Decision Problem Examples

Let us have a random outcome based on known probabilities. Think about the following lotteries. Which one you prefer? **Answer intuitively**, you may maximize MEU after that.

Lottery A

1. 80% chance to gain $400
2. 100% chance to gain $300

Which one you prefer?

Lottery B

Which one from this pair?

1. 20% chance to gain $400
2. 25% chance to gain $300

Which one you prefer?

Money Utility

Lottery

Two lotteries again

- You get \$1000000
- or a 50% chance to get \$3000000, any gain otherwise.

Money utility

- The utility of money is not linear.
- Assume I have $\$k$. The utility to have n is roughly (\$):

$$U(S_{k+n}) = -263.31 + 22.09 log(n + 150000)$$

valid from –\$150000 to \$800000.
(Mr. Beard)

moneyutility.pdf

Decision Problem – Milk Example

- The farmer has 50 cows.
- The milk from each cow is poured into a common container and transported to the diary.
- The value of the milk is $2 per cow.
- The diary checks the milk carefully
  - and if it is infected it is thrown away.
- After having milked a cow, the farmer may perform two different tests
  - T_A costs 0.06 and it has a false positive/negative rate of 0.01
  - T_B costs 0.20 and it has a false positive/negative rate of 0.001.
- We assume the farmer has clean milk from the 49 other cows.
  - (Check general problem gives to the same strategy.)
- Putting the milk into the container, the farmer will gain $100 if it is not infected, $0 otherwise.
- Throwing it away, he will gain $98 regardless of the state of the milk.

Should he perform the tests and in which order?

Definition (Decision Tree)

(Probabilistic) Decision Trees

- A **decision tree** is a model that encodes the structure of the decision problem.
- The nonleaf nodes are
  - decision nodes (rectangular boxes) D_i
  - or chance nodes (circles or ellipses) X_j
- and the leaf nodes are utility nodes (diamond shaped) U_k.
- The links in the tree have labels.
- Link from a decision is labeled with the action chosen
- a link from a chance node is labeled by a state and the conditional probability of this state $P(X = x_j|\text{path from the root to X})$.
- **A path from the root represents the time order**:
  - the state of a random variable is known iff it is on the path from the root to the decision (nonforgetting).
- an utility node is labeled by the utility of the decision scenario from the root to it.

Decision Scenario

- We require the decision tree to be complete
  - from a chance node there must be a link for each possible state
  - from a decision node there must be a link for each possible decision.
- Each path from the root to the leaf specifies a complete sequence of observations and decisions
- we call such sequence a **decision scenario**.
- The decision tree specifies all the possible scenarios in the decision problem.

milk3.png

Expected utility (=Expected Value)

- We know the value of any scenario $V(d, x, e)$
- we do not know which scenario will take place.
- We maximize the **expected utility**

$$EU(d|e) = \sum_x V(d, x, e) \cdot P(x|d, e)$$

More value functions $V_1, \ldots, V_n$ we usually sum together
$V(U) = V_1(U) + \ldots + V_n(U)$

- multiplicative composition would be much simpler to evaluate.

Functions $V_i(U)$ may depend on different subsets of the universe U.

Probabilities

We calculate the probabilities.

```
inf <- cptable(~inf, values=c(0.0007,0.9993),levels=c('yes','no'))
test <- cptable(~test+inf, values=c(1,99,99,1),levels=c('pos','neg'))
```

milk2.png

milk3.png

Definition (Strategy)

- A solution to a decision tree is a **strategy** that specifies how we should act at the various decision nodes.
- An **optimal strategy** is a strategy with the maximal expected utility.

milk3.png

EU(X;T) Expected Utility for a decision tree

Let X be a node in a decision tree T. To caculate an optimal strategy and the maximum expected utility for the subtree rooted at X do:

- If X is a utility node, then return $U(X)$.
- If X is a chance node, then return $EU(X,T) = \sum_{x \in sp(X)} EU(child(X = x), T) \cdot P(X = x | past(X))$
- If X is a decision node, then
  - mark the arc labeled: $x' = arg \max_{x \in sp(X)} EU(child(X = x), T)$
  - and return $EU(X | past(X)) = \max_{x \in sp(X)} EU(child(X = x), T)$

milk3.png

$$0.9351 * 99.94 + 0.0649 * (-0.06) = 93.45$$
$$0.999993 * 99.94 + 0.000007 * (-0.06) = 99.9393$$
$$0.0107 * 97.94 + 0.9893 * 99.9393 = 99.9170$$

Decision Trees and Decision graphs (=Influence diagrams)

- Decision Tree
  - general problem representation and evaluation
  - grows fast, sub-trees may repeat
  - requires an independent probabilistic model
- Decision Graph (Influence Diagram)
  - decisions and utilities incorporated in the probabilistic model
  - an implicit definition of the decision tree
  - a more compact evaluation.

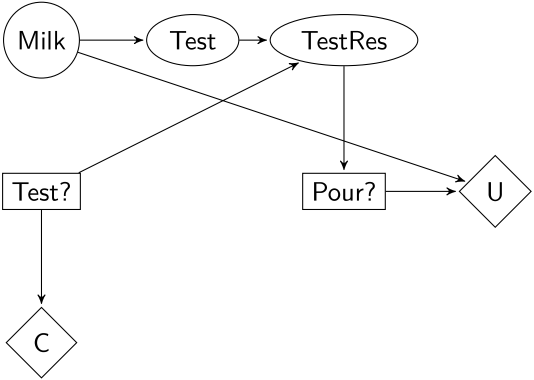
Decision graph (=Influence diagram)

Definition (Decision graph, Influence Diagram)

Decision graph is a DAG with three types of nodes and two types of tables:

- Rectangular **decision nodes** D_i have a finite domain of mutually exclusive values (decision choices). No table attached (will be attached as a solution)
- Elliptical **random nodes** are the same as in Bayesian networks: finite domain and a conditional probability table given parents
- Diamond **utility nodes** have no children and represent a function from the parent configurations to real numbers (values).
- Edges into random nodes represent conditioning as in Bayesian networks.
- **Edges into decision nodes** represent information flow: the random value is known before the decision is made
- We assume **non forgetting**.
- **Directed path ordering all decision is required**. (May contain also random variables).

Example - Milk (T.D.Nielsen)



Tables

- $P(Milk)$,
- $P(Test|Milk)$,
- $P(TestRes|Test, Test?)$,
- $U(Pour?, Milk)$,
- $C(Test?)$.

Artificial node *TestRes* to solve the asymmetry: the Test cannot be observed unless $Test = yes$.

Temporal ordering: $Test? \prec \{TestRes\} \prec Pour? \prec \{Milk, Test\}$.

Poker Probabilities

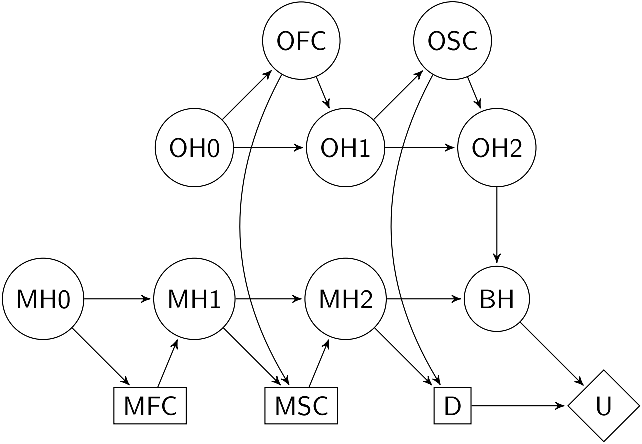
pokerprob.png

Example - Poker (T.D.Nielsen)

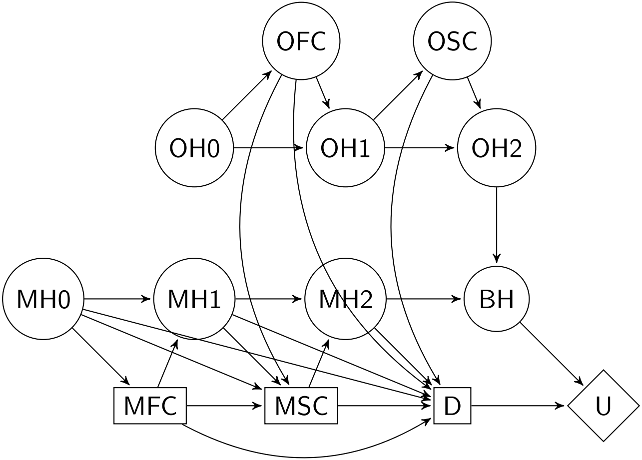
- Each player gets 5 cards
- FC the first choice: the player may change up to 3 cards
- SC the second choice: the player may change up to 2 cards
- each player may 'call' or 'fall'
- the highest hand takes the bank.

