CS761 Artificial Intelligence

20. Planning with Uncertainty: Decision Networks

Jiamou Liu
The University of Auckland

Recap: Planning with Uncertainty

Classical planning is a planning task with the following assumptions:

- Finite set of states as the search space.
- State changes defined only by agents' actions.
- Deterministic actions: Each action in a state has one outcome, which can be foreseen by the agent.
- Perfect information
- Goals must be achieved.

Planning with uncertainty differs from classical planning:

- Infinite set of probabilistic outcomes as search space.
- State changes stochastically.
- Non-deterministic actions.
- Imperfect information
- Goals may not be achieved. Need to evaluate desirability of outcomes.

One-off Decision Problem

One-off decision problem

- A set of decision variables $V_1, \ldots, V_k$
- A set of random variables $R_1, \ldots, R_\ell$
- State space $S = \prod_{i=1}^{k} \mathrm{dom}(V_i) \times \prod_{j=1}^{\ell} \mathrm{dom}(R_j)$.
- Conditional probability $\mathbf{P}(R_1, \ldots, R_\ell \mid V_1, \ldots, V_k)$.
- Preferences/utility over S.
- **Goal:** Choose values for $(V_1, \ldots, V_k)$ to maximise the expected utility, i.e., $\arg\max_{(v_1,\ldots,v_k)\in\prod_{i=1}^{k}\mathrm{dom}(i)} E(u \mid v_1, \ldots, v_k)$ where $E(u \mid v_1, \ldots, v_k)$ is

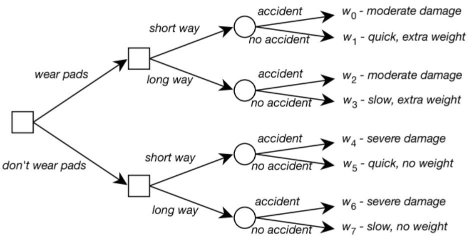
$$\sum_{(r_1,\ldots,r_\ell)\in\prod_{j=1}^{\ell}\mathrm{dom}(R_j)} P(r_1, \ldots, r_\ell \mid v_1, \ldots, v_k) u(v_1, \ldots, v_k, r_1, \ldots, r_\ell)$$

One-off decision problem can be described by two types of knowledge representations:

- **Decision tree:** factored representation
- **Decision network:** structured representation (to be covered in this lecture)

Knowledge representation 1: Decision trees. A factored representation that connects variables with outcomes, but not dependencies between variables.

Example. [delivery robot] A decision tree illustrates the connections of actions to outcomes.



Conditional probability:

Way	Acc	**P**$(Acc \mid Way)$
0	0	0.8
0	1	0.2
1	0	0.99
1	1	0.01

Utility function: $u : S \to \mathbb{R}$

Pad	Way	Acc	Outcome	$u(Pad, Way, Acc)$
0	0	0	w_5	100
0	0	1	w_4	3
0	1	0	w_7	80
0	1	1	w_6	0
1	0	0	w_1	95
1	0	1	w_0	35
1	1	0	w_3	75
1	1	1	w_2	30

Knowledge representation 2: Single-stage decision networks. A structured representation which represents dependencies relations among (decision and random) variables.

Definition

A single-stage decision network (SSDN) is an extension of a Bayesian network with three kinds of nodes:

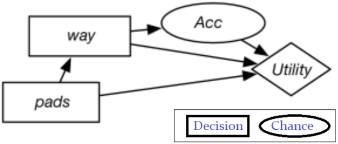
- Decision nodes, represent decision variables. If there are multiple decision nodes, then they are arranged in a total ordering

  $$D_1, D_2, D_3, \ldots, D_k$$

  The parents of a decision node D_i are $D_1, \ldots, D_{i-1}$.
- Chance nodes $C_1 \ldots, C_\ell$, represent random variables. Each chance node has a conditional probability, given its parents.
- A single utility node, represent the utility.

Note: The chance nodes take the same role as nodes in a Bayesian network with local Markov property.

Example. [delivery robot] The example above has the following decision network.



Definition

- A policy for a single-stage decision network is an assignment of a value to each decision variable. Each policy has an expected policy.
- An optimal policy is a policy whose expected utility is maximal.
- The value of a decision network is the expected utility of an optimal policy of the network.

The SSDN-based one-off decision problem asks for the optimal policy and the value of a given single-stage decision network.

