Finish Markov Decision Processes

Last Class

Computer Science cpsc322, Lecture 37

(Textbook Chpt 9.5)

April, 8, 2009

Lecture Overview

- **Recap: MDPs and** More on MDP Example
- Optimal Policies
  - Some Examples
- Course Conclusions

Planning under Uncertainty

Single Stage and Sequential Decisions

Primary Application "**Decision Support Systems**" e.g.,

- **Medicine:** Help doctor/patient to select test(s)/therapy
- **Finance:** Help Venture Capitalist in investment decisions

Decision Processes

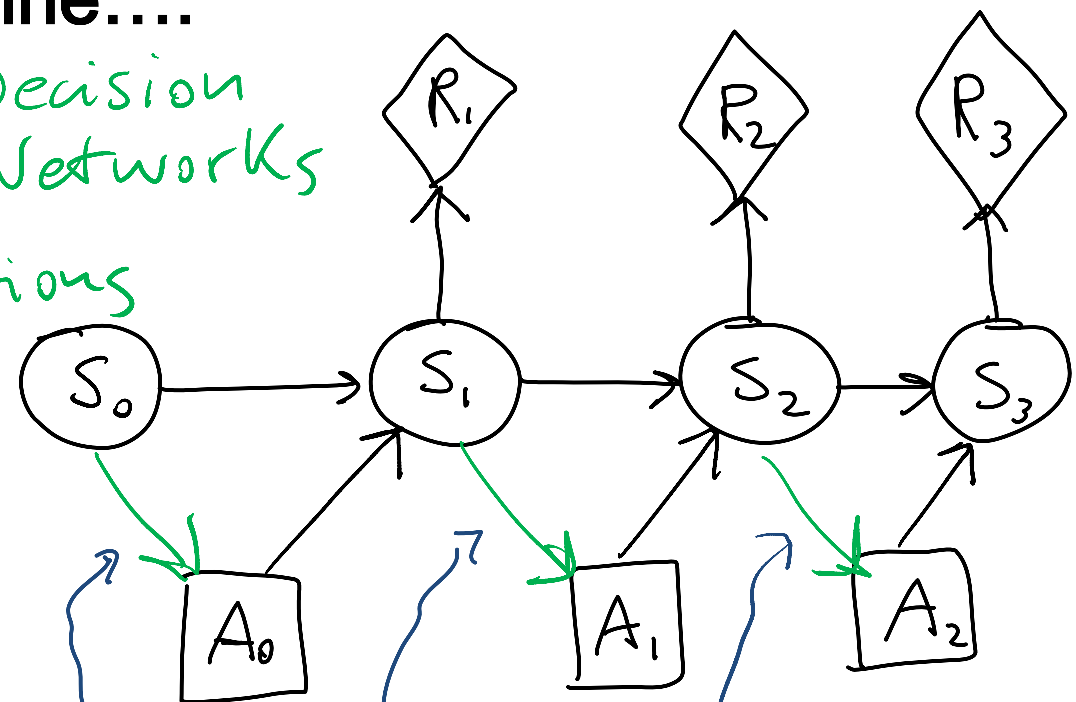
Primary Application " **(Semi-)Autonomous Agents**"

- **Robots:** on a planet, in a volcano, under the Ocean
- System **helping** older adults with cognitive disabilities
- System **monitoring** nuclear Plant
- **Control and coordination** of unmanned aerial vehicles

