1 handout: slides

Beyond STRIPS

Comparison

Forward: state space

- $+$: expressivity
- $-$: irrelevant states

Backward: sets of states

- $+$: relevant states
- $-$: larger space, reachable states, expressivity

Partial-order: plan space

- $+$: small space
- $+/-$: least commitment
- $-$: poor heuristics

STRIPS Extensions

negated goals: no problem with CWA

disjunctive precondition: for regression, just branch

conditional effects: for regression, if we need the effect, plan for the condition

universal preconditions and effects: just ground goals and preconditions

Setting

STRIPS assumes static, deterministic world, discrete time, single discrete actions.

1. time, resources
2. concurrent actions
3. abstraction: hierarchical planning
4. uncertainty: eg, disjunctive effects
5. execution monitoring, replanning
6. continuous state
7. multiple (self-interested) agents

Class Outline

1. search: heuristics, CSPs, games
2. knowledge representation: FOL, resolution
3. planning: STRIPS, MDPs
4. learning: RL, supervised, unsupervised
5. KR with uncertainty: HMMs, Bayes nets

Markov Decision Processes

Examples

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

Probability

propositional

domain: discrete or continuous

0–1, sum to 1

distribution of continuous = density

$E(X) = \int x\, P(X = x)\, dx$

$P(X = x_1)$ written as $P(x_1)$ or if X is true/false, $P(x)$

conditional (=posterior): $P(x|y) = P(x \wedge y)/P(y)$

Markov Decision Process (MDP)

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).

Markov Decision Process (MDP)

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:

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \ldots$$

Markov Decision Process (MDP)

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:
$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) \ldots$$

find:

policy: $\pi(s) = a$
optimal policy: π^*
proper policy: reaches terminal state

What to do?

$$\pi^*(s) =$$

What to do?

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

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

What to do?

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

$$U^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s]$$

What to do?

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

$$U^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s]$$

The key:

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

(Richard Bellman, 1957)

Break

- asst 3
- textbook
- final project proposals due next Monday in class
  - project handout
  - convince me it's interesting and doable
- project presentations: Wed May 9, 9am-noon

Solving MDPs

Value Iteration

Repeated Bellman updates:

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

Value Iteration

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, guaranteed to reach equilibrium.
Equilibrium is unique solution to Bellman equations!

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!