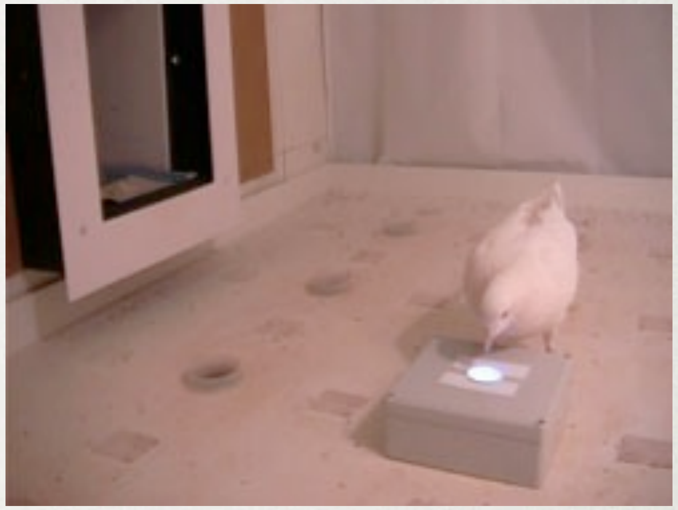
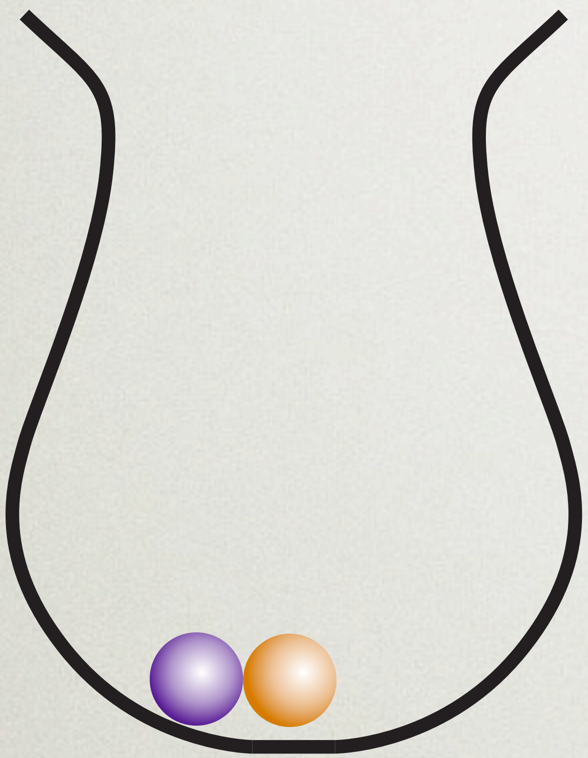
Thorndike (1911)



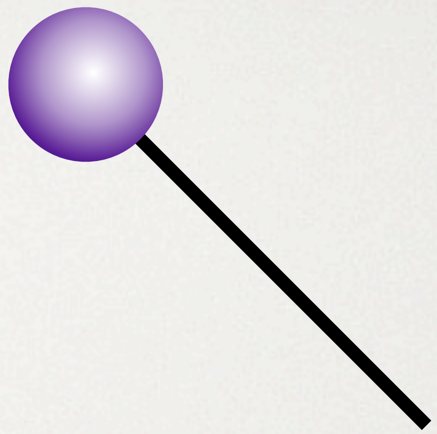
Skinner

Herrnstein (1970)