Decision Processes: MDPs

To manage an **ongoing (indefinite… infinite) decision process, we combine….**

Markov Chains & Decision Networks



Markovian

Stationary

Assumptions

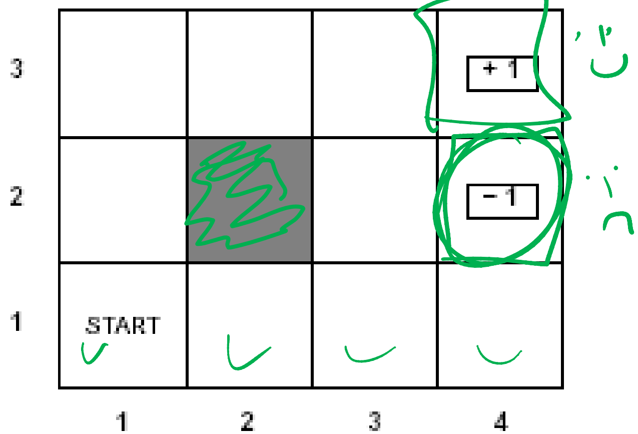
Utility not just at the end

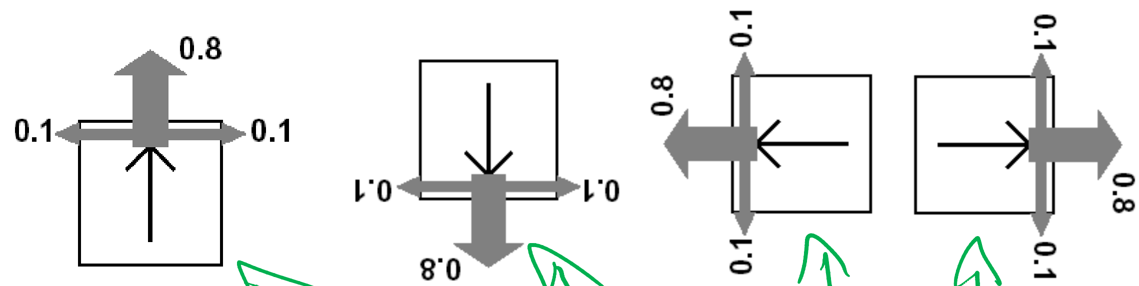
BUT

Sequence of rewards

Fully Observable

Example MDP: Scenario and Actions





Agent moves in the above grid via actions *Up, Down, Left, Right*

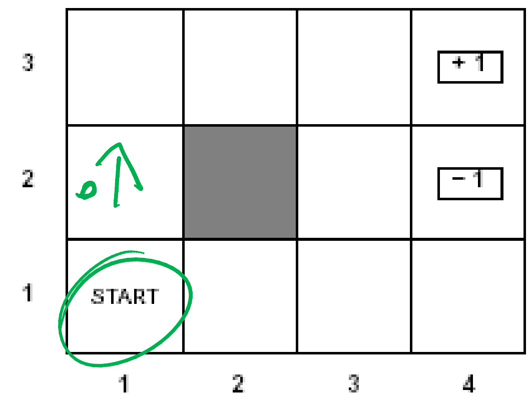
Each action has:

- 0.8 probability to reach its intended effect
- 0.1 probability to move at right angles of the intended direction
- If the agents bumps into a wall, it says there

Eleven states ((3,4) and (2,4) are terminal states)

$$R(s) = \begin{cases} -0.04 & \text{(small penalty) for nonterminal states} \\ \pm 1 & \text{for terminal states} \end{cases}$$

