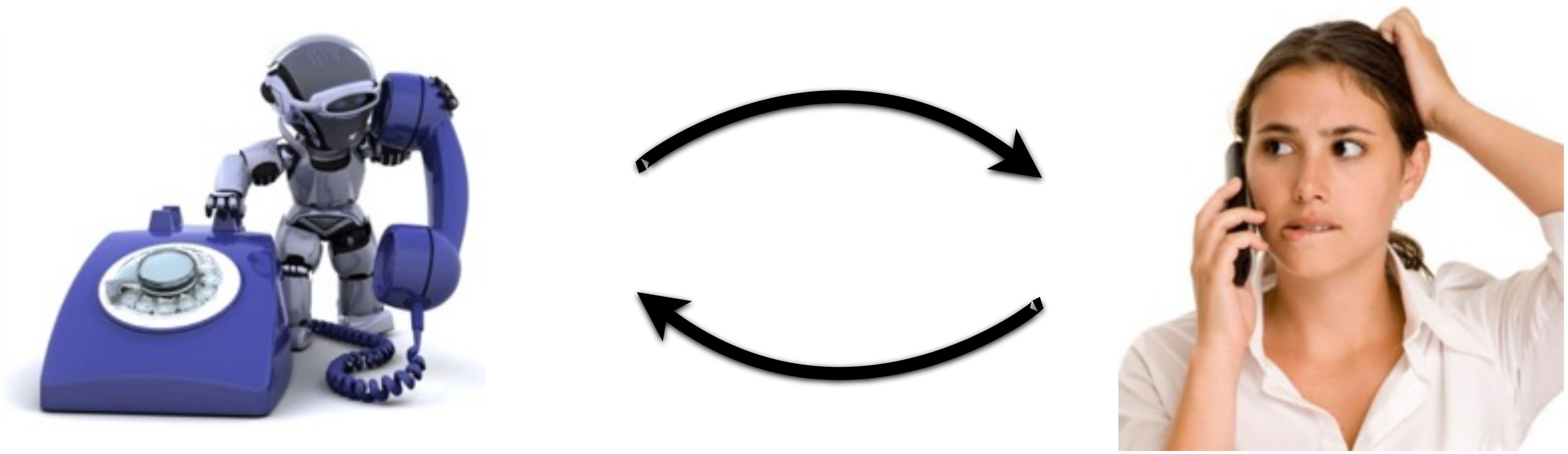
Introduction to Markov Decision Processes

Fall - 2013

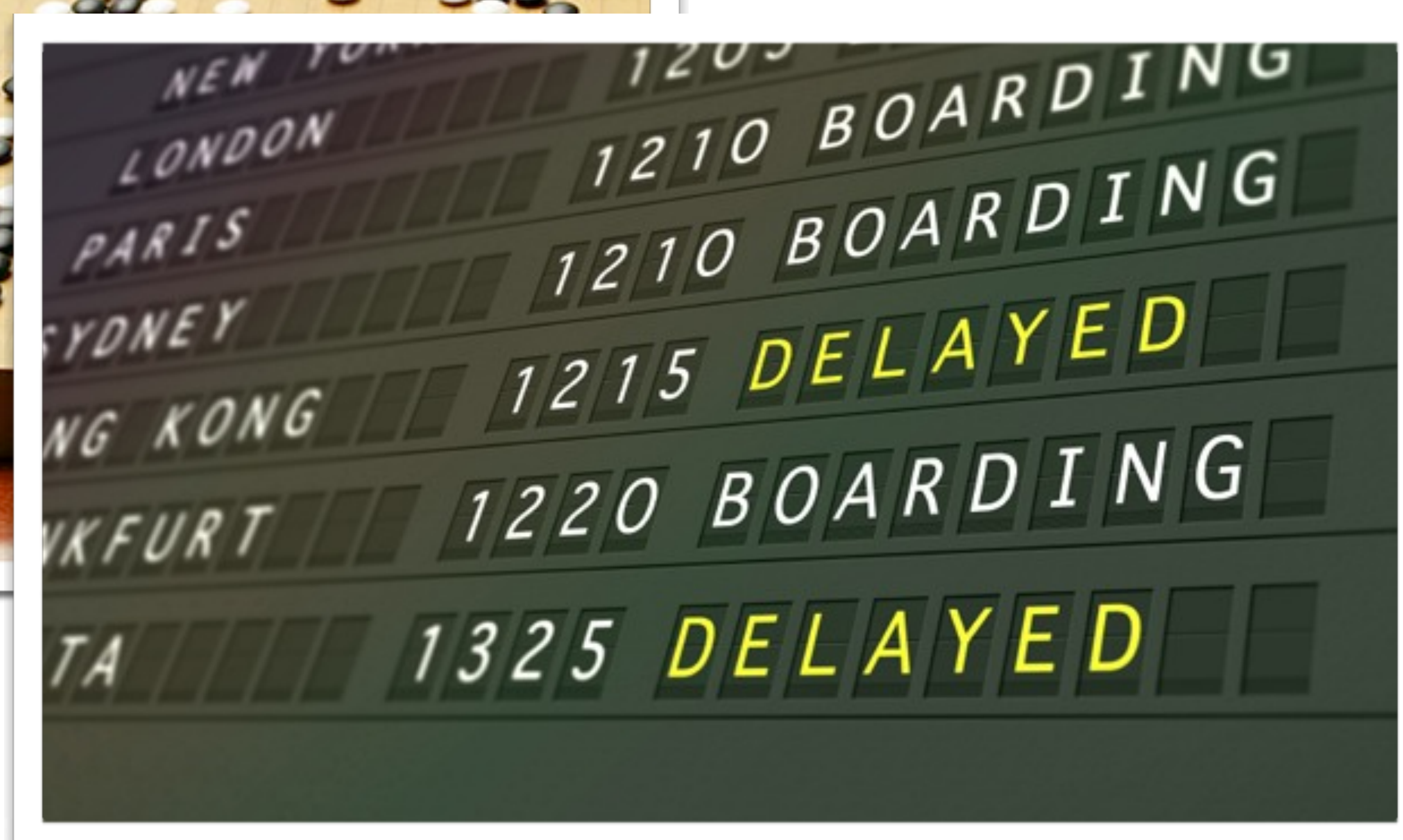
Alborz Geramifard
Research Scientist at Amazon.com
*This work was done during my postdoc at MIT.

Motivation

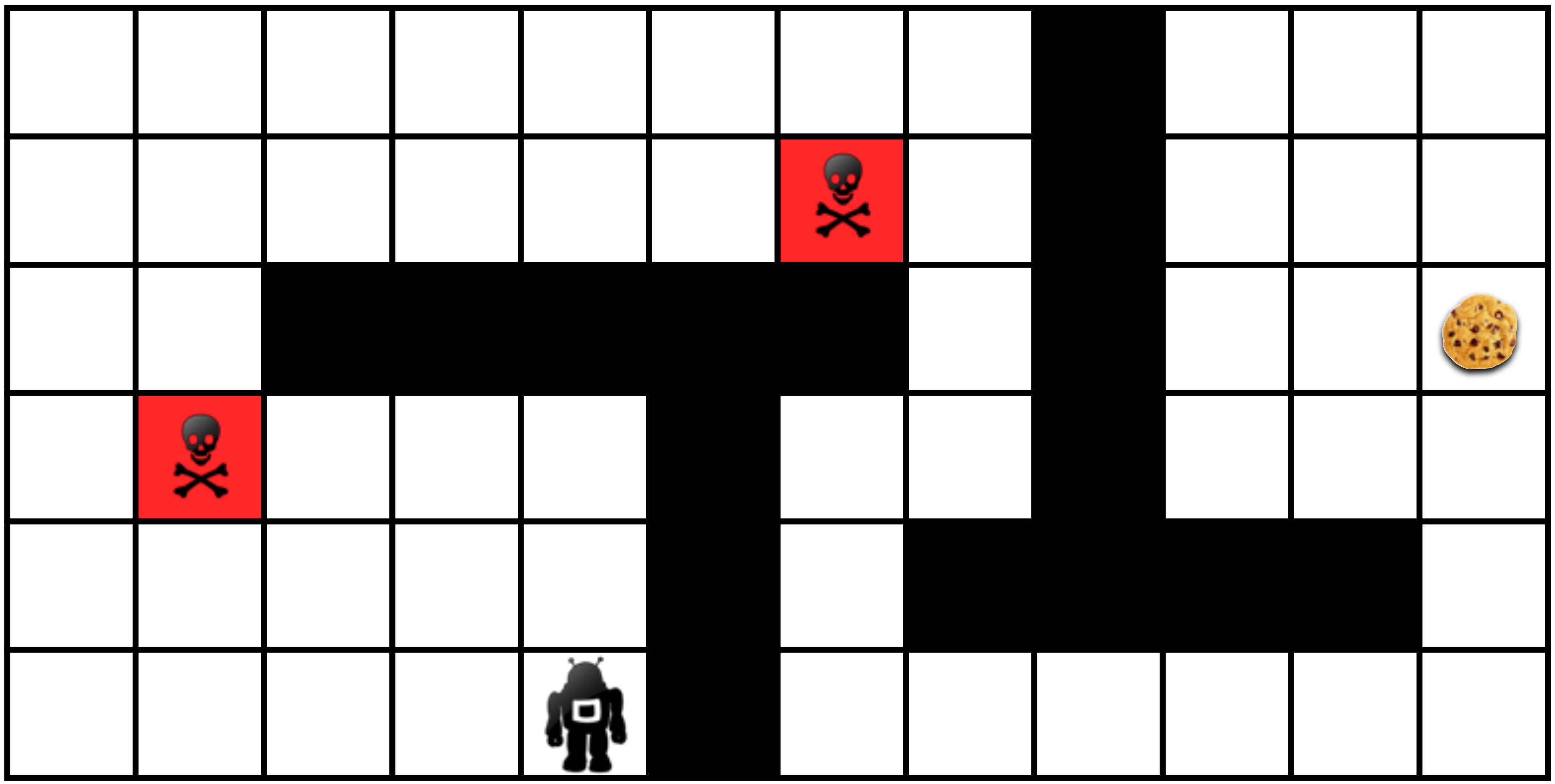


- **Understand** the customer's need in a **sequence** of interactions.
- **Minimize** a notion of accumulated frustration level.

