

Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 5

Jan, 20, 2021

Slide credit POMDP: C. Conati and P. Viswanathan

Lecture Overview

Markov Decision Processes

-
- Finding the Optimal Policy
 - Value Iteration
- **From Values to the Policy**

Rewards and Optimal Policy

Filtering for HMM (more when we will do temporal models)

Partially Observable Markov Decision Processes

- Formal Specification and example
 - Belief State
 - Belief State Update

Value iteration: Visual Summary

Iteration 0

3	0	0	0	+1
2	0		0	-1
1	0	0	0	0
	1	2	3	4

FIRST STEP

3	0.812	0.868	0.912	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388
	1	2	3	4

REPEAT FOR MANY SIMILAR STEPS

➤ Apply the 9 Bellman Equations

$$V^{(i)}(1,1) = -0.04 + 1 * \max \begin{cases} 0.8V^{(i)}(1,2) + 0.1V^{(i)}(2,1) + 0.1V^{(i)}(1,1) & \text{UP} \\ 0.9V^{(i)}(1,1) + 0.1V^{(i)}(1,2) & \text{LEFT} \\ 0.9V^{(i)}(1,1) + 0.1V^{(i)}(2,1) & \text{DOWN} \\ 0.8V^{(i)}(2,1) + 0.1V^{(i)}(1,2) + 0.1V^{(i)}(1,1) & \text{RIGHT} \end{cases}$$



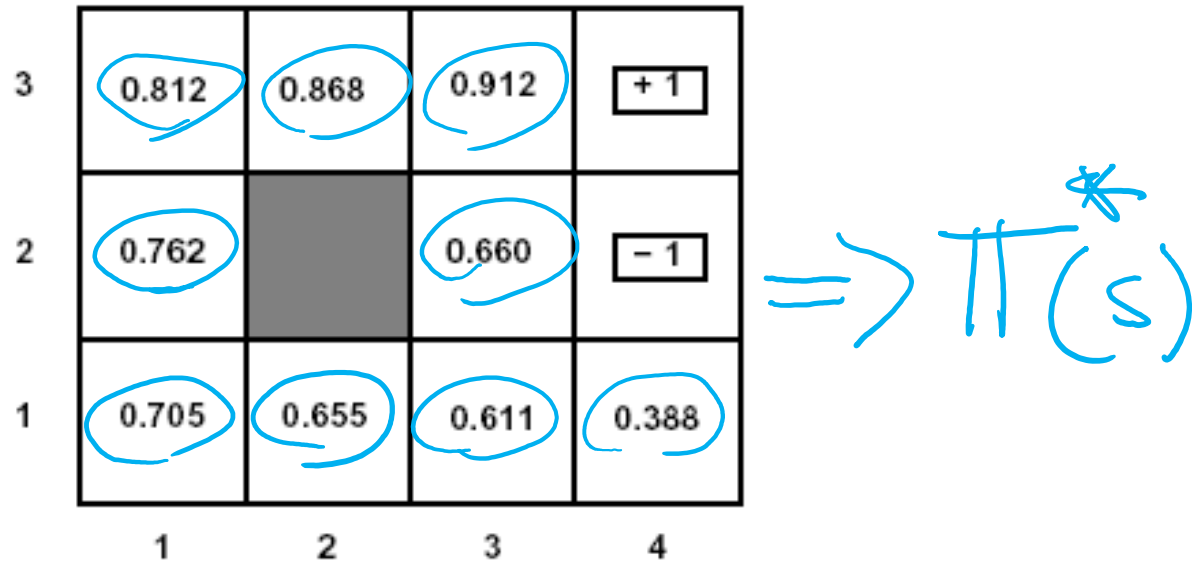
Iteration 1

3	-0.04	-0.04	0.76	+1
2	-0.04		-0.04	-1
1	-0.04	-0.04	-0.04	-0.04
	1	2	3	4



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

Value Iteration: from state values V to π^*



- Now the agent can choose the action that **maximizes its expected utility**

Example: from state values V to π^*

To find the best action in (1,1)

- I should consider all the *actions*
- For each *action* see
 - in which *states* I can end up in
 - with what *probability*
 - And their *value*

3	0.812	0.868	0.912	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388
	1	2	3	4

The expected value/utility of an *action* is the sum across all the states of value of each state times its probability

So for state (1,1) the expected utility of **RIGHT** is

$$0.8 V(2,1) + 0.1 V(1,2) + 0.1 V(1,1) =$$

$$(0.8 * 0.655) + (0.1 * 0.762) + (0.1 * 0.705)$$

Example: from state values V to π^*

- The action of the optimal policy is the one that **maximizes the expected utility**

- For state (1,1)

3	0.812	0.868	0.912	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388
	1	2	3	4

$$\pi^*(1,1) = \arg \max \left[\begin{array}{l} 0.8 \overset{.762}{V(1,2)} + 0.1 \overset{.655}{V(2,1)} + 0.1 \overset{.705}{V(1,1)} \quad \text{UP } \star \\ 0.9 \overset{.705}{V(1,1)} + 0.1 \overset{.762}{V(1,2)} \quad \text{LEFT} \\ 0.9 \overset{.705}{V(1,1)} + 0.1 \overset{.655}{V(2,1)} \quad \text{DOWN} \\ 0.8 \overset{.705}{V(2,1)} + 0.1 \overset{.762}{V(1,2)} + 0.1 \overset{.655}{V(1,1)} \quad \text{RIGHT} \end{array} \right]$$

