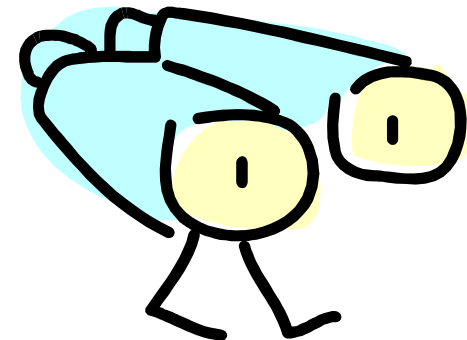

Introduction to Partially Observable Markov Decision Processes (POMDPs)

Guy Shani
Machine Learning and Applied Statistics
Microsoft Research

Overview

- Agenda:
 - Introduce POMDP models and notations.
- Structure:
 - MDPs
 - POMDPs.
 - Exact Value Iteration

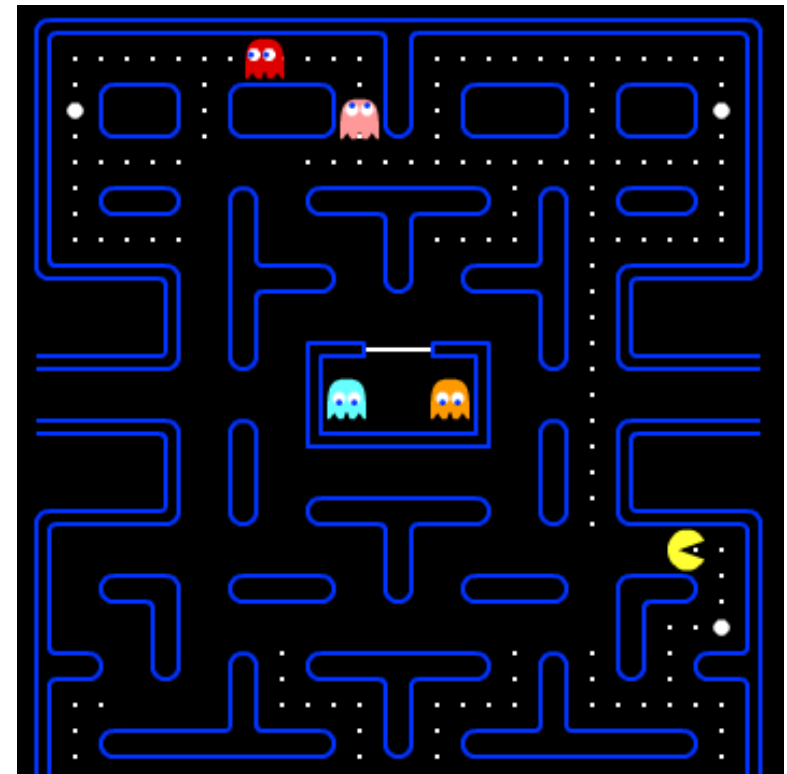


Control under Uncertainty

- Factory management
 - Scheduling item manufacture
 - Assigning jobs to machines
- Assistive agents
 - Helping people perform tasks
- Robotics
 - Mars Rover
- Dialog systems
 - Asking questions to gather information

Markov Decision Process - MDP

- Model agents in a stochastic environment.
- State – an encapsulation of all the relevant environment information:
 - Agent location
 - Can the agent eat monsters?
 - Monsters location
 - Gold coins location
- Action – affect the environment:
 - Moving up, down, left, right
- Stochastic effects –
 - Movement can sometime fail
 - Monster movements are random
- Reward – received for achieving goals
 - Collecting coins
 - Eating a monster



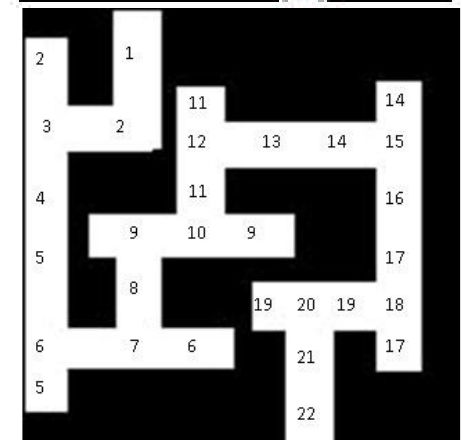
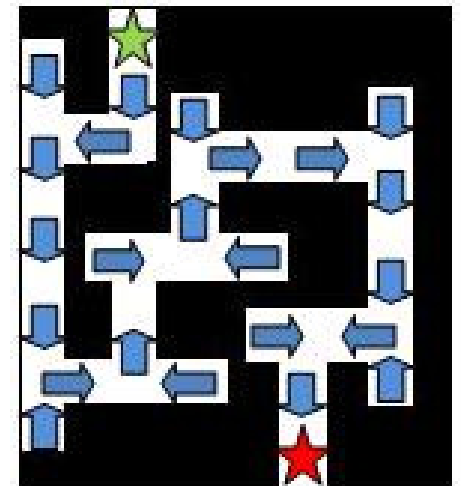
MDP Formal Definition

