# Obfuscation code[1]

**Anae Sobhani**

Postdoc Research Fellow

TLO

2018-06-08

[1] (based on Caspar Chorus "A Simple Model of Obfuscation-based Decision-making by Human and Artificial agents" paper)

# 1. Agent Decision Process (one rule):

- A= Action set= $\{a_1, a_2, \ldots, a_j\}$
  - J=J= # actions
- R= Rule set= $\{r_1, r_2, \ldots, r_k\}$
  - K=K= # rules
- $S_{kj}$= Score matrix $\in \{+, 0, -\}$
  - L= #/size of "0" in each row of the $S_{kj}$ matrix
- P $(a_j|r_k)$= probability of $S_{kj}$ matrix elements (i.e. action-rule probability)
- P $(r_k)$= prosecutor rule probability
- P $(r_k|a_j)$= prosecutor rule posterior probability
- $H_j$ = Agent action entropy
- $C_i$= Agent choice

**1) Set the random score matrix = $S_{kj}$**

- Assign (+, 0, – ) to each element randomly

**restrictions:**

- if there is one element in the row is (+), then the rest must be (-) in that row: strong rule

- if there is no (+), then that rule (that matrix row) is called weak rule

e.g.

| | $a_1$ | $a_2$ | $a_3$ | |
|---|---|---|---|---|
| $r_1$ | + | – | – | Strong rule |
| $r_2$ | 0 | 0 | – | Weak rule |
| $r_3$ | – | 0 | 0 | Weak rule |
| $r_4$ | – | + | – | Weak rule |

**2) Calculate P $(a_j|r_k)$ matrix (action and rule probability for each of the matrix elements):**

- If "+", then P $(a_j|r_k)$= 1.
- If "0", then P $(a_j|r_k)$=1/L.
  *(L= size/# of "0" in the row)*
- If "-", then P $(a_j|r_k)$~0.
  (since for latter calculations we would need to compute the log(P $(a_j|r_k)$), we have to set the P $(a_j|r_k)$ of the "-" elements as **close to 0**.

e.g.

| | $a_1$ | $a_2$ | $a_3$ | |
|---|---|---|---|---|
| $r_1$ | 1 | 0 | 0 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_1) = 1$ |
| $r_2$ | 1/2 | 1/2 | 0 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_2) = 1$ |
| $r_3$ | 0 | 1/2 | 1/2 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_3) = 1$ |
| $r_4$ | 0 | 0 | 0 | $\sum_{j=}^{J} \mathrm{P}\ (a_j \vert r_4) = 1$ |

**3) Calculate P ($r_k$) (rule probability):**

- P ($r_k$)= $\frac{1}{K}$
  *(K= number of rules)*

e.g.

P ($r_k$)= 1/4

**4) Calculate P ($r_k|a_j$) matrix (posterior probability) for each element:**

- P ($r_k|a_j$)= $\frac{\mathrm{P}\ (a_j|r_k)\ .\ \mathrm{P}\ (r_k)}{\sum_{k=1}^{K}\ (\mathrm{P}\ (a_j|r_k)\ .\ \mathrm{P}\ (r_k))}$

e.g.

| | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|
| $r_1$ | $\frac{1 * \frac{1}{4}}{\left(1*\frac{1}{4}\right)+\left(\frac{1}{2}*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)}$ | 0 | 0 |
| $r_2$ | 1/3 | 1/4 | 0 |
| $r_3$ | 0 | 1/4 | 1 |
| $r_4$ | 0 | 1/2 | 0 |
| | $\sum_{k=1}^{K} \mathrm{P}\ (a_1 \vert r_k) = 1$ | $\sum_{k=1}^{K} \mathrm{P}\ (a_2 \vert r_k) = 1$ | $\sum_{k=1}^{K} \mathrm{P}\ (a_3 \vert r_k) = 1$ |

**5) Calculate $H_j$ action matrix for the agent:**

- $H_j = -\sum_{k=1}^{k}[P(r_k|a_j) \,.\, \log(P(r_k|a_j))]$

e.g.

| $a_1$ | $a_2$ | $a_3$ |
|---|---|---|
| $-[(2/3 * \log(2/3)) + (1/3 * \log(1/3)) + (0 * \log(\sim 0)) + (0 * \log(\sim 0))]$ | 0.452 | 0 |

**6) Generate the choice ($C_i$) (or selected action) for the agent:**
C= $a_j$ with Max ($H_j$)

e.g.

