Chapter 5:
Solving Probabilistic Networks

Numerical **I**ntelligent **S**ystems **L**aboratory

Introduction

- The purpose of the knowledge base is to support our reasoning about events and decisions in a domain with inherent uncertainty.
- Expert system = Knowledge base + Inference engine.
- Knowledge base: Bayesian network / Influence diagram (probabilistic networks).
- Inference engine: Set of methods that applies the knowledge on the evidence to compute solutions.

5.1. Probabilistic Inference

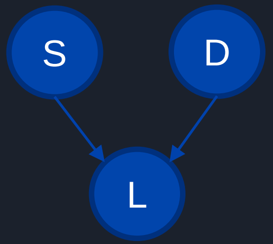
5.1.1. Inference in Discrete Bayesian Networks

Let $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P})$. $P(Y|\varepsilon)$ posterior marginal, $Y \in \mathcal{X}$. Assuming $\varepsilon = \emptyset$, if we apply the chain rule:

$$P(Y) = \sum_{X \in \mathcal{X} \setminus \{Y\}} P(\mathcal{X})$$

Prior marginal distribution

$$= \sum_{X \in \mathcal{X} \setminus \{Y\}} \prod_{X_v \in \mathcal{X}} P(X_v \mid X_{\mathrm{pa}(v)}).$$



$$P(L) = \sum_S \sum_D P(S) P(L \mid S, D) P(D).$$

$$P(L, S, D) = P(L|S, D) P(S, D)$$

$$P(L, S, D) = P(L|S, D) P(S) P(D)$$

$$P(L) = \sum_D \sum_S P(L, S, D)$$

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

To calculate the posterior marginal $P\left(X_{vj} \middle| \varepsilon\right)$, let $\varepsilon = \{\varepsilon_1, \dots \varepsilon_m\}$ be the set of evidence defined on $\mathcal{X}(\varepsilon)$:

$$
\begin{aligned}
P(X_{v_j} \mid \varepsilon) &= \eta(\cancel{P(\varepsilon \mid X_{v_j})} P(X_{v_j})) \\
&= \frac{\cancel{P(\varepsilon \mid X_{v_j}) P(X_{v_j})}}{P(\varepsilon)} = \frac{P(X_{v_j}, \varepsilon)}{P(\varepsilon)} \\
&\propto P(X_{v_j}, \varepsilon) \\
&= \sum_{Y \in \mathcal{X} \setminus \{X_{v_j}\}} P(\mathcal{X}, \varepsilon) \\
&= \sum_{Y \in \mathcal{X} \setminus \{X_{v_j}\}} \prod_{X_{v_i} \in \mathcal{X}} P(X_{v_i} \mid X_{\mathrm{pa}(v_i)}) \mathcal{E}_{\varepsilon} \\
&= \sum_{Y \in \mathcal{X} \setminus \{X_{v_j}\}} \prod_{X_{v_i} \in \mathcal{X}} P(X_{v_i} \mid X_{\mathrm{pa}(v_i)}) \prod_{X \in \mathcal{X}(\varepsilon)} \mathcal{E}_X
\end{aligned}
$$

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

- In general, probabilistic inference is an NP-hard task.
- The complexity of inference is polynomial when the graph is a poly-tree.
- The complexity is linear when the graph is a tree.
- The inference task is, in principle, solved by performing a sequence of multiplications and additions.
- The most critical problem related to the efficiency of the inference process is that of finding the optimal order in which to perform the computations.

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

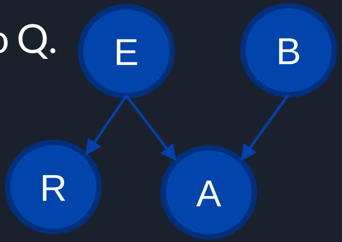
Query-based inference

We can consider the inference as a task of computing the posterior distribution of a set of variables.

Definition 5.1 (Query). Let $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P})$ be a Bayesian network model. A query Q is a three-tuple $Q = (\mathcal{N}, \mathcal{T}, \varepsilon)$ where $\mathcal{T} \subseteq \mathcal{X}$ is the target set and ε is the evidence set.

The solution of Q is the posterior distribution $P(\mathcal{T}|\varepsilon)$.

