Decision Theory

Descriptive Decision Theory

Descriptive Decision Theorie tries to simulate human behavior in finding the right or best decision for a given problem

Example:

- Company can chose one of two places for a new store
- Option 1: 125.000 EUR profit per year
- Option 2: 150.000 EUR profit per year

  Company should take Option 2, because it maximized the profit.

Often, there are multiple target values, which should be optimal

Example (additional Information):

- Option 1: 2.000.000 EUR sales per year
- Option 2: 1.800.000 EUR sales per year

  There is a conflict to handle

Decisions under Uncertainty

In real world not every thing is known, so there are uncertainties in the model

Example:

- There are plans for restructure the local traffic, which changes the predicted profit
- Option 1: 125.000 EUR profit per year
- Option 2: 80.000 EUR profit per year

  With modification Option 1 is the better one and without modification Option 2 is the better one

To model these variations in the environment we use so called Decision Tables

	z_1 (no modification)	z_2 (restructure)
a_1 (Option 1)	$125.000 = e_{11}$	$125.000 = e_{12}$
a_2 (Option 2)	$150.000 = e_{21}$	$80.000 = e_{22}$

Probability-based Decisions

In many cases probabilities could be assigned to each option

Objective Probabilities based on mathematic or statistic background

Subjective Probabilities based on intuition or estimations

Example:

- The management estimates the probability for the restructure to 30%

  The decision can be chosen by expectation value

	z_1 (no modification) $p_1 = 0.7$	z_2 (restructure) $p_2 = 0.3$	Expectation Value
a_1 (Option 1)	$125.000 = e_{11}$	$125.000 = e_{12}$	125.000
a_2 (Option 2)	$150.000 = e_{21}$	$80.000 = e_{22}$	129.000

Option 2 has the higher expectation value and should be used

Domination

An alternative a_1 dominates a_2 if the value of a_1 is always greater of (or equal to) the value of a_2

$\forall_j e_{1j} \geq e_{2j}$ ´

Example:

	z_1	z_2
a_1	$150.000 = e_{11}$	$90.000 = e_{12}$
a_2	$125.000 = e_{21}$	$80.000 = e_{22}$

Alternative a_2 could be dropped

Domination - Example 2

Some more alternatives:

	z_1	z_2	z_3	z_4	z_5	
a_1	0	20	10	60	25	dominated by a_3
a_2	-20	80	10	10	60	
a_3	20	60	20	60	50	
a_4	55	40	60	10	40	
a_5	50	10	30	5	20	dominated by a_4

- a_3 dominated a_1
- a_4 dominated a_5

  Alternatives a_1 and a_5 could be dropped

Probability Domination

	z_1 $p_1 = 0.3$	z_2 $p_2 = 0.2$	z_3 $p_1 = 0.4$	z_4 $p_2 = 0.1$
a_1	20	40	10	50
a_2	60	30	50	20

Probability Domination means that the cumulated probability for the payout for is always higher

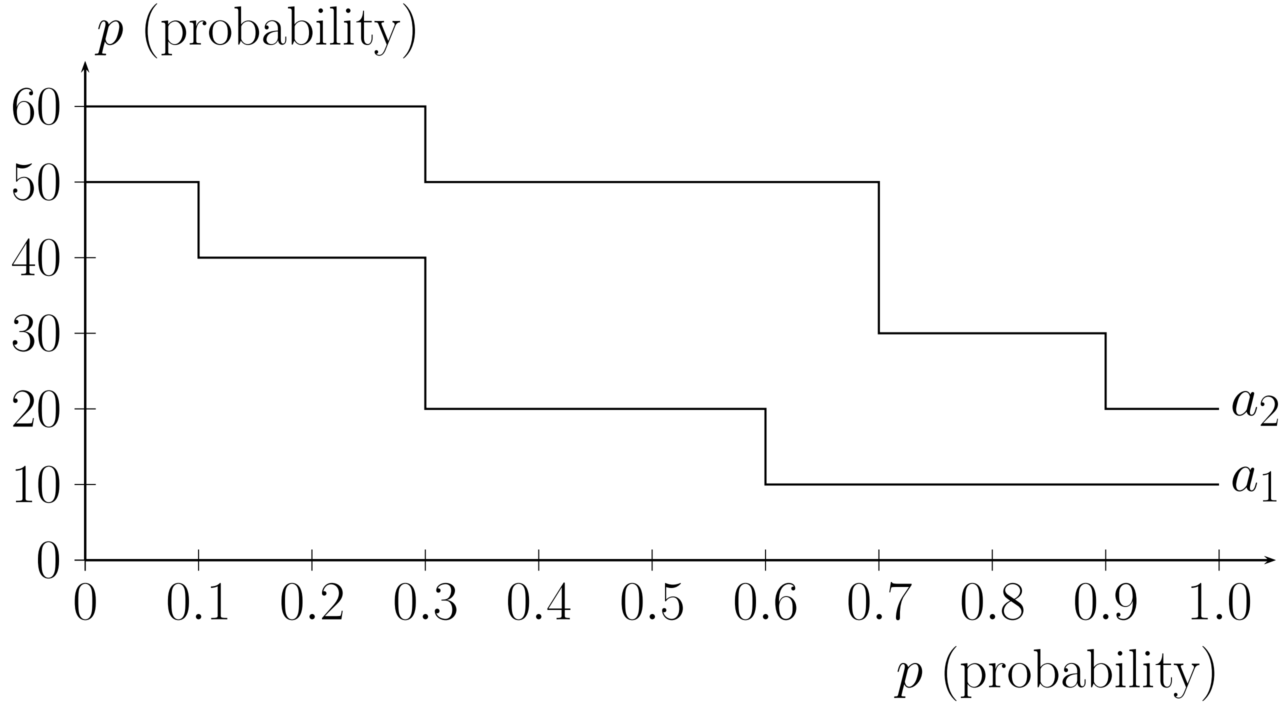
Algorithm:
- Order payout by value in a decreasing order
- Cumulate probabilities

Example:
- a_1 : 50(0.1) 40(0.2) 20(0.3) 10(0.4)
- a_2 : 60(0.3) 50(0.4) 30(0.2) 20(0.1)

Probability Domination

Example:

- a_1 : 50(0.1) 40(0.2) 20(0.3) 10(0.4)
- a_2 : 60(0.3) 50(0.4) 30(0.2) 20(0.1)



a_2 dominates a_1.

