Goals and Preferences

Alice . . . went on "Would you please tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where —" said Alice.
"Then it doesn't matter which way you go," said the Cat.

Lewis Carroll, 1832–1898
Alice's Adventures in Wonderland, 1865
Chapter 6

Review

Decision network:

- Directed acyclic graph (DAG) with three sorts of nodes: decision (rectangle), random (ellipse), utility (diamond)
- Domain for the decision and random variables.
- Unique utility node
- Arcs into a decision node represent the information that will be available when the decision is made
- For each random variable, there is factor representing the conditional probability for the random variable given its parents
- There a factor on the parents of the utility node
- No factors are (initially) associated with the decision nodes

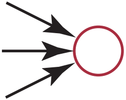
Overview

At the end of the class you should be able to:

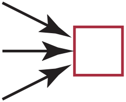
- model a user's preferences and utility when there is uncertainty
- build a simple model that includes actions, uncertainty and utilities.
- Find an optimal policy in a decision network.
- Determine the value of information and control

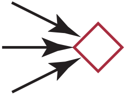
Decisions Networks

A decision network is a graphical representation of a finite sequential decision problem, with 3 types of nodes:



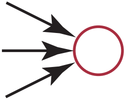
- A random variable is drawn as an ellipse. Arcs into the node represent probabilistic dependence. Each random variable has a domain and an associated factor.



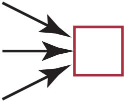


Decisions Networks

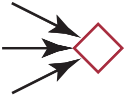
A decision network is a graphical representation of a finite sequential decision problem, with 3 types of nodes:



- A random variable is drawn as an ellipse. Arcs into the node represent probabilistic dependence. Each random variable has a domain and an associated factor.

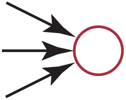


- A decision variable is drawn as an rectangle. Arcs into the node represent information available when the decision is make. Each decision variable has a domain, but no associated factor.

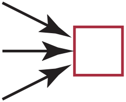


Decisions Networks

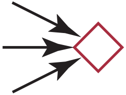
A decision network is a graphical representation of a finite sequential decision problem, with 3 types of nodes:



- A random variable is drawn as an ellipse. Arcs into the node represent probabilistic dependence. Each random variable has a domain and an associated factor.

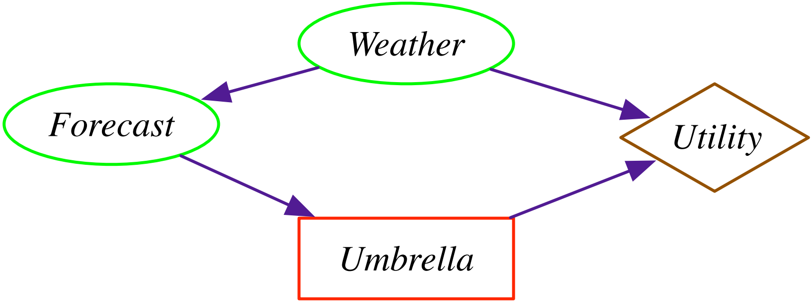


- A decision variable is drawn as an rectangle. Arcs into the node represent information available when the decision is make. Each decision variable has a domain, but no associated factor.



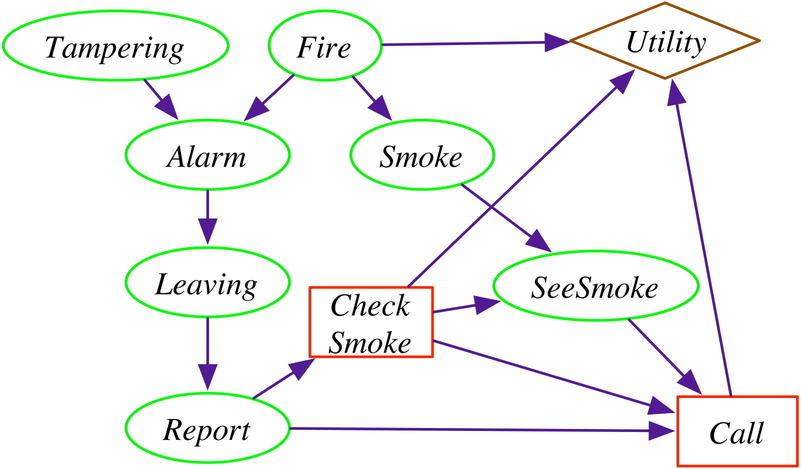
- A utility node is drawn as a diamond. Arcs into the node represent variables that the utility depends on. The utility node has no domain, and a factor on the parents of the node.