When inferring Q we can eliminate all barren and nuisance variables with respect to Q.

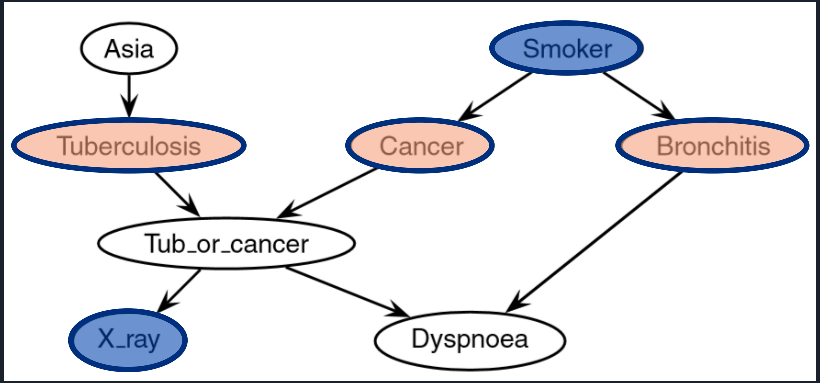


5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

Barren and Nuisance Variables Example

$\varepsilon = \{S = yes, X = yes\}$. Compute $P(Y|\varepsilon), Y = \{T, L, B\}$



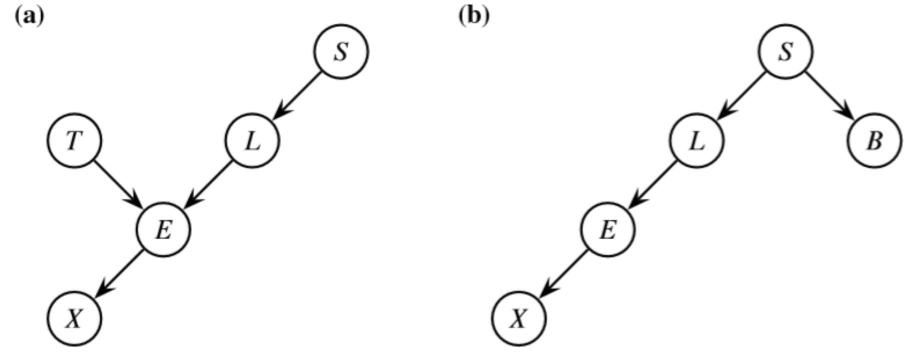
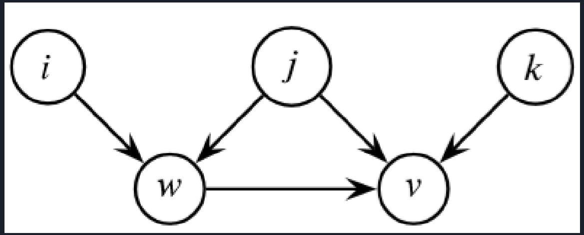


Fig. 5.1 The relevant networks for computing (**a**) $P(T\,|\,\varepsilon)$ and $P(L\,|\,\varepsilon)$ and (**b**) $P(B\,|\,\varepsilon)$

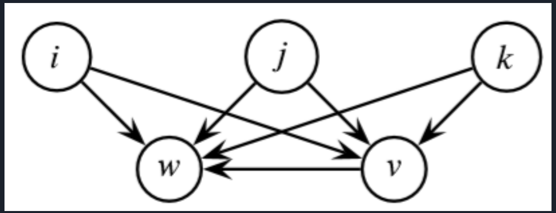
5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

Arc Reversal



$$P(X_v | X_i, X_j, X_k) = \sum_{X_w} \overbrace{P(X_w | X_i, X_j) P(X_v | X_w, X_j, X_k)}^{P(X_v, X_i, X_j, X_w, X_k)}$$



$$P(X_w | X_v, X_i, X_j, X_k) = \frac{P(X_w | X_i, X_j) P(X_v | X_w, X_j, X_k)}{P(X_v | X_i, X_j, X_k)}.$$

5.1. Probabilistic Inference
5.1.1. Inference in Discrete Bayesian Networks

Arc Reversal

For inference, we can perform a series of arc reversals and barren variable eliminations on the DAG $\mathcal{G}$ until a desired marginal or conditional is obtained.