Multi Criteria Decisions

Optimization for multiple Targets

Complementary Targets

- Selling left foot shoes / Selling right foot shoes
- One could be avoided

Independent Targets

- Could be optimized separately

Competitive Targets

- Increase profit and sales
- Decrease environment pollution

Multi Criteria Decisions - Example

	Price	Sales e_1	Profit e_2	Environment Pollution e_3
a_1	15	800	7000	-4
a_2	20	600	7000	-2
a_3	25	400	6000	0
a_4	30	200	4000	0

Efficient Alternatives

- Only focus on alternatives which are not dominated by others
- Example: Drop a_4

Finding a decision

- If multiple alternatives are effective we need an algorithm to choose the preferred one
- Simplest algorithm: Chose one target (most important, alphabetical) and optimize for this value

Multi Criteria Decisions - Utility Function

Goal find a function $U(e_1, e_2, \ldots, e_n)$ as a combination of all targets, which could be optimized

Linear combination

- Simplest variant: Linear combination of all targets
- $U(e_1, e_2, \ldots, e_i) = \sum_{i=1}^{n} \omega_i \cdot e_i$

Example

- $\omega_1 = 10, \quad \omega_2 = 1, \quad \omega_3 = 500$

	Price	Sales e_1	Profit e_2	Environment Pollution e_3	$U(e_1, e_2, e_3)$
a_1	15	800	7000	-4	**13000**
a_2	20	600	7000	-2	12000
a_3	25	400	6000	0	10000

Decision under Uncertainty

	z_1	z_2	z_3	z_4
a_1	60	30	50	60
a_2	10	10	10	140
a_3	-30	100	120	130

Think about, how you would decide!

Decision Rules

- Maximin - Rule
- Maximax - Rule
- Hurwicz - Rule
- Savage-Niehans - Rule
- Laplace - Rule

Maximin - Rule

	z_1	z_2	z_3	z_4	Minimum
a_1	60	30	50	60	**30**
a_2	10	10	10	140	10
a_3	-30	100	120	130	-30

Chose the one with the highest minimum

Contra: To pessimistic, only focus on one column

Example

	z_1	z_2	z_3	z_4	Minimum
a_1	1,000,000	1,000,000	0.99	1,000,000	0.99
a_2	1	1	1	1	**1**

Maximax - Rule

	z_1	z_2	z_3	z_4	Maximum
a_1	60	30	50	60	60
a_2	10	10	10	140	**140**
a_3	-30	100	120	130	130

Chose the one with the highest maximum

Contra: To optimistic, only focus on one column

Example

	z_1	z_2	z_3	z_4	Maximum
a_1	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000
a_2	1,000,001	1	1	1	**1,000,001**

	z_1	z_2	z_3	z_4	Max	Min	$\Phi(a_i)$
a_1	60	30	50	60	60	30	$0.4 \cdot 60 + 0.6 \cdot 30 = 42$
a_2	10	10	10	140	140	10	$0.4 \cdot 140 + 0.6 \cdot 10 = \mathbf{62}$
a_3	-30	100	120	130	130	-30	$0.4 \cdot 130 + 0.6 \cdot (-30) = 34$

Combination of Maximin and Maximax - Rule
$\Phi(a) = \lambda \cdot \max(e_i) + (1 - \lambda) \cdot \min(e_i)$
λ represents readiness to assume risk
Contra: Only focus on two column
Example $(\min(a_1) < \min(a_2), \max(a_1) < \max(a_2) \Rightarrow$ chose $a_2)$

	z_1	z_2	z_3	z_4	Max	Min
a_1	1,000,000	1,000,000	1,000,000	0.99	1,000,000	0.99
a_2	1,000,001	1	1	1	**1,000,001**	**1**

Savage-Niehans - Rule

	z_1	z_2	z_3	z_4
a_1	**60**	30	50	60
a_2	10	10	10	**140**
a_3	-30	**100**	**120**	130

Rule of minimal regret

Algorithm:

- Find the maximal value for every column
- Subtract value from maximal value
- Use alternative with the lowest regret

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	60 - 60 = 0	70	70	80	**80**
a_2	60 - 10 = 50	90	110	0	110
a_3	60 - (-30) = 90	0	0	10	90

Savage-Niehans - Rule II

	z_1	z_2	z_3	z_4
a_1	1,000	1,000,000	1,000,000	1,000,000
a_2	1,001	0	0	0

Another example

we chose a_1

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	1	0	0	0	**1**
a_2	0	1,000,000	1,000,000	1,000,000	1,000,000

Savage-Niehans - Rule III

	z_1	z_2	z_3	z_4
a_1	1,000	1,000,000	1,000,000	1,000,000
a_2	1,001	0	0	0
a_3	2,000,000	-1,000,000	-1,000,000	-1,000,000

Same example, but we add alternative a_3

Now we chose a_2

Regret Table:

	z_1	z_2	z_3	z_4	Max
a_1	1,999,000	0	0	0	1,999,000
a_2	1,998,999	1,000,000	1,000,000	1,000,000	**1,998,999**
a_2	0	2,000,000	2,000,000	2,000,000	2,000,000

What this means in real life:

- Student think about swimming a_1 and running a_2
- The fun factor is depending on the weather $z_1 \ldots z_4$
- Student decides to go swimming
- He talk to a friend and presents his plans for the evening
- The friend mentioned to go for a BBQ a_3
- With the option for BBQ the student decides to go running

Laplace - Rule

	z_1	z_2	z_3	z_4	Mean
a_1	60	30	50	60	50
a_2	10	10	10	140	42.5
a_3	-30	100	120	130	**80**

Chose the one with the highest mean value

Contra:

- Not every condition has the same probability
- Duplication of one condition could change the result

Most people would also chose a_3 in this example

Rule - Axioms

The following axioms should be fulfilled by the rules

Addition to a column
The decision should not be changed, if a fixed value is added to a column

Additional rows
The preference relation between two alternatives should not be changed, if a new row is added