Umbrella Decision Network



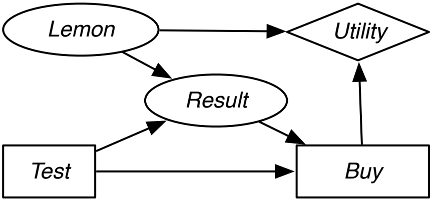
You don't get to observe the weather when you have to decide whether to take your umbrella. You do get to observe the forecast.

Decision Network for the Alarm Problem



Clicker Question

The decision network:



requires which probabilities to be specified:

A $P(Utility \mid Buy, Lemon)$, $P(Lemon)$, $P(Result \mid Lemon, Test)$, $P(Test)$, $P(Buy \mid Test, Result)$

B $P(Lemon)$, $P(Result \mid Lemon, Test)$, $P(Test)$, $P(Buy \mid Test, Result)$

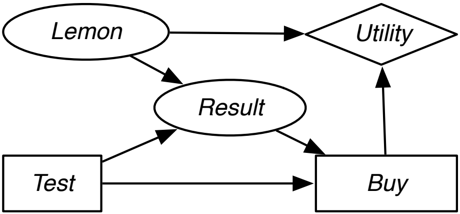
C $P(Utility \mid Buy, Lemon)$, $P(Lemon)$, $P(Result \mid Lemon, Test)$

D $P(Lemon)$, $P(Result \mid Lemon, Test)$

E $P(Utility \mid Lemon)$, $P(Lemon)$, $P(Result \mid Lemon)$, $P(Buy \mid Result)$

Clicker Question

The decision network:



requires how many factors be specified initially:

A 2

B 3

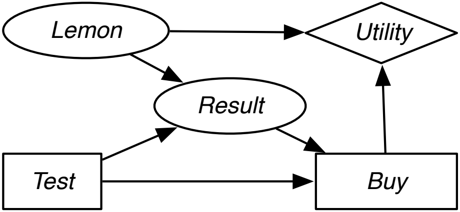
C 4

D 5

E 6

Clicker Question

In the decision network:



the initial factor that isn't a (conditional) probability is a factor on which variables?

A *Lemon*, *Result*, *Test*, *Buy*, *Utility*

B *Lemon*, *Result*, *Test*, *Buy*

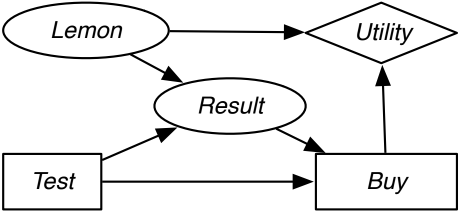
C *Result*, *Test*, *Buy*

D *Test*, *Buy*

E *Lemon*, *Buy*

Clicker Question

According to the network:



when does the agent know the value *Result*?

A Never

B Initially

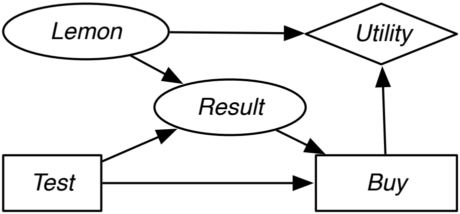
C After *Test* and before *Buy*

D After *Buy* and before *Test*

E After both *Test* and *Buy*

Clicker Question

According to the network



when does the agent know the value *Lemon*?