We can extend the CPT operations to the set of CPT and the utility function table:

- Restriction
- Multiplication
- *X*-Sum

Example.
Multiplication:

Way	Acc	$\mathbf{P}(Acc \mid Way)$
0	0	0.8
0	1	0.2
1	0	0.99
1	1	0.01

$\times$

Pad	Way	Acc	$u(Pad, Way, Acc)$
0	0	0	100
0	0	1	3
0	1	0	80
0	1	1	0
1	0	0	95
1	0	1	35
1	1	0	75
1	1	1	30

$=$

Pad	Way	Acc	$f(Pad, Way, Acc)$
0	0	0	0.8×100
0	0	1	0.2×3
0	1	0	0.99×80
0	1	1	0.01×0
1	0	0	0.8×95
1	0	1	0.2×35
1	1	0	0.99×75
1	1	1	0.01×30

Acc-Sum:

Pad	Way	$E(u \mid Pad, Way)$
0	0	$0.2 \times 3 + 0.8 \times 100 = 80.6$
0	1	$0.01 \times 0 + 0.99 \times 80 = 79.2$
1	0	$0.2 \times 35 + 0.8 \times 95 = 83$
1	1	$0.01 \times 30 + 0.99 \times 75 = 74.55$

Note. We will refer to the utility function table as a “CPT” as well.

We can then extend the variable elimination algorithm for Bayesian networks to SSDN.

VariableEliminationSSDN(*SSDN*)

INPUT: *SSDN*
OUTPUT: An optimal policy and the value of *SSDN*.

Prune all nodes that are not ancestors of the utility node.
for every chance node C **do**
    Multiply all CPTs that contains C
    Perform C-Sum to the CPT to eliminate C
end for
▷ Now there is a single CPT left derived from utility.
Let v be the maximum value in the last CPT.
Let d be the assignment that gives the maximum value.
return d, v

Note about SSDN.

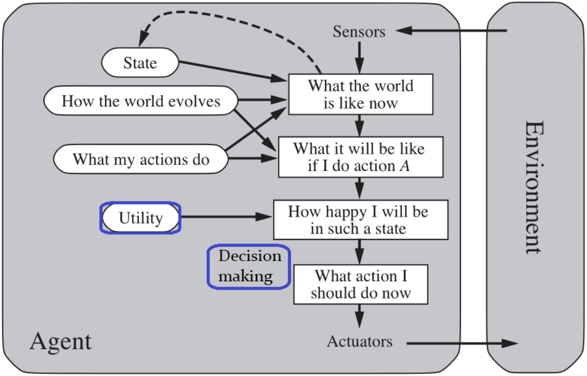
- No chance node can be a parent of a decision node.
- Decisions are made *without* the agent making any observation.

Sequential Decision Problem

Sequential decision problem: The agent makes observations before making decisions, which affects further observations, and then further decisions, etc.

$$\mathit{Observations}_1 \rightsquigarrow \mathit{Decisions}_1 \rightsquigarrow \mathit{Observations}_2 \rightsquigarrow \mathit{Decisions}_2 \rightsquigarrow \cdots \rightsquigarrow \mathit{Outcome}$$

Utility-based agent.



For sequential decision problem, we can use two types of knowledge representations:

- Finite horizon: Decision network
  There is a fixed sequence of decisions to be made.
- Indefinite horizon or infinite horizon: Markov decision process (to be covered in the next lecture)
  The agent does not know a priori the sequence and number of decisions to be made.

For sequential decision problem, we can use two types of knowledge representations:

- Finite horizon: Decision network
  There is a fixed sequence of decisions to be made.
- Indefinite horizon or infinite horizon: Markov decision process (to be covered in the next lecture)
  The agent does not know a priori the sequence and number of decisions to be made.

Definition

A decision network is a probabilistic graph model that extends a SSDN to allow both chance nodes and decision nodes to be parents of decision nodes.

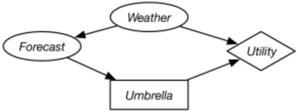
- The graph is acyclic.
- Decision nodes are ordered as $D_1, \dots, D_k$.
- (no forgetting property) If an edge connects a node X to D_i, then there is an edge from X to all nodes D_j where $j > i$[a].
- A CPT $\mathbf{P}(C_i \mid \text{parents}(C_i))$ is assigned to every chance node.
- A utility function $u(\text{parents}(u))$ is given.

[a] In this course we always assume no forgetting property for all DN

Example. [weather] Should the agent take an umbrella?

- Chance node: $Weather$: $\{norain, rain\}$, $Forecast$: $\{sunny, cloudy, rainy\}$
- Decision node: $Umbrella$: $\{0, 1\}$.

