〈研究ノート〉

動的離散選択モデルの構造推定 2（サーベイ論文）

── 動学ゲーム ──

楠田　康之*

要　旨

本稿では，動学ゲームについて，理論モデルの枠組み，推定アルゴリズム，シミュレーションについて説明する．まず，動学ゲームの基本モデルとして参入・退出をともなう寡占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と，いくつかの代替的なアルゴリズムを説明する．それらのアルゴリズムを使って，モンテカルロ法によって生成されたデータより構造推定を行なって比較する．さらに，応用例として売却利益と参入費用をともなう参入・退出モデルを解説して推定を行なう．

キーワード：動的離散選択モデル，動学ゲーム，マルコフ完全均衡（MPE），EP フレームワーク，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

楠田（2018）では，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選択モデルの構造推定について概観した．本稿では，それに続く議論として，動学ゲームにおける理論モデルの枠組み，推定アルゴリズム，シミュレーションについて説明する．シングルエージェントのモデルが労働経済学などで広く用いられるのに対し，動学ゲームはおもに産業組織論（特に寡占市場）の分析に役立つ．そこで，まずこの節では，動学ゲームの特徴を明らかにし，そこで用いられる均衡概念を考え，産業組織論における先行研究を取り上げながら，モデルの枠組みと推定アルゴリズムの発展について簡単にまとめる．

1. 1　シングルエージェントモデルから動学ゲームへ

動学ゲームとは

多くの寡占市場は，複数の企業（プレイヤー）が長期的な意味で相互に関係している．そこで

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

は，価格や生産量のような選択（行動）だけではなく，いったん選ぶと容易に変更することができず，長期的な意味で市場や企業の状態に影響を与えるような企業の選択が存在する．費用をともなう参入と退出，または，市場の状態に影響を与えるような投資がそのような選択である．そこで，長期の寡占市場モデルを，動的離散選択モデルとして考える場合，「長期的な選択」（参入・退出，投資など）と「短期的な選択」（価格，生産量など）の 2 つを分けて考えなければならない．このような各企業の選択は，市場の変化や競争関係を通じて，他の企業の利潤や選択にも影響するので，各企業間の関係は，動的に戦略的な相互依存関係を持っている．その意味で，このような寡占市場モデルは，**動学ゲーム（dynamic game）**として分析される．ここでの動学ゲームとは，価格・生産量競争ゲームを各期に行なう繰り返しゲームである．

マルコフ完全均衡（MPE）

そのような動学ゲームにおいて考えられる均衡概念は，Maskin and Tirole（1988a, 1988b）によって提唱された**マルコフ完全均衡（Markov Perfect Equilibrium, MPE）**である．MPE では，各プレイヤーの行動が利得にかかわる（payoff-relevant）状態のみの関数（**マルコフ戦略**）になっていると仮定されており，それによって，膨大にありうる部分ゲーム完全均衡を排除することができる．MPE においては，(1) すべてのプレイヤーが，他のプレイヤーの現在および将来における行動に関して予測して信念を形成し，その信念のもとで利得の割引価値総和の期待値を最大化し，(2) すべてのプレイヤーが形成する信念が，他のプレイヤーのとる行動と整合的となっていなければならない．具体的には，すべてのプレイヤーは，各期において，他のプレイヤーの戦略を所与としてベルマン方程式を設定して DP 問題を解き，それにより選択された行動と状態によって，次の状態が実現する．そのような過程において，特定の期に依存せずに，状態によって決まる行動は，他のプレイヤーの戦略に対する最適反応となっている．マルコフ完全均衡とは，そのようなプレイヤーの戦略が互いに最適反応となっていることを言うのである．（本稿では，Aguirregabiria and Mira, 2007 にしたがって，条件付き選択確率を用いた MPE を使うが，第 3. 2 節において，マルコフ戦略を用いた定義にも触れる．）

1. 2　産業組織論における動学ゲーム

EP フレームワーク（PM フレームワーク）

産業組織論の中で，動学ゲームおよび MPE を寡占市場の分析に用いることは，Ericson and Pakes（1995）までさかのぼることができる（以下，EP）．そこで設定された枠組み（フレームワーク）とは，長期的な寡占市場モデルにおいて，市場の中にいる既存企業が，各期に操業を継続するか，操業を停止して市場から退出するかのどちらかを選択し，かつ，潜在的にその市場に参入を考えている企業がいて，各期に参入するかしないかのどちらかを選択する．さらに，操業している企業はある種の投資を行なうことで，市場および自分の「状態」を変えることができる．このような枠組みの中で，各企業は最適な選択（戦略）をとり，それが繰り返されることで，均

衡における市場の企業数や，市場の状態および各企業の状態が決定する．EP は，このような枠組みにおいて長期的な寡占市場モデルの MPE を考察した．この枠組みは，その後の理論・実証モデルの基本となるもので，**Ericson and Pakes フレームワーク（EP フレームワーク）**などと呼ばれている（Doraszelski and Pakes, 2007）．さらに，このモデルは分析上とても複雑なので，Pakes and McGuire（1994）は，参入・退出を純粋戦略に限定した上で，より計算を容易にしたモデルを提示した．（そのため，EP フレームワークは **PM フレームワーク**とも呼ばれる．）ところが，離散的な戦略である参入・退出を純粋戦略に限定すると，モデルの中の MPE の存在が保証されないことが知られるようになった．その問題に対して，Doraszelski and Satterthwaite（2005, 2010）は，参入・退出を混合戦略として考える代わりに，企業が退出の際に得られる利益（売却利益）と参入費用を確率変数とし，その企業以外に観察できない私的情報であると仮定して，モデルを**不完備情報（incomplete information）**のゲームとして設定することを提案した[1]．その結果，モデルには必ず MPE が存在することが証明され，また，推定も容易になることが示されている[2]．なお，本稿の第 5 節で解説するモデルでは，売却利益と参入費用を私的情報としている[3]．

動学ゲームの推定アルゴリズムの発展

動学ゲームの構造推定を行なう場合，大きな問題は計算の複雑さと時間である．楠田（2018）で説明した「ネステッド不動点アルゴリズム」を使うと長い時間を要するので，計算が容易で一致推定量を得られるような推定法が提案されてきた．Bajari, Benkard, and Levin（2007），Pakes, Ostrovsky, and Berry（2007），Pesendofer and Schmidt-Dengler（2008）などは，「2 ステップアルゴリズム」と総称されるさまざまな推定法を提案している．これらの手法は計算が容易であり，DP 問題を解く必要がないが，少ないサンプルより推定する場合はバイアスが大きくなるという欠点がある．Aguirregabiria and Mira（2007）は，「ネステッド疑似尤度アルゴリズム（NPL）」を動学ゲームにも適用し，参入・退出モデルを推定している．ところが，NPL は収束しない場合があるということが Pesendofer and Schmidt-Dengler（2010）によって示された．Kasahara and Shimotsu（2012）は，この NPL の欠点を補うために，NPL を修正したアルゴリズムを提示している．さらに，Egesdal, Lai, and Su（2015）は，2 ステップや NPL に変わる推定法として，別のアルゴリズムを提唱している．これは，楠田（2018）で説明した「均

1　ここで，完備情報ゲームとは，すべてのプレイヤーがゲームの構造（利得，行動）を知っているゲームであり，そうでなければ不完備情報ゲームである．売却利益や参入費用を各企業の「タイプ」と考えれば，それは他の企業に観察できない私的情報であるので，このモデルは不完備情報ゲームである．

2　さらに，Doraszelski and Satterthwaite（2005, 2010）は，連続型の選択の場合でも，企業の投資がある条件（unique investment choice, UIC）を満たせば，MPE の存在が保証されることを示している．

3　ただし，このモデルにおいて企業の長期的な選択は参入・退出のみであり，その他の投資は考えられていない．

衡制約付き最適化アルゴリズム（MPEC）」を動学ゲームに適用し，それにより均衡に関する情報を完全に用いることによって最尤推定量を求めるものである．

産業組織論での応用例

参入・退出をともなう寡占市場の動学的な研究は，古くから産業組織論の中で大きな関心を集めていた．EP 以前の代表的な実証研究として，Bresnahan and Reiss（1990, 1991a, 1991b）と Berry（1992）がある．これらの研究は，参入阻止ゲームにもとづいて戦略的参入を可変利潤と既存企業の数に関連付けた研究の嚆矢であり，その議論の中にすでに複数均衡の問題を含んでいる．参入の意思決定は，可変利潤や既存企業数によって決定され，実証分析ではそれらの「しきい値」が推定されている．

EP 以後の理論分析としては，企業合併（merger）を長期的な戦略として分析した Gowrisankaran（1999），病院の参入・退出，投資の Gowrisankaran and Town（1997），生産能力拡大のための投資の Besanko and Doraszelski（2004），Chen（2009），広告投資の Doraszelski and Markovich（2007）などがある．また，EP 以後の実証分析としては，ハンバーガーチェーンの参入モデルを分析した Toivanen and Waterson（2000），航空産業の Benkard（2004），ネットスケープとマイクロソフト間の“ブラウザ戦争”の Jenkins, Liu, Matzkin, and McFadden（2004），レンタルビデオ店の Seim（2006），沖縄のコンビニエンスストアの Nishida（2009）などがある．ただし，これらの中には Bresnahan and Reiss にもとづいた静学的な研究と，動学ゲームと MPE にもとづいた研究が存在する．

最近では，Aguirregabiria and Ho（2012）が航空産業を推定して，複数均衡の問題を論じている．Fan（2016）は，Pakes et al.（2007）の推定アルゴリズムにもとづき，レンタルビデオ店の参入・退出モデルを推定した．さらに，Luo, Xiao, and Xiao（2018）は，ファーストフードチェーン市場（KFC とマクドナルド）の推定を行ない，複数均衡と市場の異質的な観察されない状態を含むゲームの推定を提示した[4]．

本稿の目的

楠田（2018）のシングルエージェントの動的離散選択モデルでは，NFXP と NPL の 2 つのアルゴリズムに焦点を当てたが，本稿では，動学ゲームの構造推定を行なうためのアルゴリズムとして，NPL を含めた 4 つのアルゴリズム（NPL，2STEP，NPL-Λ，MPEC）について検討し，その導出を行ない，それらのアルゴリズムを用いた推定の実際について説明する．シングルエージェントの場合と同じく，それらの導出において重要なポイントは，条件付き選択確率（CCP）を用いるということである．つまり，シングルエージェントの場合は，条件付き選択確率による

4　また，産業組織論の分野ではないが，松村など（2012）は，動学ゲームモデルを用いて鉄道混雑メカニズムを推定している興味深い研究である．

最適化問題の解を考えたが，本稿の動学ゲームの場合は，ゲームの均衡解を条件付き選択確率によって表現する．シングルエージェントの場合と違う点は，その条件付き選択確率が他のプレイヤーの戦略をもあらわしているということである．したがって，ある条件付き選択確率からある条件付き選択確率へ写す写像の不動点は，ゲームの均衡点と考えることができる．

本稿の基本モデルは，ほぼ Aguirregabiria and Mira（2007）に依拠したが，Aguirregabiria and Mira（2010）と Egesdal et al.（2015）などにも多くを負っている．条件付き選択確率を用いた均衡条件の導出と推定はとても有用なものである一方，若干の欠点も含んでいる．そこで，その NPL より議論を始め，その拡張として他のアルゴリズムを考えていく．さらに，Pakes et al.（2007）によって別のモデルを考え，代替的なアルゴリズムの説明も行なう．本稿では，いくつかのアルゴリズムを説明する上で，それぞれの特色を強調することにつとめた．（ただし，それらに対する理解不足や誤解などがあれば，すべて筆者の責任である．）

なお，推定の数値計算のプログラミングに際して，Python（パイソン）を用いたことが本稿の新しい試みである．Python のもっとも大きな利点は，オープンソースであり無償で使えるということであろう．さらに，昨今のモジュール（Python から特定の用途に応じて使えるようにされたプログラム群）の充実により，数値計算やデータ処理が容易になってきており，プログラマー人口の増大とともに学習がしやすいプログラミング言語であるといえよう．これらのことにより，構造推定に関する研究の参入障壁を下げる上で大きな役割が期待される．なお，この分野の先行研究でも用いられる AMPL（アンプル）と KNITRO（ナイトロ）についても適宜説明した．

本稿の構成は以下の通りである．第 2 節では，動学ゲームの基本モデルについて説明し，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．第 3.1 節でネステッド疑似尤度アルゴリズム，第 3.2 節で 2 ステップアルゴリズム，第 3.3 節で修正ネステッド疑似尤度アルゴリズム，第 3.4 節で均衡制約付き最適化アルゴリズムを解説する．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，そのデータよりいくつかのアルゴリズムを使って構造推定を行なう．第 5 節では，応用例として売却利益と参入費用をともなう参入・退出モデルを解説し，推定を行なう．最後に，第 6 節をまとめとする．

2　基本モデル

この節では，Aguirregabiria and Mira（2007）のモデルにしたがい，動学ゲーム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，MPE を定義する．さらに，そのモデルを推定するための準備として，3 つの関数を導出する．

2.1　動的ゲームの基本設定

基本となる動学ゲームを，以下のように設定しよう．シングルエージェントの動的離散選択モ

デルと同様に，期間を t であらわし，意思決定を行なう**企業（プレイヤー）**を $i \in I \equiv \{1, 2, \ldots, N\}$ であらわす．各企業は各期に**状態**を観察した上で，**行動** $a_{it} \in A \equiv \{a^0, a^1, \ldots, a^{|A|-1}\}$，$|A| < \infty$ を選択するとする．この状態を，誰にでも観察できる x_t（**共有知識**）と企業 i にしか観察できない ε_{it}（**私的情報**）の 2 つに分けよう．一般に，状態 x_t は市場の状態（**需要シフター**，操業中の企業数など）や各企業の状態（店舗数，操業年数，規模，市場シェアなど）を含む．また，$\varepsilon_t \equiv (\varepsilon_{1t}, \varepsilon_{2t}, \ldots, \varepsilon_{Nt})$ は各企業固有の需要シフターや費用などである．次に，企業 i が t 期に得られる利潤は，状態 x_t および ε_{it} とすべての企業の行動 $a_t \equiv (a_{1t}, a_{2t}, \ldots, a_{Nt})$ に依存して決まるとしよう[5]．この利潤を $\Pi_i(x_t, \varepsilon_{it}, a_t)$ と書くと，企業 i は次のような利潤の割引総和の条件付き期待値を最大化するように，行動を選択する．

$$E\left(\sum_{\tau=0}^{\infty} \beta^{\tau} \Pi_i(x_{t+\tau}, \varepsilon_{i,t+\tau}, a_{t+\tau}) \,|\, x_t, \varepsilon_{it} \right). \tag{1}$$

ただし，$\beta \in (0, 1)$ は期間を通じて不変ですべての企業に共通な割引因子である．ここで，状態 (x_t, ε_t) の推移に関する企業の主観的確率（以下，**信念**）は，すべての企業に共通な**推移確率** $p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, a_t)$ にしたがうとする．この推移確率は共有知識である．つまり，すべての企業は，他の企業がこの推移確率にしたがって行動することを知っていると仮定する．シングルエージェントの動的離散選択モデルと同様に，推定したい構造パラメーターは構成要素（Π, p, β）を決定するパラメーターである．そこで，この動学ゲームのモデルでも次のような標準的な仮定をもうけよう．

仮定 AS「加法分離性」（Additive Separability）

利潤関数 Π_i は次のように 2 つの部分に分けられる．

$$\Pi_i(x_t, \varepsilon_{it}, a_t) = \pi_i(x_t, a_t) + \varepsilon_{it}(a_{it}). \tag{2}$$

ここで，ε_{it} は企業 i の選択する行動 $a_{it} \in A \equiv \{a^0, a^1, \ldots, a^{|A|-1}\}$ に対応して生じる $(|A| \times 1)$ 次元のベクトルであり，$a_{it} = a^j$ ならば $\varepsilon_{it}(a_{it})$ はその $(j+1)$ 番目の要素である．

