CS 440
Introduction to Artificial Intelligence

Lecture 17:

Hidden Markov Models

March 24, 2020

Markov Processes

- Consider an environment
  - Environment may transition to different states
    - Due to actions selected by agent
    - Due to things outside of agent's control
- Example: Autonomous car
  - State of road changes based on what agent does as well as what other agents do
    - Agent may turn, change lanes, accelerate/decelerate, ect.
    - Other cars may move, change lanes, cut you off, ect.
- Very complicated
  - Impossible to predict exactly
- Given state of environment possible to estimate future state
  - Give you position of cars with current speed
    - Predict likely position of cars after certain amount of time has passed

Markov Processes

- Markovian assumption
  - Future state only depend on current state
  - Only matters where cars currently are on road
    - Doesn't matter what maneuvers they took to get there
  - Assumption makes solving problems a lot easier
    - Only need to keep track of current state
      - As opposed to history of previous states
    - Only need to reason over current state

RUTGERS

Types of Markov Processes

- **Discrete vs continuous**
- **Passive vs active**
  - **Active process: The agent's actions influence process**
- **Observable state vs partially observable state**
  - **Observable state: agent can observe state directly**
  - **Partially observable state:**
    - **Example: Wumpus world**

	Observable State	Hidden State
Passive	Markov Chain	Hidden Markov Model
Active	Markov Decision Process (MPD)	Partially Observable Markov Decision Process (POMPD)

Markov Decision Problem

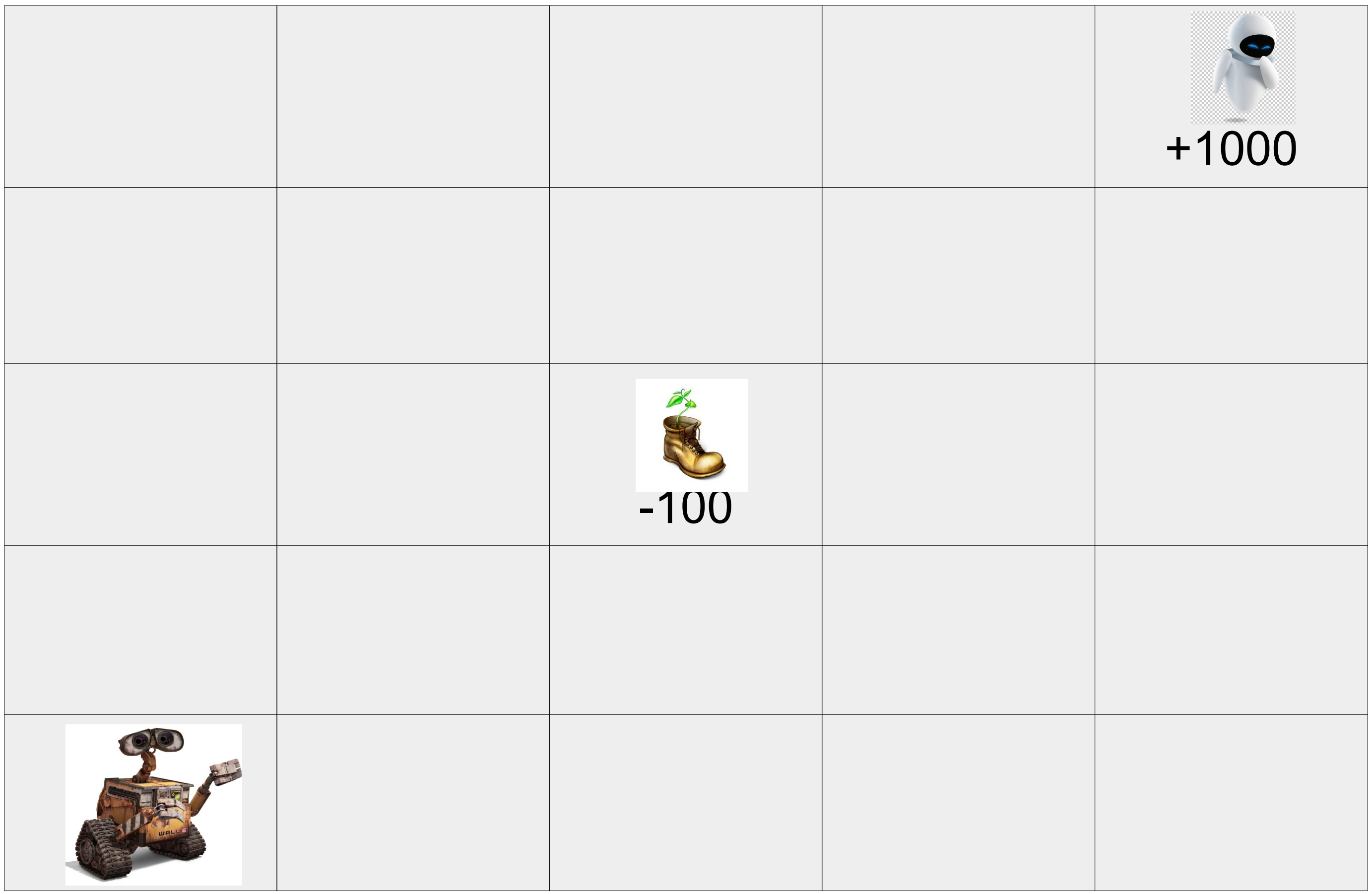
- **May be discrete or continuous**
- **Active - agents actions effect state**
- **Fully observable**
- **May be discrete or continuous**