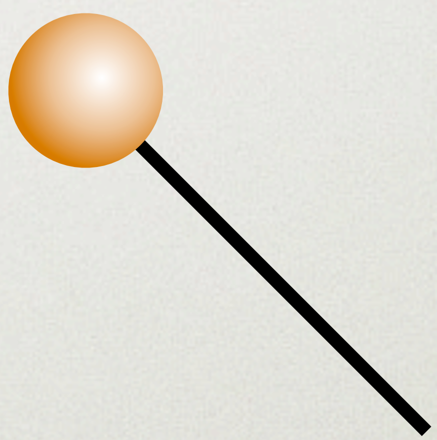
Roth and Erev



Urn that tracks your dispositions

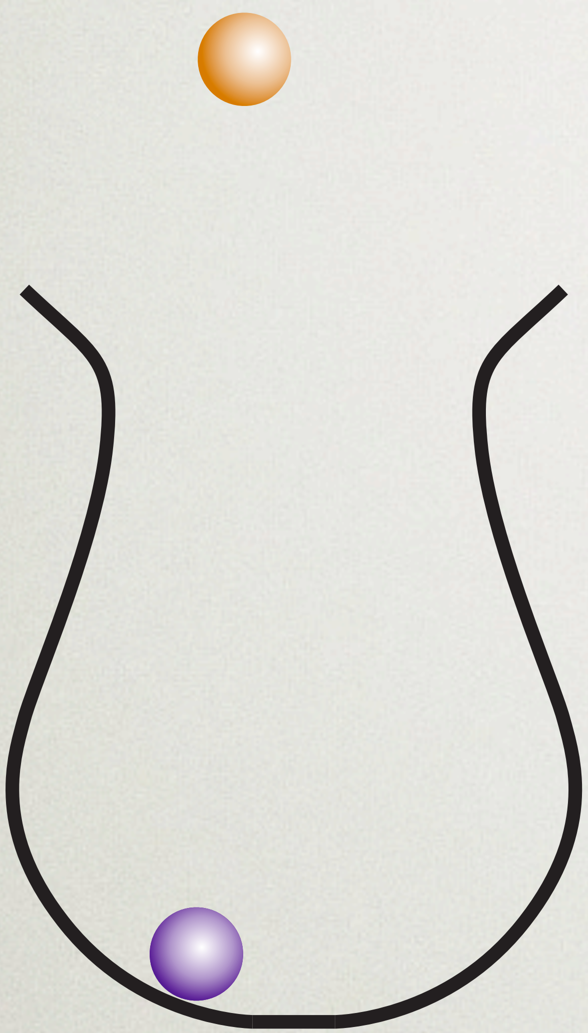


Slot machine 1
pays off with probability p

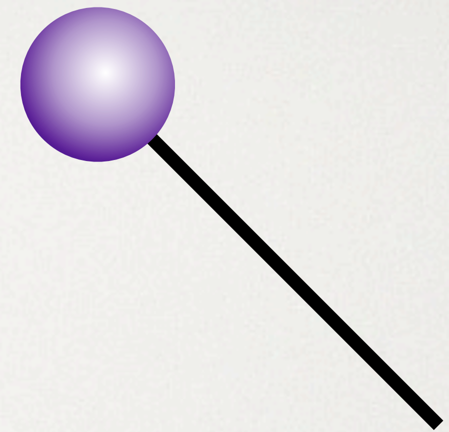


Slot machine 2
pays off with probability q

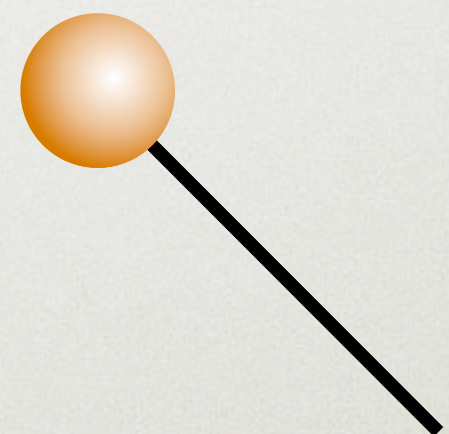
1. Draw a random ball from the urn



Urn that tracks your dispositions

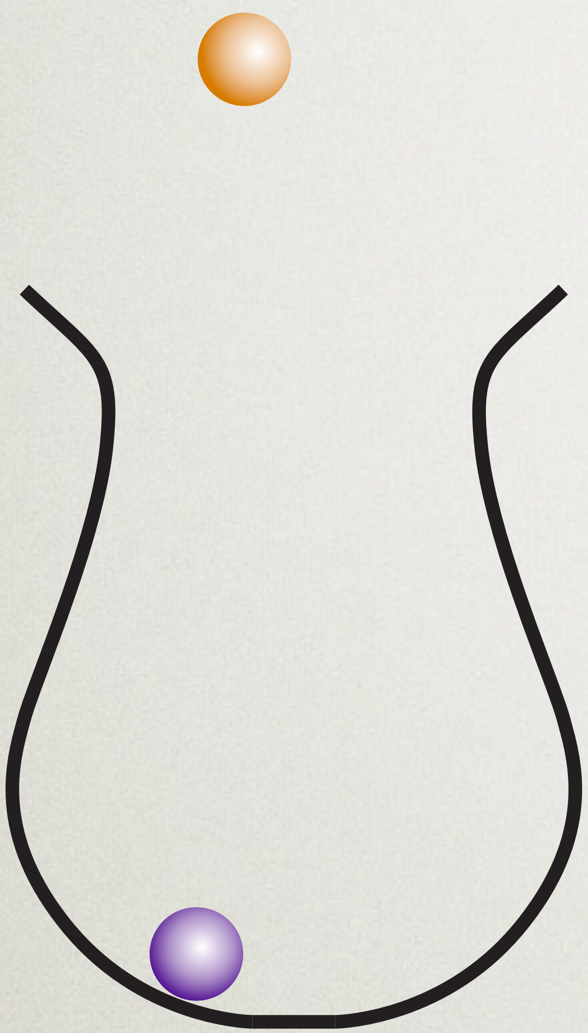


Slot machine 1
pays off with probability p

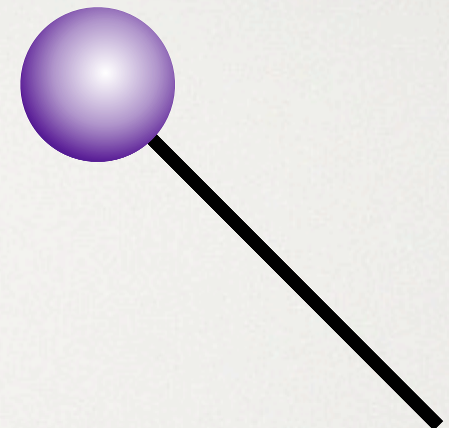


Slot machine 2
pays off with probability q

1. Draw a random ball
from the urn

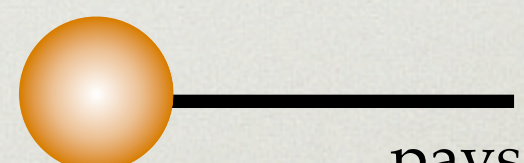


Urn that tracks your dispositions

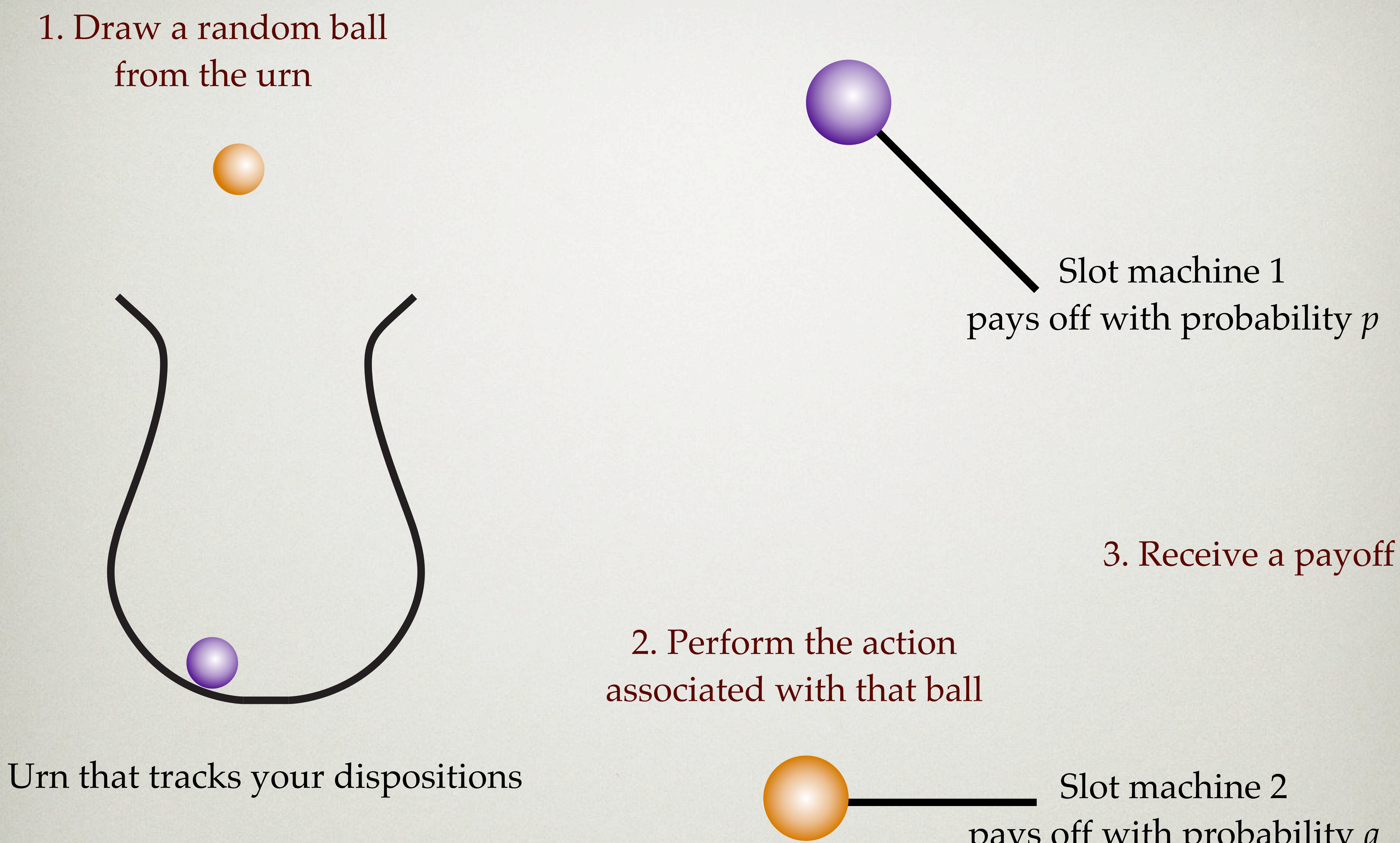


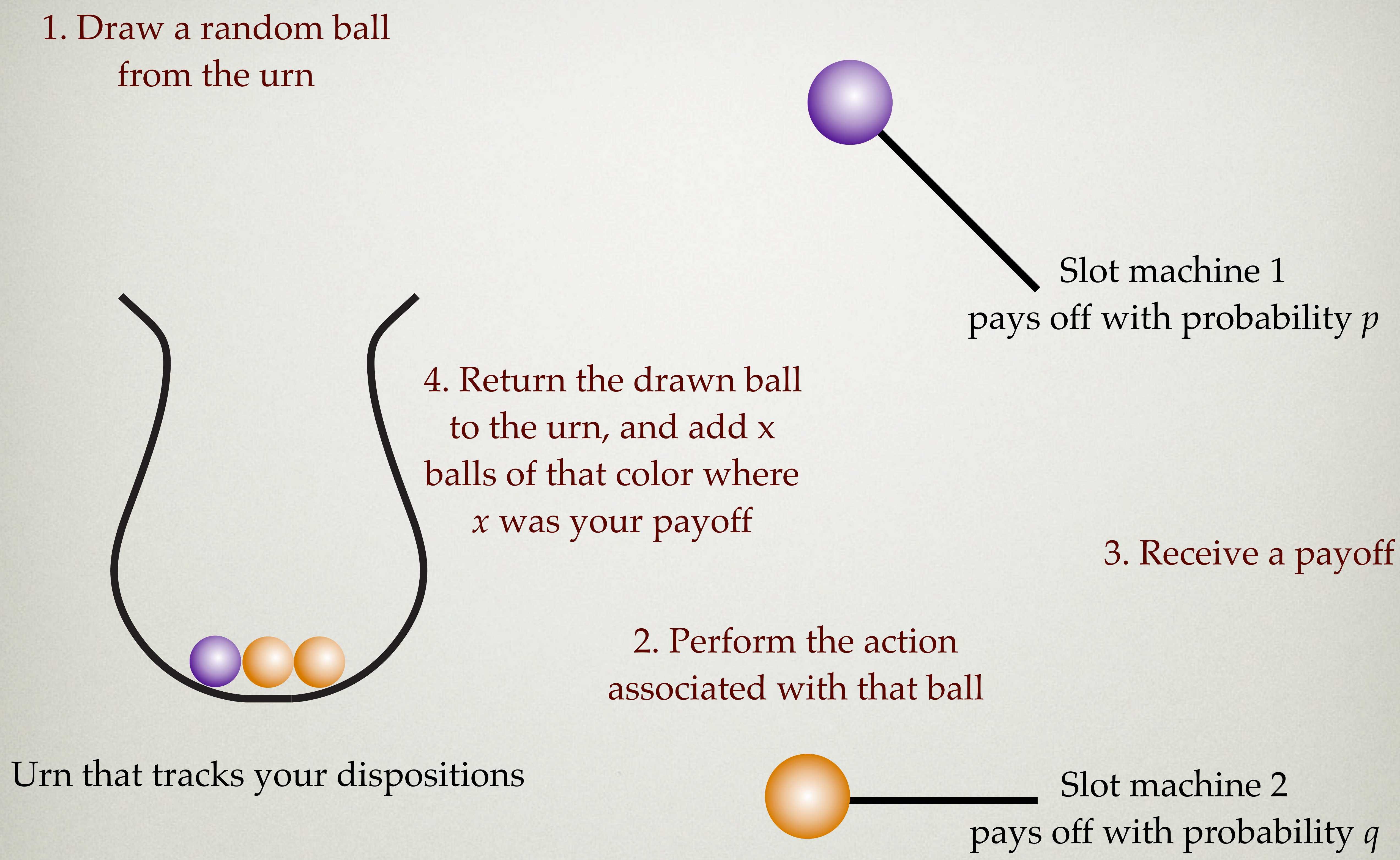
Slot machine 1
pays off with probability p

2. Perform the action
associated with that ball

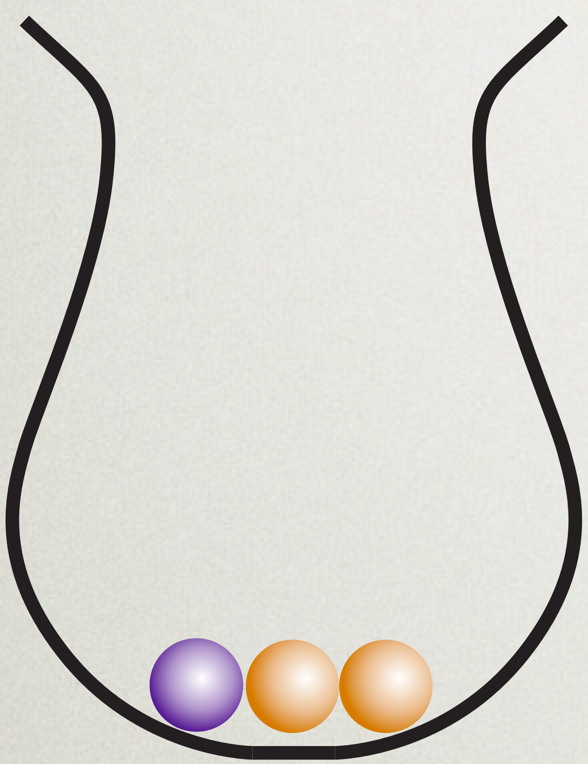


Slot machine 2
pays off with probability q

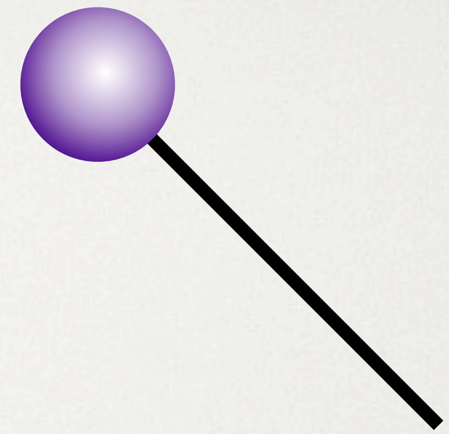

1. Draw a random ball from the urn
Urn that tracks your dispositions
Slot machine 1 pays off with probability p
3. Receive a payoff
2. Perform the action associated with that ball
Slot machine 2 pays off with probability q