- Markov property – action effects depend only on the current state.
- MDP is defined by the tuple $\langle S, A, tr, R \rangle$.
- S – state space
- A – action set
- tr – state transition function: $tr(s, a, s') = pr(s' | s, a)$
- R – reward function: $R(s, a)$

Policies and Value Functions

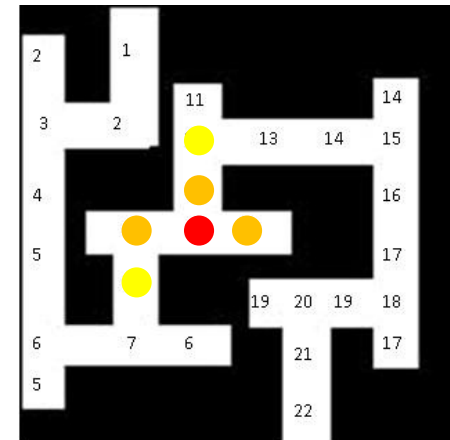
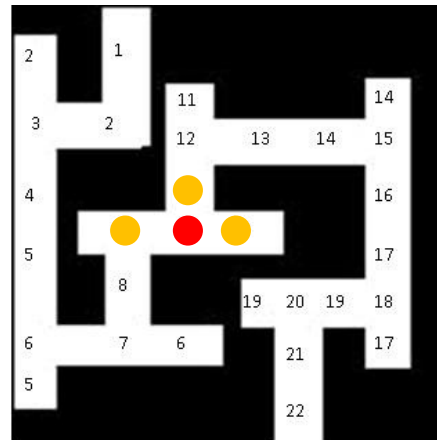
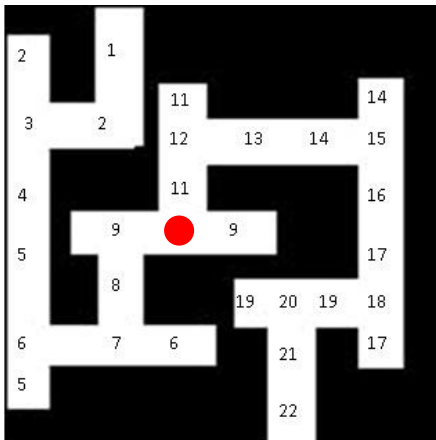
- Policy – specifies an action for each state.
- Optimal policy – maximizes the collected rewards:
 - Sum: $\sum_{t=0}^{\infty} r_t$
 - Average: $\frac{1}{T} \sum_{t=0}^T r_t$
 - Discounted sum: $\sum_{t=0}^{\infty} \gamma^t r_t$
- Value function – assigns a value to a state

$$\pi_V(s) = \arg \max_a R(s, a) + \gamma \sum_{s'} tr(s, a, s') V(s')$$



Value Iteration (Bellman 1957)

- Dynamic programming method.
- Value is updated from reward states backwards.
- Update is known as a backup.



Value Iteration (Bellman 1957)

Initialize – $V_0(s) = 0, n = 0$

While V has not converged

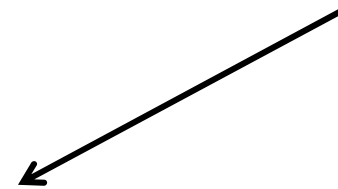
For each s

$$V_{n+1}(s) = \max_a R(s, a) + \gamma \sum_{s'} tr(s, a, s') V_n(s')$$

$n = n + 1$

- Known to converge to V^* - the optimal value function.
- π^* - the optimal policy corresponds to the optimal value function.

