

ナッシュ均衡の計算

尾山 大輔

2023 年 4 月 24 日

ナッシュ均衡の計算アルゴリズム

- ▶ Support enumeration
 - ▶ Vertex enumeration
 - ▶ Lemke-Howson
 - ▶ ...
-
- ▶ von Stengel, B. (2007). "Equilibrium Computation for Two-Player Games in Strategic and Extensive Form," Chapter 3, Algorithmic Game Theory.
 - ▶ (一般的にナッシュ均衡の計算は難しい問題 (2 人ゲームでも).
サイズの大きなゲームのナッシュ均衡を計算するのは現実的でない.)

混合戦略ナッシュ均衡 (2 人ゲームのケース)

純粋戦略 $s_1 \in S_1$, $s_2 \in S_2$, 混合戦略 $\sigma_1 \in \Delta(S_1)$, $\sigma_2 \in \Delta(S_2)$ に対して

$$u_1(s_1, \sigma_2) = \sum_{s_2 \in S_2} \sigma_2(s_2) u_1(s_1, s_2), \quad u_2(\sigma_1, s_2) = \sum_{s_1 \in S_1} \sigma_1(s_1) u_2(s_1, s_2)$$

と書くことにする.

1. (σ_1^*, σ_2^*) がナッシュ均衡であるとは, すべての $\sigma_1 \in \Delta(S_1)$, $\sigma_2 \in \Delta(S_2)$ に対して

$$\begin{aligned} \sum_{s_1 \in S_1} \sigma_1^*(s_1) u_1(s_1, \sigma_2^*) &\geq \sum_{s_1 \in S_1} \sigma_1(s_1) u_1(s_1, \sigma_2^*) \\ \sum_{s_2 \in S_2} \sigma_2^*(s_2) u_2(\sigma_1^*, s_2) &\geq \sum_{s_2 \in S_2} \sigma_2(s_2) u_2(\sigma_1^*, s_2) \end{aligned}$$

が成り立つことである.

2. (σ_1^*, σ_2^*) がナッシュ均衡であるための必要十分条件は,

$$u_1(s_1, \sigma_2^*) = \max_{s'_1 \in S_1} u_1(s'_1, \sigma_2^*) \quad (\text{すべての } s_1 \in \text{supp}(\sigma_1^*) \text{ に対して})$$

$$u_2(\sigma_1^*, s_2) = \max_{s'_2 \in S_2} u_2(\sigma_1^*, s'_2) \quad (\text{すべての } s_2 \in \text{supp}(\sigma_2^*) \text{ に対して})$$

が成り立つことである.

ただし,

▶ $\text{supp}(\sigma_1) = \{s_1 \in S_1 \mid \sigma_1(s_1) > 0\} \cdots \sigma_1$ のサポート (support)

▶ $\text{supp}(\sigma_2) = \{s_2 \in S_2 \mid \sigma_2(s_2) > 0\} \cdots \sigma_2$ のサポート (support)

3. (σ_1^*, σ_2^*) がナッシュ均衡であるための必要十分条件は,

$$\text{supp}(\sigma_1^*) \subset br_1(\sigma_2^*), \quad \text{supp}(\sigma_2^*) \subset br_2(\sigma_1^*)$$

が成り立つことである.

ただし,

▶ $br_1(\sigma_2) = \{s_1 \in S_1 \mid u_1(s_1, \sigma_2) \geq u_1(s'_1, \sigma_2) \ \forall s'_1 \in S_1\}$

▶ $br_2(\sigma_1) = \{s_2 \in S_2 \mid u_2(\sigma_1, s_2) \geq u_2(\sigma_1, s'_2) \ \forall s'_2 \in S_2\}$

Support Enumeration アルゴリズム (2 人ゲームのケース)

何の工夫もないアルゴリズム

- ▶ 入力

非退化な 2 人有限戦略ゲーム

- ▶ 出力

入力されたゲームのすべての (混合戦略) ナッシュ均衡

非退化な (nondegenerate) ゲーム

- ▶ 2人ゲームが非退化であるとは、プレイヤー 1 と 2 のどんな混合戦略 $\sigma_1 \in \Delta(S_1)$, $\sigma_2 \in \Delta(S_2)$ に対しても

$$|br_2(\sigma_1)| \leq |\text{supp}(\sigma_1)|$$

$$|br_1(\sigma_2)| \leq |\text{supp}(\sigma_2)|$$

が成り立つことである.

