Decision Theory: Markov Decision Processes

CPSC 322 – Decision Theory 3

Textbook §12.5

Lecture Overview

1. Recap
2. Finding Optimal Policies
3. Value of Information, Control
4. Markov Decision Processes
5. Rewards and Policies

Policies

- A policy specifies what an agent should do under each circumstance.
- A policy is a sequence $\delta_1, \ldots, \delta_n$ of decision functions

$$\delta_i : dom(pD_i) \rightarrow dom(D_i).$$

This policy means that when the agent has observed $O \in dom(pD_i)$, it will do $\delta_i(O)$.

Expected Value of a Policy

- Possible world ω satisfies policy δ, written $\omega \models \delta$, if the world assigns the value to each decision node that the policy specifies.
- The expected utility of policy δ is

$$\mathbb{E}(U|\delta) = \sum_{\omega \models \delta} P(\omega)U(\omega)$$

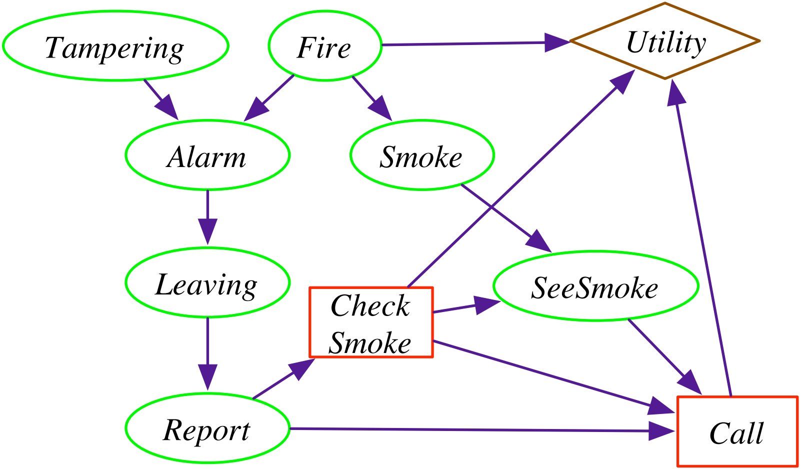
- An optimal policy is one with the highest expected utility:

$$\delta^* \in \arg\max_{\delta} \mathbb{E}(U|\delta).$$

Counting Policies

- If a decision D has k binary parents, how many assignments of values to the parents are there? 2^k
- If there are b possible actions, how many different decision functions are there? b^{2^k}
- If there are d decisions, each with k binary parents and b possible actions, how many policies are there? $\left(b^{2^k}\right)^d$

Decision Network for the Alarm Problem



Lecture Overview

Finding the optimal policy

- Remove all variables that are not ancestors of a value node
- Create a factor for each conditional probability table and a factor for the utility.
- Sum out variables that are not parents of a decision node.
- Select a variable D that is only in a factor f with (some of) its parents.
  - this variable will be one of the decisions that is made latest
- Eliminate D by maximizing. This returns:
  - the optimal decision function for D, $\arg\max_D f$
  - a new factor to use in VE, $\max_D f$
- Repeat till there are no more decision nodes.
- Sum out the remaining random variables. Multiply the factors: this is the expected utility of the optimal policy.

Complexity of finding the optimal policy

- Recall: If there are d decisions, each with k binary parents and b possible actions, there are $\left(b^{2^k}\right)^d$ policies
- Doing variable elimination lets us find the optimal policy after considering only $d \cdot b^{2^k}$ policies
- The dynamic programming algorithm is much more efficient than searching through policy space.
- However, this complexity is still doubly-exponential—we'll only be able to handle relatively small problems.

Lecture Overview

Value of Information

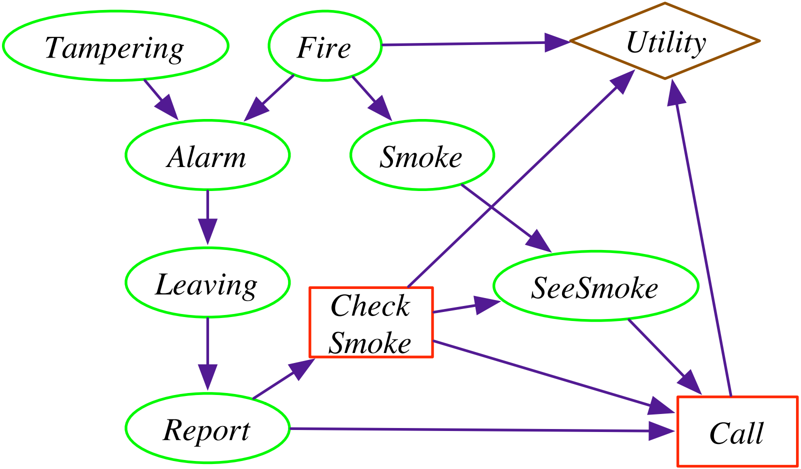
- How much you should be prepared to pay for a sensor?
  - E.g., how much is a better weather forecast worth?

Definition (Value of Information)

The value of information X for decision D is the utility of the the network with an arc from X to D minus the utility of the network without the arc.

- The value of information is always non-negative.
- It is positive only if the agent changes its action depending on X.