A Never

B Initially

C After *Test* and before *Buy*

D After *Buy* and before *Test*

E After both *Test* and *Buy*

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 decision function for decision node D_i is a function π_i that specifies what the agent does for each assignment of values to the parents of D_i.
  When it observes O, it does $\pi_i(O)$.

Policies

- A decision function for decision node D_i is a function π_i that specifies what the agent does for each assignment of values to the parents of D_i.
  When it observes O, it does $\pi_i(O)$.
- A policy is a sequence of decision functions; one for each decision node.

Expected Utility of a Policy

- Possible world ω satisfies policy π if ω assigns the value to each decision node that the policy specifies.
- The expected utility of policy π is

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

Expected Utility of a Policy

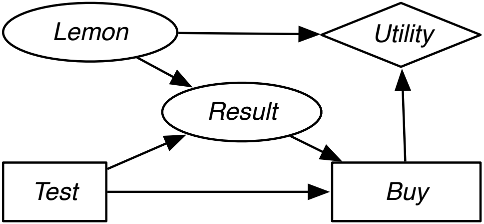
- Possible world ω satisfies policy π if ω assigns the value to each decision node that the policy specifies.
- The expected utility of policy π is

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

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

Clicker Question

Consider the decision network



where all variables are Boolean.

How many decision functions are there for *Test*?

A 2^2

B 2^4

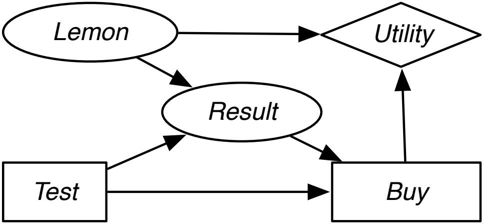
C 2^5

D 2

E There is not enough information to tell.

Clicker Question

Consider the decision network



where all variables are Boolean.
How many decision functions are there for *Buy*?

A 2^2

B 2^4

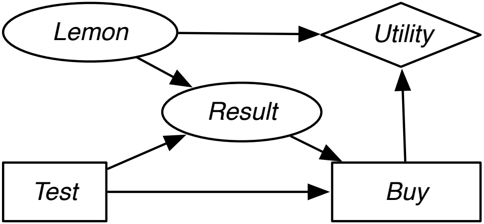
C 2^5

D 5

E There is not enough information to tell.

Clicker Question

Consider the decision network



where all variables are Boolean.
How many policies are there?

A 2^2

B 2^4

C 2^5

D 5

E There is not enough information to tell.

Finding an optimal policy

- Suppose the random variables are $X_1, \ldots, X_n$, and utility depends on $V_{i_1}, \ldots, V_{i_k}$ (random and/or decision variables)

$$\mathcal{E}(u \mid \pi) \quad =$$

Finding an optimal policy

- Suppose the random variables are $X_1, \ldots, X_n$, and utility depends on $V_{i_1}, \ldots, V_{i_k}$ (random and/or decision variables)

$$\mathcal{E}(u \mid \pi) = \sum_{X_1,\ldots,X_n} P(X_1, \ldots, X_n \mid \pi) \times u(V_{i_1}, \ldots, V_{i_k})$$

$$= \sum_{X_1,\ldots,X_n}$$

Finding an optimal policy

- Suppose the random variables are $X_1, \ldots, X_n$, and utility depends on $V_{i_1}, \ldots, V_{i_k}$ (random and/or decision variables)

$$\begin{aligned}
\mathcal{E}(u \mid \pi) &= \sum_{X_1,\ldots,X_n} P(X_1, \ldots, X_n \mid \pi) \times u(V_{i_1}, \ldots, V_{i_k}) \\
&= \sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} P(X_i \mid parents(X_i)) \times u(V_{i_1}, \ldots, V_{i_k})
\end{aligned}$$

Finding an optimal policy

- Suppose the random variables are $X_1, \ldots, X_n$, and utility depends on $V_{i_1}, \ldots, V_{i_k}$ (random and/or decision variables)

$$\begin{aligned}\mathcal{E}(u \mid \pi) &= \sum_{X_1,\ldots,X_n} P(X_1, \ldots, X_n \mid \pi) \times u(V_{i_1}, \ldots, V_{i_k}) \\ &= \sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} P(X_i \mid parents(X_i)) \times u(V_{i_1}, \ldots, V_{i_k})\end{aligned}$$

  Idea:
  - Sum out all of the random variables to compute expected utility.

