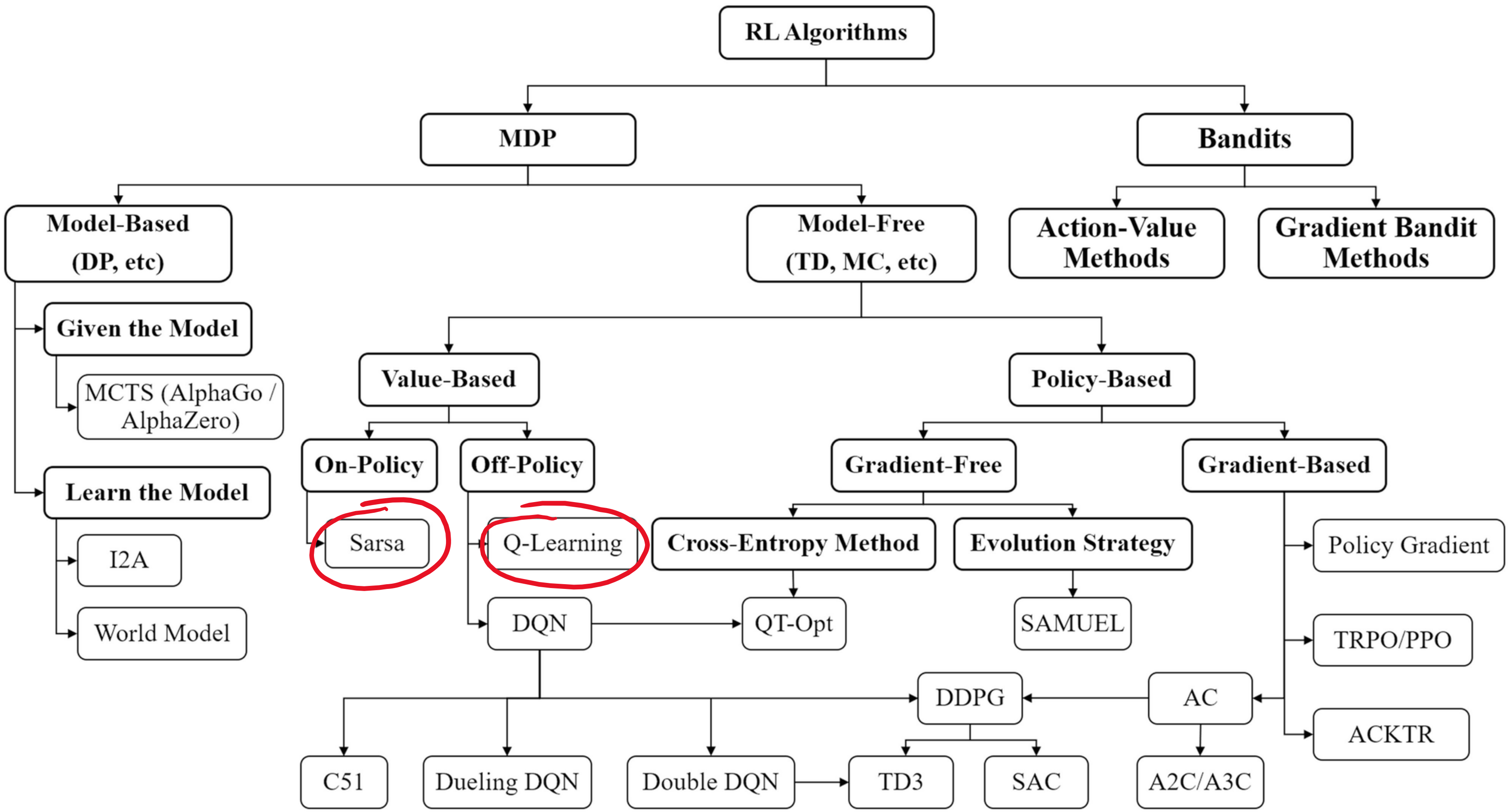
Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 8

Jan, 27, 2021

NOT REQUIRED for 422! Map of reinforcement learning algorithms. Boxes with thick lines denote different categories, others denote specific algorithms



