HMM MDP RL

Mike Stilman (and a very frustrated robot)
robot@cmu.edu

The Robotics Institute
Carnegie Mellon University

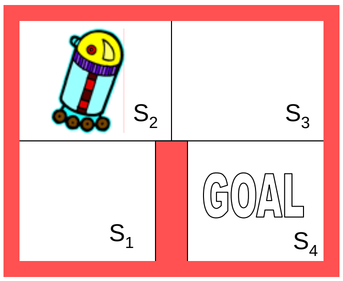
Machine Learning 10-701 Course Review, Fall 2005

Dec 14th, 2005

THE WORLD

A robot is trying to get to the goal

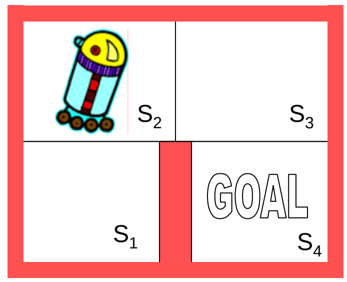
- Robot = R
- Goal = S_4
- 4 States



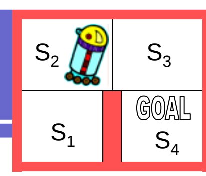
Markov?

Can we represent this as a Markov System?

- Why?

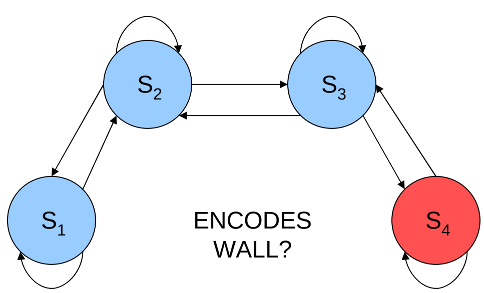


Ok - Markov



Markov Model:

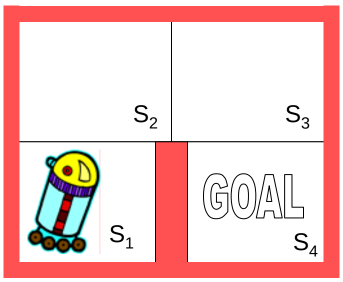
- Legitimate Transitions



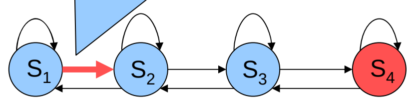
Elementary Markov Learning

- Lets Watch the Robot

It Does Sequence:
$S_1\ S_2\ S_1\ S_1\ S_2\ S_2\ S_1$
$S_2\ S_3\ S_2\ S_2\ S_2\ S_2\ S_3\ S_4$
$S_4\ S_3\ S_3\ S_3\ S_3$



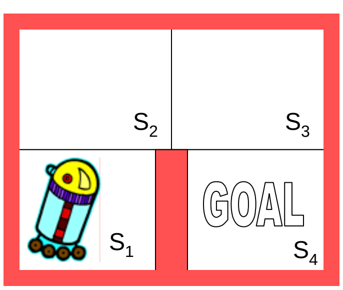
$P(S_{t+1} = S_2 \mid S_t = S_1) = ?$



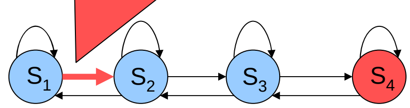
Elementary Markov Learning

- Lets Watch the Robot

It Does Sequence:
$S_1\ S_2\ S_1\ S_1\ S_2\ S_2\ S_1$
$S_2\ S_3\ S_2\ S_2\ S_2\ S_2\ S_3\ S_4$
$S_4\ S_3\ S_3\ S_3\ S_3$



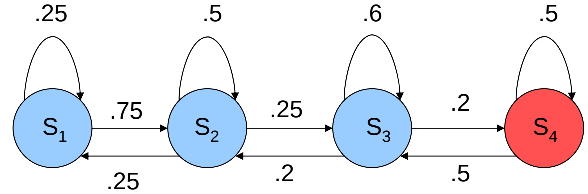
$$P(S_{t+1} = S_2 \mid S_t = S_1) = \frac{\#\, S_1 \rightarrow S_2}{\#\, S_1} = .75$$



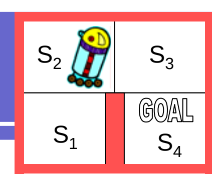
Elementary Markov Learning

- Lets Watch the Robot

It Does Sequence:
S_1 S_2 S_1 S_1 S_2 S_2 S_1
S_2 S_3 S_2 S_2 S_2 S_2 S_3 S_4
S_4 S_3 S_3 S_3 S_3



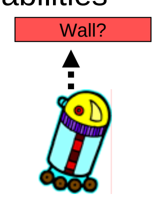
HMM: Observations

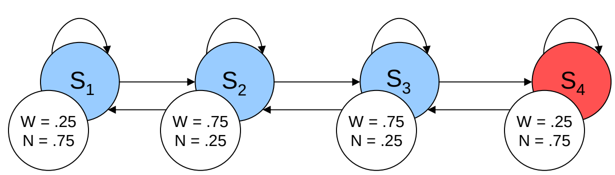


OK, we learned transition probabilities
Lets expand our model:

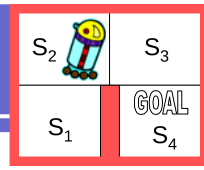
- Robots have sensors!
- Our robot can only look up
- ¼ of the time it's wrong

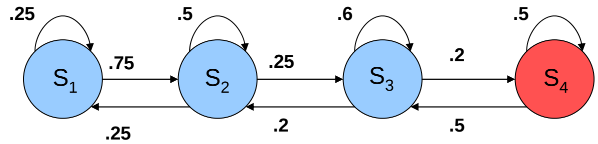
(hence frustrated)





HMM: Transitions



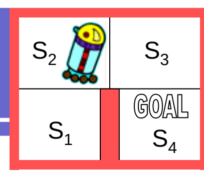


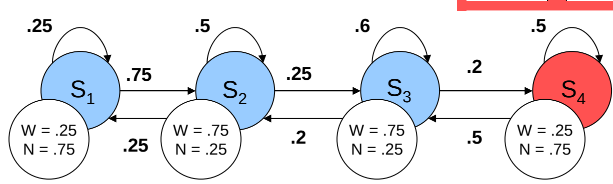
A

	S1	S2	S3	S4
S1	.25	.75	0	0
S2	.25	.5	.25	0
S3	0	.2	.6	.2
S4	0	0	.5	.5

Transition Matrix

HMM: Observations





A

	S1	S2	S3	S4
S1	.25	.75	0	0
S2	.25	.5	.25	0
S3	0	.2	.6	.2
S4	0	0	.5	.5

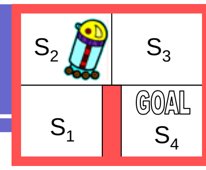
Transition Matrix

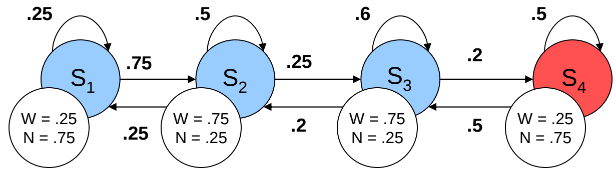
B

	W	N
S1	.25	.75
S2	.75	.25
S3	.75	.25
S4	.25	.75

Observation Matrix

HMM: λ Complete Model





A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

Transition Matrix

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

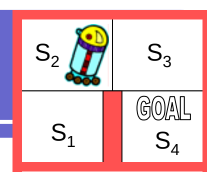
Observation Matrix

π

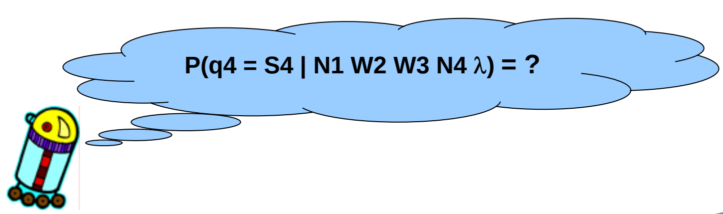
	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

Initial

HMM: Where am I?



- The robot has observed:
  - No Wall
  - Wall
  - Wall
  - No Wall
- What is the probability that the robot is now at the goal?



HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \; q_t = S_i \mid \lambda)$

- Initialization: $\alpha_1(i) = P(O_1 \; q_t = S_i \mid \lambda) = \pi(S_i)\, P(O_1|S_i)$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

	N	W	W	N
S_1	.5 × .75 = .375			
S_2	.3 × .25 = .075			
S_3	.1 × .25 = .025			
S_4	.1 × .75 = .075			

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \; q_t = S_i \mid \lambda)$

- Induction: $\alpha_{t+1}(i) = [\, \Sigma_i \, \alpha_t(i)\, A_{ij} \,]\, B_j\,(O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

	N	W	W	N
S_1	.375 →	(.375 × .25 + .075 × .25) × .25		
S_2	.075 ↗			
S_3	.025			
S_4	.075			

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t\ \ q_t = S_i \mid \lambda)$

- Induction: $\alpha_{t+1}(i) = [\ \sum_i \alpha_t(i)\ A_{ij}\ ]\ B_j\ (O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375	.1125		
S_2	.075	(.375 × .75 + .075 × .5 .025 × .2) × .75		
S_3	.025			
S_4	.075			

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t\ \ q_t = S_i \mid \lambda)$

- Induction: $\alpha_{t+1}(i) = [\ \sum_i \alpha_t(i)\ A_{ij}\ ]\ B_j\ (O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375	.1125		
S_2	.075	.2428		
S_3	.025	(.075 × .25 + .025 × .6 .075 × .5) × .75		
S_4	.075			

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \;\; q_t = S_i \mid \lambda)$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

- Induction: $\alpha_{t+1}(i) = [\, \Sigma_i \, \alpha_t(i) \, A_{ij} \,] \, B_j \, (O_{t+1})$

