

CS 486/686

Decision Networks

Yuntian Deng

Lecture 12

RN 16.5–16.6 · PM 9.1–9.4

Search ›

Uncertainty ›

Decisions ›

Learning

Learning goals

- **Model** a one-off decision problem as a decision network (chance / decision / utility nodes).
- Find the **optimal policy** by computing the expected utility of every policy.
- Find the **optimal policy** via the variable elimination algorithm.
- Quantify the **value of information**: what's a piece of evidence worth?
- Apply VEA to a multi-decision network with an information ordering.

Decision theory: how should an agent act?

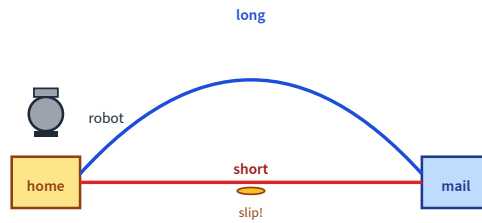
- **Probability theory:** what should the agent *believe* from the evidence?
- **Utility theory:** what does the agent *want*? (Assign a real number to each world.)

Maximum expected utility (MEU)

A rational agent picks the action that **maximizes its expected utility**.

Decision network = Bayes net + actions + utilities.

Running example: a mail-pickup robot



A robot must fetch the mail. There's a **short, slippery** route and a **long, safe** route. It can wear **pads** (slow it down, but reduce damage if it slips).

What should the robot do?

Three kinds of nodes in a decision network



Chance node

random variable (BN)



Decision node

action (we choose)

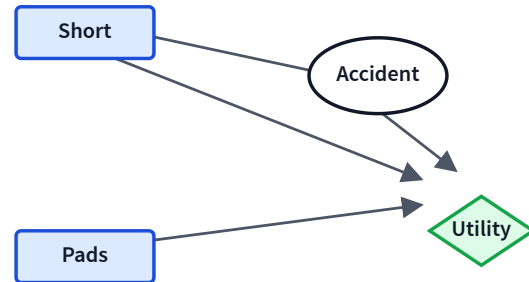


Utility node

agent's happiness

For the robot: A (chance) = whether an accident occurs; P (decision) = wear pads;
 S (decision) = take short route; U (utility) = the robot's happiness.

The robot's decision network



$P(A|\neg S) = 0$, $P(A|S) = q \cdot \text{arcs to Utility revealed on click.}$

Q1. Which variables directly influence the robot's happiness?

- A. P only
- B. S only
- C. A only
- D. Two of (A), (B), (C)
- E. **All of** (A), (B), (C)

E. Pads (weight), Short (speed), Accident (damage) all matter — all three are parents of Utility.

The robot's utility table

P	S	A	Description	U
$\neg P$	$\neg S$	$\neg A$	slow, no weight	6
$\neg P$	$\neg S$	A	<i>impossible (long route + accident)</i>	—
$\neg P$	S	$\neg A$	quick, no weight	10
$\neg P$	S	A	severe damage	0
P	$\neg S$	$\neg A$	slow, extra weight	4
P	$\neg S$	A	<i>impossible</i>	—
P	S	$\neg A$	quick, extra weight	8
P	S	A	moderate damage	2

Pads slow the robot down (lower utility), but cushion impact in an accident.

Reading the utility function

Q2. When an accident does *not* happen, which is true?

- A. Robot prefers no pads over pads.
- B. Robot prefers long route over short.
- C. Both (A) and (B).
- D. Neither (A) nor (B).

A. No pads is faster ($6 > 4$, $10 > 8$).
But the robot *does* prefer the short route ($10 > 6$).

When an accident does happen?

- The robot must have taken the short route (long route never crashes).
- Now the robot prefers **wearing pads**:
 $U(P, S, A) = 2 > 0 = U(\neg P, S, A)$.
Pads cushion the fall.

Evaluating a decision network

How do we choose the optimal action?

evaluate-decision-network(DN, evidence)

1. Fix every evidence variable to its observed value.
2. For each assignment to the decision variables: compute the posterior of the utility node's parents, then compute the expected utility.
3. Return the assignment with the maximum expected utility.

For the robot: enumerate the 4 combinations of (P, S) .