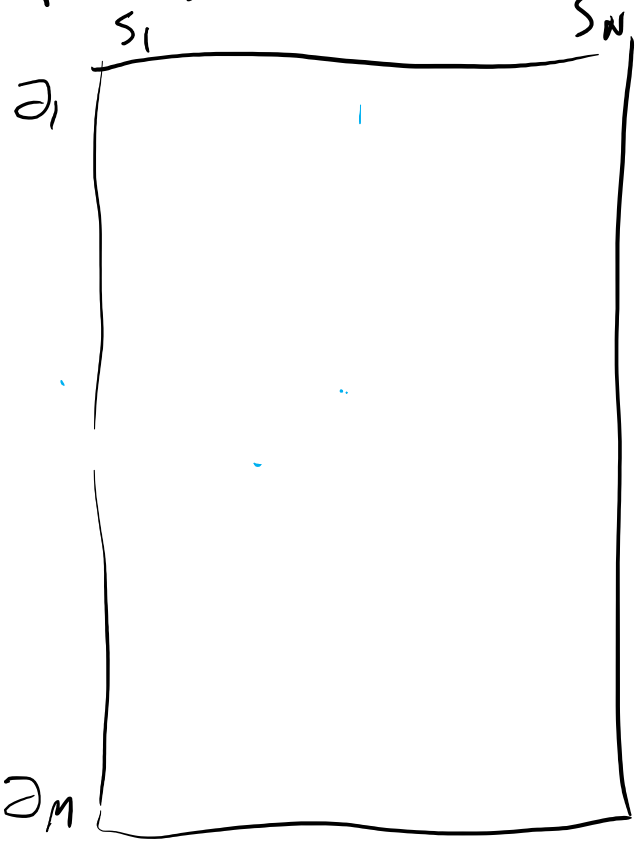
Lecture Overview

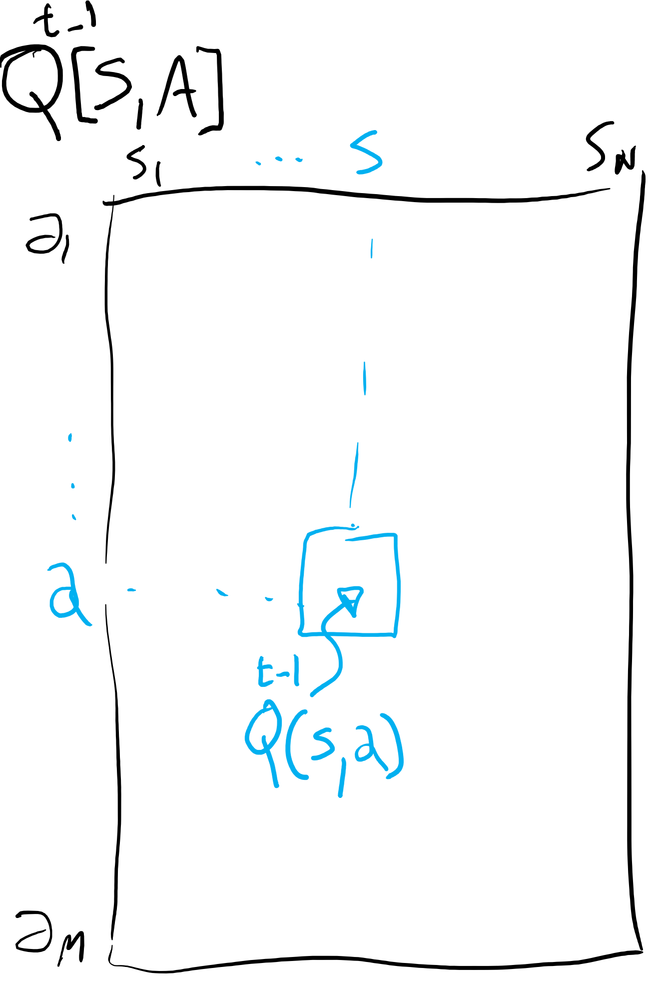
Finish Q-learning

- Algorithm Summary
- Example

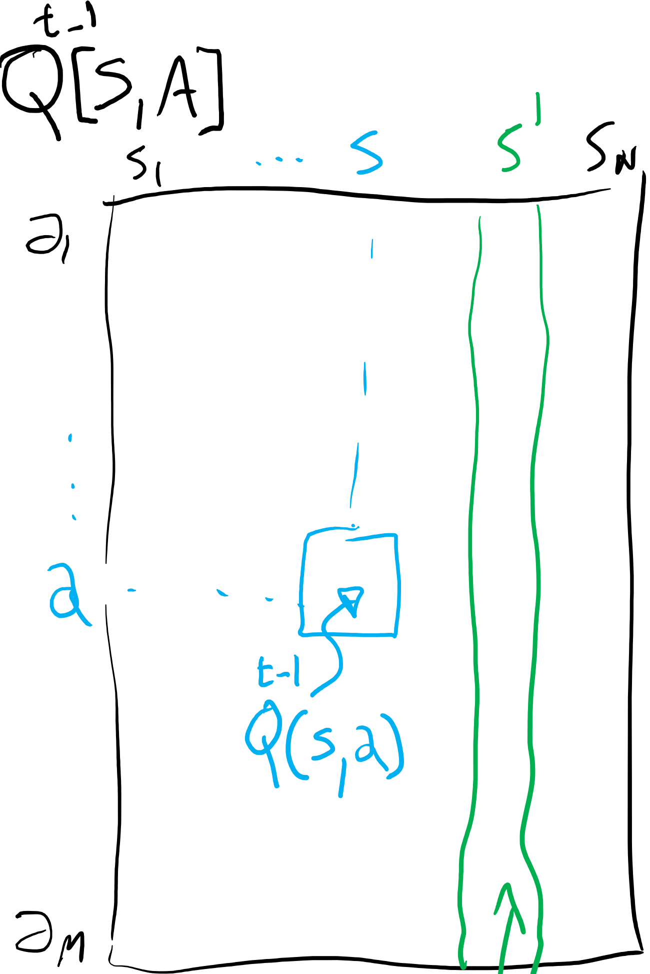
- Exploration vs. Exploitation

$Q^{t-1}[S, A]$





sars'