Example MDP: Underlying info structures



Four actions *Up, Down, Left, Right*

Eleven States: {(1,1), (1,2)…… (3,4)}

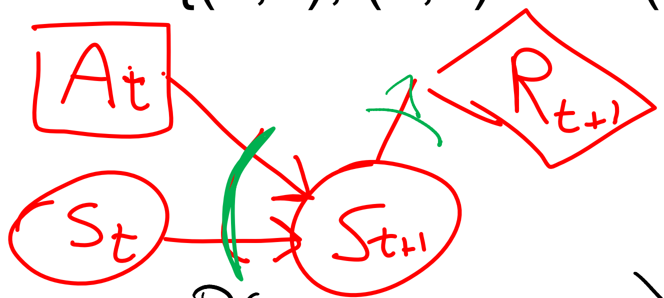


Table $4 \times 11 \times 11$ $P(S_{t+1} \mid S_t, A_t)$

Up

	1,1	2,1	1,2	3,1 …
1,1	.1	.8	.1	0000
1,2	0	.2	0	.8
⋮				

Down

L

R

$P(S_0)$

1,1	1
⋮	0
	0
	0
	0
	0
	0

$R(s)$

11	-.04
⋮	⋮
(2 4)	-1
(3 4)	+1

Lecture Overview

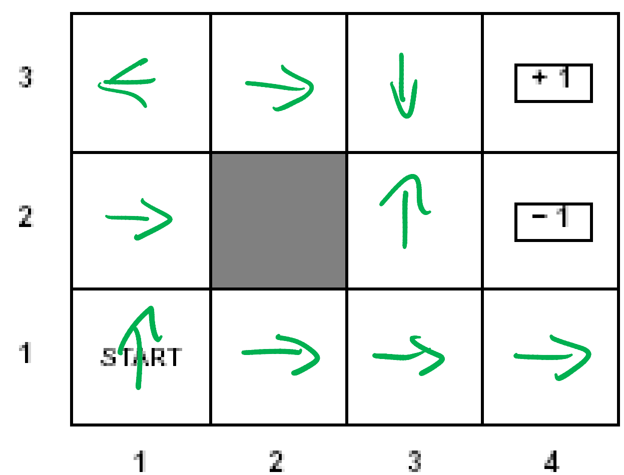
- **Recap: MDPs and** More on MDP Example
- Optimal Policies
  - Some Examples
- Course Conclusions

MDPs: Policy

- The robot needs to know what to do as the decision process unfolds…
- It starts in a state, selects an action, ends up in another state selects another action….
- Needs to make the same decision over and over: Given the current state what should I do?