Poker Decision Graph

- Each player gets 3 cards
- FC the first choice: the player may change up to 3 cards
- SC the second choice: the player may change up to 2 cards
- each player may 'call' or 'fall'
- the highest hand takes the bank.

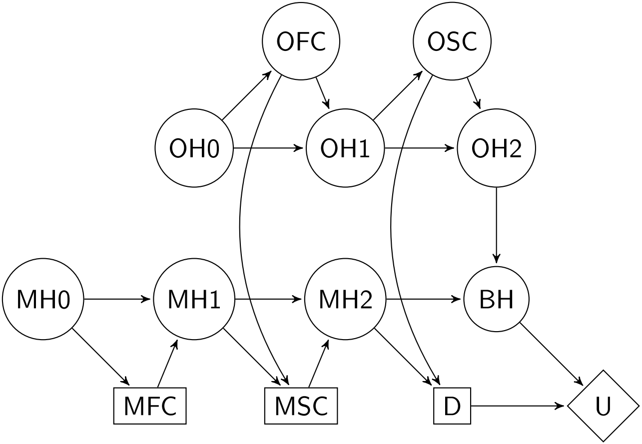


Poker – Non-forgetting Information Arcs



Partial Temporal Ordering

Temproral ordering: $\{MH0\} \prec MFC \prec \{MH1, OFC\} \prec MSC \prec \{MH2, OSC\} \prec D \prec \{OH0, OH1, OH1, GH\}$.



Decision Graph Evaluation

Definition

The **optimal strategy** for a decision graph is defined as the optimal strategy of a decision tree representing the same decision problem.

- Decision graph requires the **temporal ordering** which makes sufficient to evaluate a single decision tree.
  - Assume the temporal ordering of decisions $D_1, \ldots, D_n$.
  - We denote I_0 the set of random variables observable by D_1 (the parents of D_1)
  - generally, the set I_i are parents of D_{i+1} that are not parents of any previous D_i
  - I_n random variables that do not have any decision child.
- We get **a partial temporal ordering** of decision and random variables $I_0 \prec D_1 \prec I_1 \prec \ldots \prec D_n \prec I_n$. This ordering must be fulfilled in the decision tree.
  - The elements of a set I_k may be ordered arbitrary.

Chain Rule for Decision Graphs

Definition (Chain Rule for Decision Graphs)

Let $\mathcal{O}$ be the random variables and $D_1, \ldots, D_n$ decisions in a decision graph. Then

$$P(\mathcal{O}|D_1, \ldots, D_n) = \Pi_{X \in \mathcal{O}} P(X|pa(X)).$$

- According this rule we are able to calculate all conditional probabilities in the decision tree.
- In each utility leaf we sum appropriate values from all utility nodes in the decision graph $\sum_i V_i(\mathcal{O}, D_1, \ldots, D_n)$.
- The same optimal strategy can be evaluated also by a more compact way.

The Optimal Strategy

- For a given temporal ordering $I_0 \prec D_1 \prec I_1 \prec \ldots \prec D_n \prec I_n$ is the optimal strategy for D_i:

$$\sigma_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) =$$

$$argmax_{D_i} \sum_{I_i} max_{D_{i+1}} \ldots max_{D_n} \sum_{I_n} P(\mathcal{O}|D_1, \ldots, D_n) V(\mathcal{O}, D_1, \ldots, D_n)$$

- The expected value of the strategy starting in D_i is: $\rho_i(I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}) = \frac{1}{P(I_0, \ldots, I_{i-1}|D_1, \ldots, D_{i-1})} \cdot$

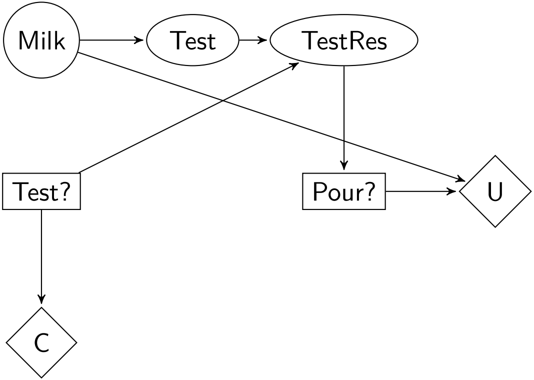
$$max_{D_i} \sum_{I_i} max_{D_{i+1}} \ldots max_{D_n} \sum_{I_n} P(\mathcal{O}|D_1, \ldots, D_n) V(\mathcal{O}, D_1, \ldots, D_n).$$

- The solution may be stored in the form of a **policy network**
  - Replace each decision D_i by a chance node D_i^o with parents $I_0, D_1, I_1, \ldots, D_{i-1}, I_{i-1}$.
  - For each parent configuration, set $P(D_i^o = d_j | pa(D_i^o)) = 1$ for the optimal decision $\sigma_i(pa(D_i^o))$
  - zero for all other choices.

Variable Elimination Algorithm Initialization

- $\Phi_0 \leftarrow$ all probability potentials $P(O_i|pa(O_i))$.
- $\Psi_0 \leftarrow$ all utility potentials $V_j(pa(V_j))$.
- We will sequentially eliminate all variables **in the reversed temporal order**. For each decision, we remember its strategy at the time it is eliminated.

Example - Milk Elimination Start

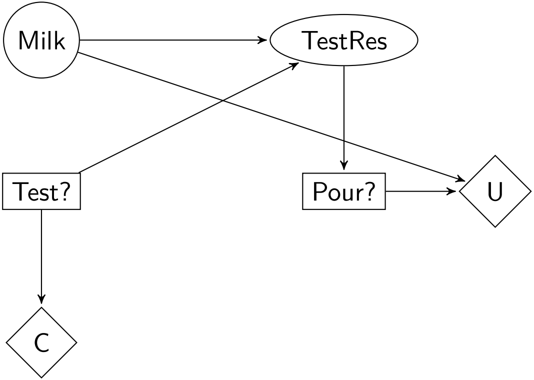


Tables

- $P(Milk)$,
- $P(Test|Milk)$,
- $P(TestRes|Test, Test?)$,
- $U(Pour?, Milk)$,
- $C(Test?)$.

Temporal ordering: $Test? \prec \{TestRes\} \prec Pour? \prec \{Milk, Test\}$.

Example - Milk Eliminate Test



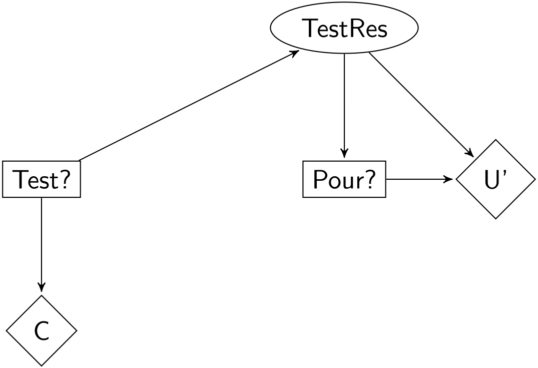
Tables

- $P(Milk)$,
- $\varphi(TestRes|Milk, Test?)$,
- $U(Pour?, Milk)$,
- $C(Test?)$.

$$\varphi(TestRes|Milk, Test?) \leftarrow \sum_{Test} P(Test|Milk)P(TestRes|Test, Test?)$$

Temporal ordering: $Test? \prec \{TestRes\} \prec Pour? \prec \{Milk\}$.

Example - Eliminate Milk

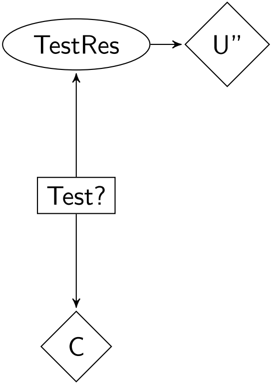


Tables

- $P(TestRes|Test?) \leftarrow \sum_{Milk} P(Milk)\varphi(TestRes|Milk, Test?)$,
- $U' \leftarrow \frac{1}{P(TestRes|Test?)} \sum_{Milk} P(Milk)\varphi(TestRes|Milk, Test?)U(Pour?, Milk)$,
- $C(Test?)$.

Temporal ordering: $Test? \prec \{TestRes\} \prec Pour?$.

