Estimating Learning 估計學習理論模型

Joseph Tao-yi Wang (王道一) Experimetrics Module 7, EE-BGT

Estimating Learning

Joseph Tao-yi Wang

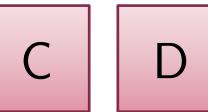
Outline: Estimating Learning (Experimetrics, Ch. 18)

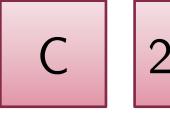
- 1. Directional Learning (DL): Selten and Stoecker (1986)
- 2. Reinforcement Learning (RL)
- 3. Belief Learning (BL)
- 4. EWA Learning: Camerer and Ho (ECMA 1999)
 - ► Experience-Weighted Attraction a Hybrid of RL and BL

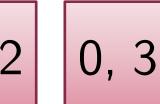
Directional Learning Theory

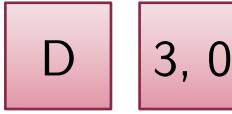
- Adjust behavior in response to previous outcome
 - Selten and Stoecker (1986)
 - Finitely Repeated Prisoner's Dilemma (PD)
 - SPE: Always Defect
- Stylized Facts

- Tacit Cooperation Until Close to End
- ▶ Want to Defect 1st (then Keep Defect)
- Decision: Which Round to Defect



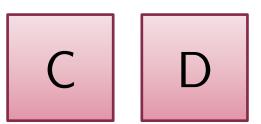


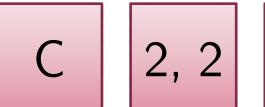




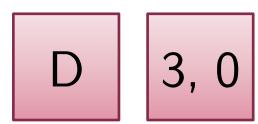
Directional Learning Theory

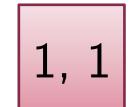
- \blacktriangleright Play N Supergames with a different opponent each time
 - Adjust next intended deviation period:
- If Deviated First:
 - May gain if deviated later
- ► If Deviated Later:
 - May gain if deviate early
- ▶ If Deviate in the Same Round:
 - May gain if deviate 1 period earlier











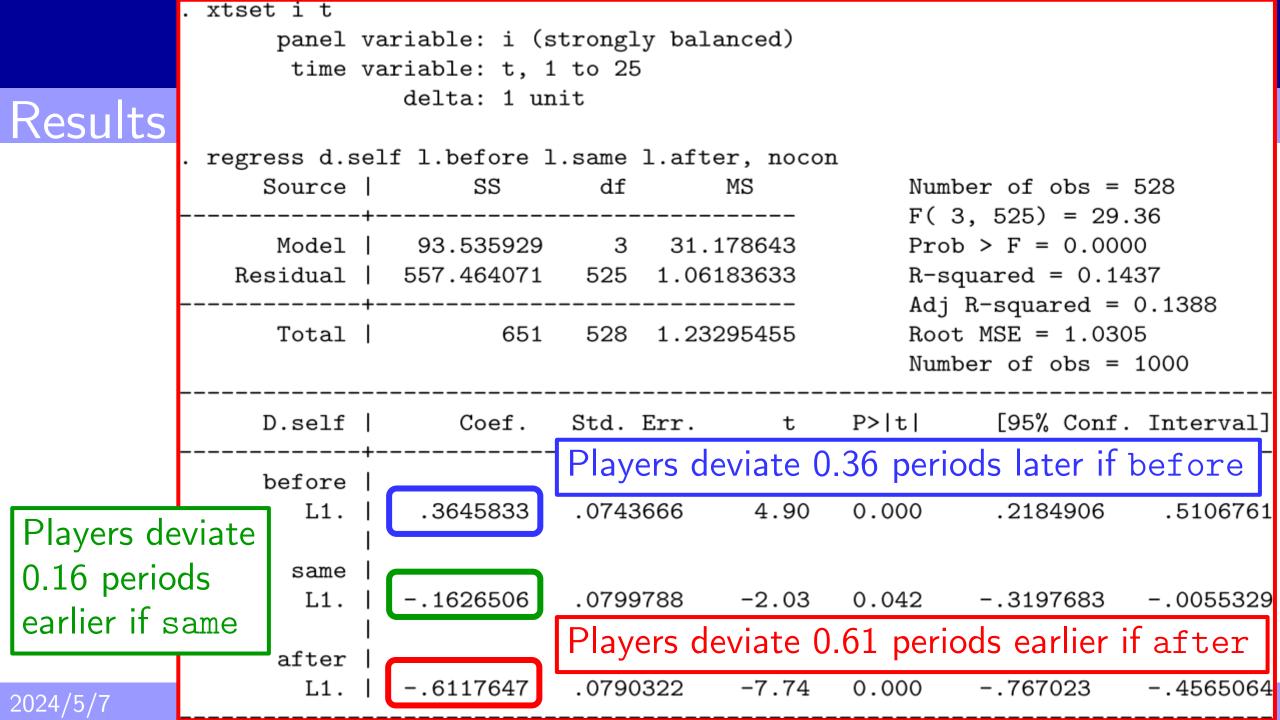
The Data: Table B1 of Selten and Stoecker (1986)

- ▶ *n*=35 subjects play 25 supergames (of 10-round PD)
 - ▶ Play the same opponent within 10 rounds of PD, but
 - Randomly rematch in between: selten-stoecker.dta
- Intended Deviation Period of each supergame: self
 - self/other = 1-10 (period)
 - self/other = 11 (later than opponent, but unobserved)
 - self/other = 12 (never deviate)
- Deviate before/same/after their opponent

Simple Linear Regression

- Predict difference in self with before/same/after
 - > d.self Difference in self
 - >l.before Lagged before
 - >l.same Lagged same
 - >l.after Lagged after
- **STATA** Command:
 - xtset i t

regress d.self l.before l.same l.after, nocon No constant term



Pursue-Evade Game (Rosenthal et al. 2003)

- Data: 100 pairs of 50 rounds pursue_evade_sim.dta
- Payoff Table

Player 1 (Pursuer): L (left) or R (right)
y1 = 0 if Pursuer choose L; y1 = 1 if Pursuer choose R
Player 2 (Evader): L (left) or R (right)
y2 = 0 if Evader choose L; y2 = 1 if Evader choose R

Pursue-Evade Game (Rosenthal et al. 2003)

- Find Two Players: i = 1, 2
- ▶ Rounds: t = 1, 2, ..., T = 50
- Five Actions: $s_i^0 = \mathbf{L}$, $s_i^1 = \mathbf{R}$
 - Relabel as
- Actions j = 0 (L) and j = 1 (R)
- Strategy of Players *i* in round *t* is $s_i(t)$
- Strategy of Players -i in round t is $s_{-i}(t)$
- Players *i*'s Payoff in round *t* is $\pi_i(s_i(t), s_{-i}(t))$

1, -1

0.0

R

R

0, 0

2. -2

Learning

Attraction to action j = 0, 1 after round t is $A_i^j(t)$

- Initial Attractions to action j = 0, 1 is $A_i^j(0)$
 - Normalize one of initial attractions to 0 for each player
- Choice Probability obtained by logistic transformation

$$P_{i}^{j}(t) = \frac{\exp\left[\lambda A_{i}^{j}(t-1)\right]}{\exp\left[\lambda A_{1}^{j}(t-1)\right] + \exp\left[\lambda A_{0}^{j}(t-1)\right]} \bullet \text{Irrelevant } (\lambda = 0)$$

$$\bullet i = 1, 2; j = 0, 1; t = 1, 2, ..., T; \lambda = \text{Sensitivity to attractions}$$

JOSEDH

Reinforcement Learning (RL)

- Erev and Roth (1998)
- Update attractions in response to previous payoffs
 - Choices "reinforced" only by previous payoffs

$$\underline{A_i^j(t)} = \phi \underline{A_i^j(t-1)} + I(s_i(t) = s_i^j) \pi_i(s_i^j, s_{-i}(t))$$

▶
$$i = 1, 2; j = 0, 1; t = 1, 2,...,T$$

Recency parameter:

 $\blacktriangleright \phi = 0$: Only most recent payoff is remembered

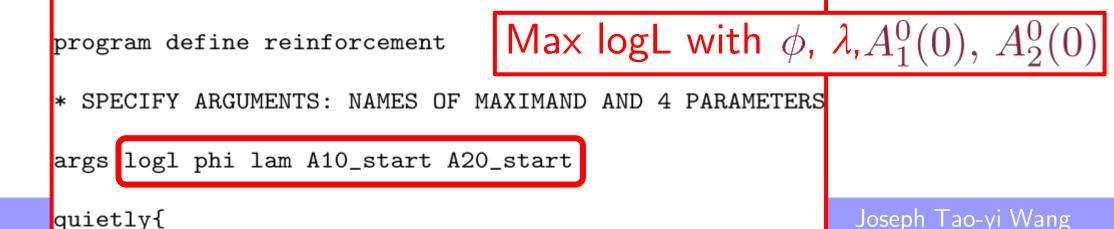
 $\blacktriangleright \phi = 1$: All past payoffs have equal weight

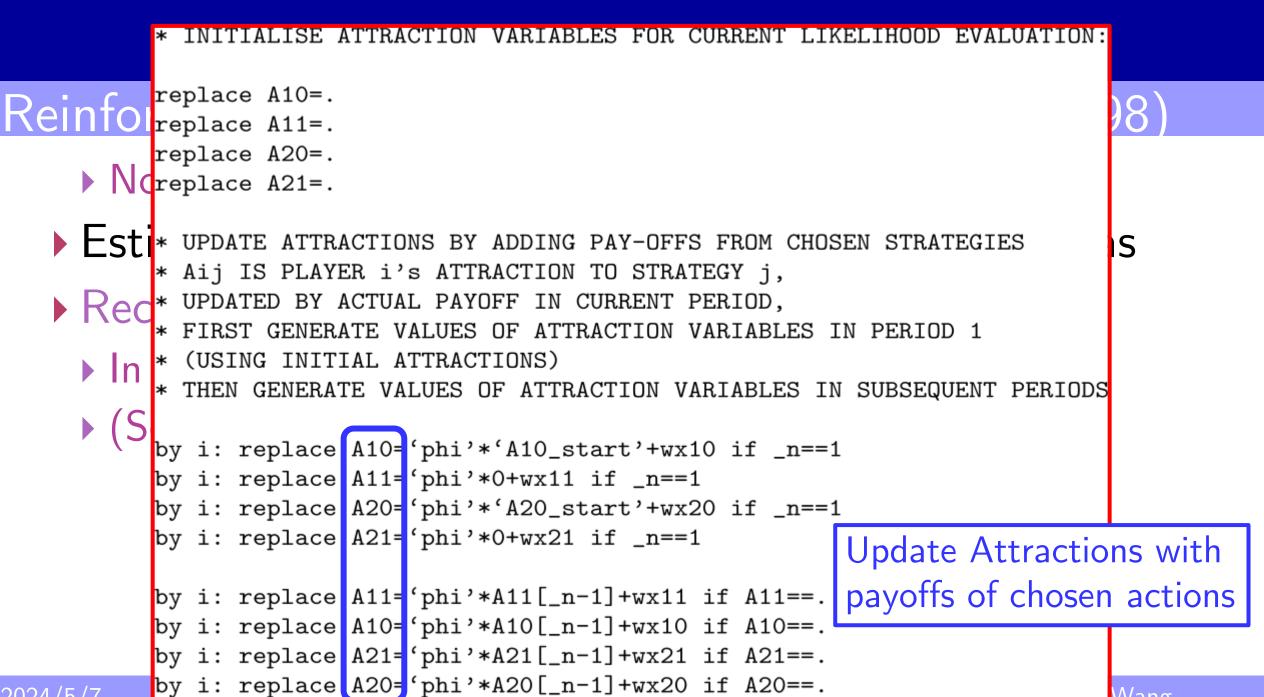
Reinforcement Learning (RL)