In the Apple-Jack problem, if we want to calculate $P(L)$ using arc reversals:

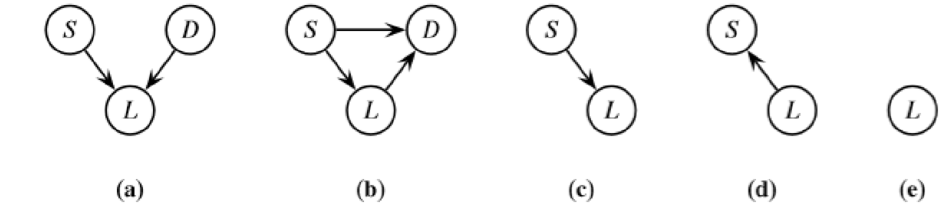


Fig. 5.3 Computing $P(L)$ by arc reversal

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

Graphical Representation of Inference

$$
\begin{aligned}
P(X_i) &= \sum_{X\in\mathcal{X}\setminus\{X_i\}} \prod_{X_v\in\mathcal{X}} P(X_v \mid X_{\mathrm{pa}(v)}) \\
&= \sum_{X\in\mathcal{X}\setminus\{X_i\}} \prod_{\phi\in\mathcal{P}\setminus\mathcal{P}_Y} \phi \prod_{\phi'\in\mathcal{P}_Y} \phi' \\
&= \sum_{X\in\mathcal{X}\setminus\{X_i,Y\}} \prod_{\phi\in\mathcal{P}\setminus\mathcal{P}_Y} \phi \; \boxed{\sum_{Y} \prod_{\phi'\in\mathcal{P}_Y} \phi'} \\
&= \sum_{X\in\mathcal{X}\setminus\{X_i,Y\}} \phi_Y \prod_{\phi\in\mathcal{P}\setminus\mathcal{P}_Y} \phi .
\end{aligned}
$$

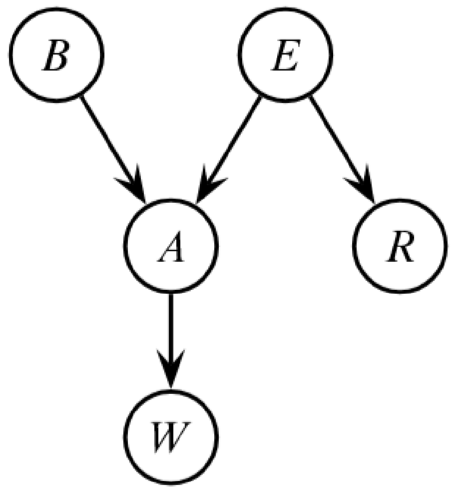
We want to eliminate Y

ϕ_Y: Probability potential obtained by eliminating Y

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

Graphical Representation of Inference



$$P(A) = \sum_E P(E) \sum_B P(B) P(A \mid B, E) \sum_R P(R \mid E) \sum_W P(W \mid A).$$

$$P(W) = \sum_A P(W \mid A) \sum_E P(E) \sum_B P(B) P(A \mid B, E) \sum_R P(R \mid E).$$

To solve a Bayesian network model $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P})$ is to compute the posterior marginal $P(X|\varepsilon)$ given a set of evidence ε for all variables $X \in \mathcal{X}$.

5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

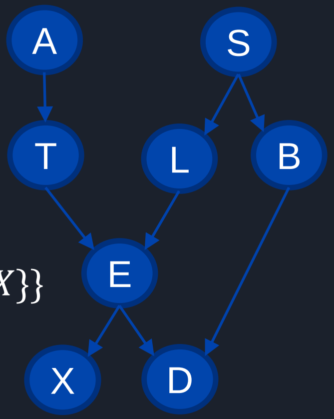
Junction Trees (*Join Tree, Markov Tree*)