Example - Eliminate Pour?

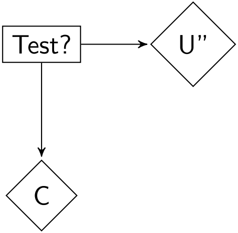


Tables

- $P(TestRes|Test?)$,
- $\max_{Pour?} U'(TestRes, Pour?)$,
- $C(Test?)$.

Temporal ordering: $Test? \prec \{TestRes\}$.

Example - Eliminate TestRes



Tables

- $U'' \leftarrow \sum_{TestRes} P(TestRes|Test?) \max_{Pour?} U'(TestRes, Pour?)$,
- $C(Test?)$.

Eliminate Test?

- $\max_{Test?}[U''(Test?) + C(Test?)]$.

Variable Elimination Algorithm (Decision Graphs)!

Eliminate X means:

1. $\Phi_X = \{\phi \in \Phi_{i-1} | X \in dom(\phi)\}$
   $\Psi_X = \{\psi \in \Psi_{i-1} | X \in dom(\psi)\}$
2. If X is a **random variable**
   $\phi_X = \sum_X \Pi\Phi_X$
   $\psi_X = \frac{1}{\phi_X} \sum_X \Pi\Phi_X \left(\sum \Psi_X\right)$
3. else X #**decision**
   $\phi_X = max_X \Pi\Phi_X$
   $\psi_X = max_X \left(\sum \Psi_X\right)$
4. always
   $\Phi_i = \Phi_{i-1} \setminus \Phi_X \cup \{\phi_X\}$
   $\Psi_i = \Psi_{i-1} \setminus \Psi_X \cup \{\psi_X\}$

For each decision D_i we store the optimal policy $\sigma_i(past) = argmax_{sp(D_i)} \psi_{D_i}$.

https://pypi.org/project/pycid/

Unconstrained Influence Diagrams

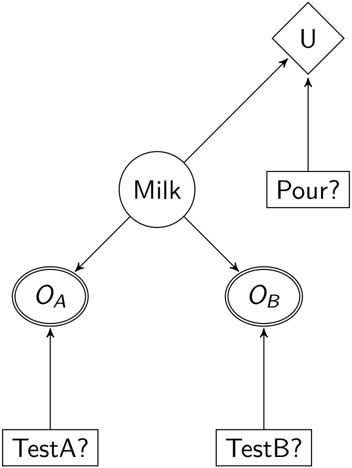
Definition (Unconstrained Influence Diagram)

An **Unconstrained Influence Diagram (UID)** U

- is a DAG
- over decision variables $\mathcal{D}_U$, chance variables $\mathcal{O}_U$ and utility variables.
- utility variables have no children.
- There are two types of chance variables
  - observables (double circled)
  - nonobservables (single circled).
- A nonobservable cannot have a decision as a child.
- Any decision has a cost (to simplify the graph).
- The partial temporal order induced by U is denoted by $\prec_U$.

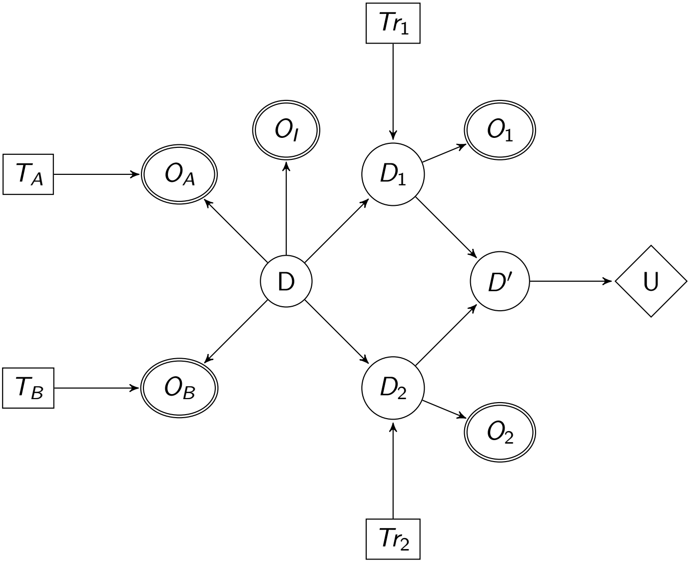
- An observable can be observed when all its antecedent decision variables have been decided on.
- In the case we say the observation is free and we release an observable when the last decision in its ancestral set is taken.

Example - UID Two Tests



Temporal ordering of decision is not fixed.

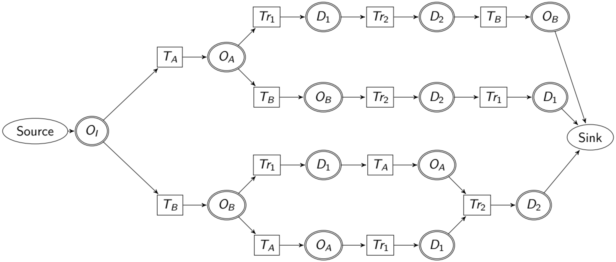
Example - UID Two Tests, Two Treatments



S-DAG – Solution Strategy for a UID

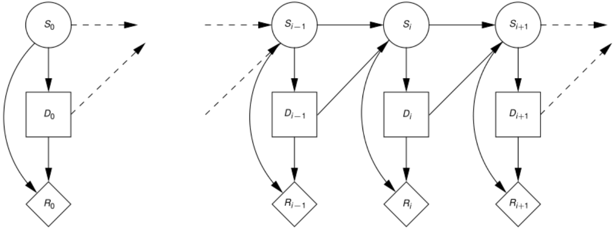
Definition (S–DAG)

Let U be a UID. An S–DAG is a directed acyclic graph G. The nodes are labeled with variables from $\mathcal{D}_U \cup \mathcal{O}_U$ such that each maximal directed path in G represents an admissible ordering of $\mathcal{D}_U \cup \mathcal{O}_U$.
We add tho unary nodes Source and Sink, Source is the only node with no parents and Sink is the only node with no children.
A **strategy for U** is a step policy for each node of the S-DAG together with a decision policy for each decision node.



Further Variants of IDs

- LIMIDs - Limited Memory IDs
- languages for asymmetric decision scenarios (Valuation networks,AIDs)
- CEG - Chain Event Graphs - closed to the coalescent decision trees.
- Repetitive in the time



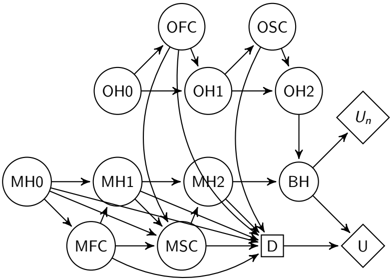
Influence Diagrams: Ommited Topics

- Strong Junction Tree - slightly more effective evaluation
- Approximate inference - Monte Carlo Sampling
- LIMIDs - Limited Memory IDs - intentionally restrict the domains for decisions
- languages for asymmetric decision scenarios (Valuation networks,AIDs)
- CEG - Chain Event Graphs - closed to the coalescent decision trees.
- Unconstrainded influence diagrams (no ordering on decisions required).

$\Rightarrow$ Reasoning on the structure of the influence diagram.

- Influence diagram $\mathcal{M}$ consists of
  - a DAG graph $\mathcal{G}$
  - a list of probability and utility potentials.

Value of Information



- Are all edges material?
- Is there a structural criterion?

Definition (Materiality, Schachter 2016)

For a single-decision influence diagram (or SCIM) $\mathcal{M}$, let $\mathcal{M}_{X \not\to D}$ be the model $\mathcal{M}$, modified by removing the edge $X \to D$, and let maximal expected utility in a model be $\mathcal{V}^*(\mathcal{M}) = max_\pi \mathbb{E}^\pi[\mathcal{U}]$.
The observation $X \in pa(D)$ is **material** if $\mathcal{V}^*(\mathcal{M}_{X \not\to D}) < \mathcal{V}^*(\mathcal{M})$.

Reference: Everitt, Tom & Carey, Ryan & Langlois, Eric & Ortega, Pedro & Legg, Shane. (2021). Agent Incentives: A Causal Perspective.

Nonrequeisite observation

Definition (Nonrequeisite observation, Lauritzen and Nilsson 2001)

- Let $\mathbf{U}^D = \mathbf{U} \cap desc(D)$ be the utility nodes downstream of D.
- An observation $X \in pa(D)$ in a single-decision ID (CID) $\mathcal{G}$ is **nonrequisite** if:

$$X \perp_d \mathbf{U}^D | (pa(D) \cup \{D\} \setminus \{X\}).$$

  In this case, the edge $X \rightarrow D$ is also called nonrequisite.
- Otherwise, X and $X \rightarrow D$ are **requisite**.