($|X| \cdots$ 有限集合 X の要素の数)

- ▶ 非退化なゲームにおいては、どんなナッシュ均衡 (σ_1^*, σ_2^*) に対しても

$$|\text{supp}(\sigma_1^*)| = |\text{supp}(\sigma_2^*)|$$

が成り立つ.

$$\text{▶ } \because |\text{supp}(\sigma_1^*)| \leq |br_1(\sigma_2^*)| \leq |\text{supp}(\sigma_2^*)| \leq |br_2(\sigma_1^*)| \leq |\text{supp}(\sigma_1^*)|$$

Support Enumeration アルゴリズム (2 人ゲームのケース)

$S_1 = \{1, \dots, \ell\}$, $S_2 = \{1, \dots, m\}$ とする.

混合戦略をそれぞれ $x = (x_1, \dots, x_\ell)$, $y = (y_1, \dots, y_m)$ と書くことにする.

- ▶ 各 $k = 1, \dots, \min\{\ell, m\}$ と, $|I| = |J| = k$ なる純粋戦略の部分集合の組 $I \subset S_1$, $J \subset S_2$ に対して,

1. 2 組の連立線形方程式 (それぞれ $(k+1)$ 変数 $(k+1)$ 本)

$$\sum_{j \in J} u_1(i, j) y_j = u \text{ (すべての } i \in I \text{ に対して)}, \quad \sum_{j \in J} y_j = 1$$

$$\sum_{i \in I} u_2(i, j) x_i = v \text{ (すべての } j \in J \text{ に対して)}, \quad \sum_{i \in I} x_i = 1$$

を解く.

2. それぞれの解に対して,

$$y_j > 0 \text{ (すべての } j \in J \text{ に対して)}, \quad x_i > 0 \text{ (すべての } i \in I \text{ に対して)}$$

が成り立つかチェックする.

3. それぞれの解に対して,

$$u \geq \sum_{j \in J} u_1(i, j) y_j \text{ (すべての } i \notin I \text{ に対して)}$$

$$v \geq \sum_{i \in I} u_2(i, j) x_i \text{ (すべての } j \notin J \text{ に対して)}$$

が成り立つかチェックする.

例

	1	2
1	3, 3	3, 2
2	2, 2	5, 6
3	0, 3	6, 1

▶ サポートのサイズ $k = 1 \dots$ 純粋戦略

$(x, y) = ((1, 0, 0), (1, 0))$ がナッシュ均衡

例

	1	2
1	3, 3	3, 2
2	2, 2	5, 6
3	0, 3	6, 1

▶ サポートのサイズ $k = 2$

(i) $\{1, 2\} \subset S_1, \{1, 2\} \subset S_2$

1. $3y_1 + 3y_2 = u, 2y_1 + 5y_2 = u, y_1 + y_2 = 1$ を解くと,

$$y_1 = \frac{2}{3} > 0, y_2 = \frac{1}{3} > 0, u = 3 > (\text{戦略 3 の期待利得}) = 2$$

2. $3x_1 + 2x_2 = v, 2x_1 + 6x_2 = v, x_1 + x_2 = 1$ を解くと,

$$x_1 = \frac{4}{5} > 0, x_2 = \frac{1}{5} > 0, v = \frac{14}{5}$$

$\therefore (x, y) = ((\frac{4}{5}, \frac{1}{5}, 0), (\frac{2}{3}, \frac{1}{3}))$ はナッシュ均衡