Bellman update

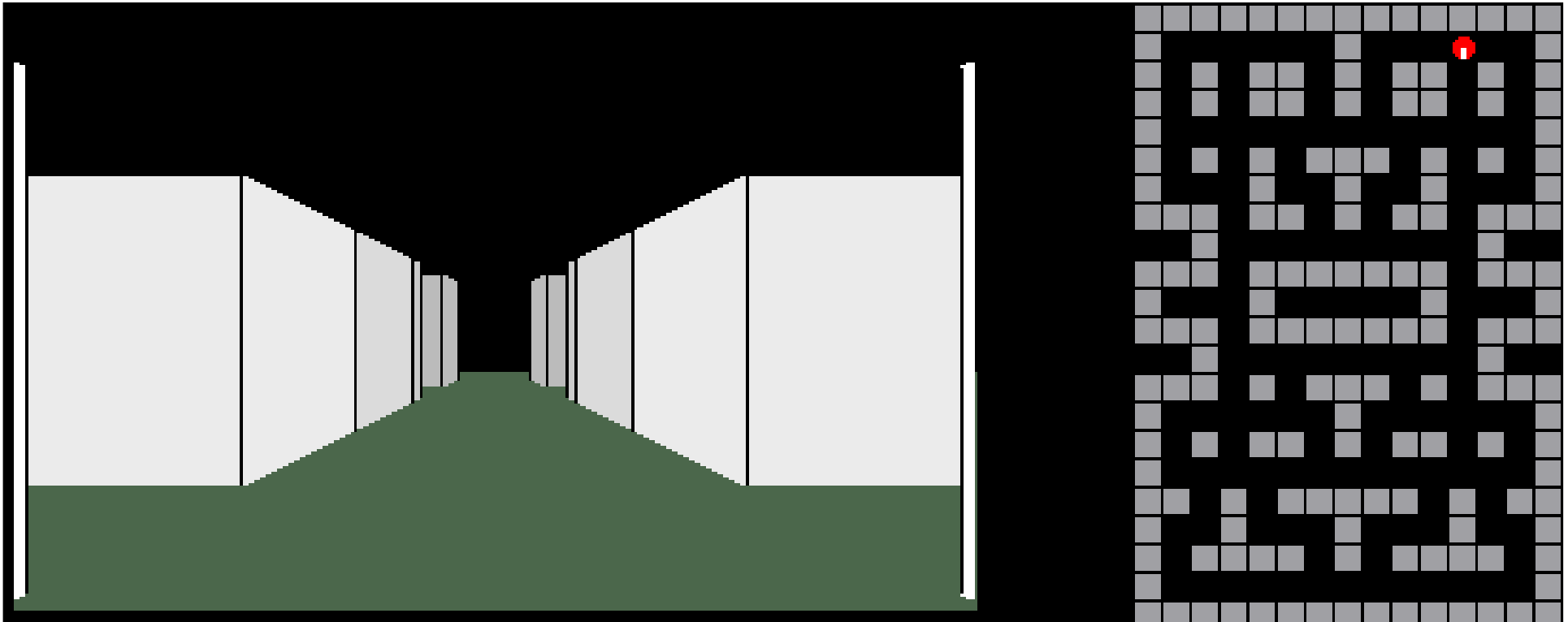


Policy Iteration (Howard 1960)

- Intuition – we care about policies, not about value functions.
- Changes in the value function may not affect the policy.
- Expectation-Maximization.
- Expectation – fix the policy and compute its value.
- Maximization – change the policy to maximize the values.

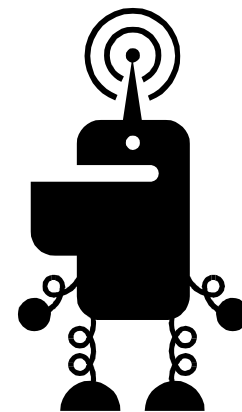
Partial Observability

- Real agents cannot directly observe the state.
- Sensors – provide partial and noisy information about the world.



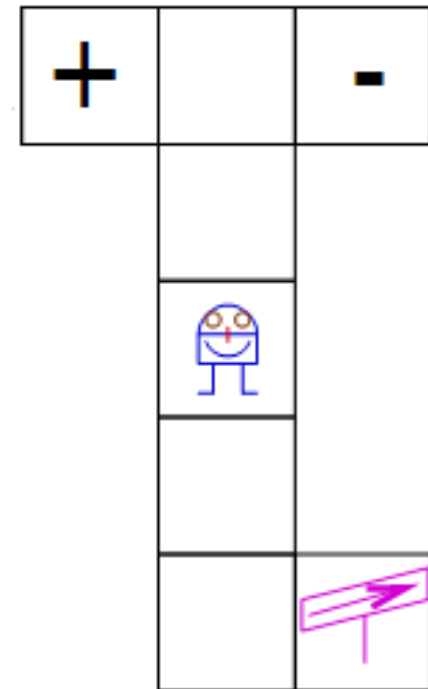
Partially Observable MDP - POMDP

- The environment is Markovian.
- The agent cannot directly view the state.
- Sensors give observations over the current state.
- Formal POMDP model:
 - $\langle S, A, tr, R \rangle$ – an MDP (the environment)
 - Ω – set of possible observations
 - $O(a,s,o)$ – observation probability given action and state – $pr(o|a,s)$.