- Recall d-separation criterion.
- We distinguish;
  - the graphical structure $\mathcal{G}$
  - the model including the probability tables (and structural equations later) $\mathcal{M}$.

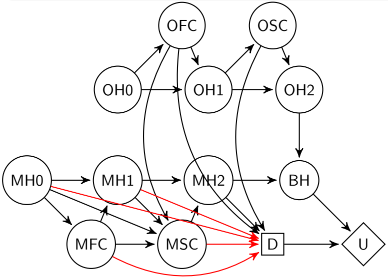
Value of Information

Definition (Value of Information)

- A node x has **value of information** *VoI* in a ID (SCIM) $\mathcal{M}$ if it is material in the model $\mathcal{M}_{X \to D}$ obtained by adding the edge $X \to D$ to $\mathcal{M}$.
- A ID (CID) $\mathcal{G}$ **admits VoI** for X if X as VoI in a ID $\mathcal{M}$ compatible with $\mathcal{G}$.

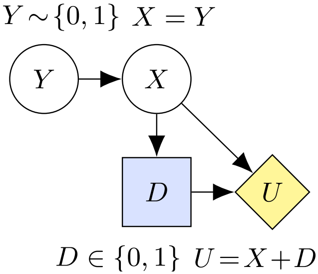
Theorem (Value of information criterion)

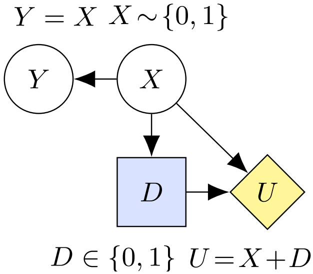
A single decision ID (CID) $\mathcal{G}$ admits VoI for $X \in V \setminus desc(D)$ if and only if X is a requisite observation in $\mathcal{G}_{X \to D}$, the graph obtained by adding $X \to D$ to $\mathcal{G}$.



Causality

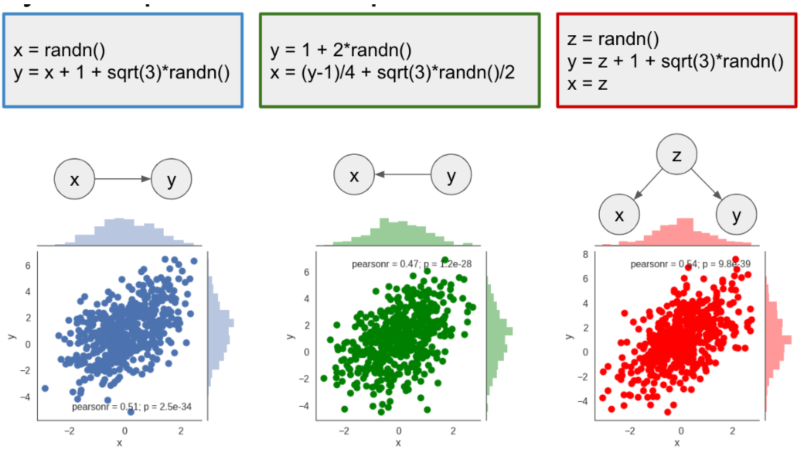
- Generally, a link in a BN does not have causal meaning.
  - The probabilistic relation between *Rain* and *WetGrass* may be represented by a link in any direction.
  - In an ID, the links from decision and the descendants need to represent causality.
  - Still, the link $X \to Y$ does not have to represent causality.
  - Further, we define causal models with all links causal.





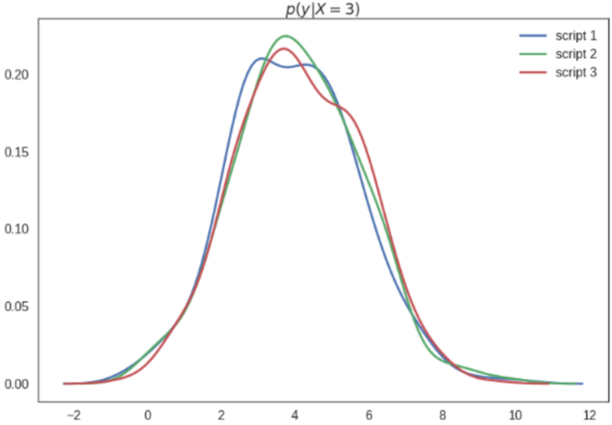
Causal Inference Example

https://www.inference.vc/causal-inference-2-illustrating-interventions-in-a-toy-example/

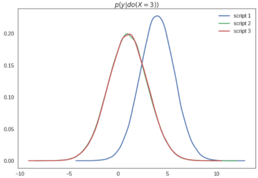


Observation

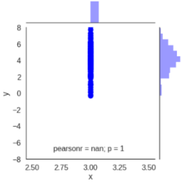
- The conditional probability $p(y|X = 3)$ is similar in all three cases.



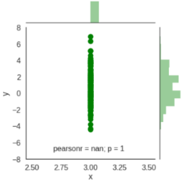
- The intervention sets the value $X = 3$ 'constantly'.
- The distributions differ.



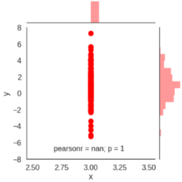

```
x = randn()
x = 3
y = x + 1 + sqrt(3)*randn()
x = 3
```



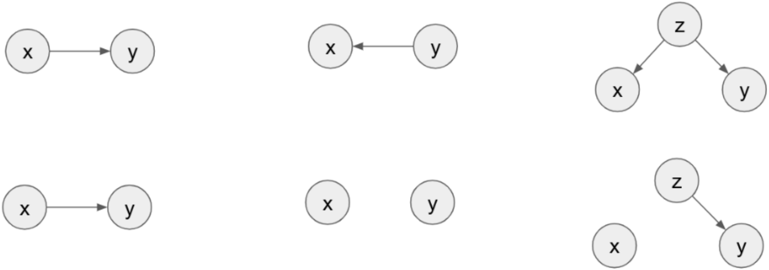

```
y = 1 + 2*randn()
x = 3
x = (y-1)/4 + sqrt(3)*randn()/2
x = 3
```




```
z = randn()
x = 3
x = z
x = 3
y = z + 1 + sqrt(3)*randn()
x = 3
```



Probabilistic model of the intervention



$P(y|do(X)) = p(y|x)$ $\quad$ $P(y|do(X)) = p(y)$ $\quad$ $P(y|do(X)) = p(y)$

- **do(X)** operator **disconnects** X **from its parents** and enters the evidence.

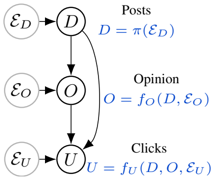
! We need a **causal graph**, not an arbitrary Bayesian network.

Structural Causal Model

Definition (Structural Causal Model, Pearl 2009, Chapter 7)

A **structural causal model** is a tuple $\langle \mathcal{E}, \mathbf{V}, \mathbf{F}, P \rangle$, where

- $\mathcal{E}$ is a set of **exogenous variables**
- $\mathbf{V}$ is a set of **endogenous variables**
- $\mathbf{F} = \{f_V\}_{V \in \mathbf{V}}$ is a collection of functions
  $f_V : dom(pa(V) \cup \mathcal{E}_V) \to dom(V)$
- The uncertainty is encoded through a probability distribution $P(\varepsilon)$ such that the exogenous variables are mutually independent.



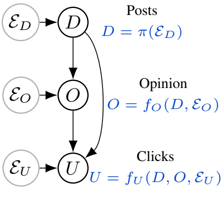
Structural Causal Influence Model

Definition (Submodel, Intervention)

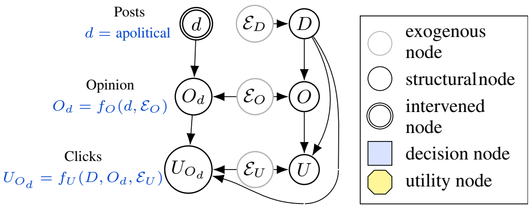
Let $\mathcal{M} = \langle \mathcal{E}, \mathbf{V}, \mathbf{F}, P \rangle$ be an SCM, X a set of variables in V, and x a particular realization on X. The submodel $\mathcal{M}_x$ represents the effect of an **intervention** $do(X = x)$, and is formally defined as the SCM $\langle \mathcal{E}, \mathbf{V}, \mathbf{F}_x, P \rangle$, where

$$\mathbf{F}_x = \{f_V | V \notin X\} \cup \{X = x\}.$$

The original functional relationships of $X \in \mathbf{X}$ are replaced with the constant function $X = x$.