- In general:
$$\pi^*(s) = \arg \max_a \sum_{s'} P(s'|s, a) V^{\pi^*}(s')$$

Value Iteration: from state values V to π^*

- Now the agent can choose the action that implements the MEU principle: maximize expected utility

$$\pi^*(s) = \arg \max_a \sum_{s'} P(s'|s, a) V^{\pi^*}(s')$$

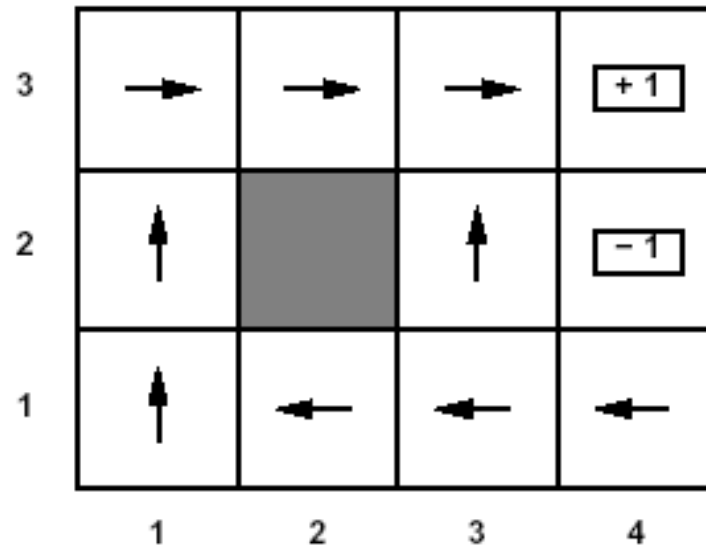
states reachable from s by doing a

Probability of getting to s' from s via a

expected value of following policy π^* in s'

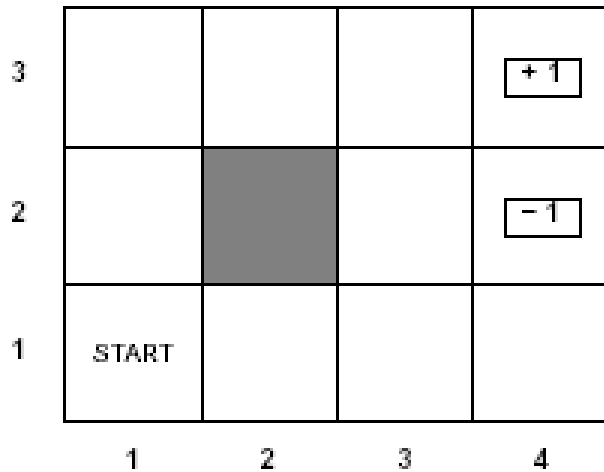
Optimal policy

➤ This is the policy that we obtain....

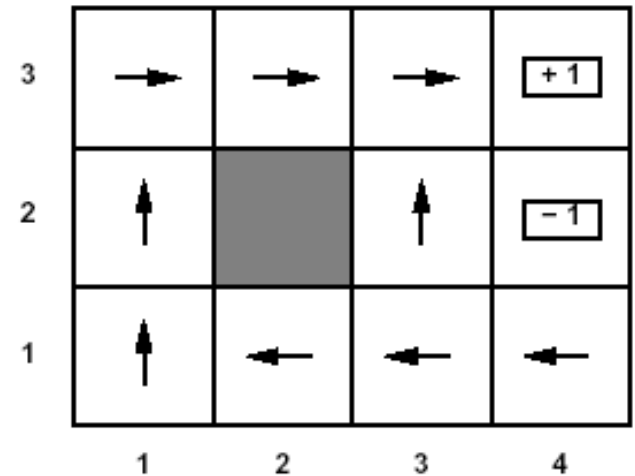


Optimal policy

- Reward structure for our example



- This is the policy that we obtain by applying Value Iteration to our example

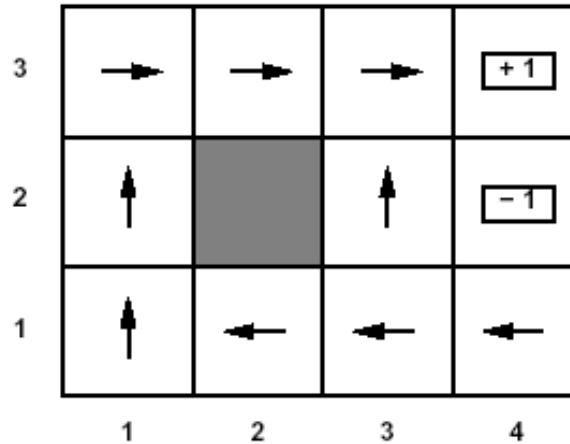


$$R(s) = \begin{cases} -0.04 & \text{(small penalty) for nonterminal states} \\ \pm 1 & \text{for terminal states} \end{cases}$$

Rewards and Optimal Policy

Optimal Policy when reward in non-terminal states is -0.04

*computed
by Value
Iteration*



Is it possible that the optimal policy changes if the reward in the non-terminal states changes?