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

私的情報 ε_{it} は i, t に関して互いに独立で同一の確率分布 $g(\varepsilon_{it})$ にしたがう[6]．さらに，推移確率は次のような積の形となる．

$$p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, a_t) = f(x_{t+1} | x_t, a_t) \cdot \prod_{i=1}^{N} g(\varepsilon_{i,t+1}). \tag{3}$$

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

5　シングルエージェントの場合と同様に，本稿でも，上付きの添字で「可能な値の候補」，下付きの添字で「プレイヤー」や「期」をあらわす．

6　本稿では，単純化のため，私的情報 ε_{it} は状態 x_t に依存しないと仮定する．

仮定 DIS「離散型の状態」（Discrete Support）

ある市場の状態 x_t のとる範囲（状態空間）は離散かつ有限である．つまり，ある $|X|<\infty$ が存在して，$x_t \in X \equiv \{x^1, x^2, \ldots, x^{|X|}\}$ と書ける．

つまり，利潤関数は状態によって加法的に分離され，推移確率は積の形で分離される．私的情報 ε_t は共有知識 x_t に影響せず，影響されることもない．また，私的情報は，期に関して独立に発生する．ここで，シングルエージェントの場合との違いは，各企業の利潤と推移確率がすべての企業の私的情報に依存していることである．これにより，ある企業の最適化問題は，他の企業が行動を選択する確率に依存していることになる．さらに，

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

私的情報 $\{\varepsilon_{it}(a),\ a \in A\}$ は互いに独立であり，それぞれタイプⅠ型極値分布

$$g(\epsilon) = \exp[-\exp(-\epsilon+\gamma)] \tag{4}$$

にしたがう．ただし，γ はオイラー定数である．

を追加しよう．シングルエージェントの問題で見たように，この仮定により，モデルは推定しやすいものとなる．

以上の基本設定のもとで，ベルマン方程式を定式化し，均衡戦略を定義する．

2.2　ベルマン方程式と均衡戦略

企業の戦略的関係を明確にするために，ここからしばらくは 2 プレイヤーモデルを考えよう．各企業は，状態を観察した上で，期に依存しない純粋戦略を持つとする[7]．すなわち，企業 i は t にかかわらず，状態 (x, ε_i) を観察すれば，1 つの行動 a_i を選択する．これを $\sigma_i(x_i, \varepsilon_i) = a_i$ であらわす．つまり，σ_i は定常的なマルコフ戦略である．

いま，戦略の組 $\sigma \equiv (\sigma_1, \sigma_2)$ を 1 つ固定し，各企業ともお互いこの戦略にしたがっていることを想定しているとしよう．この σ のもとで，企業 i の**条件付き選択確率**は次のように定義できる．

$$P_i(a_i|x) \equiv \Pr(\sigma_i(x, \varepsilon_i) = a_i|x) = \int I\{\sigma_i(x, \varepsilon_i) = a_i\}\, \mathrm{d}g(\varepsilon_i). \tag{5}$$

ここで，$I(\bullet)$ は指示関数である．

次に，企業の利潤と推移確率を定式化しよう．まず，企業 i は，相手の企業 j が戦略 σ_j にしたがっている前提のもとで行動 $a_j \in A$ を選択することを $P_j(a_j|x)$ の確率で予測するとしよう．すると，行動 a_1 を選択する企業 1 の期待利潤は，その相手企業の条件付き選択確率を条件とす

7　このため，以下では添字 t を省略する．

る期待値

$$\tilde{\pi}_1(x, a_1; \mathbf{P}_2) \equiv \sum_{a_2 \in A} P_2(a_2|x)\, \pi_1(x, a_1, a_2) \tag{6}$$

であらわすことができる．ただし，$\mathbf{P}_2$ は，$P_2(a_2|x)$ を要素とする行列とする．同様に，現在の状態が x であるとき，行動 a_1を選択する企業 1 が次の状態を x' と予測する期待推移確率は，やはり相手企業の条件付き選択確率を条件として次のように書ける．

$$\tilde{f}_1(x'|x, a_1; \mathbf{P}_2) \equiv \sum_{a_2 \in A} P_2(a_2|x)\, f(x'|x, a_1, a_2). \tag{7}$$

つまり，状態 x' が起きる確率は，自分の行動 a_1 だけでなく，相手の行動 a_2 にも依存しているが，企業 1 はその確率を $P_2(a_2|x)$ で評価する．すると，企業 1 が相手の条件付き確率が $\mathbf{P}_2$ であると予測するとき，状態 (x, ε_1) を観察した企業 1 の価値関数は次のようなベルマン方程式で定式化できる．

$$V_1(x, \varepsilon_1; \mathbf{P}_2) = \max_{a_1 \in A} \Big\{ \tilde{\pi}_1(x, a_1; \mathbf{P}_2) + \varepsilon_1(a_1) + \beta \sum_{x' \in X} \Big[\Big(\int V_1(x', \varepsilon_1'; \mathbf{P}_2) \mathrm{d}g(\varepsilon_1') \Big) \tilde{f}_1(x'|x, a_1; \mathbf{P}_2) \Big] \Big\} \tag{8}$$

すなわち，企業 1 の価値関数 V_1 は，相手の行動に関する期待利潤，期待推移確率，次の期の価値関数の 3 つを通じて，企業 2 の条件付き選択確率 $\mathbf{P}_2$ に依存していることになる．相手の条件付き確率を $\mathbf{P}_1$ と予測するときの企業 2 の価値関数 $V_2(x, \varepsilon_2; \mathbf{P}_1)$ も，同様に定式化できる．

ここで，シングルエージェントの場合と同様に，次のような**期待価値関数**を定義しよう．

$$\bar{V}_1(x; \mathbf{P}_2) \equiv \int \max_{a_1 \in A} \{ v_1(x, a_1; \mathbf{P}_2) + \varepsilon_1(a_1) \} \mathrm{d}g(\varepsilon_1). \tag{9}$$

ただし，$v_1(x, a_1; \mathbf{P}_2)$ は**選択価値関数**（choice-specific value function），つまり，

$$v_1(x, a_1; \mathbf{P}_2) \equiv \tilde{\pi}_1(x, a_1; \mathbf{P}_2) + \beta \sum_{x' \in X} \bar{V}_1(x'; \mathbf{P}_2)\, \tilde{f}_1(x'|x, a_1; \mathbf{P}_2) \tag{10}$$

である．つまり，この期待価値関数は，企業 i が自分の私的情報 ε_i を観察する前の事前の期待値をあらわす．さらに，$\bar{V}_1$ をすべての状態でまとめたベクトルを以下では $\bar{\mathbf{V}}_1$ であらわすことにする．

一般に，企業の数が N のときは，企業 $i \in I$ に対して，

$$\tilde{\pi}_i(x, a_i; \mathbf{P}_{-i}) \equiv \sum_{\mathbf{a}_{-i} \in A^{N-1}} \Big(\prod_{j \neq i} P_j(\mathbf{a}_{-i}[j]|x) \Big) \pi_1(x, a_i, \mathbf{a}_{-i}),$$

$$\tilde{f}_i(x'|x, a_i; \mathbf{P}_{-i}) \equiv \sum_{\mathbf{a}_{-i} \in A^{N-1}} \Big(\prod_{j \neq i} P_j(\mathbf{a}_{-i}[j]|x) \Big) f(x'|x, a_i, \mathbf{a}_{-i}) \tag{11}$$

とあらわすことができる．ただし，$\mathbf{a}_{-i}$ は i 以外の企業の行動のベクトル，つまり $\mathbf{a}_{-i} \equiv (a_j), j \neq i$ とし，$\mathbf{a}_{-i}[j]$ はその j 番目の要素とする．$\mathbf{P}_{-i}$ も同様である．(11) 式より，$v_i(x, a_i; \mathbf{P}_{-i})$ も

同様に定義できる．そこで，これを用いて，この動学ゲームの均衡を次のように定義しよう．

定義（マルコフ完全均衡，Markov Perfect Equilibrium, MPE）

すべての企業 $i\in I$ とすべての状態 (x, ε_i) について，戦略 $\sigma_i^*(x, \varepsilon_i)$ が次を満たすとき，$\boldsymbol{\sigma}^*\equiv\{\sigma_i^*(x, \varepsilon_i): i\in I\}$ を**マルコフ完全均衡**と呼ぶ．

$$\sigma_i^*(x, \varepsilon_i)=\underset{a_i\in A}{\operatorname{argmax}}\{v_i(x, a_i; \mathbf{P}^*_{-i})+\varepsilon_i(a_i)\}. \tag{12}$$

ただし，$P_j^*(a_j|x)\equiv\int I\{\sigma_j^*(x, \varepsilon_j)=a_j\}\mathrm{d}g(\varepsilon_j)$ とし，$\mathbf{P}^*_{-i}$ はそれをすべての $j\neq i$ に関してまとめたものである．

すなわち，この定義は，すべての企業に対して，「他の企業が $\boldsymbol{\sigma}^*$ にしたがっているとき，その行動を条件付き選択確率 $\mathbf{P}^*_{-i}$ で評価すれば，自分も $\boldsymbol{\sigma}^*$ にしたがうことが最適となっている」という意味である．ここで，MPE は，行動空間上で定義されているが（$A^N\to A^N$），これを条件付き選択確率の空間で定義することもできる．次節では，Aguirregabiria and Mira（2007）にしたがい，条件付き選択確率を用いた均衡を考える．

2.3　推定のための準備

基本モデルの説明の最後に，次節で行なう推定のための準備として，いくつかの式を求めよう．ここでの目的は，上で定義した戦略の組 $\sigma\equiv(\sigma_1, \sigma_2)$ の代わりに条件付き選択確率の組 $\mathbf{P}\equiv(\mathbf{P}_1, \mathbf{P}_2)$ を用いて均衡を表現することである．この $\mathbf{P}$ と，期待価値関数の組 $\bar{\mathbf{V}}\equiv(\bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2)$ を用いれば，推定のための式は 3 つの関数にまとめることができて，後で説明する推定のための各アルゴリズムは，それらの関数により容易に表現することができる．

まず，企業 2 が $\mathbf{P}_2$ をとるときの企業 1 の条件付き選択確率は次のようにあらわすことができる．

$$P_1(a_1|x)=\int I\left(a_1=\underset{a_1'\in A}{\operatorname{argmax}}\{v_1(x, a_1'; \mathbf{P}_2)+\varepsilon_1(a_1')\}\right)\mathrm{d}g(\varepsilon_1) \tag{13}$$

ここで，この右辺は $\bar{V}_1$ と $\mathbf{P}_2$ に依存していることに注目しよう．そこで，この右辺を $\bar{V}_1$ と $\mathbf{P}_2$ の関数として $\Lambda_1(a_1|x; \bar{V}_1, \mathbf{P}_2)$ と書くことにしよう．これは，「条件付き選択確率を用いた最適反応」に他ならない．企業 2 に関して $\Lambda_2(a_2|x; \bar{V}_2, \mathbf{P}_1)$ も同様に定義すると，それらをすべての {企業，状態，行動} でまとめた $\{\Lambda_i(a_i|x; \bar{V}_i, \mathbf{P}_j): i\in I, x\in X, a_i\in A\}$ は $\bar{\mathbf{V}}$ と $\mathbf{P}$ の関数と考えることができる．そこで，これを $\boldsymbol{\Lambda}(\bar{\mathbf{V}}, \mathbf{P})$ であらわす．ここで，シングルエージェントの場合との違いは，シングルエージェントの場合の $\boldsymbol{\Lambda}$ は $\bar{\mathbf{V}}$ のみの関数であったのに対し（楠田，2018, p.51），動学ゲームの場合の $\boldsymbol{\Lambda}$ は $\mathbf{P}$ の関数にもなっているということである．さらに，**仮定 CLOGIT** より，ε_1 がタイプ Ⅰ 型極値分布にしたがうとすれば，(13) 式の右辺は，

$$\Lambda_1(a_1|x;\bar{V}_1,\mathbf{P}_2)=\frac{\exp\{v_1(x,a_1;\mathbf{P}_2)\}}{\sum_{a'\in A}\exp\{v_1(x,a';\mathbf{P}_2)\}} \tag{14}$$

とあらわすことができる．（この導出については，楠田（2018）の付録 A を参照のこと．）

次に，(9) 式の期待価値関数を別の表現であらわそう．ある戦略 $\boldsymbol{\sigma}$ のもとで，企業 1 が自分の条件付き選択確率を $\mathbf{P}_1$ とし，相手の条件付き選択確率を $\mathbf{P}_2$ で予測するとしよう．すると，企業 1 の期待価値関数は

$$\bar{V}_1(x;\mathbf{P}_2)=\sum_{a_1\in A}P_1(a_1|x)\left[v_1(x,a_1;\mathbf{P}_2)+E\left(\varepsilon_1(a_1)|x_1,\sigma_1(x,\varepsilon_1)=a_1\right)\right] \tag{15}$$

と書くことができる．さらに，**仮定 CLOGIT** を仮定すれば，この式の［•］の第 2 項は $-\ln P_1(a_1|x)$ に等しくなる（楠田（2018）の付録 A を参照）ので，σ_1 を用いずに表現できることになる．ここで，(15) 式の右辺は $\bar{V}_1$ と $\mathbf{P}\equiv(\mathbf{P}_1,\mathbf{P}_2)$ に依存しているので，この右辺を $\bar{V}_1$ と $\mathbf{P}$ の関数として $\Phi_1(x;\bar{V}_1,\mathbf{P})$ であらわすことにしよう．$\Phi_2(x;\bar{V}_2,\mathbf{P})$ も同様に定義できるので，これをすべての｛企業，状態｝でまとめた $\{\Phi_i(x;\bar{V}_i,\mathbf{P}):i\in I,x\in X\}$ は，$\bar{\mathbf{V}}$ と $\mathbf{P}$ との関数と考えることができる．これを $\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P})$ であらわそう．

さらに，(15) 式は $\bar{V}_1$ に関して解くことができるので，シングルエージェントの場合と同様に，(15) 式をすべての状態でまとめて行列・ベクトルで表現して，式を整理すると，

$$\begin{aligned}\bar{\mathbf{V}}_1(\mathbf{P}_2)=&\left[\mathbf{I}_M-\beta\sum_{a_1\in A}\mathbf{P}_1(a_1)*\tilde{\mathbf{F}}(a_1;\mathbf{P}_2)\right]^{-1}\\&\times\sum_{a_1\in A}\mathbf{P}_1(a_1)*\{\tilde{\boldsymbol{\pi}}_1(a_1;\mathbf{P}_2)+\mathbf{E}(a_1;\mathbf{P}_1)\}\end{aligned} \tag{16}$$

と書ける．ただし，

$$\tilde{\mathbf{F}}(a_1;\mathbf{P}_2)=\begin{bmatrix}\tilde{f}_1(x^1|x^1,a_1;\mathbf{P}_2) & \cdots & \tilde{f}_1(x^{|X|}|x^1,a_1;\mathbf{P}_2)\\ \tilde{f}_1(x^1|x^2,a_1;\mathbf{P}_2) & \cdots & \tilde{f}_1(x^{|X|}|x^2,a_1;\mathbf{P}_2)\\ \vdots & \ddots & \vdots\\ \tilde{f}_1(x^1|x^{|X|},a_1;\mathbf{P}_2) & \cdots & \tilde{f}_1(x^{|X|}|x^{|X|},a_1;\mathbf{P}_2)\end{bmatrix},$$

