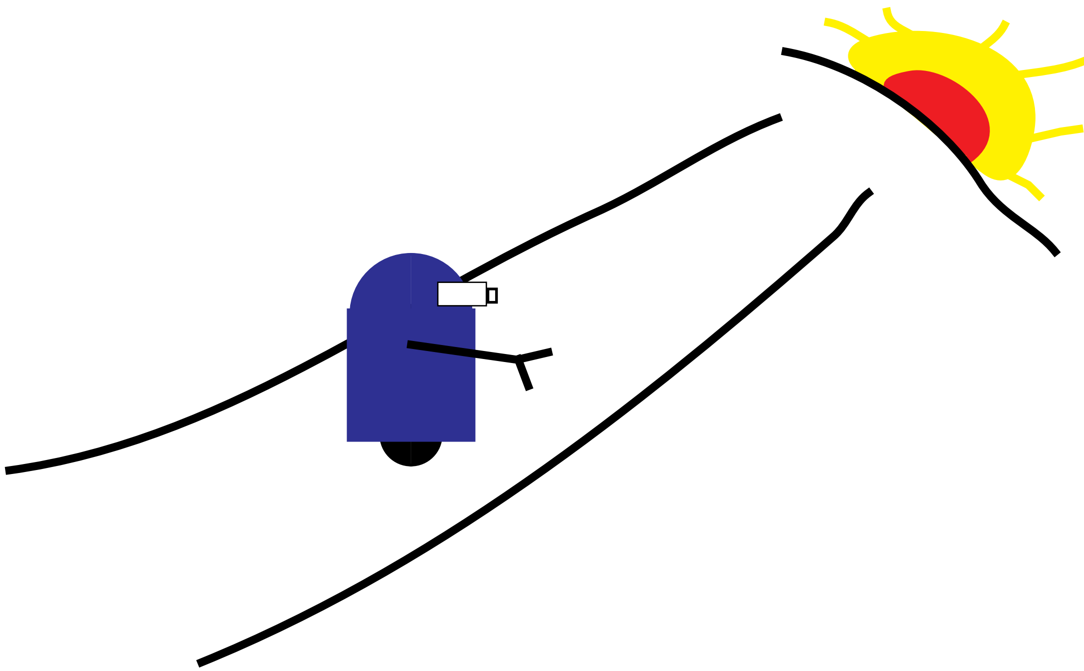


Conclusions

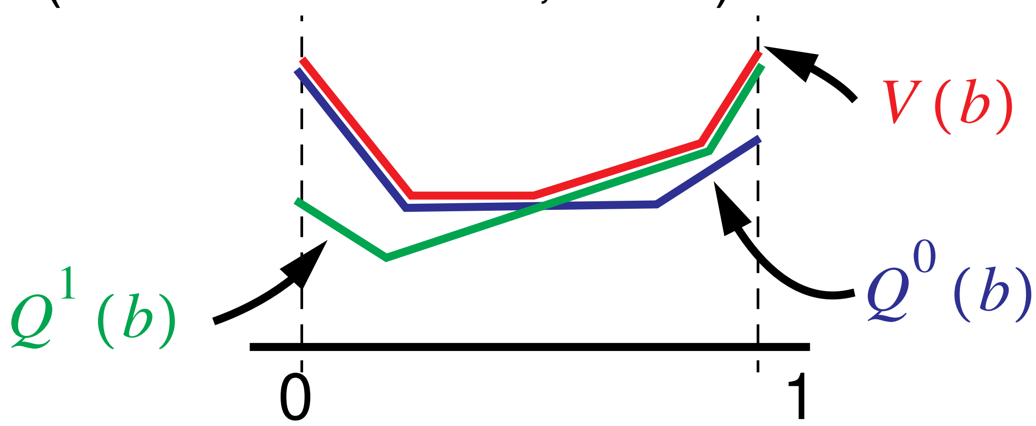
- MDPs are a useful framework for planning in domains where the outcomes of actions are uncertain
- POMDPs are a useful framework for thinking about planning in domains where observations are noisy and not completely informative
- POMDPs allow graceful combination of acting to change the state of the world and acting to gain information
- exact POMDP planning is hopelessly intractable
- approximation methods may make POMDP planning useful for medium-sized problems

Richer Representations

In some problems, no linear representation of the Q-functions can do well.

Possibilities:

- allow multiple linear segments for each Q-function
- use smooth convex approximation for the value function (Parr and Russell, 1995)



Data

$Q_{\mathbf{MDP}}$ biased initial q_i^a values, 75,000 updates

	57 States		89 States	
	Goal %	Median	Goal %	Median
Linear-Q	11.4	>251	5.7	>251
Random	46.2	>251	25.9	>251
$Q_{\mathbf{MDP}}$	47.4	>251	25.9	>251
Human	100.0	15	100.0	29
Biased Linear-Q	95.2	15	58.0	51

More updates will yield better policies.

Adding Bias

Problems:

- $Q_{\mathbf{MDP}}$ will not improve with time
- Linear-Q needs good values to start

Hybrid Solution: bias initial values with $Q_{\mathbf{MDP}}$ values

Experimental Results

Random initialization of q_i^a , 75,000 updates

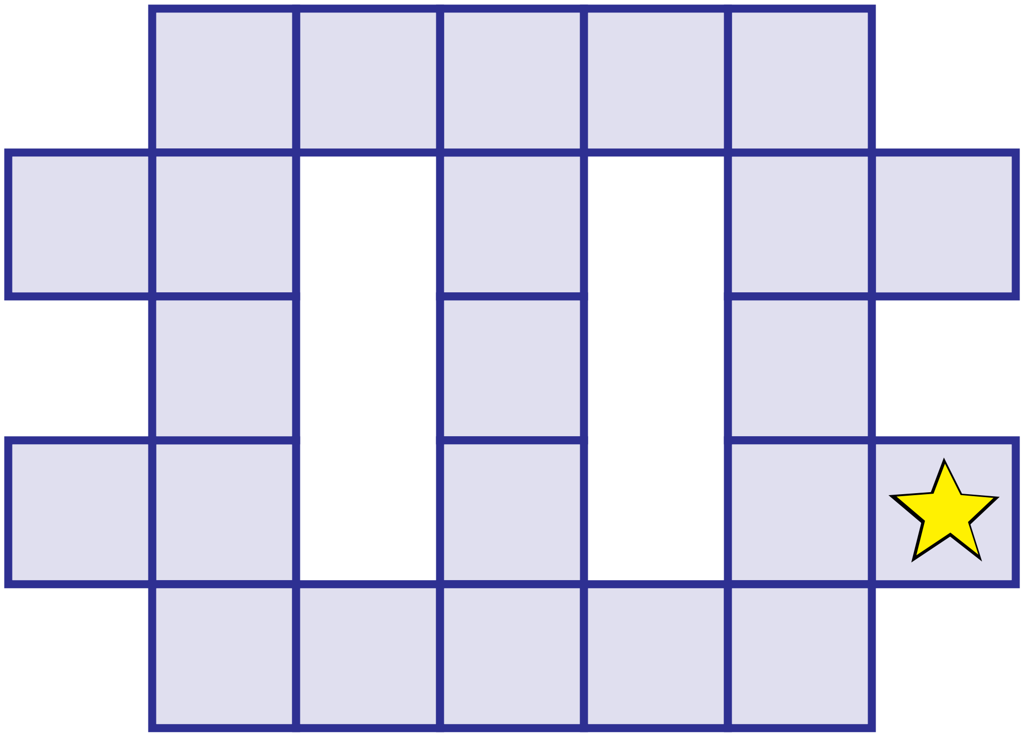
	57 States		89 States	
	Goal %	Median	Goal %	Median
Linear-Q	11.4	>251	5.7	>251
Random	46.2	>251	25.9	>251
$Q_{\mathtt{MDP}}$	47.4	>251	25.9	>251
Human	100.0	15	100.0	29

Nothing does very well.