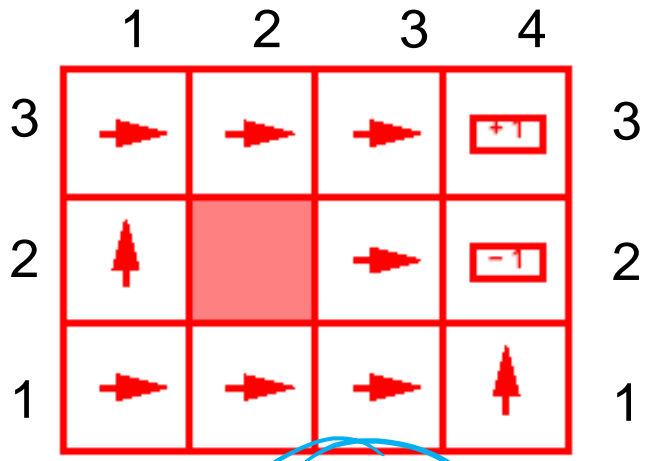
A. Yes

B. No

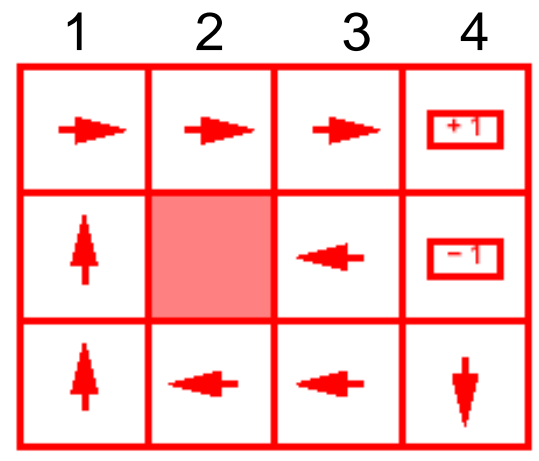
Rewards and Optimal Policy iclicker.

If $r = -2$, what would be a reasonable policy?

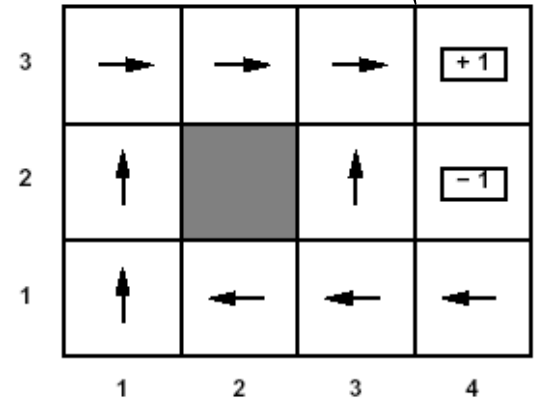
Optimal Policy
 $r = -0.4$



A.

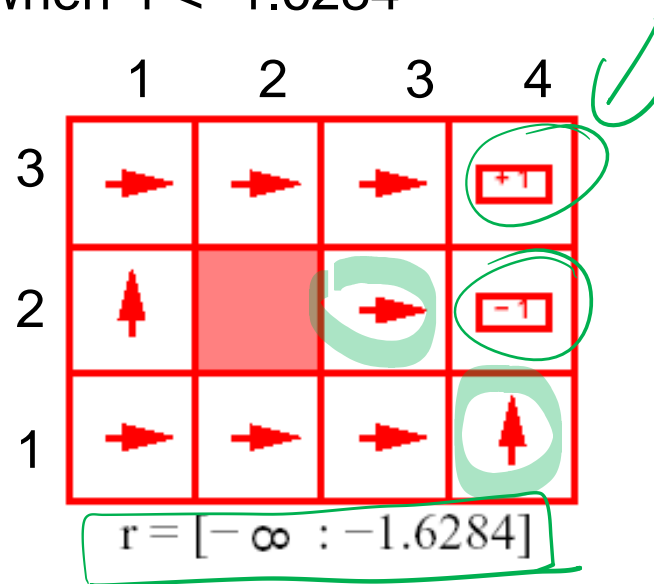


B.

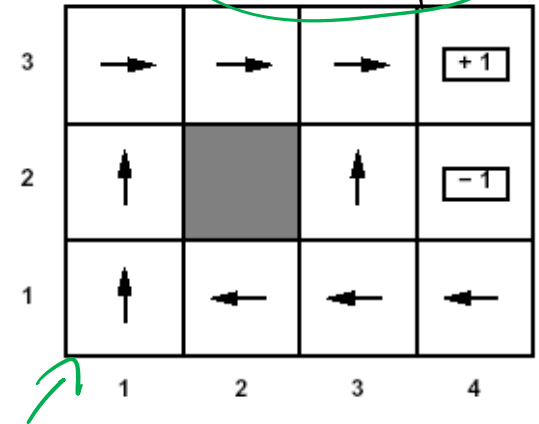


Rewards and Optimal Policy

Optimal Policy when $r < -1.6284$



Optimal Policy
 $r = -0.4$

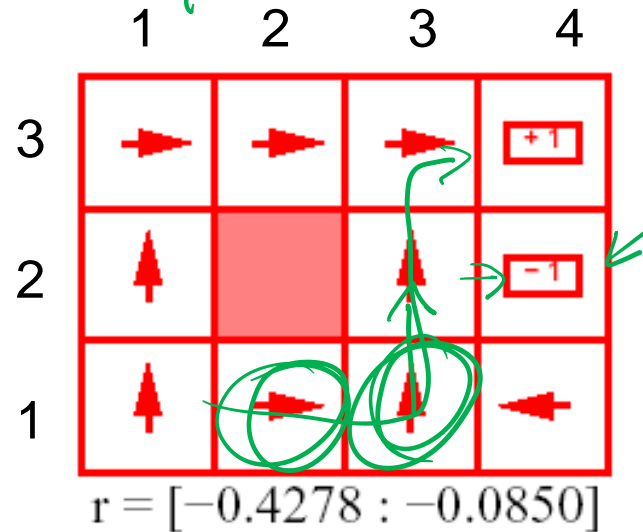


Why is the agent heading straight into (4,2) from its surrounding states?