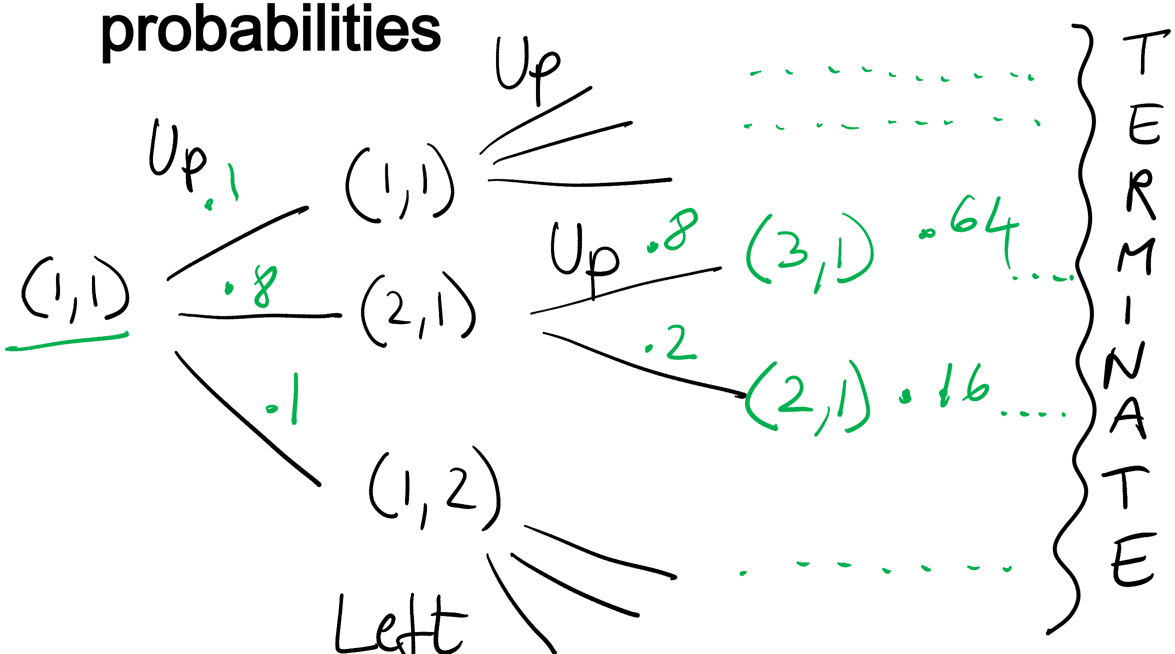
policy

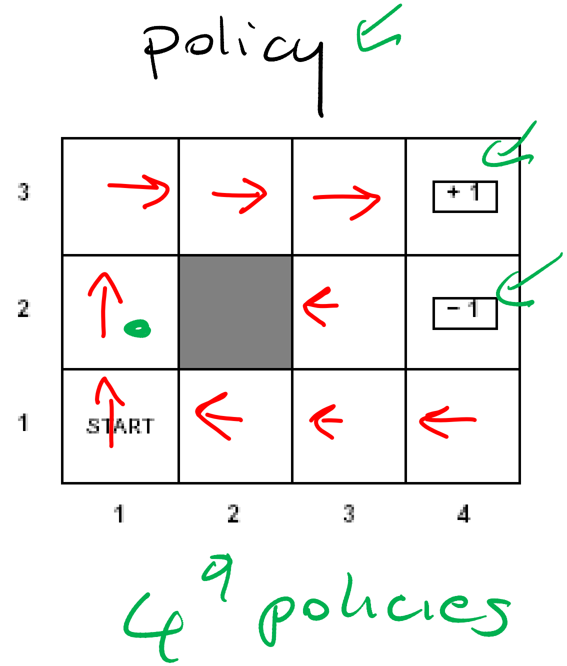
- So a policy for an MDP is a single decision function $\pi(s)$ that specifies what the agent should do for each state s



How to evaluate a policy

A **policy** can generate a **set of state sequences** with different **probabilities**





Each state sequence has a corresponding reward. Typically the sum of the rewards for each state in the sequence

-.04 -.04 +1

$\sum (1,1) \rightarrow (1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (3,4)$

+.72

MDPs: optimal policy

Expected total reward of a policy

$$\sum \overbrace{P(s_0, s_1, \ldots, s_{Terminal})}^{\text{probability}} * \overbrace{\sum R(s_0) \ldots R(s_T)}^{\text{reward}}$$

For all the sequences of states generated by the policy

we sum the product of its probability times its reward

Optimal policy maximizes *expected total reward*

Lecture Overview

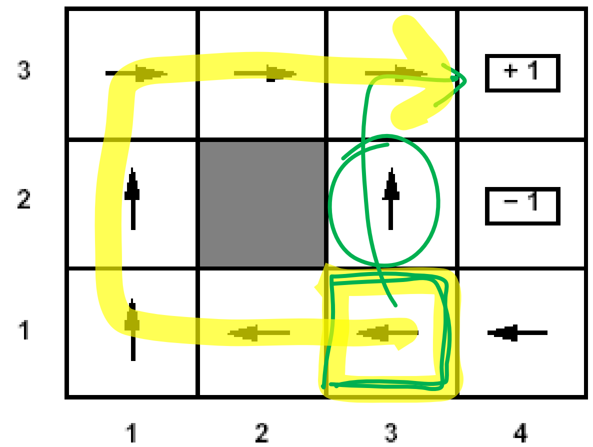
- Recap: MDPs and More on MDP Example
- Optimal Policies
  - Some Examples
- Course Conclusions

Can be computed effectively by an algoritm called VALUE ITERATION. We do not cover it in 322

Rewards and Optimal Policy

Optimal Policy when penalty in non-terminal states is $\boxed{-0.04}$

computed by Value Iteration



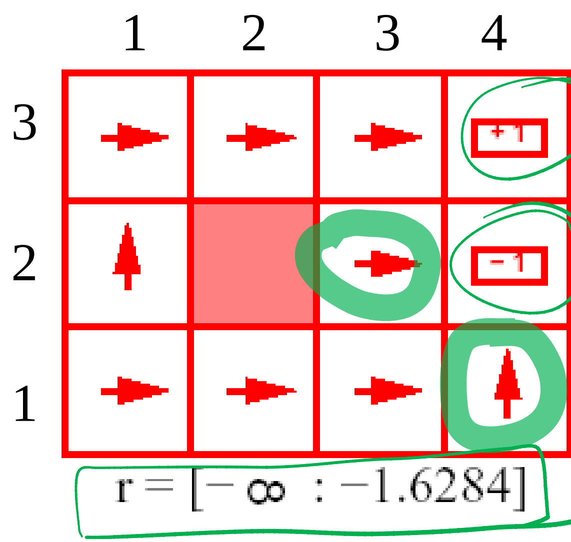
Note that here the cost of taking steps is small compared to the cost of ending into (2,4)