Finding an optimal policy

- Suppose the random variables are $X_1, \ldots, X_n$, and utility depends on $V_{i_1}, \ldots, V_{i_k}$ (random and/or decision variables)

$$\begin{aligned}
\mathcal{E}(u \mid \pi) &= \sum_{X_1,\ldots,X_n} P(X_1, \ldots, X_n \mid \pi) \times u(V_{i_1}, \ldots, V_{i_k}) \\
&= \sum_{X_1,\ldots,X_n} \prod_{i=1}^{n} P(X_i \mid parents(X_i)) \times u(V_{i_1}, \ldots, V_{i_k})
\end{aligned}$$

Idea:

  - Sum out all of the random variables to compute expected utility.
  - Choose the policy to maximize the sum: when a decision variable is in a factor with only its parents, select maximum value.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable
    — D is only in a factor f with (some of) its parents.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable
    — D is only in a factor f with (some of) its parents.
  - Eliminate D by maximizing.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable
    — D is only in a factor f with (some of) its parents.
  - Eliminate D by maximizing. This returns:
    - an optimal decision function for D: $\arg\max_D f$
    - a new factor: $\max_D f$

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable
    — D is only in a factor f with (some of) its parents.
  - Eliminate D by maximizing. This returns:
    - an optimal decision function for D: $\arg\max_D f$
    - a new factor: $\max_D f$
- until there are no more decision nodes.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable
    — D is only in a factor f with (some of) its parents.
  - Eliminate D by maximizing. This returns:
    - an optimal decision function for D: $\arg\max_D f$
    - a new factor: $\max_D f$
- until there are no more decision nodes.
- Sum out the remaining random variables.

Finding an optimal policy

- Create a factor for each conditional probability table and a factor for the utility.
- Repeat:
  - Sum out random variables that are not parents of a decision node.
  - Let D be last decision variable — D is only in a factor f with (some of) its parents.
  - Eliminate D by maximizing. This returns:
    - an optimal decision function for D: $\arg\max_D f$
    - a new factor: $\max_D f$
- until there are no more decision nodes.
- Sum out the remaining random variables.
- Multiply the factors: this is the expected utility of an optimal policy.

Initial factors for the Umbrella Decision

Weather	Value
norain	0.7
rain	0.3

Weather	Fcast	Value
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	Umb	Value
norain	take	20
norain	leave	100
rain	take	70
rain	leave	0

Eliminating By Maximizing

f:

Fcast	Umb	Val
sunny	take	12.95
sunny	leave	49.0
cloudy	take	8.05
cloudy	leave	14.0
rainy	take	14.0
rainy	leave	7.0

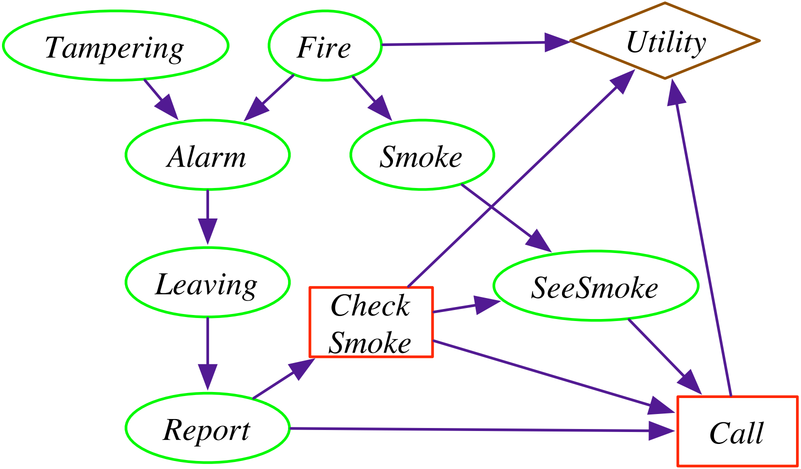
$\max_{Umb} f$:

Fcast	Val
sunny	49.0
cloudy	14.0
rainy	14.0

$\arg\max_{Umb} f$:

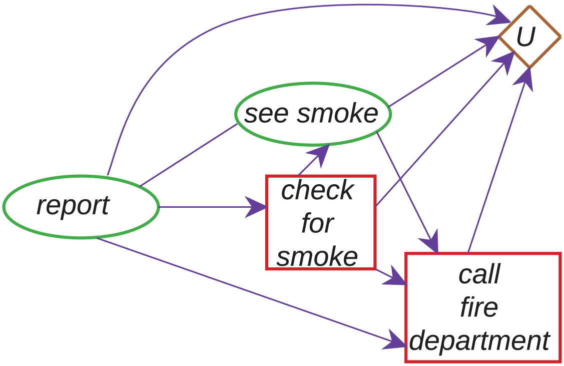
Fcast	Umb
sunny	leave
cloudy	leave
rainy	take

Decision Network for the Alarm Problem

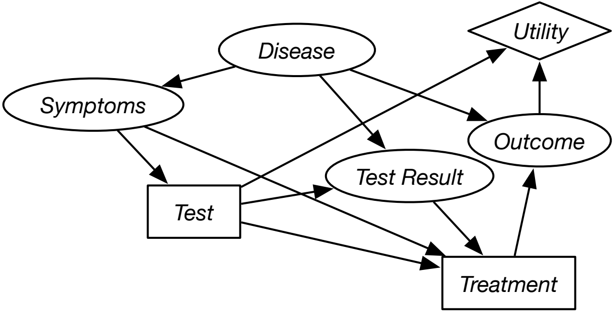


Reduced Alarm Example

Eliminate the non-observed variables for the final decision.

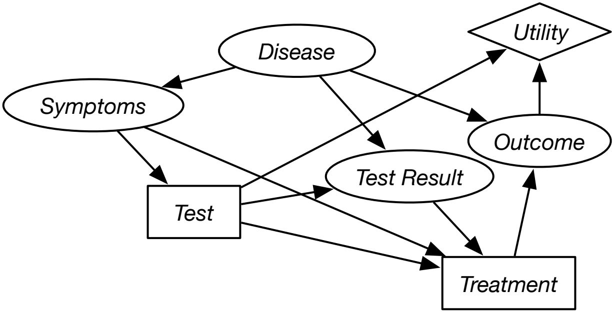


Exercise



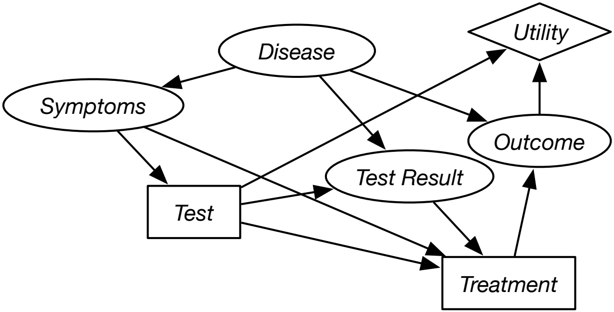
What are the factors?

Exercise



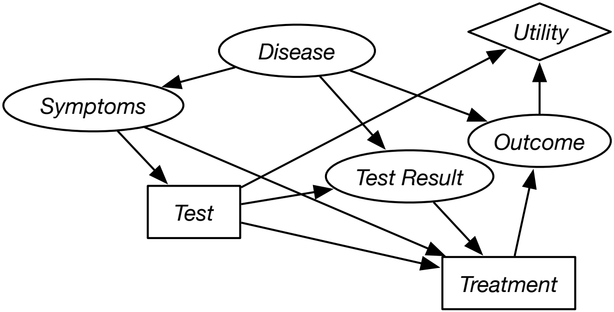
What are the factors?
Which random variables get summed out first?