(b) SCM

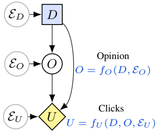


(c) SCM with nested counterfactual

Causal influence diagram

Definition (Causal influence diagram)

A **causal influence diagram** is a DAG $\mathcal{G}$ where the vertex set $\mathbf{V}$ is partitioned into **structure nodes** $\mathbf{X}$, **decision nodes** $\mathbf{D}$, and **utility nodes** $\mathbf{U}$. Utility nodes have no children.



(a) SCIM

Definition (Structural causal influence model)

A **structural causal influence model** is a tuple $\mathcal{M} = \langle \mathcal{G}, \mathcal{E}, \mathbf{F}, P \rangle$ where

- $\mathcal{G}$ is a CID with finite-domain variables V partitioned into X, D, U where utility variable domains are a subset of $\mathbb{R}$. We say that $\mathcal{M}$ is **compatible with $\mathcal{G}$**.
- $\{\mathcal{E}_V\}_{V \in \mathbf{V}}$ is a set of **exogenous variables**, one for each endogenous variable,
- $\mathbf{F} = \{f_V\}_{V \in \mathbf{V} \setminus \mathbf{D}}$ is a collection of **structural functions** $f_V : dom(pa(V) \cup \mathcal{E}_V) \to dom(V)$,
- $P(\varepsilon)$ such that the exogenous variables are mutually independent.

Response Incentives

Definition (Response Incentives)

Let $\mathcal{M}$ be a single-decision SCIM. A policy π **responds** to a variable X if there exists some intervention $do(X = x)$ and some setting $\mathcal{E} = \varepsilon$, such that $D_x(\varepsilon) \neq D(\varepsilon)$.
The variable $X \in \mathbf{X}$ has a **response incentive** if all optimal policies responds to X.
A CID **admits** a response incentive on X if it is compatible with a SCIM that has a response incentive on X.

Definition (Minimal reduction)

The **minimal reduction** $\mathcal{G}^{min}$ of a single-decision CID $\mathcal{G}$ is the result of removing from $\mathcal{G}$ all information links from nonrequisite observations.

Theorem (Response incentive criterion)

A single decision CID $\mathcal{G}$ admits a response incentive on $X \in \mathbf{X}$ if and only if the minimal reduction $\mathcal{G}^{min}$ has a directed path $X \dashrightarrow D$.

Incentivised unfairness

Definition (Counterfactual fairness, Kusner et al. 2017)

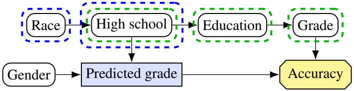
A policy is **counterfactually fair** with respect to a sensitive attribute A if

$$P^{\pi}(D_{a'} = d | pa(D), a) = P^{\pi}(D = d | pa(D), a)$$

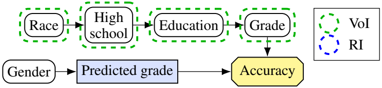
for every decision $d \in dom(D)$, every context $pa(D) \in dom(pa(D))$ and every pair of attributes $a, a' \in dom(S)$ with $P(pa(D), a) > 0$.

Theorem (Counterfactual fairness and response incentives)

In a single-decision SCIM $\mathcal{M}$ with a sensitive attribute $A \in \mathbf{X}$, all optimal policies π^* are counterfactually unfair with respect to A if and only if A has a response incentive.



(a) Admits response incentive on race



(b) Admits no response incentive on race

Value of Control

Definition (Value of control)

In a single-decision SCIM $\mathcal{M}$, a non-decision node X has **positive value of control** if

$$max_{\pi}\mathbb{E}^{\pi}[\mathcal{U}] < max_{\pi,g^X}\mathbb{E}^{\pi}[\mathcal{U}_{g^X}]$$

where $g^X : dom(pa(X) \cup \{\mathcal{E}_X\}) \rightarrow dom(X)$ is a soft intervention at X, i.e. new structural function for X that respects the graph.
A CID $\mathcal{G}$ **admits positive value of control** for X if there exists a SCIM $\mathcal{M}$ compatible with $\mathcal{G}$ where X has positive value of control.

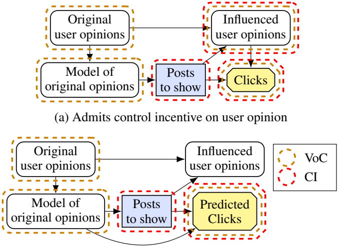
Theorem (Value of control criterion)

A single decision CID $\mathcal{G}$ admits positive value of control for a node $X \in \mathbf{V} \setminus \{D\}$ if and only if there is a directed path $X \dashrightarrow U$ in the minimal reduction $\mathcal{G}^{min}$.

Definition (Control Incentive)

In a single-decision SCIM $\mathcal{M}$, there is a **control incentive** on $X \in \mathbf{V}$ if for every optimal policy π^*, there exists a setting for parents of the decision $pa(D) \in dom(pa(D))$ with $P(pa(D)) > 0$ and an alternative decision $d \in dom(D)$ such that $\mathbb{E}^{\pi^*}[\mathcal{U}_{X_d}|pa(D)] \neq \mathbb{E}^{\pi^*}[\mathcal{U}|pa(D)]$.
A CID $\mathcal{G}$ **admits a control incentive** on X if there exists a SCIM $\mathcal{M}$ compatible with $\mathcal{G}$ in which there is a control incentive on X.

- In Pearl's terminology, a control incentive means that D has a **natural indirect effect** on U via X under all optimal policies.
- Can be viewed as **instrumental goal**.



(a) Admits control incentive on user opinion

(b) Admits no control incentive on user opinion

Theorem (Control incentive criterion)

A single decision CID $\mathcal{G}$ admits a control incentive on $X \in \mathbf{V}$ if and only if there is a directed path from the decision D to a utility node $U \in \mathbf{U}$ that passes through X, i.e. a directed path $D \dashrightarrow X \dashrightarrow U$.

Counterfactual

https://www.inference.vc/causal-inference-3-counterfactuals/

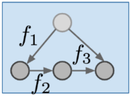
inFERENCe

Example (Counterfactual)

Given that Ferenc Huszár have a beard, and that Ferenc Huszár have a PhD degree, and everything else we know about him, with what probability would he have obtained a PhD degree, had he never grown a beard.

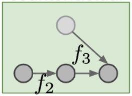
ϵ_1	ϵ_2	ϵ_3	
0.1	0.3	0.7	...
0.7	0.1	0.0	...
0.4	0.8	0.6	...
1.0	0.2	1.0	...
0.7	0.3	0.5	...

observed, factual



0	1	1	0
0	0	1	1
1	0	1	0
1	1	1	1
1	1	0	0

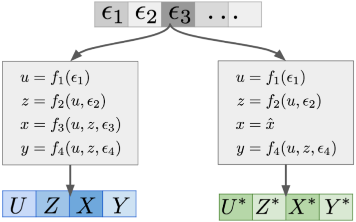
imagined, counterfactual



0	1	1	0
0	0	1	1
0	0	1	0
0	1	0	1
0	1	0	0

$p($🎓$^*|$🧔$^* = 0,$ 🧔 $= 1,$ 💍 $= 1,$ 💪 $= 1,$ 🎓 $= 1)$

Counterfactual II



We set: $p(y|do(X = \hat{x})p(y^*|X^* = \hat{x})$.

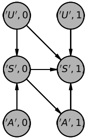
We may notice:

$$
\begin{aligned}
p(y|do(X = \hat{x})) &= p(y^*|X^* = \hat{x}) \\
= \int_{x,y,u,v} p(\quad y^* & \quad |X^* = \hat{x}, X = x, Y = y, U = u, Z = z)p(x, y, u, z)dxdydudz \\
&= \mathbb{E}_{p_{X,Y,U,Z}} p(y^*|X^* = \hat{x}, X = x, Y = y, U = u, Z = z).
\end{aligned}
$$

that is, $p(y|do(X = \hat{x})$ is the average of counterfactuals over the observable population.