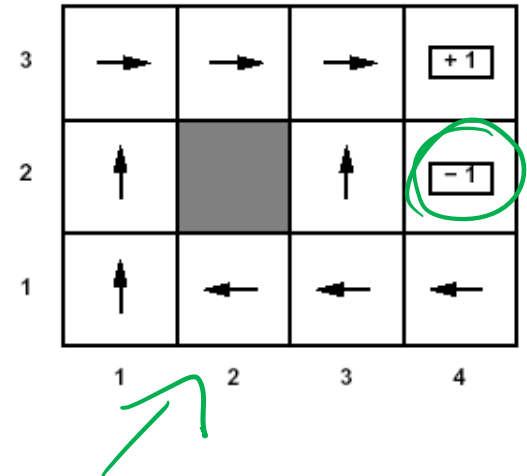
(column, row)
 to indicate state)

Rewards and Optimal Policy

Optimal Policy when $-0.427 < r < -0.085$



Optimal Policy
 $r = .04$

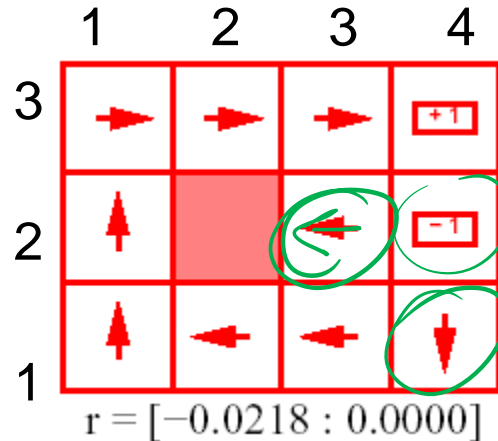


The cost of taking a step is high enough to make the agent take the shortcut to (4,3) from (3,1)

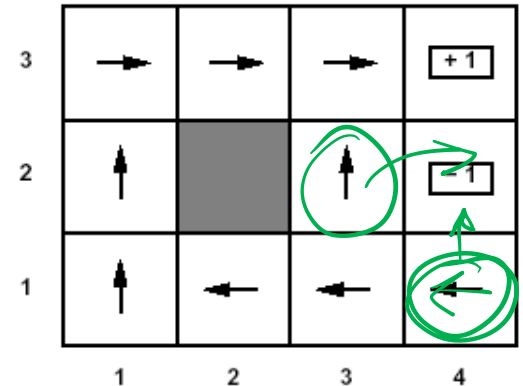
(column, row)
to indicate state

Rewards and Optimal Policy

Optimal Policy when $-0.0218 < r < 0$



Optimal Policy
 $r = 0.04$



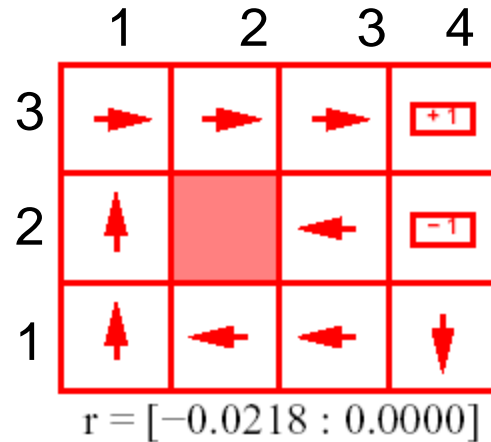
Why is the agent heading straight into the obstacle from (3,2)? And into the wall in (4,1)?

see next slide

(column, row)
to indicate state

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 (4,2)

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

Rewards and Optimal Policy

Optimal Policy when $r > 0$?

