

# ROB311 Quiz 2

Hanhee Lee

February 18, 2025

## Contents

<b>1</b>	<b>Bayesian Networks</b>	<b>2</b>
1.1	Junction . . . . .	2
1.1.1	Causal Chain . . . . .	2
1.1.2	Common Cause . . . . .	2
1.1.3	Common Effect . . . . .	3
1.2	Dependence Separation . . . . .	4
1.2.1	Blocked . . . . .	4
1.2.2	Blocked Undirected Path . . . . .	4
1.2.3	Independence . . . . .	4
1.2.4	Consequence of Dependence Separation . . . . .	4
<b>2</b>	<b>Probabilistic Inference</b>	<b>5</b>
2.1	Problem Setup . . . . .	5
2.2	Method 1: Bayesian Network Inference . . . . .	5
2.2.1	Markov Blanket . . . . .	5
2.2.2	Graphical Interpretation . . . . .	5
2.2.3	Elimination Ordering . . . . .	5
2.2.4	Elimination Width . . . . .	5
2.2.5	Heuristics for Elimination Ordering . . . . .	5
2.3	Method 2: Inference via Sampling . . . . .	6
2.3.1	Inference via Sampling with Likelihood Weighting . . . . .	6
2.4	Canonical Problems: . . . . .	7
2.4.1	Path Blocked? . . . . .	8
2.4.2	Independence . . . . .	8
2.4.3	Hypergraph . . . . .	9
2.4.4	Bayesian Inference . . . . .	9
2.4.5	Inference via Sampling . . . . .	10
<b>3</b>	<b>Markov</b>	<b>11</b>
3.1	General . . . . .	11
3.1.1	Random Process . . . . .	11
3.1.2	Markov Process . . . . .	11
3.2	Markov Chains (MCs) . . . . .	11
3.2.1	Bayesian Network . . . . .	11
3.3	Markov Reward Processes (MRPs) . . . . .	12
3.3.1	Bayesian Network . . . . .	12
3.4	Markov Decision Processes (MDPs) . . . . .	13
3.4.1	Setup . . . . .	13
3.4.2	Bayesian Network . . . . .	16
3.4.3	Intuition on Formulae . . . . .	16
3.5	Canonical Examples . . . . .	17
3.5.1	Markov Chains . . . . .	17
3.5.2	Markov Reward Processes . . . . .	17
3.5.3	Markov Decision Processes . . . . .	18

# Probabilistic Inference Problems

## 1 Bayesian Networks

**Definition:** Vertices represent random variables and edges represent dependencies between variables.

### 1.1 Junction

**Definition:** A **junction**  $\mathcal{J}$  consists of three vertices,  $X_1$ ,  $X_2$ , and  $X_3$ , connected by two edges,  $e_1$  and  $e_2$ :



Figure 1

- $X_1$  and  $X_2$  are not independent,  $X_2$  and  $X_3$  are not independent, but when is  $X_1$  and  $X_3$  independent?

#### 1.1.1 Causal Chain

**Definition:** A causal chain is a junction  $\mathcal{J}$  s.t.



Figure 2

- $X_1$  and  $X_3$  are not independent (unconditionally), but are independent given  $X_2$ .

**Notes:**

- **Analogy:** Given  $X_2$ ,  $X_1$  and  $X_3$  are independent. Why?  $X_2$ 's door closes when you know  $X_2$ , so  $X_1$  and  $X_3$  are independent.
- **Distinction b/w Causal and Dependence:**  $X_1$  and  $X_2$  are dependent. However, from a causal perspective,  $X_1$  is influencing  $X_2$  (i.e.  $X_1 \rightarrow X_2$ ).

**Warning:**  $X_1$  is influencing  $X_2$  and  $X_2$  is influencing  $X_3$ .

#### 1.1.2 Common Cause

**Definition:** A common cause is a junction  $\mathcal{J}$  s.t.



Figure 3

- $X_1$  and  $X_3$  are not independent (unconditionally), but are independent given  $X_2$ .

**Notes:**

- **Analogy:** Given  $X_2$ ,  $X_1$  and  $X_3$  are independent. Why? Consider the following example:
  - Let  $X_2$  represent whether a person smokes or not,  $X_1$  represent whether they have yellow teeth,  $X_3$  represent whether they have lung cancer.
- Without knowing  $X_2$ , observing  $X_1$  provides information about  $X_3$  because yellow teeth are associated with smoking, which in turn increases the likelihood of lung cancer.
- If  $X_2$  is known, then knowing whether a person has yellow teeth provides no additional information about whether they have lung cancer beyond what is already known from smoking status.