Experimental Set-up

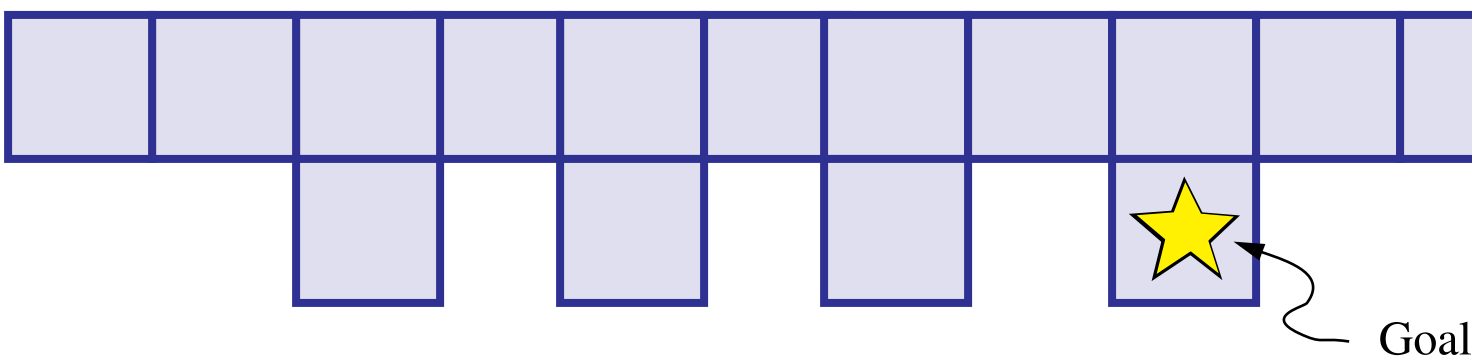
- Run algorithm to generate a policy 21 times.
- Use 75,000 updates to generate each policy.
- Evaluate each policy with 251 attempts to reach goal.
- If goal not reached within 251 steps, stop attempt.
- For each experiment, compute percentage of times goal reached and median steps to goal.
- Reported results are the median of the goal percentage and median of the median steps to goal for each experiment.

Example: 89 States



Example: 57 States

- states: robot's location and orientation (N, S, E, W)
- actions: forward, turn-left, turn-right, turn-around, no-op
- observations: binary wall detection in 4 directions + goal
- noise: in both actions and observations



Linear-Q Algorithm

```
for all a and i, randomly initialize q_i^a
initialize belief state b
loop
   perform action a, (best or random)
   get experience ( R, O )
   update belief state, b′
   update coefficients of Q^a
```

$$\Delta Q^a_i = \eta [(R + \gamma V(b')) - b \cdot Q^a] b_i$$

```
end loop
```

Reduces to Q-learning in the model, if the problem is completely observable

Learning Instead of Planning

The planning problem is too complex; learn from simulated experience.

$$\text{Model} \rightarrow \text{Experience} \rightarrow Q^a \rightarrow \text{Policy}$$

Why simulations?

- space is huge
- many points not visited
- concentrate on getting the common points correct

Simple Approach: $Q_{\mathbf{MDP}}$ method

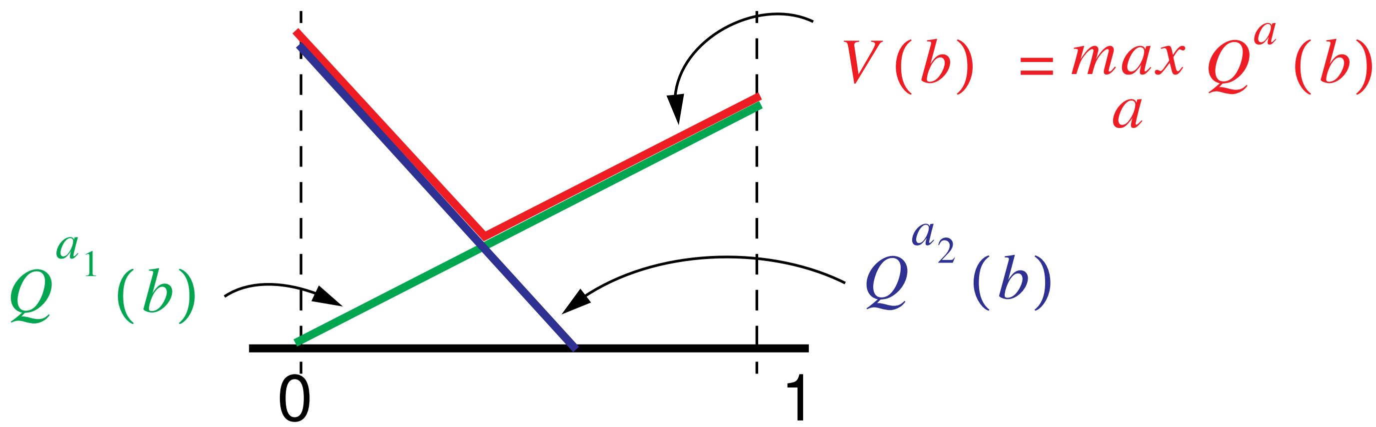
- Ignore observations and solve underlying MDP. (Recall: POMDP = MDP + observations.)
- This gives values for each action-state pair.
- Use these values for q_i^a .
- Equivalent to having uncertainty for a single step.

Advantage: relatively fast to compute.

Disadvantage: suboptimal; will not take actions to gain information.

Linear Q-Functions

We consider linear approximations to the Q-functions.



The maximization over a set of linear functions yields a convex function.

Q-Functions

$Q^a(b)$ is the value of performing a particular action, a, and acting optimally thereafter.

Value function: $V(b) = \max_a Q^a(b)$

Policy at a belief state is the action whose Q-function is largest.

Policy: $Action(b) = \arg\max_a Q^a(b)$

Best Exact Algorithms

Cannot effectively solve problems with only:

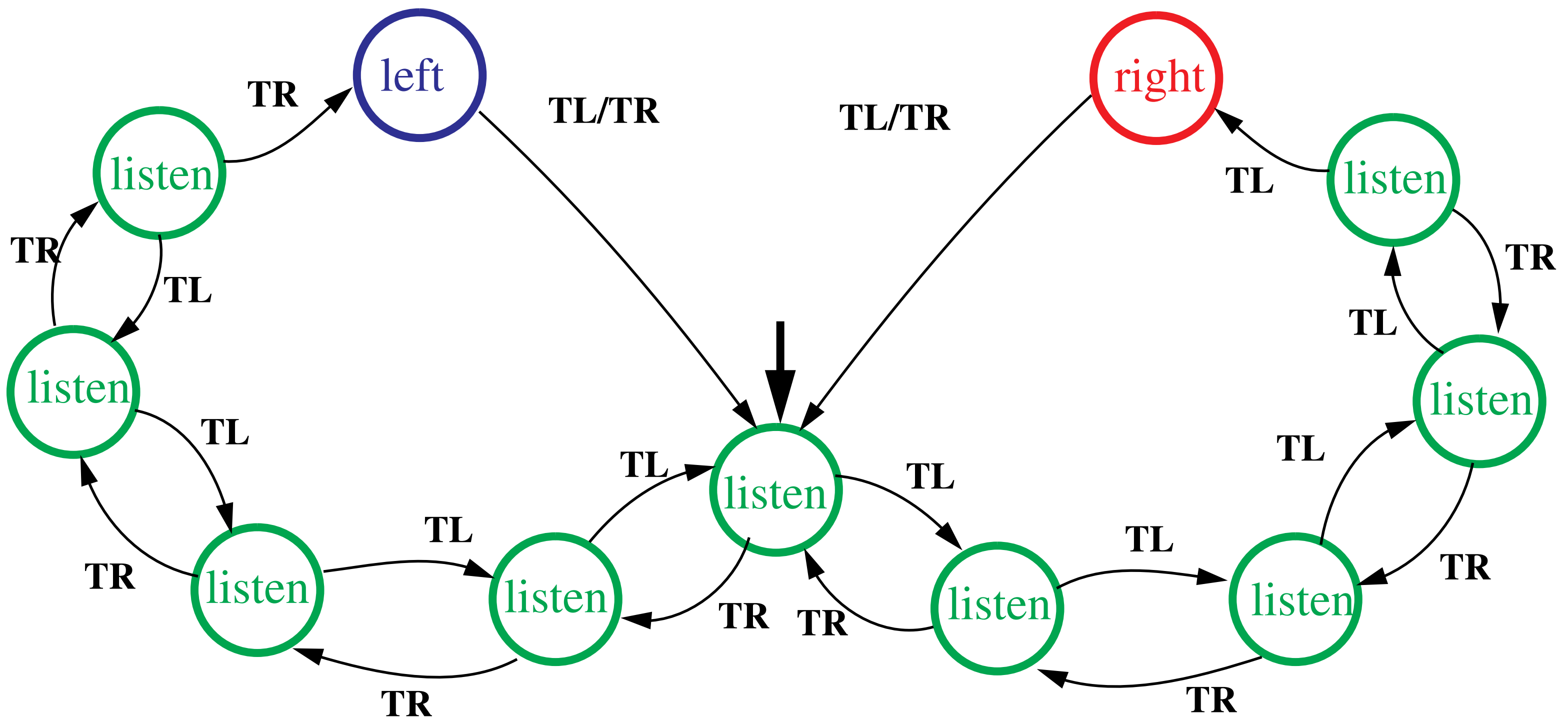
- 8 states
- 8 actions
- 8 observations

We propose an approximation scheme which will give reasonable results on problems with:

- 89 states
- 5 actions
- 17 observations

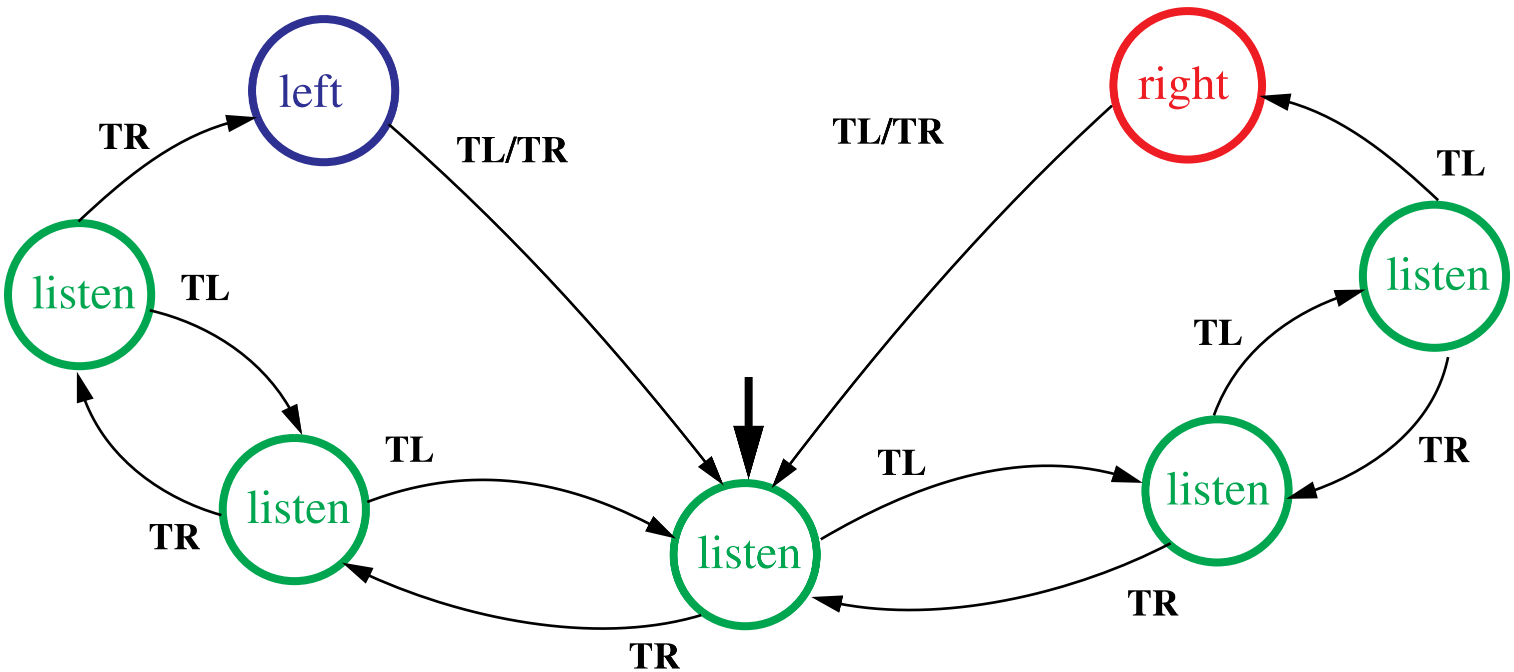
Another Tiger Variation

Reliability of listening = 0.65



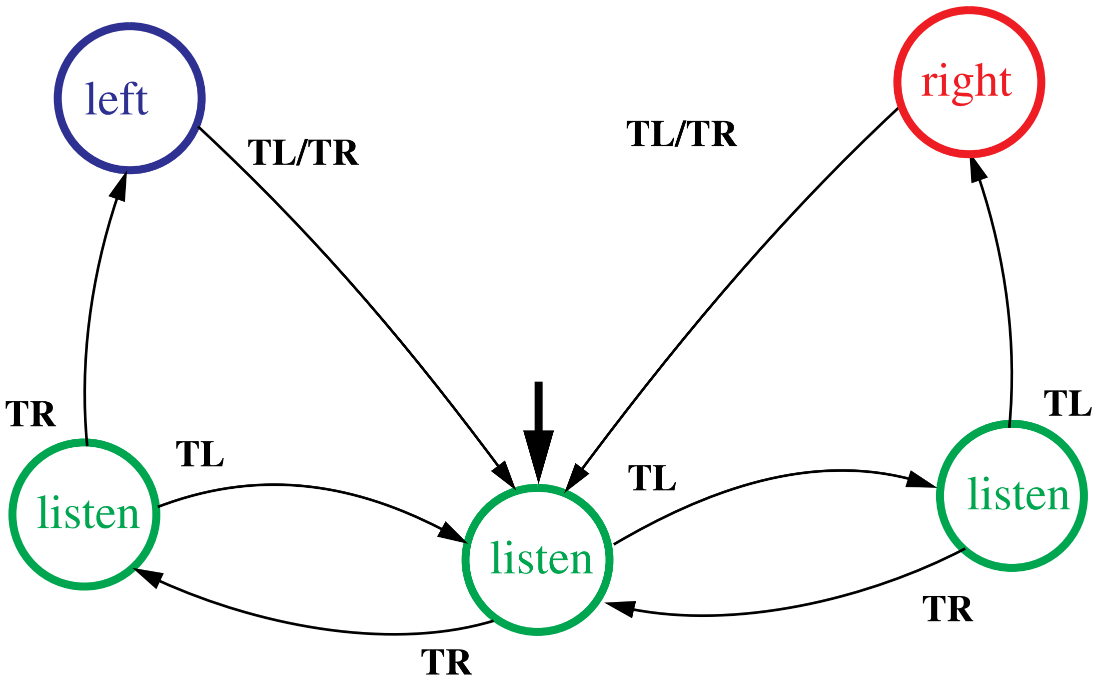
Tiger Variation

Reliability of listening = 0.75



Tiger Policy Graph

Reliability of listening = .85

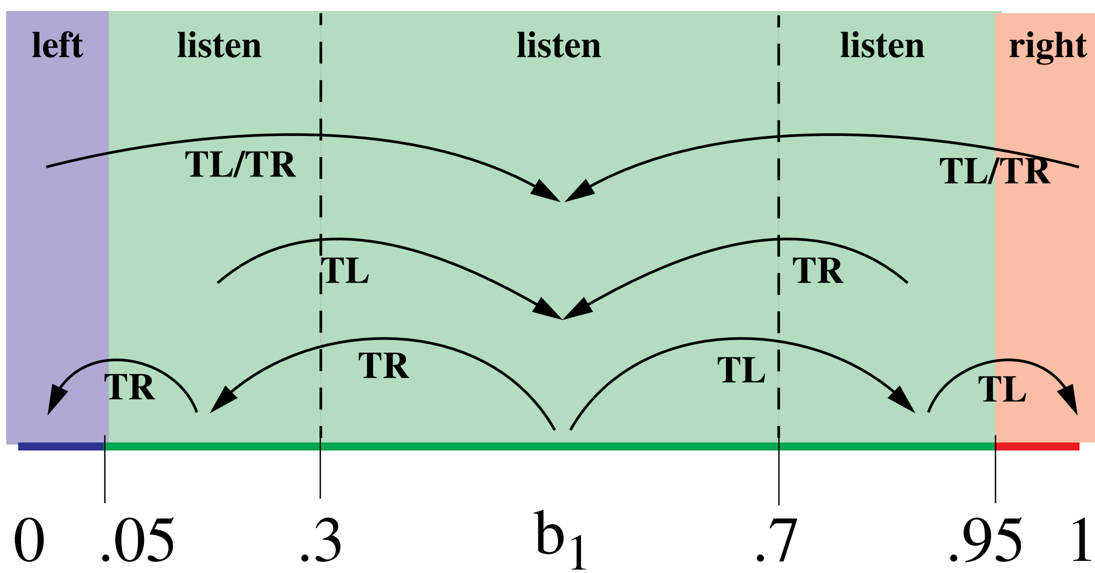


No need for belief states.