More on causality: Sucar, Luis Enrique. "Probabilistic Graphical Models: Principles and Applications." Probabilistic Graphical Models (2021)

Dynamic Bayesian Networks

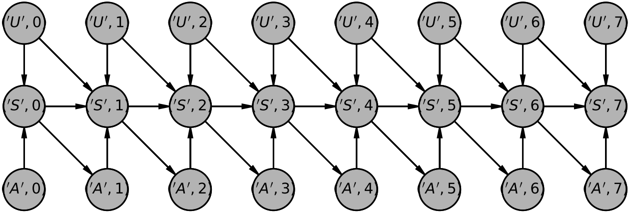


- We ame to monitor a process in time.
- We asume a constant BN that represents
  - intra edges and parameters inside one time slice
  - inter edges and parameters from one time slice to another
  - initial probability distributions

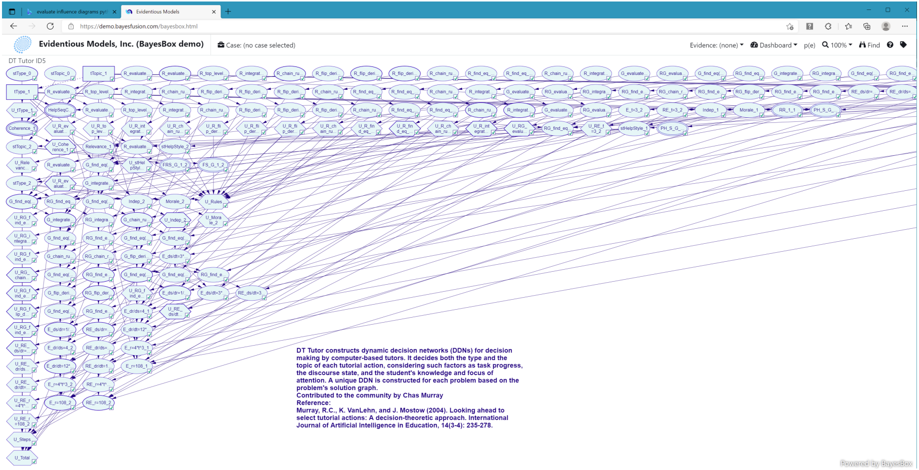
?

Are variables U0 and A0 independent given A3?

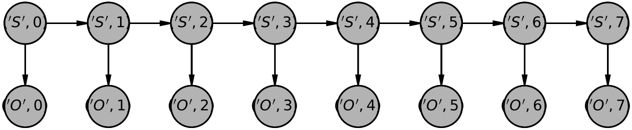
- Observations usually make non-observable variables dependent.
- Hidden Markov models 'join' each time slice to one 'product' variable S.
  - DBN is always useful for the input specification.



Dynamic Influence Diagram



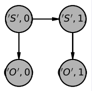
Hidden Markov Model



Definition (Hidden Markov Model)

Hidden Markov Model is defined by

- p the number of hidden states, possible values of S_i
- m the number of observation O_i per state
- N the length of observation/prediction sequence
- Initial probability distribution $P(S_0)$
- State transition probabilities, $P(S_{t+1} = j | S_t = i)$
- Observation distribution per state $P(O_t = k | S_t = i)$.

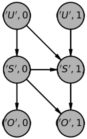


- Filtering, Smoothing = a special case of the evidence propagation
- Baum-Welch algorithm = a special case of the EM algorithm.

HMM and LSTM comparison

Manie Tadayon and Greg Pottie: Comparative Analysis of the Hidden Markov Model and LSTM: A Simulative Approach, (2020), https://arxiv.org/abs/2008.03825

- The authors simulated data from a DBN.
- Learned a HMM and a LSTM and compared the results.
- Several DBNs, HMMs and LSTMs were tested.



- DHMM has much less parameters to train.
  - LSTM 4484 parameters
  - DHMM 27 parameters
- It may perform well. It may perform even better than LSTM with little training data.

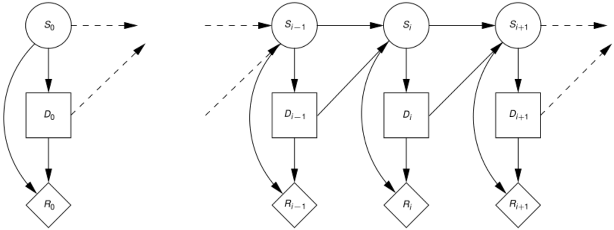
Number of Samples	LSTM Accuracy(%)	DHMM Accuracy(%)
8000	61.59	60.95
3000	58.36	60.01
1000	56.12	57.90
50	33.84	50.16
10	30.23	37.20

Markov Decision Processes

- We assume a finite set of states S in each time t
- **First order Markov property** the state $t+1$ does not depend on $t-i$, $i > 0$ given the state t, that is:

$$S_{t+1} \perp\!\!\!\perp S_{t-i} | S_t$$

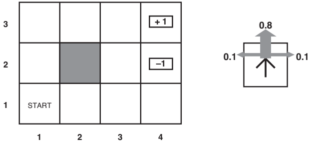
- Higher order Markov processes condition by more time slots.



Definition (Markov Decision Processes MDP)

Markov Decision Processes is defined by:

- Finite set of states S, $S_i = S$ for any time $t \in \mathbb{N}_0$,
- Initial state s_0
- The set of possible actions (decisions) at any time A
- Transition matrix $T(s, a, s^|) \equiv P(s^| | s, a)$
- Reward(=utility) $R(s, a, s^|)$ for each state (and action).
- (discount factor $\gamma \in < 0, 1 >$).



(a) (b)

$R(s) = -0.04, \gamma = 1$

Cumulative payoff

The reward is summed through the time.
There are two approaches:

- finite horizon MDP – we set the number of steps $n \in \mathbb{N}$ in advance
  - this leads to a standard influence diagram (=decision graph)
- infinite horizon and a **discount factor** γ, $0 < \gamma < 1$ to make the infinite sum finite. We maximize

$$\mathbb{E}\left(U(s_0, \ldots, s_t, \ldots)\right) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t)\right) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) | \pi\right)$$

  - We maximize the expected value due to probabilistic outcome of actions.
- γ corresponds to the interest rate $\frac{1}{\gamma} - 1$ we have to pay.
- the sum is finite since $U(s_0, \ldots, s_t, \ldots) \leq \frac{R_{max}}{(1-\gamma)}$.

Strategy (policy)

A solution is a **strategy** π^* that maximizes the expected reward.

$$\pi^* = argmax_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) | \pi\right]$$

- For a finite horizon, the strategy is not stationary. It depends on the number of steps to the end. $\pi : History \rightarrow A$
- Infinite horizon leads to a **stationary** strategy. The optimal choice of an action does not depend on the number of steps passed.
- It is easier to represent a stationary strategy $\pi : S \rightarrow A$.
- In case of certainity to reach a goal state we may use $\gamma = 1$.

Value Iteration Algorithm for MDP

Value Iteration Algorithm for MDP

input: MDP, states S, transitions T, reward $R \geq 0$, discount f. γ,ε

vars: U, U', vectors of utilities of states S, initialize $U' \leftarrow 0^{|S|}$

$\quad\quad\delta$ maximal U change in the current cycle

repeat

$\quad U \leftarrow U'; \delta \leftarrow 0$

$\quad$**for each** state s in S **do**

$\quad\quad U'[s] \leftarrow R[s] + \gamma max_a \sum_{s'} T(s,a,s')U(s')$

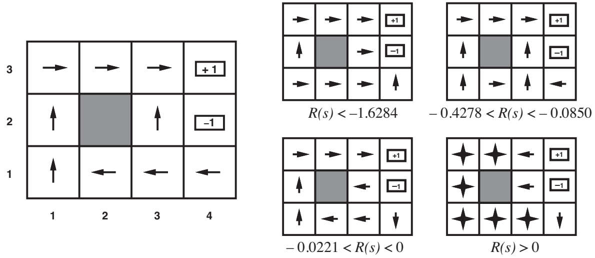
$\quad\quad$**if** $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$

until $\delta < \varepsilon(1-\gamma)/\gamma$

return U'

3	0.812	0.868	0.918	+ 1
2	0.762		0.660	–1
1	0.705	0.655	0.611	0.388

Bellman Equations for the Optimal Strategy



(a) (b)

- The evaluation of $POLICY\_VALUE(\pi, U, MPD)$ requires solution of $|S|$ linear **Bellman equations** for $U[s]$.

