Reinforcement Learning Lab

Odelia Schwartz
2016

Rescorla-Wagner rule (1972)

- Minimize difference between received reward and predicted reward
- Binary variable u (1 if stimulus is present; 0 if absent)
- Predicted reward v
- Linear weight w

$$v = wu$$

- If stimulus u is present:

$$v = w$$

based on Dayan and Abbott book

Rescorla-Wagner rule (1972)

- Minimize squared error between received reward r and predicted reward v:

$$(r - v)^2$$

(average over presentations of stimulus and reward)

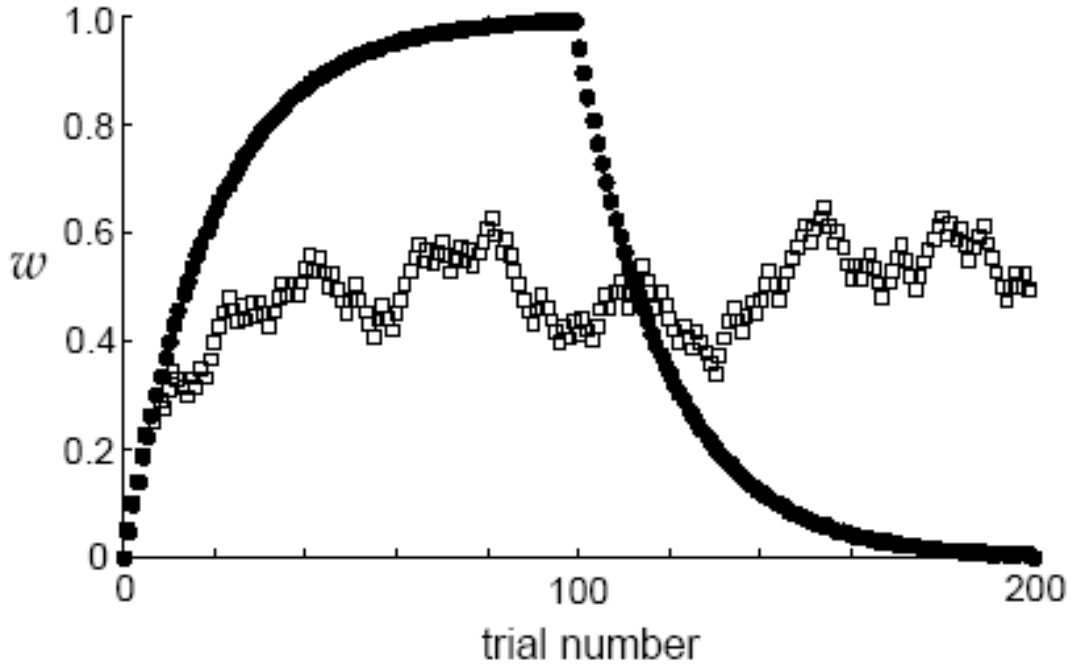
- Update weight:

$$w \rightarrow w + \varepsilon(r - v)u$$

ε learning rate

Also known as delta learning rule: $\delta = r - v$

Acquisition and extinction



- Solid: First 100 trials: reward (r=1) paired with stimulus; next 100 trials no reward (r=0) paired with stimulus (learning rate .05)
- Dashed: Reward paired with stimulus randomly 50 percent of time

From Dayan and Abbott book

Temporal Difference Learning

Want $v_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \ldots$

(here t represents time within a trial; reward can come at any time within a trial. Sutton and Barto interpret v_t as the prediction of total future reward expected from time t onward until the end of the trial)

Based on Dayan slides; Daw slides

Temporal Difference Learning

Want $v_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \ldots$

(here t represents time within a trial; reward can come at any time within a trial. Sutton and Barto interpret v_t as the prediction of total future reward expected from time t onward until the end of the trial)

Prediction error:

$$\delta_t = (r_t + r_{t+1} + r_{t+2} + r_{t+3} \ldots) - V_t$$

Temporal Difference Learning

Want $v_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \cdots$

(here t represents time within a trial)

But we don't want to wait forever for all future rewards…

$r_{t+1}; r_{t+2}; r_{t+3} \cdots$

Temporal Difference Learning

Want $v_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} \ldots$

(here t represents time within a trial)

Recursion "trick": $v_t = r_t + v_{t+1}$

Based on Dayan slides; Daw slides

Temporal Difference Learning

From recursion want:

$$v_t = r_t + v_{t+1}$$

Error:

$$\delta_t = r_t + v_{t+1} - v_t$$

Temporal Difference Learning

From recursion want: $v_t = r_t + v_{t+1}$

Error: $\delta_t = r_t + v_{t+1} - v_t$

Update: $v_t \rightarrow v_t + \varepsilon(r_t + v_{t+1} - v_t)$

$= (1- \varepsilon)v_t + \varepsilon(r_t + v_{t+1})$

RV versus TD

- Rescorla-Wagner error: (n represents trial)