	N	W	W	N
S_1	.375	.1125		
S_2	.075	.2428		
S_3	.025	.0534		
S_4	.075	.0106		

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \;\; q_t = S_i \mid \lambda)$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

- Induction: $\alpha_{t+1}(i) = [\, \Sigma_i \, \alpha_t(i) \, A_{ij} \,] \, B_j \, (O_{t+1})$

	N	W	W	N
S_1	.375	.1125 →	(.1125 × .25 + .2482 × .25) × .25	
S_2	.075	.2428 ↗		
S_3	.025	.0534		
S_4	.075	.0106		

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \; q_t = S_i \mid \lambda)$

- Induction: $\alpha_{t+1}(i) = [\, \sum_i \alpha_t(i) A_{ij} \,] B_j (O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375	.1125	.0222	
S_2	.075	.2428	.1623	
S_3	.025	.0534	.0735	
S_4	.075	.0106	.0040	

HMM: Where am I?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

FORWARD ALGORITHM

$\alpha_t(i) = P(O_1 \ldots O_t \; q_t = S_i \mid \lambda)$

- Induction: $\alpha_{t+1}(i) = [\, \sum_i \alpha_t(i) A_{ij} \,] B_j (O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375	.1125	.0222	**.0346**
S_2	.075	.2428	.1623	**.0281**
S_3	.025	.0534	.0735	**.0217**
S_4	.075	.0106	.0040	**.0125**

HMM: Where am I?

- We found **$P(N_1\ W_2\ W_3\ N_4\ q_4 = S_4 \mid \lambda) = .0125$**
- Is this the probability of being in S_4 given that we have seen N_1, W_2, W_3 and N_4?

HMM: Where am I?

- We found $P(N_1\ W_2\ W_3\ N_4\ q_4 = S_4 \mid \lambda) = .0125$
- Is this the probability of being in S_4 given that we have seen N_1, W_2, W_3 and N_4?

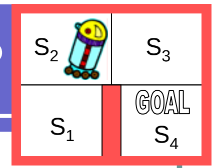
NO

$$\mathbf{P(q_4 = S_4 \mid N_1\ W_2\ W_3\ N_4\ \lambda)} = \frac{P(N_1\ W_2\ W_3\ N_4\ q_4 = S_4 \mid \lambda)}{P(N_1\ W_2\ W_3\ N_4)}$$

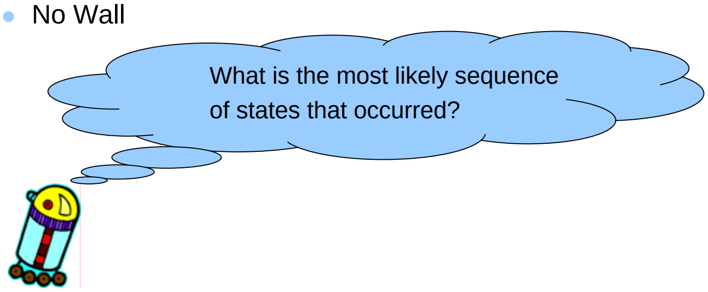
$= .0125 / (.0346+.0281+.0217+.0125)$
$= .129$

13% in State 4
36% in State 1

HMM: How Did I Get There?



- The robot has observed:
  - No Wall
  - Wall
  - Wall
  - No Wall



HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \dots q_t} P(q_1 \dots q_t = i, O_1 \dots O_t \mid \lambda)$$

- Initialization: $\alpha_1(i) = P(O_1\ q_t = S_i \mid \lambda) = \pi(S_i)\, P(O_1|S_i)$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	$.5 \times .75 = .375$			
S_2	$.3 \times .25 = .075$			
S_3	$.1 \times .25 = .025$			
S_4	$.1 \times .75 = .075$			

HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}]\, B_j(O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375 →	max(**.375** × **.25** .075 × .25) ×.25		
S_2	.075			
S_3	.025			
S_4	.075			

HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}]\, B_j(O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

	N	W	W	N
S_1	.375 →	.024		
S_2	.075	max (**.375** × **.75** .075 × .5 .025 × .2) × .75		
S_3	.025			
S_4	.075			

HMM: How Did I Get There?

VITERBI ALGORITHM

$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}] B_j(O_{t+1})$

	N	W	W	N
S_1	.375 →	.024		
S_2	.075 ↘	.211		
S_3	.025	max (.075 × .25 .025 × .6 **.075 × .5**) × .75		
S_4	.075 ↗			

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: How Did I Get There?

VITERBI ALGORITHM

$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}] B_j(O_{t+1})$

	N	W	W	N
S_1	.375 →	.024		
S_2	.075 ↘	.211		
S_3	.025 ↗	.028		
S_4	.075 →	.009		

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}] B_j(O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

	N	W	W	N
S_1	.375	.024	.013	
S_2	.075	.211	.079	
S_3	.025	.028	.040	
S_4	.075	.009	.001	

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$$

- Induction: $\delta_t(i) = [\max_i \delta_t(i) A_{ij}] B_j(O_{t+1})$

B

	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

	N	W	W	N
S_1	.375	.024	.013	.015
S_2	.075	.211	.079	.005
S_3	.025	.028	.040	.006
S_4	.075	.009	.001	.006

π

	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

HMM: How Did I Get There?