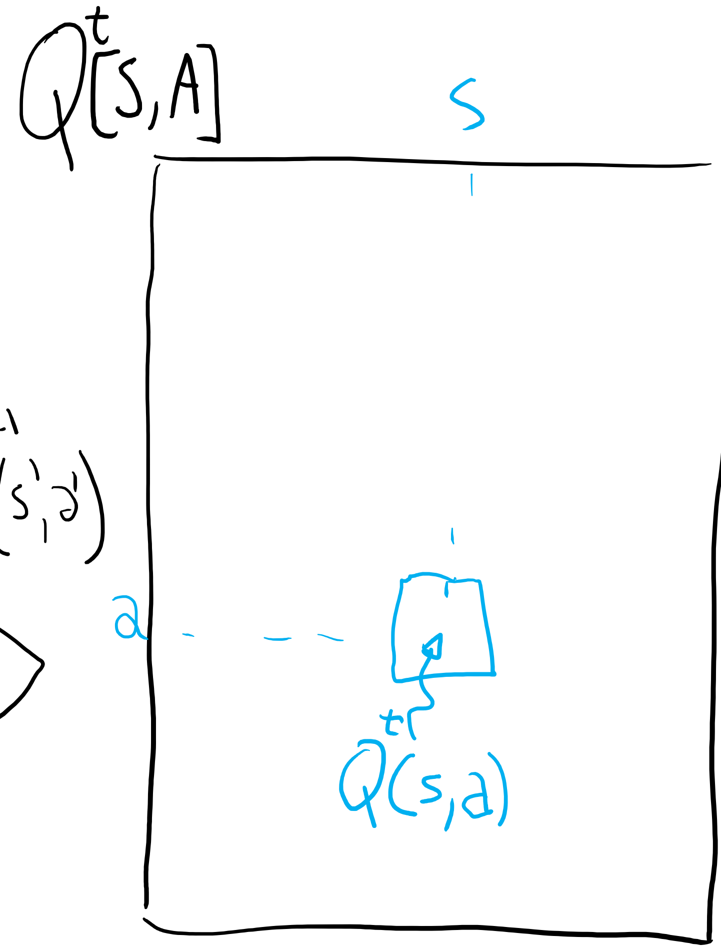


$\max_{a'} Q^{t-1}(s',a')$

sars'

$$Q(s,a) = r + \gamma \max_{a'} Q^{t-1}(s',a')$$

$\Longrightarrow$

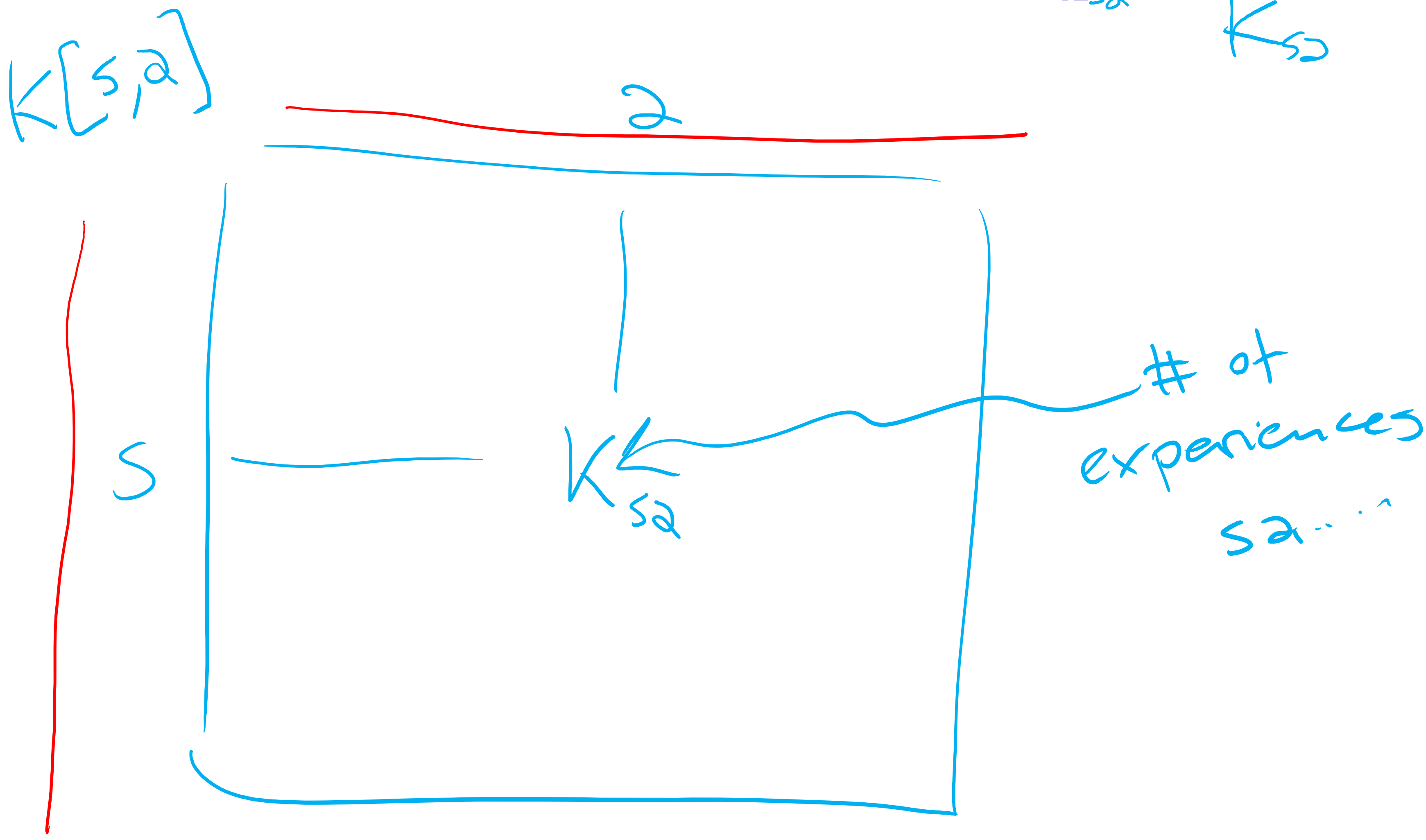


TD

$$A^t = A^{t-1} + \alpha_k \left(r^t - A^{t-1} \right)$$

$$Q^t(s,a) = Q^{t-1}(s,a) + \alpha_k \left(\left(r + \gamma \max_{a'} Q^{t-1}(s',a') \right) - Q^{t-1}(s,a) \right)$$