Max ($H_1, H_2, H_3$) = 0.452, so $C_i$= agent choice= $a_2$

## 2. Naïve onlooker Decision Process (one rule):

**1) Set the random score matrix = $S_{kj}$**

- Assign (+, 0, – ) to each element randomly

**Note: Same restrictions as previous code (agent decision making code)**

e.g:

| | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|
| $r_1$ | 0 | 0 | – |
| $r_2$ | 0 | 0 | – |
| $r_3$ | – | 0 | 0 |

**2) Compute all possible binary choice sets based on actions ($a_j$): $C_n$**

e.g.

- $C_1$= $\{a_1, a_2\}$

| | $a_1$ | $a_2$ |
|---|---|---|
| $r_1$ | 0 | 0 |
| $r_2$ | 0 | 0 |
| $r_3$ | – | 0 |

- $C_2$= $\{a_2, a_3\}$

| | $a_1$ | $a_3$ |
|---|---|---|
| $r_1$ | 0 | – |
| $r_2$ | 0 | – |
| $r_3$ | – | 0 |

- $C_3$= $\{a_1, a_3\}$

| | $a_2$ | $a_3$ |
|---|---|---|
| $r_1$ | 0 | – |
| $r_2$ | 0 | – |
| $r_3$ | 0 | 0 |

**3) Calculate $H_j$ action matrix for the agent for each choice set ($C_n$): $H_{j|Cn}$ matrix**

**3.1: For each $C_n$, do steps 2 to 5 from the agent Decision Process (one rule) and compute the $H_j$**

**3.1.1.: Calculate P $(a_{j|Cn}|r_k)$ matrix**

Note: there is no restrictions in here and $\sum_{j=1}^{J} \mathrm{P}\ (a_j|r_j)$ in each row can be less than 1.

**3.1.2. : Calculate P $(r_k)$ (rule probability)**

**3.1.3.: Calculate P $(r_k|a_{j|Cn})$ matrix (posterior probability) for each element:**

Note: there is no restrictions in here and $\sum_{k=1}^{K} \mathrm{P}\ (a_j|r_k) = 1$ in each column can be less than 1.

**3.1.4.: Calculate $H_{j|Cn}$ action matrix for the agent**

**3.2: Repeat 3.1.1 to 3.1.4 for rest of the choice sets ($C_n$)**

**4) Compute expected entropy for the Naïve onlooker for each choice set ($C_n$): E($H_{j|Cn}$)**

**E($H_{j|Cn}$) =** $$\sum_{j=1}^{J} \left[ H_{j|C} \cdot \sum_{k=1}^{K} \frac{P(a_{j|C}|r_k)}{K} \right]$$

e.g.

- for $C_1$:

  E($H_{j|C1}$)= 0.3*((0.5+0.5)/3)+.45*((.5+.5+1)/3)=0.4

- for $C_2$:

  E($H_{j|C2}$)=0.38

- for $C_3$:

  E($H_{j|C3}$)=0.20

**5) Sort the E($H_{j|Cn}$) smallest to larges**

e.g.

1) E($H_{j|C3}$)= 0.20

2) $E(H_{j|C2})$=0.38
3) $E(H_{j|C1})$= 0.4

**6) Choose the selected naïve onlooker choice set: Naïve onlooker chosen choice set ($C_n$))**

**Naïve onlooker choice set ($C_n$))= $C_n$ |min ($E(H_{j|Cn})$)**

e.g.