A

	S_1	S_2	S_3	S_4
S_1	.25	.75	0	0
S_2	.25	.5	.25	0
S_3	0	.2	.6	.2
S_4	0	0	.5	.5

VITERBI ALGORITHM

$$\delta_t(i) = \max_{q_1 \ldots q_t} P(q_1 \ldots q_t = i, O_1 \ldots O_t \mid \lambda)$$

- Now, Just Trace it Back!

B

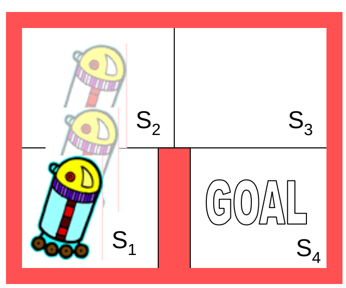
	W	N
S_1	.25	.75
S_2	.75	.25
S_3	.75	.25
S_4	.25	.75

	N	W	W	N
S_1	.375	.024	.013	**.015**
S_2	.075	.211	.079	.005
S_3	.025	.028	.040	.006
S_4	.075	.009	.001	.006

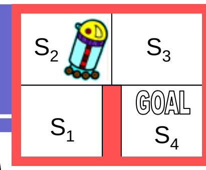
π

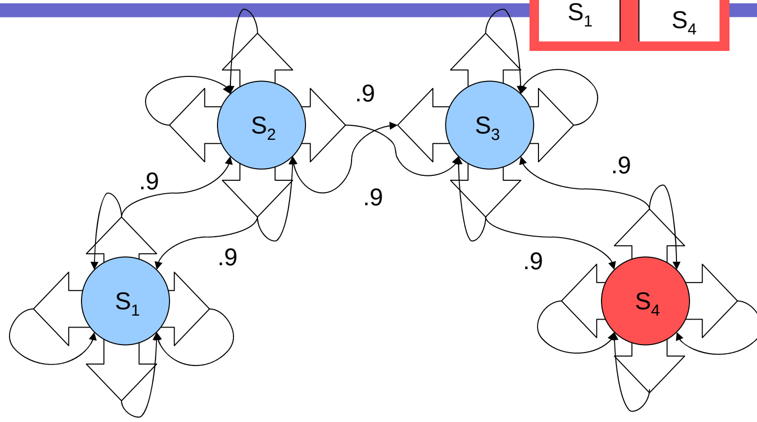
	π
S_1	.5
S_2	.3
S_3	.1
S_4	.1

Robot's POLICY is…not great



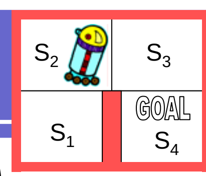
MDPs to the Rescue!

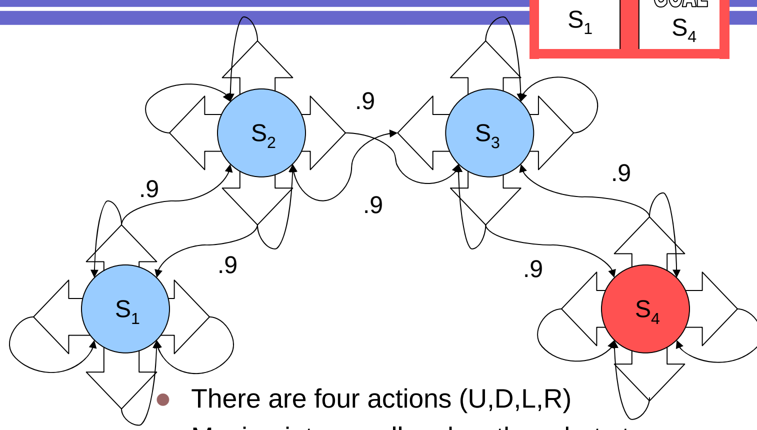




Assume the world is fully observable

MDPs to the Rescue!





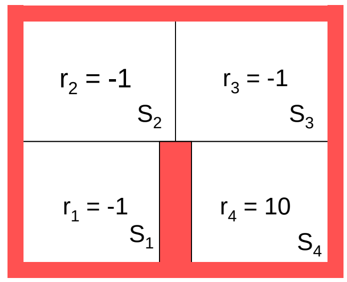
- There are four actions (U,D,L,R)
- Moving into a wall makes the robot stay
- Moving to free space works 90% of the time. (Otherwise robot stays)

Lets be Optimal

Find the optimal policy when:

- Rewards for leaving a state (r)
- Discount factor $\gamma = .9$

- **What is $J^*(S_i)$?**



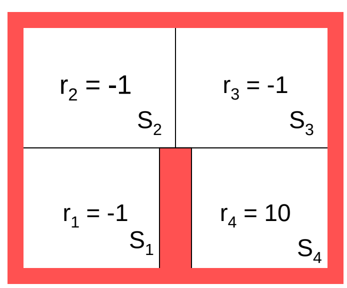
Lets be Optimal

Find the optimal policy when:

- Rewards for leaving a state (r)
- Discount factor $\gamma = .9$

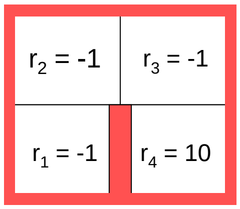
- **What is $J^*(S_i)$?**

$$J^*(S_i) = r_i + \gamma \max_a [\textstyle\sum_j P(j \mid i,a) J^*(j)]$$



Value Iteration $J^*(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J^*(j)]$

Iterative Local Optimization: Implements Dynamic Programming



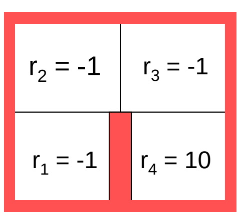
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$				
$J_2(S_i)$				
$J_3(S_i)$				
$J_4(S_i)$				

Initialize to 0 – or something meaningful from policy

How do we update?

Value Iteration $J^*(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J^*(j)]$

Iterative Local Optimization: Implements Dynamic Programming



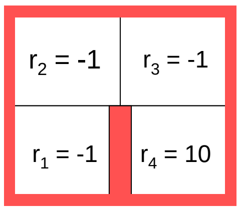
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1			
$J_2(S_i)$				
$J_3(S_i)$				
$J_4(S_i)$				

$$J_{t+1}(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J_t(j)]$$

$$J_1(S_1) = -1 + .9 \max_a [1 \times 0, 1 \times 0, 1 \times 0, .9 \times 0 + .1 \times 0]$$

Value Iteration $J^*(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J^*(j)]$

Iterative Local Optimization: Implements Dynamic Programming



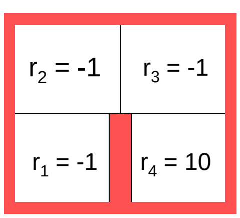
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$				
$J_3(S_i)$				
$J_4(S_i)$				

$$J_{t+1}(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J_t(j)]$$

$J_1(S_1) = -1 + .9 \max_a [1 \times 0, 1 \times 0, 1 \times 0, .9 \times 0 + .1 \times 0]$
…
$J_1(S_4) = 10 + .9 \max_a [1 \times 0, 1 \times 0, 1 \times 0, .9 \times 0 + .1 \times 0]$

Value Iteration $J^*(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J^*(j)]$

Iterative Local Optimization: Implements Dynamic Programming



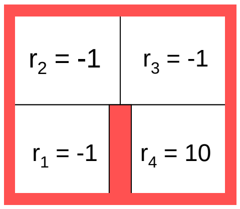
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$	-1.9	-1.9	7.0	19
$J_3(S_i)$				
$J_4(S_i)$				

$$J_{t+1}(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J_t(j)]$$

$J_2(S_1) = -1 + .9 \max_a [1 \times -1, 1 \times -1, 1 \times -1, .9 \times -1 + .1 \times -1]$
…
$J_2(S_3) = -1 + .9 \max_a [1 \times -1, 1 \times -1, 1 \times -1, .9 \times 10 + .1 \times -1]$
$J_2(S_4) = 10 + .9 \max_a [1 \times 10, 1 \times 10, 1 \times 10, .9 \times -1 + .1 \times 10]$

Value Iteration $J^*(S_i) = r_i + \gamma \ \max_a [\ \Sigma_j P(j \mid i,a) J^*(j)\ ]$

Iterative Local Optimization: Implements Dynamic Programming



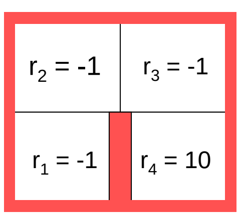
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$	-1.9	-1.9	7.0	19
$J_3(S_i)$	-2.7	4.5	15.0	27.1
$J_4(S_i)$				

$J_{t+1}(S_i) = r_i + \gamma \ \max_a [\ \Sigma_j P(j \mid i,a) J_t(j)\ ]$

Repeat until convergence…

Value Iteration $J^*(S_i) = r_i + \gamma \ \max_a [\ \Sigma_j P(j \mid i,a) J^*(j)\ ]$

Iterative Local Optimization: Implements Dynamic Programming



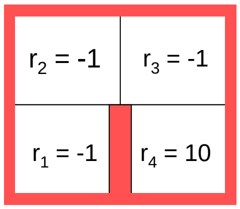
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$	-1.9	-1.9	7.0	19
$J_3(S_i)$	-2.7	4.5	15.0	27.1
$J_4(S_i)$	2.4	11.6	22.3	34.39

$J_{t+1}(S_i) = r_i + \gamma \ \max_a [\ \Sigma_j P(j \mid i,a) J_t(j)\ ]$

Repeat until convergence…

Value Iteration $J^*(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J^*(j)]$