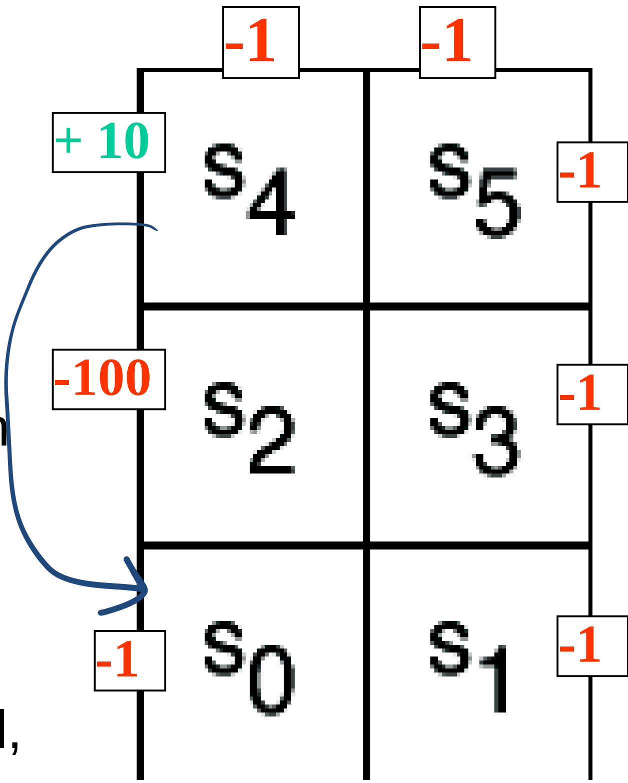
Clarification on the $\alpha_{\kappa_{sa}} = \frac{1}{K_{sa}}$



controller Q-learning(S,A)
inputs:
    S is a set of states
    A is a set of actions
    γ the discount
    α is the step size
internal state:
    real array $Q[S,A]$
    previous state s
    previous action a
begin
    initialize $Q[S,A]$ arbitrarily
    observe current state s
    repeat forever:
        select and carry out an action a
        observe reward r and state s'
        $Q[s,a] \leftarrow Q[s,a] + \alpha\,(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$
        $s \leftarrow s'$;
    end-repeat
end

Example

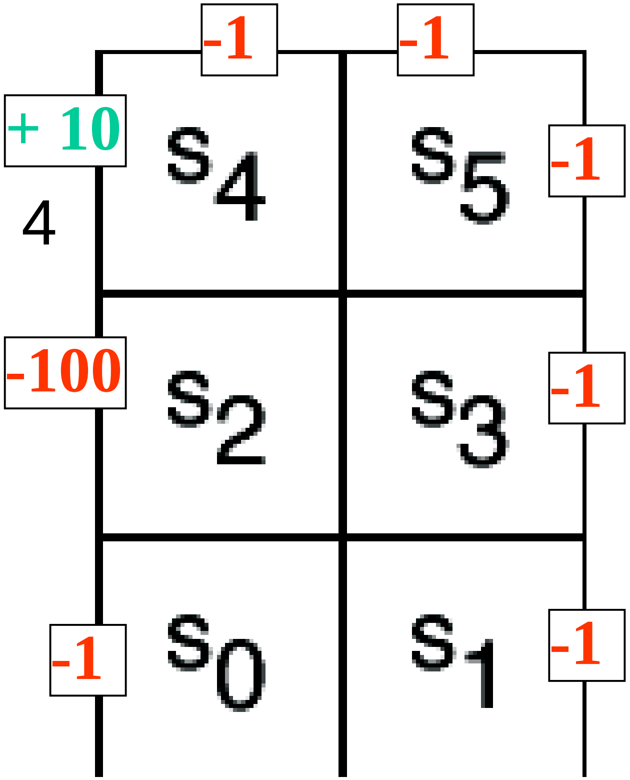
- Six possible states <s_0,..,s_5>
- 4 actions:
  - *UpCareful:* moves one tile up unless there is wall, in which case stays in same tile. Always generates a penalty of -1
  - *Left:* moves one tile left unless there is wall, in which case
    - ✓ stays in same tile if in s_0 or s_2
    - ✓ Is sent to s_0 if in s_4
  - *Right:* moves one tile right unless there is wall, in which case stays in same tile
  - *Up:* 0.8 goes up unless there is a wall, 0.1 like *Left*, 0.1 like *Right*



Reward Model:

- -1 for doing *UpCareful*
- Negative reward when hitting a wall, as marked on the picture

Example



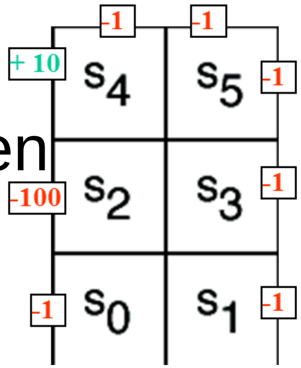
- The agent **knows** about the 6 states and 4 actions
- Can perform an action, fully observe its state and the reward it gets
- **Does not know** how the states are configured, nor what the actions do
  - **no transition model, nor reward model**

Example (variable α_k)

- Suppose that in the simple world described earlier, the agent has the following sequence of experiences

$$<s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0>$$

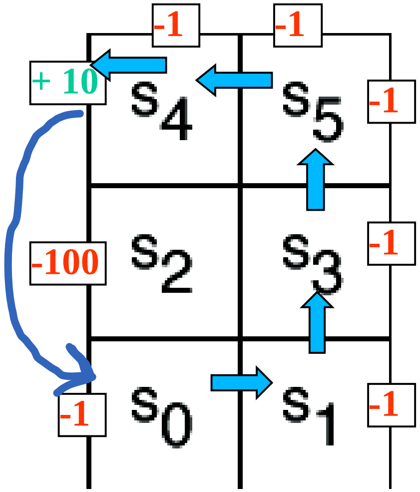
- And repeats it *k* times (not a good behavior for a Q-learning agent, but good for didactic purposes)



- Table shows the first 3 iterations of Q-learning when
  - *Q[s,a]* is initialized to 0 for every *a* and *s*
  - $\alpha_k = 1/k$, $\gamma = 0.9$

Iteration	$Q[s_0, right]$	$Q[s_1, upCare]$	$Q[s_3, upCare]$	$Q[s_5, left]$	$Q[s_4, left]$
1	0	-1	-1	0	10
2	0	-1	-1	4.5	10
3	0	-1	0.35	6.0	10