$\tilde{\boldsymbol{\pi}}(a_1;\mathbf{P}_2)=\left(\tilde{\pi}_1(x^1,a_1;\mathbf{P}_2),\ldots,\tilde{\pi}_1(x^{|X|},a_1;\mathbf{P}_2)\right)'$，$\mathbf{E}(a_1;\mathbf{P}_1)=-\left(\ln P_1(a_1|x^1)),\ldots,\ln P_1(a_1|x^{|X|})\right)'$ である．ここで，(16) 式の右辺は $\mathbf{P}\equiv(\mathbf{P}_1,\mathbf{P}_2)$ に依存している．そこで，この右辺を $\mathbf{P}$ の関数として $\boldsymbol{\Gamma}_1(\mathbf{P})$ と書き，$\bar{\mathbf{V}}_2(\mathbf{P}_1)$ についても同様に表現できるので，$\boldsymbol{\Gamma}_i(\mathbf{P})$ をすべての企業でまとめた $\{\boldsymbol{\Gamma}_i(\mathbf{P}):i\in I\}$ を $\boldsymbol{\Gamma}(\mathbf{P})$ であらわすことにする．

いま，$(\bar{\mathbf{V}},\mathbf{P})$ の関数として $\boldsymbol{\Lambda}$ が，$\mathbf{P}$ の関数として $\boldsymbol{\Gamma}$ が得られている．そこで，$\boldsymbol{\Psi}(\bullet)\equiv\boldsymbol{\Lambda}(\boldsymbol{\Gamma}(\bullet),\bullet)$ と定義すれば，これは，$\mathbf{P}$ から $\mathbf{P}$ への関数となる．したがって，条件付き選択確率を用いた MPE は，$\mathbf{P}=\boldsymbol{\Psi}(\mathbf{P})$ を満たすような不動点としてあらわすことができる．

表 1　先行研究と推定アルゴリズム

研究分析	推定方法
Aguirregabiria and Mira（2007）	ネステッド疑似尤度アルゴリズム（NPL）
Bajari, Benkard, and Levin（2007）	2 ステップアルゴリズム（2STEP）
Pakes, Ostrovsky, and Berry（2007）	2 ステップアルゴリズム（2STEP）
Pesendofer and Schmidt-Dengler（2008）	2 ステップアルゴリズム（2STEP）
Kasahara and Shimotsu（2012）	修正されたネステッド疑似尤度アルゴリズム（NPL-Λ）
Egesdal, Lai, and Su（2015）	均衡制約付き最適化アルゴリズム（MPEC）[10]

3　推定アルゴリズム

ここからは，いくつかの推定アルゴリズムについて述べる．まず，動学ゲームの場合には，「次元の呪い」のために，**ネステッド不動点アルゴリズム（Nested Fixed Point Algorithm, NFXP）**を用いるのは一般的には困難であるとされてきた[8]．そこで，推定の正確さを犠牲にせずに，計算量や計算時間を短縮できるような代替的な推定アルゴリズムが提案されている．表 1 に，そのような代表的なアルゴリズムを記す[9]．

この研究分析の中で，Aguirregabiria and Mira（2007）（以下，AM2007）の推定アルゴリズムは，シングルエージェントと同様に，NXFP の外部ループと内部ループを交換して，条件付き選択確率と構造パラメーターを逐次的に更新していくアルゴリズムである．ところが，この推定アルゴリズムによって構造パラメーターが収束するという保証がないことがわかっている．Kasahara and Shimotsu（2012）（以下，KS2012）は，この欠点を修正し，不動点が得られるように推定アルゴリズムの改良を行った．2 ステップアルゴリズムとしては，楠田（2018）の 5.1 節で説明した Hotz and Miller の“逆写像法（inversion）”により，観察された条件付き選択確率から価値関数の差を求める方法と，観察された条件付き選択確率を一度だけ用いて価値関数を計算する方法が考えられる．Bajari et al.（2007）（以下，BBL2007）は前者の方法を用いている．Pakes et al.（2007）（以下，POB2007）は，各企業が参入・退出するモデルを二項分布によって定式化し，既存企業と参入企業の価値関数を計算して推定を行っている．一方，Pesendofer and Schmidt-Dengler（2008）は，漸近的な最小二乗推定量（asymptotic least squares estimator）を考え，上記のアルゴリズムによる推定量が，均衡条件に与えた「ウェイト」が異なるだ

8　ただし，Seim（2006）のように NFXP を用いた分析も存在する．また，現在では高速な計算が可能なソルバーも利用可能となってきている．本稿でも，NFXP による推定を試みる．

9　シングルエージェントの場合と同じく，これらの日本語名は便宜上のものである．

10　このアルゴリズムによる推定量は，（疑似ではない）尤度関数を最大化するので，MLE 推定量に他ならない．しかし，本稿では，他のアルゴリズムとの違いを強調するために，シングルエージェントの場合と同様に，このアルゴリズムを Mathematical Program with Equilibrium Constraints（MPEC）と呼ぶことにする．

表 2　推定のための 3 つの関数

ラムダ関数：$\boldsymbol{\Lambda}(\bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta}) \to \mathbf{P}$

$$\Lambda_i(a_i|x; \bar{V}_i, \mathbf{P}_{-i}; \boldsymbol{\theta}) = \frac{\exp\{v_i(x, a_i; \mathbf{P}_{-i}; \boldsymbol{\theta})\}}{\Sigma_{a' \in A} \exp\{v_i(x, a'; \mathbf{P}_{-i}; \boldsymbol{\theta})\}}, \ i \in I, \ x \in X, \ a \in A.$$

ファイ関数：$\boldsymbol{\Phi}(\bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta}) \to \bar{\mathbf{V}}$

$$\Phi_i(x; \bar{V}_i, \mathbf{P}; \boldsymbol{\theta}) = \sum_{a_i \in A} P_i(a_i|x) \left[v_i(x, a_i; \mathbf{P}_{-i}; \boldsymbol{\theta}) - \ln\left(P_i(a_i|x; \boldsymbol{\theta})\right) \right], \ i \in I, \ x \in X.$$

ガンマ関数：$\boldsymbol{\Gamma}(\mathbf{P}; \boldsymbol{\theta}) \to \bar{\mathbf{V}}$

$$\boldsymbol{\Gamma}_i(\mathbf{P}; \boldsymbol{\theta}) = \left[\mathbf{I}_M - \beta \sum_{a_i \in A} \mathbf{P}_i(a_i) * \tilde{\mathbf{F}}(a_i; \mathbf{P}_{-i}; \boldsymbol{\theta}) \right]^{-1} \sum_{a_i \in A} \mathbf{P}_i(a_i) * \left\{ \tilde{\boldsymbol{\pi}}_i(a_i; \mathbf{P}_{-i}; \boldsymbol{\theta}) + \mathbf{E}(a_i; \mathbf{P}_i; \boldsymbol{\theta}) \right\}, \ i \in I.$$

けで，すべてこのクラスの推定量に包含され，漸近正規性を持つことを示している[11]．Egesdal et al.（2015）（以下，ELS2015）は，2 ステップアルゴリズム，KS2015 と比較する形で，均衡条件を制約とした最適化問題を定式化することで推定を行った．この第 3 節では，NPL，2 ステップ，修正された NPL，均衡制約付き最適化の 4 つの推定アルゴリズムを順を追って見ていこう[12]．

ここで，いま一度，推定アルゴリズムに必要な 3 つの関数を（便宜的に名付けて）表 2 にまとめておく．いま，推定したい構造パラメーターのベクトルを $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{|\Theta|})$ としよう．第 2 節で導出した関数は $\boldsymbol{\theta}$ にも依存するので，以下では，$\boldsymbol{\Lambda}(\bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta})$，$\boldsymbol{\Phi}(\bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta})$，$\boldsymbol{\Gamma}(\mathbf{P}; \boldsymbol{\theta})$ などと書くことにする．

ここで，ラムダ関数とファイ関数は，$(\bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta})$ の関数であるのに対し，ガンマ関数は $(\mathbf{P}; \boldsymbol{\theta})$ の関数である．ファイ関数とガンマ関数は，ともに $\bar{\mathbf{V}}$ を求める関数であるが，ガンマ関数は $\bar{\mathbf{V}}$ に関して解かれた形になっているのに対して，ファイ関数は $\bar{\mathbf{V}}$ に関して解かれていないという違いがある．

第 4 節で述べるように，サンプルとして多数の市場を観察したデータが得られるとしよう．市場を $m \in \{1, \ldots, M\}$ として，期間 $t \in \{1, \ldots, T\}$ の各企業 $i \in I$ のデータを観察したとき，推定のための目的関数は，

$$L\left((\mathbf{x}, \mathbf{a}), \bar{\mathbf{V}}, \mathbf{P}; \boldsymbol{\theta}\right) \equiv \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \ln \Lambda_i(a_{imt}|x_{imt}; \bar{V}_i, \mathbf{P}_{-i}; \boldsymbol{\theta}) \tag{17}$$

として定式化することができる[13]．ただし，$(\mathbf{x}, \mathbf{a}) \equiv \left\{ (x_{imt}, a_{imt}) : i \in \{1, \ldots, N\}, \ m \in \{1, \ldots,\right.$

11　Bugni and Bunting（2018）の議論も参照．

12　この他にも，利潤関数がパラメーターに関して線形であれば（$\pi_i(x_t, a_t; \boldsymbol{\theta}) = \mathbf{z}_i(x_t, a_t) \cdot \boldsymbol{\theta}$），選択価値関数を利潤の期待割引総和と私的情報の期待割引総和に分けることで（$v_i(x_t, a_t) = \tilde{\mathrm{z}}_i(x_t, a_t) + \tilde{e}_i(x_t, a_t)$），推定することもできる．Aguirregabiria and Mira（2007）と Aguirregabiria and Mira（2010）を参照．

13　NPL や修正された NPL では，この関数は推定の過程で逐次的に更新されるので，尤度関数ではなく**疑似尤度関数（pseudo-likelihood function）**と呼ばれる．

$M\}$, $t \in \{1, \ldots, T\}\big\}$ とする．

3.1　ネステッド疑似尤度アルゴリズム（AM2007）

前節で見たように，$\mathbf{P}$ から $\mathbf{P}$ への関数$\mathbf{\Psi}$を定義すれば，この不動点が MPE となる．**ネステッド疑似尤度アルゴリズム（Nested Pseudo-Likelihood Algorithm, NPL）**とは，シングルエージェントの場合と同様に，$\mathbf{P}=\mathbf{\Psi}(\bullet;\boldsymbol{\theta})$ と $\boldsymbol{\theta}$ を逐次的に更新していきながら$\mathbf{P}=\mathbf{\Psi}(\mathbf{P};\boldsymbol{\theta})$ の解を求める方法である．

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, . . . 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，$\boldsymbol{\theta}$ の次の候補を見つける．

$$\boldsymbol{\theta}^{(k)}=\underset{\theta}{\operatorname{argmax}}\ \frac{1}{M}L\left((\mathbf{x},\mathbf{a}),\mathbf{\Gamma}(\mathbf{P}^{(k-1)};\boldsymbol{\theta}),\mathbf{P}^{(k-1)};\boldsymbol{\theta}\right), \tag{18}$$

ステップ 2　次のようにして，$\mathbf{P}$ の次の候補を見つける．

$$\mathbf{P}^{(k)}=\mathbf{\Lambda}[\mathbf{\Gamma}(\mathbf{P}^{(k-1)};\boldsymbol{\theta}^{(k)}),\mathbf{P}^{(k-1)};\boldsymbol{\theta}^{(k)}]. \tag{19}$$

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

シングルエージェントの場合で見たように，NPL は再帰的に 2 ステップアルゴリズムを繰り返すことで，計算時間を短縮しながら推定を行なうことができる．ところが，この方法で推定された値が，真の値へ収束するとは限らないことが指摘されている．例えば，Pesendorfer and Schmidt-Dengler（2010）は反例を用いて，計算した結果が真の値とは異なる値に収束することを示している．

3.2　2 ステップアルゴリズム（BBL2007 など）

次に，2 ステップアルゴリズムとして，まず，BBL2007 にしたがい，逆写像を用いる方法（inversion）を考えよう．ここでは，まず，(9) 式のベルマン方程式を他の企業の条件付き選択確率$\mathbf{P}_2$の関数ではなく，$\boldsymbol{\sigma}$の関数と考えることにしよう．つまり，すべての企業が戦略$\boldsymbol{\sigma}=(\sigma_1,\sigma_2)$にしたがっているとき，企業 1 の期待価値関数は次のようにあらわすことができる．

$$\bar{V}_1(x;\sigma)\equiv\int\Big\{\pi_1\big(x,\sigma_1(x,\varepsilon_1),\sigma_2(x,\varepsilon_2)\big)+\varepsilon_1(\sigma_1(x,\varepsilon_1))$$

$$+\beta\sum_{x'\in X}\bar{V}_1(x;\sigma)f\big(x'\,|\,x,\sigma_1(x,\varepsilon_1),\sigma_2(x,\varepsilon_2)\big)\Big\}\mathrm{dG}(\varepsilon_1,\varepsilon_2). \tag{20}$$

ただし，$G(\varepsilon_1, \varepsilon_2)$ は $\varepsilon_1, \varepsilon_2$ に関する同時確率分布とする．つまり，この $\bar{V}_1$ は（9）式とは異なり，企業 1 の私的情報だけではなく，すべての企業の私的情報に関する期待値となっている．この（20）式より，

$$\bar{V}_i(x;\boldsymbol{\sigma}^*) \geq \bar{V}_i(x;\sigma_i, \boldsymbol{\sigma}^*_{-i}), \quad \forall \sigma_i \tag{21}$$

がすべての企業とすべての状態 x について成り立つとき，$\boldsymbol{\sigma}^*$は MPE である．次に，企業 2 が σ_2 にしたがっているという前提で，企業 1 が状態 x を観察し，行動 a_1 をとるとき，前節で定義した v_1 を ε_2 に関して期待値をとったものを，次のようにあらわそう．

$$\tilde{v}_1(x, a_1) \equiv \int \Big\{ \pi_1\big(x, a_1, \sigma_2^*(x, \varepsilon_2)\big) + \beta \sum_{x' \in X} \bar{V}_1(x';\boldsymbol{\sigma}^*) f\big(x' \mid x, a_1, \sigma_2^*(x, \varepsilon_2)\big) \Big\} \mathrm{d}g(\varepsilon_2). \tag{22}$$

これは，今期の行動に関して企業 2 が σ_2^* にしたがっており，次の期以降はすべての企業が$\boldsymbol{\sigma}^*$にしたがうとするとき，今期のみ企業 1 が a_1 をとるときの価値関数の期待値である．したがって，企業 1 が最適な行動 a_1 を選択するならば，次の関係が成り立っていなければならない．

$$\tilde{v}_1(x, a_1) + \varepsilon_1(a_1) \geq \tilde{v}_1(x, a_1') + \varepsilon_1(a_1'), \quad \forall a_1'. \tag{23}$$

この関係を使えば 2 ステップアルゴリズムを用いて，条件付き選択確率より価値関数を推定することができる（Hotz and Miller, 1993）．つまり，私的情報 ε_i が独立にタイプ I 型極値分布にしたがうとすれば，任意の a_1, a_1' について，次の関係が得られる．

$$\tilde{v}_1(x, a_1') - \tilde{v}_1(x, a_1) = \ln\big(P_1(a_1' \mid x)\big) - \ln\big(P_1(a_1 \mid x)\big) \tag{24}$$

これにより，観察されたデータから状態 x のときに a_1 または a_1' が起きた頻度を求めて P_1 とすれば，この式より $\tilde{v}_1$ を推定することができる[14]．本稿では便宜的に，このような方法を「逆写像法」と呼んでおこう．BBL2007 は，$(\min\{\bar{V}_i(x;\boldsymbol{\sigma}^*) - \bar{V}_i(x;\sigma_i, \boldsymbol{\sigma}^*_{-i}), 0\})^2$ を最小化するようなパラメーターを推定しているが，本稿では，疑似尤度を最大化する方法を用いる．

