

Learning in experimental 2×2 games の再現と考察

07-161048

齊藤一織

夏の合宿で実験の再現を行った Stationary Concepts for Experimental 2x2-Games(Reinhard Selten Thorsten Chmura,2008) の内容を纏め、さらに発展である Learning in experimental 2×2 games(Chmura,Goerg,Selten,2011)で行われた実験を再現し、さらに追加の実験を行いその考察をする。

ナッシュ均衡は各プレーヤーが合理的行動を行った結果生じる均衡であるが、実際のプレーヤーの行動の結果はナッシュ均衡とは必ずしも一致しないということが実験によって知られている。Selten,Chmura (以下 SC) は、一つの混合戦略均衡のみを持つ 12 種類の 2×2 ゲームの実験を行い、ナッシュ均衡 (以下 NE)、質的応答均衡 (以下 QRE)、action-sampling equilibrium(以下 ASE)、payoff-sampling equilibrium(以下 PSE)、impulse balance equilibrium(以下 IBE)の 5 つの均衡概念で比較を行い、その結果 IBE、PSE、ASE、QRE、NE の順に予測として優れている、という結論を出した。(これに対し、Brunner,Camerer,Goeree らは、SC の論文内の QRE と ASE のパラメータ設定の不備を指摘し修正を加えたところ、QRE と ASE は他の NE 以外の定常概念とほぼ変わらない結果を示した。)

【論文内容のまとめ】

・ゲームの構造

今回分析の対象となるのは、図 1 で示されるようなそれぞれの純粋戦略が正の確率で用いられるような一つの均衡点しか持たない完全混合 2×2 ゲームである。行側のプレーヤーをプレーヤー1、列側のプレーヤーをプレーヤー2 とする。 a_L, a_R, b_U, b_D などの利得パラメータは全て非負である。 c_L, c_R は U,D を選んだ際のプレーヤー1 の利得の差であり、同様に d_U, d_D は L,R を選んだ際のプレーヤー2 の利得の差であり、 c_L, c_R, d_U, d_D は全て正の数である。図 1 の周囲の矢印は最適反応の方向である。矢印が反時計回りとなる完全混合 2×2 ゲームも考えられるが、その場合行を交換することで図 1 のような形式に変換できるため、今回は図 1 の構造のゲームのみを対象とする。ここで、 $p = (p_U, p_D)$ と $q = (q_L, q_R)$ を、それぞれプレーヤー1 と 2 の混合戦略とする。 p_U と p_D はプレーヤー1 が U または D をとる確率を、 q_L と q_R はプレーヤー2 が L または R を選択する確率を指す。

・ナッシュ均衡

混合戦略ナッシュ均衡の場合、 p_U, p_D, q_L, q_R は以下の通りになる。

$$p_U = \frac{d_D}{d_U + d_D}, p_D = \frac{d_U}{d_U + d_D}, q_L = \frac{c_R}{c_L + c_R}, q_R = \frac{c_L}{c_L + c_R} \cdots (1)$$

ナッシュ均衡は最適反応の構造にのみ依存するため、これは利得の差 c_L, c_R, d_U, d_D にのみ依存する。

・質的応答均衡

質的応答均衡(Richard D. McKelvey and Thomas R. Palfrey 1995)では、各プレーヤーが行動選択時にミスをするものと仮定されている。 $E_U(q)$ と $E_R(q)$ をそれぞれ q に対して U または D を選んだ際のプレーヤー1 の期待利得とし、同様に $E_L(p)$ と $E_R(p)$ をそれぞれ p に対して L または R を選んだ際のプレーヤー2 の期待利得とする。 p_U と q_L は以下のように定義される。

$$p_U = \frac{e^{\lambda E_U(q)}}{e^{\lambda E_U(q)} + e^{\lambda E_D(q)}}, q_L = \frac{e^{\lambda E_L(p)}}{e^{\lambda E_L(p)} + e^{\lambda E_R(p)}} \cdots (2)$$