We could ask about the value of information for Smoke



Value of Control

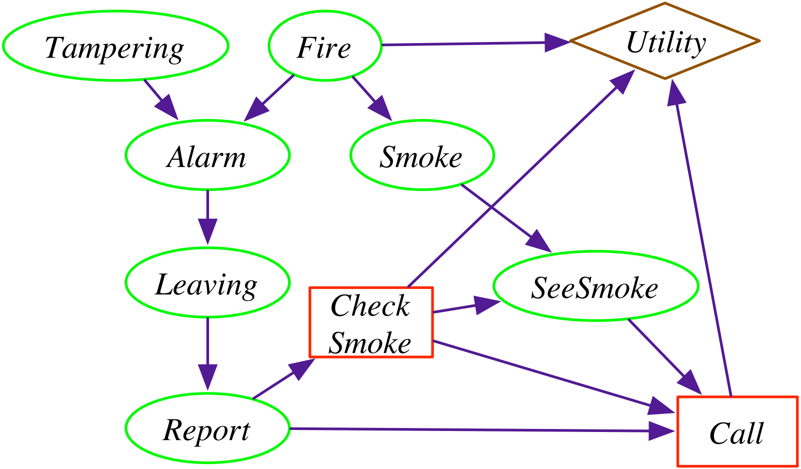
- How useful is it to be able to set a random variable?

Definition (Value of Control)

The value of control of a variable X is the value of the network when you make X a decision variable minus the value of the network when X is a random variable.

- You need to be explicit about what information is available when you control X.
  - If you control X without observing, controlling X can be worse than observing X.
  - If you keep the parents the same, the value of control is always non-negative.

We could ask about the value of control for Tampering



Lecture Overview

Agents as Processes

Agents carry out actions:

- forever: infinite horizon
- until some stopping criteria is met: indefinite horizon
- finite and fixed number of steps: finite horizon

Decision-theoretic Planning

What should an agent do under these different planning horizons, when

- actions can be noisy
  - the outcome of an action can't be fully predicted
  - there is a stationary, Markovian model that specifies the (probabilistic) outcome of actions
- the world (i.e., state) is fully observable
- the agent periodically gets rewards (and punishments) and wants to maximize its rewards received

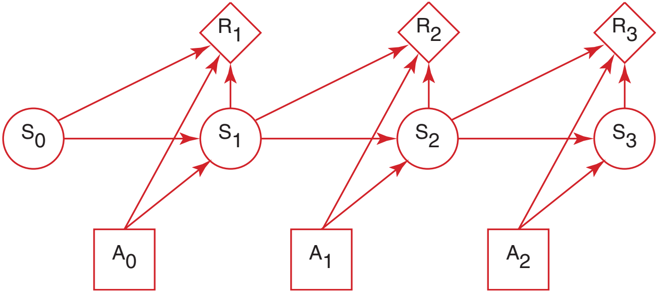
Stationary Markov chain

Start with a stationary Markov chain.

- Recall: a stationary Markov chain is when for all $t > 0$, $P(S_{t+1}|S_t) = P(S_{t+1}|S_0, \ldots, S_t)$.
- We specify $P(S_0)$ and $P(S_{t+1}|S_t)$.

Decision Processes

- A Markov decision process augments a stationary Markov chain with actions and values:



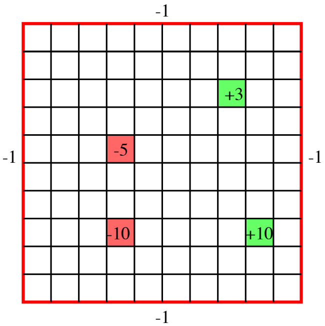
Markov Decision Processes

Definition (Markov Decision Process)

A Markov Decision Process (MDP) is a 5-tuple $\langle S, A, P, R, s_0 \rangle$, where each element is defined as follows:

- S: a set of states.
- A: a set of actions.
- $P(S_{t+1}|S_t, A_t)$: the dynamics.
- $R(S_t, A_t, S_{t+1})$: the reward. The agent gets a reward at each time step (rather than just a final reward).
  - $R(s, a, s')$ is the reward received when the agent is in state s, does action a and ends up in state s'.
- s_0: the initial state.

Example: Simple Grid World



- Actions: up, down, left, right.
- 100 states corresponding to the positions of the robot.
- Robot goes in the commanded direction with probability 0.7, and one of the other directions with probability 0.1.
- If it crashes into an outside wall, it remains in its current position and has a reward of -1.
- Four special rewarding states; the agent gets the reward when leaving.

Planning Horizons

The planning horizon is how far ahead the planner can need to look to make a decision.

- The robot gets flung to one of the corners at random after leaving a positive (+10 or +3) reward state.
  - the process never halts
  - infinite horizon
- The robot gets +10 or +3 entering the state, then it stays there getting no reward. These are absorbing states.
  - The robot will eventually reach the absorbing state.
  - indefinite horizon

Information Availability

What information is available when the agent decides what to do?

- fully-observable MDP the agent gets to observe S_t when deciding on action A_t.
- partially-observable MDP (POMDP) the agent has some noisy sensor of the state. It needs to remember its sensing and acting history.

We'll only consider (fully-observable) MDPs.

Lecture Overview

Rewards and Values

Suppose the agent receives the sequence of rewards $r_1, r_2, r_3, r_4, \ldots$. What value should be assigned?

- total reward:

$$V = \sum_{i=1}^{\infty} r_i$$

- average reward:

$$V = \lim_{n \to \infty} \frac{r_1 + \cdots + r_n}{n}$$

- discounted reward:

$$V = \sum_{i=1}^{\infty} \gamma^{i-1} r_i$$

  - γ is the discount factor, $0 \leq \gamma \leq 1$