### 1.1.3 Common Effect

**Definition:** A common effect is a junction  $\mathcal{J}$  s.t.



Figure 4

- $X_1$  and  $X_3$  are independent (unconditionally), but are not independent given  $X_2$  or any of  $X_2$ 's descendants.

**Notes:**

- **Analogy:** Consider the following example:
  - Let  $X_2$  represent whether the grass is wet,  $X_1$  represent whether it rained,  $X_3$  represent whether the sprinkler was on.
- Without knowing whether the grass is wet ( $X_2$ ), the occurrence of rain ( $X_1$ ) and the sprinkler being on ( $X_3$ ) are independent events. The rain may occur regardless of the sprinkler, and vice versa.
- However, once we observe that the grass is wet ( $X_2$ ), the two events become dependent:
  - If we learn that the sprinkler was not on, then the wet grass must have been caused by rain.
  - If we learn that it did not rain, then the wet grass must have been caused by the sprinkler.

## 1.2 Dependence Separation

### 1.2.1 Blocked

**Definition:**  $\mathcal{J} = (\{X_1, X_2, X_3\}, \{e_1, e_2\})$  is **blocked** given  $\mathcal{K} \subseteq \mathcal{V}$  if  $X_1$  and  $X_3$  are independent given  $\mathcal{K}$ .

### 1.2.2 Blocked Undirected Path

**Definition:** An undirected path,

$$p = \langle (X_1, e_1, X_2), \dots, (X_{|p|-1}, e_{|p|-1, |p|}, X_{|p|}) \rangle,$$

is **blocked** given  $\mathcal{K} \subseteq \mathcal{V}$  if any of its junctions,

$$\mathcal{J}^{(n)} = \{(X_{n-1}, X_n, X_{n+1}), (e_{n-1}, e_n)\},$$

is blocked given  $\mathcal{K}$ .

### 1.2.3 Independence

**Theorem:** Any two variables,  $X_1$  and  $X_2$ , in a Bayesian network,  $\mathcal{B} = (\mathcal{V}, \mathcal{E})$ , are independent given  $\mathcal{K} \subseteq \mathcal{V}$  if every undirected path is blocked.

### 1.2.4 Consequence of Dependence Separation

**Theorem:** For any variable,  $X \in \mathcal{V}$ , it can be shown that  $X$  is independent of  $X$ 's non-descendants,  $\mathcal{V} \setminus \text{des}(X)$ , given  $X$ 's parents,  $\text{pts}(X)$ .

Notes:

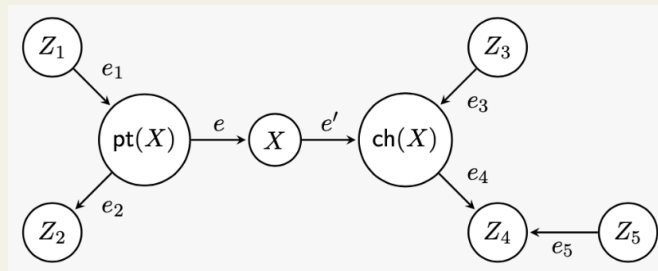


Figure 5

- Intuition

## 2 Probabilistic Inference

### 2.1 Problem Setup

**Definition:** Given a Bayesian network,  $\mathcal{B} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{X_1, \dots, X_{|\mathcal{V}|}\}$ , we want to find the value of:

$$\text{pr}(\mathbf{Q} \mid \mathbf{E}) := \text{pr}(Q_1, \dots, Q_{|\mathbf{Q}|} \mid E_1, \dots, E_{|\mathbf{E}|}) = \frac{\sum_{\mathcal{V} \setminus (\mathbf{Q} \cup \mathbf{E})} p(X_1, \dots, X_{|\mathcal{V}|})}{\sum_{\mathcal{V} \setminus \mathbf{E}} p(X_1, \dots, X_{|\mathcal{V}|})}$$

$$\text{pr}(\mathbf{Q} \mid \mathbf{E}) \propto \sum_{\mathcal{V} \setminus (\mathbf{Q} \cup \mathbf{E})} \left( p(X_1) \prod_{i \neq 1} p(X_i \mid \text{pts}(X_i)) \right)$$

- $\mathbf{Q} = \{Q_1, \dots, Q_{|\mathbf{Q}|}\}$ : Query variables
- $\mathbf{E} = \{E_1, \dots, E_{|\mathbf{E}|}\} \subseteq \mathcal{V}$ : Evidence variables
- $\mathbf{Q} \cap \mathbf{E} = \emptyset$ .