Domination
If a_1 dominates a_2, a_2 could not be optimal

Join of equal columns
The preference relation between to alternatives should not change, if two columns with the same outcomes are joined to a common column

Decision Rules Conclusion

Rule	Example Result	Addition to a row	Additional Rows	Domination	Join of equal Rows
Maximin	a_1		$\surd$		$\surd$
Maximax	a_2		$\surd$		$\surd$
Hurwicz	a_2		$\surd$		$\surd$
Savage-Niehans	a_1	$\surd$		$\surd$	$\surd$
Laplace	a_3	$\surd$	$\surd$	$\surd$	

No Rule fulfills all axioms $\Rightarrow$ no perfect rule

Common usage: Remove duplicate Columns and use Laplace

Better: Define subjective probabilities and use them

Decision Graphs / Influence Diagrams

Preference Orderings

A *preference ordering* $\succeq$ is a ranking of all possible states of affairs (worlds) S

- these could be outcomes of actions, truth assignments, states in a search problem, etc.
- $s \succeq t$: means that state s is *at least as good as* t
- $s \succ t$: means that state s *is strictly preferred to* t

We insist that $\succeq$ is

- reflexive: i.e., s $\succeq$ s for all states s
- transitive: i.e., if s $\succeq$ t and t $\succeq$ w, then s $\succeq$ w
- connected: for all states s,t, either s $\succeq$ t or t $\succeq$ s

Why Impose These Conditions?

Structure of preference ordering imposes certain “rationality requirements” (it is a weak ordering)

E.g., why transitivity?

- Suppose you (strictly) prefer coffee to tea, tea to OJ, OJ to coffee
- If you prefer X to Y, you will trade me Y plus $1 for X
- I can construct a “money pump” and extract arbitrary amounts of money from you

Utilities

Rather than just ranking outcomes, we must quantify our degree of preference

- e.g., how much more important is *chc* than ∼*mess*

A *utility function* $U : S \to \mathbb{R}$ associates a realvalued *utility* with each outcome.

- $U(s)$ measures your *degree* of preference for s

Note: U induces a preference ordering $\succeq_U$ over S defined as: s $\succeq_U$ t iff $U(s) \geq U(t)$

- obviously $\succeq_U$ will be reflexive, transitive, connected

Expected Utility

Under conditions of uncertainty, each decision d induces a distribution Pr_d over possible outcomes

- $Pr_d(s)$ is probability of outcome s under decision d

The *expected utility* of decision d is defined

The *principle of maximum expected utility (MEU)* states that the optimal decision under conditions of uncertainty is that with the greatest expected utility.

$$EU(d) = \sum_{s \in S} Pr_d(s) U(s)$$

Decision Problems: Uncertainty

A *decision problem under uncertainty* is:

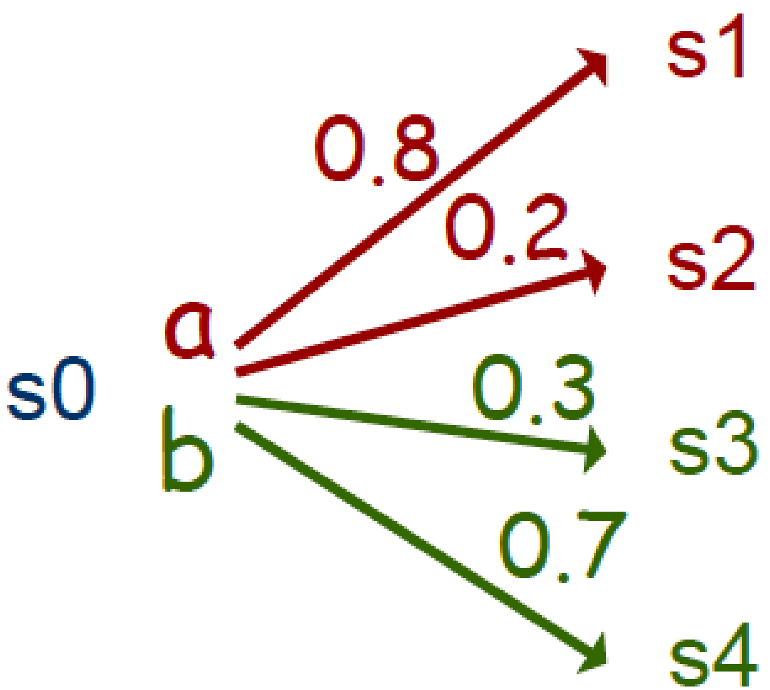
- a set of *decisions* D
- a set of *outcomes* or states S
- an *outcome function* $Pr : D \to \Delta(S)$
  - $\Delta(S)$ is the set of distributions over S (e.g., Prd)
- a *utility function* U over S

A solution to a decision problem under uncertainty is any $d^* \in D$ such that $EU(d^*) \succeq EU(d)$ for all $d \in D$

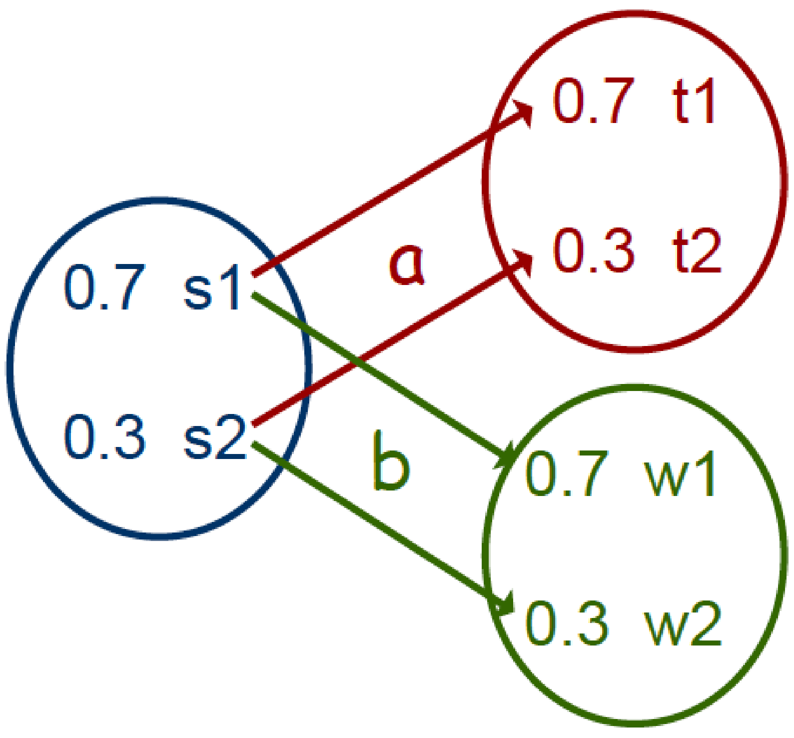
Again, for single-shot problems, this is trivial

Note that this viewpoint accounts for both:

- uncertainty in action outcomes
- uncertainty in state of knowledge
- any combination of the two



Stochastic actions



Uncertain knowledge

Expected Utility: Notes

Why MEU? Where do utilities come from?