λ は合理性の程度を示すパラメータとして解釈される。 $\lambda \rightarrow 0$ となるとプレーヤーは完全に非合理的となり、

どの戦略も等確率でプレーし、 $\lambda \rightarrow \infty$ となると完全に合理的となり、ナッシュ均衡に近づく。

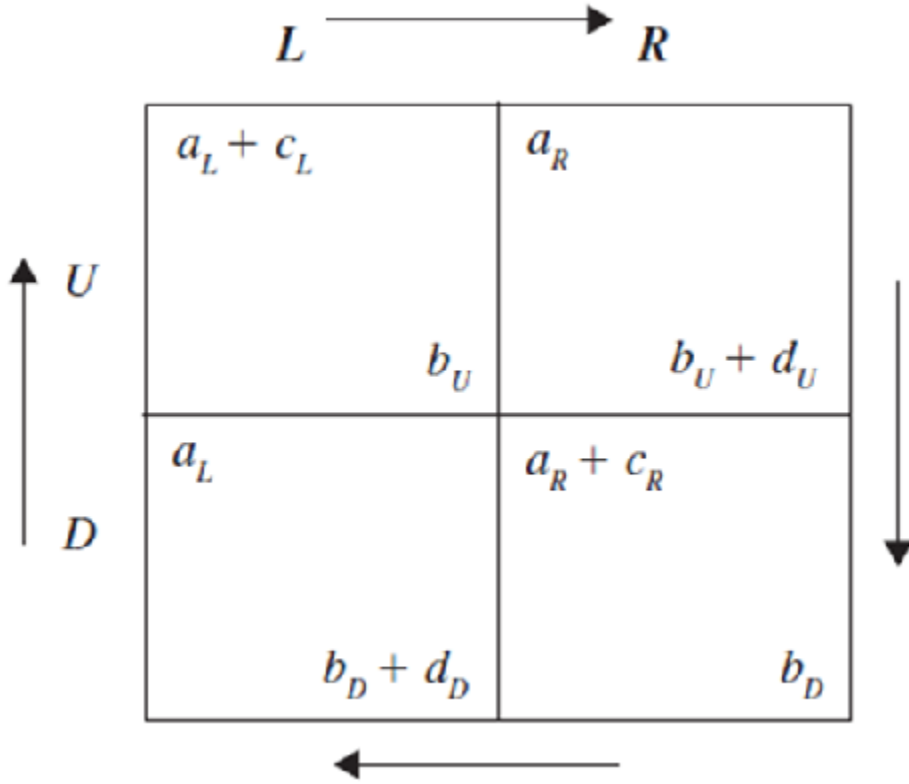


図 1 (Selten, Chumra, (2008) Figure1 より)

• action-sampling equilibrium

ASE は、プレーヤーは相手がプレーした戦略について n 個のサンプルを取り、このサンプルに対して最適な行動を取る、というものである。 k をプレーヤー1が取ったサンプル内の L の数とし、 m をプレーヤー2のサンプル内の D の数とする。プレーヤー1 は、 $U, D, (1/2, 1/2)$ をそれぞれ $kc_L > (n-k)c_R, kc_L < (n-k)c_R, kc_L = (n-k)c_R$ の時にプレーする。プレーヤー2 は $L, R, (1/2, 1/2)$ をそれぞれ $md_D > (n-m)d_U, md_D < (n-m)d_U, md_D = (n-m)d_U$ の時にプレーする。 $\alpha_U(k)$ を k に対してプレーヤー1 が U を選択する確率とし、 $\alpha_L(m)$ を m に対してプレーヤー2 が L を選択する確率とすると、 $\alpha_U(k), \alpha_L(m)$ は以下ようになる。

$$\alpha_U(k) = \begin{cases} 1 & \text{for } \frac{k}{n} > \frac{c_R}{c_L + c_R} \\ \frac{1}{2} & \text{for } \frac{k}{n} = \frac{c_R}{c_L + c_R} \\ 0 & \text{else} \end{cases}, \quad \alpha_L(m) = \begin{cases} 1 & \text{for } \frac{m}{n} > \frac{d_U}{d_U + d_D} \\ \frac{1}{2} & \text{for } \frac{m}{n} = \frac{d_U}{d_U + d_D} \\ 0 & \text{else} \end{cases}$$

以上より、 p_U と q_L は以下の通りになる。

$$p_U = \sum_{k=0}^n \binom{n}{k} q_L^k (1 - q_L)^{n-k} \alpha_U(k), \quad q_L = \sum_{m=0}^n \binom{n}{m} (1 - p_U)^m p_U^{n-m} \alpha_L(m)$$