Which means the agent is rewarded for every step it takes

Rewards and Optimal Policy

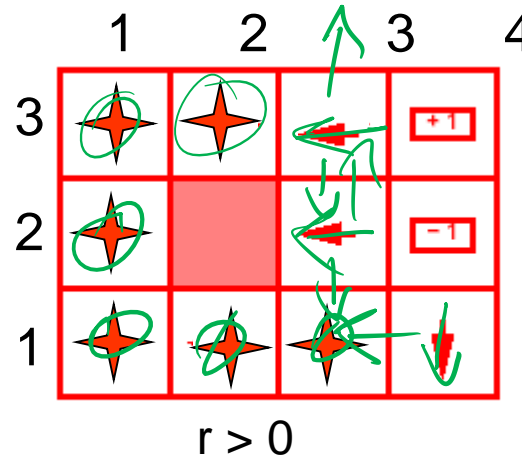
Optimal Policy when $r > 0$

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



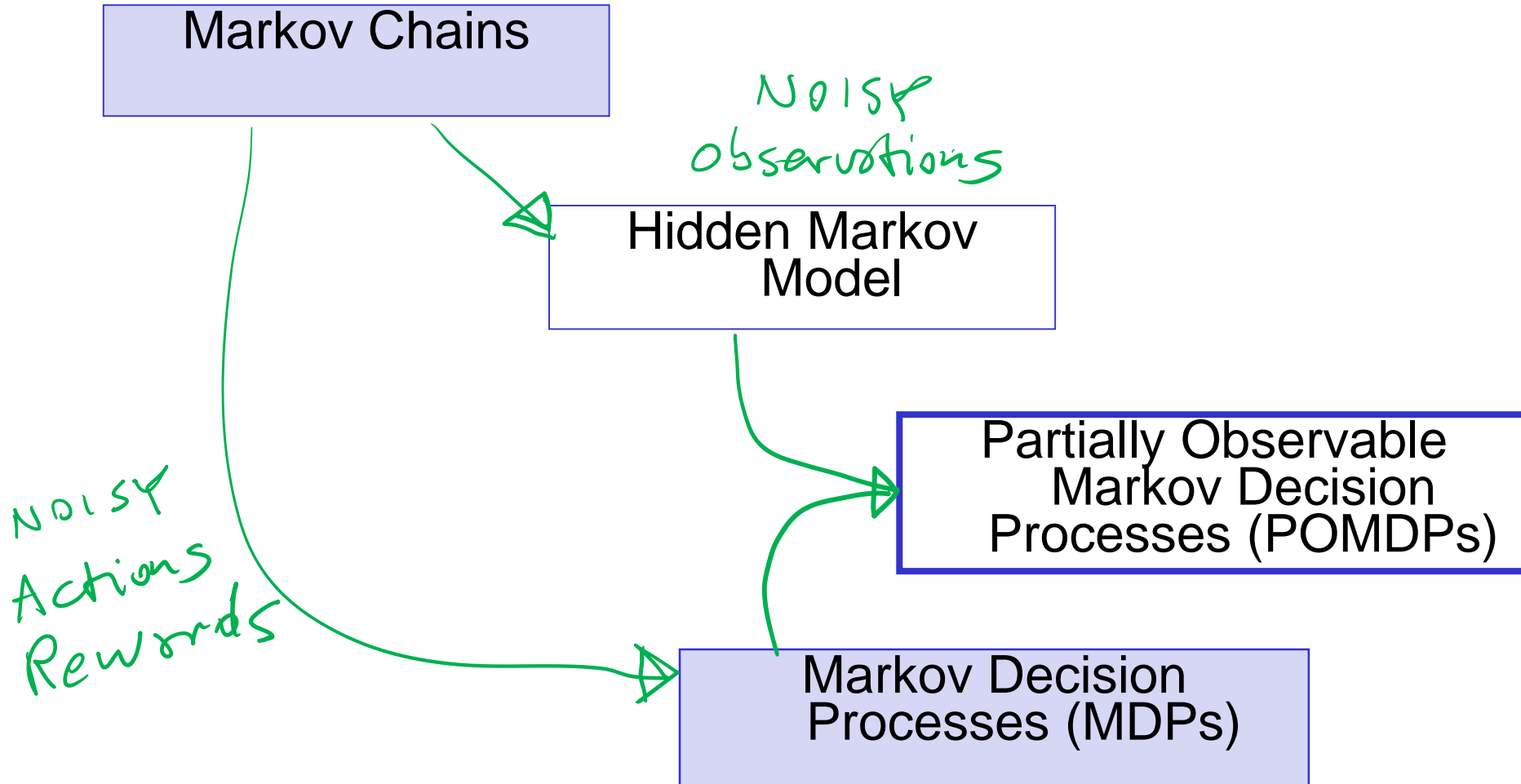
MDPs scalability (not required)

- **Modern optimal algorithms** draw from a vast repertoire of techniques, like **graph algorithms**, **heuristic search**, **compact value function representations**, and **simulation-based approaches**. E.g.,
 - Only compute V for states “reachable” from S_0
 - Do not compute V for really bad states (based on heuristics)
- An enormous number of **approximation algorithms** have been suggested that exploit several intuitions, such as **inadmissible heuristics**, **interleaving planning and execution**, special processing for dead-end states, **domain determinization** ideas, **hybridizing multiple algorithms**, and **hierarchical problem decompositions**.

explore the vast space of optimality-efficiency tradeoffs



Markov Models



Lecture Overview

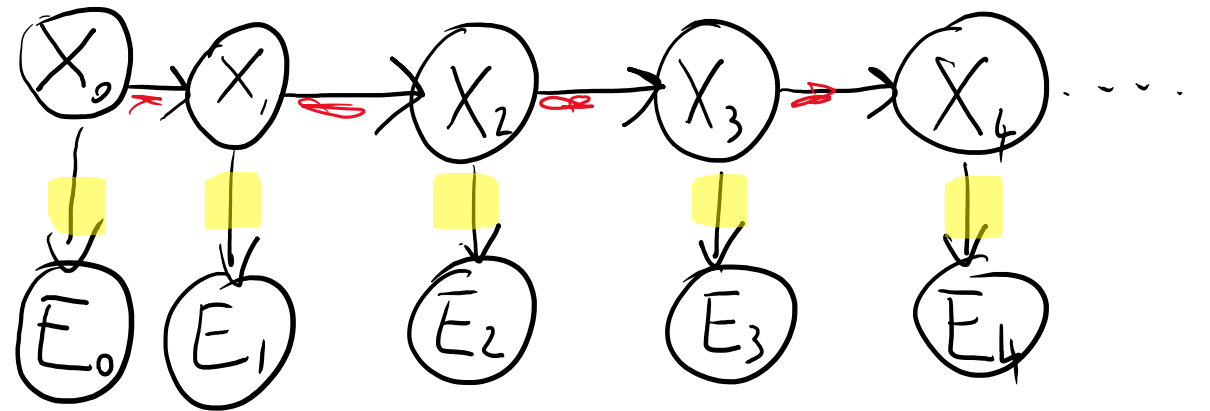
Filtering for HMM (more when we will do temporal models)