1. Draw a random ball
from the urn
Slot machine 1
pays off with probability p
4. Return the drawn ball
to the urn, and add x
balls of that color where
x was your payoff
3. Receive a payoff
2. Perform the action
associated with that ball
Urn that tracks your dispositions
Slot machine 2
pays off with probability q

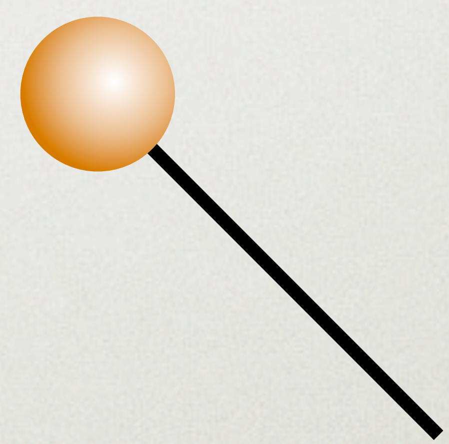
Repeat



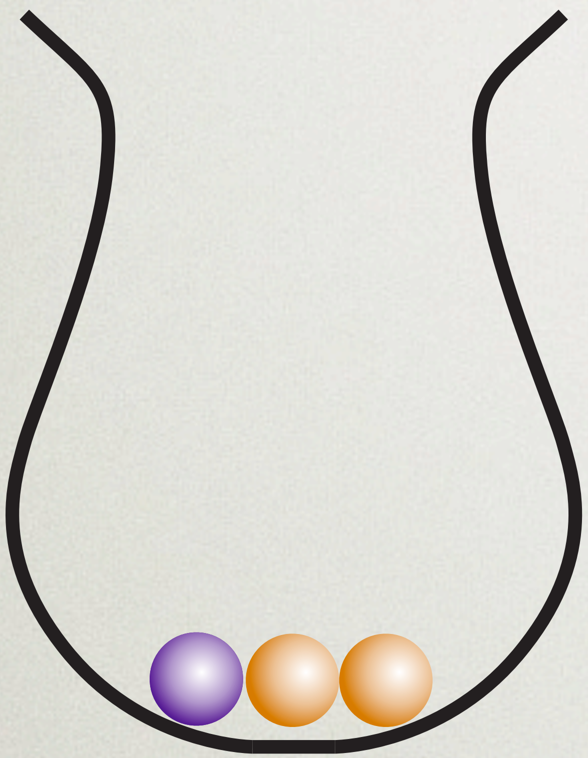
Urn that tracks your dispositions



Slot machine 1
pays off with probability p



Slot machine 2
pays off with probability q



Urn that tracks your dispositions

Law of effect:

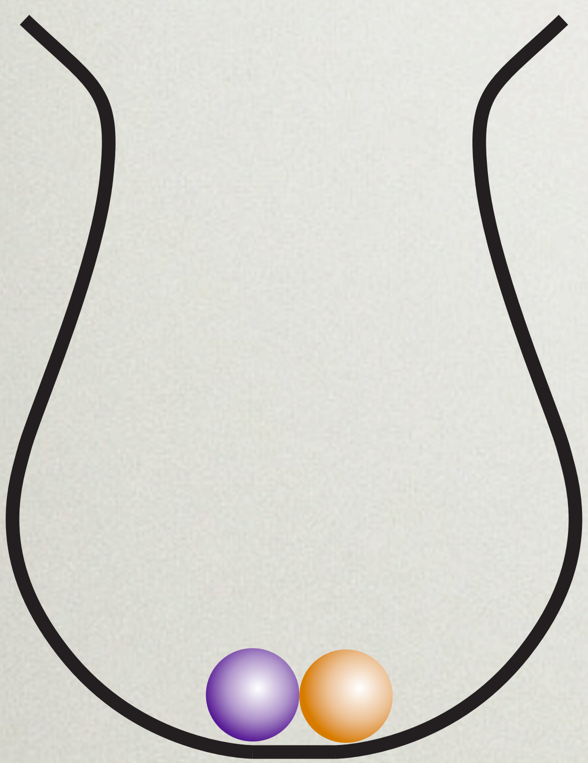
Actions that are followed by the satisfaction of desires are reinforced so that, when the desire recurs the action that satisfied the desire will be more likely to be performed again.

Law of practice:

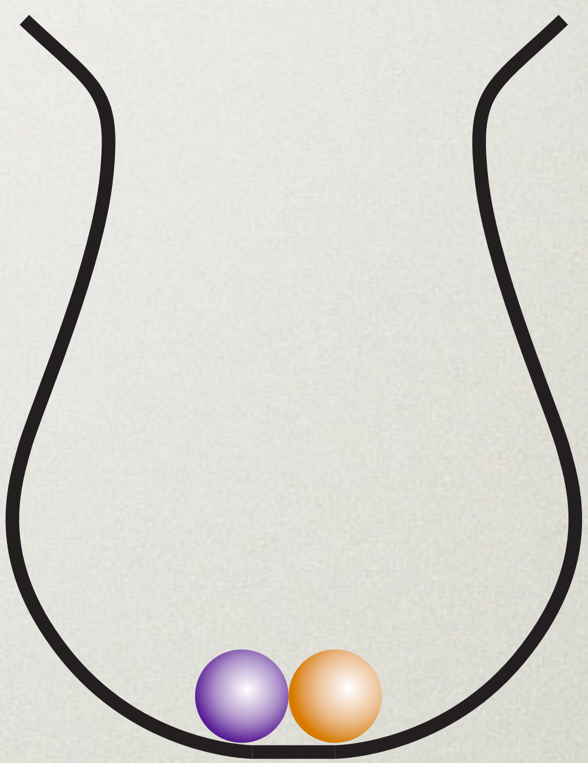
Learning slows down over time.

This model can also be used for games
Roth and Erev (1995)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



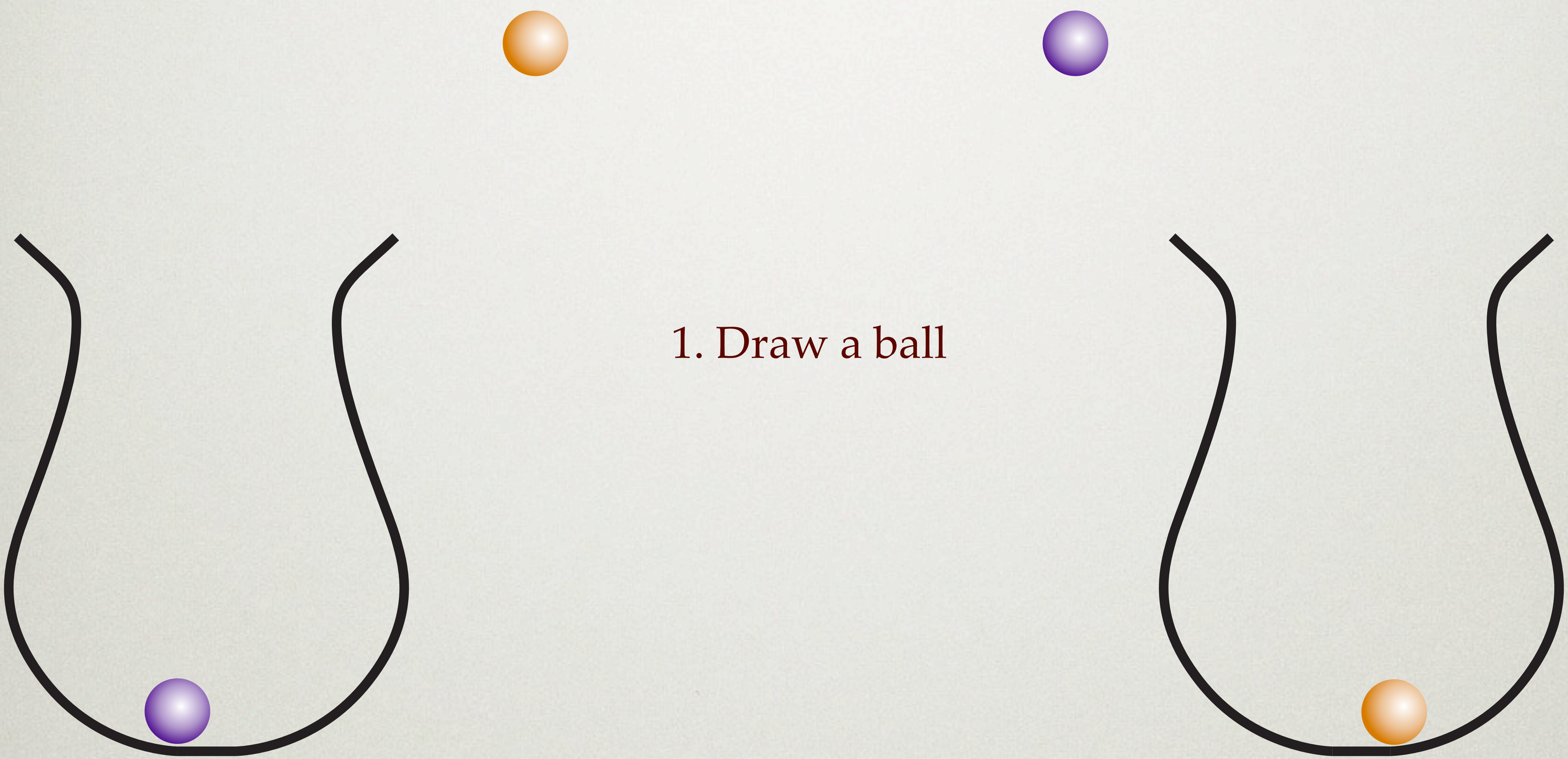
Player 1's urn



Player 2's urn

This model can also be used for games
Roth and Erev (1995)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