### 2.2 Method 1: Bayesian Network Inference

#### 2.2.1 Markov Blanket

**Definition:** The **Markov blanket** of a variable  $X$ , denoted  $\text{mbk}(X)$ , consists of the following variables:

- $X$ 's children
- $X$ 's parents
- The other parents of  $X$ 's children, excluding  $X$  itself.

which is when a variable,  $X$ , is "eliminated", the resulting factor's scope is the Markov blanket of  $X$ .

#### 2.2.2 Graphical Interpretation

**Definition:** Pictorially, eliminating  $X$  is equivalent to replacing all hyper-edges that include  $X$  with their union minus  $X$ , and then removing  $X$ .

#### 2.2.3 Elimination Ordering

**Definition:** The order that the variables are eliminated.

#### 2.2.4 Elimination Width

**Definition:** The **elimination width** of a sequence of hyper-graphs is the # of variables in the hyper-edge within the sequence with the most variables.

#### 2.2.5 Heuristics for Elimination Ordering

**Definition:** Choose the elimination ordering to minimize the elimination width using the following heuristics:

1. Eliminate variable with the fewest parents.
2. Eliminate variable with the smallest domain for its parents, where

$$|\text{dom}(\text{pts}(X))| = \prod_{Z \in \text{pnt}(X)} |\text{dom}(Z)|.$$

3. Eliminate variable with the smallest Markov blanket.
4. Eliminate variable with the smallest domain for its Markov blanket, where

$$|\text{dom}(\text{mbk}(X))| = \prod_{Z \in \text{embk}(X)} |\text{dom}(Z)|.$$

## 2.3 Method 2: Inference via Sampling

**Definition:** Generate a large # of samples and then approximate as:

$$p(\mathbf{Q} \mid \mathbf{E}) \approx \frac{\# \text{ of samples w/ } \mathbf{Q} \text{ and } \mathbf{E}}{\# \text{ of samples w/ } \mathbf{E}}.$$

- As # of samples  $\rightarrow \infty$ , the approximation becomes exact.

### 2.3.1 Inference via Sampling with Likelihood Weighting

**Motivation:** Most of the samples are wasted since they are not consistent with the evidence.

**Definition:** Generate a large # of samples and then approximate as:

$$p(\mathbf{Q} \mid \mathbf{E}) \approx \frac{\text{weight of samples w/ } \mathbf{Q} \text{ and } \mathbf{E}}{\text{weight of samples w/ } \mathbf{E}}.$$

- Weight for each sample: Probability of forcing the evidence, i.e. probability of the evidence given the sample.

## 2.4 Canonical Problems:

### Example:

1. **Given:** Caveman is deciding whether to go hunt for meat. He must take into account several factors:
  - Weather
  - Possibility of over-exertion
  - Possibility encountering lion

These factors can result in Cavemen's death. His decision will ultimately depend on the **chances** of his death.

2. **Binary Variables:**

- $W = \{\text{Sun, Rainy}\}$ : Weather
- $H$ : Whether the Cavemen goes hunting or not.
- $L$ : Whether the Cavemen encounters a lion or not.
- $T$ : Whether the Cavement is tired or not.
- $D$ : Whether the Cavemen dies or not

3. **Problem:** Cavemen must decide whether to go hunting or not.

- He must consider the conditional probabilities (i.e. dependence) of each event.

**Warning:** Have to be discrete.

### 2.4.1 Path Blocked?

**Process:**

- Know when a path is blocked. More than one path b/w 2 variables, then all paths need to be blocked.

**Example:**

### 2.4.2 Independence

**Process:**

- 1.

**Example:**

1. **Given:** Bayesian network.

