小西貞則・越智義道・大森裕浩 (2008). 『計算統計学の方法 — ブートストラップ・EM アルゴリズム・MCMC —』朝倉書店.
- Lee, D., & Wolpin, K. I. (2010). Accounting for wage and employment changes in the US from 1968-2000: A dynamic model of labor market equilibrium. *Journal of Econometrics*, 156, 68-85.
- Lucarelli, C. (2006). An analysis of the medicare prescription drug benefit. PARC Working Paper Series. University of Pennsylvania.
- McFadden, D. (1981). Econometric models of probabilistic choice. In C. Manski, & D. McFadden (Eds.), *Structural Analysis of Discrete Data with Econometric Applications* (pp. 198-272). Cambridge, MA: MIT Press.
- Miller, R. (1997). Estimating models of dynamic optimization with microeconomic data. In H. H. Pesaran, & P. Schmidt (Eds.), *Handbook of Applied Econometrics, Vol. 2* (pp. 246-299). Amsterdam: North-Holland.
- Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica*, 55, 999-1033.
- Rust, J. (1988). Maximum likelihood estimation of discrete control processes. *SIAM Journal of Control and Optimization*, 26, 1006-1024.
- Rust, J. (1994). Structural estimation of Markov decision processes. In R. E. Engle, & D. McFadden (Eds.), *Handbook of econometrics, Vol. 4* (pp. 3082-3139). Amsterdam: North-Holland.
- Rust, J. (1996). Numerical dynamical programming in economics. In H. Amman, D. Kendrick, & J. Rust (Eds.), *Handbook of Computational Economics, Vol. 1* (pp. 619-729). Amsterdam: North-Holland.
- Rust, J., & Phelan, C. (1997). How social security and medicare affect retirement behavior in a world of incomplete markets. *Econometrica*, 65, 781-831.
- Rust, J., & Rothwell, G. (1995). Optimal response to a shift in regulatory regime: The case of the US nuclear power industry. *Journal of Applied Econometrics*, 10, S75-S118.
- Stokey, N. L., & Lucas, R. E., Jr. (1989). *Recursive Methods in Economic Dynamics*. Cambridge, MA: Harvard University Press.
- Todd, P. E. & Wolpin, K. I. (2006). Assessing the impact of a school subsidy program in Mexico: Using a social experiment to validate a dynamic behavioral model of child schooling and fertility. *American Economic Review*, 96, 1384-1417.
- Train, K. E. (2009). *Discrete Choice Methods with Simulation (2nd Edition)* New York, NY: Cambridge University Press.

付録 A 条件付き選択確率, $E(a)$, 積分価値関数の導出

ここでは、ロジット型モデルにおける [1] 条件付き選択確率, [2] (9) 式の $E(a)$, [3] 積分価値関数を導出する。

[1] まず, $v^j \equiv u(x^m, a^j) + \beta F(x^m, a^j) \bar{V}$ としよう。(この節を通じて x^m は 1 つ固定しておく。) すると, t 期において個人 i が行動 a^j を選択する確率は,

$$\begin{aligned} & \Pr\left\{v^j + \varepsilon_{it}(a^j) > v^h + \varepsilon_{it}(a^h); \forall h \neq j\right\} \\ & = \Pr\left\{\varepsilon_{it}(a^h) < \varepsilon_{it}(a^j) + v^j - v^h; \forall h \neq j\right\} \end{aligned} \quad (40)$$