なお，別の 2 ステップアルゴリズムとして，条件付き選択確率を一度だけ用いて価値関数を計算する方法を考えることもできる．つまり，初期値 $\mathbf{P}^{(0)}$ として，観察されたデータ $\{(x_{imt}, a_{imt})\}$ より各企業が行動を選択した頻度を用いると，推定されるパラメーターは

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{\mathrm{argmax}} \frac{1}{M} L\left((\mathbf{x}, \mathbf{a}), \boldsymbol{\Gamma}(\mathbf{P}^{(0)};\boldsymbol{\theta}), \mathbf{P}^{(0)};\boldsymbol{\theta}\right) \tag{25}$$

となる（LSE2015 も参照）．言い換えれば，この方法を再帰的に繰り返す方法が NPL である．

14　ただし，この方法で推定できるのは $\tilde{v}_1$ の「差」のみである．したがって，ある $a_1 = a^0$ に関して，$\tilde{v}_1(x, a^0) = 0$ と基準化しておく必要がある．

3.3　修正されたネステッド疑似尤度アルゴリズム（KS2012）

Kasahara and Shimotsu（2012）は，関数$\boldsymbol{\Lambda}$が縮小写像でない場合はNPLが収束しないとし，上の（19）式を修正して，$\mathbf{P}$を更新する方法を提案している．つまり，（19）式を，$\lambda\in[0,1]$でウェイトづけされた関数

$$\mathbf{P}^{(k)}=\{\boldsymbol{\Lambda}[\boldsymbol{\Gamma}(\mathbf{P}^{(k-1)};\boldsymbol{\theta}),\mathbf{P}^{(k-1)};\boldsymbol{\theta}]\}^{\lambda}\cdot(\mathbf{P}^{(k-1)})^{1-\lambda} \tag{26}$$

に代えて，上記のNPLアルゴリズムを行なう．ここで，$\lambda=1$ならば結果はNPLに等しく，$\lambda=0$ならば，上の2番目の2ステップアルゴリズムに等しくなる．KS2012は，これを**NPL-$\boldsymbol{\Lambda}$アルゴリズム**と名付けている．**KS2012**は，2ステップアルゴリズムで評価したヤコビ行列のスペクトラム半径を用いることを提案しているが，その値が小さくなれば，収束するまで多くの計算が必要となる[15]．**ELS2015**は，シミュレーションの中で$\lambda=\mathbf{0.5}$を採用している．

3.4　均衡制約付き最適化アルゴリズム（ELS2015）

ELS2015は，先行研究のもとで，このモデルを均衡制約付き最適化問題（MPEC）として解くことを提案している．表2の$\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})$と$\boldsymbol{\Lambda}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})$は，それぞれ，均衡における$\bar{\mathbf{V}}$と$\mathbf{P}$に関する「制約」として考えることができる．したがって，ある$\boldsymbol{\theta}$のもとで，MPEは，

$$\left\{(\bar{\mathbf{V}},\mathbf{P})\mid\begin{array}{l}\bar{\mathbf{V}}=\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})\\ \mathbf{P}=\boldsymbol{\Lambda}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})\end{array}\right\} \tag{27}$$

とあらわすことができる．本稿ではこれを「均衡制約」と呼ぶことにしよう．ここで，$\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})$は，$\boldsymbol{\Gamma}(\mathbf{P};\boldsymbol{\theta})$のように$\bar{\mathbf{V}}$について解く必要がなく，逆行列を計算することもないので，解を求める時間を短縮することができる．$\boldsymbol{\theta}$は，$\bar{\mathbf{V}}=\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})$と$\mathbf{P}=\boldsymbol{\Lambda}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})$を均衡制約として，$(\boldsymbol{\theta},\bar{\mathbf{V}},\mathbf{P})$に関して目的関数を最適化することで推定することができるので，均衡制約付き最適化問題は

$$\begin{aligned}&\max_{\boldsymbol{\theta},\bar{\mathbf{V}},\mathbf{P}}\ \frac{1}{M}L\left((\mathbf{x},\mathbf{a}),\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta}\right)\\ &\text{subject to: }\ \bar{\mathbf{V}}=\boldsymbol{\Phi}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})\\ &\qquad\qquad\mathbf{P}=\boldsymbol{\Lambda}(\bar{\mathbf{V}},\mathbf{P};\boldsymbol{\theta})\end{aligned} \tag{28}$$

と定式化できる．

ここで，このアルゴリズムは，$\bar{\mathbf{V}}$と$\mathbf{P}$を変数とすることで，DP問題を解くことなく，静学的な最適化問題として$\boldsymbol{\theta}$を求めている．ところが，制約条件が複雑な形をしているために解析的に解を求めるのが困難であり，変数と制約条件式の数が多いので，専用の「ソルバー」を用いる必要がある．この場合，この最適化問題の変数の数は$|\Theta|+N(1+|A|)\cdot|X|$，制約条件式の数

15　この場合のスペクトラム半径とは，2ステップアルゴリズムの推定量で評価した$\boldsymbol{\Psi}(\boldsymbol{P};\boldsymbol{\theta})$のヤコビ行列の固有値のうち最大のものである．

は $N(1+|A|)\cdot|X|$ となる．

4　モンテカルロ・シミュレーション

この節では，上で説明した 4 つの推定アルゴリズム ― ネステッド疑似尤度（NPL），逆写像法による 2 ステップ（2STEP），修正されたネステッド疑似尤度（NPL-Λ），均衡制約付き最適化（MPEC）― を比較するために，具体的なモデルを考えて，擬似的なデータを生成し，そのデータより推定を行なう．モデルは AM2007 にしたがい，複数の市場に出店するスーパーマーケットのモデルを用いよう．このモデルでは，各プレイヤーの行動は出店するかしないかの 2 択であり，計算が容易なものとなる．以下では，モデルを定式化し，データを生成し，ネステッド不動点アルゴリズム（NFXP）を加えた 5 つのアルゴリズムにより推定を行なっていく．

4.1　実験モデルの定式化

AM2007 で用いられた参入・退出モデルを，ここでも用いることにしよう．このモデルの特徴は，市場の状態および前期のプレイヤーの存在の有無によって，企業が操業するかしないかを決める単純なゲームであるということである．まず，企業（プレイヤー）をスーパーマーケット・チェーンとする．各企業は，小さく分割された市場 $m\in\{1,\ldots,M\}$ に参入するかしないかを考え，すでに参入している場合は退出するかしないかを決定する．各企業は，複数の市場で操業することができる．つまり，企業は**グローバル・プレイヤー**（global player）であるとする．各市場は十分に小さく，各企業は 1 つの市場にたかだか 1 店舗しか出店しない[16]．そこで，企業 i が市場 m で t 期に操業する（active）という選択を $a_{imt}=1$，操業しないという選択を $a_{imt}=0$ であらわそう．すると，t 期に市場 m で操業するときの企業 i の利潤は，次のような形で定式化できる．

$$\begin{aligned}&\Pi_{imt}(x_{mt},\varepsilon_{imt},a_{imt}=1,\mathbf{a}_{-i,mt};\boldsymbol{\theta})\\&\quad\equiv\theta_{RS}\ln(S_{mt})-\theta_{RN}\ln\left(1+\sum_{j\neq i}a_{jmt}\right)-\theta_{FC,i}\\&\quad\quad-\theta_{EC}(1-a_{im,t-1})+\varepsilon_{imt}(1).\end{aligned}\tag{29}$$

ただし，S_{mt} は t 期における市場 m のサイズ（離散型），$\boldsymbol{\theta}=(\theta_{RS},\theta_{RN},\theta_{EC},\theta_{FC,1},\ldots,\theta_{FC,N})$ はパラメーターである．もし，この企業が前期（$t-1$）に操業していなければ（$a_{im,t-1}=0$），今期は参入費用 θ_{EC} を払う必要がある．また，固定費用 $\theta_{FC,i}$ は企業によって異なる．この市場より

16　AM2007 では実証分析も行っているが，そこでの「市場」はチリの郡（*comuna*）である．AM2007 はチリに存在する 342 郡から大都市を除いた 189 郡を選び，そのサンプルの人口の中央値が 10,400 人となるようにしている．

得られる利潤はクールノー競争を反映しており，市場で操業する企業の数が多くなれば，個々の利潤は小さくなる．一方，企業が操業しなければ，外部の利潤として $\Pi_{imt}(x_{mt}, \varepsilon_{imt}, a_{imt}=0, \mathbf{a}_{-i,mt}; \boldsymbol{\theta}) = \varepsilon_{imt}(0)$ を得るとしよう．このモデルの状態（共有知識）は，$x_{mt}=(S_{mt}, \mathbf{a}_{m,t-1})$ である．すなわち，各期首で，各企業はその市場の状態と期首に存在している企業の数を観察し，それを前提としてその期に操業するかしないか（退出するか）を選択する．したがって，可能な S の数を $|S|$ とすれば，状態の数は，$2^N \times |S|$ となる[17]．ここで，市場の状態 S の推移確率を $f_S(S_{m,t+1}|S_{mt}; \boldsymbol{\theta})$ としよう．つまり，S は各企業の行動とは関係ない外生的なものとする．

このモデルでは，退出の際の利益または費用は考慮されておらず，各企業の参入費用は同一であることに注意しよう．第5節の別の参入・退出モデルでは，退出の際の売却利益を考慮し，各企業の参入費用が異なる場合を考えている．逆に，このモデルでは固定費用に関して異質性を仮定している．また，このモデルでは利潤の大きさに関する ε_{imt} が私的情報であるのに対し，第5節のモデルでは，退出の際の売却利益と参入費用が私的情報である場合を扱う．

(29) 式の利潤のもとで，動学ゲームを定式化していこう．まず，t 期に市場 m の企業 i が状態 $(S_{mt}, \mathbf{a}_{m,t-1})$ を観察したとき行動 a_{imt} をとる条件付き選択確率は，$P_i(a_{imt}|S_{mt}, \mathbf{a}_{m,t-1})$ となる．すると，操業を決めた企業の期待利潤は，(11) 式より次のように書ける．

$$\tilde{\pi}_i\big((S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}=1; \boldsymbol{\theta}\big) \equiv \sum_{\mathbf{a}_{-i,mt} \in A^{N-1}} \Big(\prod_{j \neq i} P_j(\mathbf{a}_{-i,mt}[j] | S_{mt}, \mathbf{a}_{m,t-1}) \Big)$$
$$\times \pi_i\big((S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}=1, \mathbf{a}_{-i,mt}; \boldsymbol{\theta}\big). \tag{30}$$

ただし，$\pi_i((S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}=1, \mathbf{a}_{-i,mt}; \boldsymbol{\theta})$ は (29) 式から $\varepsilon_{imt}(1)$ をのぞいた部分である．一方，操業しない企業の期待利潤は，$\tilde{\pi}_i((S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}=0; \boldsymbol{\theta}) = 0$ となる．次に，期待推移確率は，

$$\tilde{f}_i\big(S_{m,t+1}, a_{imt}, \mathbf{a}_{-i,mt} | (S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}; \mathbf{P}_{-i}, \boldsymbol{\theta}\big)$$
$$\equiv \sum_{\mathbf{a}_{-i,mt} \in A^{N-1}} \Big(\prod_{j \neq i} P_j(\mathbf{a}_{-i,mt}[j] | S_{mt}, \mathbf{a}_{m,t-1}) \Big) f_S(S_{m,t+1} | S_{mt}; \boldsymbol{\theta}) \tag{31}$$

である．

この $\tilde{\pi}_i((S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}; \boldsymbol{\theta})$ と $\tilde{f}_i(S_{m,t+1}, a_{imt}, \mathbf{a}_{-i,mt} | (S_{mt}, \mathbf{a}_{m,t-1}), a_{imt}; \mathbf{P}_{-i}, \boldsymbol{\theta})$ にもとづいて，各企業は (12) 式のようなマルコフ完全均衡戦略をとる．ここで，すべての企業は1つの均衡にしたがうとし，観察されたデータはその均衡から生成されるものと仮定しよう．

17　このように，動学ゲームではプレイヤーの数とともに状態の数がとても大きくなる．これは，広く「次元の呪い」として知られる．

4.2 データの生成

上のモデルにしたがって，推定に用いる擬似的なデータの生成について説明する．まず，真のパラメーターを設定した．ここでは単純化のために企業の数を $N=2$，その他のパラメーターは AM2007，ELS2015 などを参考にして，$S=\{1, 2, 3, 4, 5\}$，$\theta_{RS}=1.0$，$\theta_{RN}=2.0$，$\theta_{EC}=0.1$，$\theta_{FC,1}=1.5$，$\theta_{FC,2}=1.9$ とし，$f_S(S_{m,t+1}|S_{mt};\boldsymbol{\theta})$ は，AM2007 にしたがい，

$$f_S(S'|S)=\begin{pmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.2 & 0.8 \end{pmatrix} \tag{32}$$

とした[18]．また，プレイヤーの数が，行動の数が 2，市場の状態の数が 5 であり，状態空間のサイズが $2\times2\times5=20$ と比較的小さいことより，以下の計算では，状態 x を (S, a_1, a_2) の組とし，通し番号で並べることにした．つまり，$x^0=(1, 0, 0)$, $x^1=(1, 0, 1)$, $x^2=(1, 1, 0)$, . . . , $x^{18}=(5, 1, 0)$, $x^{19}=(5, 1, 1)$, のようにして 20 通りの状態をあらわすことにした[19]．また，$P_i(a=0|x)=1-P_i(a=1|x)$ なので，$\mathbf{P}_i$ を $P_i(a_i=1|x)$ を要素とするベクトルと見なした．

次に，データの数（市場の数）は $M=200$，$T=1$ として，以下のようにデータの生成を行った[20]．まず，上記のパラメーターにもとづく $\mathbf{P}$ を求めるために，楠田（2018）で説明した「CCP 不動点アルゴリズム」を用いた．つまり，真の値のパラメーターベクトル $\boldsymbol{\theta}^{true}$ をとして，$\bar{\mathbf{V}}^{(k)}=\boldsymbol{\Gamma}(\mathbf{P}^{(k)};\boldsymbol{\theta}^{true})$ と $\mathbf{P}^{(k+1)}=\boldsymbol{\Lambda}(\bar{\mathbf{V}}^{(k)}, \mathbf{P}^{(k)};\boldsymbol{\theta}^{true})$ を逐次的に繰り返し，$\|\mathbf{P}^{(k+1)}-\mathbf{P}^{(k)}\|<0.000001$ となった時点で収束とした[21]．次に，乱数によって $X^{old}\in\{0, 1, \ldots, 19\}$[20] を選び，(32) 式の推移確率と $\mathbf{P}$ の収束値より，次の期の状態 X^{new} を求める計算を 200 回行って，200 個のデータを作成した．さらに，X^{old} と X^{new} より推移確率を計算し，行列 $F^{dat}(X^{new}|X^{old})$ とした．

4.3 推定結果

推定は，Python 3.6.4 と，AMPL（モデリング言語），KNITRO（ソルバー）を用いた．基本的な計算プログラムとしては Python を用いて，NFXP，NPL，NPL-Λと，2STEP の目的関数の最適化（それぞれ，(18) 式と(25) 式），および MPEC（(28) 式）には AMPL と KNITRO を用いた．NFXP の内部ループ，NPL，NPL-Λの外部ループの $\boldsymbol{\theta}$ の初期値は（0.1, 0.1, 0.1, 0.1, 0.1）とし，それらの目的関数の最適化および均衡制約最適化における $\boldsymbol{\theta}$ の初期値は特に定めな

18 AM2007，ELS2015 のように $\theta_{EC}=1.0$ とすると，この $N=2$ ケースでは $\mathbf{P}$ の収束が確認できなかった．

19 Python の辞書型配列を使えば，$x=x^j$ のときの S^j, a_1^j, a_2^j $(j=0, 1, \ldots, 19)$ の値を簡単に取り出すことができる（付録 A 参照）．なお，Python では添字が 0 から始まる．