- underlying foundations of utility theory tightly couple utility with action/choice
- a utility function can be determined by asking someone about their preferences for actions in specific scenarios (or “lotteries” over outcomes)

Utility functions needn’t be unique

- if I multiply U by a positive constant, all decisions have same relative utility
- if I add a constant to U, same thing
- *U is unique up to positive affine transformation*

So What are the Complications?

Outcome space is large

- like all of our problems, states spaces can be huge
- don't want to spell out distributions like Pr_d explicitly
- Solution: Bayes nets (or related: *influence diagrams*)

Decision space is large

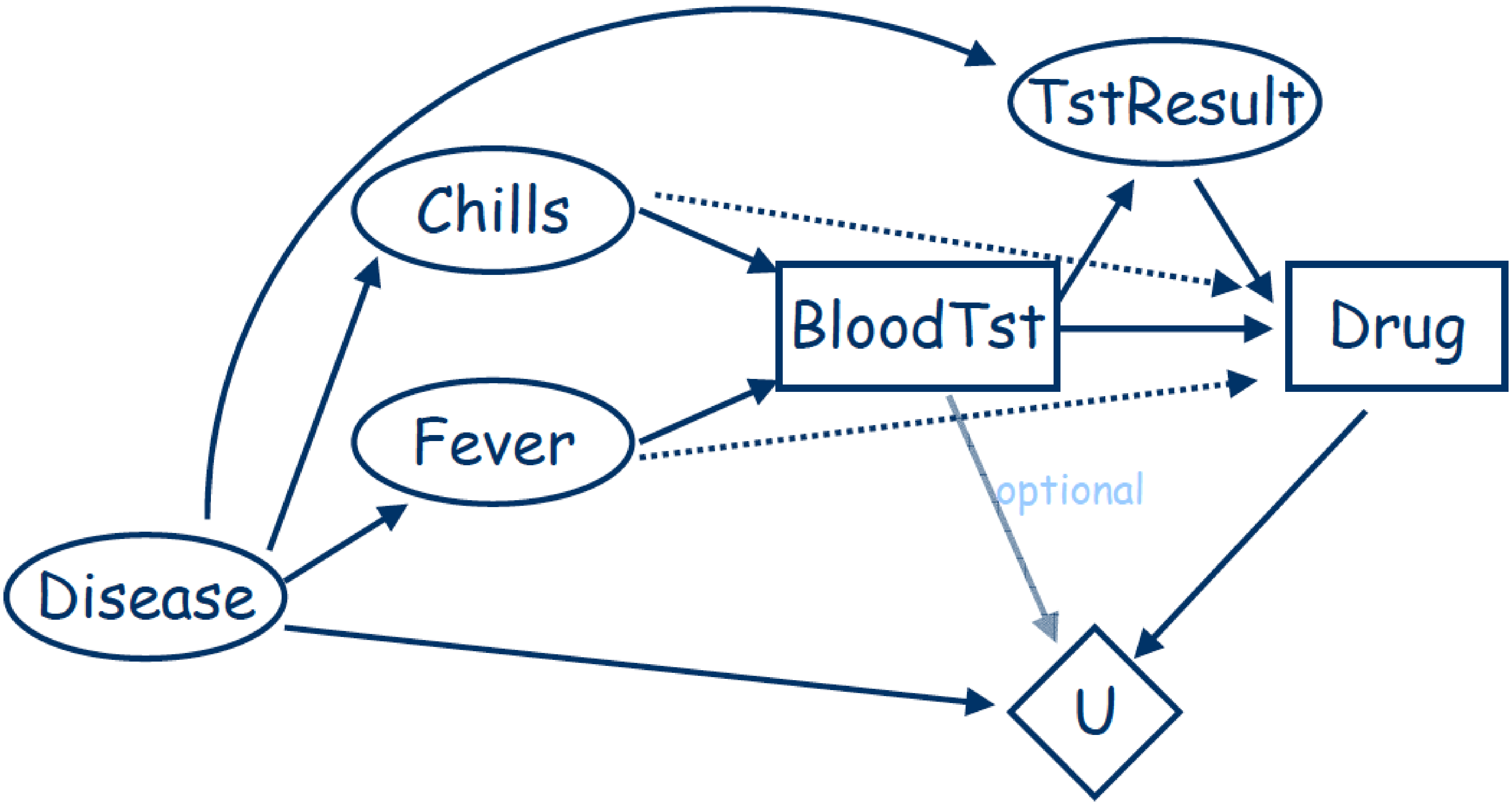
- usually our decisions are not one-shot actions
- rather they involve sequential choices (like plans)
- if we treat each plan as a distinct decision, decision space is too large to handle directly
- Soln: use dynamic programming methods to *construct* optimal plans (actually generalizations of plans, called policies. . . like in game trees)

So What are the Complications?