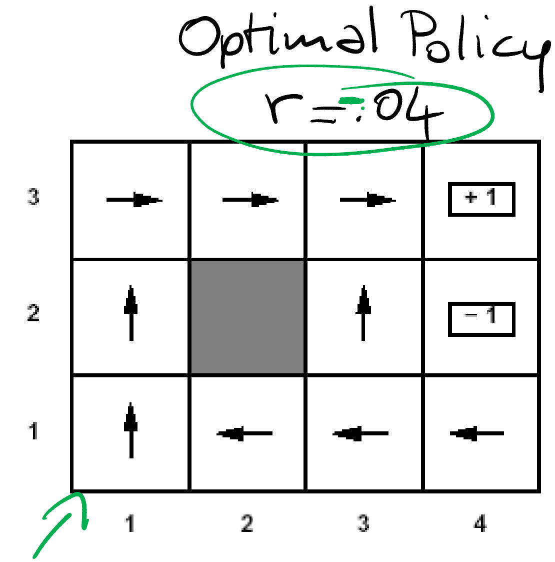
- Thus, the optimal policy for state (1,3) is to take the long way around the obstacle rather then risking to fall into (2,4) by taking the shorter way that passes next to it

May the optimal policy change if the reward in the non-terminal states (let's call it *r*) changes?

Rewards and Optimal Policy

Optimal Policy when r < -1.6284

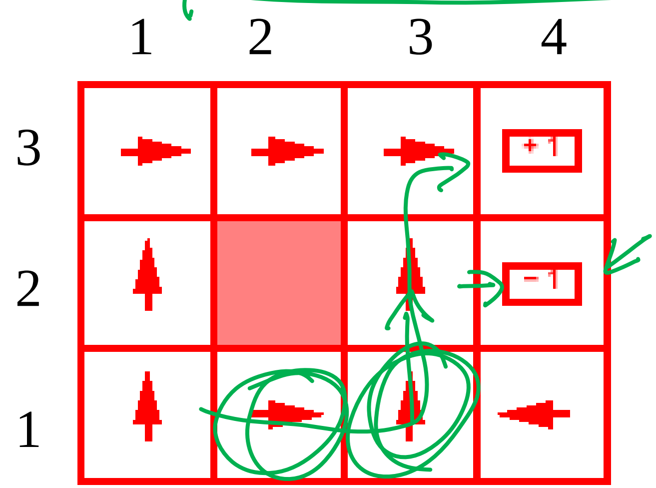




Why is the agent heading straight into (2,4) from its surrounding states?

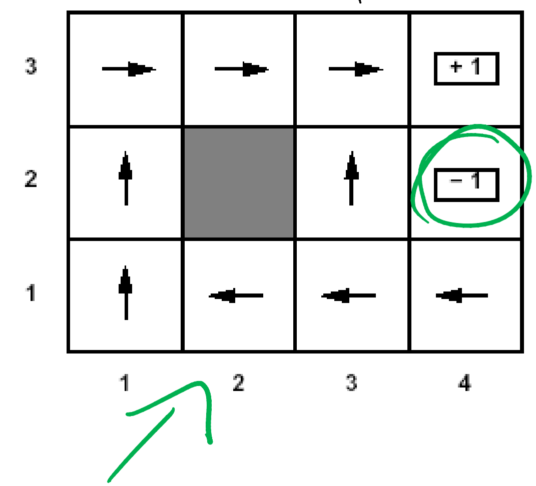
Rewards and Optimal Policy

Optimal Policy when -0.427 < r < -0.085



r = [−0.4278 : −0.0850]