20 つまり，AM2007 のように，ランダムに生成した初期の状態を一度だけ更新して 2 期間のデータのみを用いる．一方，ELS2015 は，$T=1, 10, 20$ のケースを検討している．

21 計算の結果，収束までの計算は 15 回であった．

かった．（つまり，AMPL に初期値は与えていない．）なお，すべてのケースで $\beta=0.95$，そして NPL-Λ の λ は 0.5 とした．

付録 C に，MPEC で用いた AMPL の mod ファイルを示した[22]．AMPL はモデリング言語の一種であり，最適化問題を解くのに適している．(28) 式の問題は，変数 85 個，制約条件式 80 本の問題となる[23]．変数は，$((\theta_{RS}, \theta_{RN}, \theta_{EC}, \theta_{FC,1}, \theta_{FC,2}), \mathbf{P}_1, \mathbf{P}_2, \bar{\mathbf{V}}_1, \bar{\mathbf{V}}_2,)$ なので，var として定義し，最適化問題のためのパラメーターは，β，M と X（集合），そして上で求めた $X^{old}=(S^{old}, a_1^{old}, a_2^{old}), (a_1^{new}, a_2^{new})$ と F^{dat} であるので，それらを param として定義している．そして，目的関数 L と制約条件 $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, $\mathbf{\Phi}_1$, $\mathbf{\Phi}_2$ を設定した上で，Python でそれぞれのデータごとにパラメータを計算し，Python より AMPL の API（Application Programming Interface）を呼び出して，これらの値を mod ファイルにセットした[24]．AMPL で計算した結果は，API によって再び Python に返される．その他のアルゴリズムでの最適化も，このように API より AMPL を用いて計算している．

表 3 と図 1 に推定結果を示す．表 3 には，上の実験を 1000 回行なって得た 1000 個の推定値の平均と標準偏差，および平均計算時間を示した．この結果より，全体的に，θ_{RS}, $\theta_{FC,1}$, $\theta_{FC,2}$ については 2 ステップを除いてどのアルゴリズムも大差なく，ほぼ満足のいく結果が出たと言える．θ_{EC} に関しては，どの推定結果も過大な結果が出た．θ_{RN} に関しては，どのアルゴリズムも満足のいく結果が出なかった．平均計算時間に関しては，NFXP，NPL，NPL-Λ，MPEC，2STEP の順に長くかかった．平均計算時間を比べると，NFXP は NPL の 1.6 倍ほど長くかかっている

表 3　推定結果[a]

Algorithm	True values:	θ_{RS} **1.0**	θ_{RN} **2.0**	θ_{EC} **0.1**	$\theta_{FC,1}$ **1.5**	$\theta_{FC,2}$ **1.9**	Mean Time (in sec.)
NFXP	Mean	0.935	0.998	0.149	1.407	1.924	12.03
	Std.dev.	(0.446)	(2.143)	(0.164)	(0.507)	(0.686)	
NPL	Mean	1.117	2.365	0.139	1.439	1.869	7.41
	Std.dev.	(0.846)	(3.888)	(0.166)	(0.444)	(0.562)	
2STEP	Mean	0.889	0.001	0.168	1.555	2.213	0.21
	Std.dev.	(0.241)	(0.018)	(0.180)	(0.311)	(0.332)	
NPL-Λ	Mean	1.003	1.098	0.154	1.514	2.059	7.07
	Std.dev.	(0.393)	(1.943)	(0.175)	(0.393)	(0.518)	
MPEC	Mean	1.157	2.997	0.124	1.416	1.799	0.25
	Std.dev.	(0.458)	(3.001)	(0.164)	(0.432)	(0.576)	

a　アルゴリズムは，上から「ネステッド不動点」「ネステッド疑似尤度」「2 ステップ（逆写像法）」「修正されたネステッド疑似尤度」「均衡制約付き最適化」をあらわす．Mean，Std.dev は，それぞれ 1000 個の推定値の平均と標準偏差，Mean Time は平均計算時間．すべてのケースで $\beta=0.95$ としている．

22　mod ファイルとは，AMPL で最適化問題を記述したものである．詳しくは付録 B を参照．

23　各 x に対して $P_i(a=0\mid x)=1-P_i(a=1\mid x)$ となるので，$\mathbf{P}_i$ の要素の数は 20 となる．

24　API とは，異なるプログラム言語のやりとりを行なうものである．AMPL の API には，R, MATLAB, Python のものがあり，それらのプログラムから AMPL を操作することができる．

が、このような簡単な実験でソルバーを使えば，NFXP も十分に有用であることがわかった．ただし，この実験では，NPL や NPL-Λ（$\lambda=0.5$）も平均計算時間が長く，$\mathbf{P}$ の収束に時間がかかっていることがわかる．MPEC は，非常に短い時間で（θ_{RN} 以外は）ほぼ満足のいく結果を出している．より時間のかかる複雑な推定を行なう場合は，おそらくは最適なアルゴリズムであることを示唆している．

ただし，この結果は，プレイヤーの数が少なく，状態が 2 期間のみという比較的単純なケースより得られたことに注意する必要があろう．ELS2015 は，表 4 のようないくつかのモデルを検討しているが，このケース 2 やケース 3 では，NPL が収束しない場合も多く起きたことを報告している．また，ケース 4 では，λ の値によっては NPL-Λ も収束しないことが確認されている．

表 4　ELS2015 の実験[a]

1:	$S=\{2, 6, 10\}$	$N=3: \boldsymbol{\theta}_{FC}=(1.0, 0.9, 0.8)$	$(\theta_{RN}, \theta_{RS})=(2, 1)$	$\beta=0.96$
2:	$S=\{2, 6, 10\}$	$N=3: \boldsymbol{\theta}_{FC}=(1.0, 0.9, 0.8)$	$(\theta_{RN}, \theta_{RS})=(4, 1)$	$\beta=0.96$
3:	$S=\{1, 2, 3, 4, 5\}$	$N=5: \boldsymbol{\theta}_{FC}=(1.9, 1.8, 1.7, 1.6, 1.5)$	$(\theta_{RN}, \theta_{RS})=(2, 1)$	$\beta=0.95$
4:	$S=\{1, 2, 3, 4, 5\}$	$N=5: \boldsymbol{\theta}_{FC}=(1.9, 1.8, 1.7, 1.6, 1.5)$	$(\theta_{RN}, \theta_{RS})=(4, 2)$	$\beta=0.95$
5:	$S=\{1, 2, \ldots, 10\}$	$N=5: \boldsymbol{\theta}_{FC}=(1.9, 1.8, 1.7, 1.6, 1.5)$	$(\theta_{RN}, \theta_{RS})=(2, 1)$	$\beta=0.95$
6:	$S=\{1, 2, \ldots, 15\}$	$N=5: \boldsymbol{\theta}_{FC}=(1.9, 1.8, 1.7, 1.6, 1.5)$	$(\theta_{RN}, \theta_{RS})=(2, 1)$	$\beta=0.95$

a　$\boldsymbol{\theta}_{FC}=(\theta_{FC,1}, \theta_{FC,2}, \ldots, \theta_{FC,N})$ とする．ケース 1, 2 は KS2012，ケース 3, 4 は AM2007 にしたがっている．すべてのケースで $\theta_{EC}=1.0$ としている．ケース 1, 2 は $(\theta_{RN}, \theta_{RS})$ のみ推定し，他のケースはすべてのパラメーターを推定している．それぞれのケースで，$M=400$, $T=1, 10, 20$ である．

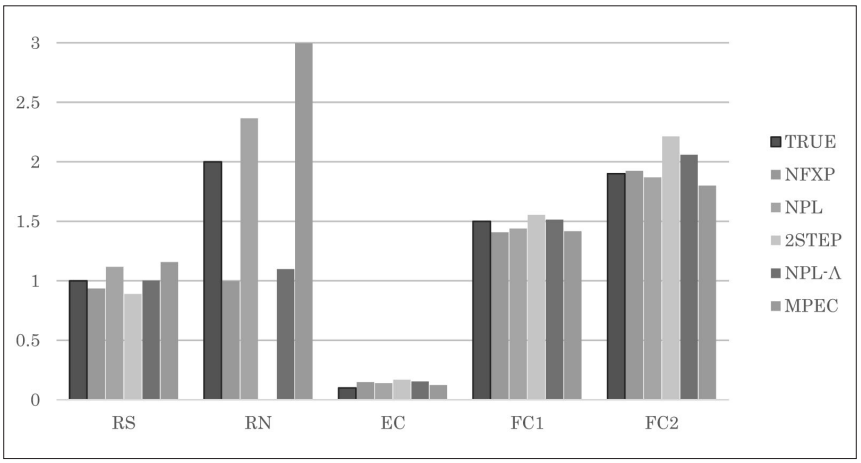


図 1　推定結果

5　応用例：売却利益と参入費用をともなう参入・退出モデル（POB2007）

この節では，応用例として POB2007 の参入・退出モデルについて概説する．このモデルでは，ある種の 2 ステップアルゴリズムを用いて，企業が参入・退出を決定する市場モデルを推定する．その 2 ステップアルゴリズムの特徴とは，まず，観察されたデータより価値関数を計算（recover）して，それにより疑似尤度関数を計算してパラメーターを推定するということである．AM2007 のモデルとこのモデルの違いは，このモデルでは，「退出する際に企業が得られる売却利益」や「参入する際に発生する参入費用」が，各企業の私的情報となっているということである．POB2007 は，この 2 ステップアルゴリズムを用いてそのような 2 つの私的情報に関するパラメーターを推定している．この節では，まずモデルの概要を設定し，POB2007 の 2 ステップによる推定法と，前節の均衡付き最適化による推定法を説明する．

5.1　参入・退出の基本モデル

POB2007 にしたがい，このモデルの設定を説明する[25]．ある 1 つの市場があり，すでに参入して操業している既存企業と，参入の可能性がある潜在的な参入企業が存在する[26]．市場で操業することにより得られる各期の利潤を $\pi(n)$ としよう．ただし，n はその期に市場で操業している企業の数とする．ここでは，企業の最大数を N とし，$n \in \{0, 1, \ldots, N\}$ としよう．すでに操業している既存企業は，各期の期首に状態（市場で操業している企業数）を観察し，次の期にも操業を継続するか，退出するかを決定する．次の期に退出する場合は，（退出費用と相殺された）**売却利益（sell-off value）**を得る．一方，この市場に参入していない企業が $N-n$ 存在する．それらの潜在的な参入企業は，次の期に参入するか，しないかを決定する．参入する場合は，**参入費用（sunk cost）**を支払う必要がある．一般に，売却利益や参入費用は，企業の私的情報なので，外部から観察するのは難しいが，ここでは動学ゲームを設定し，それにもとづいて構造推定することにより，それらの値に関するパラメーターを推定することを目的とする．以下，既存企業と参入企業の問題を順に考える．

既存企業の問題

まず，既存企業 I が直面する DP 問題は，次のように定式化できる[27]．

25　ただし，POB2007 では，状態変数の 1 つとして外生的な需要シフター z とその成長率 g を考えているが，本稿では単純化のために省略した．つまり，利潤は状態に応じて確定的である．

26　POB2007 では，市場の数が複数である場合と潜在的参入企業の数が確率変数である場合も考察している．

27　このモデルでは，既存企業は継続/退出を決定する期では操業していることに注意．

$$\max\{\pi(n)+\beta\phi,\ \pi(n)+\beta V^C(n)\}. \tag{33}$$

ただし，ϕ は売却利益，V^C は次の期以降も操業を継続するときの価値関数，β は割引因子である（$0<\beta<1$）．ここで，V^C は，次の期の企業数に依存し，その数は今期の企業数（n）と今期の期末に決定する退出数（x）および参入数（e）によって確定する．企業は売却利益 ϕ（私的情報）を観察した上で意思決定を行なう．一方，x と e は企業の意思決定の時点では未知であるので，各企業は，今期の企業数（n）と，次の期に自分自身も操業するということ（これを $\chi^I=1$ であらわす）を前提として，次の期の企業数（n'）に関する整合的な信念を形成する．売却利益 ϕ は，企業数 n とは無関係な確率分布 F にしたがうとしよう．すると，V^C は次のように書くことができる．

$$V^C(n)\equiv\sum_{n'}\int\max\{\phi',\ V^C(n')\}\mathrm{d}F(\phi')\cdot m^I(n'|n),\quad n'=n-x+e. \tag{34}$$

ここで，$m^I(n'|n)$ は，既存企業 I が n と $\chi^I=1$ のもとで形成する n' に関する信念（共有知識）であり，既存企業の認識の中での推移確率（incombents' perceived transition probability）と考えることができる．この $m^I(n'|n)$ は，「自分自身をのぞく（$n-1$）人のうち x 人が退出する確率」$b^x(x,\ n-1|n)$ と，「今の時点で参入していない（$N-n$）人のうち e 人が参入する確率」$b^e(e,\ N-n|n)$ によって形成される．つまり，$0\le x\le n-1$ と $0\le e\le N-n$ に対して，

$$m^I(n'|n)=\sum_{\substack{x,e:\\ n'=n-x+e}} b^x(x,\ n-1|n)\cdot b^e(e,\ N-n|n) \tag{35}$$

とあらわすことができる．

潜在的な参入企業の問題

一方，潜在的な参入企業 E は参入するかしないかを決定する．参入する場合は，参入費用 κ をその期に支払い，次の期より操業することができるとする．企業 E は，この参入費用 κ（私的情報）を観察した上で意思決定を行なう．すると，参入企業 E が直面する問題は，

$$\max\{\kappa,\ \beta V^E(n)\} \tag{36}$$

となる[28]．ただし，V^E は企業 E が参入して（これを $\chi^E=1$ であらわす），次の期以降も操業を継続するときの価値関数であり，次のように定式化できる．

$$V^E(n)\equiv\sum_{n'}\int\max\{\phi',\ V^C(n')\}\mathrm{d}F(\phi')\cdot m^E(n'|n). \tag{37}$$

この $m^E(n'|n)$ は参入企業 E が，n と $\chi^E=1$ のもとで形成する n' に関する信念であり，この参

28　退出の場合の売却利益とは異なり，参入費用は操業開始の 1 期前に支払うことに注意しよう．参入に十分な時間がかかることを考えれば，これは自然な設定であろう．

入企業の認識の中での推移確率（entrants' perceived transition probability）である．これは，「現在操業している n 人のうち x 人が退出する確率」$b^x(x,\, n|n)$ と，「今の時点で参入していない企業で自分自身をのぞく（$N-n-1$）人のうち（$e-1$）人が参入する確率」$b^e(e-1,\, N-n-1|n)$ によって決まる．つまり，$0\leq x\leq n$ と $1\leq e\leq N-n-1$ に対して，

$$m^E(n'|n)=\sum_{\substack{x,e:\\ n'=n-x+e}} b^x(x,\, n|n)\cdot b^e(e-1,\, N-n-1|n) \tag{38}$$

となる．

各企業の信念

いま，ある 1 人の既存企業が状態 n を観察したとき退出する確率を $P^x(n)$，ある 1 人の潜在的な参入企業が状態 n を観察したとき参入する確率を $P^e(n)$ としよう．すると，(35) 式と (38) 式の b^x と b^e は，次のような二項分布によって求めることができる．

$$\begin{aligned} b^x(x,\, r|n) &\equiv \binom{r}{x}\cdot[P^x(n)]^x\cdot[1-P^x(n)]^{r-x} \qquad r\geq x, \\ b^e(e,\, r|n) &\equiv \binom{r}{e}\cdot[P^e(n)]^e\cdot[1-P^e(n)]^{r-e} \qquad r\geq e. \end{aligned} \tag{39}$$

結果として，既存企業の認識の中の推移確率 $m^I(n'|n)$ と潜在的な参入企業の認識の中の推移確率 $m^E(n'|n)$ は，$P^x(n)$ と $P^e(n)$ に依存して決まることになる．ここで，各企業の私的情報である退出利益 ϕ と参入費用 κ について，POB2007 にしたがって次の仮定を設定しよう．