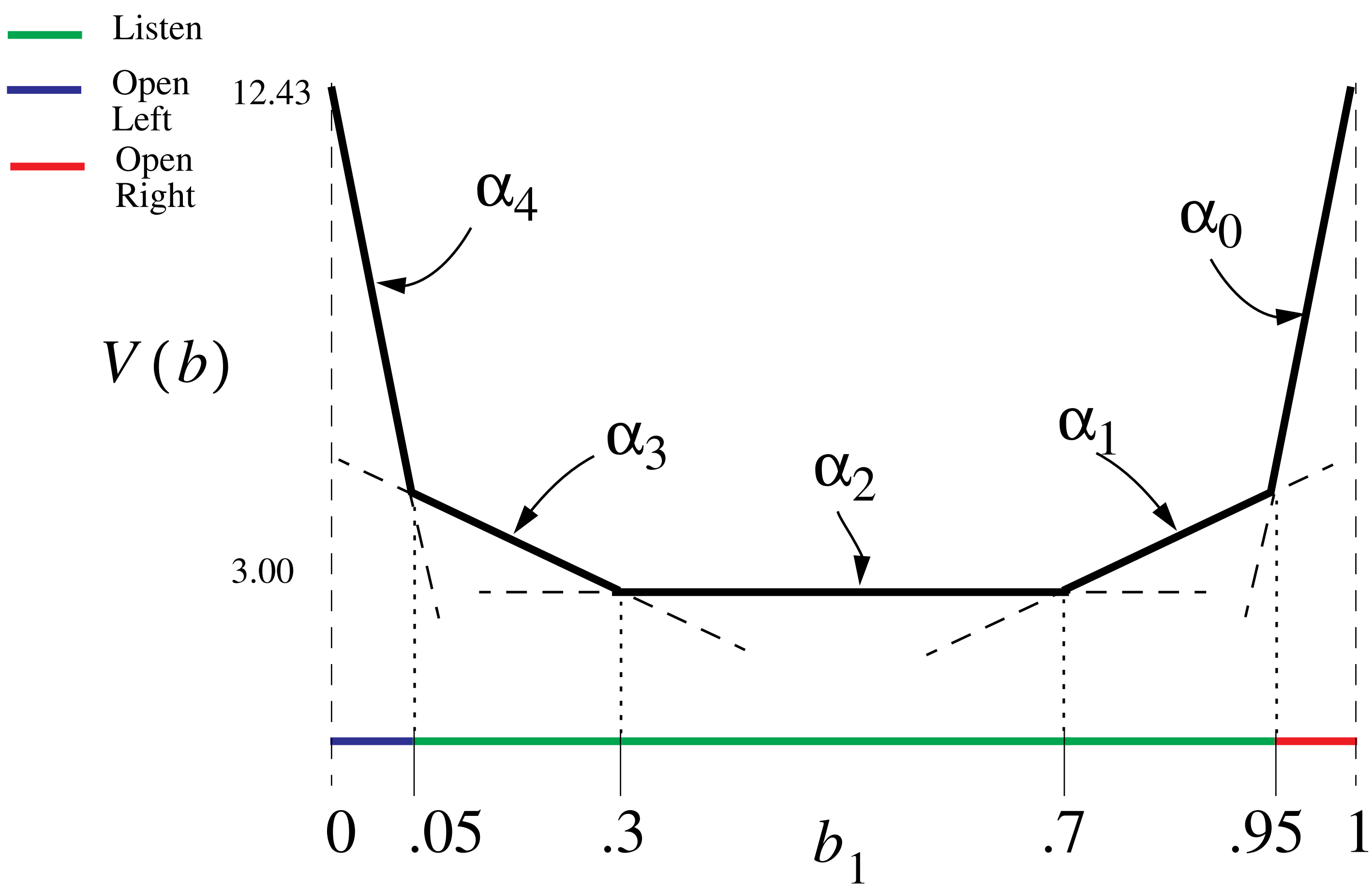
Policy Graphs

In many cases, the optimal V forms a *Markov partition* of the belief space; all b in the same region are equivalent:

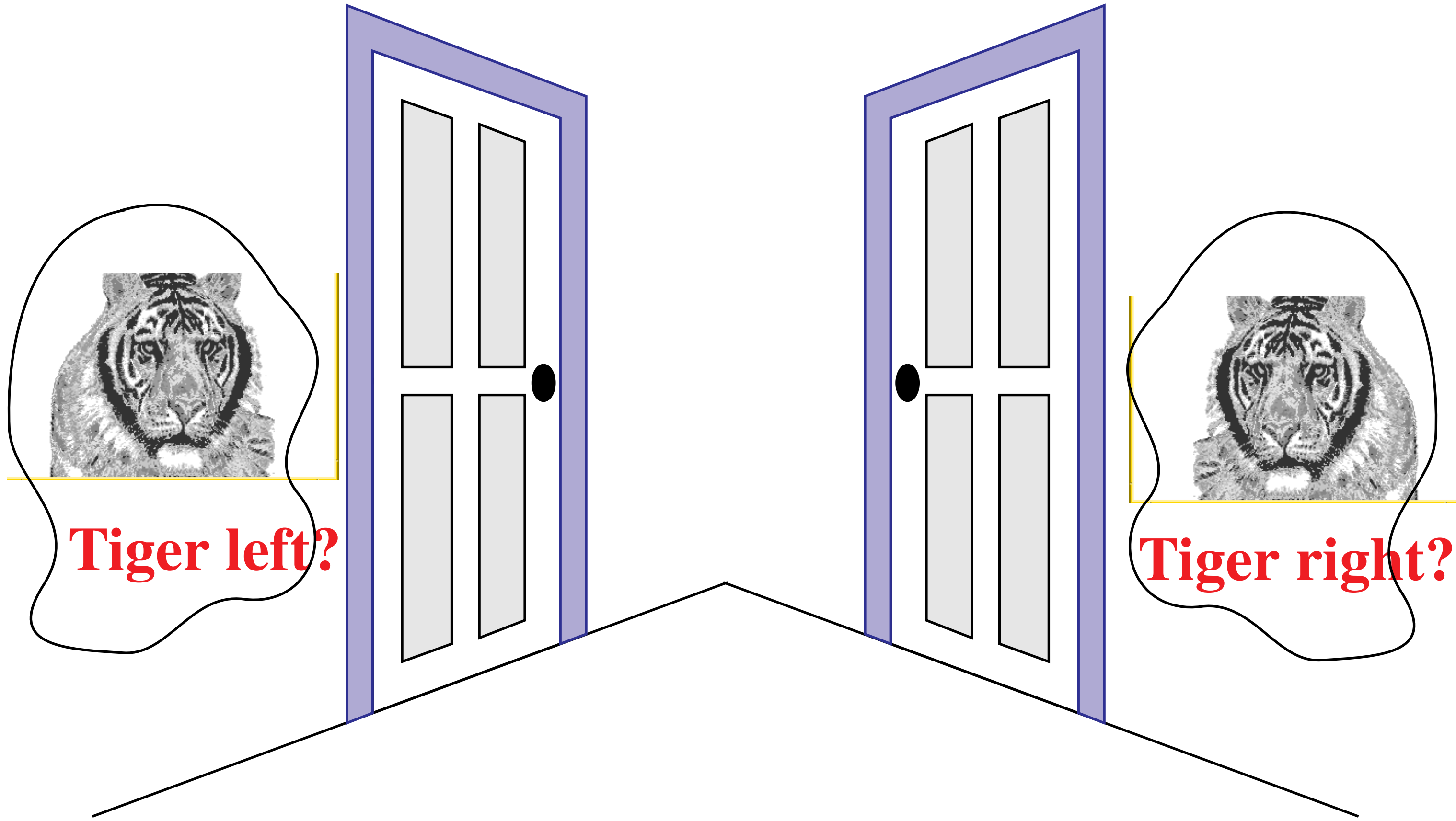
- they have the same optimal action
- they have the same next region



Tiger Value Function



Watch out for Tigers!