Formalization

- State space S
- Set of actions A
- Transition function T(s,a,s')
  - $T(s,a,s') = p(s'|s',a)$
    - Probability that you will end up in state s' if you take action a while in state s
    - Defined for all combinations of $s \in S$, $a \in A$, $s' \in S$
- Reward function R
  - Could define reward of being in a state, R(s)
  - Could define reward of performing action while in state R(s,a)
  - Could be reward of performing action that ends up in a particular state R(a,s')
- Immediate objective: Determine the best action to take given your current state
  - Action that maximizes expected future reward

Markov Chains Example

- **Robot in a grid with noisy actions**
  - **Robot can choose Left/Right/Up/Down**
  - **Actions may bring robot to wrong cell**



Possible Solutions

- **Objective: find best action**
- **Search**
  - **Branching faction equal to number of actions**
  - **For each node in search tree need probability for each state**
  - **Need to compute for every state**
  - **Can blow up quickly**

Policy

- **Policy is a mapping of states to actions**
  - **$\Pi(s) \rightarrow a$**
- **Policies are solutions to MDPs**
- **Optimal policy is a policy that maps each state to the action which maximizes expected future reward.**

Computing Policies

- Construct a policy that is optimal for next n moves
  - Define Π_n to be a policy that is optimal for n steps
  - Define R_n to be the expected reward for this policy
- Construct inductively
  - Assume you have a policy Π_i that is optimal over i steps
  - $R_{i+1}(s) = \max_{a\in A}(R(a,s) + \sum_{s\in S} p(s'|s)\, R_i(s'))$
  - $\Pi_{i+1}(s) = \text{argumax}_{a\in A}(R(a,s) + \sum_{s\in S} p(s'|s)\, R_i(s'))$
- What can you say if $R_{i+1}(s) = R_i(s)$ and $\Pi_{i+1}(s) = \Pi_i(s)$?
  - Policy won't change for all future iterations
  - $\Pi_{i+1}(s)$ is an optimal policy