The problem has the following decision network:



Weather	Umbrella	$u(Weather, Umbrella)$
norain	1	20
norain	0	100
rain	1	70
rain	0	0

Weather	$\mathbf{P}(Weather)$
norain	0.7
rain	0.3

Weather	Forecast	$\mathbf{P}(Forecast \mid Weather)$
norain	sunny	0.7
norain	cloudy	0.2
norain	rainy	0.1
rain	sunny	0.15
rain	cloudy	0.25
rain	rainy	0.6

There are three types of edges:

- Into a decision node: available information (observation/memory) when the decision is made.
- Into a chance node: probabilistic dependence.
- Into the utility node: dependence of the utility function.

Definition

- A decision function for a decision node D_i is a function

$$d_i\colon \prod_{D_j \in \text{parents}(D_i)} \text{dom}(D_j) \to \text{dom}(D_i)$$

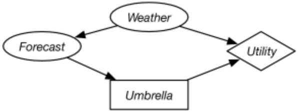
  which maps all possible values of parents(D_i) to dom(D_i).
- A policy π consists of a decision function for every decision node $D_1, \ldots, D_k$.
- States $\prod_{\text{decision node } D_i} \text{dom}(D_i) \times \prod_{\text{chance node } C_j} \text{dom}(C_j)$
- A state ω satisfies policy π if every decision node D has the same value as specified by π, given the values of its parents in ω.
- The expected utility of policy π is

$$E(u \mid \pi) = \sum_{\omega \text{ satisfies } \pi} u(\omega) \times P(\omega)$$

The DN-based sequential decision problem asks for the optimal policy (which maximises the expected utility) and the value of a given decision network.

Example. [weather] A policy is a function $\pi\colon \{sunny, cloudy, rainy\} \rightarrow \{0, 1\}$. E.g.

- Always bring the umbrella. $\pi(sunny) = \pi(cloudy) = \pi(rainy) = 1$
- Bring the umbrella only if the forecast is rainy. $\pi(rainy) = 1$, $\pi(sunny) = \pi(cloudy) = 0$.



Suppose $\pi_1(sunny) = \pi_1(rainy) = 0$, and $\pi_1(cloudy) = 1$.

The states that satisfy π_1:

Weather	Forecast	Umbrella	$u(s) \times P(s)$
norain	sunny	0	$P(norain)P(sunny \mid norain)u(norain, 0) = 0.7 \times 0.7 \times 100$
norain	cloudy	1	$P(norain)P(cloudy \mid norain)u(norain, 1) = 0.7 \times 0.2 \times 20$
norain	rainy	0	$P(norain)P(rainy \mid norain)u(norain, 0) = 0.7 \times 0.1 \times 100$
rain	sunny	0	$P(rain)P(sunny \mid rain)u(rain, 0) = 0.3 \times 0.15 \times 0$
rain	cloudy	1	$P(rain)P(cloudy \mid rain)u(rain, 1) = 0.3 \times 0.25 \times 70$
rain	rainy	0	$P(rain)P(rainy \mid rain)u(rain, 0) = 0.3 \times 0.6 \times 0$

Expected utility:

$$E(u \mid \pi_1) = 49 + 2.8 + 7 + 0 + 5.25 = 64.05$$

We can extend the variable elimination (VE) algorithm for SSDN to decision networks.

Idea: Repeatedly "eliminate decisions"

- Start from the *last* decision node
- Find an optimal decision for this node
- Reduce this decision node

VariableEliminationDN(DN)

INPUT: Decision network DN
OUTPUT: An optimal policy and the value of DN.

Create a set DF of decision functions, initially $\varnothing$
Remove all variables that are not ancestors of the utility node
while there are decision nodes remaining **do**
  Eliminate each random variable that is not a parent of a decision node
  Let D be the last decision node
  Multiply all tables that contains D
                ▷ Now D appear in only one (utility) table F_u
  Add $\arg\max_D F_u$ to DF
end while
Return DF and the product of the remaining tables.
return d, v

Example. [weather]

Step 1. Eliminate *Weather*:

Weather	P
norain	0.7
rain	0.3

×

Weather	Forecast	P
norain	sunny	0.7
norain	cloudy	0.2
norain	rainy	0.1
rain	sunny	0.15
rain	cloudy	0.25
rain	rainy	0.6