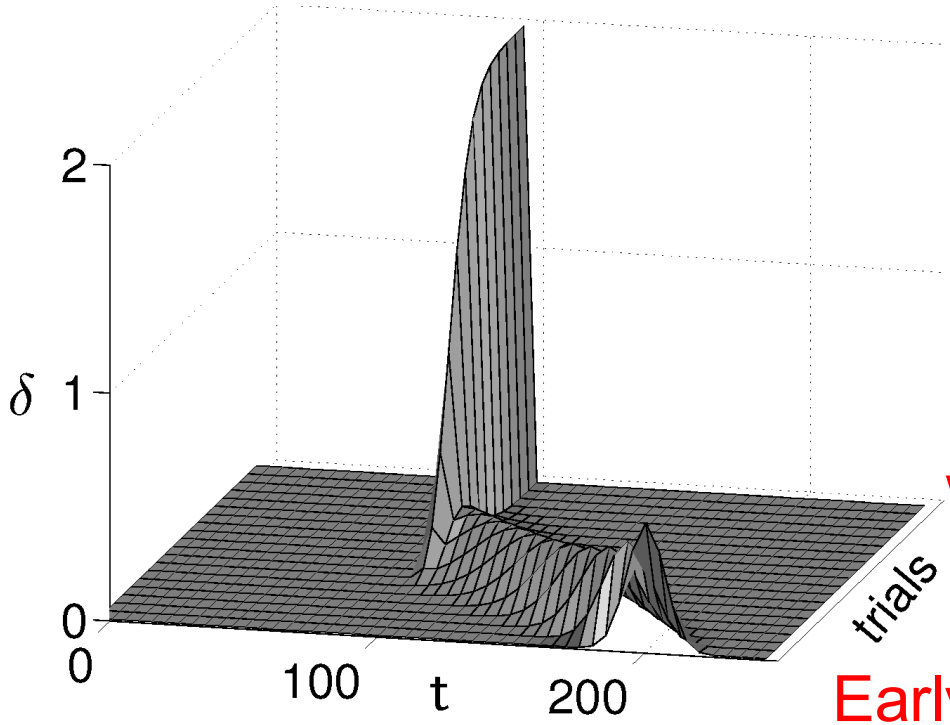
$$\delta_n = r_n - v_n$$

- Temporal Difference Error: (t is time within a trial)

$$\delta_t = r_t + v_{t+1} - v_t$$

Updates are causal

Temporal Difference Learning



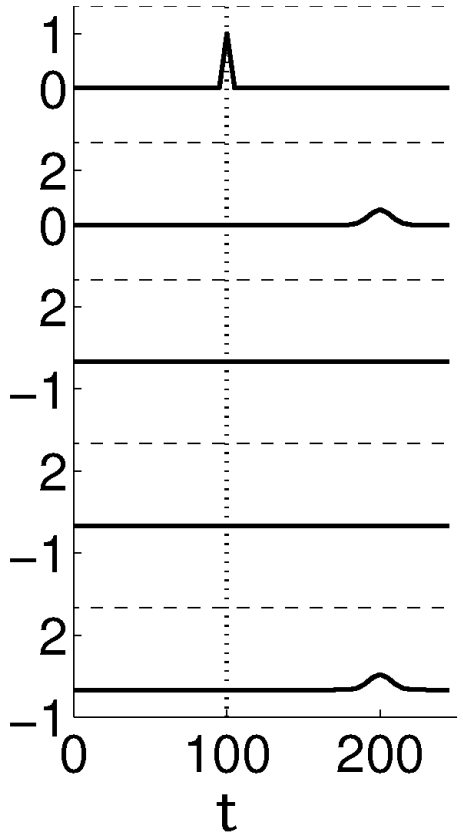
Late trials: error when stimulus

Early trials: error when reward

Dayan and Abbott Book: Surface plot of prediction error (stimulus at 100; reward at 200)