Computing Policies

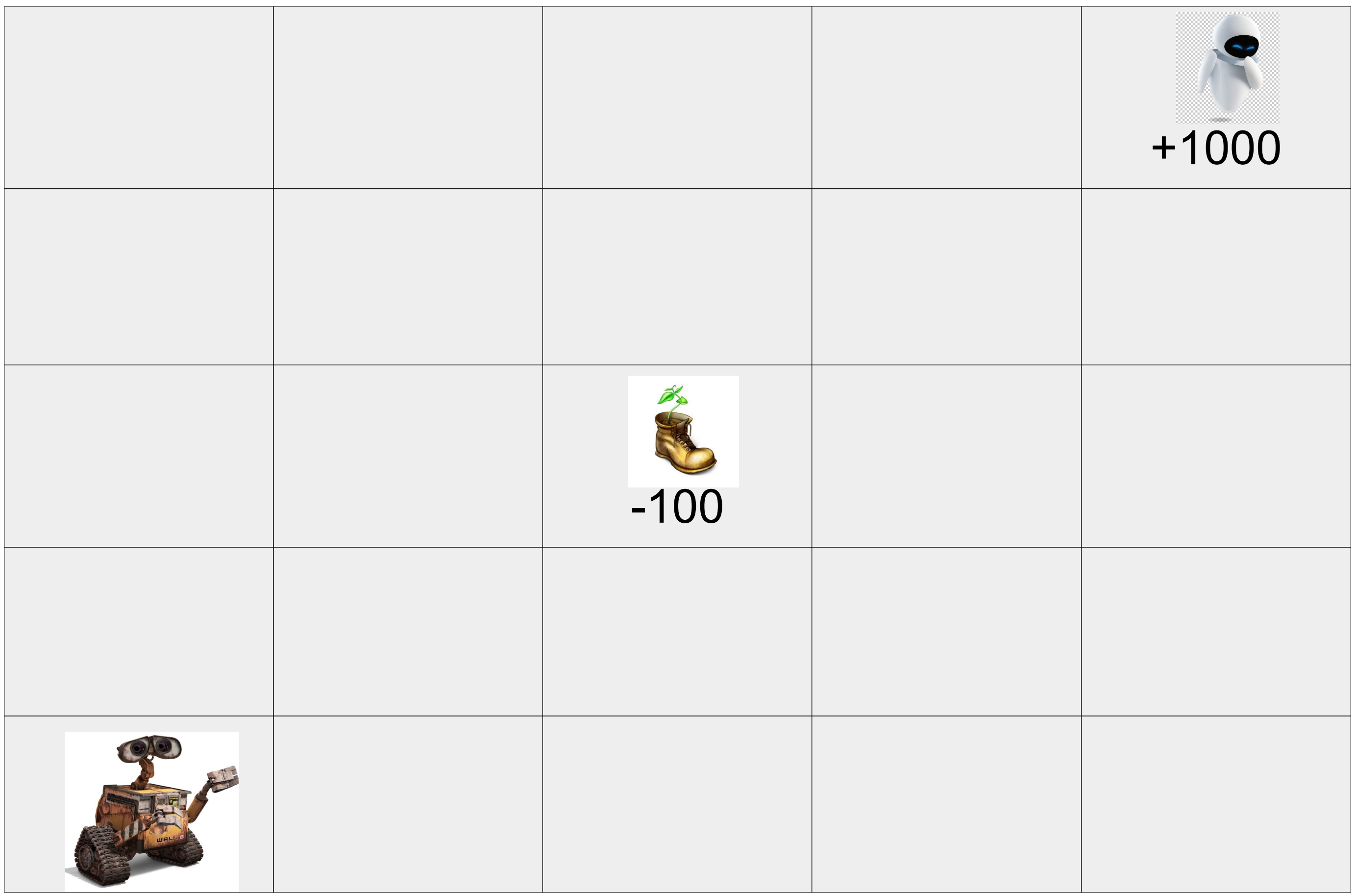
- Idea: iteratively compute $R_{i+1}(s)$ and $\Pi_{i+1}(s)$ from $R_{i+1}(s)$ until it converges to optimal
  - do
    - For all $s \in S$, $a \in A$, $s' \in S$
      - $R_{i+1}(s) = \max_{a \in A}(R(a,s) + \sum_{s \in S} p(s'|s) R_i(s'))$
      - $\Pi_{i+1}(s) = \text{argumax}_{a \in A}(R(a,s) + \sum_{s \in S} p(s'|s) R_i(s'))$
  - Until $R_{i+1}(s) = R_i(s)$ and $\Pi_{i+1}(s) = \Pi_i(s)$
- Problem
  - Does not take into account number of steps to get to goal
    - Sequence of n moves to goal yields same reward as single move to goal

Discounting

- Multiply reward of future steps by discounting factor α
- do
  - For all $s \in S$, $a \in A$, $s' \in S$
    - $R_{i+1}(s) = \max_{a \in A}(R(a,s) + \alpha \sum_{s \in S} p(s'|s)\, R_i(s'))$
    - $\Pi_{i+1}(s) = \text{argumax}_{a \in A}(R(a,s) + \alpha \sum_{s \in S} p(s'|s)\, R_i(s'))$
- Until $R_{i+1}(s) = R_i(s)$ and $\Pi_{i+1}(s) = \Pi_i(s)$

Markov Chains Example

- **Robot in a grid with noisy actions**
  - **Robot can choose Left/Right/Up/Down**
  - **Actions may bring robot to wrong cell**



Types of Markov Processes

- Discrete vs continuous
- Passive vs active
  - Active process: The agent's actions influence process
- Observable state vs partially observable state
  - Observable state: agent can observe state directly
  - Partially observable state:
    - Example: Wumpus world