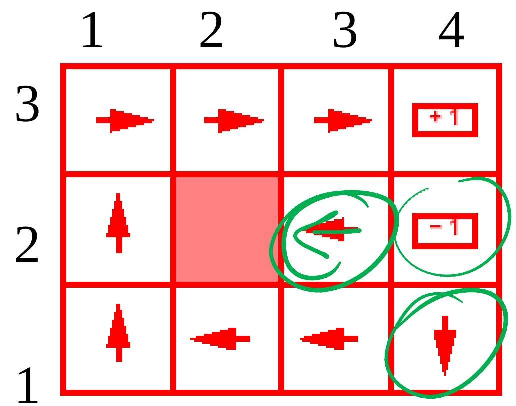
Optimal Policy
r = -.04



The cost of taking a step is high enough to make the agent take the shortcut to (3,4) from (1,3)

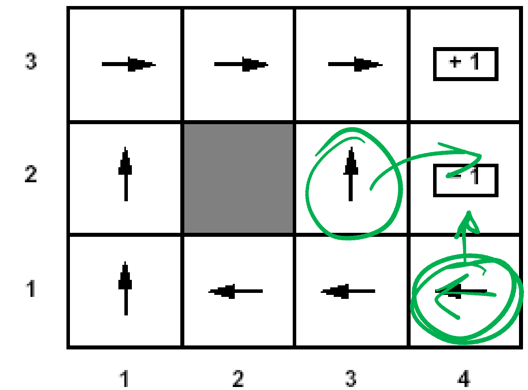
Rewards and Optimal Policy

Optimal Policy when -0.0218 < r < 0



r = [−0.0218 : 0.0000]

Optimal Policy
r=-.04

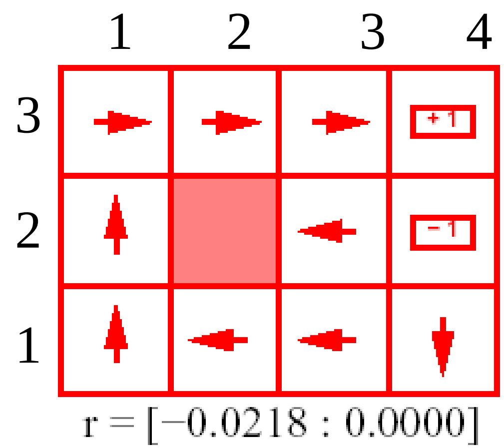


Why is the agent heading straight into the obstacle from (2,3)? And into the wall in (1,4)?

see next slide .....

Rewards and Optimal Policy

Optimal Policy when -0.0218 < r < 0



Stay longer in the grid is not penalized as much as before. The agent is willing to take longer routes to avoid (2,4)

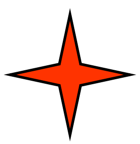
- This is true even when it means banging against the obstacle a few times when moving from (2,3)

Rewards and Optimal Policy

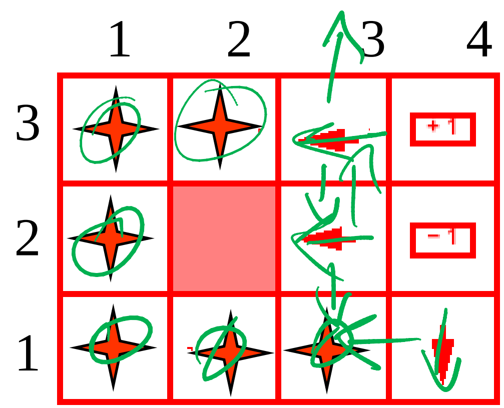
Optimal Policy when r > 0

Which means the agent is rewarded for every step it takes

it avoids terminal states completely



state where every action belong to an optimal policy



$r = [-0.0218 : 0.0000]$

Learning Goals for Monday's and today's class

You can:

- Effectively represent indefinite/infinite decision processes
- Compute the probability of a sequence of actions in a Markov Decision Process (MDP)
- Compute number of Policies of MDP
- Define the computation of the expected total reward of a policy for an MDP
- Explain influence of rewards on optimal policy

After 322 …..

becomes small :)

322 big picture

Most in 422

	Deterministic	Stochastic
CSPs	*Vars + Constraints* More sophisticated SLS	
Query	*Logics* *First Order Logics* *Temporal reasoning* *Description Logics*	*Belief Nets* More sophisticated reasoning *Markov Chains and HMMs*
Planning	*Hierarchical Task Networks* *Partial Order Planning*	*Partially Observable MDP* More sophisticated reasoning

Applications of AI

340

Machine Learning
Knowledge Acquisition
Preference Elicitation

Where are the components of our representations coming from?