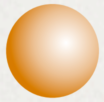
Player 1's urn

Player 2's urn

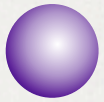
This model can also be used for games
Roth and Erev (1995)

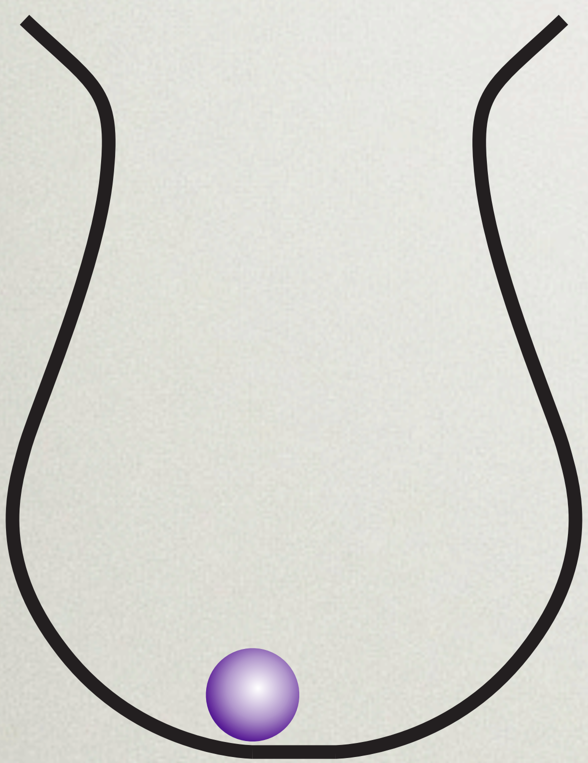
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Player 1 uses action 2

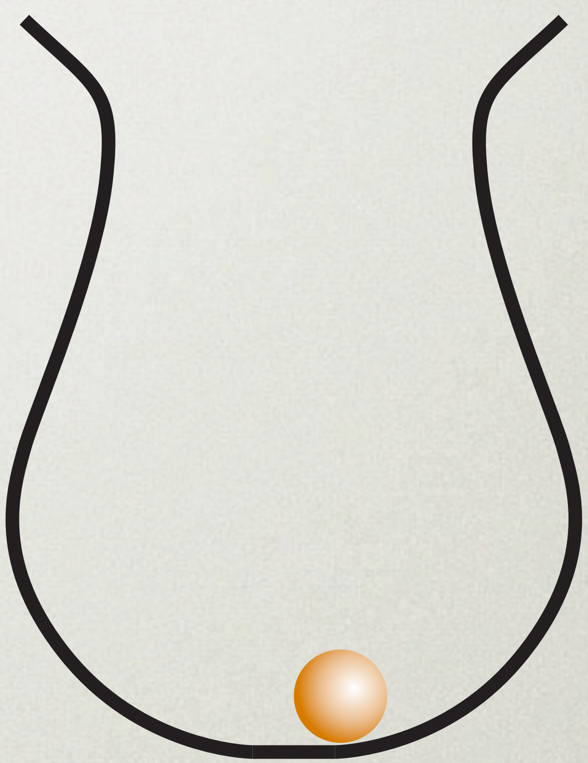


Player 2 uses action 1





1. Draw a ball
2. Perform the corresponding action



Player 1's urn

Player 2's urn

This model can also be used for games
Roth and Erev (1995)

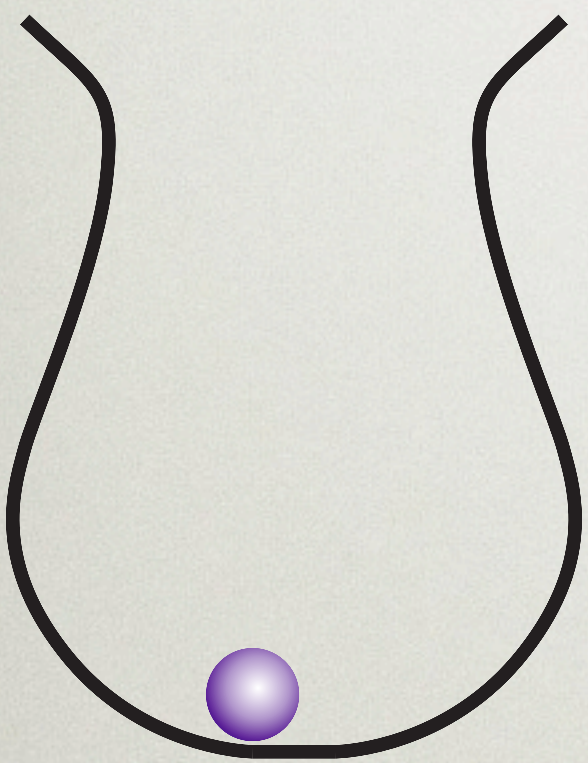
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Player 1 uses action 2

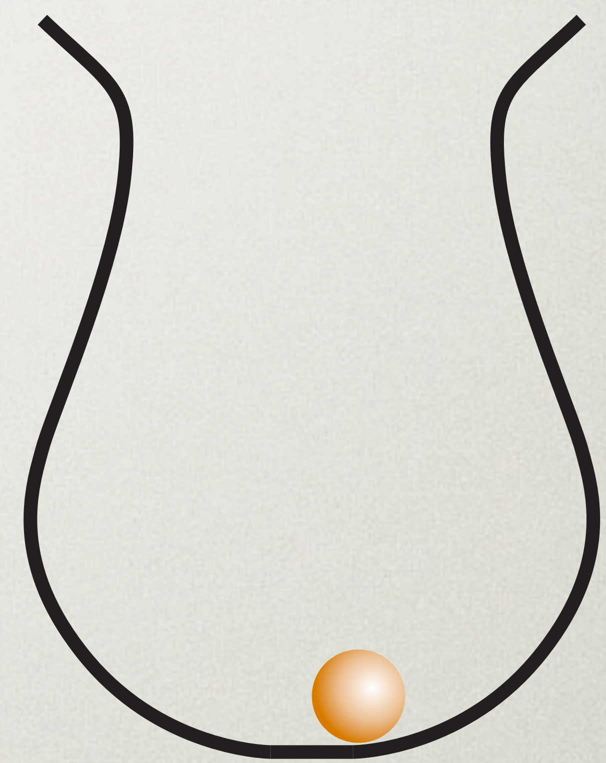
Player 2 uses action 1

Both players receive
a payoff of zero

1. Draw a ball
2. Perform the corresponding action
3. Receive payoffs



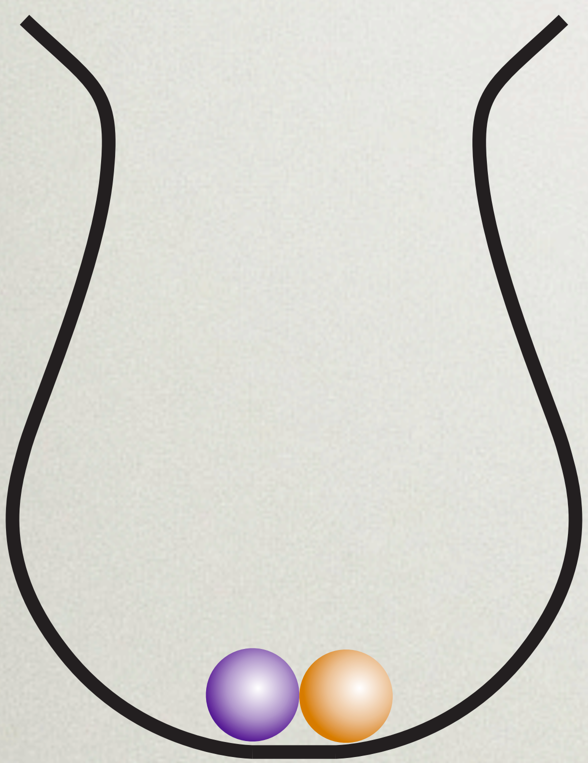
Player 1's urn



Player 2's urn

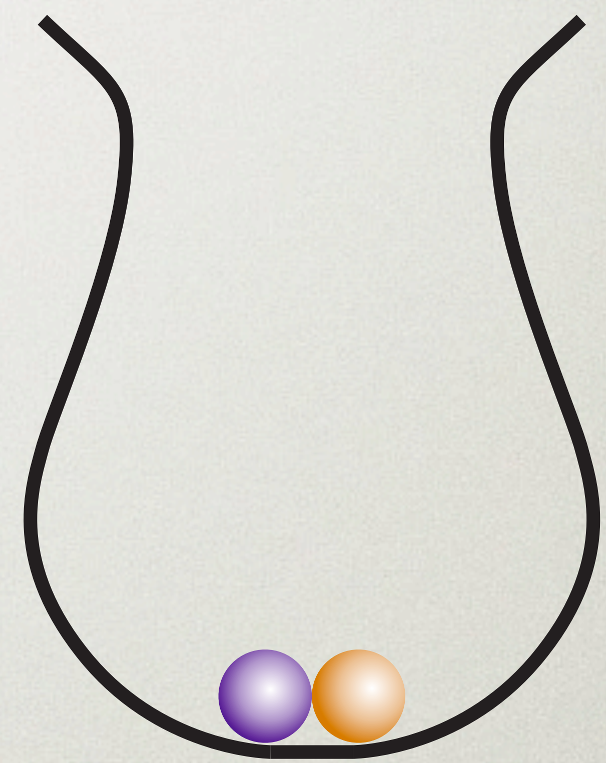
This model can also be used for games
Roth and Erev (1995)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