- ▶ Normalize Initial Attractions $A_1^1(0) = 0, A_2^1(0) = 0$
- Estimate Initial Attractions $A_1^0(0), A_2^0(0)$, as well as
- Recency parameter ϕ and Sensitivity parameter λ
 - In STATA using Maximum Likelihood
 - (See code in package)

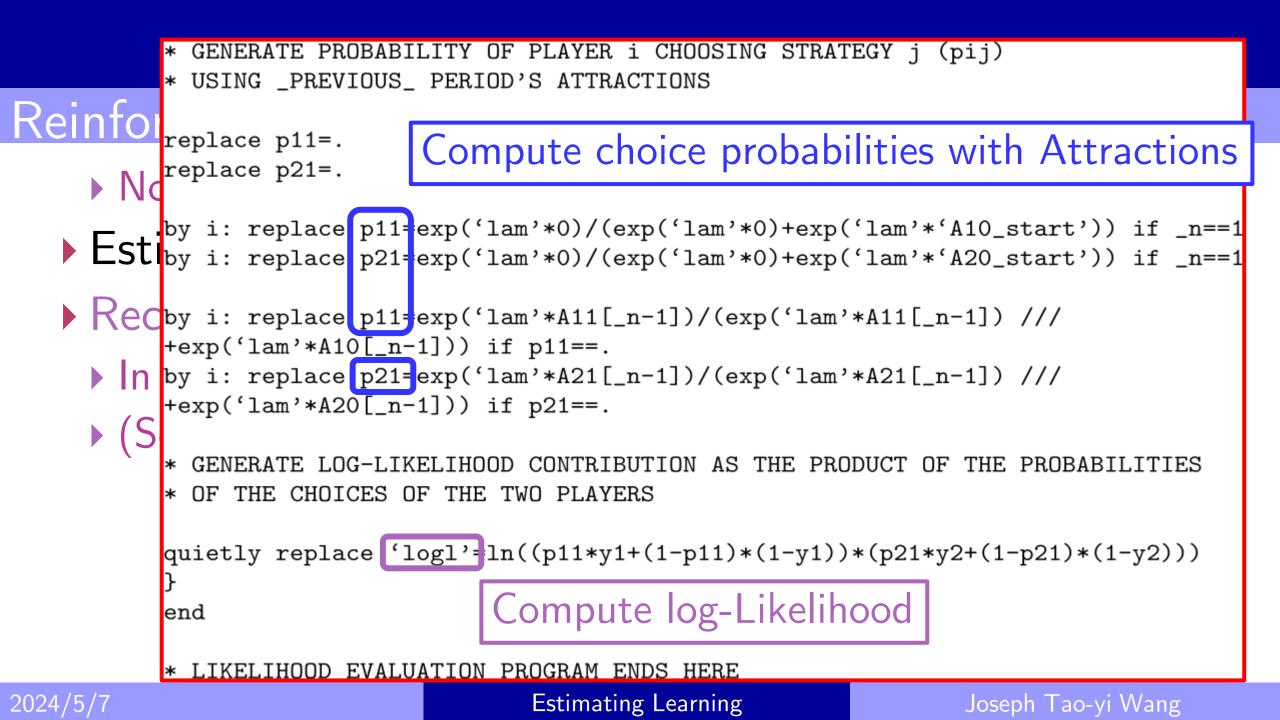
2024/5/7

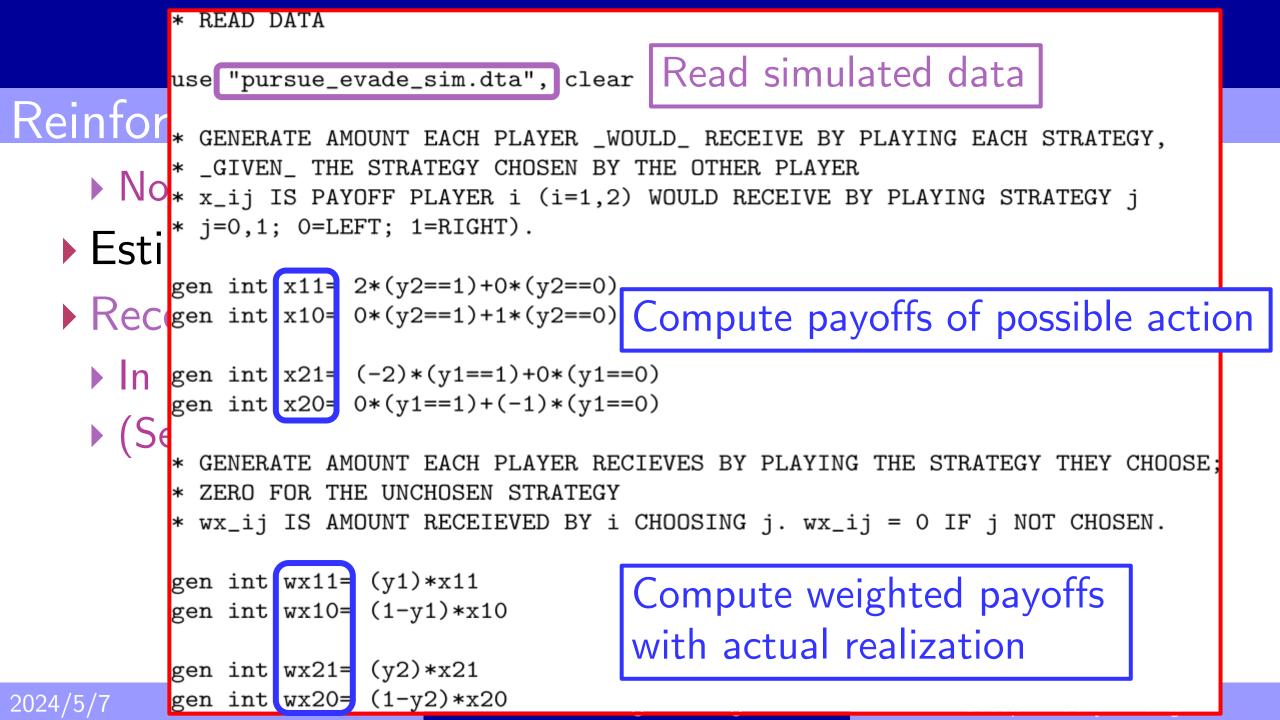
LIKELIHOOD EVALUATION PROGRAM STARTS HERE

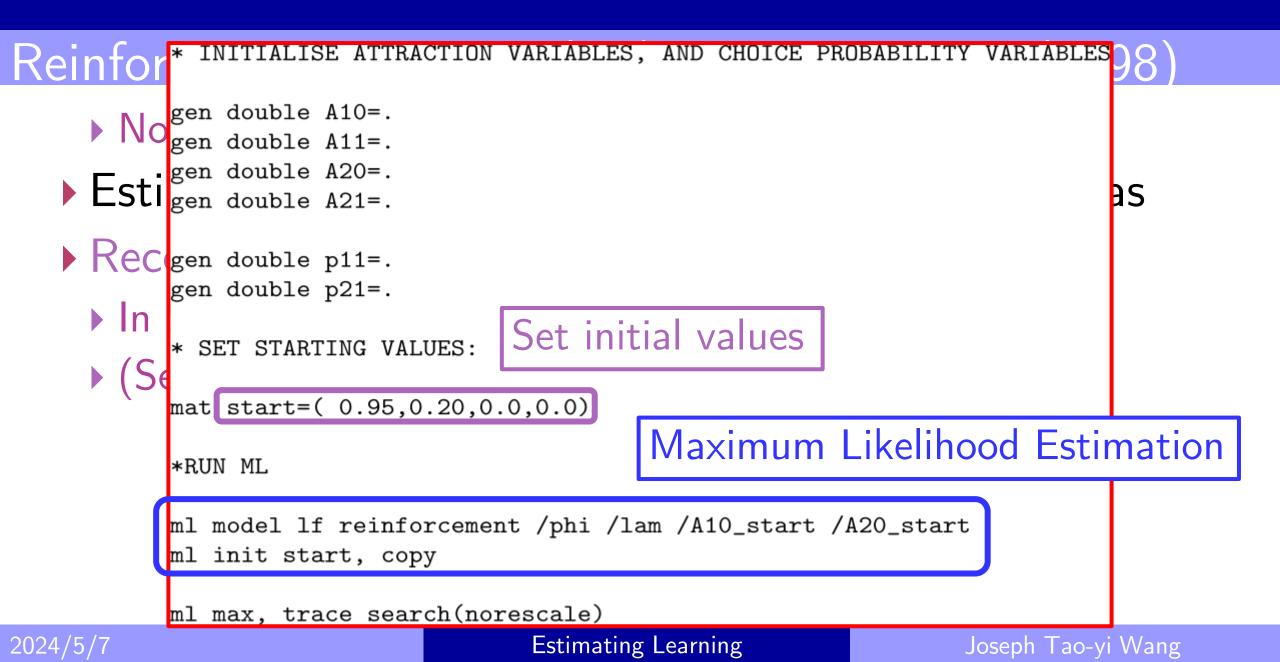


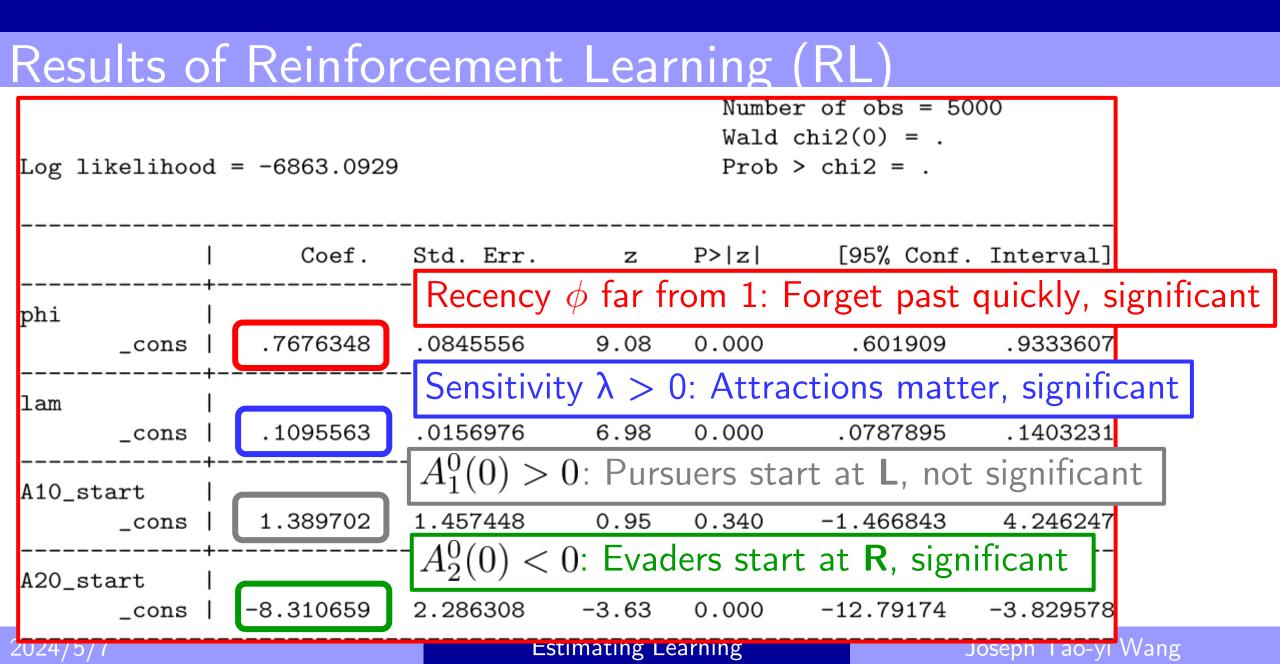


√ang









Simple Belief Learning (BL): Cournot Learning

Cournot Learning: Attractions increase by actioncorresponding payoffs given opponent actions

BR to opponent action in previous round

$$A_{i}^{j}(t) = A_{i}^{j}(t-1) + \pi_{i}(s_{i}^{j}, s_{-i}(t))$$

▶ i = 1, 2; j = 0, 1; t = 1, 2,...,T

- ▶ Normalize Initial Attractions $A_1^1(0) = 0, A_2^1(0) = 0$
- ▶ Only need to estimate Initial Attractions A⁰₁(0), A⁰₂(0) and λ using Maximum Likelihood (Too simple?!)

Belief Learning (BL): Standard Fictitious Play

- Standard Fictitious Play: Attractions is actioncorresponding average payoffs
 - Counting cards and BR to opponent actions from all rounds
- All Initial Attractions are zero: $A_i^j(0) = 0$, j = 0, 1

$$A_i^j(1) = \pi_i \left(s_i^j, s_{-i}(1) \right), A_i^j(2) = \frac{1}{2} \left[\pi_i \left(s_i^j, s_{-i}(1) \right) + \pi_i \left(s_i^j, s_{-i}(2) \right) \right]$$

$$A_i^j(3) = \frac{1}{3} \left[\pi_i \left(s_i^j, s_{-i}(1) \right) + \pi_i \left(s_i^j, s_{-i}(2) \right) + \pi_i \left(s_i^j, s_{-i}(3) \right) \right]$$

• ...,
$$A_i^j(t) = \frac{1}{t} \sum_{\tau=1}^t \pi_i(s_i^j, s_{-i}(\tau))$$

Belief Learning (BL): Experience Weight

- Express Attractions based on Experience N(t)
 - \blacktriangleright Observation Equivalents: Experience accumulated up to t
- Initial Experience is zero: N(0) = 0