Decision networks (more commonly known as *influence diagrams*) provide a way of representing sequential decision problems

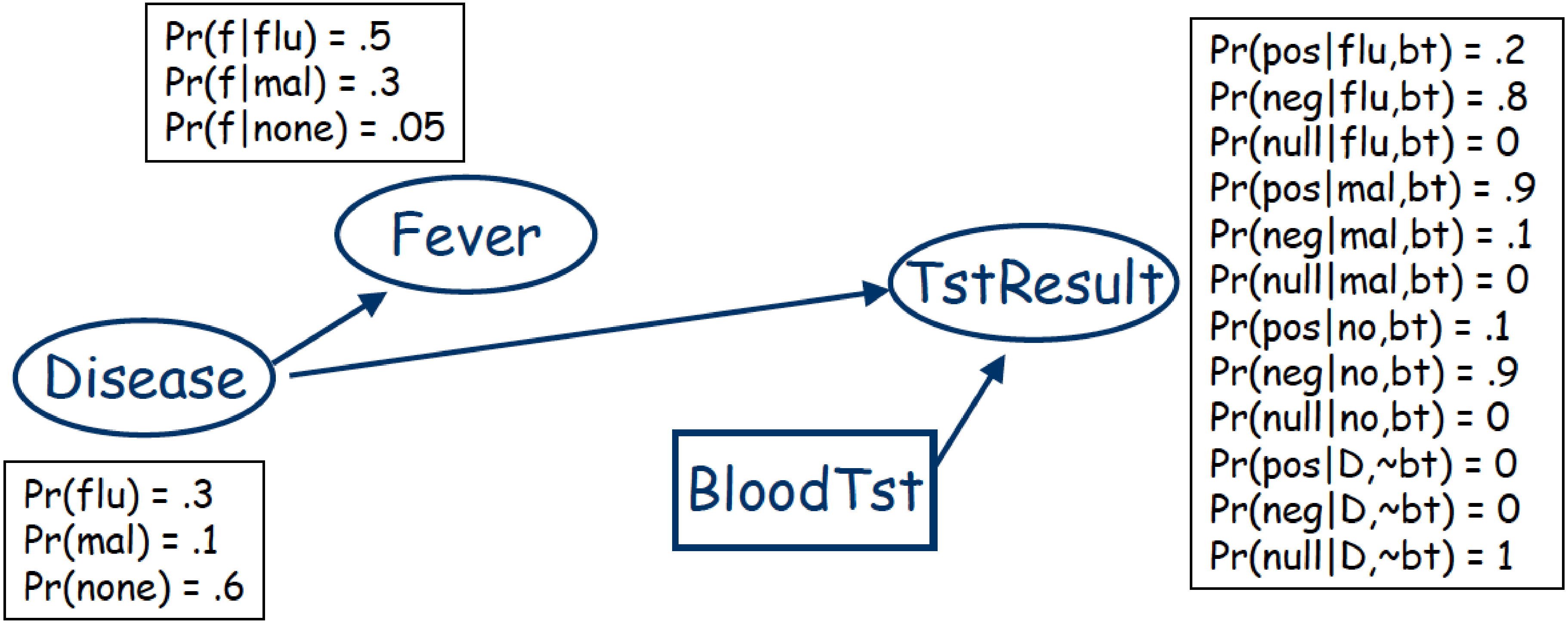
- basic idea: represent the variables in the problem as you would in a BN
- add decision variables – variables that you “control”
- add utility variables – how good different states are

Sample Decision Network



Chance nodes

- random variables, denoted by circles
- as in a BN, probabilistic dependence on parents



Decision nodes

- variables decision maker sets, denoted by squares
- parents reflect *information available* at time decision is to be made

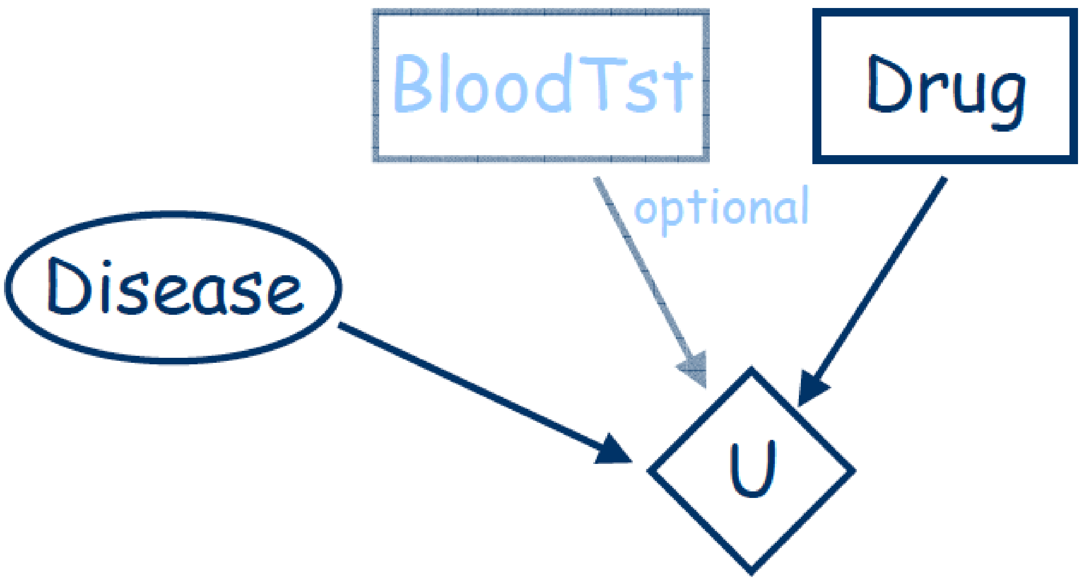
In example decision node: the actual values of Ch and Fev will be observed before the decision to take test must be made

- agent can make different decisions for each instantiation of parents (i.e., policies)

Value node

- specifies utility of a state, denoted by a diamond
- utility depends *only on state of parents* of value node
- generally: only one value node in a decision network

Utility depends only on disease and drug



U(fludrug, flu) = 20
U(fludrug, mal) = -300
U(fludrug, none) = -5
U(maldrug, flu) = -30
U(maldrug, mal) = 10
U(maldrug, none) = -20
U(no drug, flu) = -10
U(no drug, mal) = -285
U(no drug, none) = 30