Player 1's urn

1. Draw a ball
2. Perform the corresponding action
3. Receive payoffs
4. Return balls and reinforce according to payoffs



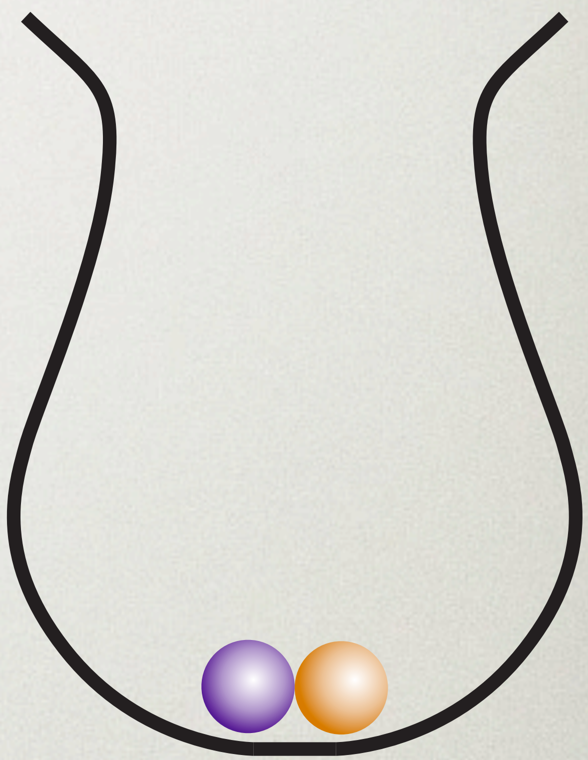
Player 2's urn

There are many possible variations of this process...

1. What number of balls are initially in the urns?
   (As described so far: urns are initially stocked with one ball of each type)

2. How does one convert balls in urns into probabilities for actions?
   (As described so far: every ball has an equal probability of being drawn and you perform the action associated with that ball)

3. How does one translate payoffs of the game into additional balls in the urn?
   (As described so far: you add a number of balls that is equal to the you payoff)



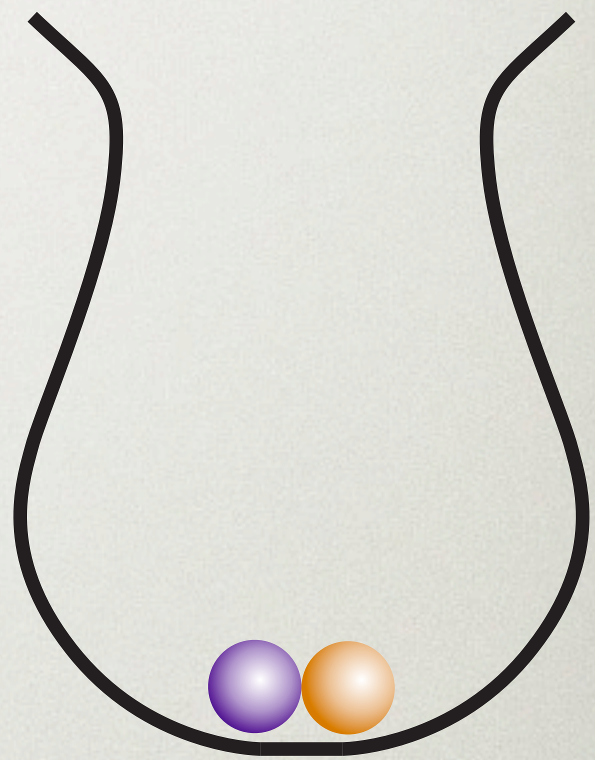
Player 1's urn

Another derivation of the replicator dynamic!

The mean field dynamic of Roth-Erev reinforcement learning (with equal initial weights, a proportional response rule, and reinforcement according the payoffs) is equal to the replicator dynamic!

Beggs (2005)

Hopkins and Posch (2005)



Player 1's urn

The replicator dynamic is everywhere!

Biological reproduction

1. Large population of individuals, reproduce continuously with rates equal to their payoffs

2. Large population of individuals, reproduce continuously with rates equal to the change in population given by the discrete-time replicator dynamic

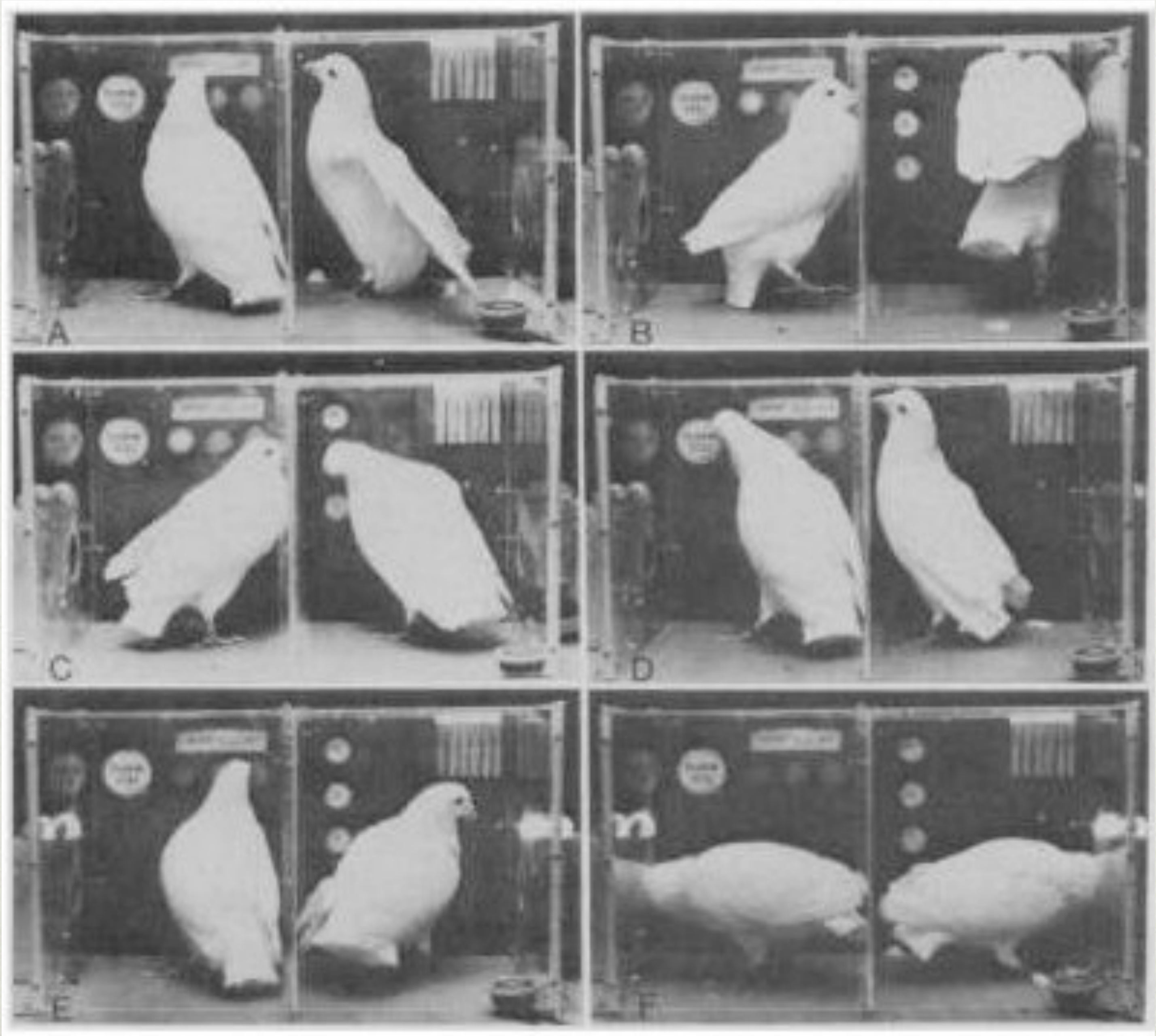
Social learning via imitation

3. Large population of individuals who revise their strategy in the game through pairwise proportional imitation

Individual learning via reinforcement

4. Some number of individuals whose dispositions for action change according to simple Roth-Erev reinforcement learning

We want to know what happens when reinforcement learners play signaling games



Epstein et al. (1980)

So what happens when reinforcement learners play signaling games?

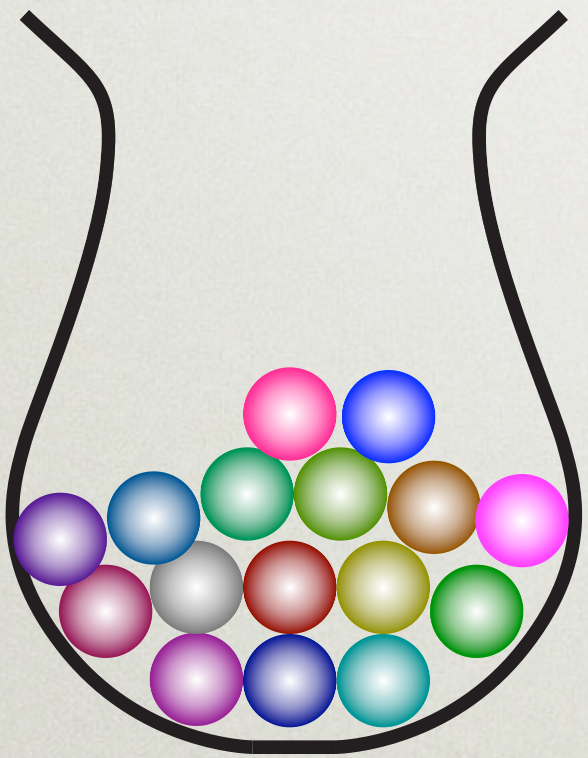
Method 1:

Two players in the game.