Chosen set= min ($E(H_{j|C3})$, $E(H_{j|C2})$, $E(H_{j|C1})$) = $E(H_{j|C3})$= 0.20

****At the end print the outputs of these steps:**

1) **score matrix = $S_{kj}$**
2) **binary choice** sets based on actions ($a_j$): **$C_n$**
3) Only Final **$H_{j|Cn}$ matrices**
5) **Sort the naïve onlooker $E(H_{j|Cn})$ smallest to larges**
6) **Naïve onlooker chosen choice set ($C_n$))**

## 3. Cynical onlooker Decision Process (one rule):

**1) Set the random score matrix = Skj**

- Assign (+, 0, – ) to each element randomly

**<u>Note: Same restrictions as previous code (agent decision making code)</u>**

e.g:

| | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|
| $r_1$ | 0 | 0 | – |
| $r_2$ | 0 | 0 | – |
| $r_3$ | – | 0 | 0 |

**2) Compute all possible binary choice sets based on actions ($a_j$): $C_n$**

e.g.

- $C_1$= $\{a_1, a_2\}$

| | $a_1$ | $a_2$ |
|---|---|---|
| $r_1$ | 0 | 0 |
| $r_2$ | 0 | 0 |
| $r_3$ | – | 0 |

- $C_2$= $\{a_2, a_3\}$

| | $a_1$ | $a_3$ |
|---|---|---|
| $r_1$ | 0 | – |
| $r_2$ | 0 | – |
| $r_3$ | – | 0 |

- $C_3$= $\{a_1, a_3\}$

| | $a_2$ | $a_3$ |
|---|---|---|
| $r_1$ | 0 | – |
| $r_2$ | 0 | – |
| $r_3$ | 0 | 0 |

**3) Calculate $H_j$ action matrix for the agent for each choice set ($C_n$): $H_{j|Cn}$ matrix**

**3.1: For each $C_n$, do steps 2 to 5 from the agent Decision Process (one rule) and compute the $H_j$**

**3.1.1.: Calculate P $(a_{j|Cn}|r_k)$ matrix**

Note: there is no restrictions in here and $\sum_{j=1}^{J} \mathrm{P}\ (a_j|r_j)$ in each row can be less than 1.

**3.1.2. : Calculate P $(r_k)$ (rule probability)**

**3.1.3.: Calculate P $(r_k|a_{j|Cn})$ matrix (posterior probability) for each element:**

Note: there is no restrictions in here and $\sum_{k=1}^{K} \mathrm{P}\ (a_j|r_k) = 1$ in each column can be less than 1.

**3.1.4.: Calculate $H_{j|Cn}$ action matrix for the agent**

**3.2: Repeat 3.1.1 to 3.1.4 for rest of the choice sets ($C_n$)**

**4) Compute entropy for the Cynical onlooker for each choice set ($C_n$): $E(H_{j|Cn})$**

**$E(H_{j|Cn})$ =** $\max_{j=1..J}\{H_{j|C}\}$

e.g.

- for $C_1$:

  $E(H_{j|C1})= 0.45$

- for $C_2$:

  $E(H_{j|C2})=0.30$

- for $C_3$:

  $E(H_{j|C3})=0.46$

**5) Sort the $E(H_{j|Cn})$ smallest to larges**

e.g.

1) $E(H_{j|C2})= 0.30$

2) $E(H_{j|C1})$=0.45
3) $E(H_{j|C3})$= 0.46

**6) Choose the selected naïve onlooker choice set: Cynical onlooker chosen choice set ($C_n$))**

**Cynical onlooker choice set ($C_n$))= $C_n$ |min ($E(H_{j|Cn})$)**

e.g.

Chosen set= min ($E(H_{j|C3})$, $E(H_{j|C2})$, $E(H_{j|C1})$) = $E(H_{j|C2})$= 0.30

****At the end print the outputs of these steps:**

1) **score matrix = $S_{kj}$**
2) **binary choice** sets based on actions ($a_j$): **$C_n$**
3) Only Final **$H_{j|Cn}$ matrices**
5) **Sort the Cynical onlooker $E(H_{j|Cn})$ smallest to larges**
6) **Cynical onlooker chosen choice set ($C_n$))**

# 4. Agent Hybrid (one rule):

**1) Set the random score matrix = $S_{kj}$**

- Assign (+, 0, – ) to each element randomly

**restrictions:**

- if there is one element in the row is (+), then the rest must be (-) in that row: strong rule