Exercise



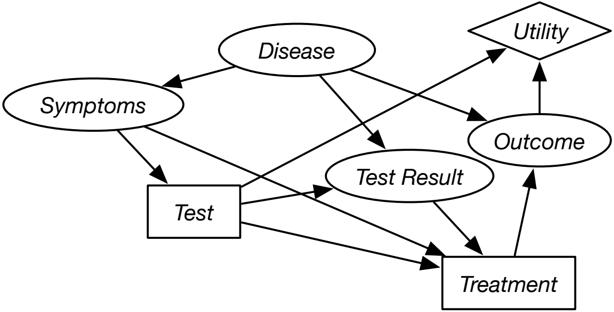
What are the factors?
Which random variables get summed out first?
Which decision variable is eliminated? What factor is created?

Exercise



What are the factors?
Which random variables get summed out first?
Which decision variable is eliminated? What factor is created?
Then what is eliminated (and how)?

Exercise



What are the factors?
Which random variables get summed out first?
Which decision variable is eliminated? What factor is created?
Then what is eliminated (and how)?
What factors are created after maximization?

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are assignments of values to the parents.

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are   different decision functions.

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are b^{2^k} different decision functions.
- The dynamic programming algorithm does optimizations

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are b^{2^k} different decision functions.
- The dynamic programming algorithm does 2^k optimizations

If there are multiple decision functions

- The number of policies is

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are b^{2^k} different decision functions.
- The dynamic programming algorithm does 2^k optimizations

If there are multiple decision functions

- The number of policies is the product of the number decision functions.
- The number of optimizations in the dynamic programming is

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are b^{2^k} different decision functions.
- The dynamic programming algorithm does 2^k optimizations

If there are multiple decision functions

- The number of policies is the product of the number decision functions.
- The number of optimizations in the dynamic programming is the sum of the number of assignments of values to parents.

Complexity of finding an optimal policy

Decision D has k binary parents, and has b possible actions:

- there are 2^k assignments of values to the parents.
- there are b^{2^k} different decision functions.
- The dynamic programming algorithm does 2^k optimizations

If there are multiple decision functions

- The number of policies is the product of the number decision functions.
- The number of optimizations in the dynamic programming is the sum of the number of assignments of values to parents.
- Searching through policy space is exponentially more complicated than dynamic programming.

Value of Information

- The value of information X for decision D is the utility of the network with an arc from X to D ($+$ no-forgetting arcs) minus the utility of the network without the arc.
- The value of information is always

Value of Information

- The value of information X for decision D is the utility of the network with an arc from X to D ($+$ no-forgetting arcs) minus the utility of the network without the arc.
- The value of information is always non-negative.
- It is positive only if

Value of Information

- The value of information X for decision D is the utility of the network with an arc from X to D ($+$ no-forgetting arcs) 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.
- The value of information provides a bound on how much an agent should be prepared to pay for a sensor. How much is a better weather forecast worth?

Value of Information

- The value of information X for decision D is the utility of the network with an arc from X to D ($+$ no-forgetting arcs) 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.
- The value of information provides a bound on how much an agent should be prepared to pay for a sensor. How much is a better weather forecast worth?
- We need to be careful when adding an arc would create a cycle. E.g., how much would it be worth knowing whether the fire truck will arrive quickly when deciding whether to call them?

Value of Control

- The value of control of a variable X is the value of the network when you make X a decision variable (and add no-forgetting arcs) minus the value of the network when X is a random variable.

Value of Control

- The value of control of a variable X is the value of the network when you make X a decision variable (and add no-forgetting arcs) 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.

Value of Control

- The value of control of a variable X is the value of the network when you make X a decision variable (and add no-forgetting arcs) 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. E.g., controlling a thermometer.

Value of Control

- The value of control of a variable X is the value of the network when you make X a decision variable (and add no-forgetting arcs) 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. E.g., controlling a thermometer.
- If you keep the parents the same, the value of control is always non-negative.