となる。記号の簡略化のために、この節を通じて $\varepsilon^j \equiv \varepsilon_{it}(a^j)$ 、 $\varepsilon^h \equiv \varepsilon_{it}(a^h)$ としよう。すると、ある h に対して、個人が a^h ではなく a^j を選択する確率は、仮定 CLOGIT より、 $g(\varepsilon^j + v^j - v^h)$ となる。（ただし、 $g(\cdot)$ は (14) 式で定義されたものである。） ε^h 、 $h=1, \dots, J$; $h \neq j$ は互いに独立なので、個人が a^j を選択する確率は $g(\varepsilon^j + v^j - v^h)$ を $h \neq j$ に関してかけ合わせたものを $g(\varepsilon^j)$ に関して積分したものである。つまり、

$$\begin{aligned}
 P(a^j | x^m) &= \int_{-\infty}^{\infty} \prod_{h \neq j} g(\varepsilon^j + v^j - v^h) dg(\varepsilon^j) \\
 &= \int_{-\infty}^{\infty} \prod_{h \neq j} \exp\{-\exp[-(\varepsilon^j + v^j - v^h) + \gamma]\} dg(\varepsilon^j) \\
 &= \int_{-\infty}^{\infty} \prod_{h \neq j} \exp\{-\exp[-(\varepsilon^j + v^j - v^h) + \gamma]\} \\
 &\quad \times \exp(-\varepsilon^j + \gamma) \cdot \exp(-\exp(-\varepsilon^j + \gamma)) d\varepsilon^j \\
 &= \int_{-\infty}^{\infty} \prod_{h \neq j} \exp\{-\exp[-(\varepsilon^j + v^j - v^h) + \gamma]\} \\
 &\quad \times \exp(-\varepsilon^j + \gamma) \cdot \exp\{-\exp[-(\varepsilon^j + v^j - v^j) + \gamma]\} d\varepsilon^j \\
 &= \int_{-\infty}^{\infty} \prod_{h=1}^J \exp\{-\exp[-(\varepsilon^j + v^j - v^h) + \gamma]\} \cdot \exp(-\varepsilon^j + \gamma) d\varepsilon^j \\
 &= \int_{-\infty}^{\infty} \exp\{-\exp(-\varepsilon^j + \gamma) \sum_{h=1}^J \exp[-(v^j - v^h)]\} \cdot \exp(-\varepsilon^j + \gamma) d\varepsilon^j \quad (41)
 \end{aligned}$$

これを解くために、 $t \equiv \exp(-\varepsilon^j + \gamma)$ として置換積分を行う。すると、 $\varepsilon^j \rightarrow -\infty$ のとき $t \rightarrow \infty$ 、 $\varepsilon^j \rightarrow \infty$ のとき $t \rightarrow 0$ 、 $d\varepsilon^j = -(1/t)dt$ であるので、

$$\begin{aligned}
 P(a^j | x^m) &= \int_{\infty}^0 \exp\left\{-t \sum_{h=1}^J \exp[-(v^j - v^h)]\right\} \cdot t \left(-\frac{1}{t}\right) dt \\
 &= \int_0^{\infty} \exp\left\{-t \sum_{h=1}^J \exp[-(v^j - v^h)]\right\} dt \\
 &= \left| \frac{\exp\left\{-t \sum_{h=1}^J \exp[-(v^j - v^h)]\right\}}{\sum_{h=1}^J \exp[-(v^j - v^h)]} \right|_0^{\infty} \\
 &= \frac{1}{\sum_{h=1}^J \exp[-(v^j - v^h)]} \\
 &= \frac{\exp(v^j)}{\sum_{h=1}^J \exp(v^h)}. \quad (42)
 \end{aligned}$$

したがって、 v^j の定義より (15) 式が得られる。

[2] $E(a)$ も同様に導出できる。これは条件付き期待値なので、

$$\begin{aligned}
 E[\varepsilon^j | x^m, \delta(x^m, \varepsilon^j) = a^j] &= \frac{1}{P(a^j | x^m)} \int_{-\infty}^{\infty} \varepsilon^j \prod_{h \neq j} g(\varepsilon^j + v^j - v^h) d\mathbf{g}(\varepsilon^j) \\
 &= \frac{1}{P(a^j | x^m)} \int_{-\infty}^{\infty} \varepsilon^j \prod_{h \neq j} \exp\left\{-\exp\left[-(\varepsilon^j + v^j - v^h) + \gamma\right]\right\} d\mathbf{g}(\varepsilon^j) \\
 &= \frac{1}{P(a^j | x^m)} \int_{-\infty}^{\infty} \varepsilon^j \exp\left[-Q \exp(-\varepsilon^j + \gamma)\right] \cdot \exp(-\varepsilon^j + \gamma) d\varepsilon^j \quad (43)
 \end{aligned}$$