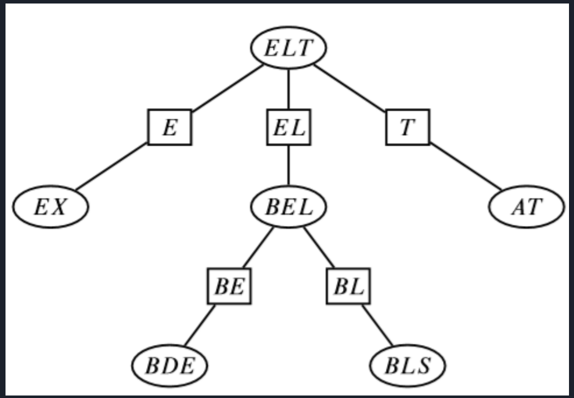
It's a secondary computational structure used to efficiently solve the task of probabilistic inference.
$\mathcal{T} = (\mathcal{C}, \mathcal{S})$ ($\mathcal{C}$: set of cliques (nodes), $\mathcal{S}$: set of separators (links))

$$P(X_i) = \sum_{X \in \mathcal{X}\setminus\{X_i\}} \prod_{X_v \in \mathcal{X}} \underbrace{P(X_v \mid X_{\mathrm{pa}(v)})}_{\text{Clique of } \mathcal{G}^m}$$

$$\mathcal{C} = \{\,\{A;T\};\{B;D;E\};\{B;E;L\};\{B;L;S\};\{E;L;T\};\{E;X\}\}$$

$$\mathcal{S} = \{\{B;E\};\{B;L\};\{E\};\{E;L\};\{T\}\}$$





5.1. Probabilistic Inference
5.1.1. Inference in Discrete Bayesian Networks

Junction Trees (*Join Tree, Markov Tree*)

- They are used for organizing the computations performed during probabilistic inferences.
- During construction, a probability potential is assigned with each C∈$\mathcal{C}$ and S∈$\mathcal{S}$.
- Inference involves:

1. For each item of evidence, an evidence function is multiplied onto an appropriate clique potential.
2. We select the *root* clique.
3. Messages are passed to the root through the separators (CollectInformation).
4. The messages passes from the root to the leaves of the tree (DistributeInformation).
5. Now, $P(X|\varepsilon)$ can be compute from any clique or separator that contains X.

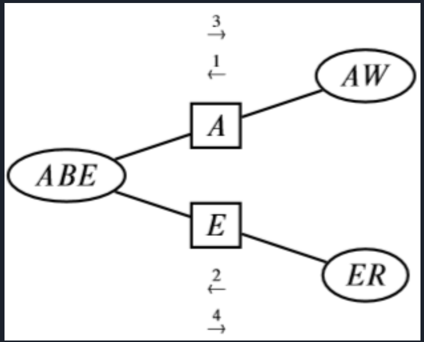
5.1. Probabilistic Inference

5.1.1. Inference in Discrete Bayesian Networks

Junction Trees: HUGIN algorithm

- Propagation of information given two adjacent cliques $\mathcal{C}_i$ and $\mathcal{C}_j$, and a separator S:

1. Calculate the updated separator potential: $$\phi_S^* = \sum_{C_j \setminus S} \phi_{C_j}.$$
2. Update the clique potential of $\mathcal{C}_i$: $$\phi_{C_i} := \phi_{C_i} \frac{\phi_S^*}{\phi_S}.$$
3. Associate the updated potential with the separator. $$\phi_S = \phi_S^*.$$



5.1. Probabilistic Inference

5.1.2. Inference in CLG Bayesian Networks

- Let $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P}, \boldsymbol{\mathcal{F}})$. $\mathcal{X} = \mathcal{X}_\Gamma \cup \mathcal{X}_\Delta$. To solve it is to compute the marginal for all $X \in \mathcal{X}$.
- To calculate $P(X)$ ($X \in \mathcal{X}_\Delta$) we use only the discrete network (continuous v. are barren).
- The prior density of X ($X \in \mathcal{X}_\Gamma$) is a mixture of Gaussian distributions:

$$f(x) = \sum_{i=0}^{n} \alpha_i f_i(x)$$

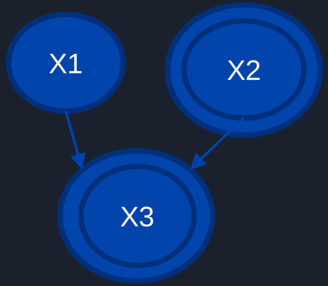
f_i: Gaussian dens. function in X
α_i: prob. of a config. of discrete variables