- Iteratively define N(t) = N(t-1) + 1, t = 1,...,T
- All Initial Attractions are zero: $A_i^j(0) = 0$, j = 0, 1
- Iteratively define (for j = 0, 1; t = 1, ..., T)
 - $A_i^j(t) = \frac{1}{N(t)} \left[N(t-1)A_i^j(t-1) + \pi_i \left(s_i^j, s_{-i}(t) \right) \right]$
 - Special Case of N(t) = t is Standard Fictitious Play!

Belief Learning (BL): Weighted Fictitious Play

- Another Special Case is Weighted Fictitious Play
 - \blacktriangleright With Recency parameter ϕ
- Initial Experience is zero: N(0) = 0
- Iteratively define $N(t) = \phi N(t-1) + 1, t = 1, \cdots, T$
- All Initial Attractions are zero: $A_i^j(0) = 0$, j = 0, 1
- Iteratively define (for j = 0, 1; t = 1,...,T)

$$A_i^j(t) = \frac{1}{N(t)} \begin{bmatrix} \phi N(t-1)A_i^j(t-1) + \pi_i(s_i^j, s_{-i}(t)) \end{bmatrix}$$

Weights are 1, ϕ , ϕ^2 , ϕ^3 , etc.

Belief Learning (BL): Weighted Fictitious Play

- Weighted Fictitious Play: Attractions is actioncorresponding average payoffs weighted by recency (exponentially discounted)
- All Initial Attractions are zero: $A_i^j(0) = 0$, j = 0, 1 $A_i^j(1) = \pi_i(s_i^j, s_{-i}(1)),$ $A_i^j(2) = \frac{1}{\phi+1} \left[\phi \pi_i(s_i^j, s_{-i}(1)) + \pi_i(s_i^j, s_{-i}(2)) \right]$ $A_i^j(3) = \frac{\phi^2 \pi_i(s_i^j, s_{-i}(1)) + \phi \pi_i(s_i^j, s_{-i}(2)) + \pi_i(s_i^j, s_{-i}(3))}{\phi^2 + \phi + 1}$, etc.

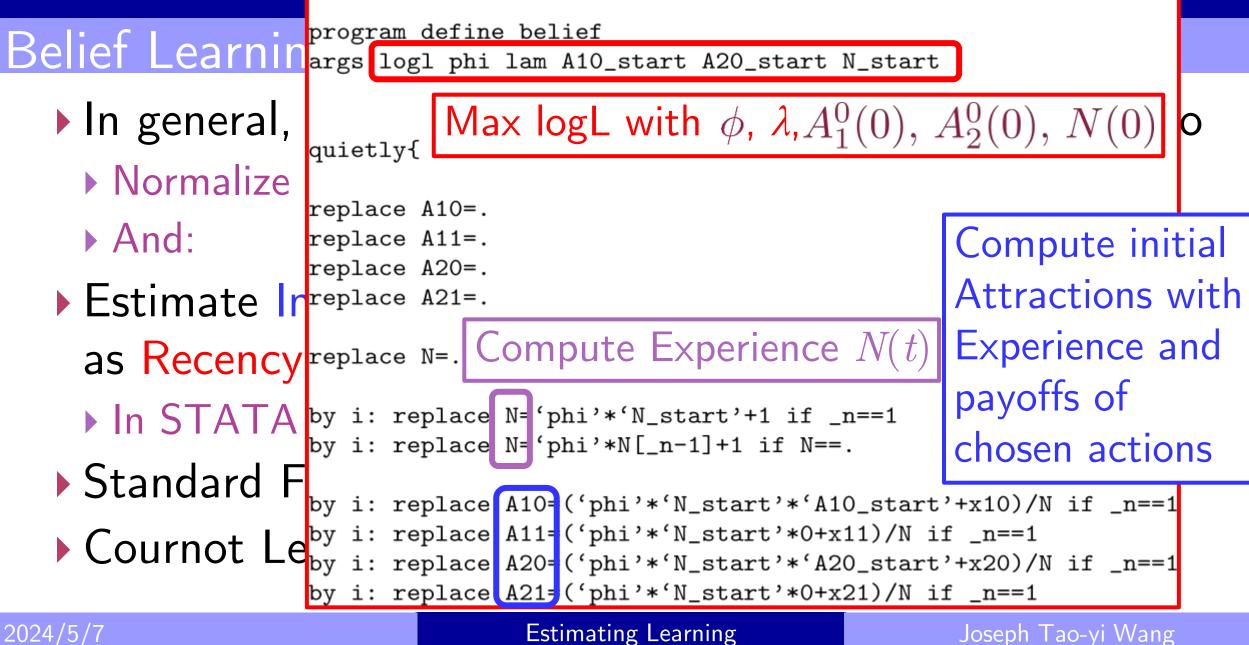
Belief Learning (BL): Weighted Fictitious Play

- In general, initial attractions and N(0) need not be zero
 - ▶ Normalize Initial Attractions $A_1^1(0) = 0, A_2^1(0) = 0$

And:

- Estimate Initial Attractions A⁰₁(0), A⁰₂(0), N(0) as well as Recency parameter φ and Sensitivity parameter λ
 In STATA using Maximum Likelihood (See code in package)
 Standard Fictitious Play if φ = 1
- Cournot Learning if $\phi = 0$