Each player has one urn.

Balls correspond to pure strategies in the signaling game.

Payoffs are given by the expected payoffs in the symmeterized signaling game (i.e., the 16 by 16 normal form payoff matrix)



Player 1's urn

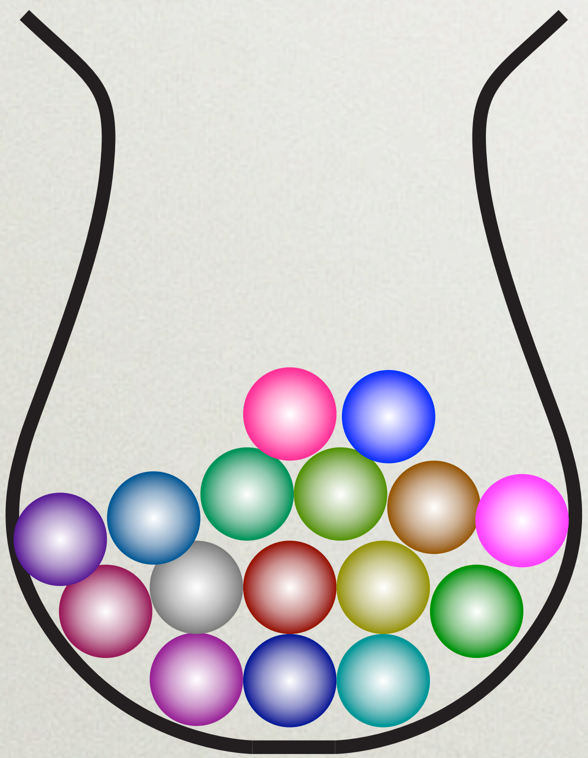
Player 2's urn

So what happens when reinforcement learners play signaling games?

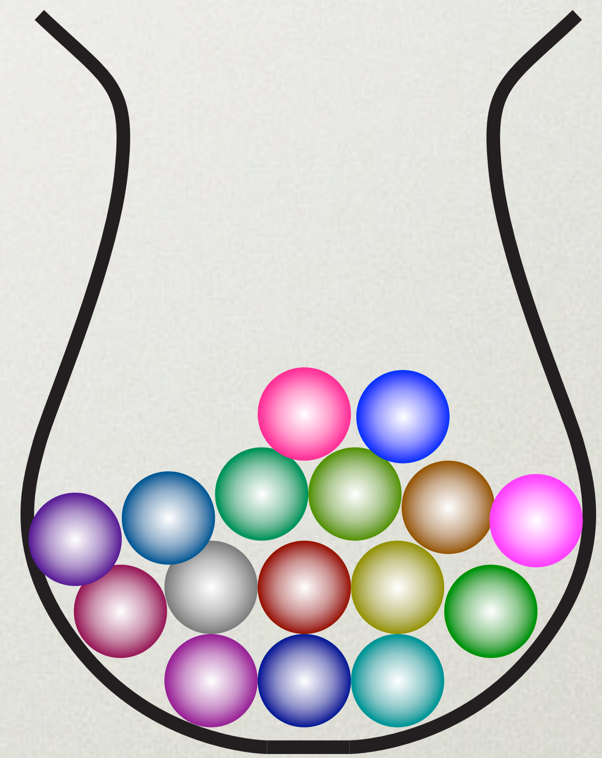
Method 1:

In the long run, this system will behave exactly like the replicator dynamic.

We already know what happens in the replicator dynamic, so we already know what will happen here!



Player 1’s urn



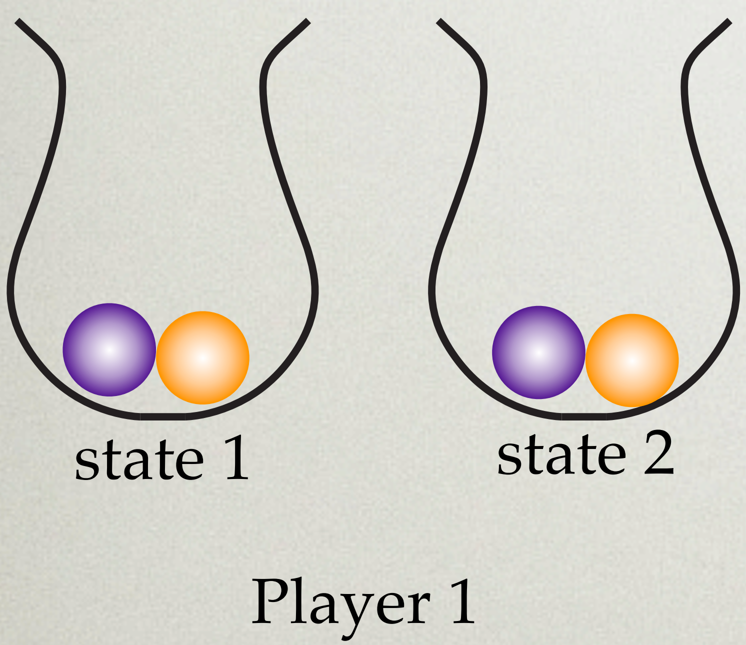
Player 2’s urn

So what happens when reinforcement learners play signaling games?

Method 2:

Player 1 gets an urn for every state
Player 2 gets an urn for every message

Note that this assumes very little sophistication. The players don't need to understand that the different states are part of the same interaction, and they don't even need to understand that they're playing a game!

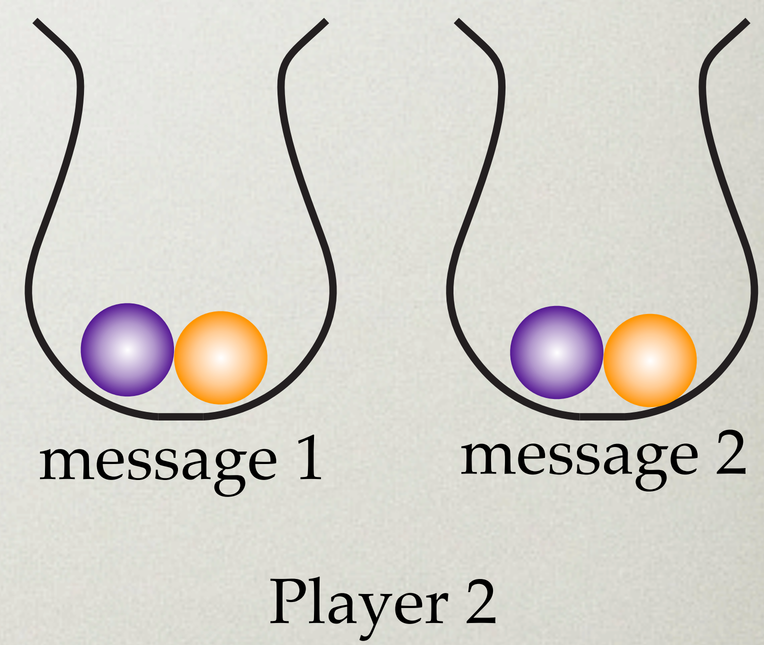


1. Nature determines the state

2. The sender chooses a message by drawing a ball from the appropriate urn

3. The receiver chooses an action by drawing a ball from the appropriate urn

4. Both players reinforce according to their payoffs



So what happens when reinforcement learners play signaling games?

Roth-Erev reinforcement method 2:

2 states, 2 messages, 2 actions, equiprobable states:

> Argiento et al. (2009, Theorem 1):
>
> In two state, two message, two action signaling game with equiprobable states...
>
> With probability 1, the player's expected payoffs $\rightarrow 1$ as the number of learning rounds $\rightarrow \infty$

So what happens when reinforcement learners play signaling games?

Roth-Erev reinforcement method 2:

Non-equiprobable states:

Simulation results (first due to Huttegger [unpublished]):
When the states are not equiprobable, some simulations appear to converge to pooling equilibria.

$P(s_1) = .8 \Rightarrow$ 22% of simulations lead to pooling

$P(s_1) = .9 \Rightarrow$ 44%

More than two states:

Simulation results (first due to Huttegger [unpublished] or Barrett [2006]):
When there are more than two states, some simulations appear to converge to partial pooling equilibria.

4 states $\Rightarrow$ ~30% of simulations lead to partial pooling

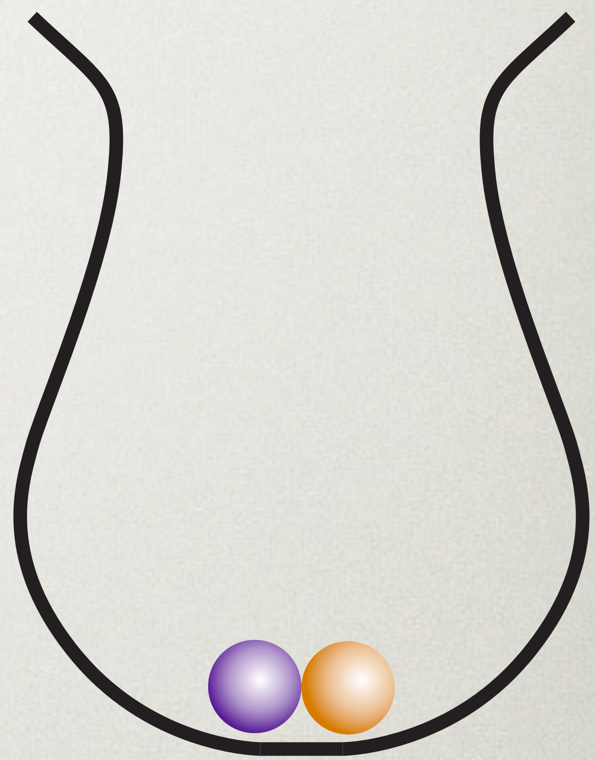
8 states $\Rightarrow$ ~60%

There are many possible variations of this process...