- Mixing factors of Gaussian dist. are joint probabilities over configurations of discrete variables I ($I \subseteq \mathcal{X}_\Delta$) . Then, the marginal of X is:

$$\mathcal{L}(X) = \sum_{i:P(I=i)>0} P(i) * \mathbb{N}(\mu(i), \sigma^2(i)).$$

5.1. Probabilistic Inference

5.1.2. Inference in CLG Bayesian Networks



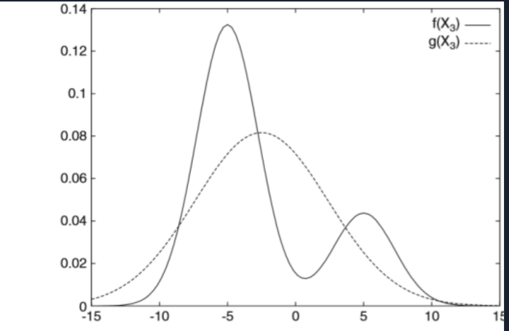


Fig. 5.14 The density function $f(X_3)$ for X_3 and its weak marginal $g(X_3)$

- To compute X_3, we eliminate variables X_1 and X_2

$$\mathcal{L}(X_3) = 0.75 * \mathbb{N}(-5, 5.1) + 0.25 * \mathbb{N}(5, 5.2)$$

mean $\mu = -2.5$ and variance $\sigma^2 = 23.88$

- $\{X_1, X_3\}$ are barren with respect to X_2.
- $\{X_2, X_3\}$ are barren with respect to X_1.
- CLG distributions are not closed under the operation of discrete variable elimination.
- Distribution $\mathbb{N}(\mu, \sigma^2)$ is used an approximation of the true marginal.

5.1. Probabilistic Inference

5.1.2. Inference in CLG Bayesian Networks

- When marginalizing over both continuous and discrete variables, we first marginalize over the continuous variables and then over the discrete variables ($\mathcal{X}_\Delta \prec \mathcal{X}_\Gamma$: strong elimination order).
- If all the ancestors of a variable are continuous, its posterior marginal is a strong marginal.
- When there is evidence, the marginal of a discrete variable is $P(X|\varepsilon)$ and the marginal of a continuous is a density function $f(x|\varepsilon)$.
- The normalization constant α is proportional to the density at the observed values of the continuous variables. The proportionality constant is $P(\varepsilon_\Delta|\varepsilon_\Gamma)$.

5.2. Solving Decision Models

5.2.1. Solving Discrete Influence Diagrams

- $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P}, \mathcal{U})$. Inference: Compute the optimal strategy $\widehat{\Delta}$ and the expected utility.

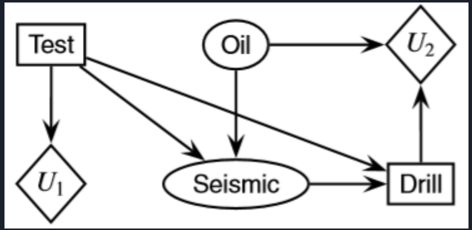
$$\mathrm{EU}(\mathcal{X}) = \prod_{X_v \in \mathcal{X}_C} P(X_v \,|\, X_{\mathrm{pa}(v)}) \sum_{w \in V_U} u(X_{\mathrm{pa}(w)}).$$

- To solve $\mathcal{N}$, we eliminate variables in the reverse order of the information of the precedence order $\prec$ ($\sum - max \sum - rule$). The principle is to average over the unknown random variables, maximize over the decision variable, and finally average over the observed random variables.

$$\mathrm{EU}(\hat{\Delta}) = \sum_{\mathcal{I}_0} \max_{D_1} \sum_{\mathcal{I}_1} \max_{D_2} \cdots \sum_{\mathcal{I}_{n-1}} \max_{D_n} \sum_{\mathcal{I}_n} \mathrm{EU}(\mathcal{X})$$

5.2. Solving Decision Models

5.2.1. Solving Discrete Influence Diagrams



$$\mathrm{EU}(\hat{\Delta}) = \max_T \sum_S \max_D \sum_O P(O)P(S \mid O, T)(U_1(T) + U_2(D, O)).$$

$$\{\} \prec \text{Test} \prec \{\text{Seismic}\} \prec \text{Drill} \prec \{\text{Oil}\}.$$

Solving an influence diagram using this method is INEFFICIENT!