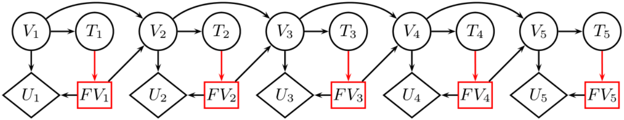
$$U_i[s] = R(s, \pi(a)) + \gamma \sum_{s^|} T(s, \pi(a), s^|) U_{i-1}[s^|]$$

Policy Iteration Algorithm for MDP

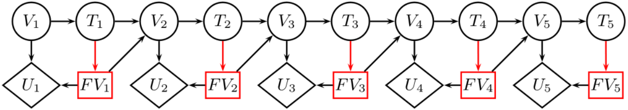
input: MDP, states S, transitions T, reward R, discount f. γ
vars: U, a vector of utilities of states S, initialize $U \leftarrow 0^{|S|}$
π policy, initialize at random
repeat
$U \leftarrow POLICY\_VALUE(\pi, U, MPD)$
unchanged? $\leftarrow$ *true*
for each state s in S **do**
if $max_a \sum_{s^|} T(s, a, s^|)U[s^|] > \sum_{s^|} T(s, \pi[s], s^|)U[s^|]$
then
$\pi[s] \leftarrow argmax_a \sum_{s^|} T(s, a, s^|)U[s^|]$
unchanged? $\leftarrow$ *false*
until *unchanged*?
return π

- The only difficulty may be a huge number of states like 10^5 equations for 10^5 variables.
- There are hybrid algorithms of value and policy iteration (for example prioritized sweeping).

Example



- The process above is not a Markov process.
- $\sigma_{FV_5}(T_1, FV_1, T_2, FV_2, T_3, FV_3, T_4, FV_4, T_5)$ is a very large table.
- We approximate.
- Consider the second model and eliminate variables V_i to get a Markov process



  on $S_i \equiv T_i$.
- $\sigma_{FV_5}(T_1, FV_1, T_2, FV_2, T_3, FV_3, T_4, FV_4, T_5) = \sigma_{FV_5}(T_5^|)$.
  - $\sigma_{FV_5}(T_5^|)$ is small (not larger than the MDP specification).
- We do not have to approximate. Let us introduce the POMPD.

Partially Observable Markov Decision Processes (POMDP)

- We are not able to observe the state directly.
- Our observations are noisy.
- The ideas:
  - The process is Markov with respect to the **belief** on states.
    - = the probability distribution on states
    - there are infinitely many such distributions (a continuous space)
- Hidden Markov Model + Decisions + Rewards = Partially Observed Markov Decision Processes.

POMDP

Definition (Partially Observed Markov Decision Processes POMDP)

Partially Observed Markov Decision Processes is defined by:

- Finite set of states S, $S_i = S$ for any time $t \in \mathbb{N}_0$,
- → Initial **belief** $b_0(s) = P(S_0)$
- The set of possible actions (decisions) at any time $A = \{a_1, \ldots, a_{|A|}\}$
- → a set of observations $Z = O = \{z_1, \ldots, z_{|Z|}\}$
- Transition matrix $T(s_{t-1}, a_{t-1}, s_t) = Pr(s_t | s_{t-1}, a_{t-1})$
- → observation matrix $O(s_t, a_{t-1}, z_t) = Pr(z_t | s_t, a_{t-1})$
- Reward(=utility) $R(s, a)$ for each state (and action).
- (discount factor $\gamma \in < 0, 1 >$).

We maximize the expected cumulative reward $max_\pi \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^t R(s_t, a_t)\right]$.

MDP The policy is a function of the state $\pi(s)$

)MDP The policy is a function of the history $\pi(a_{t-1}, z_{t-1}, \ldots, z_1, a_0, b_0)$

- or a function of the **belief**: $b : S \to \langle 0, 1 \rangle$, $\pi(b)$

Tiger Example

The example is a variant of the Monty Hall problem.

- We face two doors.
  - There is a tiger behind one door,
  - there is a gold brick behind the other.
- The Tiger is left or right $S = \{left, right\}$
- We may open any door or listen $A = \{left, right, listen\}$,
- we search optimal policy for given observation and reward tables.
- We observe Z only if we **listen** - we listen the tiger left TL or right TR
- we reset the world at the beginning and after opening any door:
  - the initial belief $P(S_0) = \langle 0.5, 0.5 \rangle$
- The reward R is a function of the state and the action
  - $U(gold, l/r) = 10$, $U(tiger, l/r) = -100$, $U(*, listen) = -1$, that is

$\frac{Tiger}{Action}$	left	right
Listen	-1	-1
left	-100	10
right	10	-100

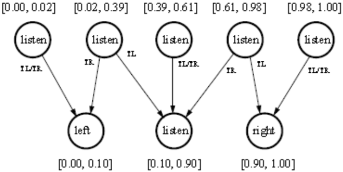
Z | S=?, A=listen	left	right
TL	0.85	0.15
TR	0.15	0.85
NoInfo	0	0

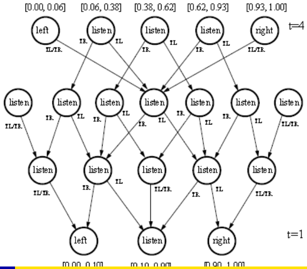
POMPD

Finite horizon POMPD t, $\gamma = 1$:

- $t = 1$
  $EU_{t=1}(A = left/right) = \frac{-100+10}{2} = -45$
  $EU_{t=1}(A = listen) = -1$
- horizon $t = 2$

$$T(s_{t-1}, a_{t-1}, s_t) = Pr(s_t | s_{t-1}, a_{t-1})$$
$$O(s_t, a_{t-1}, z_t) = Pr(z_t | s_t, a_{t-1})$$

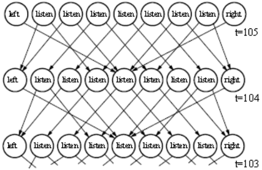




- and $t = 4$

Infinite Horizon

- $\gamma = 0.75$
- we iterate until convergence
- Then, we create a graph by joining two successive time slices together.
- We may omit nodes that are not reachable from the initial belief $b_0(s) = 0.5$.



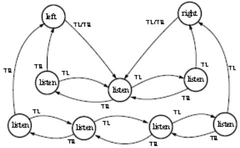


Figure 16: Policy graph for tiger example

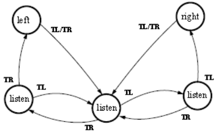


Figure 17: Trimmed policy graph for tiger example

Markov with respect to *belief* over states

- The history is aggregated in the probability distribution over states
  - history $h_t = \{a_0, z_1, a_1, \dots, z_{t-1}, a_{t-1}, z_t\}$
  - **belief** $b_t(s) = P(S = s | z_t, a_{t-1}, \dots, a_0, b_0)$,
  - initial belief $b_0(s) = P(S_0)$.
  - In the tiger example a single number $b(left)$, since the other probability is $1 - b(left)$.
- We update belief after any iteration. The update consists of:
  - a transition - we eliminate unobserved s_{t-1}
  - an observation - we condition by z_t.
- belief update

$$\begin{aligned}\tau(b_{t-1}, a_{t-1}, z_t) &= b_t(s^|) \\ &= \frac{\Sigma_s O(s^|, a_{t-1}, z_t) T(s, a_{t-1}, s^|) b_{t-1}(s)}{Pr(z_t | b_{t-1}, a_{t-1})}\end{aligned}$$

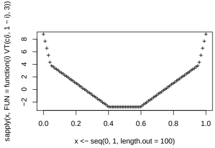
- Markov with respect to b since τ does not depend on time.

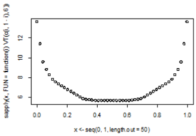
Strategy, Value function

- **Strategy (policy)** is a function $\pi(b) \rightarrow a$,
- optimal strategy maximizes the expected discounted cumulative reward

$$\pi^*(b_0) = argmax_{\pi}\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}(\gamma^t \cdot r_t)|b_0\right]$$

- **value function**
  - initial $V_0(b) = max_a \sum_{s\in S} R(s,a)b(s)$
  - recursively
    $V_t(b) = max_a \left[\sum_{s\in S} R(s,a)b(s) + \gamma \sum_{z\in Z} P(z|a,b)V_{t-1}(\tau(b,a,z))\right]$,
- optimal strategy for the horizon t:
  $\pi_t^*(b) = argmax_a \left[\sum_{s\in S} R(s,a)b(s) + \gamma \sum_{z\in Z} P(z|a,b)V_{t-1}(\tau(b,a,z))\right]$.