仮定（売却利益と参入費用）

売却利益 ϕ と参入費用 κ は，市場，期，企業に関して独立で，それぞれ次の確率分布にしたがう確率変数である．

$$F(\phi)\equiv 1-\exp(-\phi/\sigma), \tag{40}$$

$$G(\kappa)\equiv\begin{cases} 1-a\kappa\,\exp[-a(\kappa-1/a)] & \text{if } \kappa\in(1/a,\,\infty) \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

もし，各企業が合理的に行動するならば，これらの確率は条件付き選択確率と考えることができる．つまり，$P^x(n)\equiv\Pr\{\phi> V^C(n)\}$, $P^e(n)\equiv\Pr\{\kappa<\beta V^E(n)\}$ であるので，これらは次のようにあらわすことができる．

$$P^x(n)\equiv 1-F\left(V^C(n)\right), \tag{42}$$

$$P^e(n)\equiv G\left(\beta V^E(n)\right). \tag{43}$$

図 2 に，これらのイメージを示した．ここで，(42) 式と (43) 式の右辺は，それぞれ $V^C(n)$ と $V^E(n)$ の関数となっているので，これらを $\Lambda^x(n;V^C(n))$ と $\Lambda^e(n;V^E(n))$ であらわそう．さ

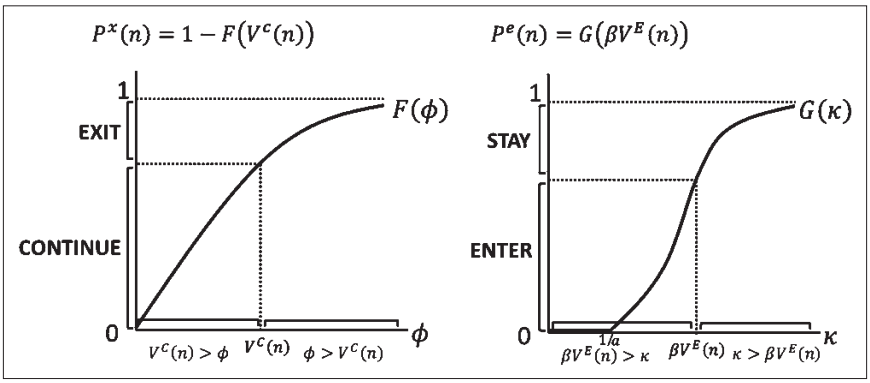


図 2　退出確率と参入確率

らに，$\mathbf{V}^C=(V^C(0),\ldots,V^C(N))'$, $\mathbf{V}^E=(V^E(0),\ldots,V^E(N))'$ とすると，(42) 式と (43) 式は状態 n でまとめて，それぞれ，$\mathbf{\Lambda}^x(\mathbf{V}^C)$, $\mathbf{\Lambda}^e(\mathbf{V}^E)$ であらわすことができる．（これらは，前節の「ラムダ関数」に相当する．）

価値関数の計算

次に，期待価値関数を次のように定義しよう．

$$EV^C(n)=\int \max\{\phi,\ V^C(n)\}\,\mathrm{d}F(\phi). \tag{44}$$

これは，次のような表現に書きかえることができる．

$$\begin{aligned} EV^C(n) &=\left(1-P^x(n)\right)V^C(n)+P^x(n)\,E[\phi'|\phi'>V^C(n)] \\ &=\left(1-P^x(n)\right)V^C(n)+P^x(n)\ \left(V^C(n)+\sigma\right) \\ &=V^C(n)+\sigma P^x(n). \end{aligned} \tag{45}$$

ここで，$E[\phi'|\phi'>V^C(n)]=V^C(n)+\sigma$ を用いている[29]．これより，

$$\begin{aligned} V^C(n) &=\sum_{n'} m^I(n'|\,n)\ \left(\pi(n')+\beta EV^C(n')\right) \\ &=\sum_{n'} m^I(n'|\,n)\ \left(\pi(n')+\beta V^C(n')+\beta\sigma P^x(n')\right) \end{aligned} \tag{46}$$

29　$f(\phi)$ を密度関数とすると，

$$\begin{aligned} E[\phi'|\phi'>V^C(n)] &\equiv \frac{1}{1-F(V^C(n))}\int_{V^C(n)}^{\infty}\phi'\cdot f(\phi')\mathrm{d}\phi' \\ &=\frac{1}{\exp(-V^C(n)/\sigma)}\int_{V^C(n)}^{\infty}(\phi'/\sigma)\cdot\exp(-\phi'/\sigma)\mathrm{d}\phi' \end{aligned}$$

なので，$\phi\equiv-\phi'/\sigma$ として置換積分すれば $V^C(n)+\sigma$ が求まる．

となる．ここで，この右辺は $V^C(n)$ と $P^x(n)$ の関数になっているので，これを $\Phi^C(n;\mathbf{V}^C, \mathbf{P}^x)$ であらわそう．ただし，$\mathbf{P}^x=\left(P^x(0),\ldots,P^x(N)\right)'$ とする．同様に，

$$V^E(n)=\sum_{n'} m^E(n'|n)\left(\pi(n')+\beta V^C(n')+\beta\sigma P^x(n')\right) \tag{47}$$

なので，(47) 式の右辺を $\Phi^E(n;\mathbf{V}^C, \mathbf{P}^x)$ としよう．（これらは，前節の「ファイ関数」である．）

さらに，(46) 式を n に関して行列・ベクトル表現にすると，

$$\mathbf{V}^C=[\mathbf{I}_M-\beta\mathbf{M}^I]^{-1}\mathbf{M}^I[\pi+\beta\sigma\mathbf{P}^x] \tag{48}$$

と書ける．ただし，$\pi\equiv\left(\pi(0),\ldots,\pi(N)\right)'$, $\mathbf{M}^I\equiv\begin{bmatrix} m^I(0|0) & \cdots & m^I(N|0) \\ m^I(0|1) & \cdots & m^I(N|1) \\ \vdots & \ddots & \vdots \\ m^I(0|N) & \cdots & m^I(N|N) \end{bmatrix}$

とする．同様に，参入企業 E に関しても次のように書ける．

$$\mathbf{V}^E=\mathbf{M}^E[\boldsymbol{\pi}+\beta\mathbf{V}^C+\beta\sigma\mathbf{P}^x]. \tag{49}$$

ただし，$\mathbf{M}^E$ は $m^E(n'|n)$ を要素とする行列である．(48) 式と (49) 式の右辺を，それぞれ $\mathbf{\Gamma}^C(\mathbf{P}^x)$ と $\mathbf{\Gamma}^E(\mathbf{V}^C, \mathbf{P}^x)$ と書くことにしよう．

5.2　推定アルゴリズム

ここからは，参入・退出モデルの推定について考える．推定の目的は，売却利益，参入費用に関するパラメーター，σ と κ を求めることである．推定したいパラメーターは，$\boldsymbol{\theta}=(\sigma,\kappa)$ にまとめられるので，以下では各関数に $\boldsymbol{\theta}$ をつけて書くことにする．

2 ステップアルゴリズム（POB2007）

POB2007 は，以下のような 2 ステップアルゴリズムを考え，シミュレーションを行っている．そのアイデアとは，観察されたデータより，各状態に対して参入が起きた頻度，および次の状態に以降した推移確率を求め，それらが漸近的に理論値に収束すると仮定して，価値関数を計算するのに用いる，というものである．つまり，各企業は観察した状態に整合的な信念を形成すると仮定していたので，十分大きなデータが観察できれば，実際に起きた参入・退出の数より“平均的な”価値関数が計算できるであろう．そして，もし真の値に近いパラメーターが推定できれば，その推定値のもとで，その平均的な価値関数は各企業が予測する期待値に近いものになっているはずである．

推定する方法は，次のように 2 段階とする．

ステップ 1：観察されたデータより価値関数を計算する．

ステップ 2：その価値関数を用いてパラメーターの推定を行なう．

そこで，まず**ステップ 1** として，ある状態のもとで参入が起きた頻度より求めた割合を $\tilde{\mathbf{p}}^x$，状態から状態へ推移した頻度より求めた割合を $\tilde{\mathbf{M}}^I$ とし，それぞれ，$\mathbf{p}^x(\boldsymbol{\theta})$ と $\mathbf{M}^I(\boldsymbol{\theta})$ に置き換えれば，（48）式は次のように推定できる[30]．

$$\hat{\mathbf{V}}^C(\boldsymbol{\theta}) = \tilde{\mathbf{A}}_1 + \tilde{\mathbf{A}}_2 \sigma. \tag{50}$$

ただし，$\tilde{\mathbf{A}}_1 \equiv [\mathbf{I}_M - \beta \tilde{\mathbf{M}}^I]^{-1} \tilde{\mathbf{M}}^I \boldsymbol{\pi}$, $\tilde{\mathbf{A}}_2 \equiv \beta [\mathbf{I}_M - \beta \tilde{\mathbf{M}}^I]^{-1} \tilde{\mathbf{p}}^x$ とする．同様に，$\tilde{\mathbf{M}}^E$ を $\mathbf{M}^E(\boldsymbol{\theta})$ に置き換えれば，（49）式は

$$\hat{\mathbf{V}}^E(\boldsymbol{\theta}) = \tilde{\mathbf{B}}_1 + \tilde{\mathbf{B}}_2 \sigma. \tag{51}$$

となる．ただし，$\tilde{\mathbf{B}}_1 \equiv \tilde{\mathbf{M}}^E [\mathbf{I}_M - \beta \tilde{\mathbf{M}}^I]^{-1} \tilde{\mathbf{M}}^I + \beta \tilde{\mathbf{M}}^E \tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}_2 \equiv \beta \tilde{\mathbf{M}}^E (\tilde{\mathbf{A}}_2 + \tilde{\mathbf{p}}^x)$ とする．

次に，**ステップ 2** では，**ステップ 1** で計算した価値関数をもとにパラメーターの推定を行なう．推定の目的関数を擬似尤度関数，推移確率および価値関数は**ステップ 1** で計算したものを用いる．ここでは，前節と同じように，サンプルとして多数の市場を観察したデータが得られるとし，市場を $m \in \{1, \ldots, M\}$，期間 $t \in \{1, \ldots, T\}$ の市場のデータが観察できるとしよう．市場 m において t 期に操業している企業の数が n_{mt}，退出が起きた回数 x_{mt}，参入が起きた回数 e_{mt} を観察したとすると，観察されるデータは $(\mathbf{n}, \mathbf{x}, \mathbf{e}) \equiv \{(n_{mt}, x_{mt}, e_{mt}) : m \in \{1, \ldots, M\},\ t \in \{1, \ldots, T\}\}$ となるので，疑似尤度関数は，

$$\begin{aligned} \mathcal{L}\left((\mathbf{n}, \mathbf{x}, \mathbf{e}); \boldsymbol{\theta}\right) = \prod_{m,t} & \left[1 - F\left(\hat{\mathbf{V}}^C(\boldsymbol{\theta}); \boldsymbol{\theta}\right)\right]^{x_{mt}} F\left(\hat{\mathbf{V}}^C(\boldsymbol{\theta}); \boldsymbol{\theta}\right)^{n_{mt} - x_{mt}} \\ & \times G\left(\hat{\mathbf{V}}^E(\boldsymbol{\theta}); \boldsymbol{\theta}\right)^{e_{mt}} \left[1 - G\left(\hat{\mathbf{V}}^E(\boldsymbol{\theta}); \boldsymbol{\theta}\right)\right]^{N - e_{mt}} \end{aligned} \tag{52}$$

となる．そこで，この対数をとり，

$$\begin{aligned} \mathrm{L}\left((\mathbf{n}, \mathbf{x}, \mathbf{e}); \boldsymbol{\theta}\right) = \sum_{m,t} \Big\{ & x_{mt} \ln\left[1 - F\left(\hat{\mathbf{V}}^C(\boldsymbol{\theta}); \boldsymbol{\theta}\right)\right] + (n_{mt} - x_{mt}) \ln F\left(\hat{\mathbf{V}}^C(\boldsymbol{\theta}); \boldsymbol{\theta}\right) \\ & + e_{mt} \ln G\left(\hat{\mathbf{V}}^E(\boldsymbol{\theta}); \boldsymbol{\theta}\right) + (N - e_{mt}) \ln \left[1 - G\left(\hat{\mathbf{V}}^E(\boldsymbol{\theta}); \boldsymbol{\theta}\right)\right] \Big\} \end{aligned} \tag{53}$$

を最大にするような $\boldsymbol{\theta}$ を求めればよい．

均衡付き最適化アルゴリズム（MPEC）

上で求めたラムダ関数（（42）式と（43）式），ファイ関数（（46）式と（47）式）を制約として用いれば，前節のように，均衡付き最適化アルゴリズムを使うことができる．つまり，

30　正確には，状態 n が起きた t の集合を $T(n)$，その数を $\#T(n)$ とすれば，$\tilde{\mathbf{p}}^x$ の要素は $\tilde{p}^x(n) \equiv \frac{1}{\#T(n)} \sum_{t \in T(n)} \frac{x_t}{n}$ というように定義できる．

$\boldsymbol{\Phi}^C(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta})$, $\boldsymbol{\Phi}^E(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta})$, $\boldsymbol{\Lambda}^x(\mathbf{V}^C; \boldsymbol{\theta})$, $\boldsymbol{\Lambda}^e(\mathbf{V}^E; \boldsymbol{\theta})$ を，均衡における $\mathbf{V}^C$, $\mathbf{V}^E$, $\mathbf{P}^x$, $\mathbf{P}^e$ に関する制約として考えると，ある $\boldsymbol{\theta}$ のもとで，MPE は

$$\left\{ (\mathbf{V}^C, \mathbf{V}^E, \mathbf{P}^x, \mathbf{P}^e) \;\middle|\; \begin{array}{l} \mathbf{V}^C = \boldsymbol{\Phi}^C(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta}) \\ \mathbf{V}^E = \boldsymbol{\Phi}^E(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta}) \\ \mathbf{P}^x = \boldsymbol{\Lambda}^x(\mathbf{V}^C; \boldsymbol{\theta}) \\ \mathbf{P}^e = \boldsymbol{\Lambda}^e(\mathbf{V}^E; \boldsymbol{\theta}) \end{array} \right\} \tag{54}$$

とあらわすことができるので，均衡制約付き最適化問題（MPEC）は

$$\begin{aligned} \max_{\boldsymbol{\theta}, \mathbf{V}^C, \mathbf{V}^E, \mathbf{P}^x, \mathbf{P}^e} \quad & \frac{1}{M} L\left((\mathbf{n}, \mathbf{x}, \mathbf{e}), \mathbf{V}^C, \mathbf{V}^E, \mathbf{P}^x, \mathbf{P}^e; \boldsymbol{\theta}\right) \\ \text{subiect to:} \quad & \mathbf{V}^C = \boldsymbol{\Phi}^C(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta}) \\ & \mathbf{V}^E = \boldsymbol{\Phi}^E(\mathbf{V}^C, \mathbf{P}^x; \boldsymbol{\theta}) \\ & \mathbf{P}^x = \boldsymbol{\Lambda}^x(\mathbf{V}^C; \boldsymbol{\theta}) \\ & \mathbf{P}^e = \boldsymbol{\Lambda}^e(\mathbf{V}^E; \boldsymbol{\theta}) \end{aligned} \tag{55}$$

と定式化できる．この場合，変数の数は $2+4N$，制約条件式の数は $4N$ となる．

なお，POB2007 では，次の 3 つの推定法を提案している．

1）疑似尤度関数（pseudo-likelihood function）を用いる．
2）データ上の参入・退出確率とモデル上の参入・退出確率の差が最小となるようなモーメント推定を行なう．
3）データ上の参入・退出確率の平均とモデル上の参入・退出確率の平均の差が最小となるようなモーメント推定を行なう．