Applications



Grid World Example



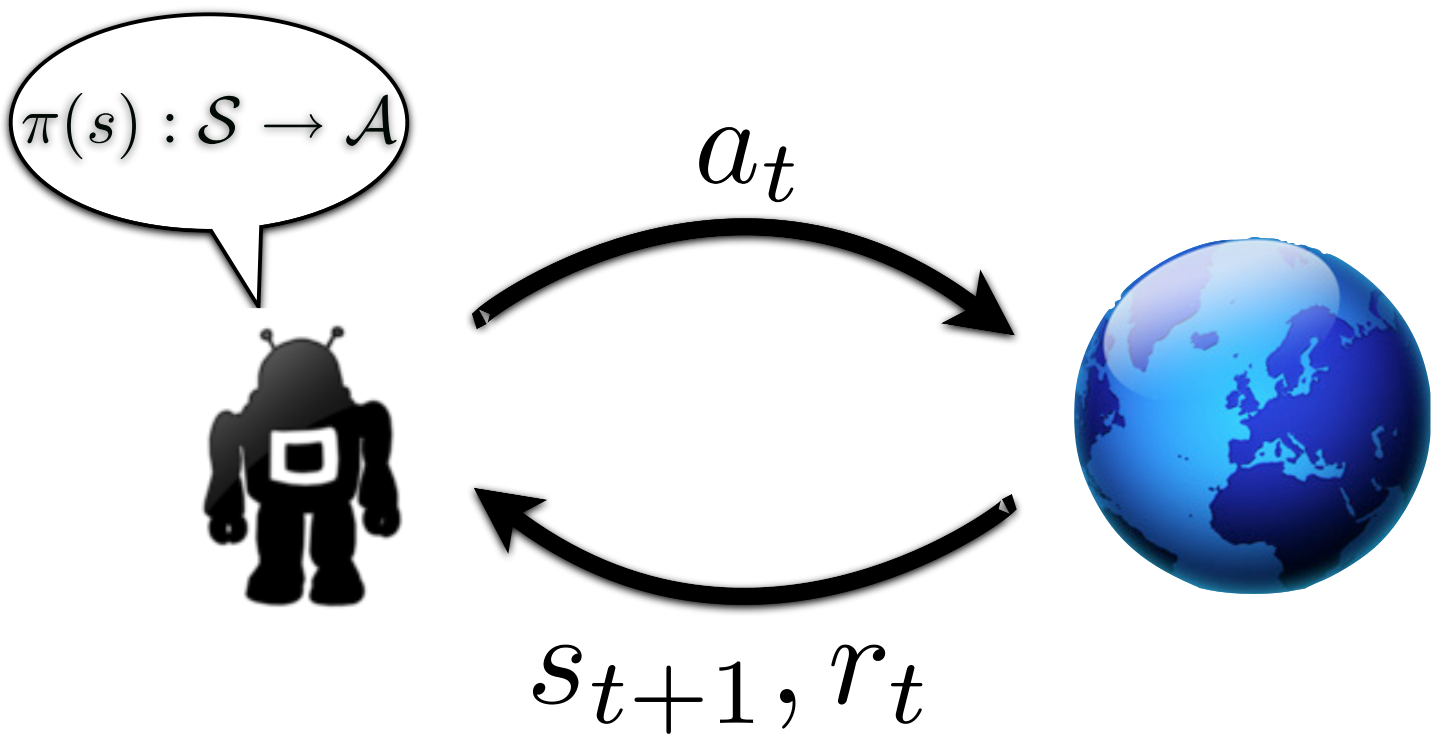
- **Goal:** Grab the cookie fast and avoid pits
- **Noisy** movement
- **Actions:** $\rightarrow, \leftarrow, \uparrow, \downarrow$