となる。ただし、 $Q = \sum_{h=1}^J \exp[-(v^j - v^h)]$ である。これを解くために、 $y \equiv -Q \exp(-\varepsilon^j + \gamma)$

として置換積分を行う。すると、

$$\begin{aligned}
 &\int_{-\infty}^{\infty} \varepsilon^j \exp\left[-Q \exp(-\varepsilon^j + \gamma)\right] \cdot \exp(-\varepsilon^j + \gamma) d\varepsilon^j \\
 &= \int_{\infty}^0 \left[\gamma - \ln\left(\frac{y}{Q}\right)\right] \exp(-y) \left(\frac{y}{Q}\right) \left(-\frac{1}{y}\right) dy \\
 &= \frac{1}{Q} \int_0^{\infty} (\ln Q - \ln y + \gamma) \exp(-y) dy \\
 &= \frac{1}{Q} \left\{ \int_0^{\infty} (\ln Q + \gamma) \exp(-y) dy - \int_0^{\infty} \ln y \exp(-y) dy \right\} \\
 &= \frac{1}{Q} \left[-(\ln Q + \gamma) \exp(-y) \right]_0^{\infty} - \frac{\gamma}{Q} \\
 &= \frac{1}{Q} (\ln Q + \gamma) - \frac{\gamma}{Q} \\
 &= \frac{\ln Q}{Q}. \quad (44)
 \end{aligned}$$

ここで、 $Q = 1/P(a^j | x^m)$ であるので、

$$E[\varepsilon^j | x^m, \delta(x^m, \varepsilon^j) = a^j] = \frac{1}{P(a^j | x^m)} \frac{\ln(1/P(a^j | x^m))}{1/P(a^j | x^m)} = \ln\left(\frac{1}{P(a^j | x^m)}\right) = \ln(1) - \ln P(a^j | x^m)$$

となり、目的の答えが得られる。(なお、極値分布が $g(\varepsilon_{it}) = \exp[-\exp(-\varepsilon_{it})]$ である場合は、 $\gamma - \ln P(a^j | x^m)$ となる。)

[3] 積分価値関数に関しては、 ε_{it} が (14) 式にしたがうとき、(8) 式の“積分価値関数版”ベルマン方程式を計算すればよい。まず、 w を次のように定義する。

$$w \equiv \max_{j=1, \dots, J} \{v^j + \varepsilon_{it}\}. \quad (45)$$

ただし、 $v^j \equiv u(x^m, a^j) + \beta \mathbf{F}(x^m, a^j) \bar{V}$ である。 ε_{it} は仮定 CI の下で IID であり、(14) 式にしたがうので、この w の確率分布は、

$$\begin{aligned} G(w) &\equiv \prod_{j=1}^J g(w - v^j) \\ &= \prod_{j=1}^J \exp\left\{-\exp[-(w - v^j) + \gamma]\right\} \\ &= \exp\left\{-\exp(-w + \gamma) \sum_{j=1}^J \exp(v^j)\right\} \\ &= \exp\left\{-D \exp(-w + \gamma)\right\}. \end{aligned} \quad (46)$$