Four expected utilities

Recall $P(A|\neg S) = 0$, $P(A|S) = q$. For each (P, S) , $EU(P, S) = \sum_a P(a|S) U(P, S, a)$.

$$EU(\neg P, \neg S)$$

$$(1)(6) + (0)(-)$$
$$= 6$$

$$EU(\neg P, S)$$

$$(1 - q)(10) + (q)(0)$$
$$= 10 - 10q$$

$$EU(P, \neg S)$$

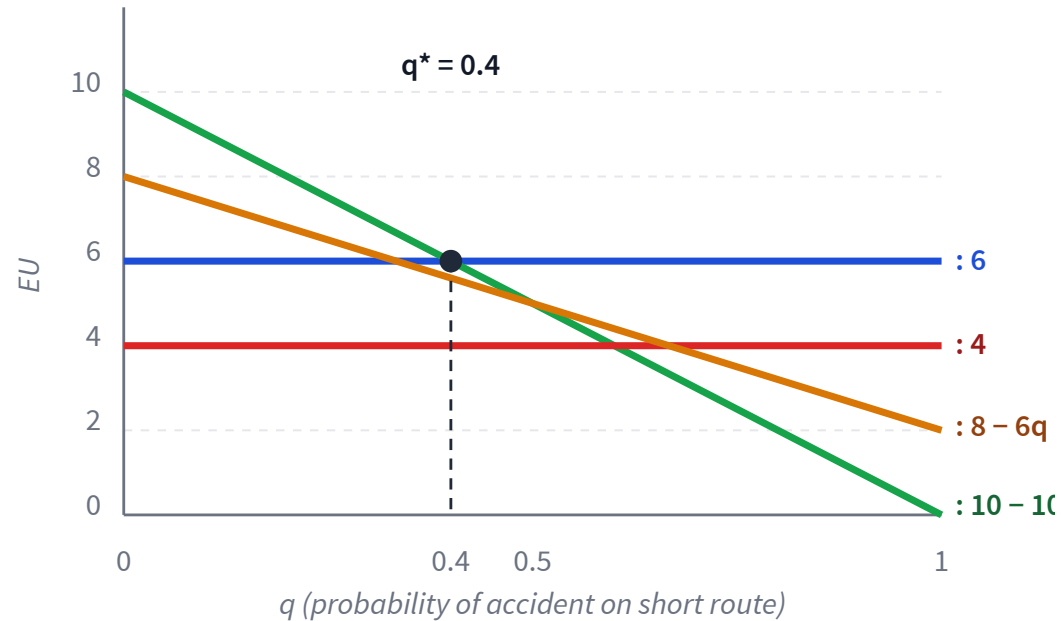
$$(1)(4) + (0)(-)$$
$$= 4$$

$$EU(P, S)$$

$$(1 - q)(8) + (q)(2)$$
$$= 8 - 6q$$

Now plot all four as functions of q , and pick the upper envelope.

Optimal policy depends on the risk q



Optimal policy

- Never wear pads.
- If $q \leq 0.4$: take the **short** route.
- If $q > 0.4$: take the **long** route.

The two upper lines ($\neg P, S$ and $\neg P, \neg S$) cross at $10 - 10q = 6 \Rightarrow q = 0.4$.

Variable elimination for a single-stage DN

Same factor algebra as L9, with a **max** instead of normalization at the end.

VEA-single-stage(DN, evidence)

1. Prune nodes that are not ancestors of the utility.
2. Sum out every chance (random) variable.
3. For the remaining factor over decisions, return the assignment with the maximum value.

If multiple decisions can be made *simultaneously*, merge them into one composite node.

Combine decisions, define factors

Step 0. Since P and S are chosen together, merge them into one 4-valued decision

$S \wedge P$. Two factors remain:

$f_1(A, S \wedge P)$ –
chance

A	$S \wedge P$	val
t	S, P	q
f	S, P	$1 - q$
t	$S, \neg P$	q
f	$S, \neg P$	$1 - q$
t	$\neg S, P$	0
f	$\neg S, P$	1
t	$\neg S, \neg P$	0
f	$\neg S, \neg P$	1

$u(A, S \wedge P)$ –
utility

A	$S \wedge P$	val
t	S, P	2
f	S, P	8
t	$S, \neg P$	0
f	$S, \neg P$	10
t	$\neg S, P$	–
f	$\neg S, P$	4
t	$\neg S, \neg P$	–
f	$\neg S, \neg P$	6

Multiply, sum out A , then maximize

Steps 2-3. Multiply: $f_2 = f_1 \times u$. Then sum out A to get $f_3(S \wedge P)$:

$$f_2 = f_1 \times u$$

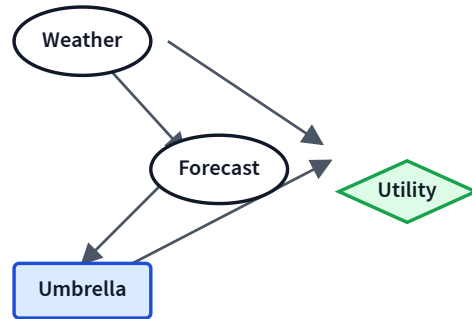
A	$S \wedge P$	val
t	S, P	$2q$
f	S, P	$8 - 8q$
t	$S, \neg P$	0
f	$S, \neg P$	$10 - 10q$
t	$\neg S, P$	0
f	$\neg S, P$	4
t	$\neg S, \neg P$	0
f	$\neg S, \neg P$	6

$$f_3(S \wedge P) = \sum_A f_2$$

$S \wedge P$	val
S, P	$8 - 6q$
$S, \neg P$	$10 - 10q$
$\neg S, P$	4
$\neg S, \neg P$	6

argmax over the 4 rows reproduces slide 11: $\neg P, S$ if $q \leq 0.4$; $\neg P, \neg S$ otherwise.