Outline

- Motivation
- Problem Formulation
- Solving MDPs
- Extensions

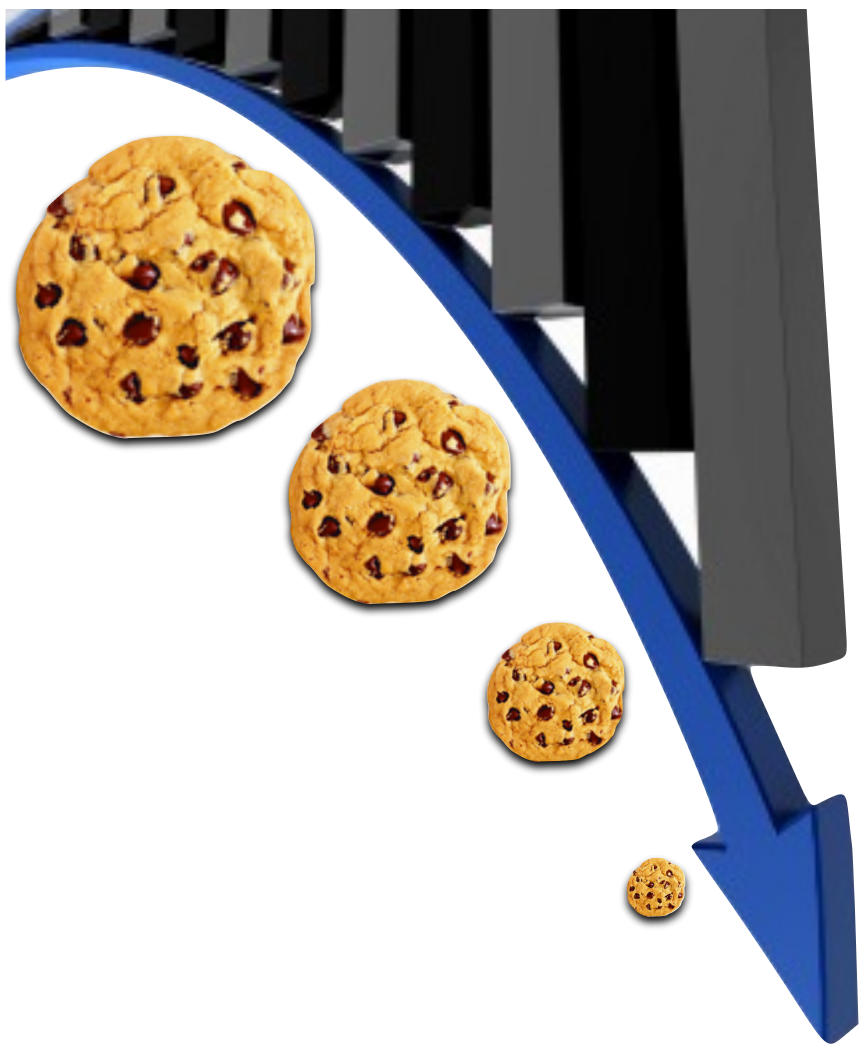
Markov Decision Process

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma)$$

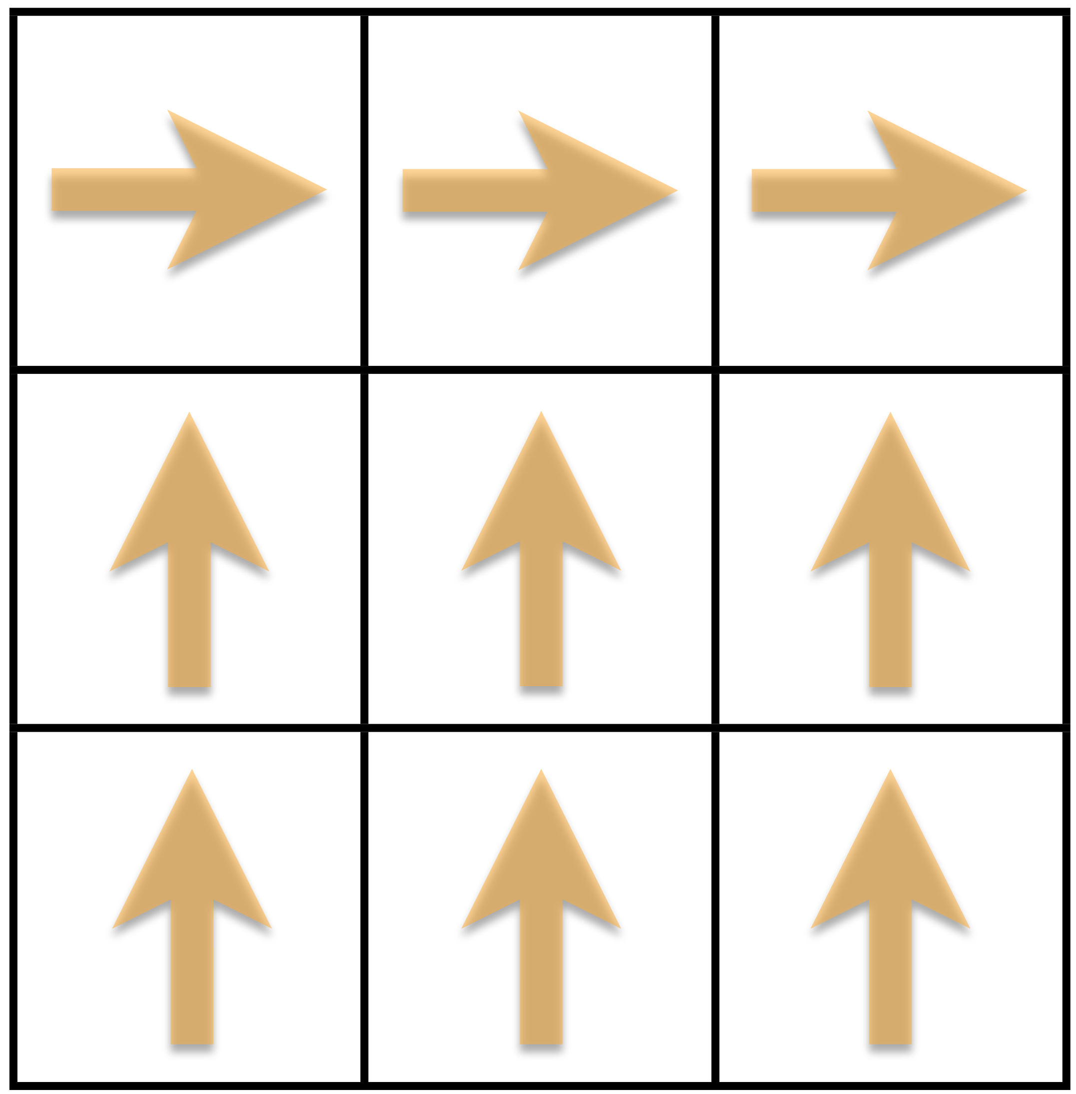


Markov Decision Process

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma)$$

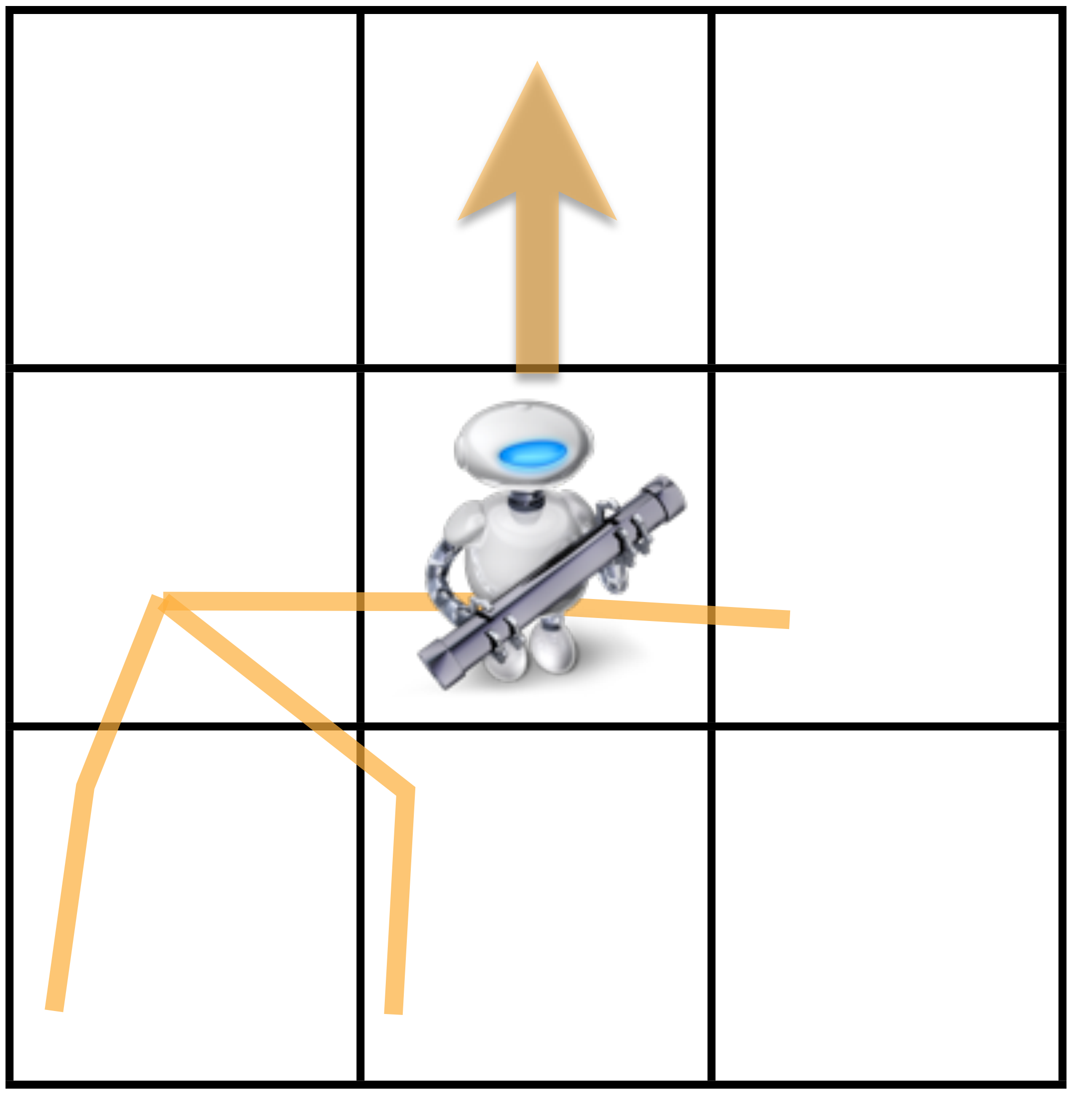


Policy (π): S → A



Assumptions

- Fully Observable
- Markovian Property



State Values

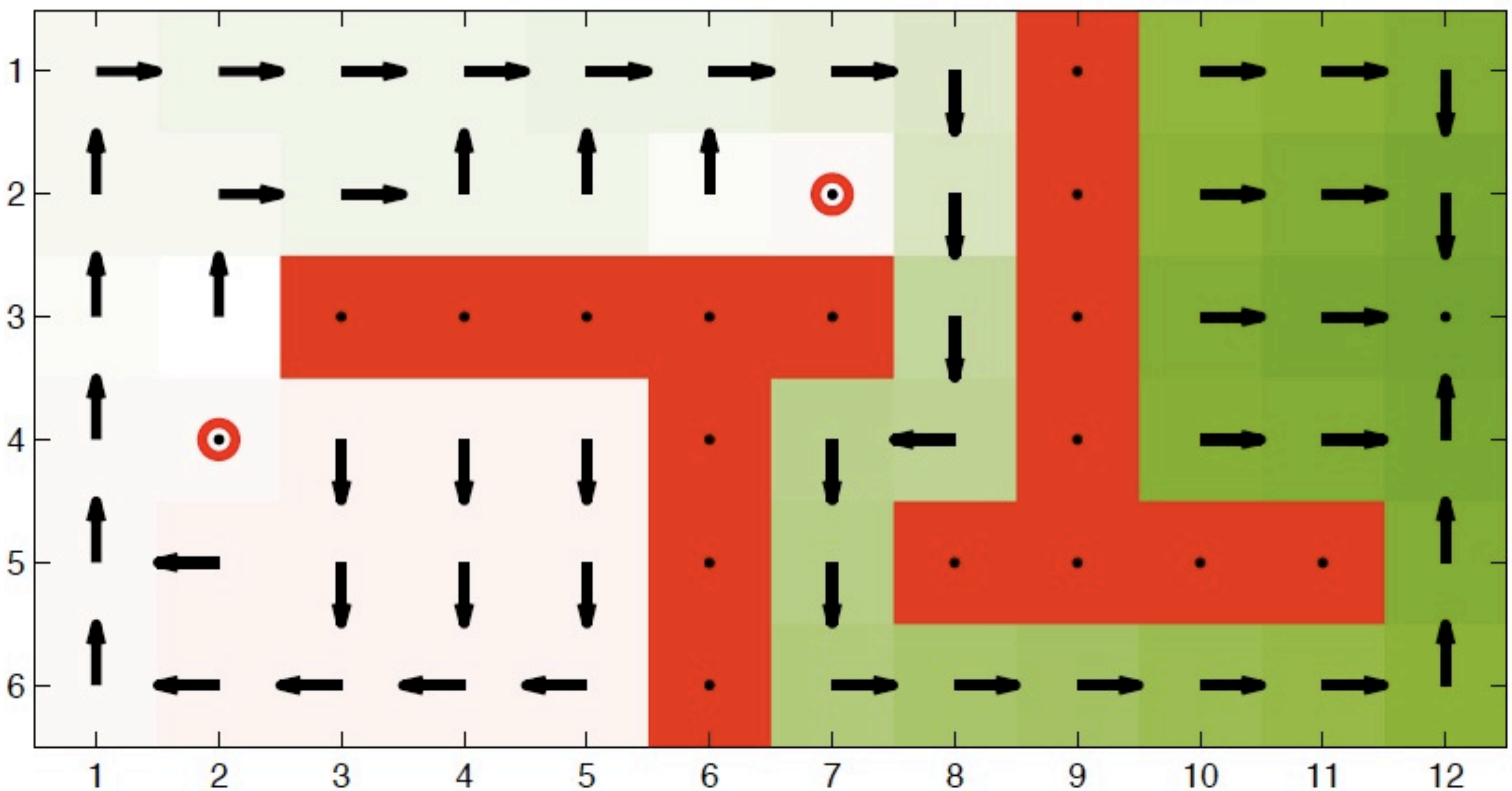
$$s_0 \xrightarrow{a_0, r_1} s_1 \xrightarrow{a_1, r_2} s_2 \quad \cdots$$

$$Q^\pi(s,a) = E\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_0 = s, a_0 = a, \pi\right]$$