その密度関数は、

$$G'(w) = D \exp(-w + \gamma) \exp(-D \exp(-w + \gamma)) \quad (47)$$

となる。ただし、 $D \equiv \sum_{j=1}^J \exp(v^j)$ 。求めたい積分価値関数は、 $\bar{V}(x^m) = \int w dG(w)$ となることに注目しよう。つまり、

$$\bar{V}(x^m) \equiv \int_{-\infty}^{\infty} w D \exp(-w + \gamma) \exp(-D \exp(-w + \gamma)) dw. \quad (48)$$

これを解くために、 $z \equiv D \exp(-w + \gamma)$ として置換積分を行う。すると、

$$\begin{aligned} \bar{V}(x^m) &\equiv \int_{\infty}^0 \left[\gamma - \ln\left(\frac{z}{D}\right) \right] z \exp(-z) \times \left(\frac{1}{z}\right) dz \\ &= \int_0^{\infty} (\ln D - \ln z + \gamma) \exp(-z) dz. \\ &= \int_0^{\infty} (\ln D + \gamma) \exp(-z) dz - \int_0^{\infty} \ln z \exp(-z) dz \\ &= \left| -(\ln D + \gamma) \exp(-z) \right|_0^{\infty} - \gamma \\ &= \ln D + \gamma - \gamma \\ &= \ln \sum_{j=1}^J \exp(v^j). \end{aligned} \quad (49)$$

v^j の定義より、(17) 式が得られる。（極値分布が $g(\varepsilon_{it}) = \exp[-\exp(\varepsilon_{it})]$ である場合は、 $\ln \sum_{j=1}^J \exp(v^j) + \gamma$ となる。）

付録 B ニュートン法と準ニュートン法について

ここでは、制約なし最大化問題を解く方法について解説する³¹。まず、アルゴリズムのイメージをつかむために、ある微分可能な関数 $f: \mathbb{R} \rightarrow \mathbb{R}$ を考え、 $f(x)=0$ となる x^* を探索したいとする。関数 f を $x^{(k)}$ の近くで一次テイラー展開すると

$$f(x) = f(x^{(k)}) = f'(x^{(k)})(x - x^{(k)}) \quad (50)$$

と $f(x)=0$ より近似式

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \quad (51)$$

が得られる。適当な $x^{(0)}$ から始めて、 $k=0, 1, \dots$ に対して (51) の探索ルールを適用すると、 $x^{(0)}, x^{(1)}, \dots$ はある値に収束する。収束した値が求めるべき x^* である。ここで、 f が 2 回連続的微分可能であり、 $f'(x)=0$ となる値をこのやり方で探索すれば、 $f(x)$ の局所的な最大点または最小点を求めることができる。

いま、2 回連続的微分可能な関数 $\ell: \mathbb{R}^N \rightarrow \mathbb{R}$; $\ell(\boldsymbol{\theta})$, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)'$ の最大点を探索したいとしよう。2 次のテイラー展開より、探索ルールは次のような式であらわされる。

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \left(\nabla^2 \ell(\boldsymbol{\theta}^{(k)}) \right)^{-1} \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \boldsymbol{\theta}} \quad (52)$$

ここで、 $\nabla^2 \ell(\boldsymbol{\theta}^{(k)})$ はヘッセ行列、 $\partial \ell(\boldsymbol{\theta}^{(k)}) / \partial \boldsymbol{\theta}$ は勾配ベクトルである。つまり、

$$\nabla^2 \ell(\boldsymbol{\theta}^{(k)}) = \begin{bmatrix} \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_1^2} & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_1 \partial \theta_2} & \dots & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_2^2} & \dots & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_2 \partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_n \partial \theta_2} & \dots & \frac{\partial^2 \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_n^2} \end{bmatrix}, \quad \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_1} \\ \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_2} \\ \vdots \\ \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \theta_n} \end{bmatrix}$$

このようにして、最大点 $\hat{\boldsymbol{\theta}}$ を探索していく方法をニュートン法と呼ぶ。

ニュートン法は概念的にわかりやすい探索ルールであるが、大域的な収束が保証されず、計算コストが大きいという欠点がある。そこで、一般的には次のような準ニュートン法が用いられることも多い。準ニュートン法では、ヘッセ行列の代わりにある正定値対称行列 $\mathbf{B}^{(k)}$ を用いて次のような探索ルールを考える。