The weather decision network



$P(W)$

W	val
norain	0.7
rain	0.3

$P(F | W)$

W	F	val
norain	sunny	0.7
norain	cloudy	0.2
norain	rainy	0.1
rain	sunny	0.15
rain	cloudy	0.25
rain	rainy	0.6

$u(W, Um)$

W	Um	val
norain	take	20
norain	leave	100
rain	take	70
rain	leave	0

Policies

A **policy** specifies the agent's action for every assignment of the decision node's parents.

Q3. How many policies are there for the weather DN?

- A. 1
- B. 3
- C. 6
- D. 8

D — 8. Forecast has 3 values (sunny, cloudy, rainy); each maps to one of 2 actions (take / leave). So $2^3 = 8$ policies.

Method 1: enumerate the policies

For any policy π , $EU(\pi) = \sum_{w,f} P(w) P(f|w) u(w, \pi(f))$. Compare two:

π_1 : take if cloudy, else leave

Cover only cloudy days; rainy days are missed.

$$EU(\pi_1) = 64.05$$

π_2 : take if rainy, else leave

Cover only rainy days; cloudy days are missed.

$$EU(\pi_2) = 77$$

Repeat for all 8 policies; pick the max. Tedious; let's use VEA instead.

Method 2: VEA on the weather DN

VEA-multi-stage(DN, evidence)

1. Prune non-ancestors of Utility. (*Nothing to prune here.*)
2. Define factors:
 $f_1(W), f_2(W, F), f_3(W, U)$.
3. While decisions remain: eliminate chance vars not parents of any decision, then max over the next decision (in info order).
4. Sum out the remaining chance vars to get the EU.

$$\text{Eliminate } W: f_1 \times f_2 \times f_3 \rightarrow f_4(W, F, U) \rightarrow f_5(F, U)$$

multiply, then \sum_W



Max over U : for each F , keep best U . π^* : sunny→leave (49), cloudy→leave (14), rainy→take (14).
 $\Rightarrow f_6(F)$

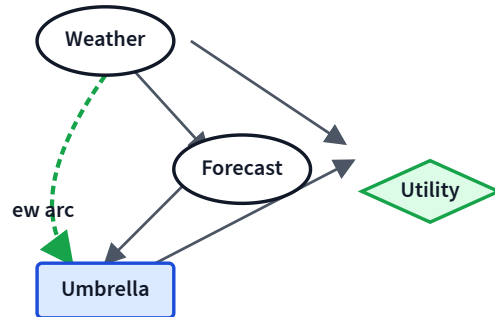


$$\text{Sum out } F: \sum_F f_6 = 49 + 14 + 14 = 77$$

Same answer as enumeration ($EU^* = 77$) — without enumerating all 8 policies.

The value of information

What if we could observe the weather *directly*, instead of only the noisy forecast?



New optimal policy (after re-running VEA):

- norain → leave (utility 100)
- rain → take (utility 70)

$$EU = 0.7 \cdot 100 + 0.3 \cdot 70 = 91.$$

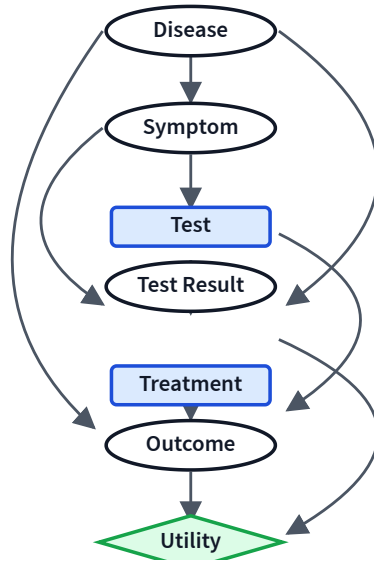
Value of information

$$VOI(W) = EU_{\text{with } W} -$$

$$EU_{\text{without}} = 91 - 77 = \mathbf{14}.$$

That's the maximum price we'd pay for a perfect forecast.