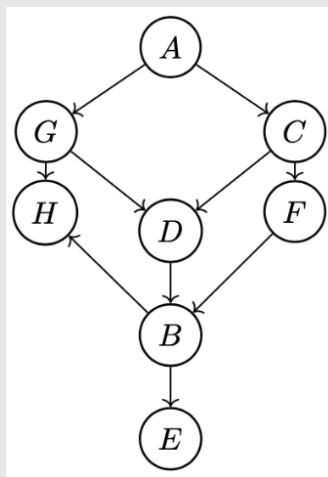


Figure 6

2. **Problem:**  $A$  and  $E$  are
  - independent if  $\mathcal{K} =$
  - not necessarily independent for  $\mathcal{K} =$



### 2.4.3 Hypergraph

#### Process:

1.

### 2.4.4 Bayesian Inference

#### Process:

1.

#### Example:

1. Given:

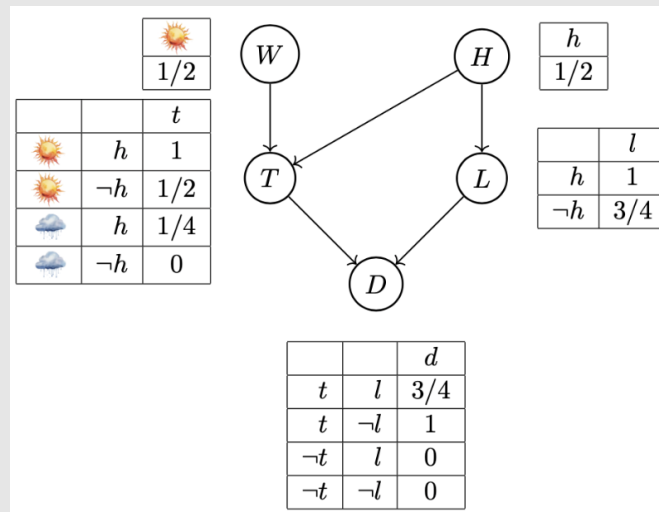


Figure 7

2. Problem:

### 2.4.5 Inference via Sampling

**Process:**

- 1.

**Example:**

1. **Given:**
2. **Problem:**

### 3 Markov

#### 3.1 General

##### 3.1.1 Random Process

**Definition:** Time-varying random variables  $S_0, S_1, S_2, \dots$

##### 3.1.2 Markov Process

**Definition:** Random process + depends on previous time step only (memoryless)

- w.l.o.g. states can contain history of previous states.

#### 3.2 Markov Chains (MCs)

**Summary:** In a **Markov Chain**, we assume that:

- there are no agents
- state transitions occur automatically
- $S_t$  is the state *after* transition  $t$
- the state transition process is stochastic and memoryless:

$$S_t \perp S_0, \dots, S_{t-2} \mid S_{t-1}$$

- $S_t$  is independent of all previous states given  $S_{t-1}$

Name	Function:
initial state distribution	$p_0(s) := \mathbb{P}[S_0 = s]$
transition distribution	$p(s' s) := \mathbb{P}[S_{t+1} = s'   S_t = s]$
Prob. that state of the env. after $T$ transitions is $s$	$p_T(s) := \mathbb{P}[S_T = s]$ $= \sum_{s'} p_{T-1}(s') p(s s')$
<ul style="list-style-type: none"> <li>• <math>p_{T-1}(s')</math>: Prob. <math>s'</math> at <math>T-1</math> (given) <ul style="list-style-type: none"> <li>– <math>p_0(s)</math>: Base case</li> </ul> </li> <li>• <math>p(s s')</math>: Prob. <math>s</math> given <math>s'</math> (from graph)</li> </ul>	

##### 3.2.1 Bayesian Network

**Notes:**  $S_0, S_1, S_2, \dots$  form a **Bayesian Network**:

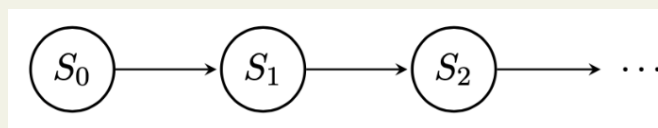


Figure 8

### 3.3 Markov Reward Processes (MRPs)

**Summary:** In a **Markov Reward Process**, we assume that:

- there is one agent
- state transitions occur automatically (i.e. agent has no control over actions)
- $S_t$  is the state *after* transition  $t$
- the state transition process is stochastic and memoryless:

$$S_t \perp S_0, \dots, S_{t-2} \mid S_{t-1}$$

- $S_t$  is independent of all previous states given  $S_{t-1}$
- $R_t$  is the reward for transition  $t$ , i.e.,  $(S_{t-1}, \emptyset, S_t)$