Iterative Local Optimization: Implements Dynamic Programming



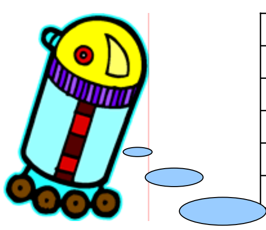
	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$	-1.9	-1.9	7.0	19
$J_3(S_i)$	-2.7	4.5	15.0	27.1
$J_4(S_i)$	2.4	11.6	22.3	34.39

$J_{t+1}(S_i) = r_i + \gamma \max_a [\sum_j P(j \mid i,a) J_t(j)]$

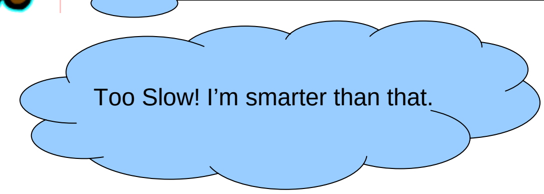
To get policy, simply do **argmax a!**

$\pi_{t+1}(S_i) = \text{argmax}_a [\sum j \, P(j \mid i,a) J_t(j)]$

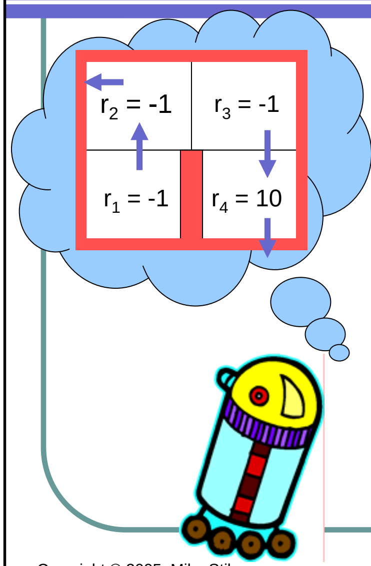
Downside…



	S_1	S_2	S_3	S_4
$J_0(S_i)$	0	0	0	0
$J_1(S_i)$	-1	-1	-1	10
$J_2(S_i)$	-1.9	-1.9	7.0	19
$J_3(S_i)$	-2.7	4.5	15.0	27.1
$J_4(S_i)$	2.4	11.6	22.3	34.39



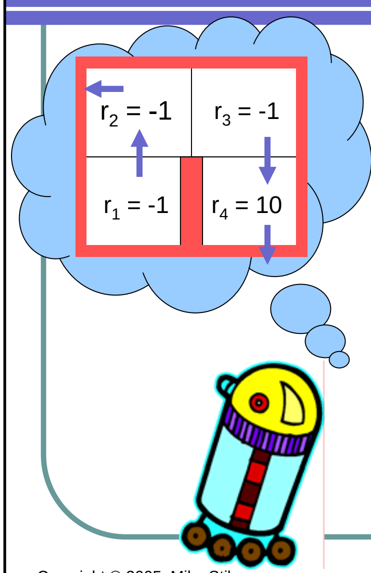
Policy Iteration



← Initial Policy

- 1) Find Value Function J*(S_i)
  - Value Iteration
    (No Max)
  - Matrix Inversion

Policy Iteration



← Initial Policy

- 1) Find Value Function $J^*(S_i)$

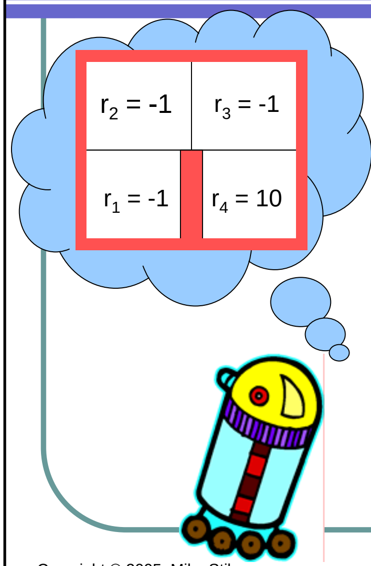
  $J^*(S_4) = 10 \sum^{\infty} \gamma^i =$
  $10 \times 1/(1-\gamma) = 100$

  $J^*(S_3) = -1 + .9\,[.9\,J^*(S_4) + .1\,J^*(S_3)]$
  $J^*(S_3) = -1 + 81 + .09\,J^*(S_3)$
  $J^*(S_3) = 87.9$

  Analogously…

  $J^*(S_2) = -1 \sum^{\infty} \gamma^i = -10$
  $J^*(S_1) = -10$

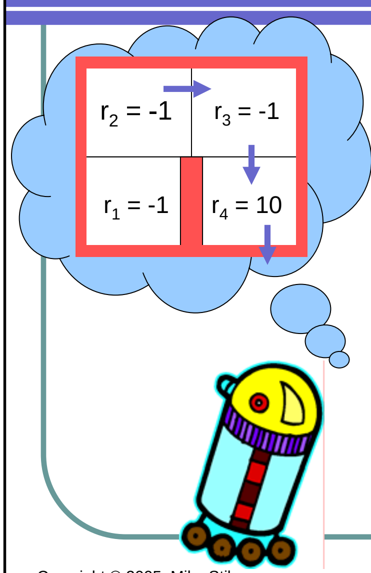
Policy Iteration



- 1) Find Value Function $J^*(S_i)$
  - $J^*(S_4) = 100$
  - $J^*(S_3) = 87.9$
  - $J^*(S_2) = -10$
  - $J^*(S_1) = -10$
- 2) Compute *Improved* Policy
  - $\pi_1(S_i) = \text{argmax}_a \, [\, r_i + \gamma \sum_j P^a_{ij} J^*(S_j) \,]$