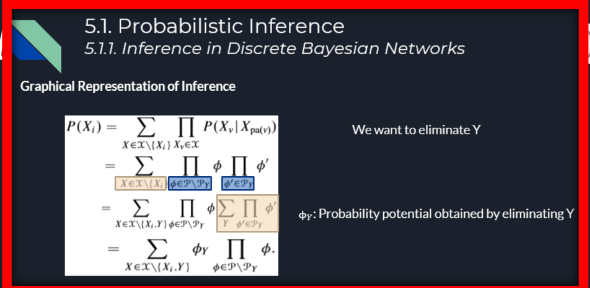
$$\mathop{\mathrm{M}}_X \rho \triangleq \sum_X \rho \quad \text{and} \quad \mathop{\mathrm{M}}_D \rho \triangleq \max_D \rho,$$

5.2. Solving Decision Models

5.2.1. Solving Discrete Influence Diagrams

Generalized Distributive Law

- If we want to eliminate Y, we split the set of utility potentials $\mathcal{U}$ in
- We also split , $\mathcal{P}$ in $\mathcal{P}_Y$ and $\mathcal{P} \backslash \mathcal{P}_Y$. $\mathcal{P}_Y = \{P \in \mathcal{P} \mid Y \in \mathrm{dom}(P)\}$.
- ϕ_Y: Probability potential obtained by eliminating Y from $\mathcal{P}_Y \cup \mathcal{U}_Y$
- ψ_Y: Utility potential obtained by eliminating Y from $\mathcal{P}_Y \cup \mathcal{U}_Y$. .



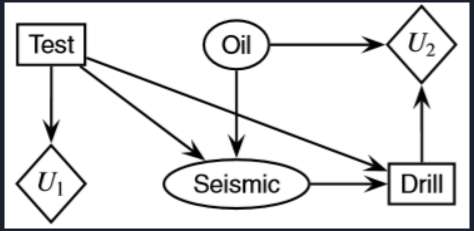
$$\mathrm{EU}(\hat{\Delta}) = \mathop{\mathrm{M}}_{X \in \mathcal{X}} \Big(\prod_{\phi \in \mathcal{P}} \phi \sum_{\psi \in \mathcal{U}} \psi \Big)$$

$$= \mathop{\mathrm{M}}_{X \in \mathcal{X}} \Big[\Big(\prod_{\phi \in \mathcal{P} \backslash \mathcal{P}_Y} \phi \prod_{\phi' \in \mathcal{P}_Y} \phi' \Big) \Big(\sum_{\psi \in \mathcal{U} \backslash \mathcal{U}_Y} \psi + \sum_{\psi' \in \mathcal{U}_Y} \psi' \Big) \Big] \qquad = \mathop{\mathrm{M}}_{X \in \mathcal{X} \backslash \{Y\}} \Big[\Big(\prod_{\phi \in \mathcal{P} \backslash \mathcal{P}_Y} \phi \Big) \Big(\Big(\sum_{\psi \in \mathcal{U} \backslash \mathcal{U}_Y} \psi \Big) \phi_Y + \psi_Y \Big) \Big]$$

$$= \mathop{\mathrm{M}}_{X \in \mathcal{X} \backslash \{Y\}} \Big[\Big(\prod_{\phi \in \mathcal{P} \backslash \mathcal{P}_Y} \phi \Big) \mathop{\mathrm{M}}_{Y} \Big(\prod_{\phi' \in \mathcal{P}_Y} \phi' \Big) \Big(\sum_{\psi \in \mathcal{U} \backslash \mathcal{U}_Y} \psi + \sum_{\psi' \in \mathcal{U}_Y} \psi' \Big) \Big] = \mathop{\mathrm{M}}_{X \in \mathcal{X} \backslash \{Y\}} \Big[\Big(\prod_{\phi \in \mathcal{P} \backslash \mathcal{P}_Y} \phi \Big) \phi_Y \Big(\sum_{\psi \in \mathcal{U} \backslash \mathcal{U}_Y} \psi + \frac{\psi_Y}{\phi_Y} \Big) \Big].$$