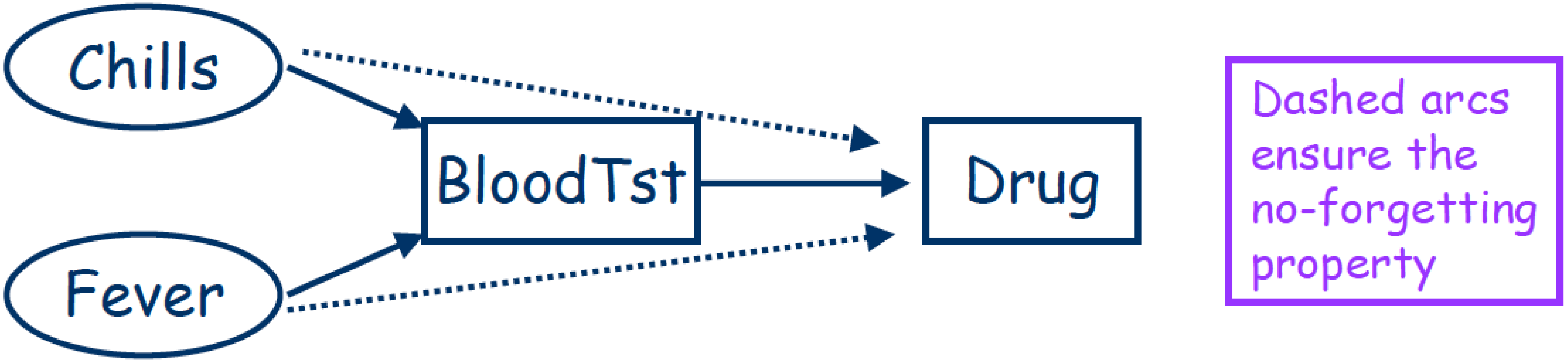
Decision Networks: Assumptions

Decision nodes are totally ordered

- decision variables $D_1, D_2, \ldots, D_n$
- decisions are made in sequence
- e.g., BloodTst (yes,no) decided before Drug (fd,md,no)

No-forgetting property

- any information available when decision D_i is made is available when decision D_j is made (for $i < j$)
- thus all parents of D_i are parents of D_j



Policies

Let $Par(D_i)$ be the parents of decision node D_i

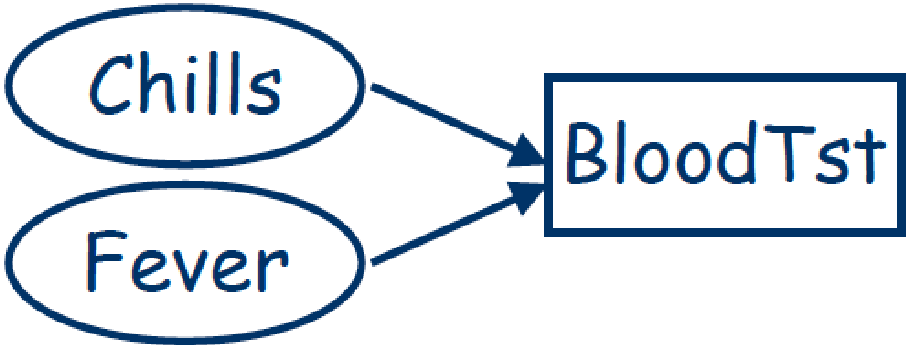
- $Dom(Par(Di))$ is the set of assignments to parents

A policy δ is a set of mappings δ_i, one for each decision node D_i

- $\delta_i : Dom(Par(D_i)) \rightarrow (D_i)$
- δ_i associates a decision with each parent assignment for D_i

For example, a policy for BT might be:

$$\begin{aligned}\delta_{BT}(c, f) &= bt\\ \delta_{BT}(c, \sim f) &= \sim bt\\ \delta_{BT}(\sim c, f) &= bt\\ \delta_{BT}(\sim c, \sim f) &= \sim bt\end{aligned}$$



Value of a Policy

Value of a policy δ is the expected utility given that decision nodes are executed according to δ

Given associates $\boldsymbol{x}$ to the set $\boldsymbol{X}$ of all chance variables, let $\delta(\boldsymbol{x})$ denote the assignment to decision variables dictated by δ

- e.g., assigned to D_1 determined by it's parents' assignment in $\boldsymbol{x}$
- e.g., assigned to D_2 determined by it's parents' assignment in $\boldsymbol{x}$ along with whatever was assigned to D1
- etc.

Value of δ:

$$EU(\delta) = \sum_{\boldsymbol{X}} P(\boldsymbol{X}, \delta(\boldsymbol{X})U(\boldsymbol{X}, \delta(\boldsymbol{X}))$$

Optimal Policies

An *optimal policy* is a policy δ^* such that $EU(\delta^*) \geq EU(\delta)$ for all policies δ

We can use the dynamic programming principle yet again to avoid enumerating all policies

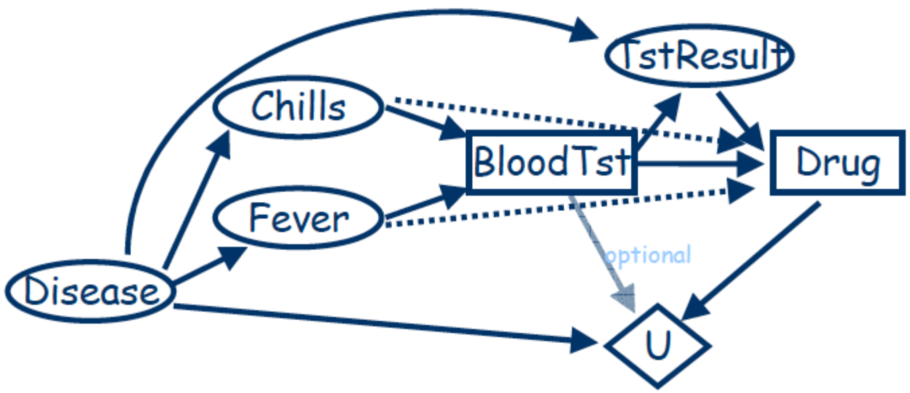
We can also use the structure of the decision network to use variable elimination to aid in the computation

Computing the Best Policy

We can work backwards as follows

First compute optimal policy for Drug (last dec'n)

- for each assignment to parents (C,F,BT,TR) and for each decision value (D = md,fd,none), *compute the expected value* of choosing that value of D
- set policy choice for each value of parents to be the value of D that has max value
- eg: $\delta_D(c, f, bt, pos) = md$



Computing the Best Policy

Next compute policy for BT given policy $\delta_D(C, F, BT, TR)$ just determined for Drug

- since $\delta_D(C, F, BT, TR)$ is fixed, we can treat Drug as a normal random variable with deterministic probabilities
- i.e., for any instantiation of parents, value of Drug is fixed by policy δ_D
- this means we can solve for optimal policy for BT just as before
- only uninstantiated vars are random vars (once we fix *its* parents)

Computing the Best Policy

How do we compute these expected values?