さらに，POB2007 は 2 種類の方法で推定した推移確率と 2 種類の価値関数を考え，合計 12 通りの推定法でモンテカルロ・シミュレーションを行ない，結果を比較している．

5.3　推定結果

最後に，上の 2 つのアルゴリズムを用いて推定を行なう．POB2007 とは異なり，利潤関数は

$$\pi(n) = 1.0 - \ln(1 + 0.05\mathrm{n}) \tag{56}$$

とし，パラメーターの真の値は，$\sigma = 1.2$, $\kappa = 0.3$ とした．これは，真の値のパラメーターのもとで，比較的短い回数で収束させるためである[31]．シミュレーションは，市場の数を $C = 1000$ とし，実験の回数を $R = 500$ 回として，500 個の推定値の平均と標準偏差を求めた．第 4. 2 節と同様に，

31　真のパラメーターによっては，条件付き選択確率が収束しない場合もあった．

ここでも，$T=1$ とし，乱数を発生させて，それぞれの市場で状態 $n\in\{0,\ldots,N\}$, $N=19$ を生成し，真の値のパラメーターのもとで収束させて求めた P_x と P_e より，それぞれ退出数 x と参入数 e のデータを生成した．推定アルゴリズムは上で説明した 2 つを使い，2STEP では，生成したデータより，上の$\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2, \tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2$ を計算して，AMPL にパラメーターとして与え，目的関数を最大化する（σ, κ）を計算させた．一方，MPEC では，(55)式の制約条件式を AMPL に記述し，$\mathbf{P}^x, \mathbf{P}^e, \mathbf{V}^C, \mathbf{V}^E$ と（σ, κ）に関して目的関数を最大化して推定を行った．

表 5 に推定結果を示す．

表 5　推定結果[a]

β	Algorithm	True values:	σ **1.2**	κ **0.3**	Mean Time (in sec.)
0.95	2STEP	Mean	1.011	0.256	0.20
		Std.dev.	(0.117)	(0.024)	
	MPEC	Mean	1.112	0.130	0.44
		Std.dev.	(0.937)	(0.624)	
0.96	2STEP	Mean	0.870	0.297	0.21
		Std.dev.	(0.160)	(0.029)	
	MPEC	Mean	1.460	0.067	0.36
		Std.dev.	(2.162)	(0.018)	

a　アルゴリズムは，「2 ステップ（POB）」「均衡制約付き最適化」をあらわす．Mean，Std.dev は，それぞれ 1000 個のデータの平均と標準偏差，Mean Time は平均計算時間．

推定のための計算時間は，2STEP よりも MPEC が 2 倍ほど長くかかった．一方，推定値の正確さは，全般的にどちらが優越しているとも言えない．σ に関しては MPEC の方がやや近い値となっているのに対し，κ に関しては 2STEP のみが満足のいく結果を示している．ただし，このパラメーターのもとでの実験では，退出確率がかなり 0 に近く，退出する企業（x）がほとんどいなかった．このようなモデルに対して，どのようなアルゴリズムが適切かという推定アルゴリズム選択の問題は，さらなる研究が待たれるところであろう．

6　おわりに

この研究ノートでは，動学ゲームの構造推定のためのいくつかの推定アルゴリズムについて説明した．最後に，本稿のまとめとして，いくつかのコメントを付す．まず，本稿の議論は，ほぼロジット型（**仮定 CLOGIT**）を仮定していたが，私的情報が他の分布（正規分布など）にしたがっている場合もある．あるいは，私的情報が系列相関していることもありえる．過去の研究分析では，そのような場合についても研究されている．また，すべてのプレイヤーが，同じ均衡にしたがっているということもとても強い仮定である．Aguirregabiria and Mira（2018）は，複数均衡と観察されない共有知識の異質性の問題を考察している（Luo, et al., 2018 も参照）．また，Aguirregabiria（2012）は，複数均衡の中で均衡が不連続的に“ジャンプ”しない条件

（ホモトピー）を論じている．さらに，実証的に，現実の企業がこれまで述べたような形の“ゲーム”を行っているのか，というより本質的な問題も考えねばならない．Weintraub, Benkard, and Roy（2008）は，"obvious equilibrium" という概念を提唱し，企業が私的情報と産業の“平均的な”状態を観察するとする興味深い考察を行っている．現実の企業が，限定された情報より最適な戦略を選択すると考えるならば，このような現実に合わせた設定も今後検討していく必要があるだろう．

最後に，各アルゴリズム間の関係に関する最近の研究として，Bugni and Bunting（2018）を見ておこう．Bugni and Bunting（2018）は，段階の政策反復法による推定量のクラスを考え，これを目的関数の規準（creteria）に関して「最大尤度（maximum likelihood）」と「最小距離（minimum distance）」の 2 種類に分けた．前者の K-ML 推定量には本稿で説明した AM2007 などがあり，後者の K-MD 推定量には Pesendofer and Schmidt-Dengler（2008）などが相当する．さらに，K＝1 とした 1-ML，1-MD 推定量が 2 ステップ推定量に当たり，K＝∞とした ∞-ML 推定量がネステッド疑似尤度による推定量となる．では，このように各アルゴリズムを 1 つのクラスの中で整理した場合，推定のための最適な K とはどのような値であろうか？ この問題は，求められる推定量に一致性や漸近正規性などを期待する場合，整理されたアルゴリズムの“リスト”より，最適なアルゴリズムをどのように選択すべきか，という問題でもある．各アルゴリズム間の関係や，性質の違いに関するさらなる研究が期待されるところである．

この研究ノートの目的は，本稿で説明してきた推定アルゴリズムに優劣をつけることではない．計算時間や，概念理解の容易さ，コンピューターやソフトウェアの計算能力の制約など，さまざまな要因によって，それぞれ対象となる個々の寡占モデルに対して最適な推定アルゴリズムが選択されるべきだ，というのが筆者の立場である[32]．ただし，昨今の計算環境の飛躍的な向上を考えれば，幅広い寡占モデルに適用可能な汎用的アルゴリズムの“決定版”が教科書に掲載される日が近いのかもしれない．しかし，その日が来たとしても，過去十数年間に渡り，多くの研究者がさまざまなアルゴリズムを提案し，この分野において広く受け入れられてきたことが忘れられることのないように願うものである．

参考文献

Aguirregabiria, V.（2012）. A method for implementing counterfactual experiments in models with multiple equilibria. *Economics Letters,* 114, 190-194.

Aguirregabiria, V., & Ho, C.-Y.（2012）. A dynamic oligopoly game of the US airline industry: Estimation and policy experiments. *Journal of Econometrics,* 168, 156-173.

Aguirregabiria, V., & Mira, P.（2007）. Sequential estimation of dynamic discrete games. *Econometrica,* 75, 1-53.

32　それゆえに，紙幅の都合上，本稿で説明できなかった K-MD クラスのアルゴリズムに関してもまとめる機会を待ちたい．

Aguirregabiria, V., & Mira, P. (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics,* 156, 38-67.

Aguirregabiria, V., & Mira, P. (2018). Identification of games of incomplete information with multiple equilibria and common unobserved heterogeneity. Manuscript.

Bajari, P., Benkard, C. L., & Levin, J. (2007). Estimating dynamic models of imperfect competition. *Econometrica,* 75, 1331-1370.

Berry, S. T. (1992). Estimation of a model of entry in the airline industry. *Econometrica,* 60, 889-917.

Besanko, D., & Doraszelski, U. (2004). Capacity dynamics and endogenous asymmetries in firm size. *RAND Journal of Economics,* 35, 23-49.

Bresnahan, T. F., & Reiss, P. C. (1990). Entry in monopoly markets. *Review of Economic Studies,* 57, 531-553.

Bresnahan, T. F., & Reiss, P. C. (1991a). Empirical models of discrete games. *Journal of Econometrics,* 48, 57-81.

Bresnahan, T. F., & Reiss, P. C. (1991b). Entry and competition in concentrated markets. *Journal of Political Economy,* 99, 977-1009.

Bugni, F. A., & Bunting, J. (2018). On the iterated estimation of dynamic discrete choice games. cemmap Working Paper, CWP13/18.

Chen, J. (2009). The effects of mergers with dynamic capacity accumulation. International *Journal of Industrial Organization,* 27, 92-109.

Doraszelski, U., & Markovich, S. (2007). Advertising dynamics and competitive advantage. *RAND Journal of Economics,* 38, 557-592.

Doraszelski, U., & Pakes, A. (2007). A framework for applied dynamic analysis in IO. in Armstrong, M., & Porter, R. (eds.), *Handbook of Industrial Organization, Vol. 3* (pp. 1887-1966). Amsterdam: North-Holland.

Doraszelski, U., & Satterthwaite, M. (2005). Foundations of Markov-perfect industry dynamics: Existence, purification, and multiplicity. Manuscript. Department of Economics. Harvard University.

Doraszelski, U., & Satterthwaite, M. (2010). Computable Markov-perfect industry dynamics. *RAND Journal of Economics,* 41, 215-243.

Egesdal, M., Lai, Z., & Su, C.-L. (2015). Estimating dynamic discrete-choice games of incomplete information. *Quantitative Economics,* 6, 567-597.

Ericson, R., & Pakes A. (1995). Markov-perfect industry dynamics: A framework for empirical work. *Review of Economic Studies,* 62, 53-82.

Fourer, R., Gay, D. M., & Kernighan, B. W. (2002). *Ampl: A Modeling Language for Mathematical Programming (2nd ed.).* Boston: Cengage Learning.

Gowrisankaran, G. (1999). A dynamic model of endogenous horizontal mergers. *RAND Journal of Economics,* 30, 56-83.

Gowrisankaran, G., & Town, R. J. (1997). Dynamic equilibrium in the hospital industry. *Journal of Economics and Management Strategy,* 6, 45-74.

Jenkins, M., Liu, P., Matzkin, R. L., & McFadden, D. L. (2004). The browser war: Econometric analysis of Markov perfect equilibrium in markets with network effects. Manuscript. University of California-Berkeley.

楠田康之（2018）．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―．日本福祉大学経済論集，56, 43-80.

久保幹雄，ジョア・ペドロ・ペドロソ，村松正和，アブドゥール・レイス（2012）．『あたらしい数理最適化：Python 言語と Gurobi で解く』近代科学社.

Luo, Y., Xiao, P., & Xiao, R. (2018). Identification of dynamic games with multiple equilibria and unobserved heterogeneity with application to fast food chains in China. Manuscript.

Maskin, E., & Tirole, J. (1988a). A theory of dynamic oligopoly, I: Overview and quantity competition

with large fixed costs. *Econometrica,* 56, 549-569.
Maskin, E., & Tirole, J. (1988b). A theory of dynamic oligopoly, II: Price competition, kinked demand curves, and Edgeworth cycles. *Econometrica,* 56, 571-599.
松村杏子，武藤茂夫，福田大輔，柳沼秀樹（2012）．混雑した都市鉄道における出発時刻選択モデルの構造推定：ゲーム理論に基づいた実証研究，土木計画学研究・講演集，45, paper no.165.
Pakes, A., & McGuire, P. (1994). Computing Markov-perfect Nash equilibria: Numerical implications of a dynamic differentiated product model. *RAND Journal of Economics,* 25, 555-589.
Pakes, A., & McGuire, P. (2001). Stochastic algorithms, symmetric Markov perfect equilibrium, and the 'curse' of dimensionality. *Econometrica,* 69, 1261-1281.
Pakes, A., Ostrovsky, M., & Berry, S. (2007). Simple estimators for the parameters of discrete dynamic games (with entry/exit examples). *RAND Journal of Economics,* 38, 373-399.
Pesendorfer, M., & Schmidt-Dengler, P. (2008). Asymptotic least squares estimators for dynamic games. *Review of Economic Studies,* 75, 901-928.
Pesendorfer, M., & Schmidt-Dengler, P. (2010). Sequential estimation of dynamic discrete games: A comment. *Econometrica,* 78, 833-842.
Seim, K. (2006). An empirical model of firm entry with endogenous product-type choices. *RAND Journal of Economics,* 37, 619-640.
Weintraub, G. Y., Benkard, C. L., & Roy, B. V. (2008). Markov perfect industry dynamics with many firms. *Econometrica,* 76, 1375-1411.

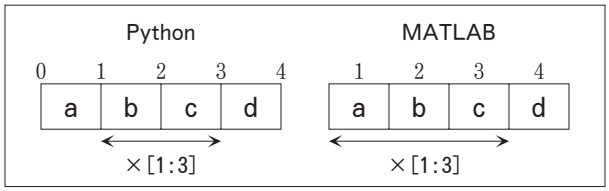
付録 A　Python についての覚書

Python は Numpy パッケージを利用することで，容易に配列演算ができる．しかし，MATLAB などに慣れている場合，気をつけなければならない点もある．今回プログラムを作った際に，覚えておきたいことを MATLAB と比較しながら以下に列挙した．

配列について

- 数値計算において配列（行列・ベクトル）の演算を行う場合，モジュール Numpy を用いる．プログラムの最初に，`import numpy as np` としておけば，`np.log(V1)` などとして Numpy の関数を使うことができる．（慣習的に numpy を np と略して記述する．）
- MATLAB の配列の添字は 1 から始まるのに対して，Python の配列は 0 から始まる．例えば，`a = np.arange(10)` として 10 個の整数からなる配列を作るとすると，［1, 2, . . . , 10］ではなく［0, 1, . . . , 9］ができる．これは，1 から始まる変数を用いる場合などで特に注意が必要である．例えば，本稿のモデルで用いた市場の状態 S＝{1, 2, 3, 4, 5} を表す配列を作れば，S[0], S[1], S[2], S[3], S[4] となる．
- 「スライシング」を用いれば，配列の複数の要素を抽出することができるが，添字の指定の仕方に注意すべきである．例えば，配列 `x=np.array(['a','b','c','d'])` から `x[1:3]` として要素を抽出する場合，MATLAB では，['a' 'b' 'c'] が抽出されるのに対し，Python では，['b' 'c'] となる．このことは，次の図のように，Python と MATLAB で，添字の意味

が違うことを考えれば理解できるだろう[33]．つまり，MATLAB で添字は要素をあらわすのに対し，Python では要素間の境界をあらわす．



- したがって，配列の一部を抽出する場合，MATLAB のやり方では次元が維持されないことがある．例えば，`A=np.array([[1,2,3],[4,5,6],[7,8,9]])` として下のような（3×3）行列を作り，この行列 A から最後の列（第 2 列）を取り出すために `b=A[:,2]` とすると，b は（3×1）列ベクトルではなくなる．

$$A=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

実際，`np.shape(b)` でサイズを確認すると，(3, 1) ではなく，(3,) と表示される．これを知らずに b を同じサイズの列ベクトルと足したりすれば，ブロードキャスト（後述）により（3×3）行列となるので注意が必要である．第 2 列を（3×1）列ベクトルとするには，`b=np.reshape(b,(3,1))` として次元を整えるか，あるいは `b=A[:,2]` ではなく `b=A[:,2:3]` として正しく取り出す必要がある．

- 次元の異なる配列同士の演算では，「ブロードキャスト」が行われる．例えば，`a=np.array([[1,2,3]])` として行ベクトルを作り，`b=np.array([[1],[2],[3]])` として列ベクトルを作ると，それらの足し算はエラーとはならずに，

$$a+b=\begin{bmatrix} (1+1) & (2+1) & (3+1) \\ (1+2) & (2+2) & (3+2) \\ (1+3) & (2+3) & (3+3) \end{bmatrix}=\begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$$

となる．これは，要素が欠けているところが自動的に補われて演算が行われるためである．

- 配列間の＊は，MATLAB では配列と配列の積をあらわすが，Python では要素間の積（アダマール積）をあらわす．Python で行列 A と B の内積の計算をするときは，`A*B` ではなく，`C=np.dot(A,B)` と書かなければならない．
- Python の辞書型配列を使えば，キーを使って要素を呼び出すことができる．例えば，

```
X = [ {'S': 1, 'a1': 0, 'a2': 0},
      {'S': 1, 'a1': 0, 'a2': 1},
      {'S': 1, 'a1': 1, 'a2': 0},
```