Finding the Value Function: Exact Methods

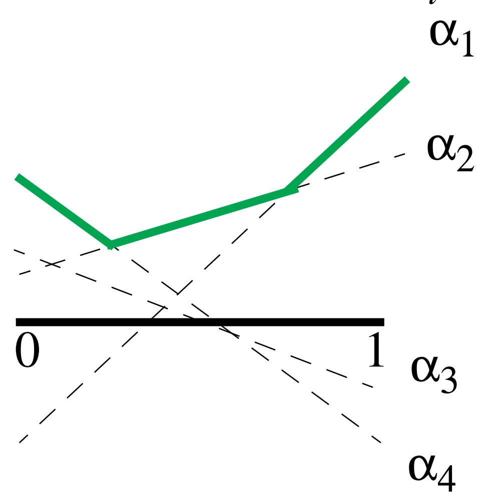
Value iteration, computing a new set of linear functions on each iteration

- single iteration can have output exponential in size of input
- even if the output is polynomial-sized, the problem is NP-complete
- we have developed the *witness* algorithm, which is polynomial in the size of the Q functions

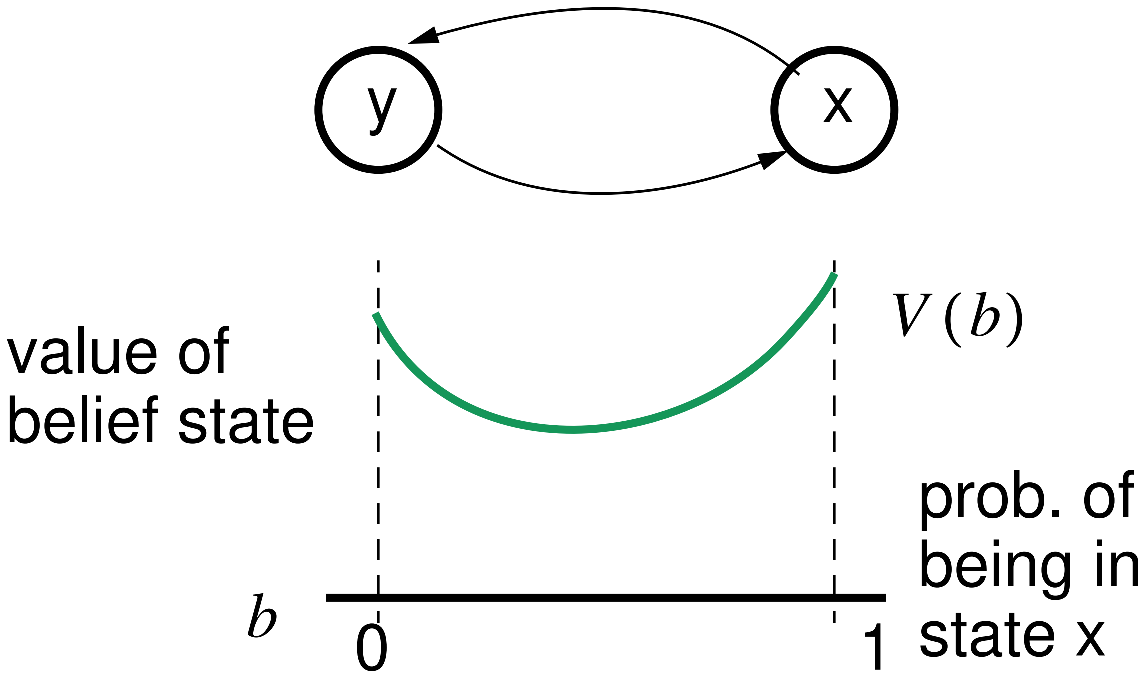
Piecewise Linearity

- finite-horizon value functions are piecewise-linear and convex
- max over a finite number of linear functions

$$V_t(b) = max_{\alpha \in V_t} \alpha \cdot b$$



Value Functions



The optimal value function is convex

Finding An Optimal Policy

Good News:

- the problem of acting given belief states is Markov

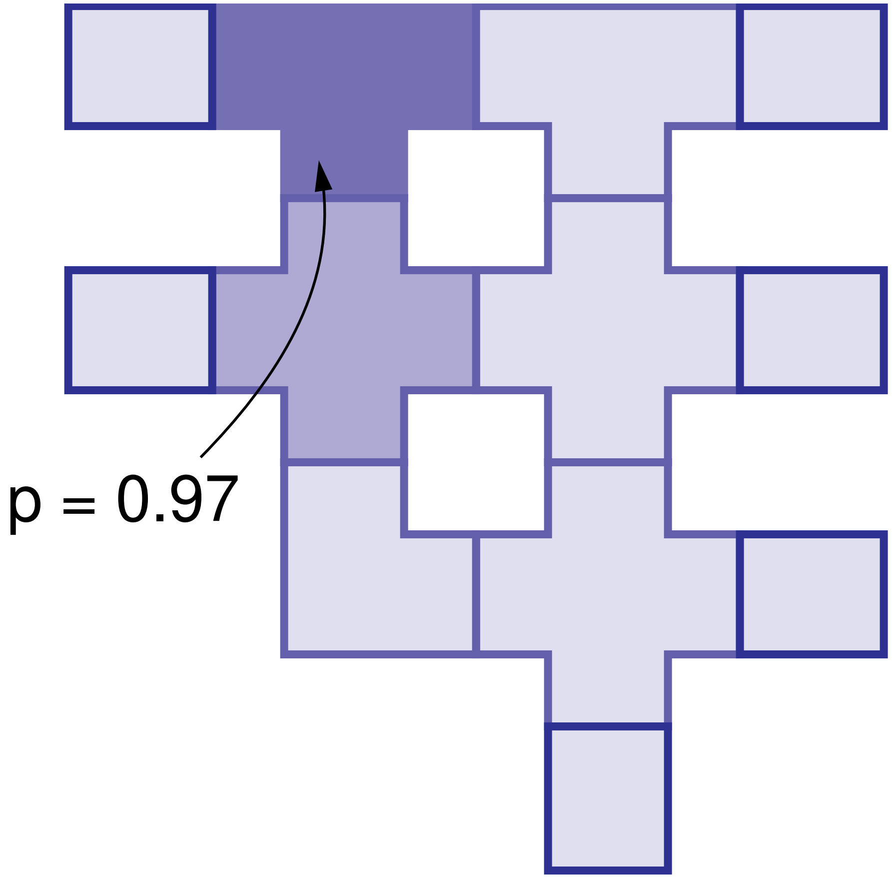
Unfortunately,

- the space of belief states is continuous

But!

- there is a finite representation of the optimal value function

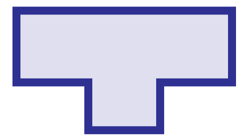
Belief State Example III



Action:

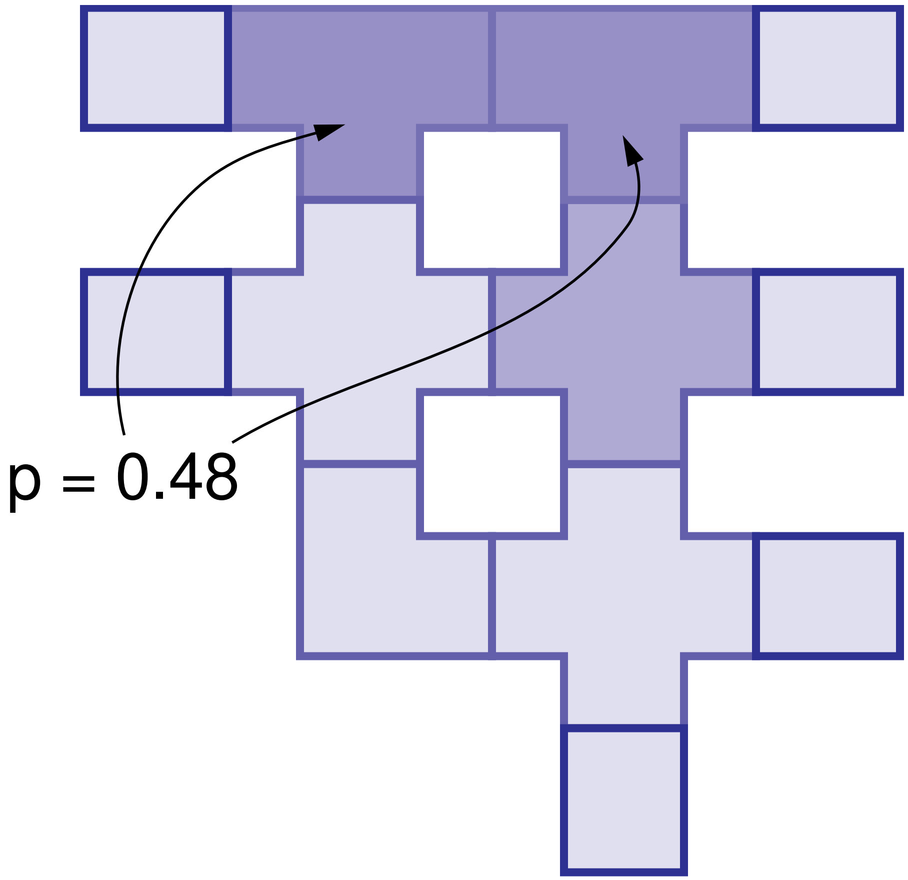
Move West

Observation:



$$b = \begin{bmatrix} 0 & 0.97 & 0.01 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