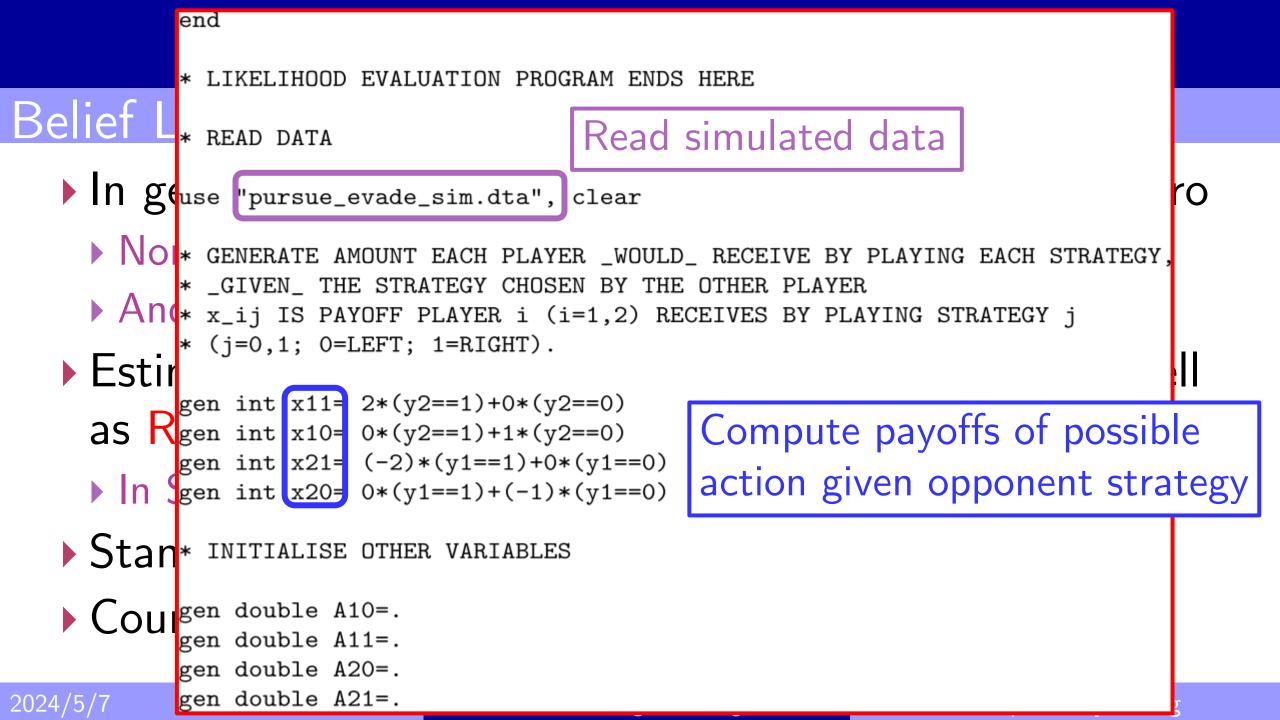
EVALUATION PROGRAM STARTS HERE

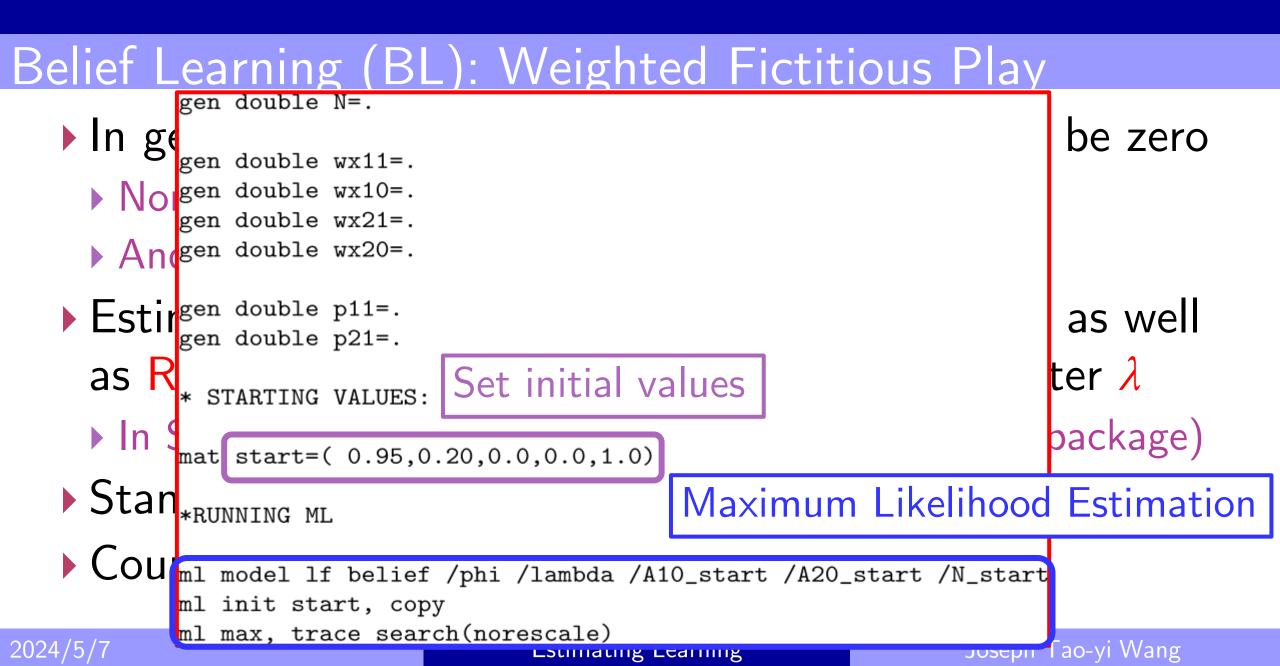


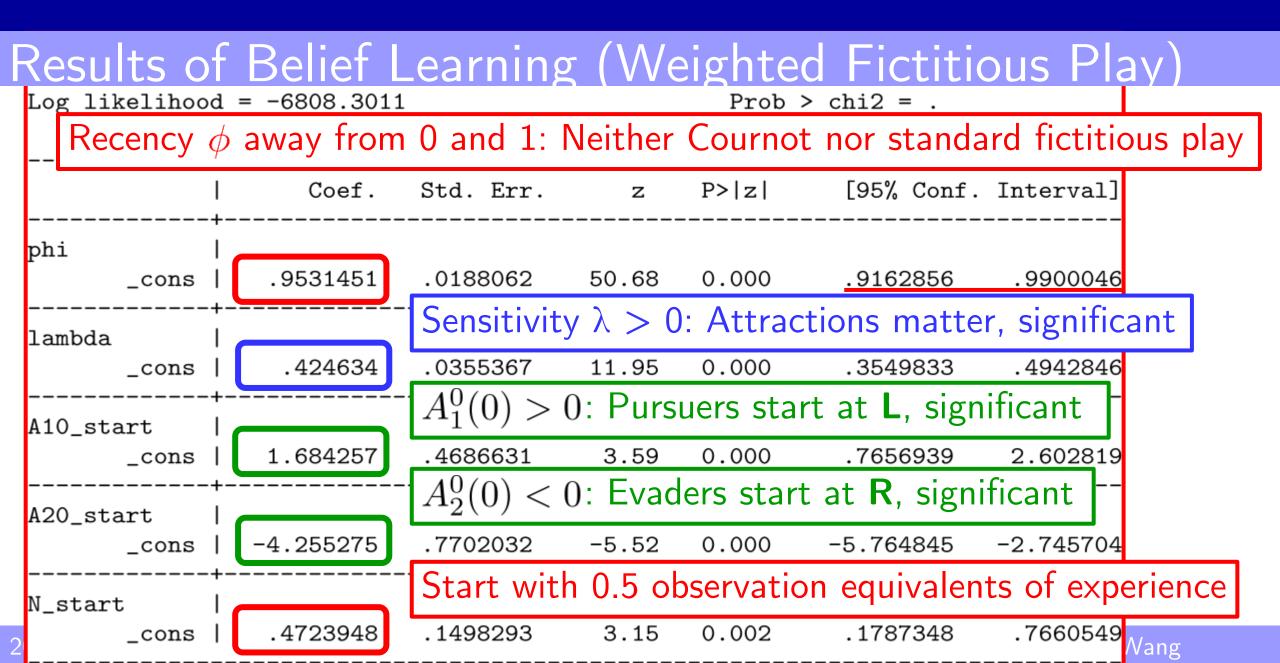
Estimating Learning

Joseph Tao-yi Wang

```
* Aij is the attraction, updated by payoff (either actual or hypothetical) in t,
     * to be used to determine choice probs in t+1
                                                              Update Attractions
    by i: replace A11=('phi'*N[_n-1]*A11[_n-1]+x11)/N if A11==.
Be by i: replace A10= ('phi'*N[_n-1]*A10[_n-1]+x10)/N if A10==.
                                                              with Experience
    by i: replace A21=('phi'*N[_n-1]*A21[_n-1]+x21)/N if A21==.
                                                              and payoffs of
   by i: replace A20=('phi'*N[_n-1]*A20[_n-1]+x20)/N if A20==.
                                                              chosen actions
    *pij are the probabilities player i choosing strategy j
    replace p11=.
                    Compute choice probabilities from Attractions
    replace p21=.
    by i: replace p11=exp('lam'*0)/(exp('lam'*0)+exp('lam'*'A10_start')) if _n==1
    by i: replace p21=exp('lam'*0)/(exp('lam'*0)+exp('lam'*'A20_start')) if _n==1
    by i: replace p11=exp('lam'*A11[_n-1])/(exp('lam'*A11[_n-1]) ///
    +exp('lam'*A10[_n-1])) if p11==.
    by i: replace p21=exp('lam'*A21[_n-1])/(exp('lam'*A21[_n-1]) ///
    +exp('lam'*A20[_n-1])) if p21==.
    replace 'logl'=ln((p11*y1+(1-p11)*(1-y1))*(p21*y2+(1-p21)*(1-y2)))
                    Compute log-Likelihood
2024/
```







Experience-Weighted Attraction (EWA) Learning Model

- EWA = Experience-Weighted + Attraction
- Has both Experience N(t) and Attractions $A_i^j(t)$

Experience N(t) accumulated as Observation Equivalents
Initial Experience estimated: N(0)
Iteratively define N(t) = ρN(t-1) + 1, t = 1, ..., T

▶ Past Experience Depreciation Rate is $\rho < 1$

Experience-Weighted Attraction (EWA) Learning Model Initial Attractions estimated: $A_i^j(0)$, j = 0, 1Attractions to different actions iteratively define $A_{i}^{j}(t) = \frac{\phi N(t-1)A_{i}^{j}(t-1) + \left[\delta + (1-\delta)I_{s_{i}(t)=s_{i}^{j}}\right]\pi_{i}(s_{i}^{j}, s_{-i}(t))}{s_{i}(t) + \left[\delta + (1-\delta)I_{s_{i}(t)=s_{i}^{j}}\right]\pi_{i}(s_{i}^{j}, s_{-i}(t))}$ N(t) $= \frac{\phi N(t-1)A_i^j(t-1) + 1 \cdot \pi_i(s_i^j, s_{-i}(t))}{N(t)} \text{ if } s_i^j \text{ chosen}$ $= \frac{\phi N(t-1)A_i^j(t-1) + \delta \cdot \pi_i(s_i^j, s_{-i}(t))}{N(t)} \text{ if not}$ $\mathsf{RL} \ (\delta = 0) \text{ vs. BL } (\delta = 1) \quad N(t) \quad (\text{for } j = 0, 1; t = 1, ..., T)$

Estimating Learning

2024/5/7

Joseph Tao-yi Wang

Experience-Weighted Attraction (EWA) Learning Model

Choice Probability obtained by logistic transformation

$$P_i^j(t) = \frac{\exp\left[\lambda A_i^j(t-1)\right]}{\exp\left[\lambda A_1^j(t-1)\right] + \exp\left[\lambda A_0^j(t-1)\right]}$$