Temporal Difference Learning

Before learning



u_t

r_t

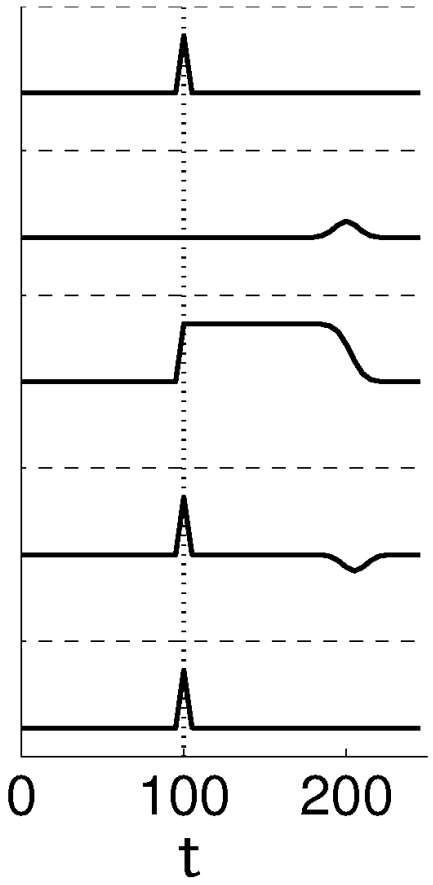
v_t

$v_{t+1} - v_t$

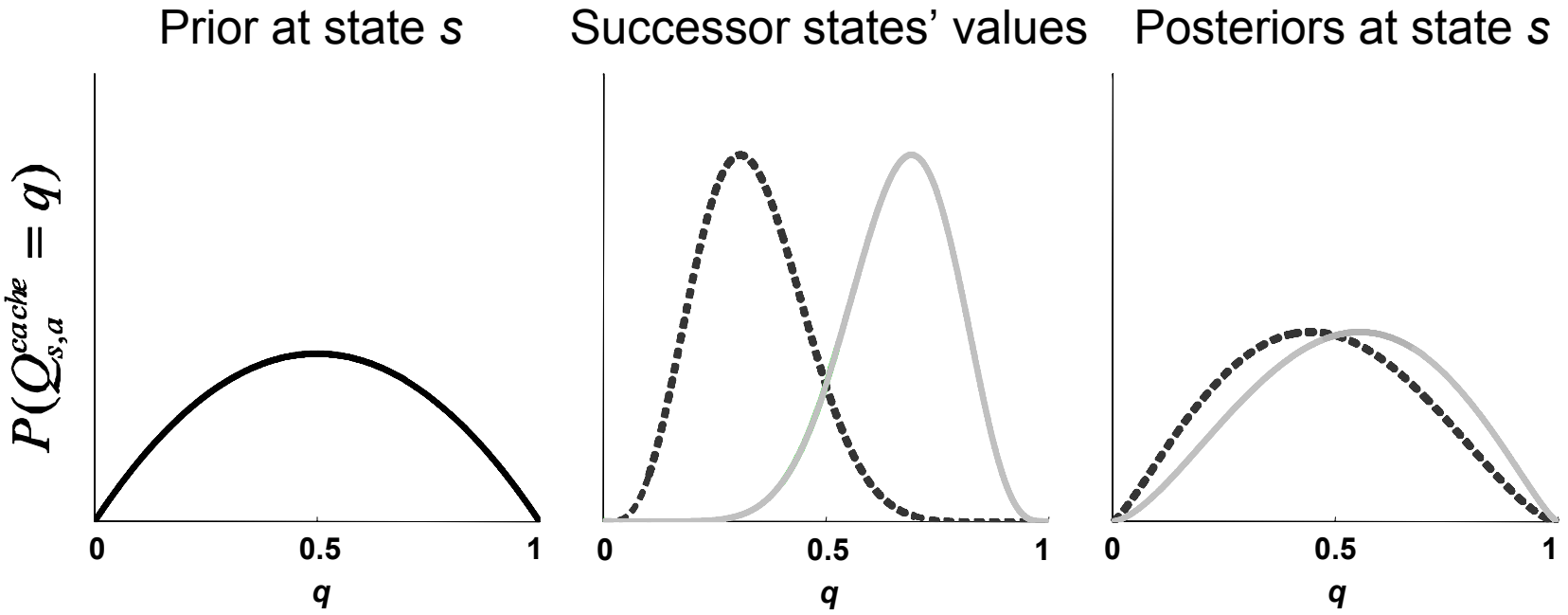
$\delta_t = r_t + v_{t+1} - v_t$

Temporal Difference Learning

After learning



Whole distribution: Daw et al. 2015 paper

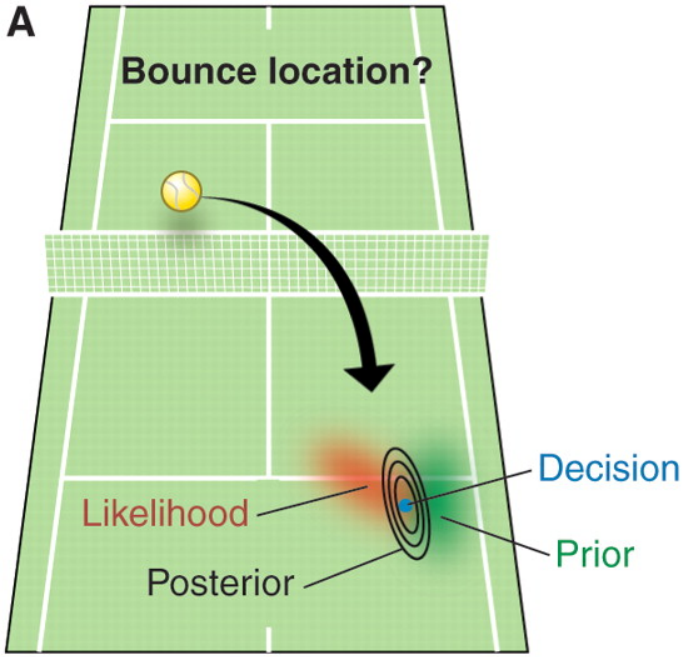


Two example Successors – more and less favorable

Prior nudged in direction of each successor

Learning both a mean and a variance: uses Beta distribution

Bayesian inference



Koerding 2007