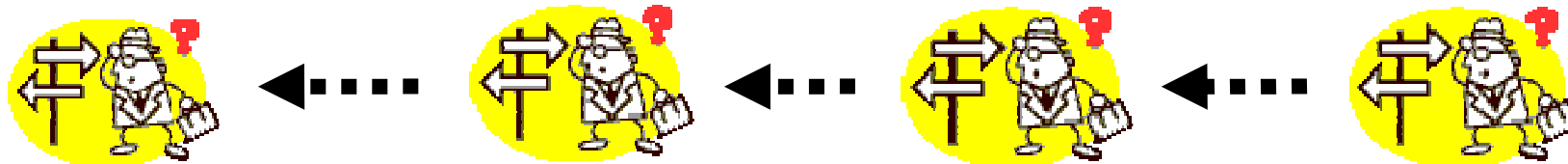
Value of Information

- POMDPs capture the value of information.
- Example – we don't know where the larger reward is – should we go and read the map?
- Answer – it depends on:
 - The difference between the rewards.
 - The cost of reading the map.
 - The accuracy of the map.
- POMDPs take all such considerations into account and provide an optimal policy.



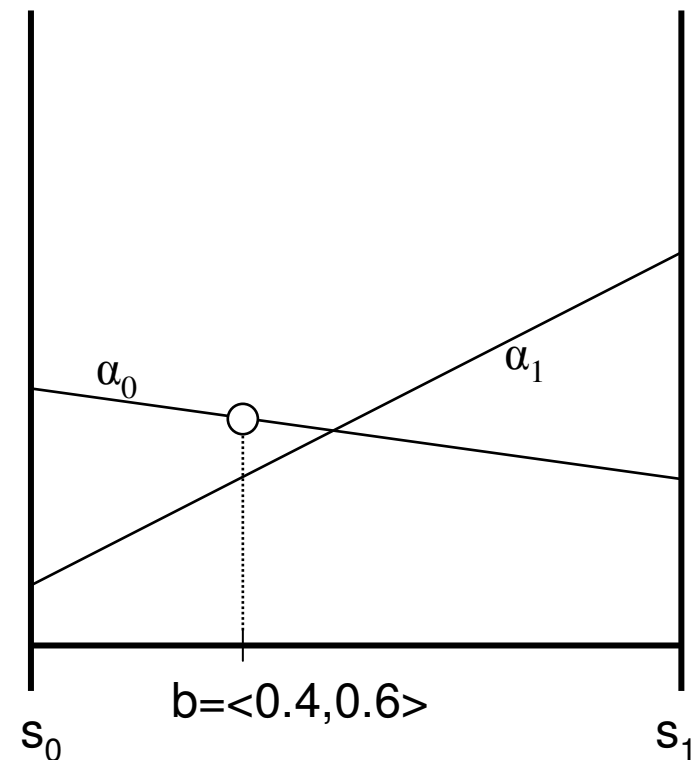
Belief States

- The agent does not directly observe the environment state.
- Due to noisy and insufficient information, the agent has a belief over the current world state.
- $b(s)$ is the probability of being at state s .
- $\tau(b, a, o)$ - a deterministic function computing the next belief state given action a and observation o .
- The agent knows its initial belief state – b_0



Value Function (Sondik 1973)

- A value function V assigns a value to a belief state b .
- V^* - the optimal value function.
- $V^*(b)$ – the expected reward if the agent will behave optimally starting from belief state b .
- V is traditionally represented as a set of α -vectors.
- $V(b) = \max_{\alpha} \alpha \cdot b$ (upper envelope).
- $\alpha \cdot b = \sum_s \alpha(s) b(s)$



Exact Value Iteration

$$S' = \bigcup_a S^a$$

$$S^a = \bigoplus_o S_o^a$$

$$S_o^a = \{g_{a,o}^\alpha : \alpha \in S\}$$

$$g_{a,o}^\alpha(s) = r_a(s) + \gamma \sum_{s'} tr(s, a, s') O(a, s', o) \alpha(s')$$

- Creates a new set of α -vectors.
- Exponential explosion of vectors.
- Dominated vectors can be pruned. (Littman et al. 1997)
- Pruning process is time consuming.

Summary

- We explained the basic concepts.
- Now to something more interesting...