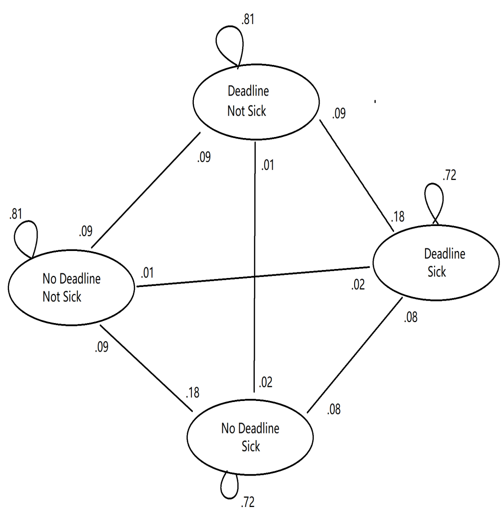
	Observable State	Hidden State
Passive	Markov Chain	Hidden Markov Model
Active	Markov Decision Process (MPD)	Partially Observable Markov Decision Process (POMPD)

Hidden Markov Model

- **May be discrete or continuous**
- **Passive - agents actions don't effect state**
- **Partially observable**
- **May be discrete or continuous**

Hidden Markov Model: Example

- Consider the following model
  - If Bob has a paper deadline at time step i there is a .9 probability he will have a paper deadline at step i+1
  - If Bob has a no deadline at time step i there is a .1 probability he will have a paper deadline at step i+1
  - If Bob is sick at time step i there is a .8 probability he will be sick at step i+1
  - If Bob is not at time step i there is a .1 probability he will be sick at step i+1
- If we know the values of deadline and sick this is a Markov Chain