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

Problem

$$\pi^* = \max_{\pi}, \forall s \in \mathcal{S}, V^{\pi}(s)$$



Outline

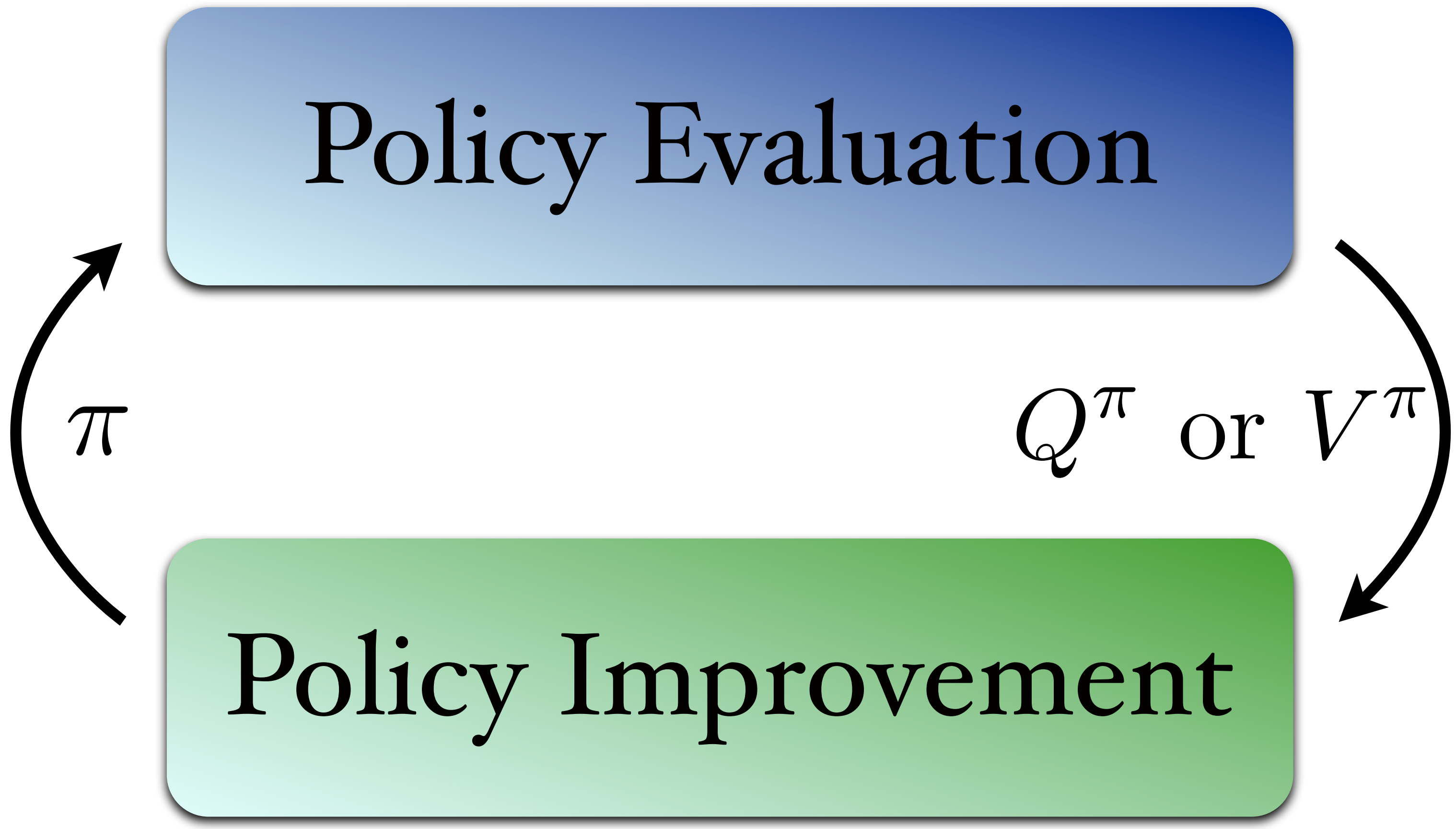
- Motivation
- Problem Formulation
- Solving MDPs
- Extensions

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}_{ss'}^{a}, \mathcal{R}_{ss'}^{a}, \gamma)$$

Assume all elements of the MDP are known.

Dynamic Programming

- Given a fixed policy (π), estimate the value of each state



- Given a fixed value function, improve the policy (π)

Loop till convergence

Policy Evaluation

$$Q^{\pi}(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma Q^{\pi}(s', \pi(s')) \right]$$

- Solve by formulating as a set of **linear equations**
- **Costly** calculation: $\mathcal{O}(|S|^3)$

Policy Improvement

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q(s,a)$$

Policy Iteration

Policy Evaluation

$$Q(s,a) \leftarrow \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q(s',a')]$$

- Improve the value of a **single** state-action pair at a time
- **Lower** computation: $\mathcal{O}(|\mathcal{S}|)$*

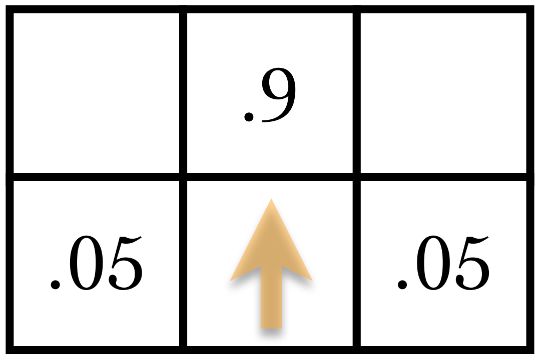
Policy Improvement

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q(s,a)$$

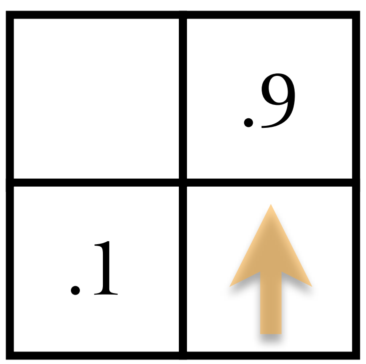
Value Iteration

Value Iteration Example

Transition Model



Transition Model



$\gamma = 1$

Reward

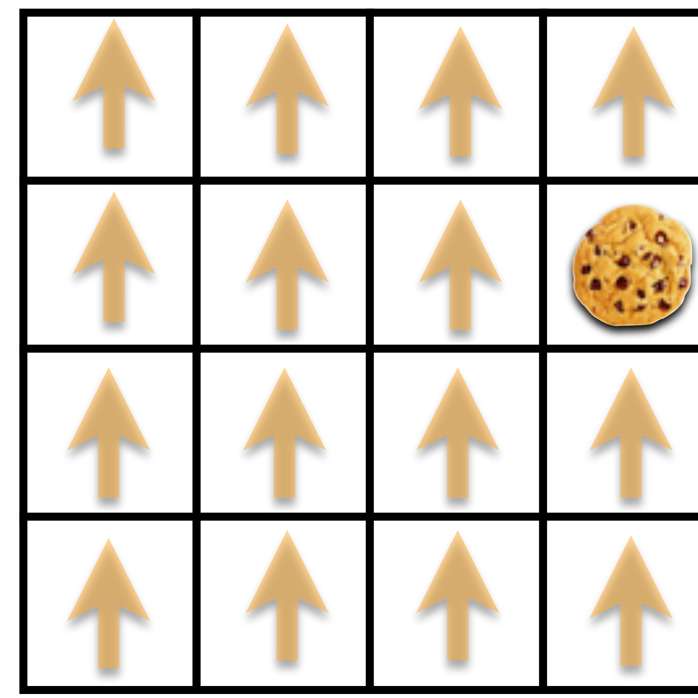
 = 100

Iteration 0:

$V^{\pi}(s)$

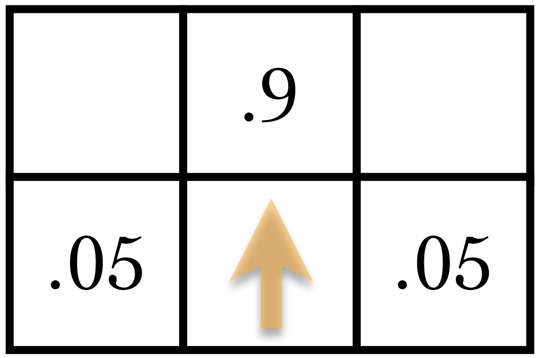
0	0	0	0
0	0	0	
0	0	0	0
0	0	0	0

Policy

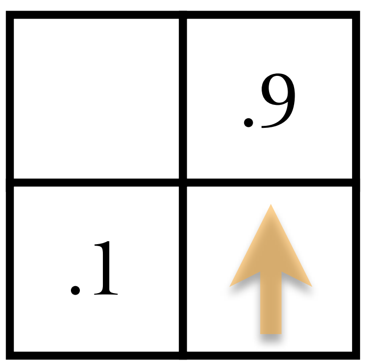


Value Iteration Example

Transition Model



Transition Model



$\gamma = 1$

Reward

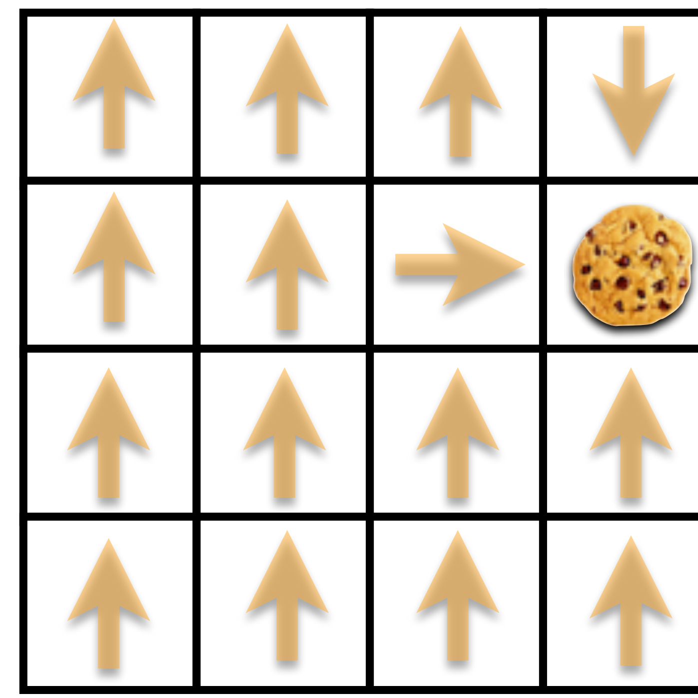
 = 100

Iteration 1:

$V^{\pi}(s)$

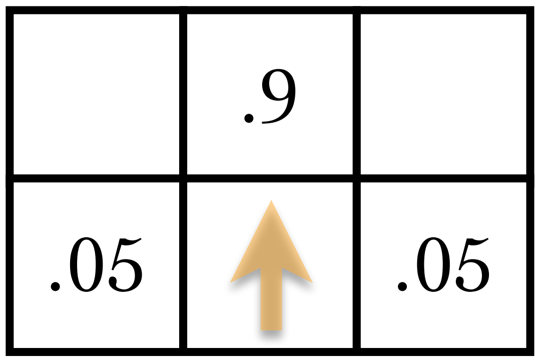
0	0	0	90
0	0	90	
0	0	0	90
0	0	0	0

Policy

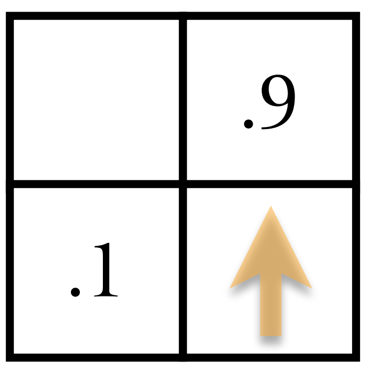


Value Iteration Example

Transition Model



Transition Model



$\gamma = 1$

Reward

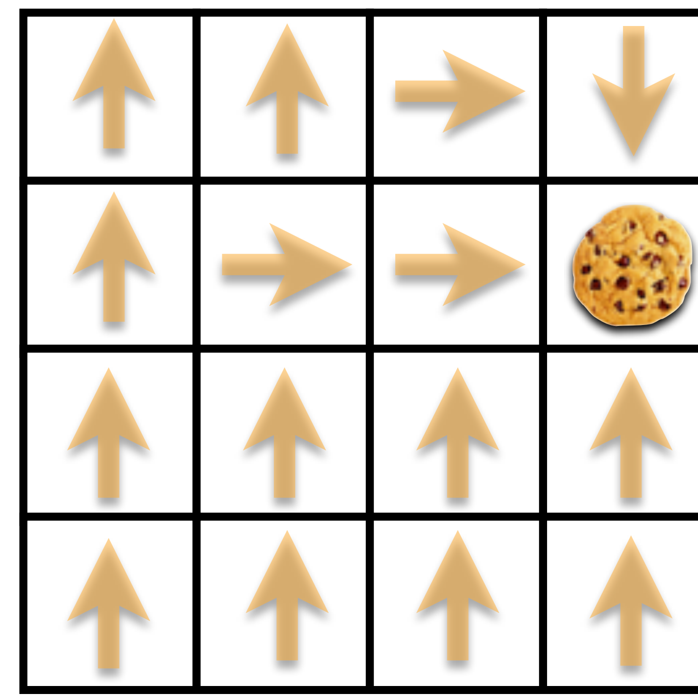
 = 100

Iteration 2:

$V^{\pi}(s)$

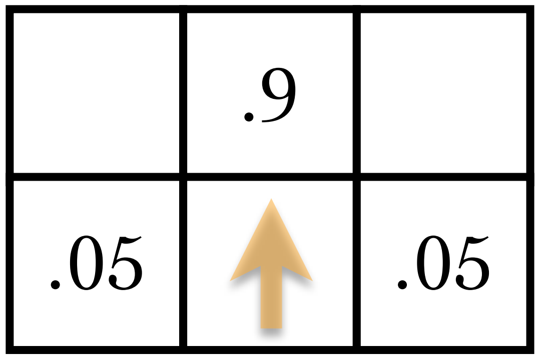
0	0	90	90
0	81	90	
0	0	85.5	90
0	0	0	81

Policy

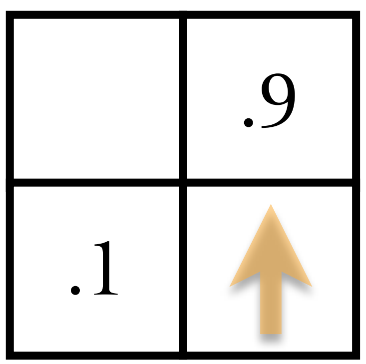


Value Iteration Example

Transition Model



Transition Model



$\gamma = 1$

Reward

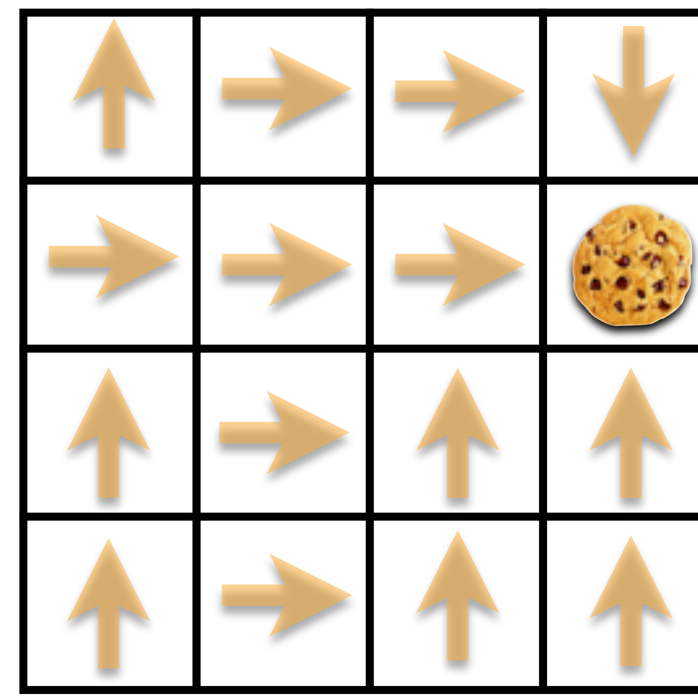
 = 100

Iteration 3:

$V^{\pi}(s)$

0	86	99	99
72.9	81	90	
0	81	89.8	98.5
0	0	81	81

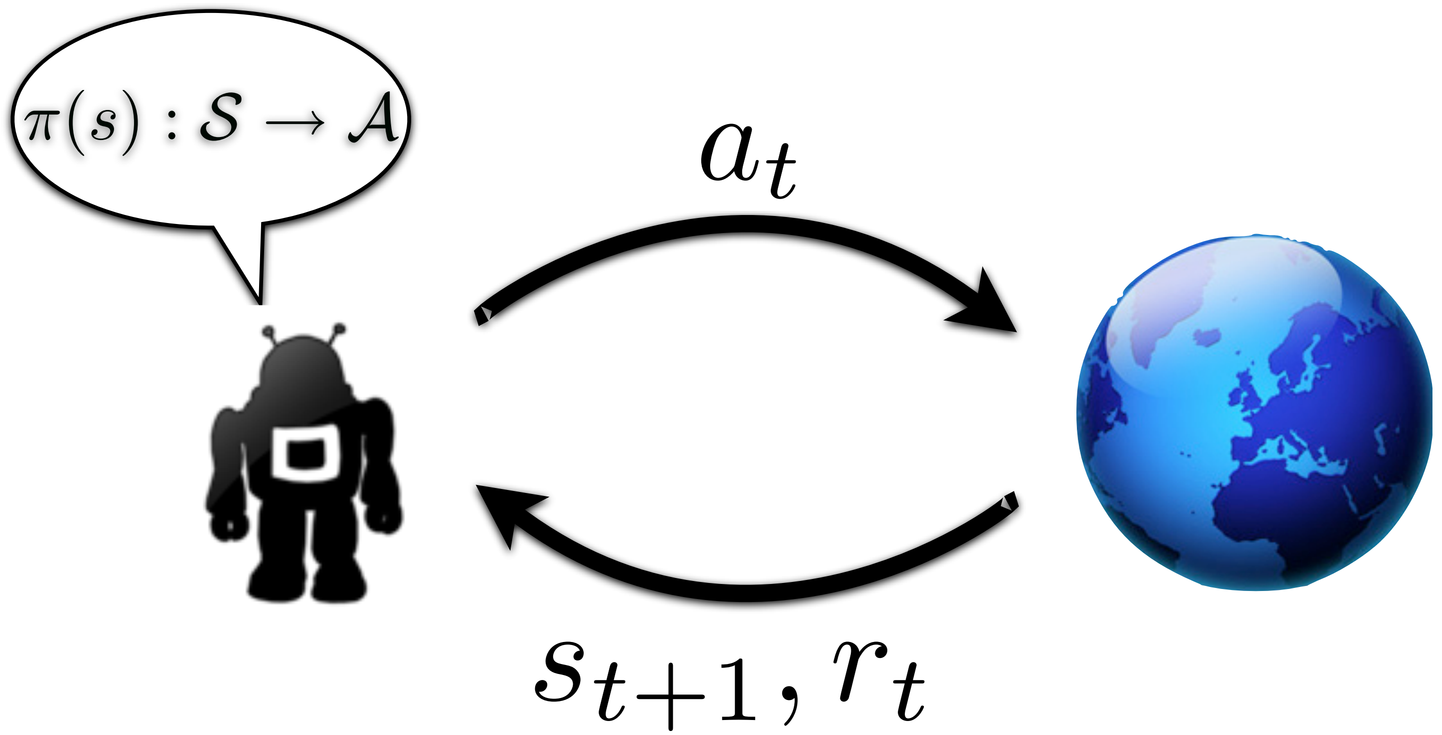
Policy



$$(\mathcal{S}, \mathcal{A}, \boxed{\mathcal{P}^{a}_{ss'}, \mathcal{R}^{a}_{ss'}}, \gamma)$$

Not known!

Reinforcement Learning



We only see this:

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2 \ldots.$$

Reinforcement Learning

[B.F. Skinner Foundation]

Reinforcement Learning

- **Unknown** $\mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}$
- What can we do with **only** samples?