1. What number of balls are initially in the urns?
   (As described so far: urns are initially stocked with one ball of each type)

2. How does one convert balls in urns into probabilities for actions?
   (As described so far: every ball has an equal probability of being drawn and you perform the action associated with that ball)

3. How does one translate payoffs of the game into additional balls in the urn?
   (As described so far: you add a number of balls that is equal to the you payoff)



Player 1's urn

Variations on simple reinforcement learning

What happens if we adjust the initial weights?

Small initial weights make pooling very unlikely

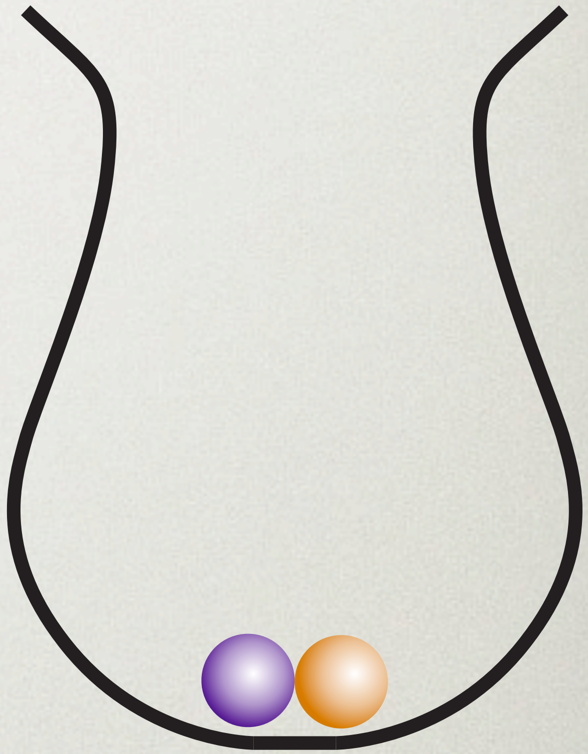
Large initial weights make pooling very likely

There are many possible variations of this process...

1. What number of balls are initially in the urns?
   (As described so far: urns are initially stocked with one ball of each type)

2. How does one convert balls in urns into probabilities for actions?
   (As described so far: every ball has an equal probability of being drawn and you perform the action associated with that ball)

3. How does one translate payoffs of the game into additional balls in the urn?
   (As described so far: you add a number of balls that is equal to the you payoff)



Player 1's urn

Variations on simple reinforcement learning

Exponential response

Instead of drawing balls with uniform probability, you draw balls with probability proportional to their number in the urn

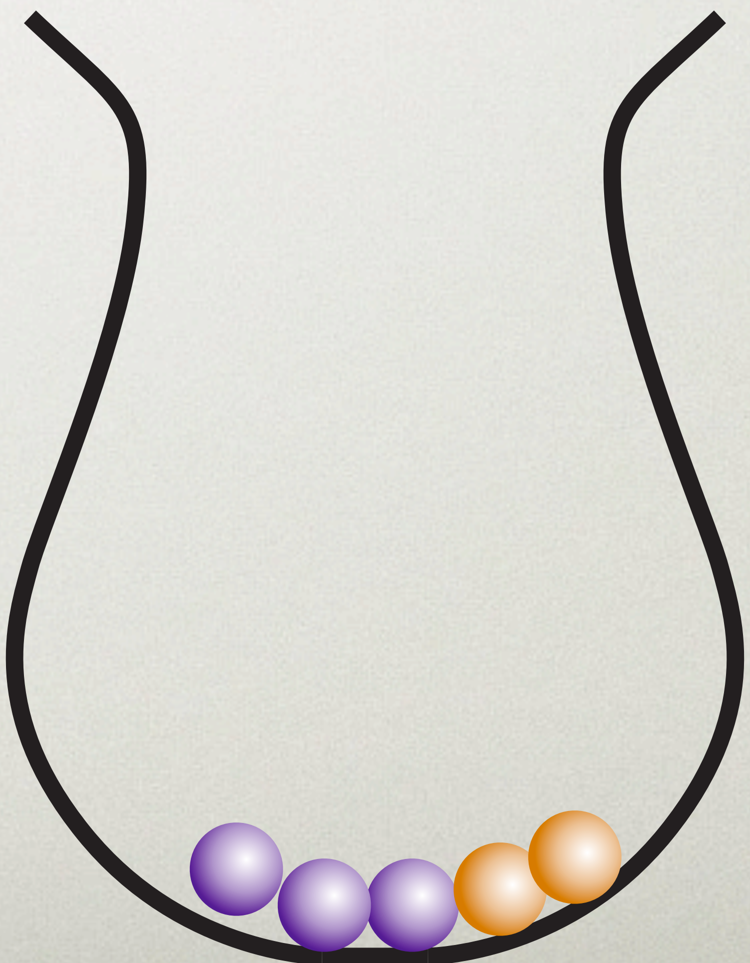
$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

$$\lambda = .001$$

$$w_{purp} = 3$$

$$w_{orange} = 2$$

$$\Pr(purp) \approx .5$$



Variations on simple reinforcement learning

Exponential response

Instead of drawing balls with uniform probability, you draw balls with probability proportional to their number in the urn

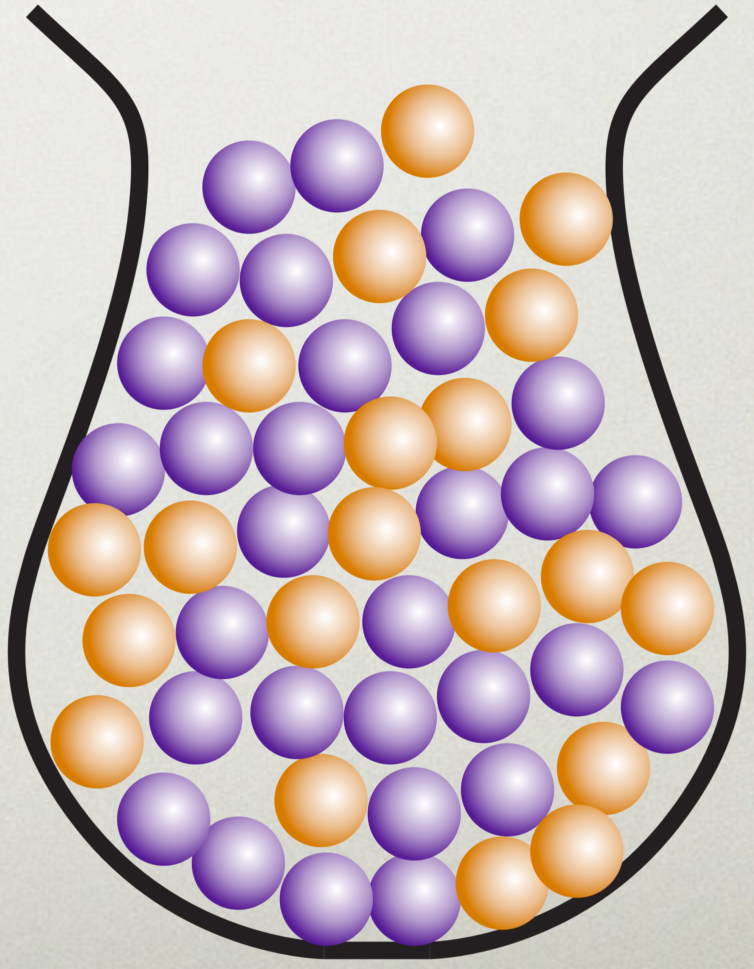
$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

$$\lambda = .001$$

$$w_{purp} = 30$$

$$w_{orange} = 20$$

$$\Pr(purp) \approx .5$$



Variations on simple reinforcement learning

Exponential response

Instead of drawing balls with uniform probability, you draw balls with probability proportional to their number in the urn

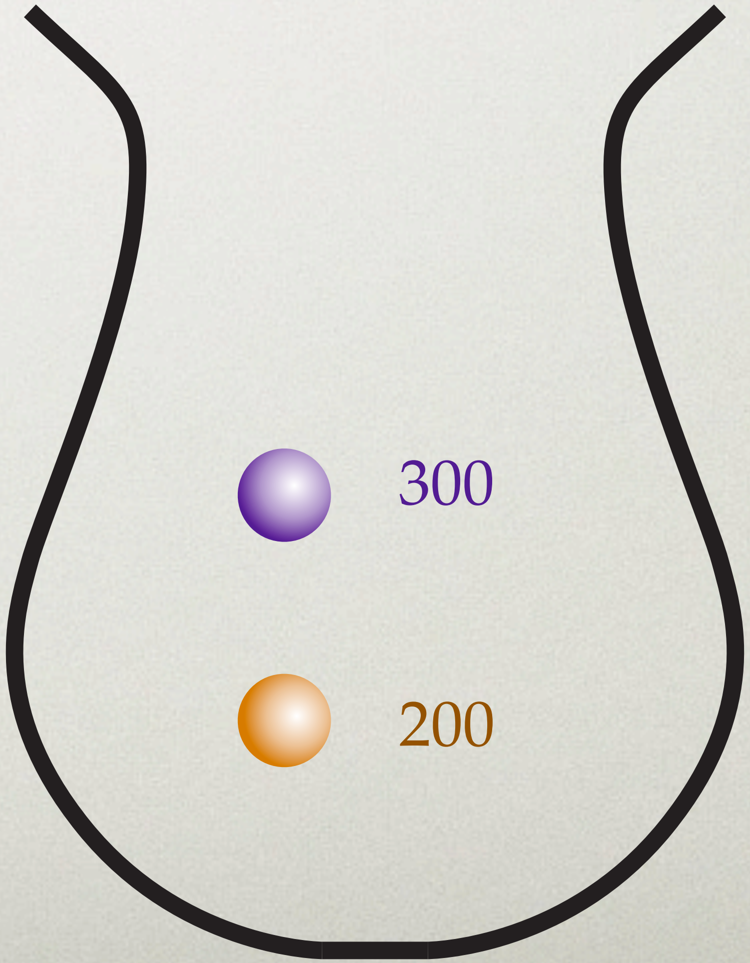
$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