ASE は差 c_L, c_R, d_U, d_D にのみ依存する。したがって、ASE は最適反応の構造にのみ依存する。

• payoff-sampling equilibrium

PSE (Osborne and Rubinstein 1998) では、各プレーヤーが自身の純粋戦略のそれぞれについて相手の行動のサンプルを n 個取り、2 つのサンプルでの利得の合計を比較し、利得の合計が高いほうの戦略をプレーする。 k_U と k_D をそれぞれ U と D に対するプレーヤー1 のサンプルにおける L の数とする。同様に、 m_L と m_R をそれぞれ

L と R に対するプレーヤー2 のサンプルにおける U の数とする。

U と D のサンプルにおけるプレーヤー1 のそれぞれの利得の合計 H_U と H_D は以下の通りである。

$$H_U = k_U(a_L + c_L) + (n - k_U)a_R, H_D = k_D a_L + (n - k_D)(a_R + c_R)$$

同様に、L と R のサンプルにおけるプレーヤー2 のそれぞれの利得の合計 H_L と H_R は以下のようになる。

$$H_L = m_U b_U + (n - m_L)(b_U + d_D), H_D = m_R(b_U + d_U) + (n - m_R)b_D$$

k_U と k_D がサンプル内の L の数である時のプレーヤー1 が U をプレーする確率 $\beta(k_U, k_D)$ と、 m_L と m_R が L と R に対するサンプル内の U の数である時のプレーヤー2 が L をプレーする確率 $\gamma(m_L, m_R)$ は以下のようになる。

$$\beta(k_U, k_D) = \begin{cases} 1 & \text{for } H_U > H_D \\ \frac{1}{2} & \text{for } H_U = H_D \\ 0 & \text{else} \end{cases}, \gamma(m_L, m_R) = \begin{cases} 1 & \text{for } H_L > H_R \\ \frac{1}{2} & \text{for } H_L = H_R \\ 0 & \text{else} \end{cases}$$

k_U, k_D, m_L, m_R は二項分布であるため、PSE における p_U と q_L の式は以下の二つで示される。

$$p_U = \sum_{k_U=0}^n \sum_{k_D=0}^n \binom{n}{k_U} \binom{n}{k_D} q_L^{k_U+k_D} (1 - q_L)^{2n-k_U-k_D} \beta(k_U, k_D)$$

$$q_L = \sum_{m_L=0}^n \sum_{m_R=0}^n \binom{n}{m_L} \binom{n}{m_R} (1 - p_U)^{m_L+m_R} p_U^{2n-m_L-m_R} \gamma(m_L, m_R)$$

・ impulse-balance equilibrium

IBE は Learning Direction Theory(Selten and Joachim Buchta 1999)に基づくものである。ゲームをプレーした後、各プレーヤーは他の戦略を取っていればより高い利得を得られていた場合に、他の戦略の方向への「衝動」をフィードバックとして受ける。この時、「こっちを選択していればより高い利得をえられたのに」という衝動よりも、「こっちを選択していれば損はしなかったのに」という損失回避的な衝動の方が重視される。これはプロスペクト理論(Kahnemann and Tversky 1979)などが元となった考えである。この二つを元に、自身の純粋戦略の衝動が等しくなるように p ないし q を選択するのが IBE である。

各プレーヤーには最低限保証される利得があり、それはプレーヤーの純粋戦略のマクシミン値、つまり純粋戦略のうち一つを用いて得られる最小の利得の最大値であり、以下の通り定義される。

$$s_1 = \max[\min(a_L + c_L, a_R), \min(a_L, a_R + c_R)]$$

$$s_2 = \max[\min(b_U, b_D + d_D), \min(b_U + d_U, b_D)]$$

s_1, s_2 はそれぞれプレーヤー1、2 のマクシミン値となる。ここで、マクシミン値よりも大きい利得の価値を下げる、つまり損失回避の方をより重視する、という操作を行う。ゲームの利得のうち、 s_i 以下のものはそのままにし、 s_i より大きい利得 x は $x^* = x - \frac{x-s_i}{2}$ と変形する。これは損失が収入の 2 倍の重さを持つように変形されたゲームとなる。利得の差 c_L, c_R, d_L, d_R は変形されたゲームでは $c_L^*, c_R^*, d_U^*, d_D^*$ と表される。プレー後、プレーヤー*i* が他の戦略を取っていればより高い利得を得られていた場合、他の戦略の方向の衝動を受ける。この衝動が利得の差 c^*, d^* である。impulse balance 均衡では、プレーヤー1 の U から D までの期待衝動は D から U までの期待衝動と等しくならなければならない。同様に、プレーヤー2 の L から R までの期待衝動は R から L までの期待衝動と一致しなければならない。これによって、以下の二つの impulse balance 方程式が得られる。