α vectors

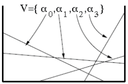
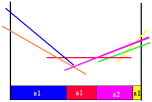


Figure 1: POMDP value function representation



$$|\Gamma_t| = O(|A| \cdot |\Gamma_{t-1}|^{|Z|})$$

- *value* function $V_t(b)$ can be represented by a finite number of hyperplanes
  - each hyperplane is represented as a vector α $V_t(b) \Leftrightarrow \Gamma_t = \{\alpha_0, \alpha_1, \dots, \alpha_m\}$.
  - initial: $\Gamma_0(b) = \{\langle R(s_1, a), R(s_2, a), \dots, R(s_{|S|}, a)\rangle\}_{a \in A}$
  - at the time t: $V_t(b) = max_{\alpha \in \Gamma_t} \sum_{s \in S} \alpha(s) b(s)$.
- From
  - $V_t(b) = max_a \left[\sum_{s \in S} R(s, a) b(s) + \gamma \sum_{z \in Z} P(z|a, b) V_{t-1}(\tau(b, a, z))\right]$:
  - $\tau(b_t, a_t, z_{t+1}) = \frac{\Sigma_s O(s^{|}, a_t, z_{t+1}) T(s, a_t, s^{|}) b_t(s)}{Pr(z_{t+1}|b_t, a_t)}$

$$V_t(b) = max_a \left[\sum_{s \in S} R(s, a) b(s) + \gamma \sum_{z \in Z} max_{\alpha \in \Gamma_{t-1}} \sum_{s' \in S} \sum_{s \in S} T(s, a, s') O(s', a, z_t) \alpha(s') b(s)\right]$$

One Step of the Time Update

- temporal sets $\forall \alpha_i \in \Gamma_{t-1}$:

$$\begin{aligned} \Gamma_t^{a,+} &\leftarrow \alpha^{a,+}(s) = R(s,a) \\ \Gamma_t^{a,z} &\leftarrow \alpha^{a,z}(s) = \gamma \sum_{s' \in S} T(s,a,s') O(s',a,z) \alpha(s'), \end{aligned}$$

- The utility for the action a summed over possible observation results z_j:

$$\Gamma_t^a = \Gamma_t^{a,+} + \Gamma_t^{a,z_1} \oplus \Gamma_t^{a,z_2} \oplus \ldots \oplus \Gamma_t^{a,z_m}$$

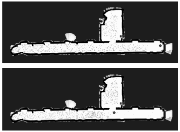
- the new value function for the time t: $\Gamma_t \leftarrow \bigcup_{a \in A} \Gamma_t^a$.
- We remove all α that are dominated by others
  - there are strategies to remove them earlier
  - or to avoid to generate many of them at all $|\Gamma_t| = O(|A| \cdot |\Gamma_{t-1}|^{|Z|})$.

https://h2r.github.io/pomdp-py/html/index.html

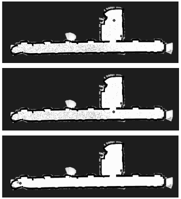
Approximation - We evaluate only some *b* points

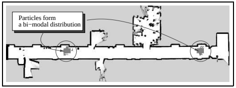
- Pineau & all.: Anytime Point-Based Approximations for Large POMDPs, JAIR 2006
- Pearl the Nursebot
- Find a person

(a) t=1
(b) t=7



c) t=12
d) t=17
e) t=29





(a) Original Belief


(b) Reconstruction

Approximation - We evaluate only some b points

- We evaluate the *belief* only in a finite number of points
- only one α vector for each point

$$\begin{aligned} \Gamma_t^{a,+} &\leftarrow \alpha^{a,+}(s) = R(s,a) \\ \Gamma_t^{a,z} &\leftarrow \alpha^{a,z}(s) = \gamma \sum_{s' \in S} T(s,a,s') O(s',a,z) \alpha(s'), \end{aligned}$$

- max for FINITE number of $b \in B$

$$\begin{aligned} \alpha_b &= argmax_a \left[\sum_{s \in S} R(s,a) b(s) + \sum_{z \in Z} argmax_{\alpha \in \Gamma_t^{a,z}} \sum_{s \in S} \alpha(s) b(s) \right] \\ \Gamma_t &= \bigcup_{b \in B} \{\alpha_b\} \end{aligned}$$

- The number of αs does not increase (with respect to the size of B).

POMDP Evaluation for the Fixed Number of B Points

procedure BACKUP(B,Γ_{t-1})
  for each action $a \in A$ **do**
    for each observation $z \in Z$ **do**
      for each solution vector $\alpha_i \in \Gamma_{t-1}$ **do**
        $\alpha^{a,z}(s) = \gamma \sum_{s' \in S} T(s,a,s')O(s',a,z)\alpha(s'), \forall s \in S$
      end for
      $\Gamma_t^{a,z} = \cup_i \alpha^{a,z}(s)$
    end for
  end for
  $\Gamma_t = \emptyset$
  for each belief point $b \in B$ **do**

$$\alpha_b = argmax_a \left[\sum_{s \in S} R(s,a)b(s) + \sum_{z \in Z} argmax_{\alpha \in \Gamma_t^{a,z}} \sum_{s \in S} \alpha(s)b(s)\right]$$

    if $\alpha_b \notin \Gamma_t$ **then**
      $\Gamma_t = \Gamma_t \cup \alpha_b$
    end if
  end for
  return Γ_t
end procedure

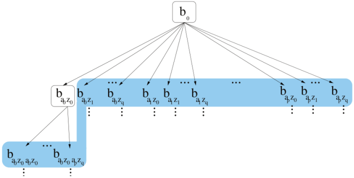
Iterative Number of Points POMDP

procedure PBVI-MAIN(B_{Init},Γ_0,N,T)
  $B = B_{Init}$
  $\Gamma = \Gamma_{Init}$
  for N expansions **do**
    for T iterations **do**
      $\Gamma = BACKUP(B, \Gamma)$
    end for
    $B_{new} = EXPAND(B, \Gamma)$
  end for
  return Γ
end procedure

T either a horizon or we select a error bound $\gamma^t ||V_0^* - V^*||$.

Expand: New Points Selection

1) at random



2) greedy maximal error improvement

- b' a new candidate
- the upper error bound in b'

$$\epsilon(b') \leq \min_{b\in B} \sum_{s\in S} \begin{cases} (\frac{R_{max}}{1-\gamma} - \alpha(s))(b'(s) - b(s)) & b'(s) \geq b(s) \\ (\frac{R_{min}}{1-\gamma} - \alpha(s))(b'(s) - b(s)) & b'(s) < b(s) \end{cases}$$

- b on the fringe, the error weighted by the probability of observations:

$$\begin{aligned} \epsilon(b) &= \max_{a\in A} \sum_{z\in Z} O(b,a,z)\epsilon(\tau(b,a,z)) \\ &= \max_{a\in A} \sum_{z\in Z} \left[\sum_{s'\in S} \sum_{s\in S} T(s,a,s')O(s',a,z)b(s) \right] \epsilon(\tau(b,a,z)). \end{aligned}$$

QMDP Approximation

- QMDP underestimates the state uncertainty in the POMDP.

procedure $\mathrm{QMDP}(\ b\ )$
    $\hat{V} = MDP\_discrete\_value\_iteration()$
    for each action $a \in A$ **do**
        for each state $s \in S$ **do**
            $Q(s,a) = R(s,a) + \gamma \sum_{s' \in S} \hat{V}(s) p(s'|a,s)$
        end for
    end for
    return $\arg\max_a \sum_{s \in S} b(s) Q(s,a)$
end procedure

Table of Content

Summary Links

- BN basics
  - Bayesian Network, Conditional Independence, Separation, d-separation, Markov Blanket, ...
  - Naive Bayes Classifier, Functions MI, KL, CMI, loglik, BIC, AIC
- BN Evaluation
  - Variable Elimination Algorithm, Junction Tree Algorithm
  - Likelihood weighting, Gibbs Sampling, (Metropolis Hastings Sampling)
- Parameter Learning
  - Frequency Ratio, Dirichlet, BDeu priors, Bayesian Learning BO, MAP, ML, Missing Data, EM algorithm
- Structure Learning
  - Chow–Liu Tree, Learning TAN Classifier
  - Myopic Structure Search, PC–Algorithm, (Structural EM)
- Gaussian Variables
  - Gaussian Graphical Models, Graphical Regression, GGM Model Selection (deviance, idev, lrt)
  - Gaussian Process, (Bayesian Optimization)
- Decisions
  - Decision Tree, DT Evaluation
  - Decision Graphs =IDs, Variable Elimination for DG
  - (Markov Decision Processes, Value Iteration Algorithm,) Partially Observed Markov Decision Processes, Policy Graph
- Variational Approximation
  - Variational Approximation (, Latent Dirichlet Allocation).