33　この図は，久保など（2012）を参考にした．

```
{'S': 1, 'a1': 1, 'a2': 1},
    ....
{'S': 5, 'a1': 1, 'a2': 0},
{'S': 5, 'a1': 1, 'a2': 1} ]
```

としておけば，xi 番目の X[xi]['S'] で S を呼び出すことができる．上の例では，上から 2 番目の状態は，X[1]['S']=1，X[1]['a1']=0，X[1]['a2']=1 となる．

その他の注意事項

- 変数から変数へ値をコピーするつもりで = を使ってはならない．例えば，P1 の値を P1prev にコピーしたいと考えて P1prev=P1 とすれば，その後で P1 の値を変更すると P1prev の値も自動的に変更される．つまり，P1prev=P1 は値のコピーではなく，値を格納する場所のコピーと考えるべきである．もし，値のみをコピーするのならば，P1prev=np.copy(P1) としなければならない．
- 関数の中からグローバル変数を参照することができる．例えば，上記のように X という辞書型配列を定義しておけば，引数を指定せずに他の関数の中で使うことができる．つまり，

```
def func_pi1(P2,THETA):
    ....
  pi1[xi]=np.log(X[xi]['S'])- ....
```

などとできる．ただし，関数の中でできるのは参照のみであり，関数の中で参照したグローバル変数に値を代入することはできない．

- インデックスに使う変数は整数型である．そうでない変数をインデックスとして使う場合は，int(xi) のように整数型に変えておく．

付録 B　AMPL と KNITRO について

この付録では，モデリング言語 AMPL とソルバー KNITRO について解説する．「モデリング言語」とは，数理モデルを記述するのに適した言語であり，問題を解く「ソルバー」と組み合わせて使うことで，比較的容易に最適化問題などの数理モデルの解を求めることができる．本稿の推定では，AMPL と KNITRO を用いたが，最近では，Python でも PuLP や Pyomo などのモジュールが利用可能である．

AMPL とは

AMPL とは，モデリング言語の一種である．AMPL は有償プログラムであるが，試用目的で，Windows，Linux，macOS 用の無償のデモ版が利用できる（2018 年 5 月現在）．デモ版の制限は，

- 線形計画問題では，500 個の変数，500 本の制約条件式＋目的関数

●非線形計画問題では，300 個の変数，500 本の制約条件式＋目的関数

となっている．また，同じ場所より，AMPL IDE もダウンロードすることができる．AMPL IDE は，AMPL のための統合開発環境である．

一方，AMPL で使うことができるソルバーとしては，CPLEX や Gurobi（線形計画問題）や，KNITRO や MINOS（非線形計画問題）などがある．デモ版でもこのようなソルバーは利用可能であるが，KNITRO のように制限がある場合もある（KNITRO の場合は，10 個の変数，10 本の制約条件式＋目的関数）．

AMPL に関する情報は，Fourer, Gay, and Kernighan（2002）が詳しいが，インターネット上でも https://ampl.com/resources/the-ampl-book/chapter-downloads/を読むことができる（2018 年 5 月現在）．

最適化問題の例

実際の最適化問題は，次のように記述することができる．

$$
\begin{aligned}
\text{minimize} \quad & 3x+4y \\
\text{subject to:} \quad & 5x+6y \geq 10 \\
& 7x+5y \geq 5 \\
& x, y \geq 0
\end{aligned}
$$

この問題は，変数 x, y に関する線形最適化問題である．この問題を AMPL で記述すれば，次のように書ける．

```
var x;
var y;
minimize obj : 3*x + 4*y;
subject to con1 : 5*x + 6*y >= 10;
subject to con2 : 7*x + 5*y >= 5;
subject to con3 : x >= 0;
subject to con4 : y >= 0;
```

つまり，目的関数 obj と，制約条件式 con1，con2，con3，con4 を記述するだけでよい．さらに，非負条件 con3，con4 は，最初の変数宣言のところで

```
var x >= 0;
var y >= 0;
```

と書けば省略できる．このようなモデルをテキストファイルに記述し，"sample.mod" と名付けておこう．AMPL はこのようなモデルファイルを読みこんで，このモデルの解を求める．

コマンドラインを使った AMPL の実際

それでは，この問題を AMPL を用いて解いてみよう．AMPL（デモ版）を起動すると，次の

ようなコマンドラインが入力待ちとなる．

```
ampl:
```

このコマンドラインに次のように入力する．

```
ampl: reset;
ampl: model sample.mod;
ampl: solve;
```

つまり，まず変数の値などをリセットし，"sample.mod" を読み込み，問題を解かせる．すると，次のようなメッセージがあらわれる．

```
MINOS 5.51: optimal solution found.
0 iterations, objective 6
```

そこで，次のように，解 x，y とそのときの目的関数の値を表示させる．

```
ampl: display x, y, obj;
```

結果は，

```
x = 2
y = 0
obj = 6
```

となる．ここで，デモ版では何もソルバーを指定しなければ，MINOS というソルバーが問題を解いている．もし，別のソルバー（例えば，KNITRO）を用いる場合は，`solve;` の前に

```
ampl: option solver knitro;
```

とすればよい．なお，KNITRO のマニュアルとしては，
https://www.artelys.com/tools/knitro_doc/ などを参照のこと．

付録 C　AMPL のプログラムコード（MPEC 用）

```
param beta;
param M;      # 市場の数（データ数）
param X_old {1..M};  # 各市場で観察された状態 X
param S_old {1..M};  # 各市場で観察された市場の状態 S
param a1_old {1..M}; # 各市場で観察された企業 1 の前期の行動 a1(t-1)
param a2_old {1..M}; # 各市場で観察された企業 2 の前期の行動 a2(t-1)
param a1_new {1..M}; # 各市場で観察された企業 1 の今期の行動 a1(t)
param a2_new {1..M}; # 各市場で観察された企業 2 の今期の行動 a2(t)

set XSET;
param S {XSET};      # 市場の状態 S
param a1 {XSET};     # 企業 1 の前期の行動 a1(t-1)
param a2 {XSET};     # 企業 2 の前期の行動 a2(t-1)
param F_dat {XSET,XSET};    # データから推定した推移確率

var thetaRS >= 0;    # 需要シフトパラメーター
var thetaRN >= 0;    # 需要の傾きパラメーター
var thetaEC >= 0;    # 参入費用パラメーター
var thetaFC1 >= 0;   # 企業 1 の固定費用パラメーター
var thetaFC2 >= 0;   # 企業 2 の固定費用パラメーター

var P1 {XSET} >= 0, <= 1;
var P2 {XSET} >= 0, <= 1;
var V1 {XSET};
var V2 {XSET};

maximize L : sum {im in 1..M} (
                          # 企業 1
                          log(
                            # 分子の部分
                              exp(
                                  a1_new[im]*( # 次の期に a1=1 のときだけ
                                    thetaRS*log(S_old[im])
                                   -thetaRN*(P2[X_old[im]]*log(2)+(1-P2[X_old[im]])*log(1))
                                   -thetaEC*(1-a1_old[im])
                                   -thetaFC1
                                   )
                                  +beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V1[xj])
                                  ) # 分子ここまで
                                 /
                            # 分母の部分
                              (
                              # 分母第 1 項
                              exp(
                                  thetaRS*log(S_old[im])
                                  -thetaRN*(P2[X_old[im]]*log(2)+(1-P2[X_old[im]])*log(1))
                                  -thetaEC*(1-a1_old[im])
                                  -thetaFC1
                                  +beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V1[xj])
                                  ) # 第 1 項ここまで
                              # 分母第 2 項
                              +exp(
                                  0+beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V1[xj])
                                  ) # 第 2 項ここまで
                              +0.000001) # 分母ここまで
                             +0.000001) # 企業 1 ここまで
                          +
                          # 企業 2
                          log(
                            # 分子の部分
                              exp(
                                  a2_new[im]*( # 次の期に a2=1 のときだけ
                                    thetaRS*log(S_old[im])
                                   -thetaRN*(P1[X_old[im]]*log(2)+(1-P1[X_old[im]])*log(1))
                                   -thetaEC*(1-a2_old[im])
```

```
                                    -thetaFC2
                                    )
                                   +beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V2[xj])
                                   ) # 分子ここまで
                                  /
                                # 分母の部分
                                  (
                                  # 分母第１項
                                  exp(
                                      thetaRS*log(S_old[im])
                                     -thetaRN*(P1[X_old[im]]*log(2)+(1-P1[X_old[im]])*log(1))
                                     -thetaEC*(1-a2_old[im])
                                     -thetaFC2
                                     +beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V2[xj])
                                     ) # 第１項ここまで
                                  # 分母第２項
                                  +exp(
                                     0+beta*(sum {xj in XSET} F_dat[X_old[im],xj]*V2[xj])
                                     ) # 第２項ここまで
                                  +0.000001) # 分母ここまで
                                 +0.000001) # 企業２ここまで
                               );

subject to Lambda1 {xi in XSET} : P1[xi]
                                = exp(
                                     thetaRS*log(S[xi])
                                     -thetaRN*(P2[xi]*log(2)+(1-P2[xi])*log(1))
                                     -thetaEC*(1-a1[xi])
                                     -thetaFC1
                                     +beta*(sum {xj in XSET} F_dat[xi,xj]*V1[xj])
                                     )/
                                  (exp(
                                     thetaRS*log(S[xi])
                                     -thetaRN*(P2[xi]*log(2)+(1-P2[xi])*log(1))
                                     -thetaEC*(1-a1[xi])
                                     -thetaFC1
                                     +beta*(sum {xj in XSET} F_dat[xi,xj]*V1[xj])
                                     )
                                  +exp(
                                     0+beta*(sum {xj in XSET} F_dat[xi,xj]*V1[xj])
                                     )
                                  +0.000001);

subject to Lambda2 {xi in XSET} : P2[xi]
                                = exp(
                                     thetaRS*log(S[xi])
                                     -thetaRN*(P1[xi]*log(2)+(1-P1[xi])*log(1))
                                     -thetaEC*(1-a2[xi])
                                     -thetaFC2
                                     +beta*(sum {xj in XSET} F_dat[xi,xj]*V2[xj])
                                     )/
                                  (exp(
                                     thetaRS*log(S[xi])
                                     -thetaRN*(P1[xi]*log(2)+(1-P1[xi])*log(1))
                                     -thetaEC*(1-a2[xi])
                                     -thetaFC2
                                     +beta*(sum {xj in XSET} F_dat[xi,xj]*V2[xj])
                                     )
                                  +exp(
                                     0+beta*(sum {xj in XSET} F_dat[xi,xj]*V2[xj])
                                     )
                                  +0.000001);

subject to Phi1 {xi in XSET} : V1[xi]
                                = P1[xi]*(
                                     thetaRS*log(S[xi])
                                     -thetaRN*(P2[xi]*log(2)+(1-P2[xi])*log(1))
                                     -thetaEC*(1-a1[xi])
```

```
                                -thetaFC1
                                +beta*(sum {xj in XSET} F_dat[xi,xj]*V1[xj])
                                -log(P1[xi]+0.000001)
                                   )
                           +(1-P1[xi])*(
                                0+beta*(sum {xj in XSET} F_dat[xi,xj]*V1[xj])
                                -log(1-P1[xi]+0.000001)
                                   );

subject to Phi2 {xi in XSET} : V2[xi]
                             = P2[xi]*(
                                thetaRS*log(S[xi])
                                -thetaRN*(P1[xi]*log(2)+(1-P1[xi])*log(1))
                                -thetaEC*(1-a2[xi])
                                -thetaFC2
                                +beta*(sum {xj in XSET} F_dat[xi,xj]*V2[xj])
                                -log(P2[xi]+0.000001)
                                   )
                           +(1-P2[xi])*(
                                0+beta*(sum {xj in XSET} F_dat[xi,xj]*V2[xj])
                                -log(1-P2[xi]+0.000001)
                                   );
```

プログラム対照表

変数/関数	コード上の名前	値	備　考
β	beta	0.95	
M	M	200	
x_{mt}	X_old	—	$=(S_{mt},\ a_{1m,t-1},\ a_{2m,t-1})$
S_{mt}	S_old	—	$(\lvert M\rvert \times 1)$ ベクトル
$a_{1m,t-1}$	a1_old	—	$(\lvert M\rvert \times 1)$ ベクトル
a_{1mt}	a1_new	—	$(\lvert M\rvert \times 1)$ ベクトル
X	XSET	$[1, \ldots, 20]$	$(\lvert X\rvert \times 1)$ ベクトル
S	S	欄外	$(\lvert X\rvert \times 1)$ ベクトル
a_1	a1	欄外	$(\lvert X\rvert \times 1)$ ベクトル
a_2	a2	欄外	$(\lvert X\rvert \times 1)$ ベクトル
$f(x_{m,t+1}\lvert x_{mt})$	F_dat	—	$(\lvert X\rvert \times \lvert X\rvert)$ 行列
θ_{RS}	thetaRS	—	真の値 1.0
θ_{RN}	thetaRN	—	真の値 2.0
θ_{EC}	thetaEC	—	真の値 0.1
$\theta_{FC,1}$	thetaFC1	—	真の値 1.5
$\theta_{FC,2}$	thetaFC2	—	真の値 1.9
$\bar{V}_1(x;\mathbf{P}_2)$	V1	—	$(\lvert X\rvert \times 1)$ ベクトル
$P_1(a_1=1\lvert x)$	P1	—	$(\lvert X\rvert \times 1)$ ベクトル
$P_1(a_1=0\lvert x)$	1-P1	—	$(\lvert X\rvert \times 1)$ ベクトル
$L((\mathbf{x},\mathbf{a}),\bar{\mathbf{V}},\mathbf{P})$	L	—	目的関数
$P_1(a_1=1\lvert x)=\Lambda_1(a_1=1\lvert x;\mathbf{P}_2)$	Lambda1	—	制約条件式（$\lvert X\rvert$本）
$\bar{V}_1(x;\mathbf{P}_2)=\Phi_1(x;\bar{V}_1,\mathbf{P})$	Phi1	—	制約条件式（$\lvert X\rvert$本）

$S=$ [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5]
$a_1=$ [0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1]
$a_2=$ [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]

楠田（2018）「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―」の訂正

p.55　誤：

$$\ell(\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{t=1}^{T_i} \ln P(a_{it},\, x_{it};\, \boldsymbol{\theta}_u,\, \boldsymbol{\theta}_g,\, \boldsymbol{\theta}_f) + \prod_{i=1}^{N}\prod_{t=1}^{T_i-1} \ln f(x_{i,\,t+1} \mid x_{it},\, a_{it};\, \boldsymbol{\theta}_f) \tag{20}$$

正：

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{t=1}^{T_i} \ln P(a_{it},\, x_{it};\, \boldsymbol{\theta}_u,\, \boldsymbol{\theta}_g,\, \boldsymbol{\theta}_f) + \sum_{i=1}^{N}\sum_{t=1}^{T_i-1} \ln f(x_{i,\,t+1} \mid x_{it},\, a_{it};\, \boldsymbol{\theta}_f) \tag{20}$$

p.63　誤：Hotz and Millerz（2010）
正：Hotz and Miller（1993）

p.64　誤：$\mathbf{EV} = T_{\boldsymbol{\theta}}(\mathbf{EV})$　(29)
正：$\mathbf{EV} = T(\mathbf{EV};\, \boldsymbol{\theta})$　(29)

同　誤：subject to: $\mathbf{EV} = T_{\boldsymbol{\theta}}(\mathbf{EV})$.　(31)
正：subject to: $\mathbf{EV} = T(\mathbf{EV};\, \boldsymbol{\theta})$.　(31)