Partially Observable Markov Decision Processes

- Formal Specification and example
 - Belief State
 - Belief State Update

Hidden Markov Model

- A **Hidden Markov Model (HMM)** starts with a Markov chain, and adds a noisy observation/evidence about the state at each time step:



- $|\text{domain}(X)| = k$
- $|\text{domain}(E)| = h$

- $P(X_0)$ specifies initial conditions (probability distrib. for start state)
- $P(X_{t+1}|X_t)$ specifies the dynamics
- $P(E_t|S_t)$ specifies the sensor model

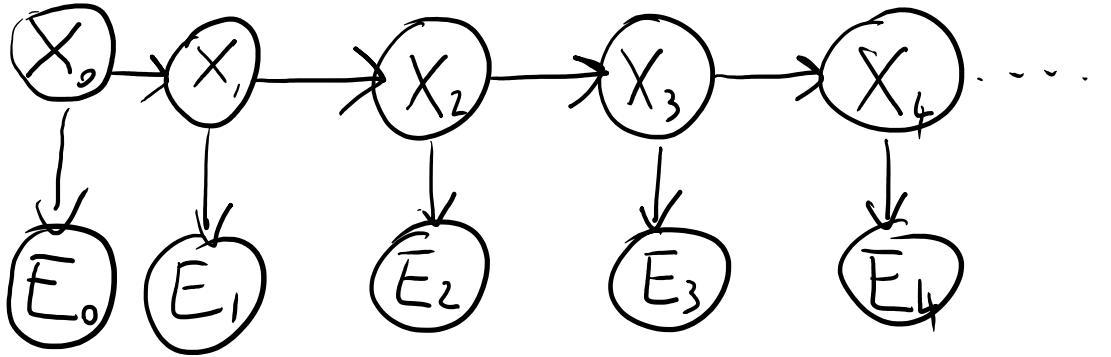
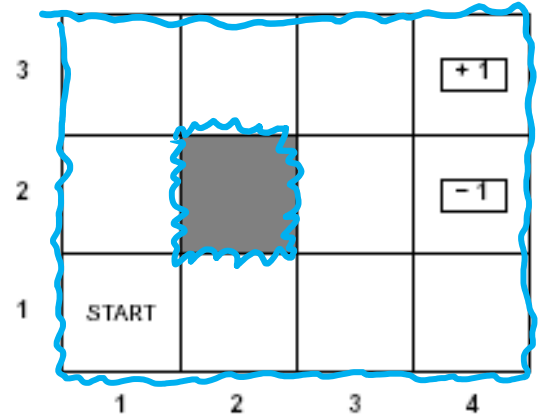
K

$K \times K$

$K \times h$ { K prob. dist. over O }

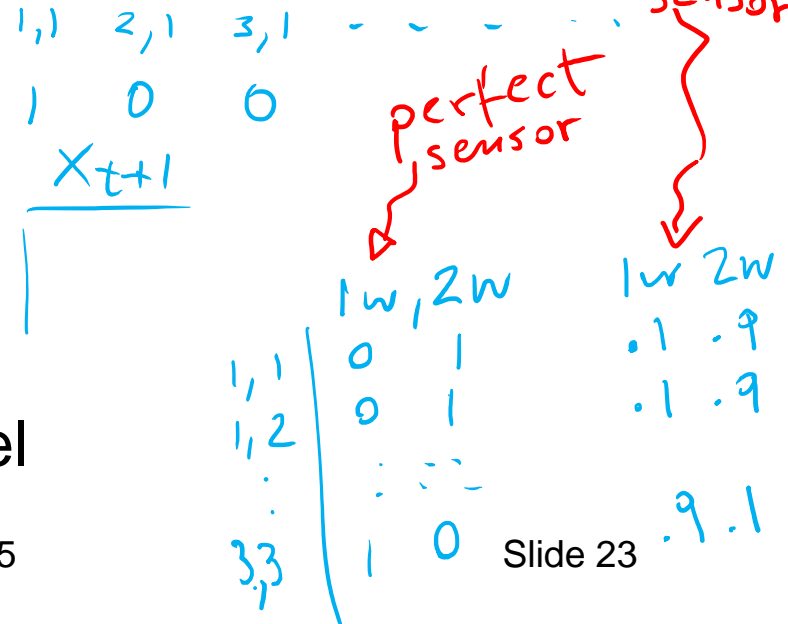
Hidden Markov Model (our example with no actions)