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - (\mathbf{B}^{(k)})^{-1} \frac{\partial \ell(\boldsymbol{\theta}^{(k)})}{\partial \boldsymbol{\theta}}. \quad (53)$$

31 詳しくは福島 (1996) などを参照。

ここで、 $\mathbf{s}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}$, $\mathbf{y}^{(k)} = (\partial \ell(\boldsymbol{\theta}^{(k+1)}) / \partial \boldsymbol{\theta}) - (\partial \ell(\boldsymbol{\theta}^{(k)}) / \partial \boldsymbol{\theta})$ とすると、 $\mathbf{B}^{(k)}$ は $\mathbf{y}^{(k)} = \mathbf{B}^{(k)} \mathbf{s}^{(k)}$ という条件を満たさなければならない（これをセカント条件と呼ぶ）。実際、 $\mathbf{B}^{(k)}$ は次のような BFGS の公式によって更新していけばよい。

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \frac{\mathbf{y}^{(k)} (\mathbf{y}^{(k)})'}{(\mathbf{y}^{(k)})' \mathbf{s}^{(k)}} - \frac{\mathbf{B}^{(k)} \mathbf{s}^{(k)} (\mathbf{s}^{(k)})' \mathbf{B}^{(k)}}{(\mathbf{s}^{(k)})' \mathbf{B}^{(k)} \mathbf{s}^{(k)}}. \quad (54)$$

つまり、準ニュートン法は、ニュートン法のようにヘッセ行列を逐次計算していく必要がなく、 $\boldsymbol{\theta}^{(k)}$ と同時に $\mathbf{B}^{(k)}$ も更新していくことで探索していく方法である。さらに、それでも $\mathbf{B}^{(k)}$ の逆行列を計算する手間がかかるので、 $\mathbf{H}^{(k)} = (\mathbf{B}^{(k)})^{-1}$ を更新するルールを用いることができる。その更新ルールは、

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \left(1 + \frac{(\mathbf{y}^{(k)})' \mathbf{H}^{(k)} \mathbf{y}^{(k)}}{(\mathbf{y}^{(k)})' \mathbf{s}^{(k)}} \right) \frac{\mathbf{s}^{(k)} (\mathbf{s}^{(k)})'}{(\mathbf{y}^{(k)})' \mathbf{s}^{(k)}} - \frac{\mathbf{H}^{(k)} \mathbf{y}^{(k)} (\mathbf{s}^{(k)})' + \mathbf{s}^{(k)} (\mathbf{y}^{(k)})' (\mathbf{H}^{(k)})'}{(\mathbf{y}^{(k)})' \mathbf{s}^{(k)}} \quad (55)$$

となることが知られている。本稿の (21) 式の $\mathbf{H}^{(k)}$ は、 $(\nabla^2 \ell(\boldsymbol{\theta}^{(k)}))^{-1}$, $(\mathbf{B}^{(k)})^{-1}$, または上の $\mathbf{H}^{(k)}$ を用いればよい。