Name	Function:
Initial state distribution	$p_0(s) := \mathbb{P}[S_0 = s]$
Transition distribution	$p(s' s) := \mathbb{P}[S_{t+1} = s'   S_t = s]$
Reward function	$r(s, s') := \text{reward for transition } (s, \emptyset, s')$
Discount factor	$\gamma \in [0, 1]$
Return after $T$ transitions	$U_T = \sum_{t=1}^T \gamma^{t-1} R_t$ $= U_{T-1} + \gamma^{T-1} R_T$ <ul style="list-style-type: none"> <li>• i.e. The (possibly discounted) sum of the rewards after <math>T</math> transitions (sequence of rewards)</li> <li>• <b>Why?</b> <ul style="list-style-type: none"> <li>– Future rewards are less valuable than immediate rewards.</li> <li>– Won't converge if sum goes to <math>\infty</math> if <math>\gamma = 1</math>.</li> </ul> </li> </ul>
Expected return after $T$ transitions	$\mathbb{E}[U_T] = \mathbb{E}[U_{T-1}] + \gamma^{T-1} \mathbb{E}[R_T]$ $= \mathbb{E}[U_{T-1}] + \gamma^{T-1} \sum_{s, s'} p_{T-1}(s) p(s' s) r(s, s')$ <ul style="list-style-type: none"> <li>• <math>p_{T-1}(s)p(s' s)</math>: Prob. <math>s \rightarrow s'</math></li> <li>• <math>r(s, s')</math>: rwd <math>s \rightarrow s'</math></li> <li>• <math>\mathbb{E}[U_0] := 0</math>: Base case</li> </ul>

#### 3.3.1 Bayesian Network

**Notes:**  $S_0, R_1, S_1, R_2, S_2, \dots$  form a **Bayesian Network**:

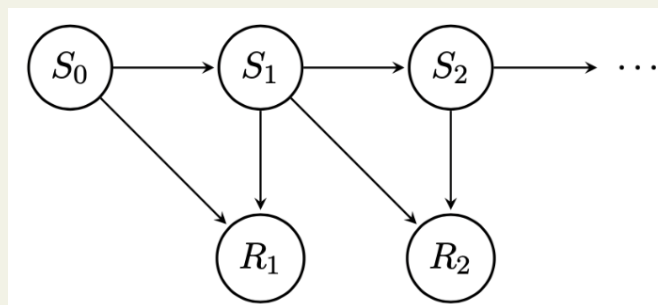


Figure 9

## 3.4 Markov Decision Processes (MDPs)

### 3.4.1 Setup

**Summary:** In a **Markov Decision Process (MDP)**, we assume that:

- there is one agent
- state transitions occur manually (after each action)
- $S_t$  is the state *after* transition  $t$
- $A_t$  is the action inducing transition  $t$
- the state transition process is stochastic and memoryless:

$$S_t \perp S_0, A_1, \dots, S_{t-2}, A_{t-1} \mid S_{t-1}, A_t$$

- $S_t$  is independent of all previous states and actions given  $S_{t-1}$  and  $A_t$
- $R_t$  is the reward for transition  $t$ , i.e.,  $(S_{t-1}, A_t, S_t)$