- if there is no (+), then that rule (that matrix row) is called weak rule

e.g.

| | $a_1$ | $a_2$ | $a_3$ | |
|---|---|---|---|---|
| $r_1$ | + | – | – | Strong rule |
| $r_2$ | 0 | 0 | – | Weak rule |
| $r_3$ | – | 0 | 0 | Weak rule |
| $r_4$ | – | + | – | Weak rule |

**2) Calculate P $(a_j|r_k)$ matrix (action and rule probability for each of the matrix elements):**

- If "+", then P $(a_j|r_k)$= 1.
- If "0", then P $(a_j|r_k)$=1/L.
  *(L= size/# of "0" in the row)*
- If "-", then P $(a_j|r_k)$~0.
  (since for latter calculations we would need to compute the log(P $(a_j|r_k)$), we have to set the P $(a_j|r_k)$ of the "-" elements as **close to 0**.

e.g.

| | $a_1$ | $a_2$ | $a_3$ | |
|---|---|---|---|---|
| $r_1$ | 1 | 0 | 0 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_1) = 1$ |
| $r_2$ | 1/2 | 1/2 | 0 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_2) = 1$ |
| $r_3$ | 0 | 1/2 | 1/2 | $\sum_{j=1}^{J} \mathrm{P}\ (a_j \vert r_3) = 1$ |
| $r_4$ | 0 | 0 | 0 | $\sum_{j=}^{J} \mathrm{P}\ (a_j \vert r_4) = 1$ |

**3) Calculate P $(r_k)$ (rule probability):**

- $\mathrm{P}\ (r_k) = \frac{1}{K}$

  *(K= number of rules)*

e.g.

P $(r_k)$= 1/4

**4) Calculate P $(r_k|a_j)$ matrix (posterior probability) for each element:**

- $\mathrm{P}\ (r_k|a_j) = \frac{\mathrm{P}\ (a_j|r_k)\ .\ \mathrm{P}\ (r_k)}{\sum_{k=1}^{K}\ (\mathrm{P}\ (a_j|r_k)\ .\ \mathrm{P}\ (r_k))}$

e.g.

| | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|
| $r_1$ | $\frac{1 * \frac{1}{4}}{\left(1 * \frac{1}{4}\right) + \left(\frac{1}{2} * \frac{1}{4}\right) + \left(0 * \frac{1}{4}\right) + \left(0 * \frac{1}{4}\right)}$ | 0 | 0 |
| $r_2$ | 1/3 | 1/4 | 0 |
| $r_3$ | 0 | 1/4 | 1 |
| $r_4$ | 0 | 1/2 | 0 |
| | $\sum_{k=1}^{K} \mathrm{P}\ (a_1 \vert r_k) = 1$ | $\sum_{k=1}^{K} \mathrm{P}\ (a_2 \vert r_k) = 1$ | $\sum_{k=1}^{K} \mathrm{P}\ (a_3 \vert r_k) = 1$ |

5) **Calculate $H_j$ action matrix for the agent:**

- $H_j = -\sum_{k=1}^{k}[\mathrm{P}\,(r_k|a_j)\,.\,\log(\mathrm{P}\,(r_k|a_j))\,]$

e.g.

| $a_1$ | $a_2$ | $a_3$ |
|---|---|---|
| $-[(2/3 * \log(2/3)) + (1/3 * \log(1/3)) + (0 * \log(\sim 0)) + (0 * \log(\sim 0))]$ | 0.452 | 0 |

6) **Choose one rule (row) = Chosen K**

e.g
chosen rule (row)=3

7) **Set gamma between 0 and 1**
- $0 < \gamma < 1$

e.g.
$\gamma = 0.3$

1) **Set $S'_{kj}$ matrix**
- If "+ or 0", then $S'_{kj}=1$.
- If "-", then $S'_{kj}=0$.