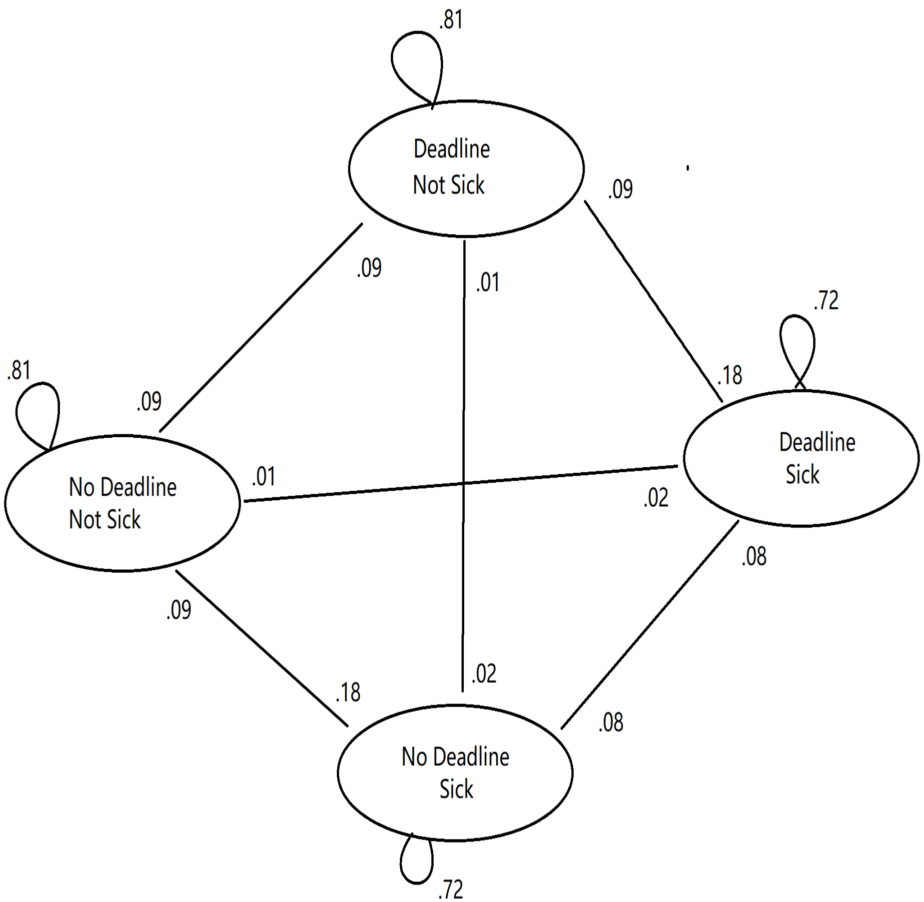
Hidden Markov Model: Example

	Not Sick	Sick
Paper	.9	.6
No Paper	.7	.1

- One an ordinary day Bob has a .7 probability of being in his office
- If Bob is busy working on a paper deadline he has a .9 probability of being in his office
- If he is sick and there is no paper deadline he has a .1 probability of being in his office
- If he is sick but there is a paper deadline he has a .6 probability of being in his office
- John does not know if Bob has a paper deadline of if he is sick
  - He only knows if Bob is in his office
  - Can he infer the probability Bob has a paper deadline or is sick based on these observations?
  - Can he predict if Bob will be in his office?

RUTGERS

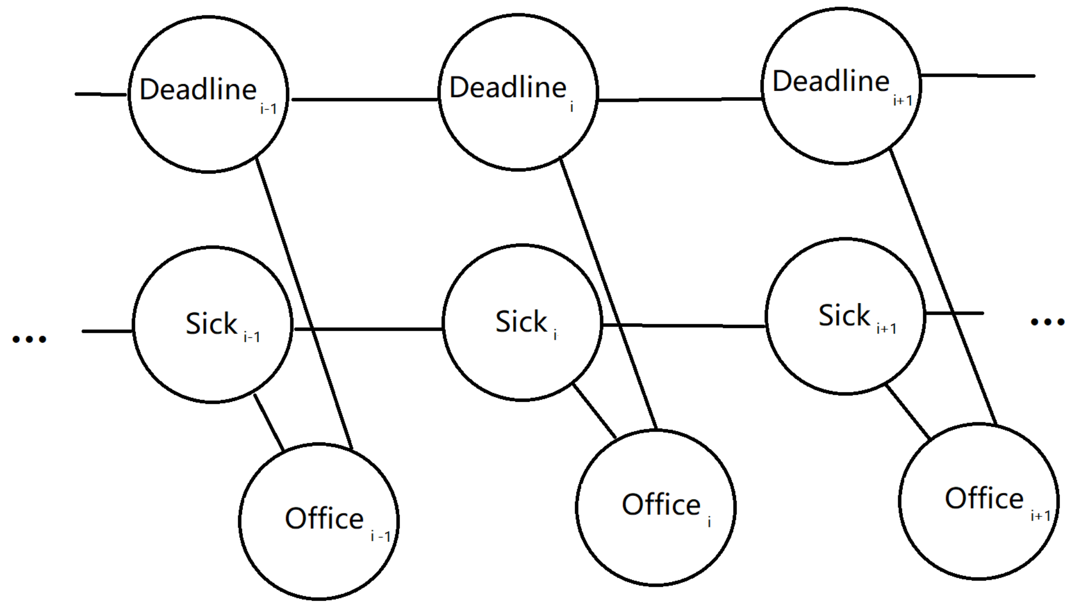
Hidden Markov Model: Example



	Not Sick	Sick
Paper	.9	.6
No Paper	.7	.1

RUTGERS

Model as Bayesian Infinite Network



- At each step Deadline and Sick are dependent on values in previous step
- Office depends on values of Deadline and Sick for that step
- Need to know probabilities of Deadline and Sick at initial step
  - Let $p(Deadline_0) = .1$
  - Let $p(Sick_0)=.1$
- At step i
  - Know value of $Office_i$
  - Compute $p(Deadline_i)$ from $p(Deadline_{i-1})$ and $Office_i$
  - Compute $p(Sick_i)$ from $p(Sick_{i-1})$ and $Office_i$

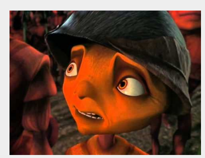
Hidden Markov Model: Solution

- At each step
  - Apply transition
    - Compute p(Deadline_i) given p(Deadline_{i-1})
    - Compute p(Sick_i) given p(Sick_{i-1})
  - Incorporate Observation
    - p(Deadline_i|Office_i)
    - p(Sick_i|Office_i)

	Initial	Transition 1	Observation 1
Observation			In office
p(Deadline)	.1	.1	.136
p(Sick)	.1	.17	.04

Hidden Markov Model Example 2

- **Food at corner of grid**
  - **Don't know where food is**
  - **And has .3 probability of moving towards food and .1 probability of moving away**

		.1		
	.1		..3	
		.3