$$p_U q_R c_R^* = p_D q_L c_L^*, \quad p_U q_L d_U^* = p_D q_R d_D^*$$

よって、IBE では p_U, q_L は以下の二つの方程式で表される。

$$p_U = \frac{q_L c_L^*}{q_L c_L^* + (1 - q_L) c_R^*}, q_L = \frac{(1 - p_U) d_D^*}{p_U d_U^* + (1 - p_U) d_D^*}$$

その後 Chumra, Goerg, Selten(2011)は、定常状態は学習過程の結果として生み出されるものであることか

ら、単純な学習モデルを用いて行動の予測がなされるかの検証を行った。impulse-matching learning と Action sampling learning の二つを提案し、比較として reinforcement learning と self-tuning EWA も紹介されている。今回はこの検証を再現しつつ、追加の実験を行う。まず、論文内で検証されている学習モデルの紹介をする。

・ impulse-matching learning

impulse-matching learning は上記の IBE と Learning Direction Theory を元にしたモデルである。IBE と同様にゲームの利得を変形させる。純粋戦略 j から i の方向の衝動の $t-1$ 期までの合計を $R_i(t)$ とする。 t 期に行動 1 または 2 が選択される確率は以下ようになる。

$$p_i(t) = \frac{R_i(t)}{R_1(t) + R_2(t)}, \text{ for } i = 1, 2$$

t 期においての純粋戦略 j から i への衝動は以下の通りである

$$r_i(t) = \max[0, \pi_i - \pi_j], \text{ for } i, j = 1, 2 \text{ and } i \neq j$$

π_i は変形後のゲームにおいて i を選択した時に得られる利得である。よって、 $t+1$ 期において衝動は以下のように更新される。

$$R_i(t+1) = R_i(t) + r_i(t)$$

第 1 期では $R_1(1) = R_2(1) = 0$ で固定され、どちらかの衝動が 0 より大きくなるまで $p_1(t) = p_2(t) = 0.5$ で固定される。

・ action-sampling learning

action-sampling learning は上記の ASE を元にした学習モデルである。各プレーヤーは t 期において、相手の $t-1$ 期までの行動履歴から n 回分の行動 a_1, \dots, a_n のサンプルをランダムに取る。 $\pi_i(a_j)$ を、相手の行動が a_j だった際の行動 i による利得とする。純粋戦略 $i=1, 2$ について、 $P_i(t) = \sum_{j=1}^n \pi_i(a_j)$ を各純粋戦略におけるサンプル内の利得の合計とする。したがって、 t 期において純粋戦略 i をとる確率は、

$$p_i(t) = \begin{cases} 1 & \text{if } P_i(t) > P_j(t) \\ 0.5 & \text{if } P_i(t) = P_j(t) \text{ for } i, j = 1, 2 \text{ and } i \neq j \\ 0 & \text{else} \end{cases}$$

となる。相手が 1, 2 の両方の戦略をとるまでは、 $p_1(t) = p_2(t) = 0.5$ で固定する。

・ reinforcement learning

reinforcement learning では、プレーヤーは各純粋戦略 i について利得の合計 $B_i(t)$ を以下のように設定する。

$$B_i(t+1) = \begin{cases} B_i(t) + \pi(t) & \text{if action } i \text{ was chosen in } t \\ B_i(t) & \text{else} \end{cases}$$

$\pi(t)$ は t 期に得られた利得を指す。第 1 期に両方の戦略が等確率で取られた後、純粋戦略 i が t 期に取られる確率は

$$p_i(t) = \frac{B_i(t)}{B_1(t) + B_2(t)}$$

となる。このモデルは全ての利得が非負であり、かつ各プレーヤーの利得のうち少なくとも 1 か所は正であることが前提となっている。reinforcement learning はナッシュ均衡に変換できる (Beggs, 2005) ため、以下では reinforcement learning をナッシュ均衡と同等の概念として取り扱う。