- $E = \# \text{ of walls } \{1w, 2w\}$



- $|\text{domain}(X)| = 11$
- $|\text{domain}(E)| = 2$

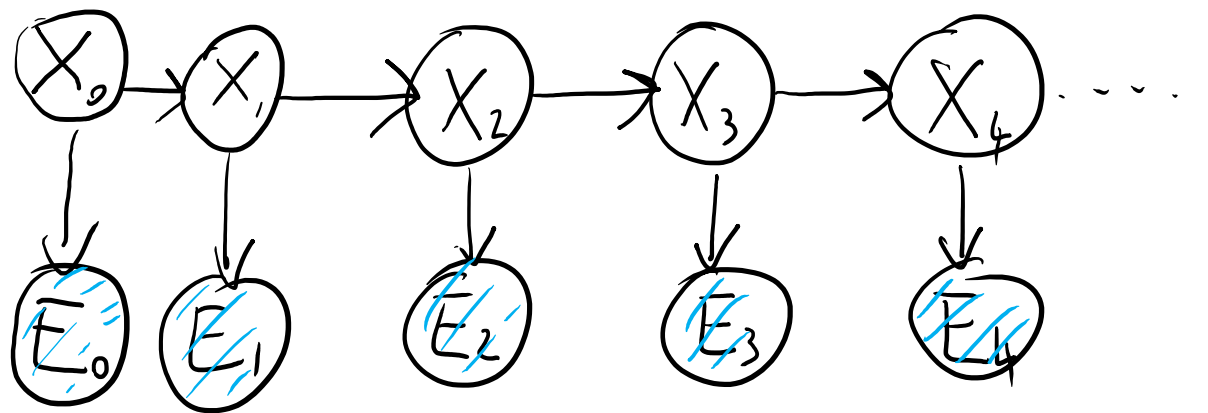
- $P(X_0)$ specifies initial conditions
- $P(X_{t+1}|X_t)$ specifies the dynamics
 $|| X ||$
- $P(E_t|S_t)$ specifies the sensor model



Useful inference in HMMs

- **In general (Filtering):** compute the posterior distribution over the current state given all evidence to date