▶
$$i = 1, 2; j = 0, 1; t = 1, 2,...,T$$

 $\lambda =$ Sensitivity to attractions

- Firelevant ($\lambda = 0$)
- Important (λ large)

Experience-Weighted Attraction (EWA) Learning Model

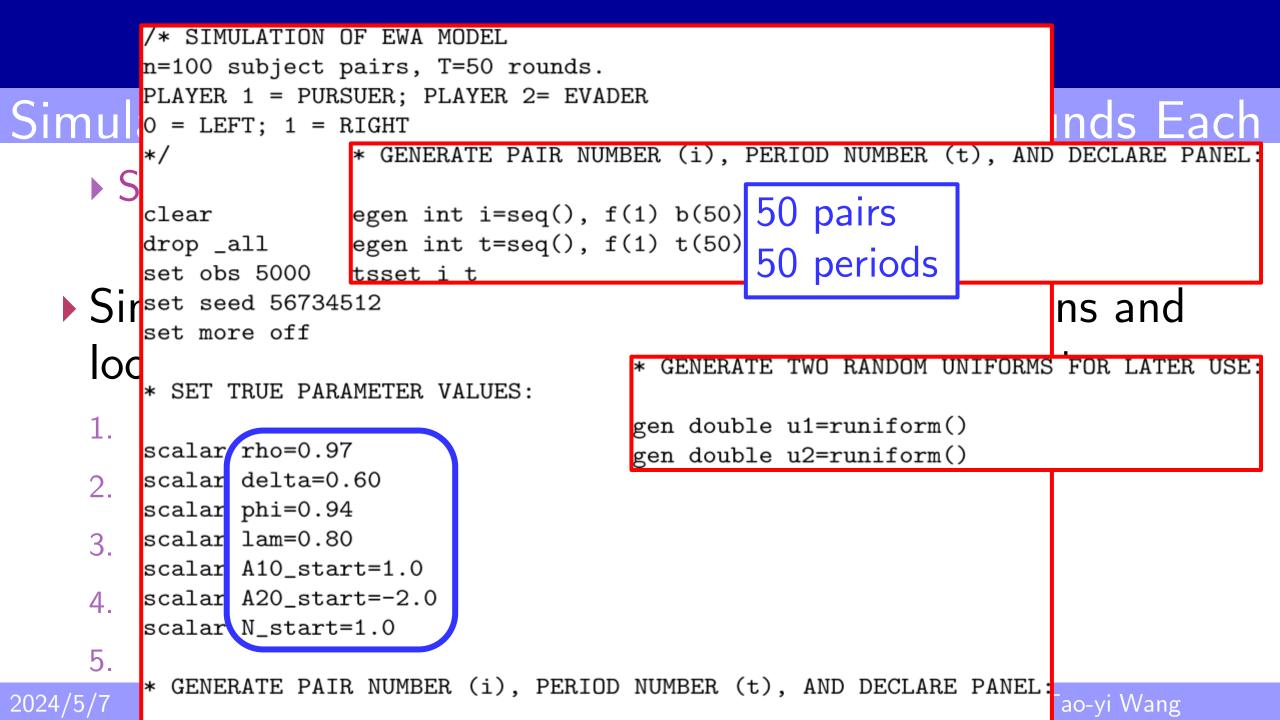
- Experience N(t)-weighted Attractions $A_i^j(t)$ model generates choice probabilities $P_i^j(t)$
- Estimate 7 parameters: $\rho, \delta, \phi, \lambda, A_1^1(0), A_2^1(0), N(0)$
- δ distinguishes RL: ($\delta = 0$) from BL ($\delta = 1$):
- 1. BL: $\delta = 1; \rho = \phi$
- 2. RL: $\delta = 0; N_0 = 1; \rho = 0$
 - Note that $A_1^1(0), A_2^1(0)$ not identified if $N_0 = 0$
 - \blacktriangleright Also, RL does not have depreciation ρ

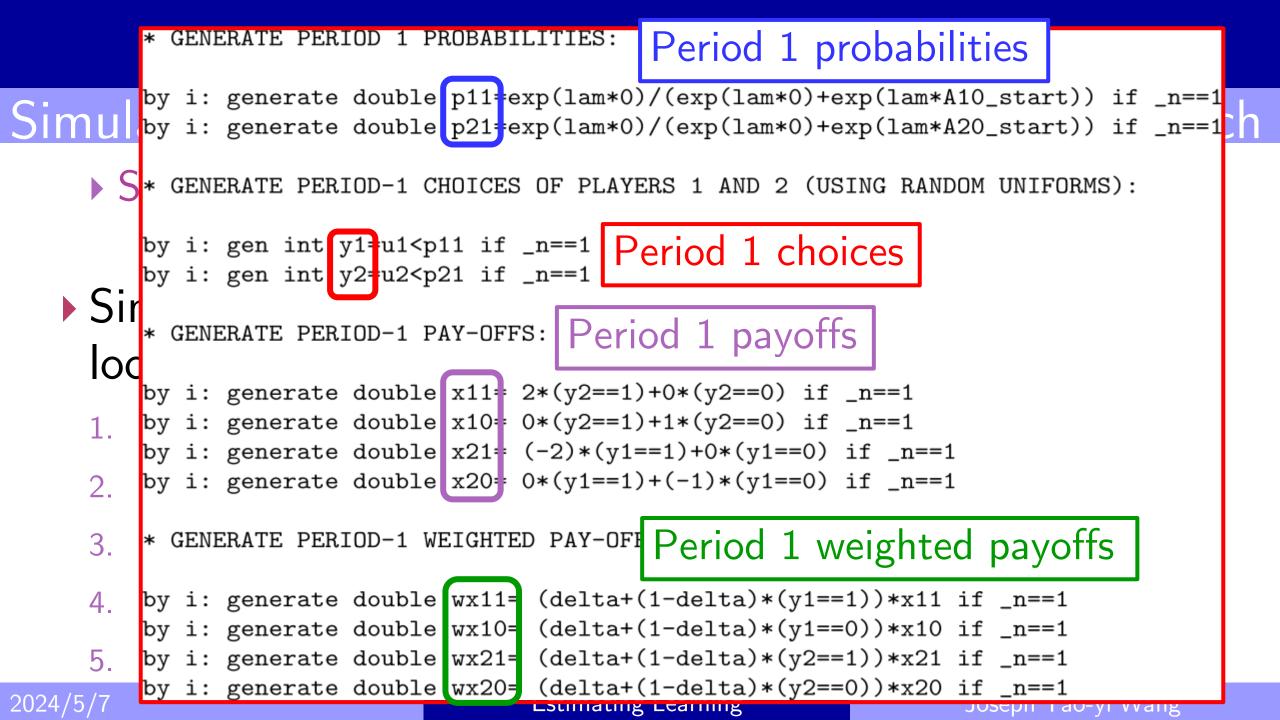
Simulating EWA for 100 Subject Pairs, 50 Rounds Each

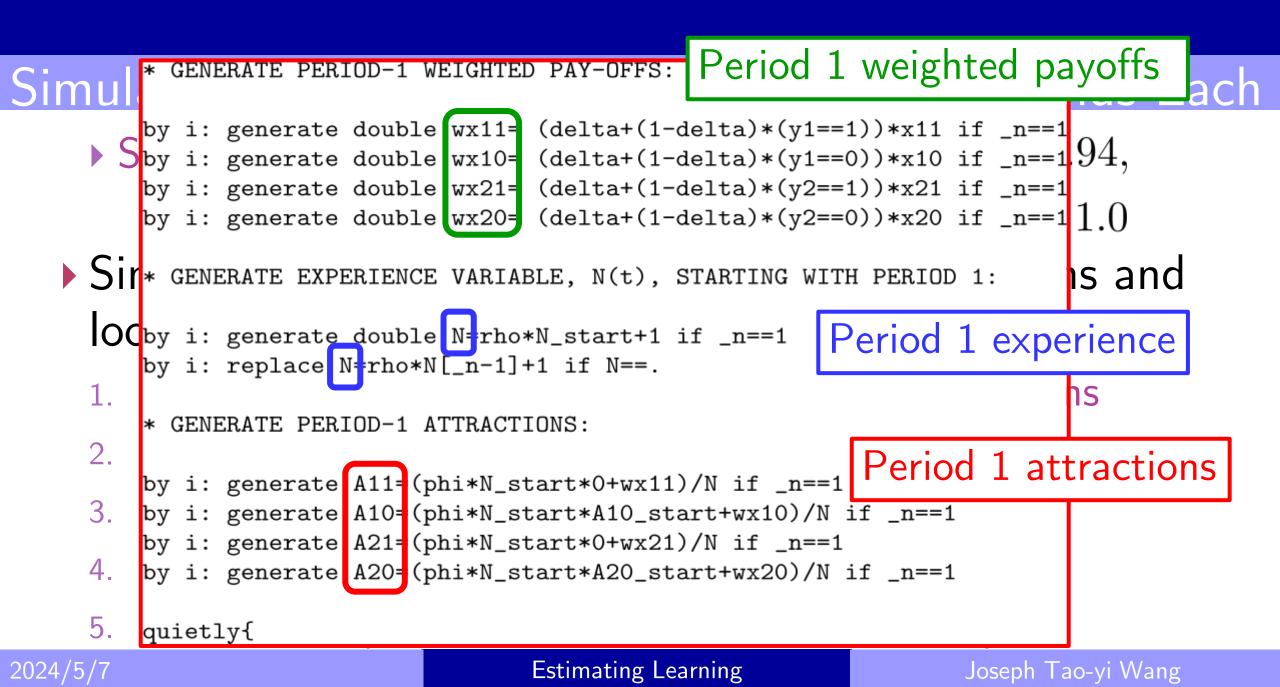
Simulate EWA model with: $\rho = 0.97, \delta = 0.60, \phi = 0.94, \phi = 0.94$

 $\lambda = 0.80, A_1^0(0) = 1.0, A_2^0(0) = -2.0, N(0) = 1.0$

- Simulate round 1 choices and resulting attractions and loop over round 2-50 with forvalues to compute:
 - 1. Choice probability p11, p21 from previous attractions
 - 2. Actual choices from probabilities
 - 3. Payoffs for each possible action
 - 4. Payoffs weighted by actual realizations
 - 5. Attractions (for next round's choice probability)