$\langle s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0 \rangle$



$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$

k=1

Q[s,a]	s_0	s_1	s_2	s_3	s_4	s_5
upCareful	0	0	0	0	0	0
Left	0	0	0	0	0	0
Right	0	0	0	0	0	0
Up	0	0	0	0	0	0

$$Q[s_0, right] \leftarrow Q[s_0, right] + \alpha_k((r + 0.9 \max_{a'} Q[s_1, a']) - Q[s_0, right]);$$

$$Q[s_0, right] \leftarrow$$

$$Q[s_1, upCareful] \leftarrow Q[s_1, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_3, a']) - Q[s_1, upCareful];$$

$$Q[s_1, upCareful] \leftarrow$$

$$Q[s_3, upCareful] \leftarrow Q[s_3, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_5, a']) - Q[s_3, upCareful];$$

$$Q[s_3, upCareful] \leftarrow$$

$$Q[s_5, Left] \leftarrow Q[s_5, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_4, a']) - Q[s_5, Left];$$

$$Q[s_5, Left] \leftarrow 0 + 1(0 + 0.9 * 0 - 0) = 0$$

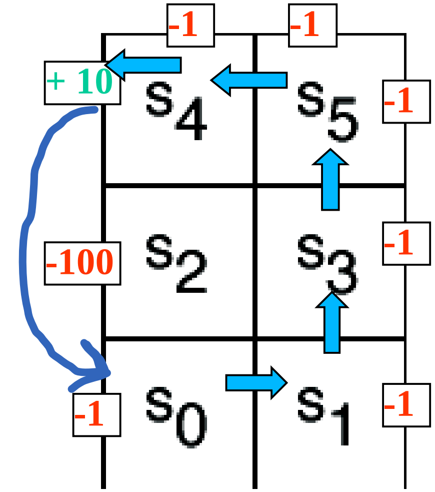
Only immediate rewards are included in the update in this first pass

$$Q[s_4, Left] \leftarrow Q[s_4, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_0, a']) - Q[s_4, Left];$$

$$Q[s_4, Left] \leftarrow 0 + 1(10 + 0.9 * 0 - 0) = 10$$

$$\langle s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0 \rangle$$

$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$



k=2

Q[s,a]	s_0	s_1	s_2	s_3	s_4	s_5
upCareful	**0**	**-1**	0	**-1**	0	0
Left	0	0	0	0	**10**	**0**
Right	0	0	0	0	0	0
Up	0	0	0	0	0	0

$$Q[s_0, right] \leftarrow Q[s_0, right] + \alpha_k((r + 0.9 \max_{a'} Q[s_1, a']) - Q[s_0, right]);$$

$$Q[s_0, right] \leftarrow 0 + 1/2(0 + 0.9 * 0 - 0) = 0$$

$$Q[s_1, upCareful] \leftarrow Q[s_1, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_3, a']) - Q[s_1, upCareful] =$$

$$Q[s_1, upCareful] \leftarrow -1 + 1/2(-1 + 0.9 * 0 + 1) = -1$$

$$Q[s_3, upCareful] \leftarrow Q[s_3, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_5, a']) - Q[s_3, upCareful] =$$

$$Q[s_3, upCareful] \leftarrow -1 + 1/2(-1 + 0.9 * 0 + 1) = -1$$

$$Q[s_5, Left] \leftarrow Q[s_5, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_4, a']) - Q[s_5, Left] =$$

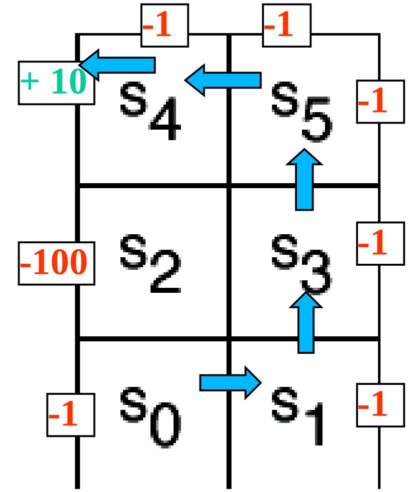
$$Q[s_5, Left] \leftarrow$$

1 step backup from previous positive reward in s4

$$Q[s_4, Left] \leftarrow Q[s_4, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_0, a']) - Q[s_4, Left] =$$

$$Q[s_4, Left] \leftarrow 10 + 1(10 + 0.9 * 0 - 10) = 10$$

$$\langle s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0 \rangle$$



$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$

k=3

Q[s,a]	s_0	s_1	s_2	s_3	s_4	s_5
upCareful	**0**	**-1**	0	**0.35**	0	0
Left	0	0	0	0	**10**	**6**
Right	0	0	0	0	0	0
Up	0	0	0	0	0	0

$$Q[s_0, right] \leftarrow Q[s_0, right] + \alpha_k((r + 0.9 \max_{a'} Q[s_1, a']) - Q[s_0, right]);$$

$$Q[s_0, right] \leftarrow 0 + 1/3(0 + 0.9 * 0 - 0) = 0$$

$$Q[s_1, upCareful] \leftarrow Q[s_1, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_3, a']) - Q[s_1, upCareful] =$$

$$Q[s_1, upCareful] \leftarrow -1 + 1/3(-1 + 0.9 * 0 + 1) = -1$$

$$Q[s_3, upCareful] \leftarrow Q[s_3, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_5, a']) - Q[s_3, upCareful] =$$

$$Q[s_3, upCareful] \leftarrow -1 + 1/3(-1 + 0.9 * 4.5 + 1) = 0.35$$

The effect of the positive reward in s4 is felt two steps earlier at the 3rd iteration