付録 C プログラムコード

ここでは、第 4 節で推定のために作成したプログラム（MATLAB コード）の一部を掲載する。

NFXP.m

```
% NFXP アルゴリズム

clear

beta = 0.975;

% 真のパラメーターの設定
RC = 11.7257; theta11 = 2.4569; theta30 = 0.0937; theta31 = 0.4475;
theta32 = 0.4459; theta33 = 0.0127; theta34 = 0.0002;

% パラメーターの真の値
THETA_TRUE = [RC, theta11, theta30, theta31, theta32, theta33, theta34]';

M = 175; % X 空間の次元
XMAX = 5000; % 最大マイル数

N = 5; % バスの数
T = 120; % 期間

% パラメーターの真の値を使ってデータを生成する。
[x, a] = func_data(M, XMAX, N, T, THETA_TRUE, beta);

%% パラメーターの初期値
RC = 12; theta11 = 3;
THETA3 = func_theta3(M, XMAX, N, T, x, a); % THETA3 は先に推定しておく。
THETA(:, 1) = [RC, theta11, THETA3']';

%% Outer-loop (パラメーターの更新)

tic % 計算時間の計測開始
```

```

kMAX = 200; % Outer-loop の最大計算回数
logL = ones(1, kMAX)*(-1e+12); % 対数尤度の初期値

for k = 1:kMAX

    fprintf('NFXP:    k = %d,    logL = %f\n', k, logL(k));

    %% Inner-loop (不動点を求める.)

    % P0 の初期値
    P0(:, 1) = ones(M, 1)*0.6;

    iterMAX = 100; % Inner-loop の最大計算回数
    ERROR = 1e-12; % 収束を終了させる値
    X = linspace(0, XMAX, M)'; % 0 から XMAX までを M 個に分割したもの

    for iter = 1:iterMAX

        V(:, iter) = func_Phi(X, M, P0(:, iter), THETA(:, k), beta); % 関数Φ
        P0(:, iter+1) = func_Lambda(X, M, V(:, iter), THETA(:, k), beta); % 関数Λ

        if (pdist([P0(:, iter), P0(:, iter+1)]')) < ERROR
            break;
        end

    end

    %% Inner-loop 終わり

    % 勾配ベクトルとヘッセ行列の計算
    Vbar = V(:, iter);
    GRAD = func_grad(M, XMAX, N, T, x, a, Vbar, THETA(:, k), beta);
    H = func_Hessian(M, XMAX, N, T, x, Vbar, THETA(:, k), beta);

    % パラメータの更新
    THETA(1:2, k+1) = THETA(1:2, k) - H\GRAD*0.02;
    THETA(3:7, k+1) = THETA(3:7, k); % theta3 はそのまま更新

    % 対数尤度の計算
    logL(k+1) = func_logL(M, XMAX, N, T, x, a, Vbar, THETA(:, k+1), beta);

    % 対数尤度が下がったら終了する.
    if logL(k+1) < logL(k)
        break;
    end

end

% Outer-loop 終わり

[~, maxk] = max(logL); % 最大対数尤度
RESULT = THETA(:, maxk);
TIME = toc; % 計算時間

```

func_data.m

```

function [x, a] = func_data(M, XMAX, N, T, THETA, beta)

theta30 = THETA(3);
theta31 = THETA(4);
theta32 = THETA(5);
theta33 = THETA(6);

%% theta の初期値のもとで不動点を求める.