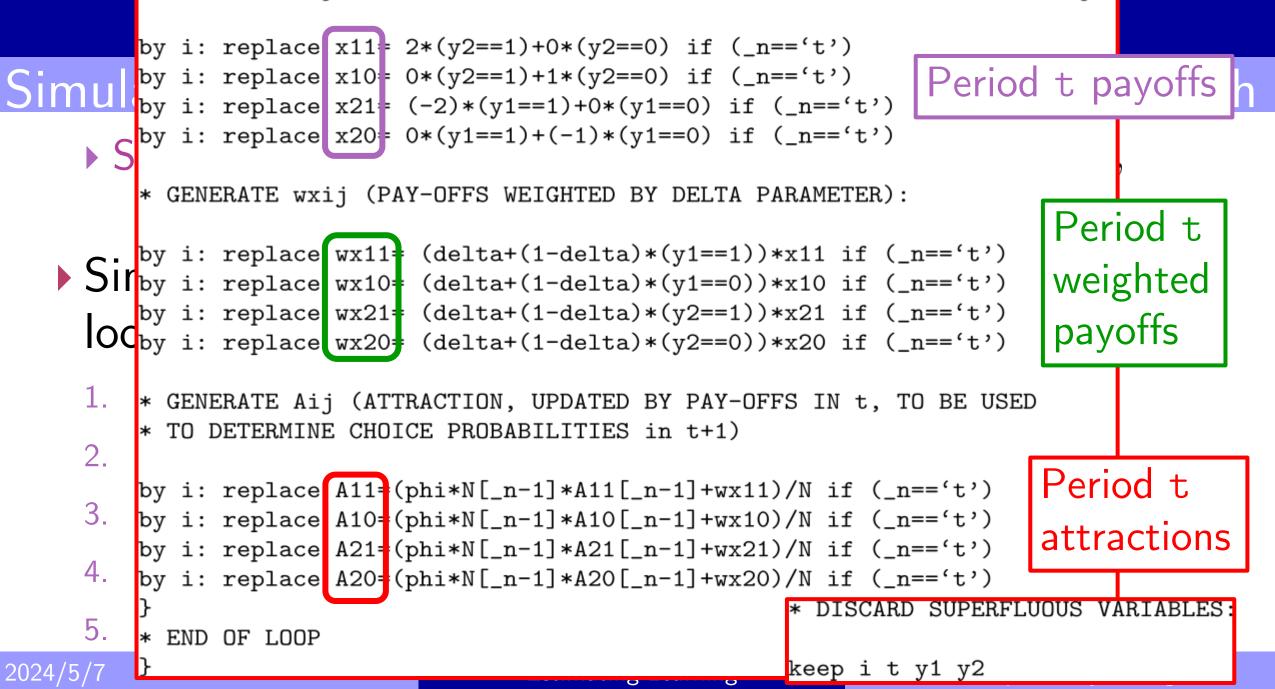






quietly{ * LOOP OVER PERIODS STARTS HERE |Loop over t = 2-50forvalues t = 2(1)50 { GENERATE p11 AND p21 (PROBABILITIES OF PLAYERS 1 and 2 CHOOSING STRATEGY 1): Sir by i: replace p11=exp(lam*A11[_n-1])/(exp(lam*A11[_n-1])+exp(lam*A10[_n-1])) /// OC if $(_n=-'t')$ by i: replace p21=exp(lam*A21[_n-1])/(exp(lam*A21[_n-1])+exp(lam*A20[_n-1])) /// if (_n=='t') 1. Period t probabilities 2. * GENERATE y1 AND y2 (CHOICES OF PLAYERS 1 AND 2) USING RANDOM UNIFORMS: 3. by i: replace y1=0 if (_n=='t') by i: replace y1= (u1<p11) if (_n=='t') 4. Period t choices by i: replace y2=0 if (_n=='t') 5. by i: replace y2= (u2<p21) if (_n=='t') 2024/5/7 Estimating Learning Joseph Tao-yi Wang

* GENERATE xij (PAY-OFF PLAYER i WOULD HAVE RECEIVED WITH STRATEGY j):

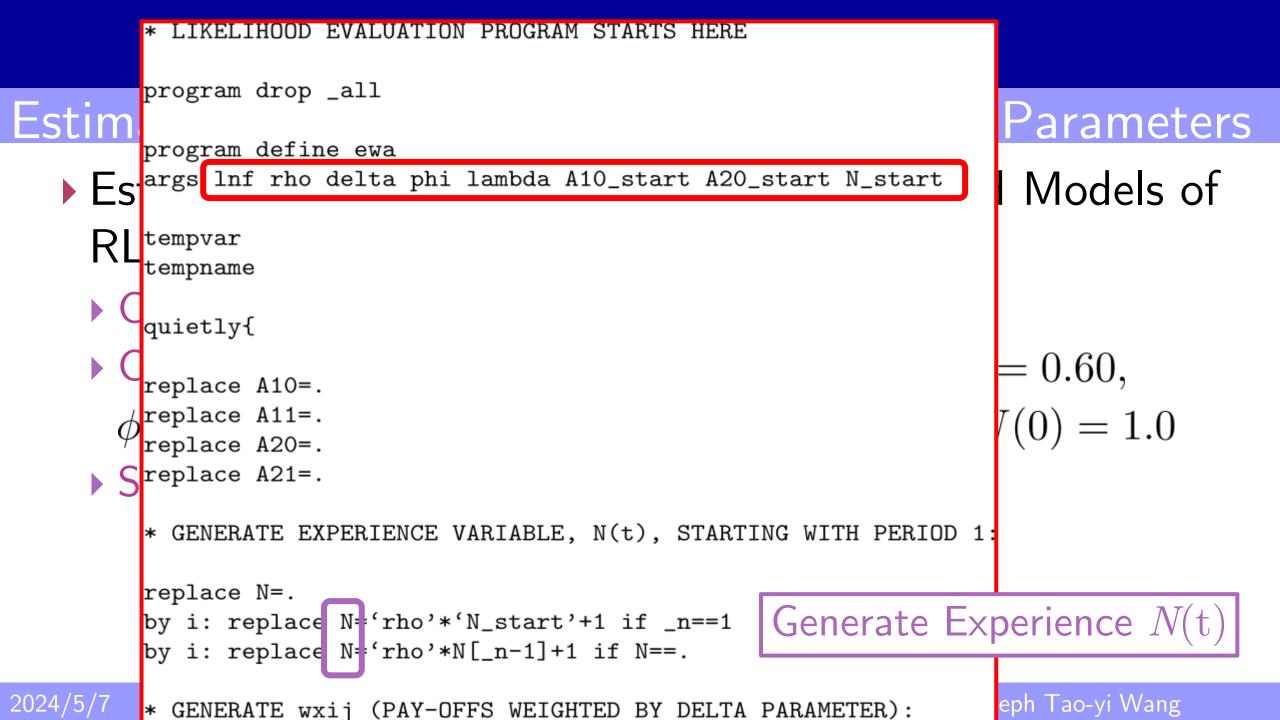


Estimating the Full EWA Model to Recover Parameters

- Estimate the Full EWA Model, the Restricted Models of RL and BL, and:
 - Conduct LR Tests to see if EWA performs better
 - Compare uncovered parameters with: $\rho = 0.97, \delta = 0.60,$

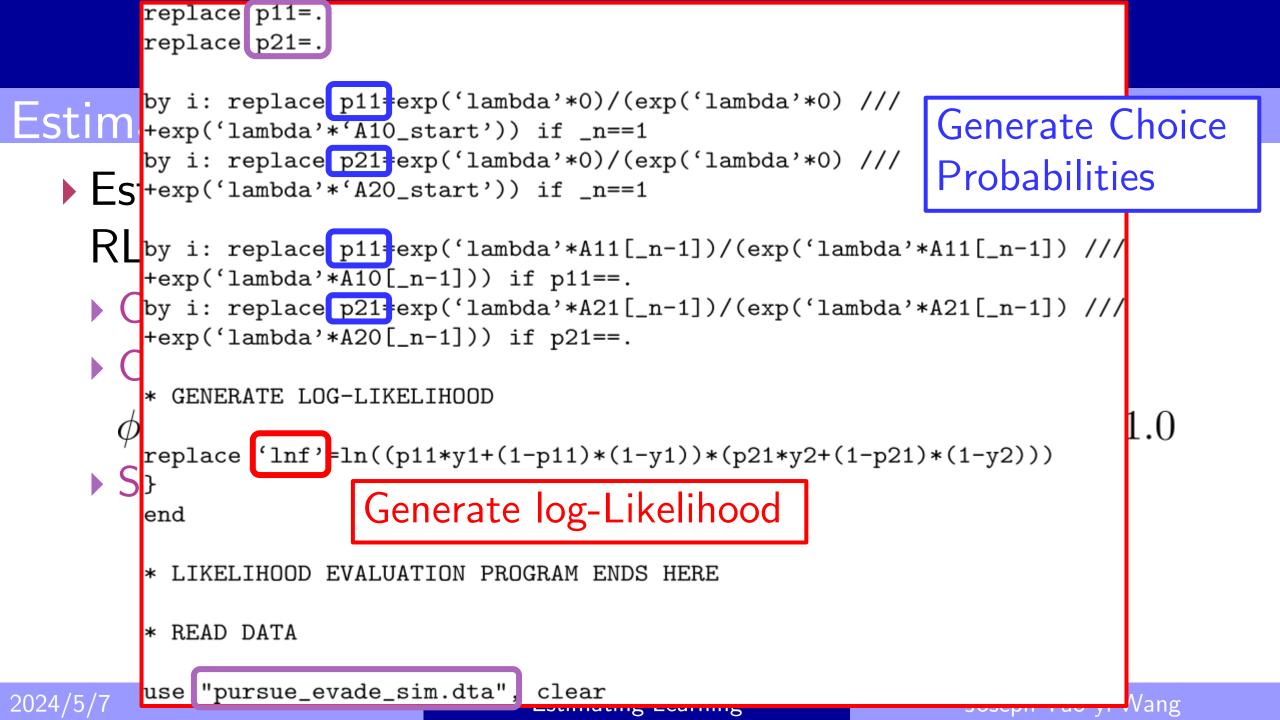
$$\phi = 0.94, \lambda = 0.80, A_1^0(0) = 1.0, A_2^0(0) = -2.0, N(0) = 1.0$$