・ self-tuning experience-weighted learning

self-tuning EWA は experience-weighted attraction model (Camerer & Ho, 1999) を元にした概念であり、

Ho, Camerer & Chong(2007)によって紹介された。 p_i は各戦略 i がどの程度魅力的かを示す $A_i(t)$ に基づいて行われる。 $A_i(t)$ はプレーヤーが戦略 s_i を取り、相手が戦略 $s^m(t)$ を選択した際に得られる利得 $\pi(s_i, s^m(t))$ に基づいて決まる。 $A_i(t)$ の関数は経験による重みづけを行う $N(t)$ 、変化決定関数 $\phi(t)$ 、注意関数 $\delta(t)$ に依存し、以下のようになる。

$$A_i(t) = \frac{\phi(t)N(t-1)A_i(t-1) + [\delta(t) + (1 - \delta(t))I(s_i, s(t))]\pi(s_i, s^m(t))}{N(t)}$$

$I(x, y)$ は $s(t) = s_i$ の時に 1 を、それ以外の時に 0 を返す指示関数である。前期の A_i をどの程度重く見るかを決定する $N(t)$ は以下のように定義される。

$$N(t) = N(t-1)\phi(t-1) + 1, N(0) = 1$$

変化決定関数 $\phi(t)$ はプレーヤーの環境がどの程度変わったかを示す。 $\phi(t)$ は以下のように定義される。

$$\phi(t) = 1 - \frac{1}{2}S(t)$$

$S(t)$ は surprise index であり、マッチした相手プレーヤーの最近の行動が過去の全ての行動からどの程度逸脱しているのかを測っている。surprise index は相手プレーヤーの行動が $t-1$ 期から t 期にかけて変化しなければ 0、そうでなければ 1 となる。つまり相手が $t-1$ 期と t 期で同じ行動をしていれば $\phi(t) = 1$ となり、違う行動をしていれば 0.5 となる。注意関数 $\delta(t)$ は過去の利得の重みづけを行い、より高い利得を得ていたであろう戦略へと注意を向ける。単一の混合戦略均衡を持つゲームではこれらの利得は $1/W$ で重みづけされる。 W は均衡でプレーされる戦略の個数である。よって、 2×2 ゲームでは $W=2$ となる。

$$\delta(t) = \begin{cases} \frac{1}{W} & \text{if } \pi(s_j, s_m(t)) > \pi(t) \\ 0 & \text{else} \end{cases}$$

self-tuning EWA の注意関数 $\delta(t)$ は learning-direction theory の考えを汲むものである。

t 期に戦略 i が取られる確率はロジット反応関数のように計算される。

$$p_i(t) = e^{\lambda A_i(t-1)} / \sum_{j=1}^2 e^{\lambda A_j(t-1)}$$

λ は反応感度であり、これは実際のデータに合わせる必要がある。実験開始時は $p_1 = p_2 = 0.5$ とする。

【実験内容】

Selten, Chumra, (2008)で行われた実験は図 2 の 12 通りの 2×2 ゲームである。Game1~6 (左列) が定和ゲーム、Game7~12 が非定和ゲームとなっている。Game1 と Game7 のように隣り合ったゲームは同じ最適反応構造をもっており、ナッシュ均衡と QRE、ASE が一致する。Selten, Chumra, (2008)による実験では、各ゲーム 8 人のグループでセッションが行われた。1 セッションは 200 ラウンドで、8 人のうち 4 人が行側のプレーヤー、残りの 4 人が列側のプレーヤーとなり、毎ラウンドごとにランダムな相手とゲームを行う。プレーヤーは 1 ラウンド終了ごとに相手プレーヤーの選択した戦略、自分が今期に得た利得、現在のラウンド数、自分がこれまでに得た利得についてのフィードバックを受ける。Game1~6 の定和ゲームは各ゲーム 12 のグループで実験が行われ、非定和ゲームは各ゲーム 6 グループで行われた。被験者はボン大学の主に経済学を専攻している学生で、合計 864 人が実験に参加したことになる。