例

	1	2
1	3, 3	3, 2
2	2, 2	5, 6
3	0, 3	6, 1

▶ サポートのサイズ $k = 2$

(ii) $\{2, 3\} \subset S_1$, $\{1, 2\} \subset S_2$

1. $2y_1 + 5y_2 = u$, $0y_1 + 6y_2 = u$, $y_1 + y_2 = 1$ を解くと,

$$y_1 = \frac{1}{3} > 0, \quad y_2 = \frac{2}{3} > 0, \quad u = 4 > (\text{戦略 1 の期待利得}) = 3$$

2. $2x_2 + 3x_3 = v$, $6x_2 + 1x_3 = v$, $x_2 + x_3 = 1$ を解くと,

$$x_2 = \frac{1}{3} > 0, \quad x_3 = \frac{2}{3} > 0, \quad v = \frac{8}{3}$$

$\therefore (x, y) = ((0, \frac{1}{3}, \frac{2}{3}), (\frac{1}{3}, \frac{2}{3}))$ はナッシュ均衡

例

	1	2
1	3, 3	3, 2
2	2, 2	5, 6
3	0, 3	6, 1

▶ サポートのサイズ $k = 2$

(iii) $\{1, 3\} \subset S_1, \{1, 2\} \subset S_2$

1. $3y_1 + 3y_2 = u, 0y_1 + 6y_2 = u, y_1 + y_2 = 1$ を解くと,

$$y_1 = \frac{1}{2} > 0, y_2 = \frac{1}{2} > 0, u = 3 < (\text{戦略 2 の期待利得}) = \frac{7}{2}$$

2. $3x_1 + 3x_3 = v, 2x_1 + 1x_3 = v, x_1 + x_3 = 1$ を解くと,

$$x_1 = 2 > 0, x_3 = -1 < 0, v = 3$$

∴ このケースのナッシュ均衡は存在しない

QuantEcon.py による数値計算

現時点で `quantecon.game_theory` に実装されているアルゴリズム

- ▶ 2 人ゲーム
 - ▶ `support_enumeration` (ナッシュ均衡をすべて求める; 10 数戦略程度)
 - ▶ `vertex_enumeration` (ナッシュ均衡をすべて求める; 10 数戦略程度)
 - ▶ `lemke_howson` (ナッシュ均衡を 1 つ求める; 数 100 戦略程度)
- ▶ 多人数ゲーム
 - ▶ `mclennan_tourky` (ナッシュ均衡を 1 つ求める; 少数プレイヤー少数戦略)
- ▶ Contribution 大歓迎!

GameTheory.jl (Julia) にも挑戦してみよう

▶ 2 人ゲーム

- ▶ `support_enumeration` (ナッシュ均衡をすべて求める; 10 数戦略程度)
- ▶ `(vertex_enumeration)`
- ▶ `lrs_nash` (ナッシュ均衡をすべて求める; 10 数戦略程度)

有理数による厳密計算ができる

▶ 多人数ゲーム

- ▶ `hc_solve` (ナッシュ均衡をすべて求める; 少数プレイヤー少数戦略)

▶ Contribution 大歓迎！

3人以上のケース

プレイヤー 1, 2, 3 の混合戦略をそれぞれ x, y, z とする.

$|I| = |J| = |H| = k$ なる純粋戦略の部分集合の組 $I \subset S_1, J \subset S_2, H \subset S_3$ に対して, 無差別条件は

$$\begin{aligned} \sum_{j \in J} \sum_{h \in H} u_1(i, j, k) y_j z_h &= u \text{ (すべての } i \in I \text{ に対して), } & \sum_{i \in I} x_i &= 1 \\ \sum_{h \in H} \sum_{i \in I} u_2(i, j, k) z_h x_i &= v \text{ (すべての } j \in J \text{ に対して), } & \sum_{j \in J} y_j &= 1 \\ \sum_{i \in I} \sum_{j \in J} u_3(i, j, k) x_i y_j &= w \text{ (すべての } h \in H \text{ に対して), } & \sum_{h \in H} z_h &= 1 \end{aligned}$$

... $(k+1) \times 3$ 本 $(k+1) \times 3$ 変数の 2 次多項式方程式

▶ 多項式方程式ソルバーが必要

▶ `GameTheory.jl` の `lrs_nash` は `HomotopyContinuation.jl` を利用