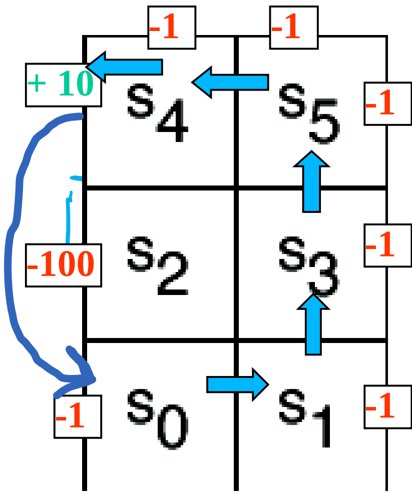
$$Q[s_5, Left] \leftarrow Q[s_5, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_4, a']) - Q[s_5, Left] =$$

$$Q[s_5, Left] \leftarrow 4.5 + 1/3(0 + 0.9 * 10 - 4.5) = 6$$

$$Q[s_4, Left] \leftarrow Q[s_4, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_0, a']) - Q[s_4, Left] =$$

$$Q[s_4, Left] \leftarrow 10 + 1/3(10 + 0.9 * 0 - 10) = 10$$

Example (variable α_k)



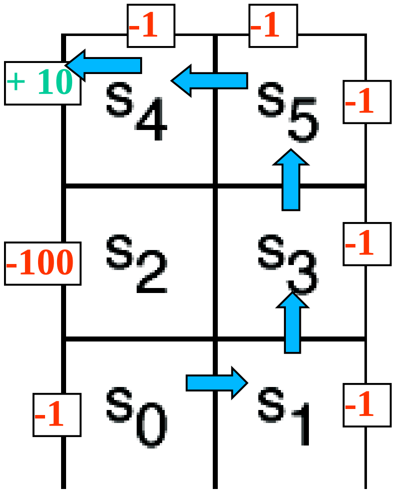
Iteration	$Q[s_0, right]$	$Q[s_1, upCare]$	$Q[s_3, upCare]$	$Q[s_5, left]$	$Q[s_4, left]$
1	0	-1	-1	0	10
2	0	-1	-1	4.5	10
3	0	-1	0.35	6.0	10
4	0	-0.92	1.36	6.75	10
10	0.03	0.51	4	8.1	10
100	2.54	4.12	6.82	9.5	11.34
1000	4.63	5.93	8.46	11.3	13.4
10000	6.08	7.39	9.97	12.83	14.9
100000	7.27	8.58	11.16	14.02	16.08
1000000	8.21	9.52	12.1	14.96	17.02
10000000	8.96	10.27	12.85	15.71	17.77
∞	11.85	13.16	15.74	18.6	20.66

- As the number of iterations increases, the effect of the positive reward achieved by moving left in s_4 trickles further back in the sequence of steps
- Q[s_4,left] starts changing only after the effect of the reward has reached s_0 (i.e. after iteration 10 in the table)

Example (Fixed α=1)

- First iteration same as before, let's look at the second

$$\langle s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0 \rangle$$



$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$

k=2

Q[s,a]	s_0	s_1	s_2	s_3	s_4	s_5
upCareful	**0**	**-1**	0	**-1**	0	0
Left	0	0	0	0	**10**	**0**
Right	0	0	0	0	0	0
Up	0	0	0	0	0	0

$$Q[s_0, right] \leftarrow 0 + 1(0 + 0.9 * 0 - 0) = 0$$

$$Q[s_1, upCareful] \leftarrow -1 + 1(-1 + 0.9 * 0 + 1) = -1$$

$$Q[s_3, upCareful] \leftarrow -1 + 1(-1 + 0.9 * 0 + 1) = -1$$

$$Q[s_5, Left] \leftarrow Q[s_5, Left] + \alpha_k((r + 0.9 \max_{a'} Q[s_4, a']) - Q[s_5, Left] =$$

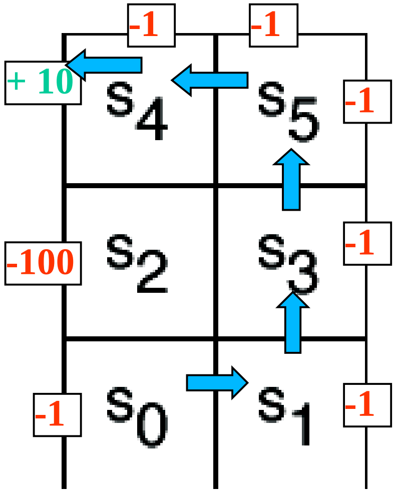
$$Q[s_5, Left] \leftarrow 0 + 1(0 + 0.9 * 10 - 0) = 9$$

New evidence is given much more weight than original estimate

$$Q[s_4, Left] \leftarrow 10 + 1(10 + 0.9 * 0 - 10) = 10$$

$\langle s_0, right, 0, s_1, upCareful, -1, s_3, upCareful, -1, s_5, left, 0, s_4, left, 10, s_0 \rangle$

$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$



k=3

Q[s,a]	s_0	s_1	s_2	s_3	s_4	s_5
upCareful	**0**	**-1**	0	**-1**	0	0
Left	0	0	0	0	**10**	**9**
Right	0	0	0	0	0	0
Up	0	0	0	0	0	0

$$Q[s_0, right] \leftarrow 0 + 1(0 + 0.9 * 0 - 0) = 0$$

$$Q[s_1, upCareful] \leftarrow -1 + 1(-1 + 0.9 * 0 + 1) = -1$$

Same here

$$Q[s_3, upCareful] \leftarrow Q[s_3, upCareful] + \alpha_k((r + 0.9 \max_{a'} Q[s_5, a']) - Q[s_3, upCareful] =$$

$$Q[s_3, upCareful] \leftarrow -1 + 1(-1 + 0.9 * 9 + 1) = 7.1$$

$$Q[s_5, Left] \leftarrow 9 + 1(0 + 0.9 * 10 - 9) = 9$$

$$Q[s_4, Left] \leftarrow 10 + 1(10 + 0.9 * 0 - 10) = 10$$

No change from previous iteration, as all the reward from the step ahead was included there