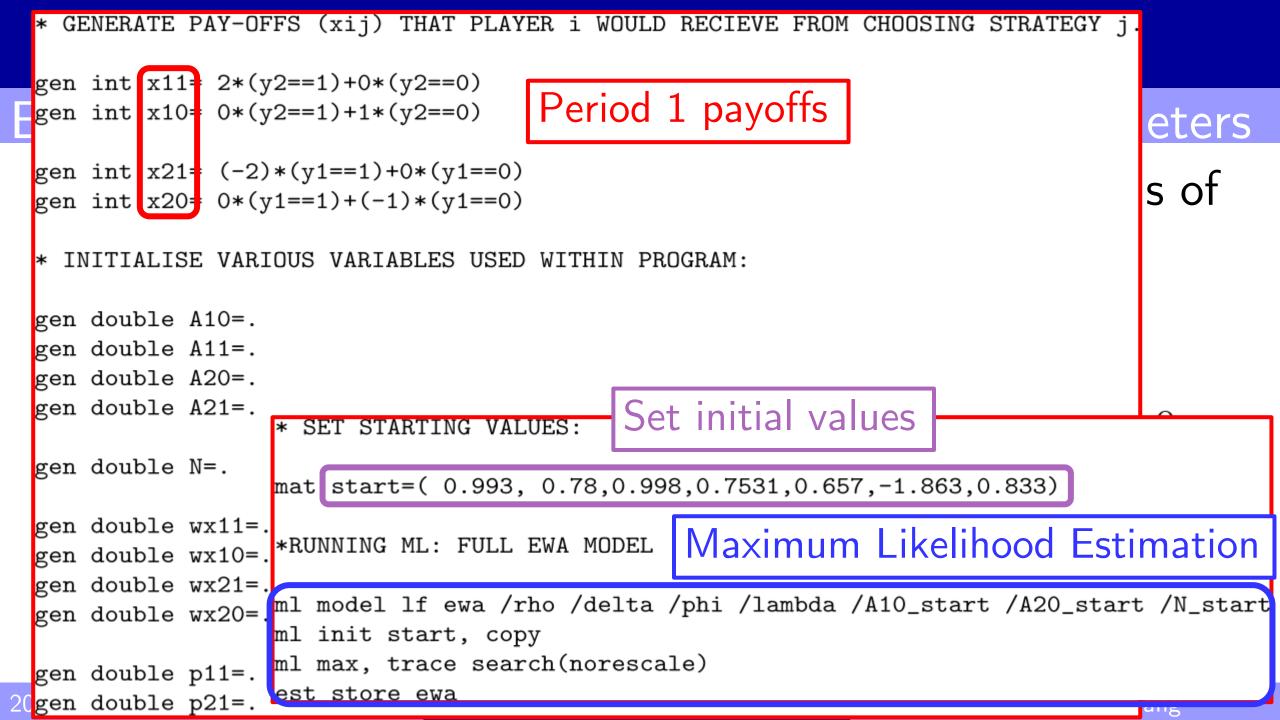
STATA Code



* GENERATE wxij (PAY-OFFS WEIGHTED BY DELTA PARAMETER):

Generate replace wx11= ('delta'+(1-'delta')*(y1))*x11 replace wx10= ('delta'+(1-'delta')*(1-y1))*x10 Weighted replace wx21= ('delta'+(1-'delta')*(y2))*x21 replace wx20= ('delta'+(1-'delta')*(1-y2))*x20 Payoffs ► Es⁻ * GENERATE PERIOD-1 ATTRACTIONS: RI by i: replace A10 ('phi'*'N_start'*'A10_start'+wx10)/N if _n==1 by i: replace A11+('phi'*'N_start'*0+wx11)/N if _n==1 by i: replace A20=('phi'*'N_start'*'A20_start'+wx20)/N if _n==1 by i: replace A21=('phi'*'N_start'*0+wx21)/N if _n==1 Generate Attractions \mathcal{O} * GENERATE ATTRACTIONS FOR t>1: >by i: replace A11=('phi'*N[_n-1]*A11[_n-1]+wx11)/N if A11==. by i: replace A10=('phi'*N[_n-1]*A10[_n-1]+wx10)/N if A10==. by i: replace A21=('phi'*N[_n-1]*A21[_n-1]+wx21)/N if A21==. by i: replace A20; ('phi'*N[_n-1]*A20[_n-1]+wx20)/N if A20==. * GENERATE p11 AND p21 (PROBABILITIES OF PLAYERS 1 and 2 CHOOSING STRATEGY 1) Generate Choice Probabilities replace p11=. 2024/5/7



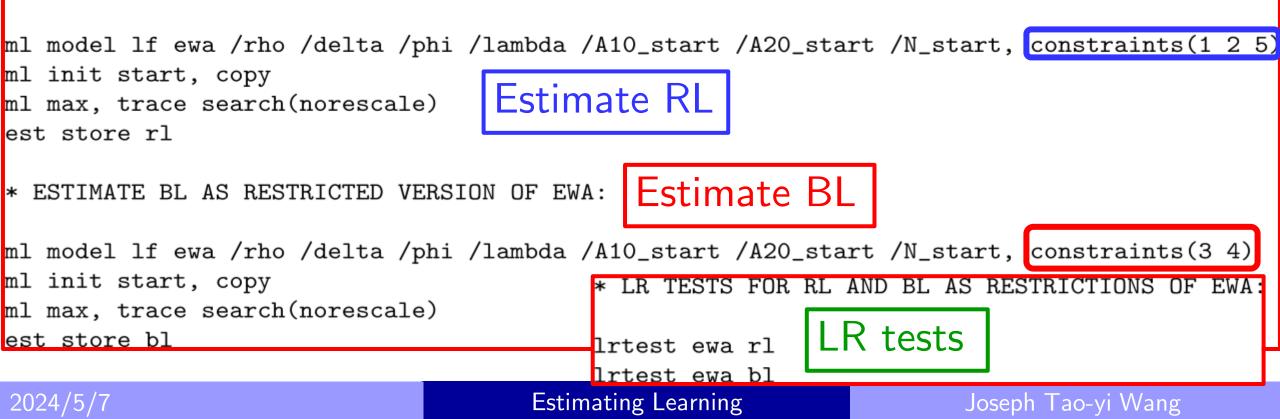


* DEFINE CONSTRAINTS REQUIRED FOR RL AND BL:

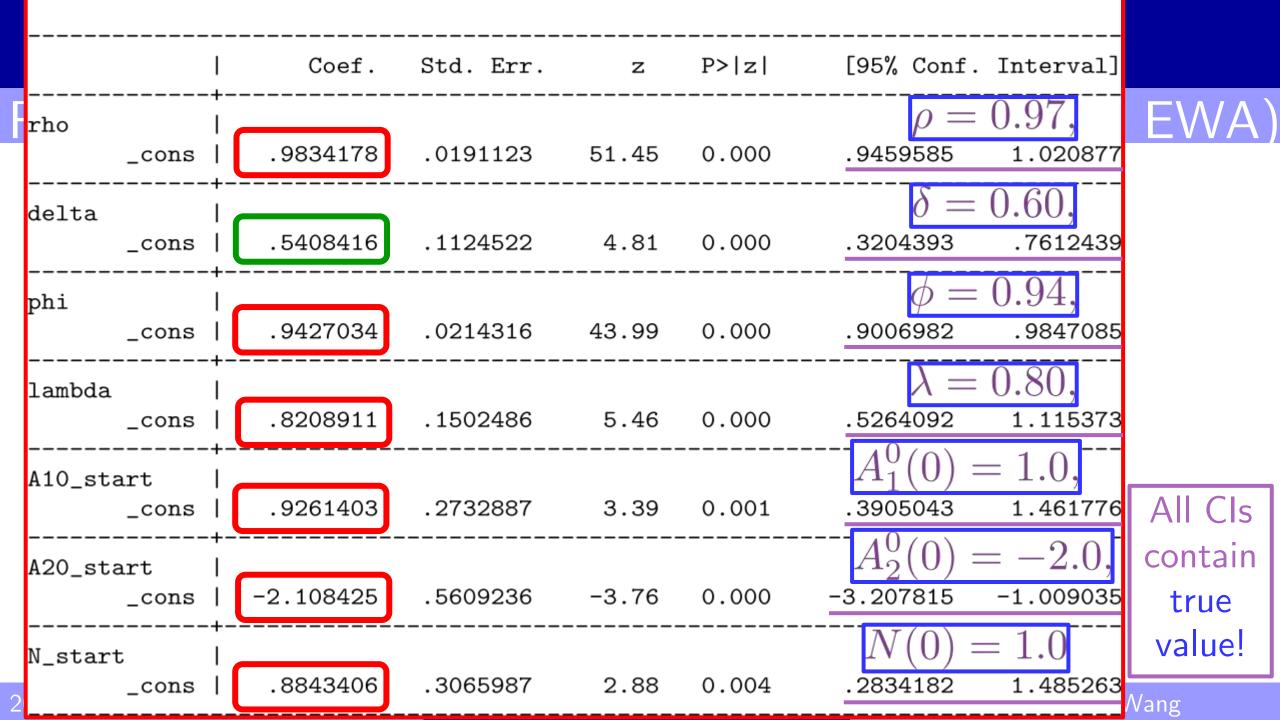
```
constraint 1 [delta]_b[_cons]=0.0
constraint 2 [rho]_b[_cons]=0.0
constraint 3 [delta]_b[_cons]=1
constraint 4 [rho]_b[_cons]=[phi]_b[_cons]
constraint 5 [N_start]_b[_cons]=1
```