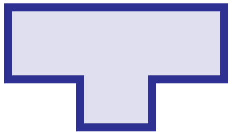
Belief State Example II



Action:

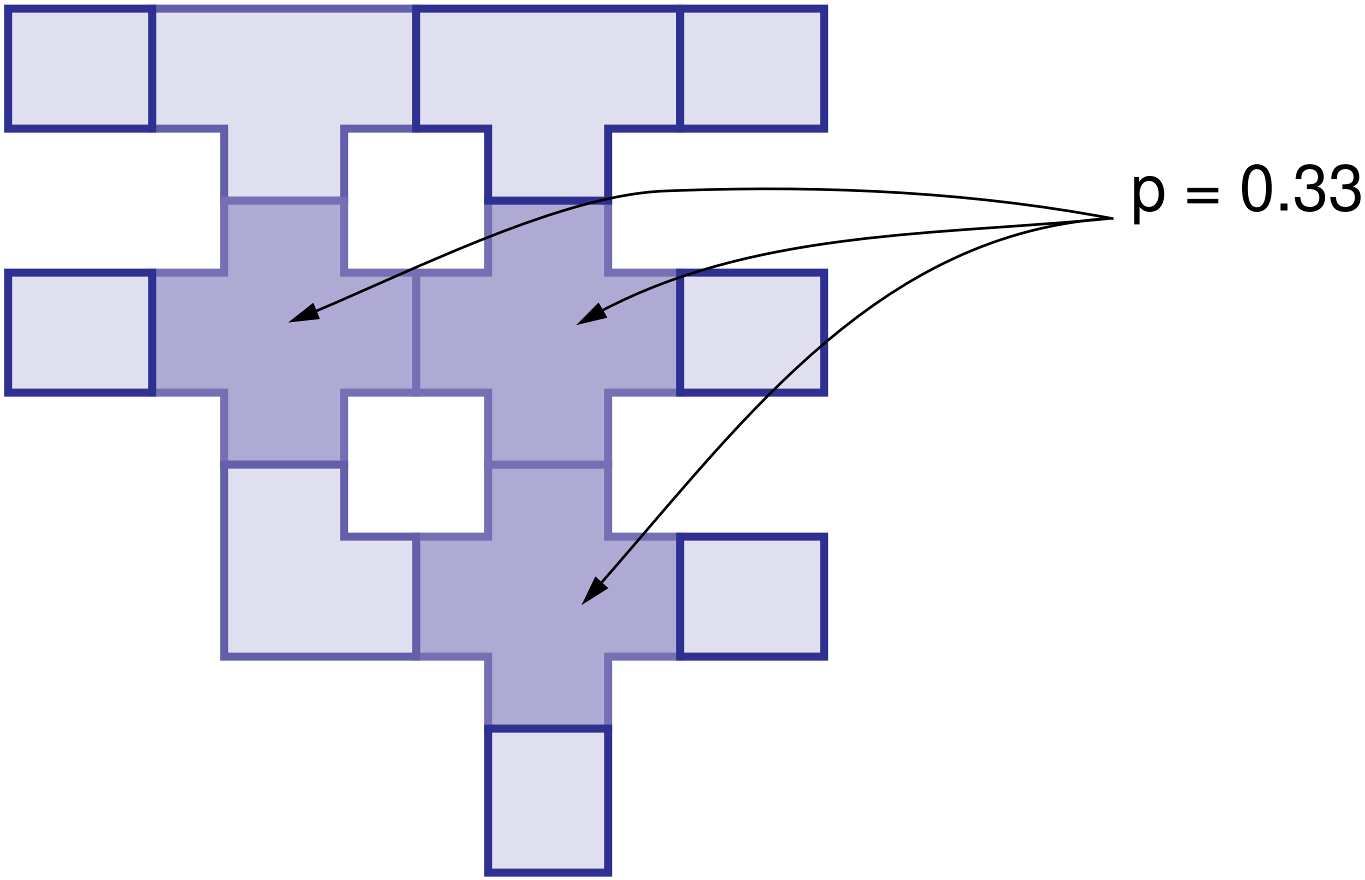
Move North

Observation:



$$b = \begin{bmatrix} 0 & 0.48 & 0.48 & 0 & 0 & 0.01 & 0.02 & 0 & 0 & 0.01 & 0 & 0 \end{bmatrix}$$

Belief State Example I



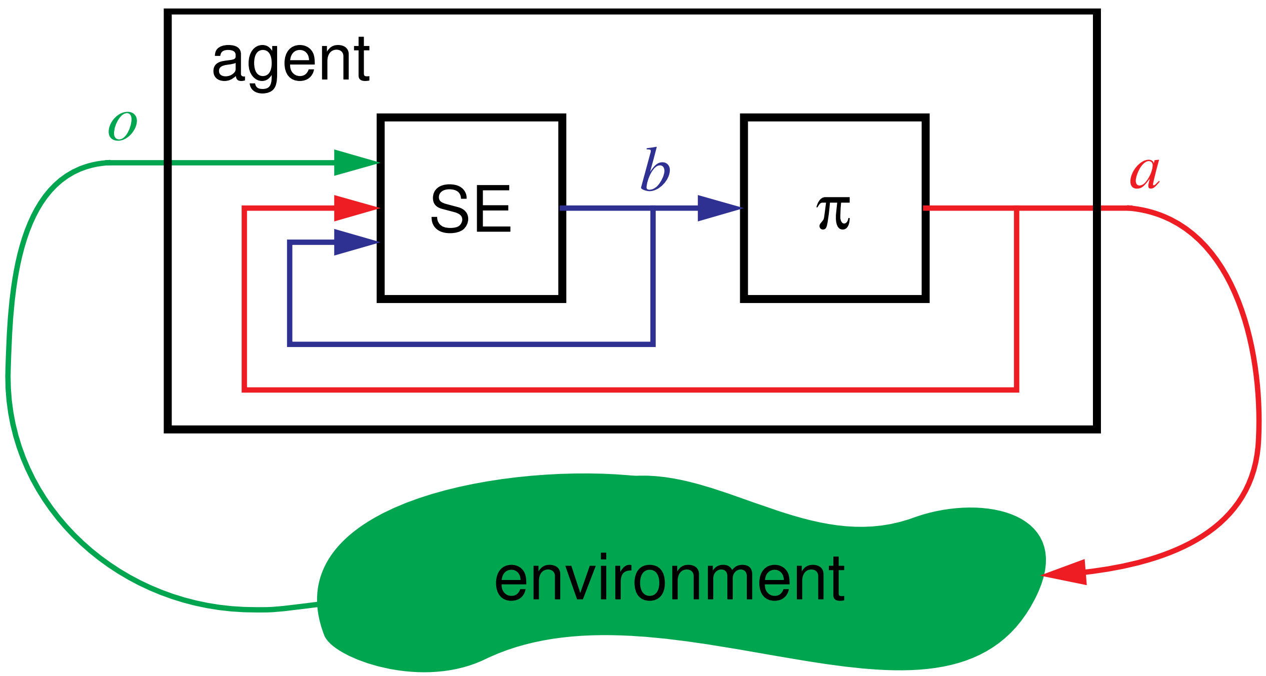
$$b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.33 & 0.33 & 0 & 0 & 0.33 & 0 & 0 \end{bmatrix}$$

Belief States

- B: probability distributions over states
- $b(s)$: probability assigned to state s
- $\mathrm{SE}(b, a, o)$: state estimator computes new belief state

$$\mathrm{SE}(b, a, o)\ (s') = \frac{\sum_{s \in S, o \in \Omega} b(s)\, T(s, a, s')\, O(s', a, o)}{\Pr(o \mid a, b)}$$

POMDP Architecture



Partially Observable MDPs

Assume that there is an underlying Markov Decision Process that the agent can stochastically observe:

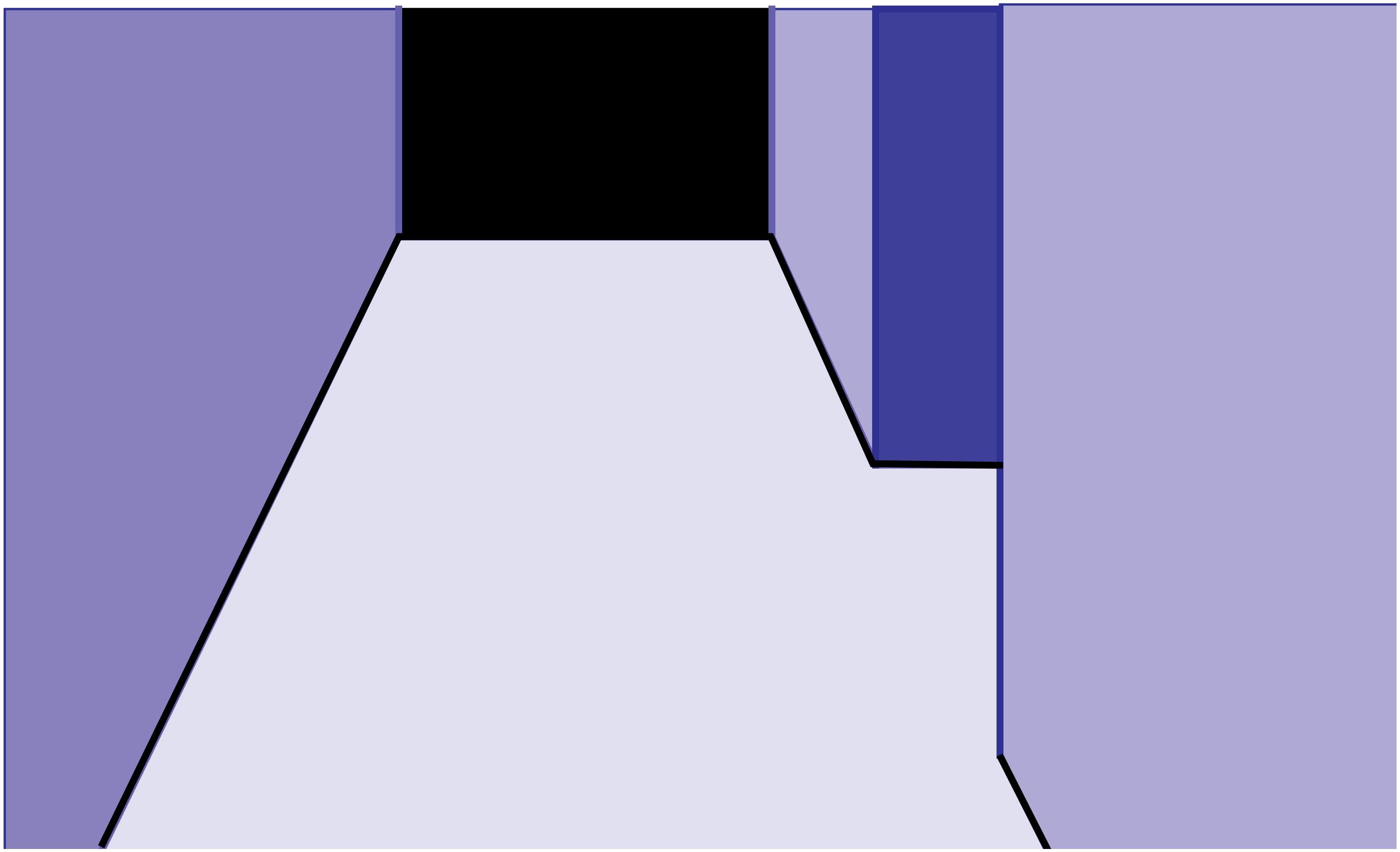
- S: finite set of world states
- A: finite set of actions
- Ω: finite set of observations
- $R(s, a)$: expected reward for action a in state s
- $T(s, a, s')$: probability of ending in state s' after taking action a in state s
- $O(s', a, o)$: probability of observing o after taking action a and ending in state s'

Dealing with Partial Observability

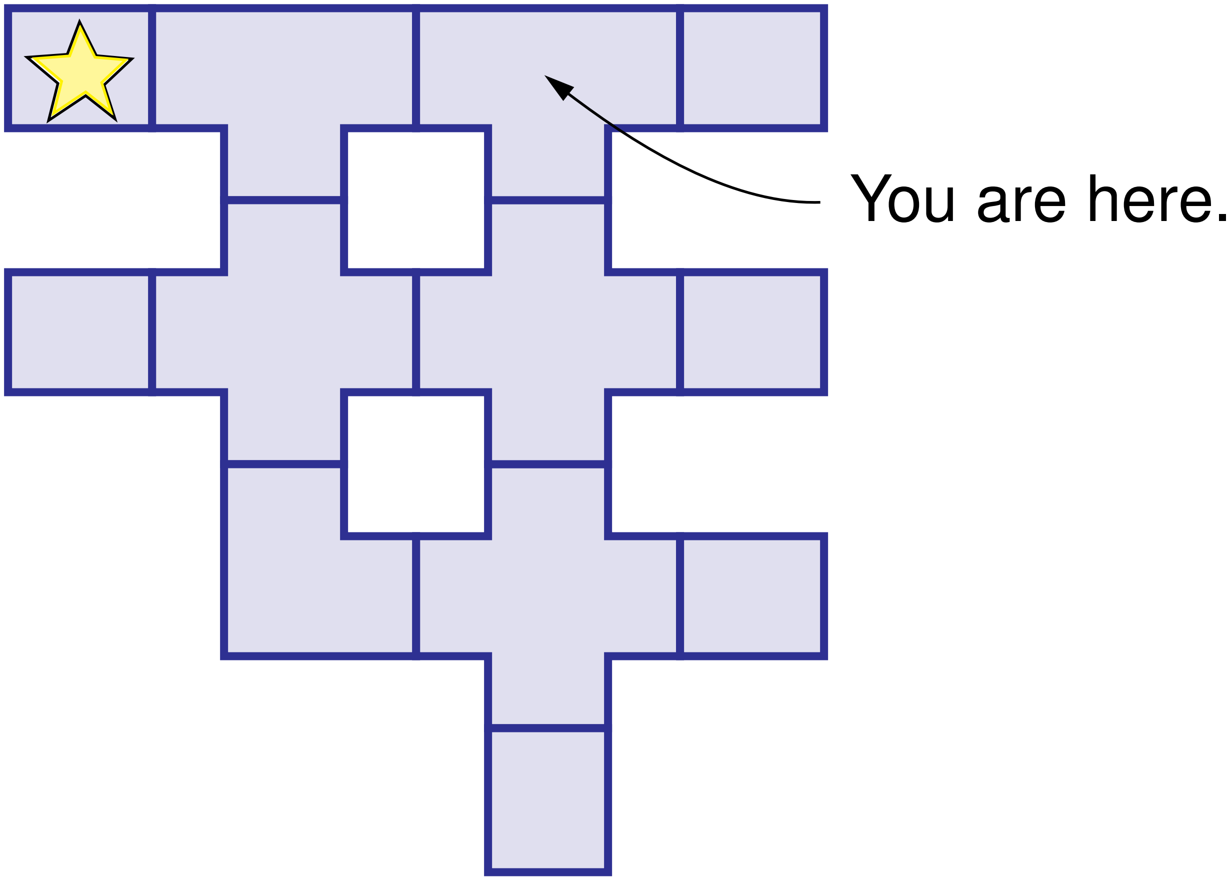
What should we do when the agent can't completely observe the state of the world?

- use observations as states
  - standard algorithms for finding optimal policies won't work
  - policies are often extremely bad
- use observations as states and try to find a stochastic state-free policy
- use something else as states; find a policy with internal state