Comparing fixed *α* and …

Iteration	$Q[s_0, right]$	$Q[s_1, upCare]$	$Q[s_3, upCare]$	$Q[s_5, left]$	$Q[s_4, left]$
1	0	-1	-1	0	10
2	0	-1	-1	9	10
3	0	-1	7.1	9	10
4	0	$\boxed{5.39}$	7.1	9	10
5	$\boxed{4.85}$	5.39	7.1	9	14.37
6	4.85	5.39	7.1	12.93	14.37
10	7.72	8.57	10.64	15.25	16.94
20	10.41	12.22	14.69	17.43	19.37
30	11.55	12.83	15.37	18.35	20.39
40	11.74	13.09	15.66	18.51	20.57
∞	11.85	13.16	15.74	18.6	20.66

variable *α*

Iteration	$Q[s_0, right]$	$Q[s_1, upCare]$	$Q[s_3, upCare]$	$Q[s_5, left]$	$Q[s_4, left]$
1	0	-1	-1	0	10
2	0	-1	-1	4.5	10
3	0	-1	0.35	6.0	10
4	0	-0.92	1.36	6.75	10
10	0.03	$\boxed{0.51}$	4	8.1	10
100	$\boxed{2.54}$	4.12	6.82	9.5	11.34
1000	4.63	5.93	8.46	11.3	13.4
10000	6.08	7.39	9.97	12.83	14.9
100000	7.27	8.58	11.16	14.02	16.08
1000000	8.21	9.52	12.1	14.96	17.02
10000000	8.96	10.27	12.85	15.71	17.77
∞	11.85	13.16	15.74	18.6	20.66

Fixed *α* generates faster update:

all states see some effect of the positive reward from <s4, left> by the 5th iteration

Each update is much larger

Gets very close to final numbers by iteration 40, while with variable *α* still not there by iteration 10^7

However:

Q-learning with fixed *α* is not guaranteed to converge

On the approximation…

$$Q(s,a) = R(s) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

True relation between Q(s.a) and Q(s'a')

$$Q[s,a] \leftarrow Q[s,a] + \alpha((r + \gamma \max_{a'} Q[s',a']) - Q[s,a])$$

Q-learning approximation based on each individual experience <*s, a, r, s'*>

➢ For the approximation to work…..

A. There is positive reward in most states

B. Q-learning tries each action an unbounded number of times

C. The transition model is not sparse

Matrix sparseness

Number of zero elements of a matrix divided by the number of elements. For conditional probabilities the max sparseness is

$$\frac{n^2 - n}{n^2}$$

need at least a 1 in each row

Density is = (1 – sparseness)

The min density for conditional probabilities is

$$\frac{n}{n^2}$$

Note: the action is deterministic!

Why approximations work…

$$Q(s,a) \;=\; R(s) + \gamma \sum_{s'} \underbrace{P(s' \mid s,a)} \max_{a'} Q(s',a')$$

True relation between Q(s.a) and Q(s'a')

$$Q[s,a] \leftarrow Q[s,a] + \alpha(\underbrace{(r + \gamma \max_{a'} Q[s',a'])} - Q[s,a])$$

Q-learning approximation based on each individual experience <*s, a, s'*>

- Way to get around the **missing transition model and reward model**
- Aren't we in danger of using data coming from unlikely transition to make incorrect adjustments?
- No, as long as Q-learning tries each action an unbounded number of times
  - Frequency of updates reflects transition model, *P(s'|a,s)*

Lecture Overview

Finish Q-learning

- Algorithm
- Example

- Exploration vs. Exploitation

What Does Q-Learning learn

- Does Q-learning gives the agent an optimal policy?

Q values

	s_0	s_1	…	s_k
a_0	$Q[s_0,a_0]$	$Q[s_1,a_0]$	….	$Q[s_k,a_0]$
a_1	$Q[s_0,a_1]$	$Q[s_1,a_1]$	…	$Q[s_k,a_1]$
…	…	…	….	…
a_n	$Q[s_0,a_n]$	$Q[s_1,a_n]$	….	$Q[s_k,a_n]$

what to do in S_2

$\arg\max_a Q[s_1, a]$

Exploration vs. Exploitation

- Q-learning does not explicitly tell the agent what to do

- just computes a Q-function Q[s,a] that allows the agent to see, for every state, which is the action with the highest expected reward

- Given a Q-function the agent can :

  - **Exploit** the knowledge accumulated so far, and chose the action that maximizes Q[s,a] in a given state (***greedy behavior***)

  - **Explore** new actions, hoping to improve its estimate of the optimal Q-function, i.e. *do not chose* the action suggested by the current Q[s,a]

Exploration vs. Exploitation

- When to explore and when the exploit?
  1. Never exploring may lead to being stuck in a suboptimal course of actions
  2. Exploring too much is a waste of the knowledge accumulated via experience

A. Only (1) is true

B. Only (2) is true

C. Both are true

D. Both are false

Exploration vs. Exploitation

- When to explore and when the exploit?
  - Never exploring may lead to being stuck in a suboptimal course of actions
  - Exploring too much is a waste of the knowledge accumulated via experience
- Must find the right compromise

Exploration Strategies

- Hard to come up with an optimal exploration policy (problem is widely studied in *statistical decision theory*)

- But intuitively, any such strategy should be *greedy in the limit of infinite exploration* (**GLIE**), i.e.
  - **Choose the predicted best action in the limit**
  - **Try each action an unbounded number of times**
- We will look at two exploration strategies
  - ε-greedy
  - soft-max

ε-greedy

- Choose a **random action with probability ε** and choose **best action with probability 1- ε**

$P(\text{random action}) = \varepsilon$

$P(\text{best action}) = 1 - \varepsilon$

- First GLIE condition (try every action an unbounded number of times) is satisfied via the ε random selection

- What about second condition?
  - Select predicted best action in the limit.