e.g
for the chosen rule 3:

| | | | |
|---|---|---|---|
| $r_3$ | 0 | 1 | 1 |

8) **Calculate the action utility for the chosen row= U'j**

- $$U_j = (1-\gamma)\cdot \mathbf{1}(s_{kj} \neq -) + \gamma \cdot \frac{H_j - H_{\min}}{H_{\max} - H_{\min}}$$

e.g.

for rule 3:

- U'1= (1-0.3)*0+0.3*[(0.276-0)/(0.452-0)]
- U'2= (1-0.3)*1+0.3*[(0.452-0)/(0.452-0)]
  - U'3= (1-0.3)*1+0.3*[(0-0)/(0.452-0)]

**9) Generate the choice ($C_j$)) (or elected action) for the agent:**

C= $a_j$ with max ($U'_j$)

e.g. C= Choice = action 2

## 5. Agent Decision Process (Multi rule):

**1) Set the random score matrix = $S_{kj}$**

- Assign (+, 0, – ) to each element randomly

e.g.

| | a1 | a2 | a3 |
|---|---|---|---|
| r1 | 0 | - | 0 |
| r2 | - | 0 | 0 |

2) **Set $S'_{kj}$ matrix**

- If "+ or 0", then $S'_{kj}$=1.
- If "-", then $S'_{kj}$=0.

e.g

| | a1 | a2 | a3 |
|---|---|---|---|
| r1 | 1 | 0 | 1 |
| r2 | 0 | 1 | 1 |

**3) Set $St_{kn}$ matrix and n: (k * n matrix size)**

- Since $B_k$ (for each rule) can only be 0 or 1, generate all potential combination of 0 and 1 for the rules

e.g

| | St1 | St2 | St3 | St4 |
|---|---|---|---|---|
| r1 | 1 | 1 | 0 | 0 |
| r2 | 1 | 0 | 1 | 0 |

n= potential states of $B_k$ combinations= 4

**4) Onlooker prior probabilistic belief about the weights attached by the agent to different rule= f(B)**

- F(B)= 1/n
  - (n= potential states of $B_k$ combinations)

e.g

F(B)= ¼=0.25

5) **Generate action utility action= $U_{jk}$ and $U_j$ matrix for each $St_{kn}$ matrix**

- For each $St_{kn}$ matrix:
  - **$U_{jk}$ (in row k)= $B_k$ * $S'_{kj}$**
  - $\boldsymbol{u_j = \mu * \sum_{k=1}^{K} u_{jk}}$

- **_______μ= 1 or manually defined**

- Do that for all $St_{kn}$ matrix

e.g.

| **St1** | **a1** | **a2** | **a3** |
|---|---|---|---|
| **r1** | (1*1) | (1*0) | (1*1) |
| **r2** | (1*0) | (1*1) | (1*1) |
| **Uj** | (1+0)=1 | (0+1)=1 | 2 |

| **St2** | **a1** | **a2** | **a3** |
|---|---|---|---|
| **r1** | (1*1) | (1*0) | (1*1) |
| **r2** | (0*0) | (0*1) | (0*1) |
| **Uj** | (1+0)=1 | (0+0)=0 | 1 |

| **St3** | **a1** | **a2** | **a3** |
|---|---|---|---|
| **r1** | (0*1) | (0*0) | (0*1) |
| **r2** | (1*0) | (1*1) | (1*1) |
| **Uj** | (0+0)=0 | (0+1)=1 | 1 |

| **St4** | **a1** | **a2** | **a3** |
|---|---|---|---|
| **r1** | (0*1) | (0*0) | (0*1) |
| **r2** | (0*0) | (0*1) | (0*1) |
| **Uj** | (0+0)=0 | (0+0)=0 | 0 |

**6) Generate the probability that a particular action is chosen given a vector of rule-weights= P $(\boldsymbol{a_j}|\boldsymbol{B})$ for each $St_{kn}$ matrix**

- $P(a_j|\boldsymbol{\beta}) = \frac{\exp(\sum_{k=1}^{K} u_{jk})}{\sum_{m=1}^{J} \exp(\sum_{k=1}^{K} u_{mk})}$

Eg.