5.2. Solving Decision Models

5.2.1. Solving Discrete Influence Diagrams



$$\mathrm{EU}(\hat{\Delta}) = \max_T \sum_S \max_D \sum_O P(O)P(S \mid O, T)(U_1(T) + U_2(D, O)).$$

$\{\} \prec \text{Test} \prec \{\text{Seismic}\} \prec \text{Drill} \prec \{\text{Oil}\}.$

$$\begin{aligned}\mathrm{EU}(\hat{\Delta}) &= \max_T \sum_S \max_D \sum_O P(O)P(S \mid O, T)(C(T) + U(D, O)) \\ &= \max_T (C(T) + \sum_S P(S) \max_D \sum_O \frac{P(O)P(S \mid O, T)}{P(S)} U(D, O)).\end{aligned}$$

5.2. Solving Decision Models

5.2.2. Solving CLQG Influence Diagrams

- $\mathcal{N} = (\mathcal{X}, \mathcal{G}, \mathcal{P}, \boldsymbol{\mathcal{F}}, \mathcal{U})$. Inference: Compute the optimal strategy $\widehat{\Delta}$ and the expected utility.

$$\mathrm{EU}(\mathcal{X}_\Delta = i, \mathcal{X}_\Gamma) = \prod_{v \in V_\Delta} P(i_v \mid i_{\mathrm{pa}(v)}) * \prod_{w \in V_\Gamma} p(y_w \mid X_{\mathrm{pa}(w)}) * \sum_{z \in V_U} u(X_{\mathrm{pa}(z)}).$$

- We apply the $\sum -max \sum -rule$ as we did with discrete influence diagrams.

- We need to eliminate the continuous random variables by integration.

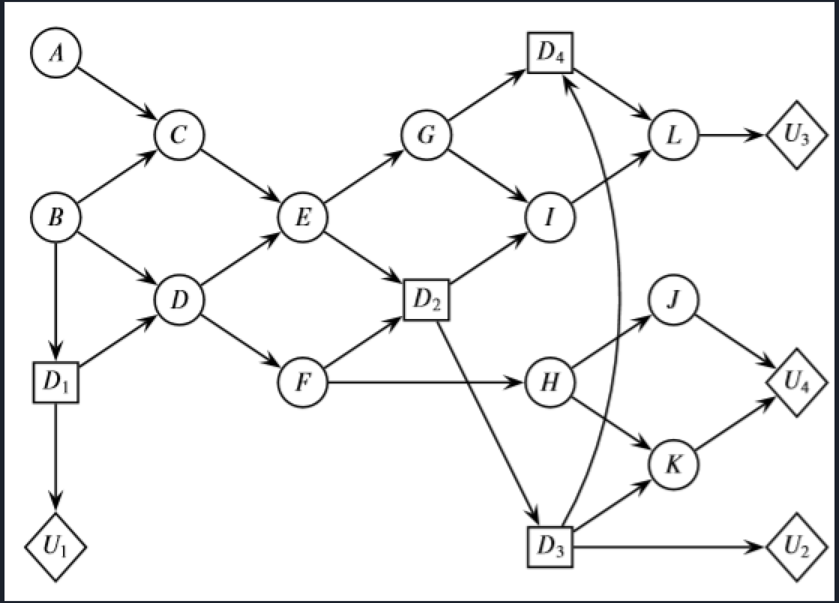
5.3. Relevance Reasoning

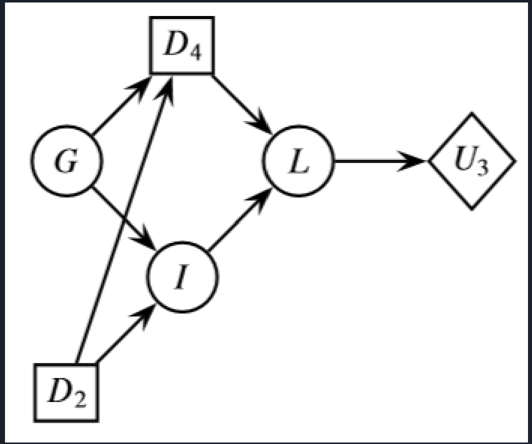
- An observation (or decision) is essential for a decision, if the outcome of the observation may impact the choice of decision option.
- A policy for a decision D could be a function of $RP(D)$ (variables observed prior D whose values have an impact on the choice of decision option for D).