- reduce ε overtime!

Soft-Max

- Takes into account improvement in estimates of expected reward function Q[s,a]
  - Choose action ***a*** in state ***s*** with a probability proportional to current estimate of **Q[s,a]**

$$\frac{e^{Q[s,a]}}{\sum_a e^{Q[s,a]}}$$

Assume only 3 actions

$Q[s_i, a]$

	s_i
a_1	2 $
a_2	3 $
a_3	1 $

prob of selecting action a

$p(a_1)$ $\frac{e^2 \approx 7.3}{e^1+e^2+e^3}$

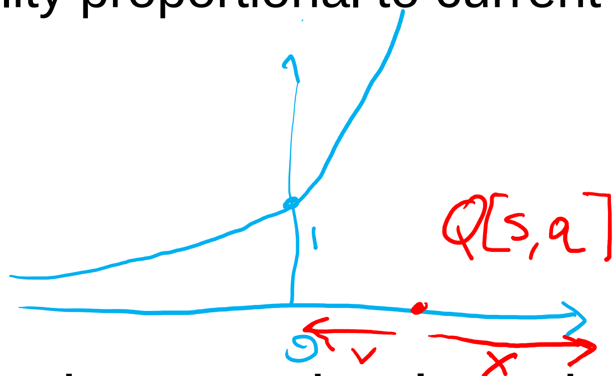
$p(a_2)$ $\frac{e^3 \approx 20}{e^1+e^2+e^3}$

$p(a_3)$ $\frac{e^1 \approx 2.7}{e^1+e^2+e^3}$

(τ controlled) Soft-Max

- Takes into account improvement in estimates of expected reward function Q[s,a]
  - Choose action ***a*** in state ***s*** with a probability proportional to current estimate of **Q[s,a]**

$$\frac{e^{Q[s,a]/\tau}}{\sum_a e^{Q[s,a]/\tau}}$$



- τ (tau) in the formula above influences how randomly actions should be chosen
  - if τ is high, the exponentials approach 1, the fraction approaches 1/(number of actions), and each action has approximately the same probability of being chosen (exploration or exploitation?)
  - as $\tau \rightarrow 0$, the exponential with the highest Q[s,a] dominates, and the current best action is always chosen (exploration or exploitation?)

if Q[s,a] close to 0 each action selected with prob $\frac{1}{\#\text{of actions}}$ UNIFORM DISTRIB.

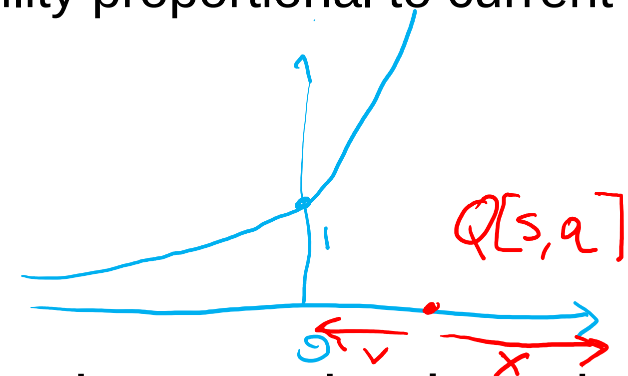
Soft-Max

- Takes into account improvement in estimates of expected reward function Q[s,a]
  - Choose action ***a*** in state ***s*** with a probability proportional to current estimate of **Q[s,a]**

$$\frac{e^{Q[s,a]}}{\sum_a e^{Q[s,a]}}$$

or controlled by τ parameter

$$\frac{e^{Q[s,a]/\tau}}{\sum_a e^{Q[s,a]/\tau}}$$



- τ (tau) in the formula above influences how randomly actions should be chosen
  - if τ is high, the exponentials approach 1, the fraction approaches 1/(number of actions), and each action has approximately the same probability of being chosen (exploration or exploitation?)
  - as $\tau \rightarrow 0$, the exponential with the highest Q[s,a] dominates, and the current best action is always chosen (exploration or exploitation?)

(τ controlled) Soft-Max example

Assume only 3 actions

$Q[s_i,a]$	s_i	$Q[s_i,a]/\tau$ $\tau=100$	$\tau=.5$
a_1	2$	.02	4
a_2	3$	.03	6
a_3	1$	.01	2

prob of selecting action a $\quad \dfrac{e^{Q[s,a]}}{\sum_a e^{Q[s,a]}}$

$p(a_1) = \dfrac{e^2}{e^1+e^2+e^3}$ $\quad p(a_2) = \dfrac{e^3}{e^1+e^2+e^3}$ $\quad p(a_3) = \dfrac{e^1}{e^1+e^2+e^3}$

$\tau = 100$ $\quad \dfrac{e^{Q[s,a]/\tau}}{\sum_a e^{Q[s,a]/\tau}}$

$\dfrac{e^{.02} \approx 1}{e^{.01}+e^{.02}+e^{.03}}$ $\quad \dfrac{e^{.03} \approx 1}{\text{same}}$ $\quad \dfrac{e^{.01} \approx 1}{}$

$\tau = .5$ $\quad \dfrac{e^{Q[s,a]/\tau}}{\sum_a e^{Q[s,a]/\tau}}$

$\dfrac{e^4 \approx 54}{e^2+e^4+e^6}$ $\quad \dfrac{e^6 \approx 403}{}$ $\quad \dfrac{e^2 \approx 7.}{}$

Learning Goals for today's class

➢**You can:**

- Explain, trace and implement Q-learning
- Describe and compare techniques to combine exploration with exploitation

TODO for Fri

- **Carefully read : A Markov decision process approach to multi-category patient scheduling in a diagnostic facility,** Artificial Intelligence in Medicine Journal, **2011**

- **Follow instructions on course WebPage** <Readings>

- Keep working on assignment-1 (due next Wed)