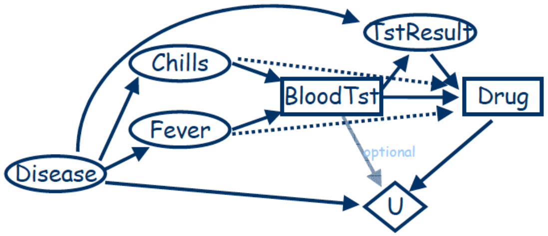
- suppose we have assigned $< c, f, bt, pos >$ to parents of $Drug$
- we want to compute EU of deciding to set $Drug = md$
- we can run variable elimination!

Treat C, F, BT, TR, Dr as evidence

- this reduces factors (e.g., U restricted to bt, md: depends on Dis)
- eliminate remaining variables (e.g., only Disease left)
- left with factor: $U() = \sum_{Dis} P(Dis|c, f, bt, pos, md)U(Dis)$

We now know EU of doing $Dr = md$ when c, f, bt, pos true

Can do same for fd, no to decide which is best



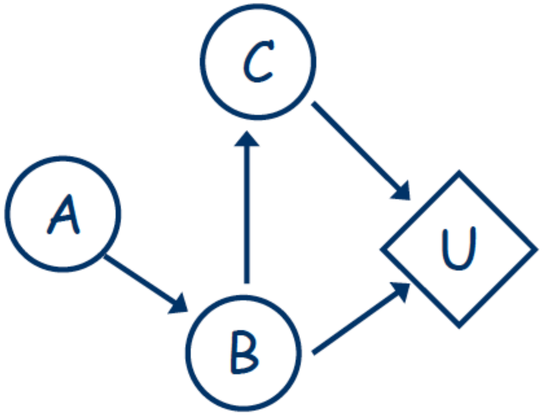
Computing Expected Utilities

The preceding illustrates a general phenomenon

- computing expected utilities with BNs is quite easy
- utility nodes are just factors that can be dealt with using variable elimination

$$
\begin{aligned}
EU &= \sum_{A,B,C} P(A,B,C)U(B,C) \\
&= \sum_{A,B,C} P(C|B)P(B|A)P(A)U(B,C)
\end{aligned}
$$

Just eliminate variables in the usual way



Optimizing Policies: Key Points

If a decision node D has no decisions that follow it, we can find its policy by instantiating each of its parents and computing the expected utility of each decision for each parent instantiation

- no-forgetting means that all other decisions are instantiated (they must be parents)
- its easy to compute the expected utility using VE
- the number of computations is quite large: we run expected utility calculations (VE) for each parent instantiation together with each possible decision D might allow
- policy: choose max decision for each parent instant'n

Optimizing Policies: Key Points

When a decision D node is optimized, it can be treated as a random variable

- for each instantiation of its parents we now know what value the decision should take

- just treat policy as a new CPT: for a given parent instantiation $\boldsymbol{x}$, D gets $\delta(\boldsymbol{x})$ with probability 1 (all other decisions get probability zero)

If we optimize from last decision to first, at each point we can optimize a specific decision by (a bunch of) simple VE calculations

- it's successor decisions (optimized) are just normal nodes in the BNs (with CPTs)

Decision Network Notes

Decision networks commonly used by decision analysts to help structure decision problems

Much work put into computationally effective techniques to solve these

- common trick: replace the decision nodes with random variables at outset and solve a plain Bayes net (a subtle but useful transformation)

Complexity much greater than BN inference

- we need to solve a number of BN inference problems
- one BN problem for each setting of decision node parents and decision node value

Decision Network Notes

In example on previous slide:

- we assume the state (of the variables at any stage) is fully observable
  - hence all time t vars point to time t decision
- this means the state at time t d-separates the decision at time t-1 from the decision at time t-2
- so we ignore "no-forgetting" arcs between decisions
  - once you *know* the state at time t, what you *did* at time t-1 to get there is irrelevant to the decision at time t-1

If the state were not fully observable, we could not ignore the "no-forgetting" arcs

A Detailed Decision Net Example

Setting: you want to buy a used car, but there's a good chance it is a "lemon" (i.e., prone to breakdown). Before deciding to buy it, you can take it to a mechanic for inspection. S/he will give you a report on the car, labelling it either "good" or "bad". A good report is positively correlated with the car being sound, while a bad report is positively correlated with the car being a lemon.

