Decision Theory: Sequential Decisions

CPSC 322 – Decision Theory 2

Textbook §12.3

Lecture Overview

1. Recap
2. Sequential Decisions
3. Finding Optimal Policies

Decision Variables

- Decision variables are like random variables that an agent gets to choose the value of.
- A possible world specifies the value for each decision variable and each random variable.
- For each assignment of values to all decision variables, the measures of the worlds satisfying that assignment sum to 1.
- The probability of a proposition is undefined unless you condition on the values of all decision variables.

Single decisions

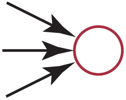
- Given a single decision variable, the agent can choose $D = d_i$ for any $d_i \in dom(D)$.
- The expected utility of decision $D = d_i$ is $\mathbb{E}(U|D = d_i)$.
- An optimal single decision is the decision $D = d_{max}$ whose expected utility is maximal:

$$d_{max} = \arg\max_{d_i \in dom(D)} \mathbb{E}(U|D = d_i).$$

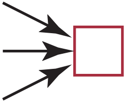
Decision Networks

- A decision network is a graphical representation of a finite sequential decision problem.
- Decision networks extend belief networks to include decision variables and utility.
- A decision network specifies what information is available when the agent has to act.
- A decision network specifies which variables the utility depends on.

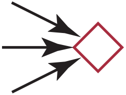
Decision Networks



- A random variable is drawn as an ellipse. Arcs into the node represent probabilistic dependence.



- A decision variable is drawn as an rectangle. Arcs into the node represent information available when the decision is made.



- A value node is drawn as a diamond. Arcs into the node represent values that the value depends on.

Lecture Overview

1. Recap
2. Sequential Decisions
3. Finding Optimal Policies

Sequential Decisions

- An intelligent agent doesn't make a multi-step decision and carry it out without considering revising it based on future information.
- A more typical scenario is where the agent:
  observes, acts, observes, acts, . . .
  - just like your final homework!
- Subsequent actions can depend on what is observed.
  - What is observed depends on previous actions.
- Often the sole reason for carrying out an action is to provide information for future actions.
  - For example: diagnostic tests, spying.

Sequential decision problems

- A sequential decision problem consists of a sequence of decision variables $D_1, \ldots, D_n$.
- Each D_i has an information set of variables pD_i, whose value will be known at the time decision D_i is made.

- What should an agent do?
  - What an agent should do at any time depends on what it will do in the future.
  - What an agent does in the future depends on what it did before.

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 $\delta_i(O)$.

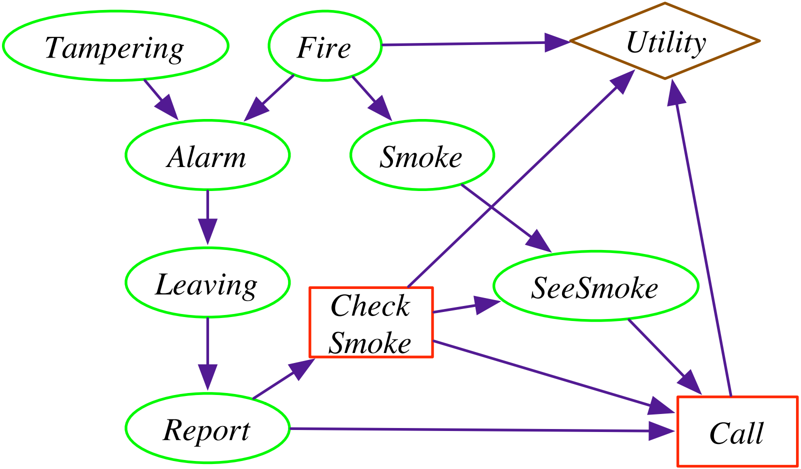
Expected Value of a Policy

- 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.

Decision Network for the Alarm Problem



Lecture Overview

1. Recap
2. Sequential Decisions
3. Finding Optimal Policies

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

- If a decision D has k binary parents, how many assignments of values to the parents are there?

Complexity of finding the optimal policy

- If a decision D has k binary parents, how many assignments of values to the parents are there? 2^k

Complexity of finding the optimal policy

- 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?

Complexity of finding the optimal policy

- 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}

Complexity of finding the optimal policy

- 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?

Complexity of finding the optimal policy

- 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? $\left(b^{2^k}\right)^d$

Complexity of finding the optimal policy

- 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? $\left(b^{2^k}\right)^d$
- 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.