The probabilities?
The utilities?
The logical formulas?

From people and from data!

Announcements

- Fill out Online Teaching Evaluations Survey.
- It closes on Apr 13th

- **FINAL EXAM:** Friday Apr 24, 3:30-6:30 pm DMP 110 (not the regular room)

6pts each ~20pts each

Final will comprise: 10 -15 short questions + 3-4 problems

- Work on all **practice exercises** **(11 have been posted!)**
- While you revise the **learning goals**, work on **review questions** - I may even reuse some verbatim ☺
- **Will post a couple of problems from previous offering** (maybe slightly more difficult /inappropriate for you because they were not informed by the learning goals) … **but I'll give you the solutions** ☺
- **Come to remaining Office hours!**

Final Exam (cont’)

- **Assignments:** 20%
- **Midterm:** 30%
- **Final:** 50%

If your final grade is >= 20% higher than your midterm grade:

- Assignments: 20%
- Midterm: 15%
- Final: 65%

Sketch of ideas to find the optimal policy for a MDP (Value Iteration)

We first need a couple of definitions

- $V^{\pi}(s)$: the expected value of following policy π in state s
- $Q^{\pi}(s, a)$, where a is an action: expected value of performing *a* in *s*, and then following policy π.

We have, by definition

$$Q^{\pi}(s,a) = \sum_{s'} P(s' \mid s, a)\left(R(s', s, a) + V^{\pi}(s')\right)$$

states reachable from *s* by doing *a*

Probability of getting to *s'* from *s* via *a*

reward of getting to *s'* from *s* via *a*

expected value of following policy π in *s'*

322 Conclusions

Artificial Intelligence has become a huge field.

After taking this course you should have achieved a reasonable understanding of the basic principles and techniques…

But there is much more…

422 Advanced AI

340 Machine Learning

425 Machine Vision

Grad courses: Natural Language Processing, Intelligent User Interfaces, Multi-Agents Systems, Machine Learning, Vision

Value of a policy and Optimal policy

We can then compute $V^{\pi}(s)$ in terms of $Q^{\pi}(s, a)$

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

Expected value of following π in s

Expected value of performing the action indicated and follow π after that

action indicated by π in s

Optimal policy π^* is one that gives the action that maximizes Q^{π^*} for each state

V^0

one eq. for each state

$$V^{(k+1)}(s) = \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V^{(k)}(s'))$$

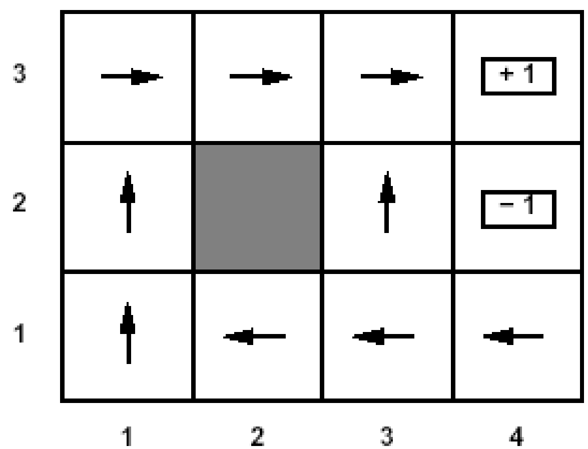
$\pi^*(s)$ = action that maximizes • at the end

Optimal Policy in our Example

Total reward of an environment history is the sum of the individual rewards

- For instance, with a penalty of -0.04 in not terminal states, reaching (3,4) in 10 steps gives a total reward of ….
- Penalty designed to make the agent go for ………solution paths