Chmura, Goerg, Selten(2011)では、人間の代わりにコンピュータに参加者として実験を行っている。グループ内の各参加者は特定の学習モデル（全参加者で共通）に従い行動する。200 ラウンド（1 セッション）終了ごとに U と L の相対度数を測定し、Selten, Chumra, (2008)の実験で得られた相対度数との距離の二乗値をとる。各学習モデルは確率的な部分を含むため、ずれを解消するために 1 種類のゲームにつき 500 セッション行い、最終

Constant sum games		Nonconstant sum games																	
Game 1	<table><tr><td>10</td><td>0</td></tr><tr><td>8</td><td>18</td></tr><tr><td>9</td><td>10</td></tr><tr><td>9</td><td>8</td></tr></table>	10	0	8	18	9	10	9	8	Game 7	<table><tr><td>10</td><td>4</td></tr><tr><td>12</td><td>22</td></tr><tr><td>9</td><td>14</td></tr><tr><td>9</td><td>8</td></tr></table>	10	4	12	22	9	14	9	8
10	0																		
8	18																		
9	10																		
9	8																		
10	4																		
12	22																		
9	14																		
9	8																		
Game 2	<table><tr><td>9</td><td>0</td></tr><tr><td>4</td><td>13</td></tr><tr><td>6</td><td>8</td></tr><tr><td>7</td><td>5</td></tr></table>	9	0	4	13	6	8	7	5	Game 8	<table><tr><td>9</td><td>3</td></tr><tr><td>7</td><td>16</td></tr><tr><td>6</td><td>11</td></tr><tr><td>7</td><td>5</td></tr></table>	9	3	7	16	6	11	7	5
9	0																		
4	13																		
6	8																		
7	5																		
9	3																		
7	16																		
6	11																		
7	5																		
Game 3	<table><tr><td>8</td><td>0</td></tr><tr><td>6</td><td>14</td></tr><tr><td>7</td><td>10</td></tr><tr><td>7</td><td>4</td></tr></table>	8	0	6	14	7	10	7	4	Game 9	<table><tr><td>8</td><td>3</td></tr><tr><td>9</td><td>17</td></tr><tr><td>7</td><td>13</td></tr><tr><td>7</td><td>4</td></tr></table>	8	3	9	17	7	13	7	4
8	0																		
6	14																		
7	10																		
7	4																		
8	3																		
9	17																		
7	13																		
7	4																		
Game 4	<table><tr><td>7</td><td>0</td></tr><tr><td>4</td><td>11</td></tr><tr><td>5</td><td>9</td></tr><tr><td>6</td><td>2</td></tr></table>	7	0	4	11	5	9	6	2	Game 10	<table><tr><td>7</td><td>2</td></tr><tr><td>6</td><td>13</td></tr><tr><td>5</td><td>11</td></tr><tr><td>6</td><td>2</td></tr></table>	7	2	6	13	5	11	6	2
7	0																		
4	11																		
5	9																		
6	2																		
7	2																		
6	13																		
5	11																		
6	2																		
Game 5	<table><tr><td>7</td><td>0</td></tr><tr><td>2</td><td>9</td></tr><tr><td>4</td><td>8</td></tr><tr><td>5</td><td>1</td></tr></table>	7	0	2	9	4	8	5	1	Game 11	<table><tr><td>7</td><td>2</td></tr><tr><td>4</td><td>11</td></tr><tr><td>4</td><td>10</td></tr><tr><td>5</td><td>1</td></tr></table>	7	2	4	11	4	10	5	1
7	0																		
2	9																		
4	8																		
5	1																		
7	2																		
4	11																		
4	10																		
5	1																		
Game 6	<table><tr><td>7</td><td>1</td></tr><tr><td>1</td><td>7</td></tr><tr><td>3</td><td>8</td></tr><tr><td>5</td><td>0</td></tr></table>	7	1	1	7	3	8	5	0	Game 12	<table><tr><td>7</td><td>3</td></tr><tr><td>3</td><td>9</td></tr><tr><td>3</td><td>10</td></tr><tr><td>5</td><td>0</td></tr></table>	7	3	3	9	3	10	5	0
7	1																		
1	7																		
3	8																		
5	0																		
7	3																		
3	9																		
3	10																		
5	0																		
U: up L: left				D: down R: right															

図 2 (Selten,Chumra,(2008)Figure5 より)

的に全種類のゲームでの距離の二乗の平均 Q を出す。これにより人間の学習過程としてある学習モデルを仮定した時に結果がどの程度現実と合致するかを測定することができる。