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2 \ldots.$$

Policy Evaluation

$$Q(s,a) \leftarrow \underbrace{\boxed{\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q(s',a')]}}_{Q^+(s,a)}$$

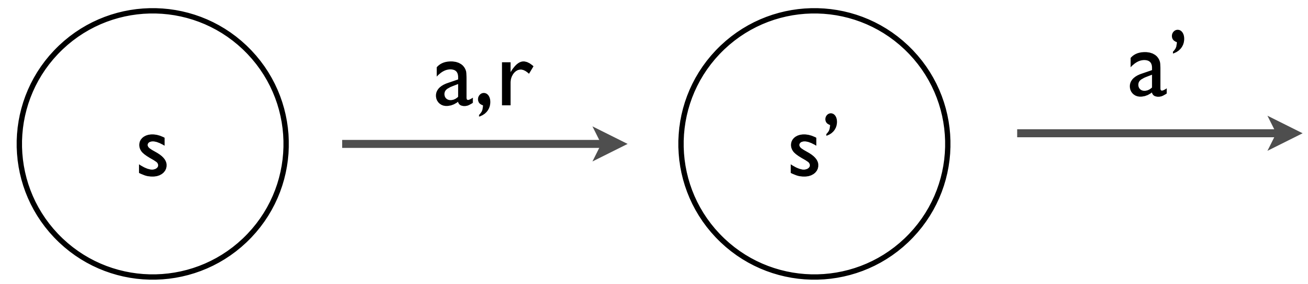
- Can we build a **noisy** estimate of $Q^+(s,a)$?

Policy Improvement

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q(s,a)$$

Value Iteration

Policy Evaluation



$$Q^+(s,a) = r_t + \gamma \max_{a'} Q(s',a')$$

$$\delta = Q^+(s,a) - Q(s,a)$$

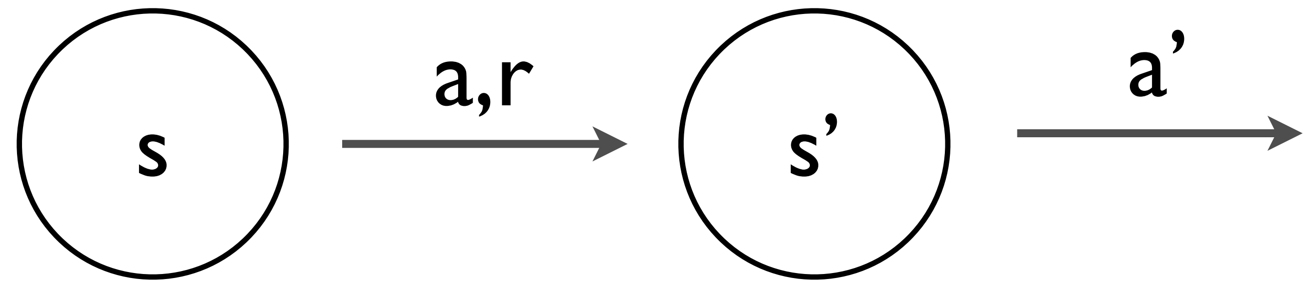
$$Q(s,a) = Q(s,a) + \alpha\delta$$

Policy Improvement

Q-Learning

$$\pi^\epsilon(s) \triangleq \begin{cases} \text{argmax}_a\, Q^\pi(s,a), & \text{with probability } 1-\epsilon \\ \texttt{UniformRandom}(\mathcal{A}), & \text{with probability } \epsilon \end{cases}$$

Policy Evaluation



$$Q^{+}(s,a) = r_t + \gamma Q(s',a')$$

$$\delta = Q^{+}(s,a) - Q(s,a)$$

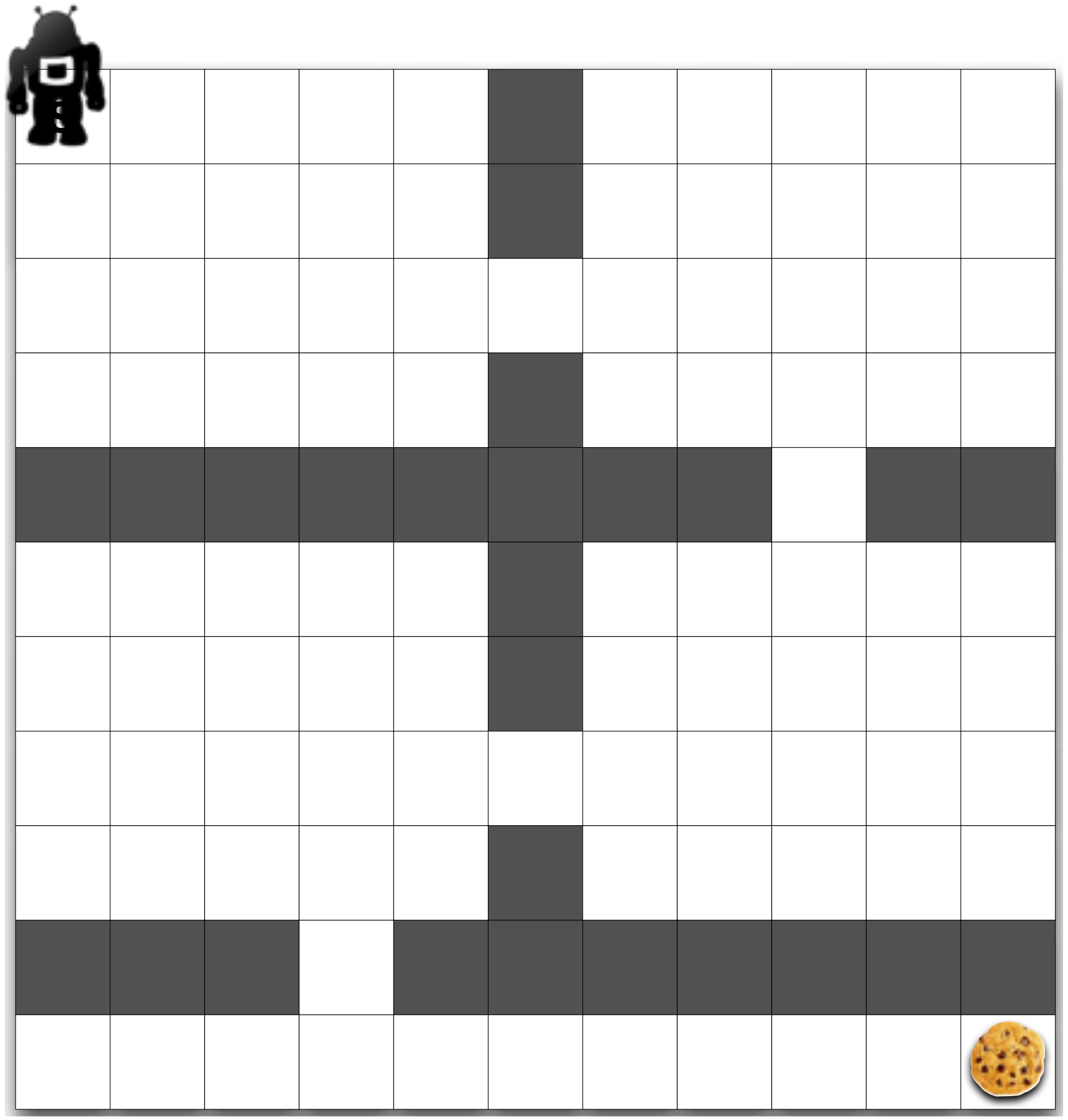
$$Q(s,a) = Q(s,a) + \alpha\delta$$

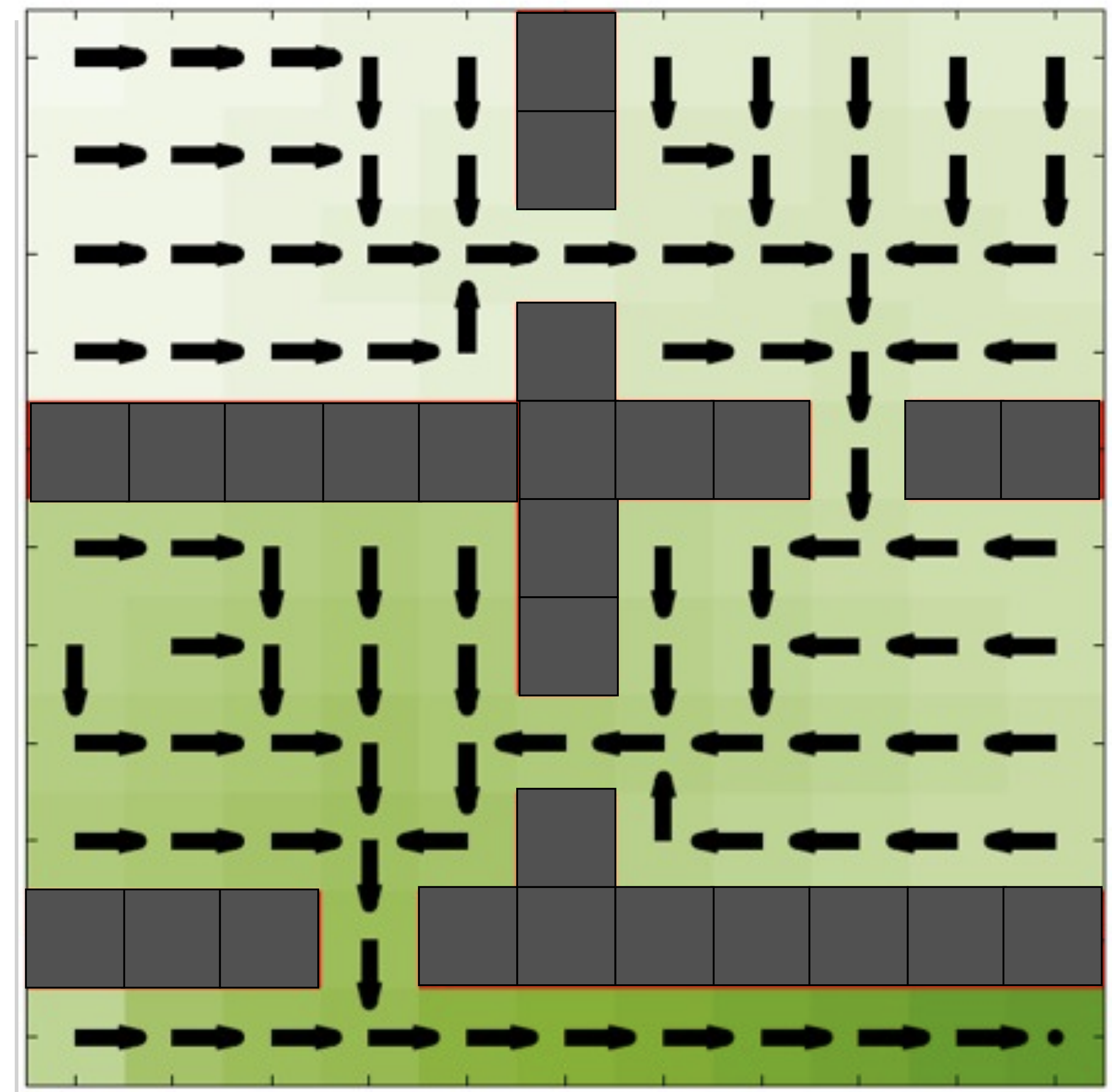
Policy Improvement

SARSA

$$\pi^{\epsilon}(s) \triangleq \begin{cases} \text{argmax}_a\, Q^{\pi}(s,a), & \text{with probability } 1-\epsilon \\ \texttt{UniformRandom}(\mathcal{A}), & \text{with probability } \epsilon \end{cases}$$