$$P(X_t | e_{0:t})$$



$$P(X_4 | e_0 \dots e_4)$$

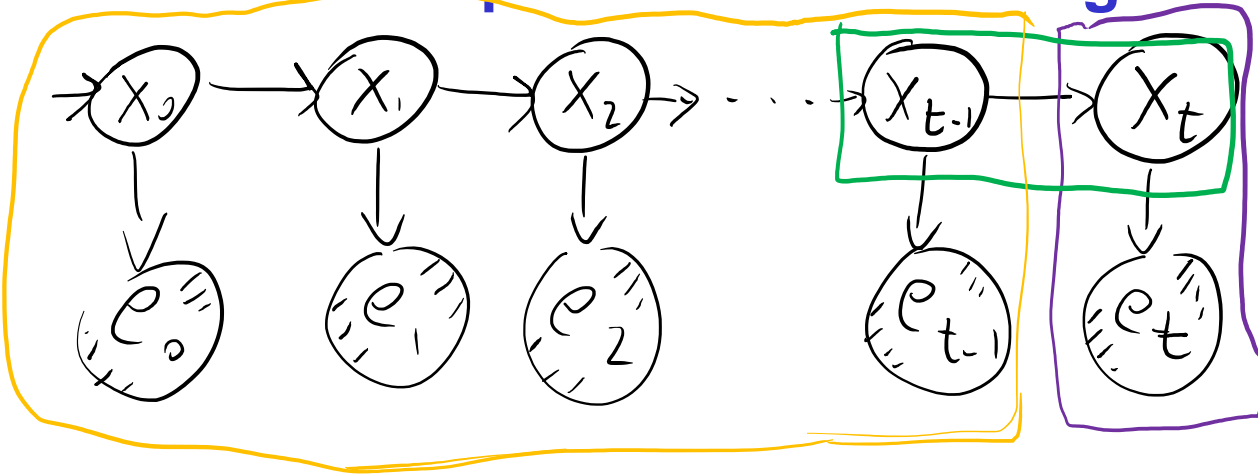


- observed
- value is known

$E_0 = e_0 \dots$

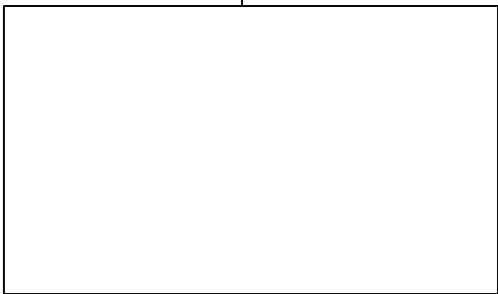
e.g. 1w 1w 2w 1w 2w

Intuitive Explanation for filtering recursive formula

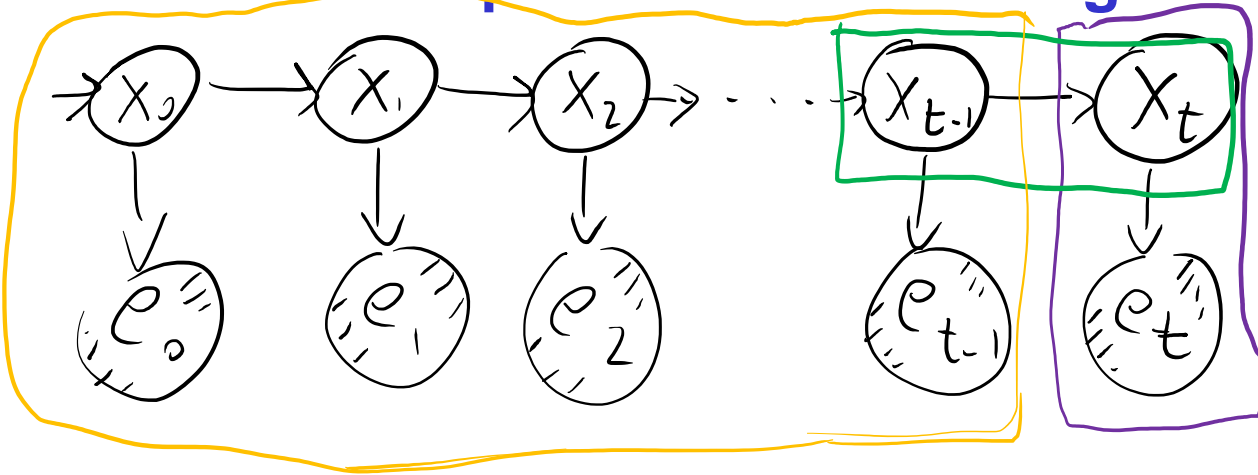


sequence of
evidences
 $e_0:e_t$

$$P(X_t | e_{0:t}) =$$



Intuitive Explanation for filtering recursive formula



sequence of evidences $e_0:e_t$

$$P(X_t | e_{0:t}) = \alpha P(e_t | X_t) * \sum_{X_{t-1}} P(X_t | X_{t-1}) * P(X_{t-1} | e_{0:t-1})$$

X_t generated evidence e_t

whatever X_{t-1} was, X_t was reached from there

and evidence $e_{0:t-1}$ must have been generated before getting to X_{t-1}

Lecture Overview

Filtering for HMM (more when we will do temporal models)

Partially Observable MDPs

- Formal Specification and example
 - Belief State
 - Belief State Update

POMDP: Intro

➤ The MDPs we looked at so far were *fully observable*

- The agent **always knows which state it is in**
- The uncertainty is in

actions effects

- Policy only depends on.....?

current state

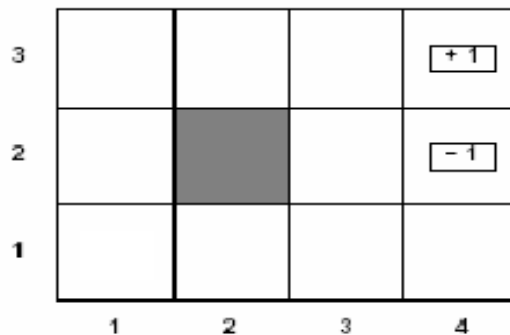
Belief States

➤ In POMDPs, the agent cannot tell for sure where it is in the space state, all it can have are *beliefs* on that

- *probability distribution over states*
- This is usually called *belief state b*
- $b(s)$ is the probability assigned by b to the agent being in state s

➤ **Example:** Suppose we are in our usual grid world, but

- the agent has no information at all about its position in non-terminal states
- It knows only when it is in a terminal state (because the game ends)



➤ What is the initial belief state, if the agent knows that it is not in a terminal state?

Belief States

➤ Initial belief state:

- $\langle 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 0, 0 \rangle$

0.111	0.111	0.111	0.000
0.111		0.111	0.000
0.111	0.111	0.111	0.111

Observation Model

- As in HMM, the agent can learn something about its actual state by *sensing* the environment:
 - **Sensor Model $P(e/s)$** : probability of observing the evidence e in state s
- A POMDP is fully specified by
 - Reward function: $R(s)$ (we'll forget about a and s' for simplicity)
 - Transition Model: $P(s' | a, s)$
 - Observation model: $P(e/s)$
- Agent's belief state is updated by computing **the conditional probability distribution over all the states given the sequence of observations and actions so far**

State Belief Update

➤ We just saw *filtering* for HMM?

- Compute conditional probability distribution over states at time t given all observations so far

$$P(X_t | e_{0:t}) = \alpha P(e_t | X_t) \sum_{x_{t-1}} P(X_t | x_{t-1}) P(x_{t-1} | e_{0:t-1})$$

Inclusion of new evidence (sensor model)

Filtering at time $t-1$

Propagation to time t

➤ State belief update is similar but includes actions

- If the agent has current belief state $b(s)$, performs action a and then perceives evidence e , the new belief state $b'(s')$ is

$$b'(s') = \alpha P(e | s') \sum_s P(s' | a, s) b(s)$$

Inclusion of new evidence:
Probability of perceiving e in s'

Sum over all the states that can take to s' after performing a

Filtering at time $t-1$:
State belief based on all observations and actions up to $t-1$

Propagation at time t : Probability of transition to s' given s and a

Learning Goals for today's class

You can:

- Compute optimal policy from output of Value Iteration
- Define and compute **filtering** on an HMM (intuitively)
- Define a **POMDP**
- Define and compute a **state belief update** for a **POMDP**
- Define a **Policy** for a POMDP

TODO for Fri

Read Textbook 9.5.6 Partially Observable MDPs

Check what to do with readings (details on course webpage)

- Carefully read the paper before class
- Send by email
 - (at least 3) questions on the assigned paper
 - a brief summary of the paper (no more than half a page)
 - First Fri 29

Assignment 1 will be out on Fri

Not required

- **Partially Observable Markov Decision Process (POMDP):** As the name suggests, POMDPs model scenarios where the agent cannot observe the world state fully [123]. A POMDP agent needs to execute actions for two reasons: for changing the world state (as in an MDP) and for obtaining additional information about the current world state. As Section 7.1.1 explains, a POMDP is a large Continuous MDP, in which a state-variable is the world state, and its value denotes the agent's belief (probability) that it is in that state. Straightforward implementations of MDP algorithms do not scale up to POMDPs and, over the years, a large number of specialized POMDP techniques have been developed, with successes in scaling the algorithms to millions of states [214]. POMDPs have also seen several applications, e.g., dialog management [241], intelligent control of workflows [65], intelligent tutoring [200], and several robotic planning applications [233].