$pi_1(S_2)$ = argmax_a [⇧⇦⇩ -1 + .9 (1 × -10),
⇨ -1 + .9 (.9 × 87.9 +.1 × -10)]=
argmax_a [-10, 69.3] = RIGHT! ⇨

Policy Iteration



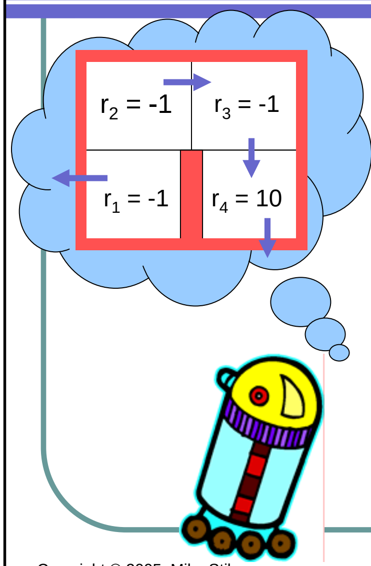
← New Policy

- 1) Find Value Function $J^*(S_i)$
  - $J^*(S_4) = 100$
  - $J^*(S_3) = 87.9$
  - $J^*(S_2) = -10$
  - $J^*(S_1) = -10$
- 2) Compute *Improved* Policy
  - $\pi_1(S_i) = \text{argmax}_a \, [\, r_i + \gamma \sum_j P^a_{ij} J^*(S_j) \,]$

$pi_1(S_3)$ = DOWN
$pi_1(S_4)$ = DOWN/LEFT/RIGHT
$pi_1(S_1)$ = argmax_a [-10, -10] !!!

What do we do?

Policy Iteration



← New Policy

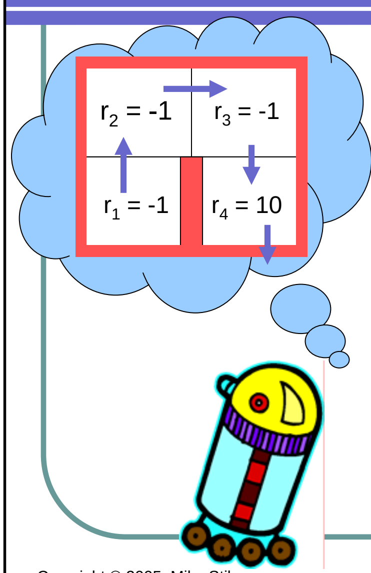
- 1) Find Value Function $J^*(S_i)$
  - $J^*(S_4) = 100$
  - $J^*(S_3) = 87.9$
  - $J^*(S_2) = -10$
  - $J^*(S_1) = -10$
- 2) Compute *Improved* Policy
  $\pi_1(S_i) = \text{argmax}_a [\, r_i + \gamma \sum_j P^a_{ij} J^*(S_j) \,]$

$pi_1(S_1) = \text{argmax}_a [\,-10, -10]$!

What do we do? Randomly choose left?

Then still bad policy, go back to step 1

Policy Iteration



← New Policy

- 1) Find Value Function $J^*(S_i)$
  - $J^*(S_4) = 100$
  - $J^*(S_3) = 87.9$
  - $J^*(S_2) = -10$
  - $J^*(S_1) = -10$
- 2) Compute *Improved* Policy
  $\pi_1(S_i) = \text{argmax}_a [\, r_i + \gamma \sum_j P^a_{ij} J^*(S_j) \,]$

$pi_1(S_1) = \text{argmax}_a [\,-10, -10]$!

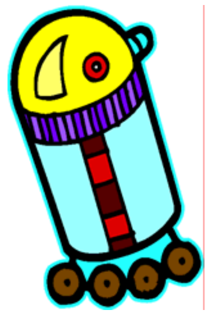
What do we do? Randomly choose up?

Both cases must converge to this solution!

We assumed a model for P(j|i,a)

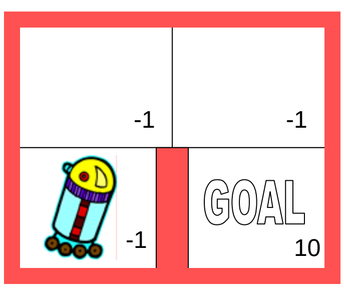
What do we do if such a model does not exist?