Chmura,Goerg,Selten(2011)の実験で用いられた学習モデルを自分で組み実験を再現したものが ipython notebook 上に記載してある。impulse-matching learning、action-sampling learning、reinforcement learning は Chmura,Goerg,Selten(2011)の論文内の結果とほぼ同じになったが、self-tuning EWA は自分で得られた結果の方が論文内の結果よりも良いものとなった。しかし、これは impulse-matching learning と action-sampling learning が他の学習モデルと Selten,Chumra,(2008)で提示されている均衡概念よりも優れた行動予測を行う、という結論を覆すものではない。また、Chmura,Goerg,Selten(2011)の 4.5.に示されている通り、action-sampling learning に impulse-matching learning で行われている利得変形を施しても結果は改善せず、reinforcement learning のみ結果が改善した。self-tuning EWA については、論文内では変形後のゲームでは結果は悪化しているが、今回得られた結果では若干改善していた。

【仮説・検証・考察】

ここで、impulse-matching learning 内の損失回避度のパラメータ（論文内では 2 で固定されている）は時間によって変化しているのではないか、という疑問を持ち、「実験開始時は損失回避的でなく、時間が経つにつれて損失回避度が上昇する」（以下パターン 1）パターンと、逆に「実験開始時は損失回避的であり、時間が経つにつれて損失回避度が下降する」（以下パターン 2）パターンを検証した。比較のために「実験開始時は損失回避的で、中盤になると損失回避度が下がり、終盤に再び損失回避的になる」（以下パターン 3）パターンも検証している。IBE や impulse-matching learning では、自身の最低限保証された利得 s_i より大きい利得 x は $x^* = x - \frac{x-s_i}{2}$ と変形されている。この時、 $\frac{x-s_i}{2}$ の分母の 2 が損失回避の程度を示している。これが 1 に近づくとき x^* は s_i に近づき、パラメータが正の無限大に近づくとき x^* は元に利得に近づく。つまりパラメータが小さくなるにつれて損失回避の度合いが高くなり、パラメータが大きくなると損失を考えず元々の利得のみを考えることになる。パターン 1 では損失回避の上限 α と下限 β を設定し、 $[\alpha, \beta]$ 間を 200 等分し、第 t 期に $200-t+1$ 番目を損失回避のパラメータとして設定している。パターン 2 では第 t 期に t 番目のものをパラメータにしている。このような設定でシミュレーションを行ったところ、 $(\alpha, \beta) = (1.5, 3)$ と $(1.5, 2)$ とした時はパターン 1 の方が良い結果となったが、 $(2, 3)$ の時のみパターン 2 の方がより良い結果となった。結果的にはどれもパラメータを 2 で固定した時の値よりも小さくなるものは無かった。パターン 1、2 とともに、 (α, β) 変化させた時の最小値はパラメータを固定した際に得られるものよりも小さくはなかったが、 (α, β) の幅が狭く、損失回避度がラウンド依存である、と言える結果ではなかった。また、このような利得変形を他の学習モデルについても適用し、 $(\alpha, \beta) = (1.5, 3), (1.5, 2), (2, 3)$ とし、シミュレーションを行ったところ、action-sampling learning では $(2, 3)$ の時のみパターン 1 の方が良い結果となり、reinforcement learning では $(1.5, 2)$ の時のみパターン 1 の方が良い結果を出し、self-tuning EWA はパターン 1 の方が良い結果となったが、 $(2, 3)$ の時はほぼ同じ結果となった。

今後は Impulse Balance Equilibrium and Feedback in First Price Auctions (Reinhard Selten, 2005) などを参考に、このような定常概念の現実での応用例について考えていきたい。

参考文献

Stationary Concepts for Experimental 2x2-Games (Reinhard Selten, Thorsten Chmura, 2008)

A Correction and Re-experiment of stationary concepts for experimental 2x2 games (Christoph Brunner, Colin F. Camerer and Jacob K. Goeree, 2011)

Stationary Concepts for Experimental 2x2 Games: Reply (Reinhard Selten, Thorsten Chmura, Sebastian J. Goerg, 2011)

Learning in experimental 2x2 games (Thorsten Chmura, Sebastian J. Goerg, Reinhard Selten, 2011)