Definition 5.4 (Requisite Observation). Let $\mathcal{N} = (\mathcal{X}, \mathcal{G} = (V, E), \mathcal{P}, \mathcal{U})$ be an influence diagram. The observation on variable $Y_v \in \mathcal{I}(D_i)$ is requisite for decision D_i in $\mathcal{N}$ if and only if $v \not\perp_{\mathcal{G}} V_U \cap \mathrm{de}(v_i) | (V_{\mathcal{I}(D_i)} \setminus \{v\})$, where v_i is the node representing D_i.

Definition 5.5 (Relevant Variable). Let $\mathcal{N} = (\mathcal{X}, \mathcal{G} = (V, E), \mathcal{P}, \mathcal{U})$ be an influence diagram. A variable $Y_v \in \mathcal{F}(D_i)$ is relevant for decision D_i if and only if $v \not\perp_{\mathcal{G}} V_U \cap \mathrm{de}(v_i) | (V_{\mathcal{I}(D_i)} \setminus \{v\})$, where v_i is the node representing D_i.

5.3. Relevance Reasoning





$$\{B\} \prec D_1 \prec \{E, F\} \prec D_2 \prec \{\} \prec D_3 \prec \{G\} \prec D_4 \prec \{A, C, D, H, I, J, K, L\}.$$

5.2. Solving Decision Models

5.2.4. Solving Limited Memory Influence Diagrams

- The LIMID representation relaxes the two fundamental assumptions of the influence diagram representation: Total order on decisions and the perfect recall of past decisions and observations.

- Single Policy Updating (SPU) algorithm: An iterative procedure for identifying (locally) optimal decision policies for the decisions of $\mathcal{N}$.

- Start an iterative process from some initial strategy where the policy at each decision is updated while keeping the remaining policies fixed until convergence.

5.2. Solving Decision Models

5.2.4. Solving Limited Memory Influence Diagrams

Single Policy Updating (SPU)

$$\delta_{D_i} : \mathcal{I}(D_i) \to \mathrm{dom}(D_i)$$

$$\delta_i'(d_i \mid \mathcal{I}(D_i) = j) = \begin{cases} 1 & \text{if } d_i = \delta_{D_i}(j), \\ 0 & \text{otherwise.} \end{cases}$$

- $\mathcal{X}_C: chance, \mathcal{X}_D: decision.$ A strategy: $\Delta = \{\delta_D : D \in \mathcal{X}_D\}$ induces:

$$P_\Delta(\mathcal{X}) = \prod_{X_v \in \mathcal{X}_C} P(X_v \mid X_{\mathrm{pa}(v)}) \prod_{D_i \in \mathcal{X}_D} \delta_i'.$$

$$\mathrm{EU}(\Delta) = \sum_{X \in \mathcal{X}} P_\Delta(\mathcal{X}) U(\mathcal{X}) = \prod_{X_v \in \mathcal{X}_C} P(X_v \mid X_{\mathrm{pa}(v)}) \prod_{D_i \in \mathcal{X}_D} \delta_i' \sum_{u \in \mathcal{U}} u.$$

5.2. Solving Decision Models

5.2.4. Solving Limited Memory Influence Diagrams

Single Policy Updating (SPU)

- Assuming Δ is the current strategy and D_i the next decision to be considered, SPU proceeds like this:

1. Retract: Retract $\delta_i{}'$ from Δ to obtain $\Delta_{-i} = \Delta \backslash \{\delta_i{}'\}$
2. Update: Identify a new policy $\delta_i{}'$: $$\hat{\delta'}_i = \arg\max_{\delta'_i} EU(\Delta_{-i} \cup \{\delta'_i\}).$$
3. Replace: $\Delta = \Delta_{-i} \cup \{\delta_i{}'\}$