×

Weather	Umbrella	u
norain	1	20
norain	0	100
rain	1	70
rain	0	0

=

Forecast	Umbrella	F_u
sunny	0	49.0
sunny	1	12.95
cloudy	0	14.0
cloudy	1	8.05
rainy	0	7.0
rainy	1	14.0

Step 2. Eliminate *Umbrella*

Forecast	F_u
sunny	49.0
cloudy	14.0
rainy	14.0

Optimal policy: $\pi^*(sunny) = 0, \pi^*(cloudy) = 0, \pi^*(rainy) = 1.$

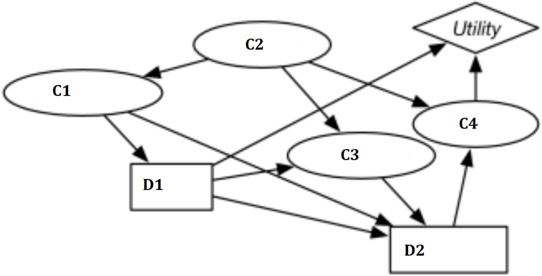
Step 3. Sum to retrieve value: $49 + 14 + 14 = 77.$

Note.

- The example above only contains one decision node.
- In many cases, the decision network contains many decision nodes: $D_1, \ldots, D_k$.

Example. A DN with two decision nodes:

Ordering of elimination: $D_2 \rightsquigarrow D_1$

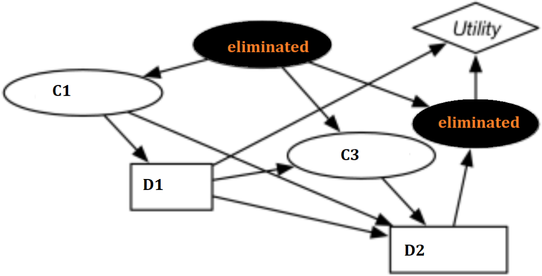


Note.

- The example above only contains one decision node.
- In many cases, the decision network contains many decision nodes: $D_1, \ldots, D_k$.

Example. A DN with two decision nodes:

Ordering of elimination: $D_2 \rightsquigarrow D_1$

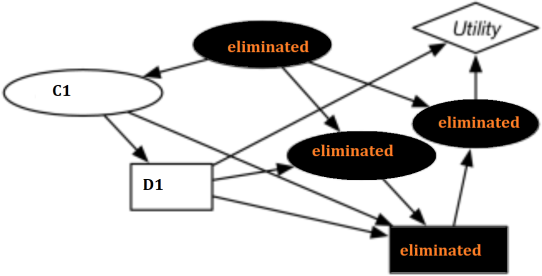


Note.

- The example above only contains one decision node.
- In many cases, the decision network contains many decision nodes: $D_1, \ldots, D_k$.

Example. A DN with two decision nodes:

Ordering of elimination: $D_2 \rightsquigarrow D_1$

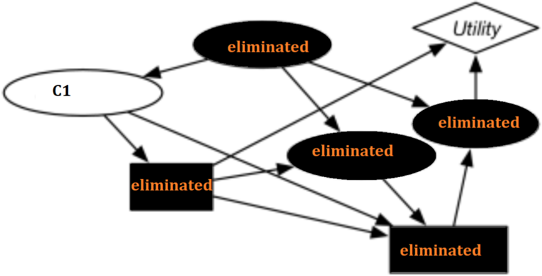


Note.

- The example above only contains one decision node.
- In many cases, the decision network contains many decision nodes: $D_1, \ldots, D_k$.

Example. A DN with two decision nodes:

Ordering of elimination: $D_2 \rightsquigarrow D_1$



The Value of Information

- The agent has imperfect information, i.e., the environment is only partially-observable.
- Having more information is beneficial to making a better decision.
- **Question.** What is the value of information?

Definition

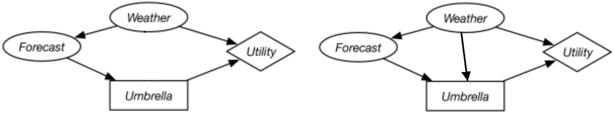
In a decision network, assume that

- X is a random variable, D is a decision variable;
- Having an edge from X to D would not create a cycle.

The value of information (VOI) about random variable X for decision D is $V' - V$ where

- V' is the value of the DN with an edge added from X to D and all decision after D;
- V is the value of the same DN without the edge from X to D.