$$\lambda = .001$$

$$w_{purp} = 30$$

$$w_{orange} = 20$$

$$\Pr(purp) \approx .53$$



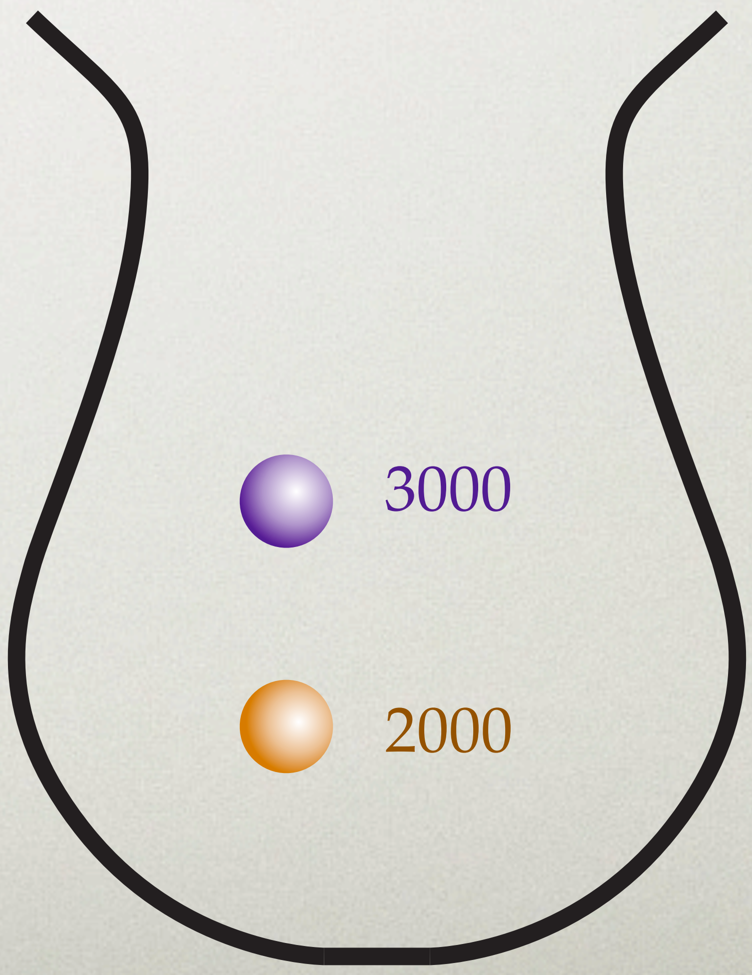
Variations on simple reinforcement learning

Exponential response

Instead of drawing balls with uniform probability, you draw balls with probability proportional to their number in the urn

$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

$$\lambda = .001$$
$$w_{purp} = 30$$
$$w_{orange} = 20$$
$$\Pr(purp) \approx .73$$



Variations on simple reinforcement learning

Exponential response

Instead of drawing balls with uniform probability, you draw balls with probability proportional to their number in the urn

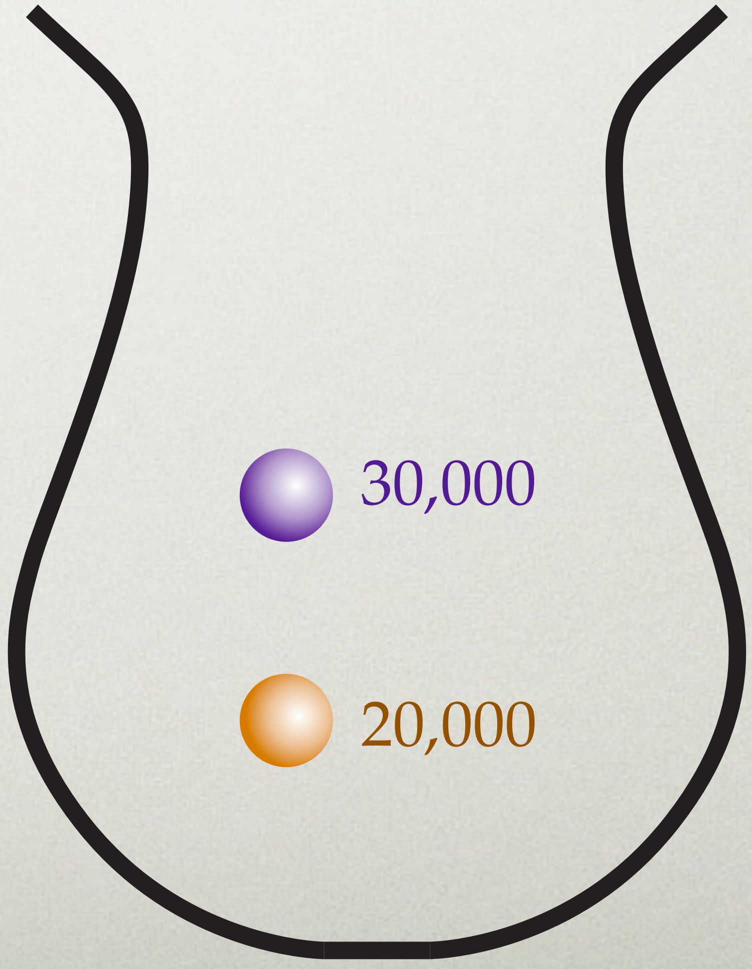
$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

$$\lambda = .001$$

$$w_{purp} = 30$$

$$w_{orange} = 20$$

$$\Pr(purp) \approx .99$$



Variations on simple reinforcement learning

Simulation results (Skyrms):

In two state games with equiprobable states, reinforcement learners that use an exponential response rule (with a wide range of reasonable learning parameters) always converge to a signaling system

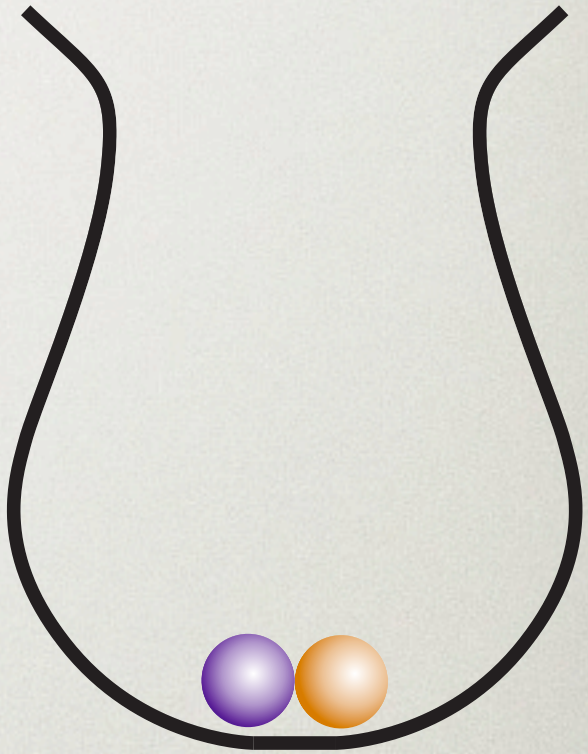
$$\Pr(\text{ball } i) = \frac{e^{\lambda w_i}}{\sum_j e^{\lambda w_j}}$$

There are many possible variations of this process...

1. What number of balls are initially in the urns?
   (As described so far: urns are initially stocked with one ball of each type)

2. How does one convert balls in urns into probabilities for actions?
   (As described so far: every ball has an equal probability of being drawn and you perform the action associated with that ball)

3. How does one translate payoffs of the game into additional balls in the urn?
   (As described so far: you add a number of balls that is equal to the you payoff)



Player 1’s urn

Variations on simple reinforcement learning

Adjustable reference points
(ARP learning)

Bereby-Meyer and Erev (1998) compared experimental data from human learning in games, and concluded that an ARP model fit the data best.

Simulation results (Barrett 2006...):

ARP reinforcement learning with parameters from estimated in the paper above, individuals always learn signaling systems even when the states are biased

Bush-Mosteller Reinforcement

Law of effect:

Actions that are followed by the satisfaction of desires are reinforced so that, when the desire recurs the action that satisfied the desire will be more likely to be performed again.

Herrnstein/Roth-Erev reinforcement learning is one way to model the law of effect, but there are others.

In Bush-Mosteller reinforcement, there is no memory of accumulated reinforcement. Instead the rewards act directly on the probabilities.

$$P_{new}(A) = (1-\alpha)P_{old}(A) + \alpha x$$

Learning parameter

Reward (scaled between 0 and 1)

Another derivation of the replicator dynamic!

The mean field dynamic of Bush-Mosteller reinforcement learning is the continuous-time replicator dynamic!

Borgers and Sarin (1997)

$$P_{new}(A) = (1 - \alpha)P_{old}(A) + \alpha x$$

But, in this case the replicator dynamic is not a good model of the long-term behavior of Bush-Mosteller reinforcement learning.

(The learning doesn't slow down over time, so, unlike Roth-Erev reinforcement learning, the Bush-Mosteller learning doesn't get closer and closer to the deterministic process.)

Bush-Mosteller reinforcement

Simulation results (Skyrms):

For a wide variety of learning parameters, Bush-Mosteller reinforcement learners will almost always learn to signal, even when the states are not equiprobable

The philosophical conundrum that we began with:

> [A] substitution of voice for gesture can only have been made by common consent, something rather difficult to put into effect by men whose crude organs had not yet been exercised; something indeed, even more difficult to conceive of having happened in the first place, for such a unanimous agreement would need to be proposed, which means that speech seems to be absolutely necessary to establish the use of speech.
>
> Rousseau, *Discourse of Inequality*

"I believe that at this point we can conclude that the possibility of learning to signal by simple reinforcement is a reasonably robust finding"

Skyrms (2011, page 101)