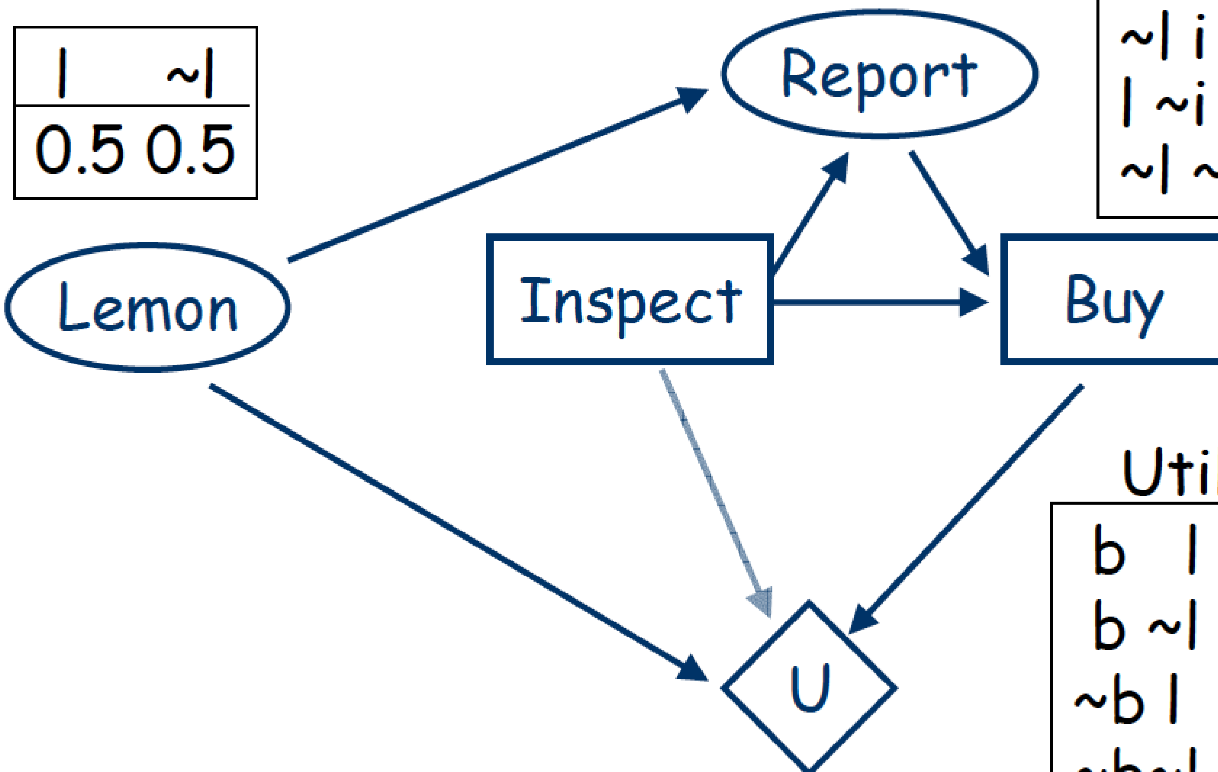
The report costs $50 however. So you could risk it, and buy the car without the report.

Owning a sound car is better than having no car, which is better than owning a lemon.

Rep: good,bad,none

	g	b	n
l i	0.2	0.8	0
~l i	0.9	0.1	0
l ~i	0	0	1
~l ~i	0	0	1

l	~l
0.5	0.5



Utility

b l	-600
b ~l	1000
~b l	-300
~b~l	-300

-50 if inspect

Evaluate Last Decision: Buy (1)

$EU(B|I,R) = \sum_L P(L|I,R,B)U(L,B)$

$I = i, R = g$:

$$\begin{aligned} EU(buy) &= P(l|i,g)U(l,buy) + P(\sim l|i,g)U(\sim l,buy) - 50 \\ &= .18 \cdot -600 + .82 \cdot 1000 - 50 = 662 \\ EU(\sim buy) &= P(l|i,g)U(l,\sim buy) + P(\sim l|i,g)U(\sim l,\sim buy) - 50 \\ &= -300 - 50 = -350 (-300 \text{ indep. of lemon}) \end{aligned}$$

So optimal $\delta_{Buy}(i,g) = buy$

Evaluate Last Decision: Buy (2)

$I = i, R = b$:

$$
\begin{aligned}
EU(buy) &= P(l|i,b)U(l,buy) + P(\sim l|i,b)U(\sim l,buy) - 50 \\
&= .89 \cdot -600 + .11 \cdot 1000 - 50 = -474 \\
EU(\sim buy) &= P(l|i,b)U(l,\sim buy) + P(\sim l|i,b)U(\sim l,\sim buy) - 50 \\
&= -300 - 50 = -350 (-300 \text{ indep. of lemon})
\end{aligned}
$$

So optimal $\delta_{Buy}(i,b) = \sim buy$

Evaluate Last Decision: Buy (3)

$I = \sim i, R = g$ (note: no inspection cost subtracted):

$$\begin{aligned} EU(buy) &= P(l|\sim i, g)U(l, buy) + P(\sim l|\sim i, g)U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \\ EU(\sim buy) &= P(l|\sim i, g)U(l, \sim buy) + P(\sim l|\sim i, g)U(\sim l, \sim buy) - 50 \\ &= -300 - 50 = -350 (-300 \text{ indep. of lemon}) \end{aligned}$$

So optimal $\delta_{Buy}(\sim i, g) = \sim buy$

So optimal policy for Buy is:

- $\delta_{Buy}(i, g) = buy; \delta_{Buy}(i, b) = \sim buy; \delta_{Buy}(\sim i, n) = buy$

Note: we don't bother computing policy for $(i, \sim n)$, $(\sim i, g)$, or $(\sim i, b)$, since these occur with probability 0

Evaluate First Decision: Inspect

$EU(I) = \sum_{L,R} P(L,R|I) U(L, \boldsymbol{\delta_{Buy}(I,R)})$,
where $P(R,L|I) = P(R|L,I)P(L|I)$

$$\begin{aligned} EU(i) &= .1 \cdot -600 + .4 \cdot -300 + .45 \cdot 1000 + .05 \cdot -300 - 50 \\ &= 237.5 - 50 = 187.5 \\ EU(\sim i) &= P(l|\sim i, n)U(l, buy) + P(\sim l|\sim i, n)U(\sim l, buy) \\ &= .5 \cdot -600 + .5 \cdot 1000 = 200 \end{aligned}$$

So optimal $\delta_{Inspect}(\sim i) = buy$

	$P(R,L\|I)$	$\boldsymbol{\delta_{Buy}}$	$U(L, \boldsymbol{\delta_{Buy}})$
g, l	0.1	buy	$-600 - 50 = -650$
$g, \sim l$	0.45	buy	$1000 - 50 = 950$
b, l	0.4	$\sim buy$	$-300 - 50 = -350$
$b, \sim l$	0.05	$\sim buy$	$-300 - 50 = -350$

Value of Information

So optimal policy is: don't inspect, buy the car

- EU = 200

- Notice that the EU of inspecting the car, then buying it iff you get a good report, is 237.5 less the cost of the inspection (50). So inspection not worth the improvement in EU.

- But suppose inspection cost \$25: then it would be worth it ($EU = 237.5 - 25 = 212.5 > EU(\sim i)$)

- The *expected value of information* associated with inspection is 37.5 (it improves expected utility by this amount ignoring cost of inspection). How? Gives opportunity to change decision ($\sim$ *buy* if bad).

- You should be willing to pay up to \$37.5 for the report