Example. [weather] Consider two instances of the weather domain:



- On the RHS, *Weather* is an extra information for decision *Umbrella*:
  - **Optimal policy:** $\pi^*(norain) = 0$, $\pi^*(rain) = 1$.
  - **Value:** $\mathbf{P}(Weather)E(u \mid Weather) = 0.7 \times 100 + 0.3 \times 70 = 91$.
- Recall the value for the DN on the LHS, 77.
- VOI of *Weather* for *Umbrella*: $91 - 77 = 14$.

Property of the value of information:

- The value of information is always non-negative.
- If an optimal decision is to do the same thing no matter the value of X, then VOI is 0.

The Value of Control

- The environment is stochastic. The agent has no control over random variables.
- Having more control is beneficial to having a better outcome.
- **Question.** What is the value of control?

Definition

In a decision network, assume that X is a random variable.
The value of control (VOI) about random variable X is $V' - V$ where

- V' is the value of the DN with the random variable X replaced by a decision node;
- V is the value of the original DN.

Decision-theoretic Expert Systems

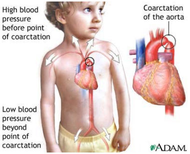
A decision-theoretic expert system is an automated tool that recommend decisions that reflect the preferences of agents as well as the available evidence. Applications: business, government, law, medical diagnosis, military strategy, management, etc.

We now demonstrate the process of designing a decision-theoretic expert system.

Example. Aortic coarctation is a kind of heart problem and can be treated by:

- surgery
- angioplasty
- medication

Question. How to build an expert system that decides on the optimal treatment?



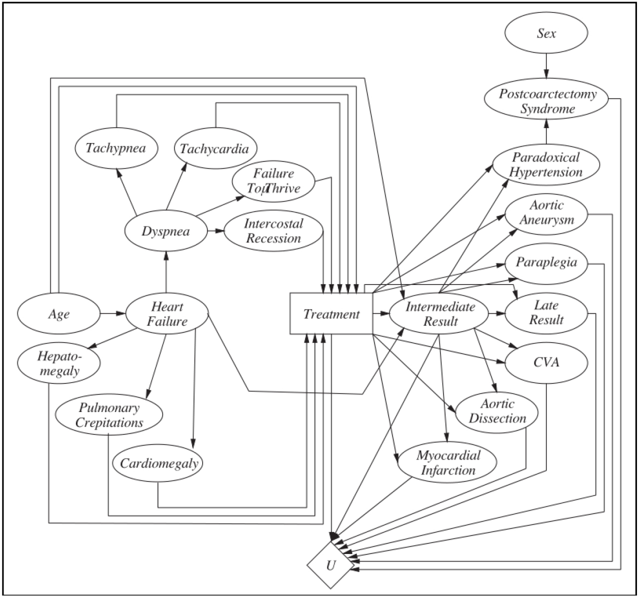
Step 1. Create a causal model.
From domain knowledge: Determine the possible symptoms, disorders, treatments, and outcomes. Draw edges.

Step 2. Simplify to a qualitative decision model.
Removing variables that are not involved in treatment decisions.
Step 3. Assign probabilities.
From literature/expert/data: conditional probabilities of random variables.

Step 4. Assign utilities.
From expert/patients: aggregated preferences of different outcomes.

Step 5. Verify and refine the model.
From expert: gold standard (input,output) of treatment plans.

Step 6. Perform sensitivity analysis.
Check if the best decision is sensitive to small changes in the assigned probabilities and utilities by systematically varying parameters.
If all variations lead to the same decision, then more confidence that it is the right decision.


Sex
Postcoarctectomy Syndrome
Tachypnea
Tachycardia
Failure To Thrive
Paradoxical Hypertension
Aortic Aneurysm
Dyspnea
Intercostal Recession
Paraplegia
Age
Heart Failure
Treatment
Intermediate Result
Late Result
Hepato-megaly
CVA
Pulmonary Crepitations
Aortic Dissection
Cardiomegaly
Myocardial Infarction
U

Summary of The Topic

The following are the main knowledge points covered:

- **One-off decision problem:**
  - Decision tree
  - Single-stage decision network
- Single-stage decision network (SSDN):
  - Decision node, Chance node, Utility node
  - Policy
  - SSDN optimal policy problem: Optimal policy, value
  - Variable elimination for SSDN: Extending CPT operations to utility function table.
- **Sequential decision problem:**
  - Decision network
  - Markov decision process
- Decision network (DN):
  - Chance node can be a parent of decision node
  - No-forgetting property
  - Decision function, policy, expected utility of policy
  - DN optimal policy problem
  - Variable elimination for DN: Eliminate decision nodes (from last to first)
- Value of information and control
- Decision-theoretic expert systems