A medical diagnosis decision network



A doctor makes two decisions:

1. Which **test** (or none) to run.
2. Which **treatment** to give.

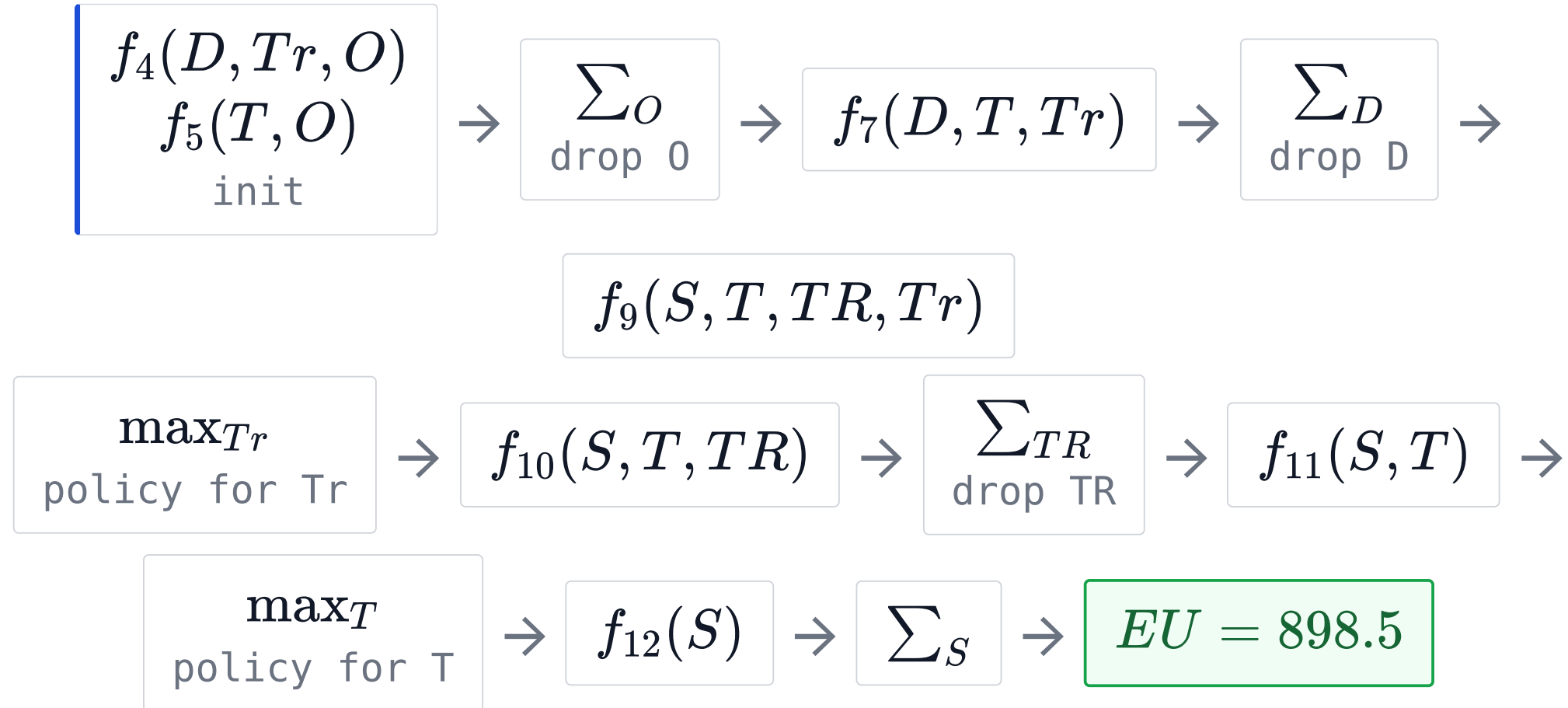
The information at decision time:

- Test \leftarrow symptom only.
- Treatment \leftarrow symptom, test, test result.

Tests cost utility but yield information for the next decision.

VEA on the medical DN: factor flow

Eliminate inner-most chance vars, then maximize the innermost decision, repeat:



Eliminate chance vars not yet needed; maximize the last decision (Treatment first, then Test).

Optimal medical policy

Test policy

S	T^*
1	0 (no test)
0	0 (no test)

Treatment policy

For every (S, T, TR) : $Tr^* = 1$. Always treat.

Expected utility = 898.5.

The cost of testing exceeds its information value in this network — skip the test and treat directly.

Learning goals (recap)

- ✓ **Model** a decision problem as a decision network.
- ✓ Find the optimal policy by **enumerating** expected utilities.
- ✓ Find the optimal policy by **variable elimination**.
- ✓ Quantify the **value of information**.
- ✓ Apply VEA to a multi-decision DN with an information ordering.

Next: sequential decisions

Today's decisions were *one-shot*. What if an agent acts **over time**, with each action changing the world?

Markov Decision Processes — rewards instead of one-shot utility, policies over states, value iteration.