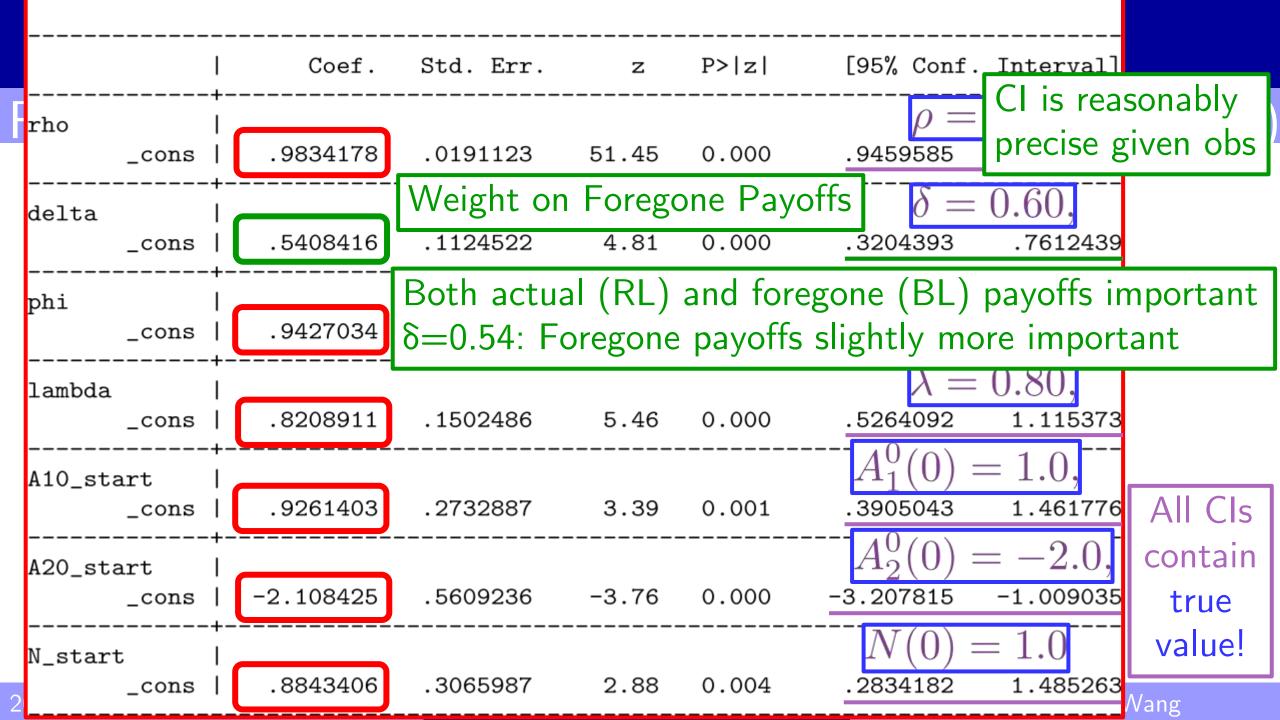


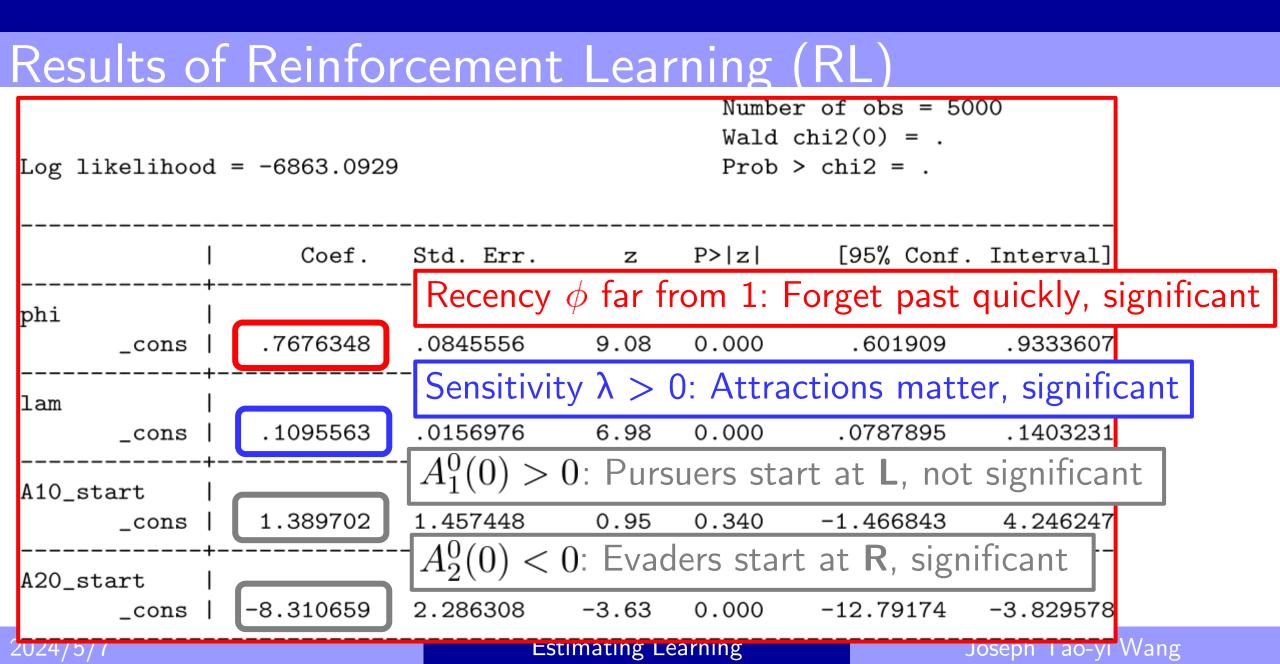
* ESTIMATE RL AS RESTRICTED VERSION OF EWA:

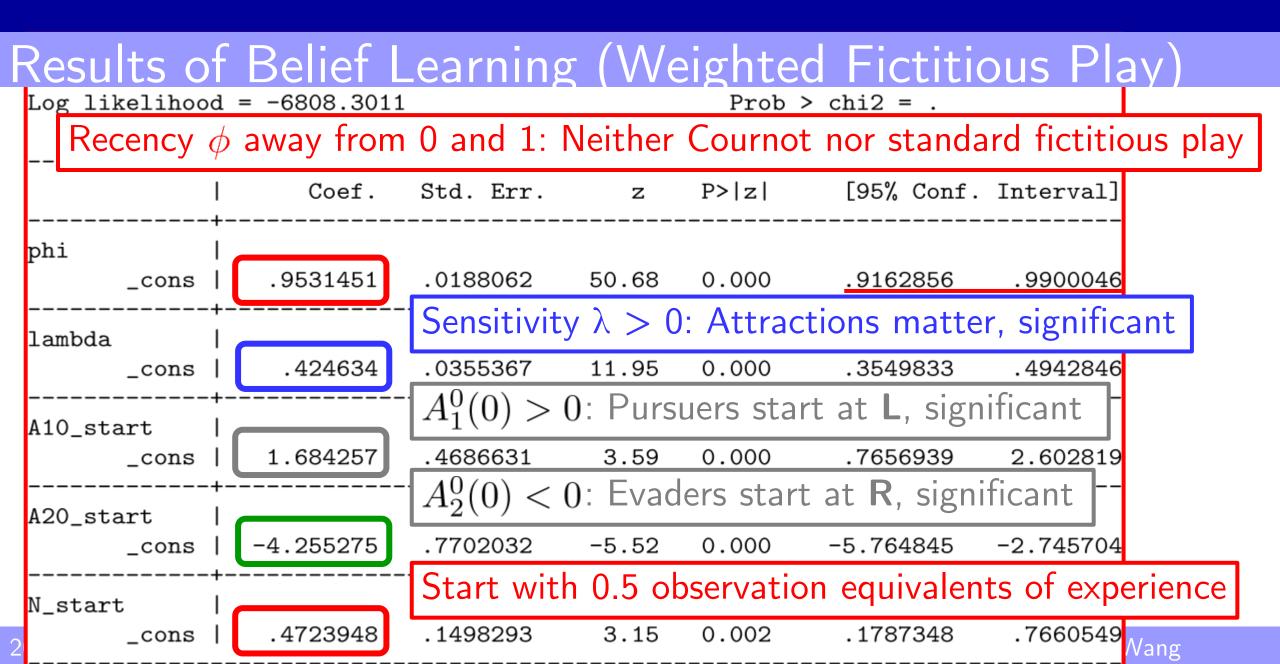


F	Resul	ts of	f Experie	enced-V	Veight	ed A	ttraction	s (Full	EWA)
	Log likelihood = -6800.9162			Number of obs = 5000 Wald chi2(0) = . Prob > chi2 = .					
			Coef.	Std. Err.	Z	P> z	[95% Conf.	Interval]	
	rho	 _cons	.9834178	.0191123	51.45	0.000	.9459585	1.020877	
	delta	 _cons	.5408416	.1124522	4.81	0.000	.3204393	.7612439	
	phi	 _cons	.9427034	.0214316	43.99	0.000	.9006982	.9847085	
	lambda	_cons	.8208911	.1502486	5.46	0.000	.5264092	1.115373	
2	10 at at	+ ~+							Nang





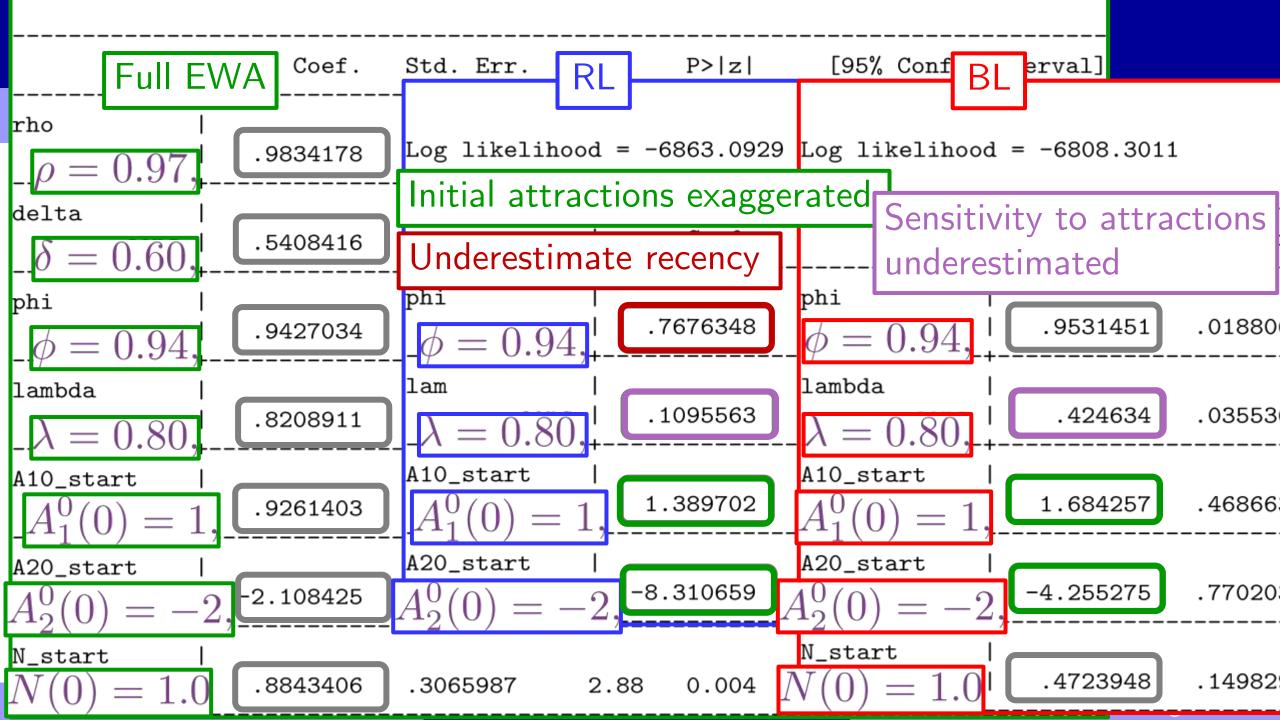




LR Test Results

Model	Log-L	LR	df	p-value
EWA	-6800.92			
RL	-6863.09	124.34	3	0.0000
BL	-6808.30	14.76	2	0.0006

Both RL and BL strongly rejected by LR test
Not surprising since we simulated EWA data with δ=0.6
In between RL and BL (but slightly closer to BL)



Acknowledgment

- This presentation is based on
 - Section 18.1-18.5 of the textbook on Experimetrics,
- An extension of the mini-course taught by Peter G. Moffatt (UEA) at National Taiwan University in Spring 2019