CS 730/730W/830: Intro AI

■ Class Outline

MDPs

Class Outline

MDPs

- 1. search: heuristics, CSPs, games
- 2. knowledge representation: FOL, resolution
- 3. planning: STRIPS, MDPs
- 4. learning: supervised, unsupervised
- 5. uncertainty: particle filters, HMMs

MDPs

- Examples
- Probability
- Definition
- \blacksquare What to do?
- Break

Solving MDPs

Markov Decision Processes

Wheeler Ruml (UNH)

Lecture 19, CS 730 – 3 / 11

Examples

- Class Outline
- MDPs
- Examples
- Probability
- Definition
- What to do?
- Break
- Solving MDPs

- 1. robot navigation
- 2. driving
- 3. business
- 4. war
- 5. diagnosis
- 6. life

Probability

■ Class Outline

MDPs

ExamplesProbability

Definition

■ What to do?

Break

Solving MDPs

propositional domain: discrete or continuous 0-1, sum to 1 distribution of continuous = density $E(X) = \int x pdf(x) dx$ $P(X = x_1)$ written as $P(x_1)$ or if X is true/false, P(x)conditional (=posterior): $P(x|y) = P(x \land y)/P(y)$

■ Class Outline	initial state: s_0
MDPs	transition model: $T(s, a, s') =$ probability of going from s to
 Examples Probability 	s' after doing a .
Definition	reward function: $R(s)$ for landing in state s.
■ What to do? ■ Break	terminal states: $sinks = absorbing states (end the trial).$

	Class	Outline
_		• • • • • •

MDPs

- Examples
- Probability
- Definition

■ What to do?

Break

Solving MDPs

initial state: s_0 transition model: T(s, a, s') = probability of going from s to s' after doing a. reward function: R(s) for landing in state s. terminal states: sinks = absorbing states (end the trial).

objective:

total reward: reward over (finite) trajectory: $R(s_0) + R(s_1) + R(s_2)$ discounted reward: penalize future by γ : $R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \dots$

Lecture 19, CS 730 – 6 / 11

|--|

MDPs

- Examples
- Probability
- Definition

■ What to do?

```
Break
```

Solving MDPs

initial state: s_0 transition model: T(s, a, s') = probability of going from s to s' after doing a. reward function: R(s) for landing in state s. terminal states: sinks = absorbing states (end the trial).

objective:

total reward: reward over (finite) trajectory: $R(s_0) + R(s_1) + R(s_2)$ discounted reward: penalize future by γ : $R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \dots$ find: policy: $\pi(s) = a$ optimal policy: π^* proper policy: reaches terminal state

Wheeler Ruml (UNH)

MDPs

- Examples
- Probability
- Definition
- What to do?
- Break

 $\pi^*(s) =$

MDPs

- Examples
- Probability
- Definition
- What to do?
- Break

Solving MDPs

$$\pi^*(s) = \operatorname*{argmax}_{a} \sum_{s'} T(s, a, s') U^{\pi^*}(s')$$

 $U^{\pi}(s) =$

MDPs

- Examples
- Probability
- Definition
- What to do?
- Break

Solving MDPs

$$\pi^*(s) = \operatorname*{argmax}_{a} \sum_{s'} T(s, a, s') U^{\pi^*}(s')$$

$$U^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^{t} R(s_{t}) | \pi, s_{0} = s]$$

Wheeler Ruml (UNH)

Lecture 19, CS 730 – 7 / 11

MDPs

- Examples
- Probability
- Definition
- What to do?
- Break

Solving MDPs

$$\pi^*(s) = \operatorname*{argmax}_{a} \sum_{s'} T(s, a, s') U^{\pi^*}(s')$$

$$U^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^{t} R(s_{t}) | \pi, s_{0} = s]$$

The key:

$$U(s) = R(s) + \gamma \max_{a} \sum_{s'} T(s, a, s') U(s')$$

(Richard Bellman, 1957)

Wheeler Ruml (UNH)

Lecture 19, CS 730 – 7 / 11

Break

Class	Outl	ine

MDPs

- Examples
- Probability
- Definition
- What to do?
- Break
- Solving MDPs

- asst 9 due Mon Apr 7
 - projects: reading is fine, but probably best to talk with me before starting serious work

MDPs

Solving MDPs

■ Value Iteration

EOLQs

Solving MDPs

Wheeler Ruml (UNH)

Lecture 19, CS 730 – 9 / 11

MDPs

Solving MDPs

Value Iteration

EOLQs

Repeated Bellman updates:

Repeat until happy for each state s $U'(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s, a, s')U(s')$ $U \leftarrow U'$

MDPs

Solving MDPs
Value Iteration

EOLQs

Repeated Bellman updates:

Repeat until happy for each state s $U'(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s, a, s')U(s')$ $U \leftarrow U'$

For infinite updates everywhere, guaranteed to reach equilibrium.

Equilibrium is unique solution to Bellman equations!

asychronous works: converges if every state updated infinitely often (no state permanently ignored)

EOLQs

- Class Outline
- MDPs
- Solving MDPs
- Value Iteration
- EOLQs

- What question didn't you get to ask today?
- What's still confusing?
- What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.

Thanks!