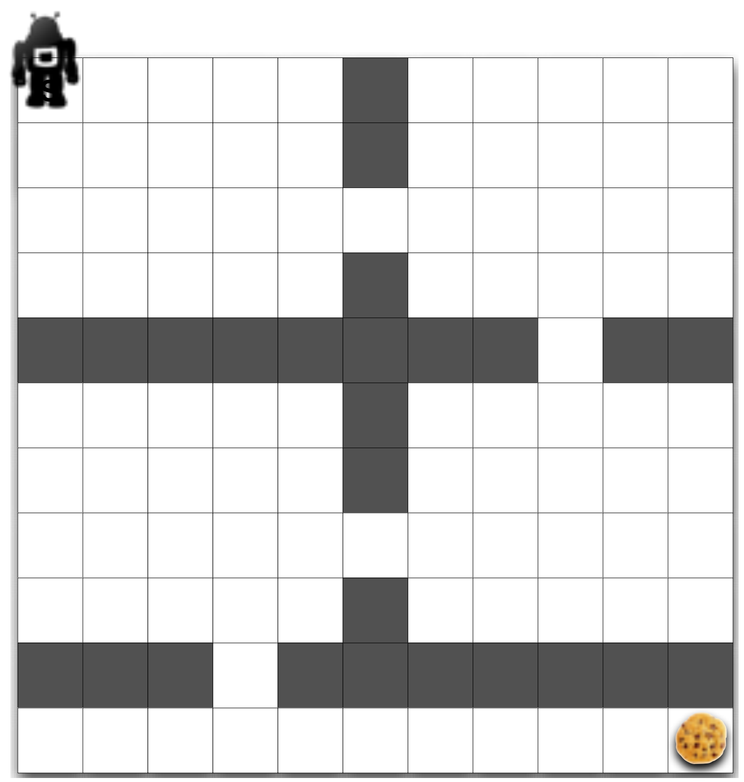
SARSA Example





- Rewards: +1 at goal, -.001 per step
- γ=.98
- Transitions: ↑↓←→, 30% noise
- |States|=95

What is the main challenge in solving MDPs with a **tabular** representation of values for every problem?



|States|=95

In practice, state spaces are **huge** ...

Huge State Spaces

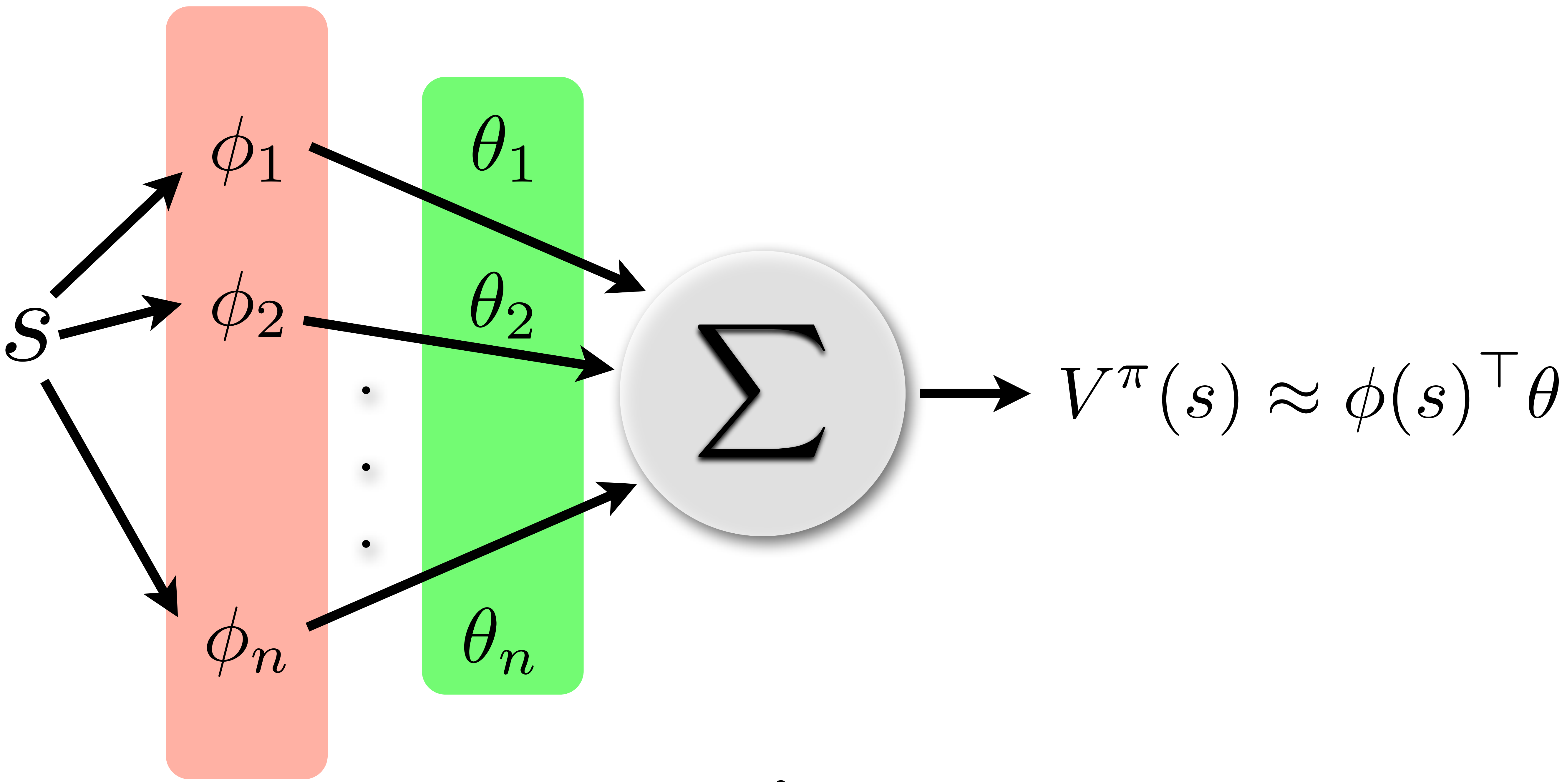
Dialog Turns	7
Frustration Level	10
Possible Sentences	10000
Caller Gender	2
Caller Location	4500

6.3 Billion Parameters

Outline

- Motivation
- Problem Formulation
- Solving MDPs
- Extensions

Linear Function Approximation



Example

State

Feature $\phi_t(s)$

Weight θ_t

Value

Male

Seattle

.

.

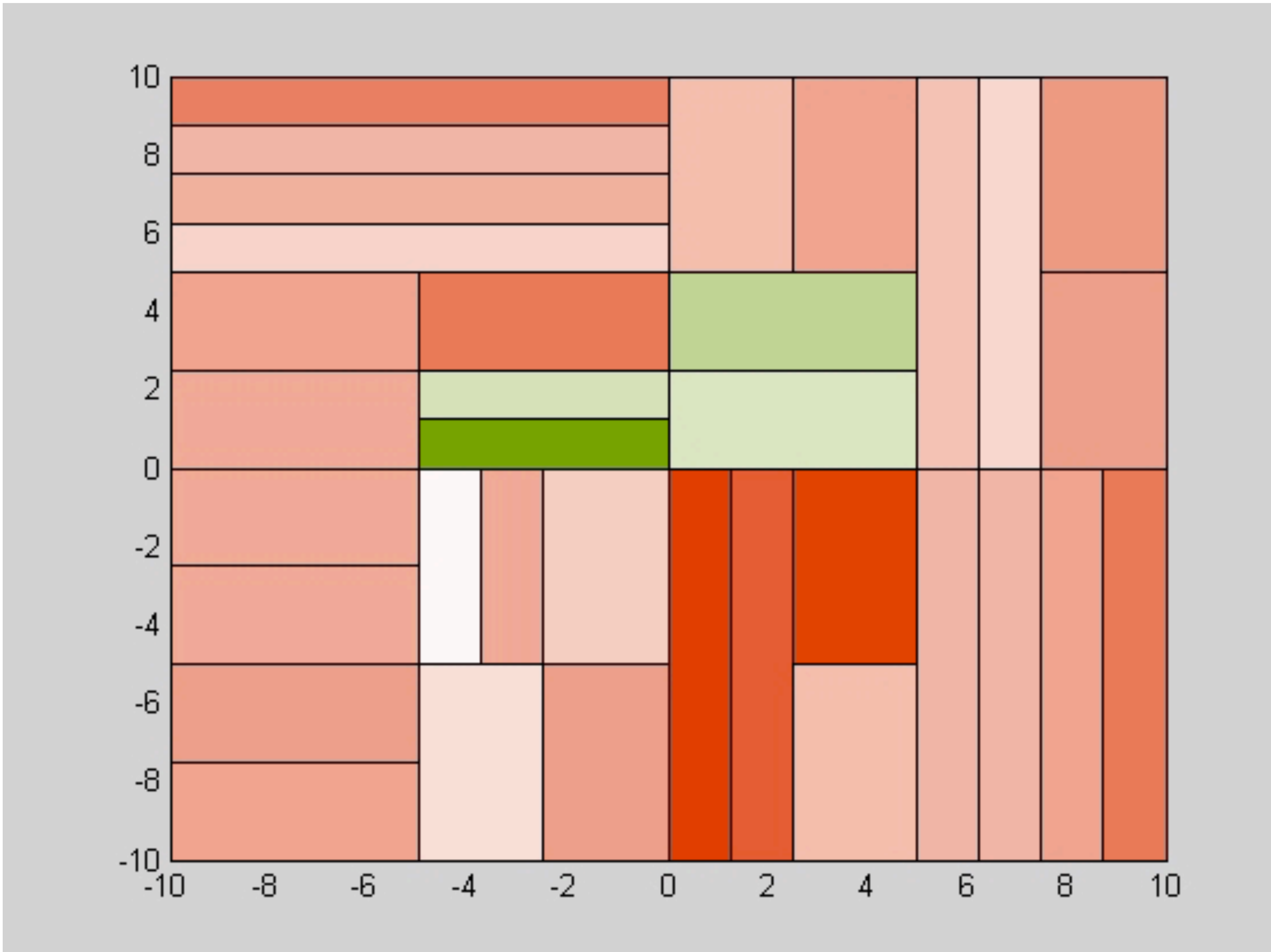
.