- Make one
- Q-Learning



Q-Learning

$Q^{t+1}(S_i,a) \leftarrow \alpha\, [\, r_i + \gamma \max_{a^1} Q^t(S_j,a^1)\,] + (1-\alpha)\, Q^t(S_i,a)$



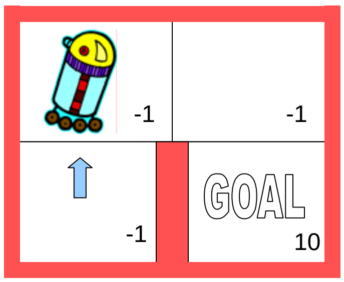
$\alpha = .7$

Q-Table

	↑	↓	←	→
S_1	0	0	0	0
S_2	0	0	0	0
S_3	0	0	0	0
S_4	0	0	0	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \, [\, r_i + \gamma \max_{a^1} Q^t(S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



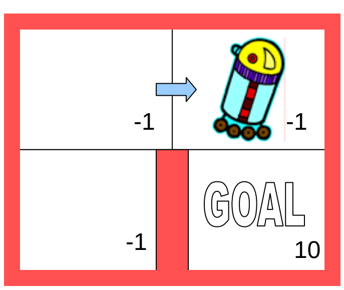
$Q^{est}(S_1, ↑) =$
$.7(-1 + .9 \max(0, 0, 0, 0)) + .3 \times 0$

Q-Table

	↑	↓	←	→
S_1	**-.7**	0	0	0
S_2	0	0	0	0
S_3	0	0	0	0
S_4	0	0	0	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \, [\, r_i + \gamma \max_{a^1} Q^t(S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



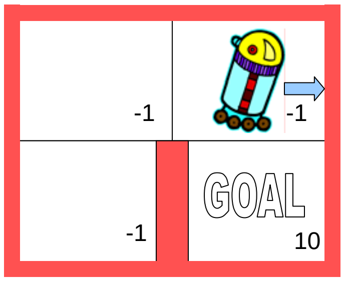
$Q^{est}(S_2, →) =$
$.7(-1 + .9 \max(0, 0, 0, 0)) + .3 \times 0$

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	**-.7**
S_3	0	0	0	0
S_4	0	0	0	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \,[\, r_i + \gamma \max_{a^1} Q^t (S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



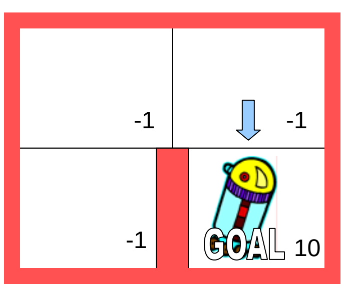
$Q^{est}(S_3, \rightarrow) =$
$.7(-1 + .9 \max(0, 0, 0, 0)) + .3 \times 0$

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	-.7
S_3	0	0	0	**-.7**
S_4	0	0	0	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \,[\, r_i + \gamma \max_{a^1} Q^t (S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



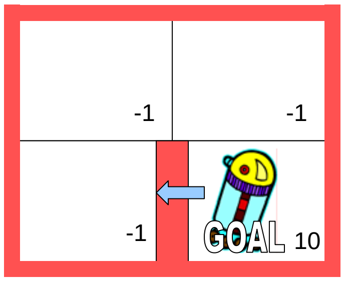
$Q^{est}(S_3, \downarrow) =$
$.7(-1 + .9 \max(0, 0, 0, 0)) + .3 \times 0$

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	-.7
S_3	0	**-.7**	0	-.7
S_4	0	0	0	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \, [\, r_i + \gamma \max_{a^1} Q^t(S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



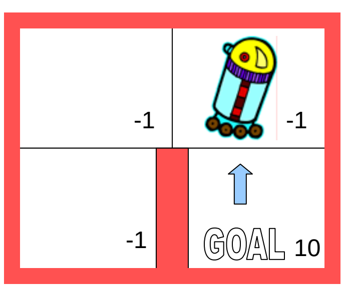
$Q^{est}(S_4$, ←) =
.7(10 + .9 max (0, 0, 0, 0)) + .3 × 0

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	-.7
S_3	0	-.7	0	-.7
S_4	0	0	**7**	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \, [\, r_i + \gamma \max_{a^1} Q^t(S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



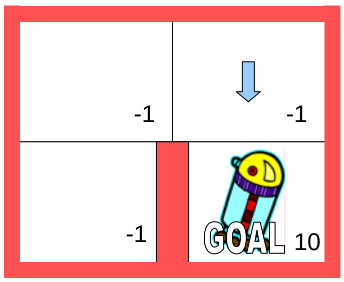
$Q^{est}(S_4$, ↑) =
.7(10 + .9 max (0, -.7, 0, -.7)) + .3 × 0

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	-.7
S_3	0	-.7	0	-.7
S_4	**7**	0	7	0

Q-Learning

$$Q^{t+1}(S_i,a) \leftarrow \alpha \, [\, r_i + \gamma \max_{a^1} Q^t (S_j,a^1) \,] + (1-\alpha)\, Q^t(S_i,a)$$



$Q^{est}(S_3,$ ↓ $) =$
$.7(-1 + .9 \max(7,0,7,0)) + .3 \times -.7$

Q-Table

	↑	↓	←	→
S_1	-.7	0	0	0
S_2	0	0	0	-.7
S_3	0	**3.5**	0	-.7
S_4	7	0	7	0