Fictitious play

三ツ國 拓真

2014/6/30

はじめに

- Fictitious Play **について** (Matching Pennies **を例として**)
- **シュミレーションプログラムのコードについての説明**
  - **工夫した点**
  - **今後の課題**

Ficititious play

- 戦略形ゲームが $t = 1, 2, ...$ の各期にプレイされる。各プレイヤーは $t + 1$ 期に、他プレイヤーが 1 期から t 期において選択した行動の比率に応じた確率で $t + 1$ 期の行動をとると予想し、自分の期待利得が最大になるような行動をとる。
- 予測に対する最適反応戦略をとるモデル。

Matching Pennies ゲームの例

- プレイヤーは０と１の２人行動は行動 ０と行動 １ の２通り
  利得は (1,-1),(-1,1) (-1,1),(1,-1)
- t 期にプレイヤー i が選択した行動は $a_i(t)$ と表す t 期にプレイヤー i が持っている「信念」は $x_i(t)$ と表す

  $x_0(t)$ は

$$x_0(t+1) = x_0(t) + \frac{1}{t+2}(a_1(t) - x_0(t))$$

  と再帰的に書くことができる.

Matching Pennies ゲームの例

- プレイヤー 0 の行動の決め方 (プレイヤー 1 についても同様．) 「プレイヤー 1 は，確率 $1 - x_0(t)$ で行動 0 をとり，確率 $x_0(t)$ で行動 1 をとる」 と考え，自分の期待利得が最大になるような行動をとる (行動 0,1 の期待利得が等しい場合は確率半々でランダムに選ぶ) 行動は信念に依存して決定される

- プレイヤー 0 の信念の決まり方 (プレイヤー 1 についても同様．) 初期信念 $x_0(0)$ は [0,1] 上の一様分布にしたがってランダムに与えられる.

- 各 $t \geq 1$ 期において、プレイヤー 1 が過去とった行動を $a_1(0), ..., a_1(t-1)$ とすると、信念 $x_0(t)$ は、

$$x_0(t) = \frac{x_0(0) + a_1(0) + ... + a_1(t-1)}{t+1}$$

で与えられる。

Matching Pennies ゲームの例

- これは、

$$x_0(t+1) = x_0(t) + \frac{1}{t+2}(a_1(t) - x_0(t))$$

と再帰的に書くことができる。

- このように信念は自らの初期信念と相手の過去の行動に依存して決まる。

コード

- **コード**

```
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
from random import uniform

p = [[1,-1], [-1, 1], [-1,1], [1, -1]]    #利得


def Fictplay(n):
    x0 = [uniform(0, 1)]
    x1 = [uniform(0, 1)]
#初期信念をランダムに設定,append を使い後に更新する
```

コード

- **コード**

```
for t in range(n):
        #期待利得
  ep0 = [p[0][0]*(1-x0[t])+p[1][0]*x0[t], p[2][0]*(
  ep1 = [p[0][1]*(1-x1[t])+p[2][1]*x1[t], p[1][1]*(

 #期待値の大きくなる行動を選択
  if ep0[0] > ep0[1]:
      a0 = 0
  else:
      a0 = 1
  if ep1[0] > ep1[1]:
      a1 = 0
  else:
      a1 = 1
```

コード

- **コード**

```
    #i+1期の信念を計算
         x0.append(x0[t]+(a1-x0[t])/(t+2))
         x1.append(x1[t] +(a0-x1[t])/(t+2))
     return x0, x1

x0, x1 = Fictplay(10000)   #t=10000までの信念の推移
plt.plot(x0, 'r-', label = 'x0(t)')
plt.plot(x1, 'b-', label = 'x1(t)')
plt.show()
x0_last = []
for i in range(100):    #Fictplay(10000)を100回繰
り返し最後の信念をヒストグラム化
    x0, x1 = Fictplay(10000)
    x0_last.append(x0[9999])
plt.hist(x0_last)
plt.show()
```

図

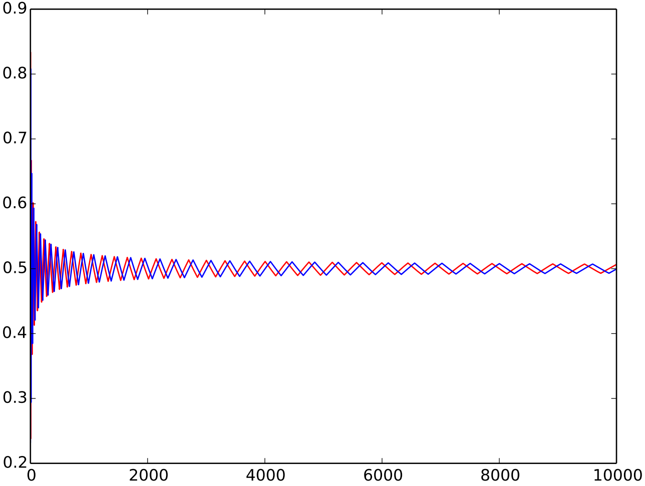

Figure : t=10000 のときの信念の推移

まとめ

- 回数を重ねるごとに 0.5 に収束しているようである。
- class で定義してみる。利得の行列表記をやってみる。コーディネーションゲームなど他のゲームのシミュレーションをやってみる。