| | **a1** | **a2** | **a3** |
|---|---|---|---|
| **St1** | [Exp (1)]/[exp(1)+ exp(1)+exp(2)] =0.21 | 0.21 | 0.58 |

$\sum_{j=1}^{J} \mathrm{P}\,(a_j|B) = 1$

| | | | | |
|---|---|---|---|---|
| **St2** | [Exp (1)]/[exp(1)+ exp(0)+exp(1)] =0.42 | 0.16 | 0.42 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j \vert B) = 1$ |
| **St3** | 0.16 | 0.42 | 0.42 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j \vert B) = 1$ |
| **St4** | 0.33 | 0.33 | 0.33 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j \vert B) = 1$ |

7) **Onlooker posterior probabilities matrix= f(B|$a_j$) matrix**

-

$$f(\boldsymbol{\beta}|a_j) = \frac{P(a_j|\boldsymbol{\beta}) \cdot f(\boldsymbol{\beta})}{\int_{\boldsymbol{B}} [P(a_j|\boldsymbol{\beta}) \cdot f(\boldsymbol{\beta})] d\boldsymbol{\beta}}$$

Eg.

| | a1 | a2 | a3 |
|---|---|---|---|
| **St1** | (0.21*0.25)/(0.21*0.25+0.16*0.25+0.42*0.25+0.33*0.25)=0.19 | | |
| **St2** | (0.16*0.25)/(0.21*0.25+0.16*0.25+0.42*0.25+0.33*0.25)=0.38 | | |
| **St3** | 0.14 | | |
| **St4** | 0.30 | | |
| | $\sum = 1$ | $\sum = 1$ | $\sum = 1$ |

8) **Calculate $H_j$ action matrix for the agent**

-

$$H_j = -\int_{\boldsymbol{B}} \left[ f(\boldsymbol{\beta}|a_j) \cdot \log \left( f(\boldsymbol{\beta}|a_j) \right) \right] d\boldsymbol{\beta}$$

e.g.

| a1 | a2 | a3 |
|---|---|---|
| - [0.19*log(0.19)]/[0.19*log(0.19)+0.38*log(0.38)+0.14*log(0.14)+0.3*log(0.3)]= 1.32 | | |

9) **Generate the choice ($C_i$)) (or selected action) for the agent:**

- C= $a_j$ with Max ($H_j$)

# 6. Agent Hybrid (Multi rule):

**1) Set the random score matrix = $S_{kj}$**

- Assign (+, 0, – ) to each element randomly

e.g.

| | a1 | a2 | a3 |
|---|---|---|---|
| **r1** | 0 | - | 0 |
| **r2** | - | 0 | 0 |

2) **Set $S'_{kj}$ matrix**

- If "+ or 0", then $S'_{kj}=1$.
- If "-", then $S'_{kj}=0$.

e.g

| | a1 | a2 | a3 |
|---|---|---|---|
| **r1** | 1 | 0 | 1 |
| **r2** | 0 | 1 | 1 |

**3) Set $St_{kn}$ matrix and n: (k * n matrix)**

- Since $B_k$ (for each rule) can only be 0 or 1, generate all potential combination of 0 and 1 for the rules

e.g

| | St1 | St2 | St3 | St4 |
|---|---|---|---|---|
| **B1** | 1 | 1 | 0 | 0 |
| **B2** | 1 | 0 | 1 | 0 |

n= potential states of $B_k$ combinations= 4

**4) Onlooker prior probabilistic belief about the weights attached by the agent to different rule= f(B)**

- F(B)= 1/n
  - (n= potential states of $B_k$ combinations)

e.g

F(B)= ¼=0.25

5) **Generate action utility action= $U_{jk}$ and $U_j$ matrix for each $St_{kn}$ matrix**

- For each $St_{kn}$ matrix:
  - **$U_{jk}$ (in row k)= $B_k$ * $S'_{kj}$**
  - $\boldsymbol{u_j} = \mu * \sum_{k=1}^{K} \boldsymbol{u_{jk}}$
  - **μ= 1 or manually defined**
- Do that for all $St_{kn}$ matrix

e.g.