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 10 \\ 20 \\ 10 \\ 10 \\ 5 \end{bmatrix}$$

V(s) ≈ 20+10+10
= 40

- What is the **right** set of features?

Adaptive Tile Coding



[Whiteson et al. 2007]

Matrix Form

$$\tilde{V}_{\boldsymbol{\theta}} = \begin{bmatrix} \text{---}\ \phi^\top(s_1)\ \text{---} \\ \text{---}\ \phi^\top(s_2)\ \text{---} \\ \vdots \\ \text{---}\ \phi^\top(s_{|\mathcal{S}|})\ \text{---} \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix} \triangleq \boldsymbol{\Phi}_{|\mathcal{S}|\times m}\boldsymbol{\theta}_{m\times 1}$$

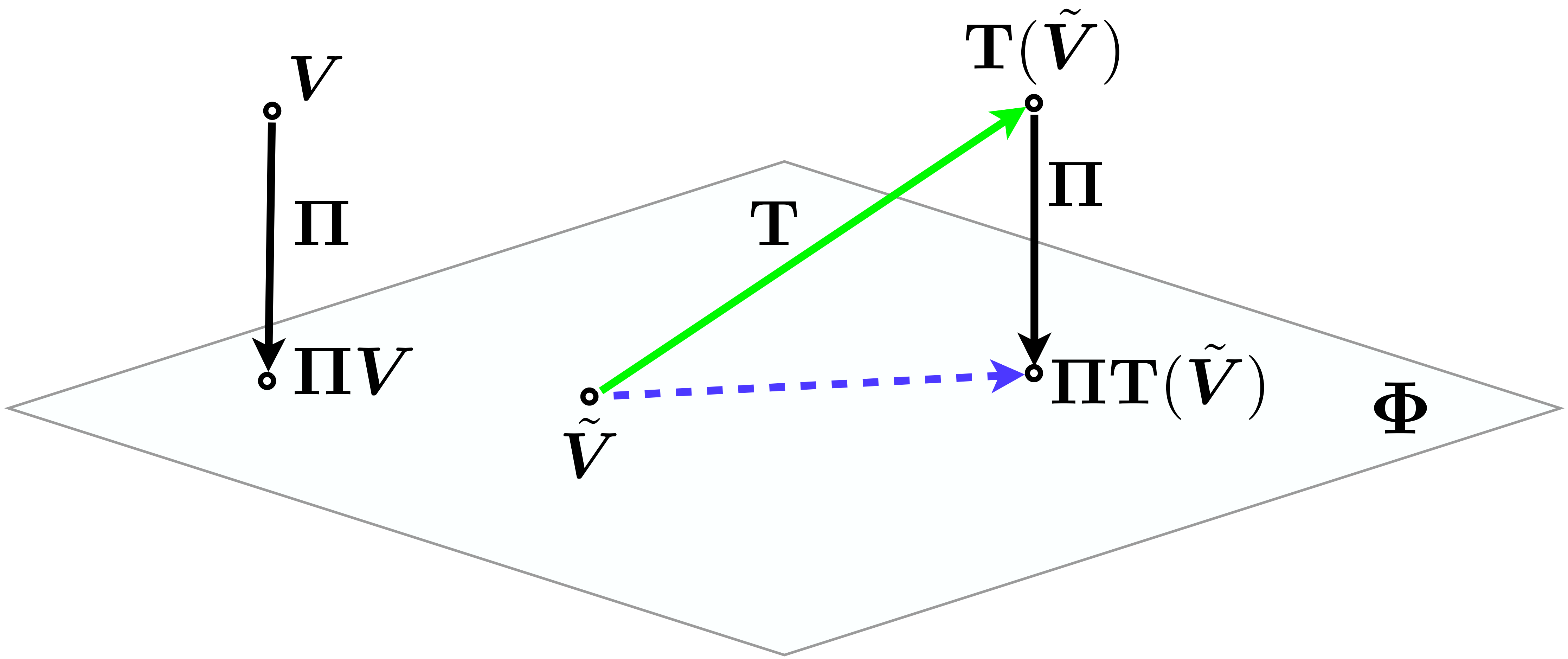
$$Q^\pi(s,a) = \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\left[\mathcal{R}^a_{ss'} + \gamma Q^\pi\left(s', \pi(s')\right)\right]$$

- Solve by formulating as a set of **linear equations**
- **Costly** calculation:

$$\mathbf{T}(\boldsymbol{V}) \triangleq \boldsymbol{R} + \gamma \boldsymbol{P}\boldsymbol{V}$$

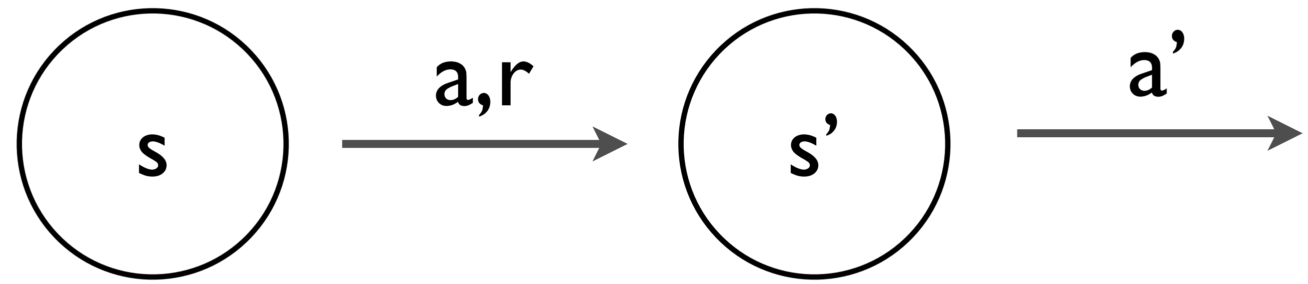
Geometric View

$$\Pi = \Phi(\Phi^T D \Phi)^{-1} \Phi^T D$$



$$\tilde{\mathbf{V}} = \boldsymbol{\Phi}\theta$$

Policy Evaluation



$$Q^+(s,a) = r_t + \gamma \max_{a'} Q(s',a')$$

$$\delta = Q^+(s,a) - Q(s,a)$$

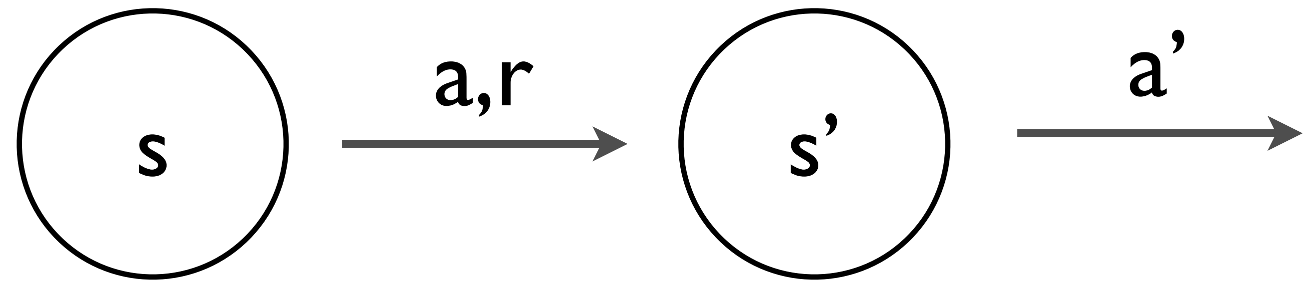
$$Q(s,a) = Q(s,a) + \alpha\delta\phi(s,a)$$

Policy Improvement

Q-Learning

$$\pi^\epsilon(s) \triangleq \begin{cases} \operatorname{argmax}_a Q^\pi(s,a), & \text{with probability } 1-\epsilon \\ \texttt{UniformRandom}(\mathcal{A}), & \text{with probability } \epsilon \end{cases}$$

Policy Evaluation



$$Q^+(s,a) = r_t + \gamma Q(s',a')$$

$$\delta = Q^+(s,a) - Q(s,a)$$

$$Q(s,a) = Q(s,a) + \alpha\delta\phi(s,a)$$

Policy Improvement

SARSA

$$\pi^\epsilon(s) \triangleq \begin{cases} \mathrm{argmax}_a\, Q^\pi(s,a), & \text{with probability } 1-\epsilon \\ \texttt{UniformRandom}(\mathcal{A}), & \text{with probability } \epsilon \end{cases}$$