Below is the optimal policy when penalty in non-terminal states is - 0.04

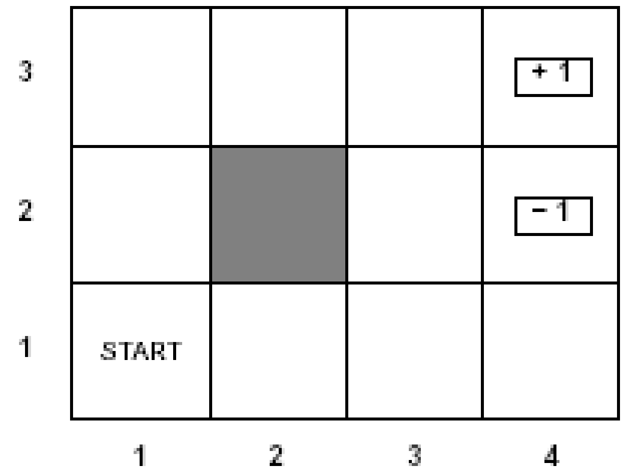


MDPs: Policy

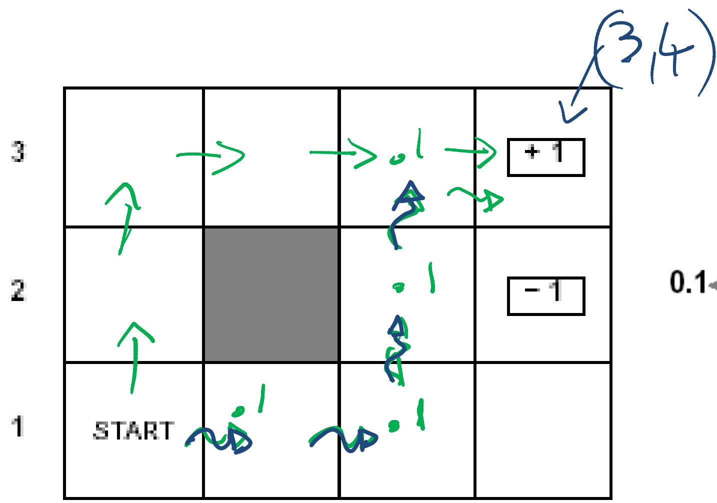
- So what is the best sequence of actions for our Robot?
- There is no best sequence of actions!
- As we saw for decision networks, our aim is to find an **optimal policy**: a set of $\delta_1, \ldots\ldots, \delta_n$ decision functions

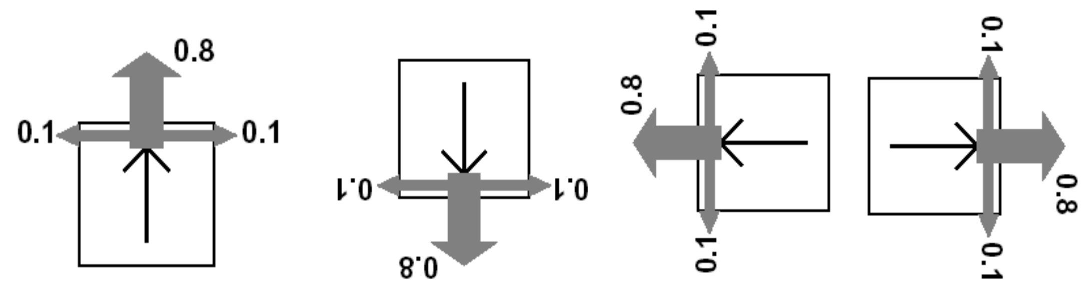
- But in an MDP the decision to be made is always…..
- Given the current state what should I do?

- So a policy for an MDP is a single decision function $\pi(s)$ that specifies what the agent should do for each state s



Example MDP: Sequence of actions





Can the sequence [*Up, Up, Right, Right, Right*] take the agent in terminal state (3,4)? $(.8)^5$

Can the sequence reach "the goal" in any other way?

$(.1)^4 \, .8$ ← with prob yes