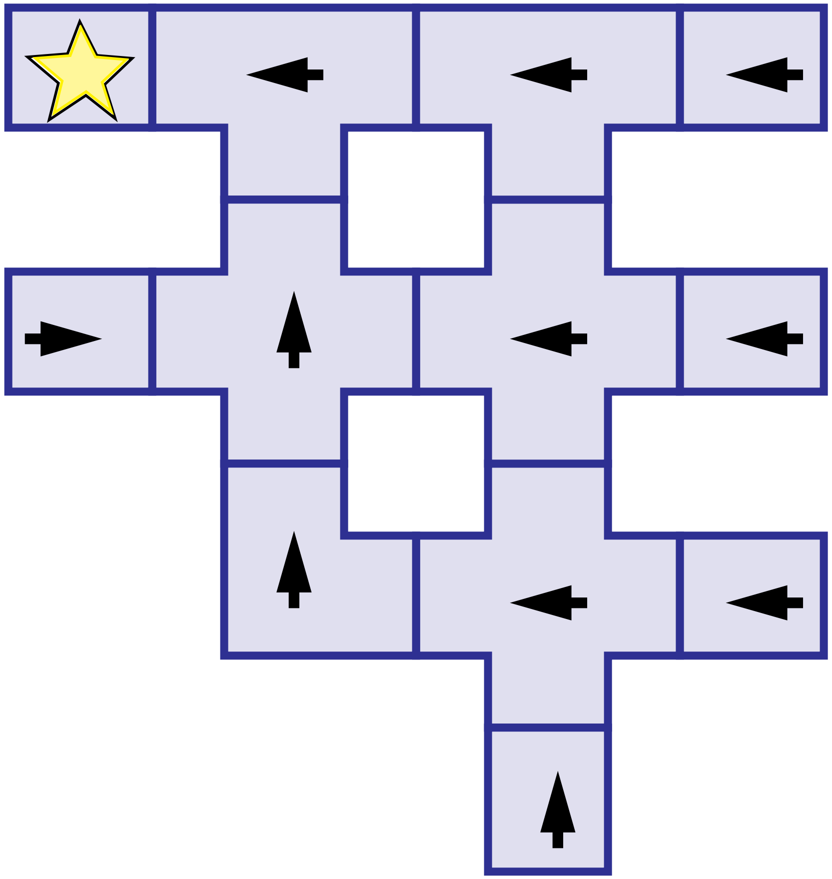
Real Observations



Ideal Observations



Sample Policy



Finding an Optimal Policy

Value iteration algorithm:

```
initialize V_0(s) arbitrarily
for t = 1 to ...
   loop for s ∈ S
```

$$V_t(s) := \max_a \left[R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_{t-1}(s') \right]$$

```
   end loop
end loop
```

An approximately optimal value function yields an arbitrarily good approximately optimal policy:

$$\left| V_t - V_{t-1} \right|_{\sup} \leq \varepsilon \Rightarrow \left| V^* - V_{\pi_t} \right|_{\sup} \leq 2\varepsilon \frac{\gamma}{1-\gamma}$$

The Value Function

Every policy has a value function:

$$V_\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_\pi(s')$$

There is a unique optimal value function:

$$V^*(s) = \max_a \left[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right]$$

Derive an optimal policy from the optimal value function:

$$\pi^*(s) = \operatorname{argmax}_a \left[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right]$$

Reward Functions and Goals

Can encode most goals as reward functions:

$\text{asap}(Q)$: enter a state satisfying Q as soon as possible

- if $Q(s)$ then $R(s) = 0$; otherwise $R(s) = -1$
- make all states such that $Q(s)$ absorbing

$\text{alap}(Q)$: stay in states satisfying Q as long as possible

- if $Q(s)$ then $R(s) = 0$; otherwise $R(s) = -1$
- make all states such that $\neg Q(s)$ absorbing

Use infinite horizon discounted model.

Many other goals possible in this rich language.

Acting Optimally

A policy $\pi : S \rightarrow A$ chooses actions as a function of world states.

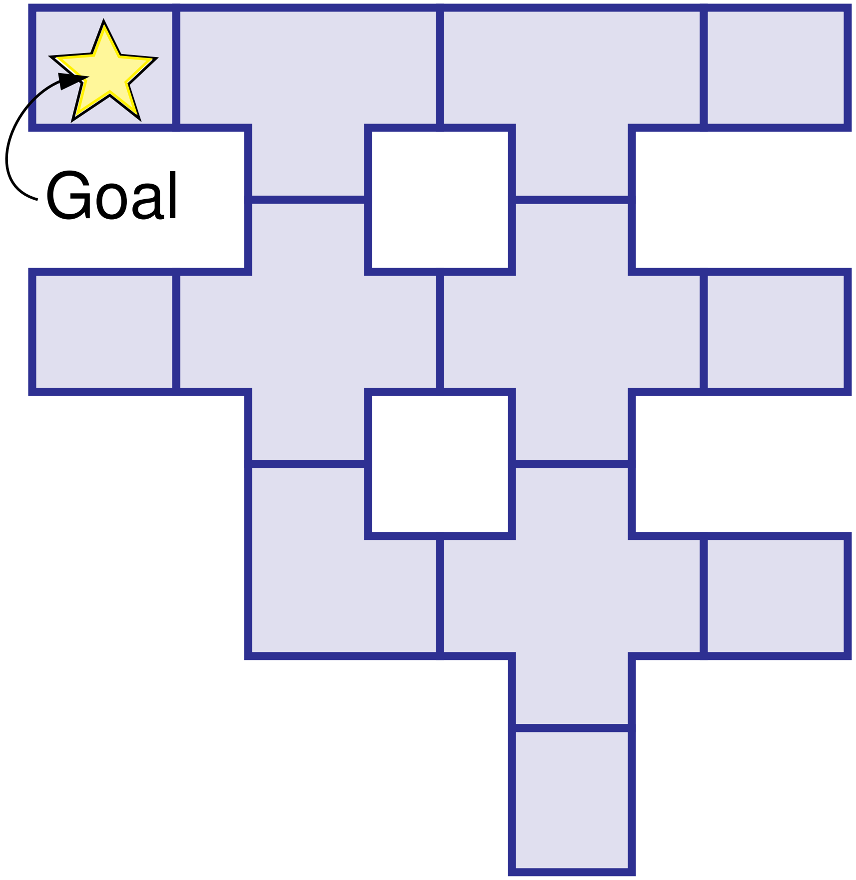
Find the policy that maximizes expected discounted future reward:

$$\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$$

Markov Decision Processes

- S: finite set of world states
- A: finite set of actions
- $R(s, a)$: expected reward for action a in state s
- $T(s, a, s')$: probability of being in state s' immediately after taking action a in state s

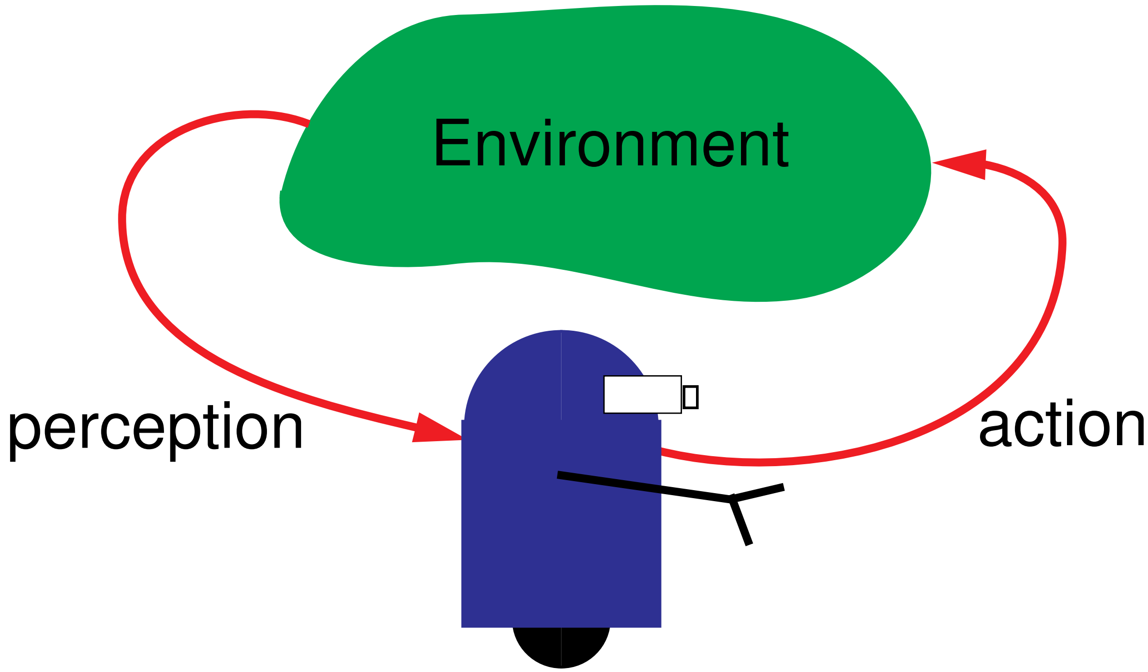
Sample Environment



Finite number of:

- locations
- actions

Agents Interact with an Environment



How should an agent act when

- effects of actions are stochastic?
- many world states look the same?
- perceptions are stochastic?

Markov decision processes enable planning with stochastic actions.

Partially observable Markov decision processes enable planning with stochastic actions and observations.

Partially Observable Markov Decision Processes for Artificial Intelligence

Leslie Pack Kaelbling
Computer Science Department
Brown University

Joint work with
Michael L. Littman and Anthony R. Cassandra