| St1 | a1 | a2 | a3 |
|---|---|---|---|
| r1 | (1*1) | (1*0) | (1*1) |
| r2 | (1*0) | (1*1) | (1*1) |
| Uj | (1+0)=1 | (0+1)=1 | 2 |

| St2 | a1 | a2 | a3 |
|---|---|---|---|
| r1 | (1*1) | (1*0) | (1*1) |
| r2 | (0*0) | (0*1) | (0*1) |
| Uj | (1+0)=1 | (0+0)=0 | 1 |

| St3 | a1 | a2 | a3 |
|---|---|---|---|
| r1 | (0*1) | (0*0) | (0*1) |
| r2 | (1*0) | (1*1) | (1*1) |
| Uj | (0+0)=0 | (0+1)=1 | 1 |

| St4 | a1 | a2 | a3 |
|---|---|---|---|
| r1 | (0*1) | (0*0) | (0*1) |
| r2 | (0*0) | (0*1) | (0*1) |
| Uj | (0+0)=0 | (0+0)=0 | 0 |

**6) Generate the probability that a particular action is chosen given a vector of rule-weights= P $(a_j|B)$ for each $St_{kn}$ matrix**

- $$P(a_j|\beta) = \frac{\exp(\sum_{k=1}^{K} u_{jk})}{\sum_{m=1}^{J} \exp(\sum_{k=1}^{K} u_{mk})}$$

- For each $St_{kn}$ matrix

Eg.

| a1 | a2 | a3 |
|---|---|---|

| | | | | |
|---|---|---|---|---|
| **St1** | [Exp (1)]/[exp(1)+ exp(1)+exp(2)] =0.21 | 0.21 | 0.58 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j\|B) = 1$ |
| **St2** | [Exp (1)]/[exp(1)+ exp(0)+exp(1)] =0.42 | 0.16 | 0.42 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j\|B) = 1$ |
| **St3** | 0.16 | 0.42 | 0.42 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j\|B) = 1$ |
| **St4** | 0.33 | 0.33 | 0.33 | $\sum_{j=1}^{J} \mathrm{P}\,(a_j\|B) = 1$ |

7) **Onlooker posterior probabilities matrix= f(B|a$_j$) matrix for each** For each St$_{kn}$ matrix

-

$$f(\beta|a_j) = \frac{P(a_j|\beta)\cdot f(\beta)}{\int_B \left[P(a_j|\beta)\cdot f(\beta)\right]d\beta}$$

- For each St$_{kn}$ matrix

Eg.

| | a1 | a2 | a3 |
|---|---|---|---|
| **St1** | (0.21*0.25)/(0.21*0.25+0.16*0.25+0.42*0.25+0.33*0.25)=0.19 | | |
| **St2** | (0.16*0.25)/(0.21*0.25+0.16*0.25+0.42*0.25+0.33*0.25)=0.38 | | |
| **St3** | 0.14 | | |
| **St4** | 0.30 | | |
| | $\sum = 1$ | $\sum = 1$ | $\sum = 1$ |

8) **Calculate H$_j$ action matrix for the agent**

-

$$H_j = -\int_B \left[f(\beta|a_j)\cdot\log\left(f(\beta|a_j)\right)\right]d\beta$$

e.g.

| a1 | a2 | a3 |
|---|---|---|
| - [0.19*log(0.19)]/[0.19*log(0.19)+0.38*log(0.38)+0.14*log(0.14)+0.3*log(0.3)]= 1.32 | | |

9) **Choose one of the $St_{kn}$ from step 3 randomly or manually**

e.g
chosen state (**$St_{kn}$**)=St3

10) **Set gamma between 0 and 1**

- 0< γ <1

e.g.
γ= 0.3

11) **Calculate the action utility for the chosen $St_{kn}$ = U'j for each**

-

$$U_j = (1-\gamma) \cdot \frac{u_j - u_{\min}}{u_{\max} - u_{\min}} + \gamma \cdot \frac{H_j - H_{\min}}{H_{\max} - H_{\min}}$$

12) **Generate the choice ($C_i$)) (or elected action) for the agent:**

C= $a_j$ with max ($U'_j$)