```

```

X = linspace(0, XMAX, M)';    % 0 から XMAX までを M 個に分割したもの

% との初期値POP1
PO(:, 1) = ones(M, 1)*0.6;
P1(:, 1) = 1 - PO(:, 1);

iterMAX = 30; % 反復計算の最大回数
ERROR = 1e-12; % 収束を終了させる値

V = zeros(M, iterMAX);

for iter = 1:iterMAX

    V(:, iter) = func_Phi(X, M, PO(:, iter), THETA, beta); % 関数Φ
    PO(:, iter+1) = func_Lambda(X, M, V(:, iter), THETA, beta); % 関数Λ

    if (pdist([PO(:, iter), PO(:, iter+1)]') < ERROR)
        break
    end

end

P1 = 1 - PO(:, iter+1);

%% データの生成

Lv_x = ones(N, T);
a = zeros(N, T);

for t = 1:T-1
    for i = 1:N

        % データ a
        if rand < P1(Lv_x(i, t)) % 交換確率より乱数が小さければ交換する.
            a(i, t) = 1;
        end

        if a(i, t) == 0

            r = rand;

            if Lv_x(i, t) < M-3

                if r < theta30
                    Lv_x(i, t+1) = Lv_x(i, t) + 0;
                elseif r < theta30 + theta31
                    Lv_x(i, t+1) = Lv_x(i, t) + 1;
                elseif r < theta30 + theta31 + theta32
                    Lv_x(i, t+1) = Lv_x(i, t) + 2;
                elseif r < theta30 + theta31 + theta32 + theta33
                    Lv_x(i, t+1) = Lv_x(i, t) + 3;
                else
                    Lv_x(i, t+1) = Lv_x(i, t) + 4;
                end

            elseif Lv_x(i, t) == M-3

                if r < theta30/(theta30 + theta31 + theta32 + theta33)
                    Lv_x(i, t+1) = Lv_x(i, t) + 0;
                elseif r < (theta30 + theta31)/(theta30 + theta31 + theta32 + theta33)
                    Lv_x(i, t+1) = Lv_x(i, t) + 1;
                elseif r < (theta30 + theta31 + theta32)/(theta30 + theta31 + theta32 + theta33)
                    Lv_x(i, t+1) = Lv_x(i, t) + 2;
            end
        end
    end
end

```



```

    else
        Lv_x(i, t+1) = Lv_x(i, t) + 3;
    end

    elseif Lv_x(i, t) == M-2

        if r < theta30/(theta30 + theta31 + theta32)
            Lv_x(i, t+1) = Lv_x(i, t) + 0;
        elseif r < (theta30 + theta31)/(theta30 + theta31 + theta32)
            Lv_x(i, t+1) = Lv_x(i, t) + 1;
        else
            Lv_x(i, t+1) = Lv_x(i, t) + 2;
        end

    elseif Lv_x(i, t) == M-1

        if r < theta30/(theta30 + theta31)
            Lv_x(i, t+1) = Lv_x(i, t) + 0;
        else
            Lv_x(i, t+1) = Lv_x(i, t) + 1;
        end

    elseif Lv_x(i, t) == M

        Lv_x(i, t+1) = Lv_x(i, t) + 0;

    end

else % a(t) == 1

    r = rand;

    if r < theta30
        Lv_x(i, t+1) = 1;
    elseif r < theta30 + theta31
        Lv_x(i, t+1) = 2;
    elseif r < theta30 + theta31 + theta32
        Lv_x(i, t+1) = 3;
    elseif r < theta30 + theta31 + theta32 + theta33
        Lv_x(i, t+1) = 4;
    else
        Lv_x(i, t+1) = 5;
    end

end

end

end

% データ x
x = X(Lv_x);

end

```

func_Phi.m

```

function V = func_Phi(X, M, P0, THETA, beta)

% 推移確率行列
FO = func_F0(M, THETA);
F1 = func_F1(M, THETA);

```

```
% 効用関数
u0 = - 0.001*THETA(2)*X;
u1 = - THETA(1)*ones(M, 1);

% P0*F0, P1*の計算F1
PF = zeros(M, M);
for i = 1:M
    PF(i, :) = P0(i).*F0(i, :);
end
POF0 = PF;

PF = zeros(M, M);
for i = 1:M
    PF(i, :) = P1(i).*F1(i, :);
end
P1F1 = PF;

V = (eye(M) - beta*(POF0 + P1F1))\((P0.*(u0 - log(P0)) + P1.*(u1 - log(P1)));

end
```

func_Lambda.m

```
function P0 = func_Lambda(X, M, V, THETA, beta)

% 推移確率行列
F0 = func_F0(M, THETA);
F1 = func_F1(M, THETA);

% 効用関数
u0 = - 0.001*THETA(2)*X;
u1 = - THETA(1)*ones(M, 1);

R0 = u0 + beta*F0*V;
R1 = u1 + beta*F1*V;

P0 = exp(R0)./(exp(R0) + exp(R1));

end
```