**Summary:**

Name	Function:
initial state distribution	$p_0(s) := \mathbb{P}[S_0 = s]$
transition distribution	$p(s' s, a) := \mathbb{P}[S_t = s'   A_t = a, S_{t-1} = s]$
reward function	$r(s, a, s') := \text{reward for transition } (s, a, s')$
a time-invariant policy for choosing actions	$\pi(a s) := \mathbb{P}[A_t = a   S_t = s]$
Maximum number of transitions	$T_{\max}$
<ul style="list-style-type: none"> <li>A Markov Decision Process can be either: <ul style="list-style-type: none"> <li><b>Finite:</b> <math>T_{\max}</math> is finite</li> <li><b>Infinite:</b> <math>T_{\max}</math> is infinite</li> </ul> </li> <li>* For infinite MDPs, we must have <math>\gamma &lt; 1</math>.</li> </ul>	
Prob. that state of the env. after $T$ transitions is $s$	$p_T(s) = \sum_{a, s'} p_{T-1}(s) \pi(a s') p(s s', a)$ <ul style="list-style-type: none"> <li><math>p_{T-1}(s)</math>: Prob. <math>s'</math> at <math>T-1</math></li> <li><math>\pi(a s')</math>: Action <math>a</math> from <math>s'</math></li> <li><math>p(s s', a)</math>: Prob. <math>s</math> given <math>s', a</math></li> </ul>
Expected return after $T$ transitions	$\mathbb{E}_\pi[U_T] = \mathbb{E}_\pi[U_{T-1}] + \gamma^{T-1} \mathbb{E}_\pi[R_T]$ <ul style="list-style-type: none"> <li><math>\mathbb{E}_\pi[R_t] = \sum_{s, a, s'} p_{T-1}(s) \pi(a s) p(s' s, a) r(s, a, s')</math></li> <li><math>\mathbb{E}_\pi[U_0] = 0</math>: Base case.</li> </ul>
Future return after $\tau$ transitions	$G_\tau = \sum_{t=\tau+1}^T \gamma^{t-(\tau+1)} R_t$ $= R_{\tau+1} + \gamma G_{\tau+1}$ <ul style="list-style-type: none"> <li>Starting at <math>\tau + 1</math> for the future return.</li> </ul>
Expected future return after $\tau$ transitions given $S_\tau = s$	$\mathbb{E}_\pi[G_\tau   S_\tau = s] = \mathbb{E}_\pi[R_{\tau+1}   S_\tau = s] + \gamma \mathbb{E}_\pi[G_{\tau+1}   S_\tau = s]$ $= \sum_{a, s'} \pi(a s) p(s' s, a) (r(s, a, s') + \gamma \mathbb{E}_\pi[G_{\tau+1}   S_{\tau+1} = s'])$ <ul style="list-style-type: none"> <li><math>\mathbb{E}_\pi[G_{T_{\max}}   S_{T_{\max}} = s] = 0</math>: Base case.</li> </ul>

**Summary:**

Name	Function:
Value function	$v_\pi(s, T) := \mathbb{E}_\pi[G_{T_{\max}-T} \mid S_{T_{\max}-T} = s]$ $= \sum_{a, s'} \pi(a \mid s) p(s' \mid s, a) (r(s, a, s') + \gamma v_\pi(s', T-1))$ <ul style="list-style-type: none"> <li>Value of state <math>s</math> under the policy <math>\pi</math> with <math>T</math> transitions remaining. <ul style="list-style-type: none"> <li>i.e. How good the state is at time <math>T</math> (e.g. If <math>v(s, T) = 5</math>, then the expected future return at <math>T</math> is 5).</li> </ul> </li> <li><math>v(s, 0) = 0</math> for all <math>s</math>: Base case</li> </ul>
Optimal action	$a^*(s, T) = \arg \max_{a \in \mathcal{A}(s)} \sum_{s'} p(s' \mid s, a) (r(s, a, s') + \gamma v_{\pi^*}(s', T-1))$ $= \arg \max_{a \in \mathcal{A}(s)} q^*(s, a, T)$
Optimal policy	$\pi^*(a \mid s, T) = \arg \max_{\pi(a \mid s, T)} \mathbb{E}_\pi[G_\tau \mid S_\tau = s] = \begin{cases} 1 & \text{if } a = a^*(s, T) \\ 0 & \text{otherwise} \end{cases}$ <ul style="list-style-type: none"> <li>Choose <math>\pi(\cdot \mid s)</math> to maximize the expected future return after <math>T</math> transitions given <math>S_\tau = s</math>.</li> <li><b>Note:</b> Policy always depends on transitions remaining so may omit.</li> </ul>
Optimal value function	$v^*(s, T) = \max_a \sum_{s'} p(s' \mid a, s) (r(s, a, s') + \gamma v^*(s', \tau+1))$ <ul style="list-style-type: none"> <li>Assume we use an optimal policy <math>\pi^*</math>.</li> <li><math>v^*(s, 0) = 0</math> for all <math>s</math>: Base case.</li> </ul>
Q function (quality)	$q_\pi(s, a, T) := \mathbb{E}_\pi[G_{T_{\max}-T} \mid S_{T_{\max}-T} = s, A_{T_{\max}-(T-1)} = a]$ $= \sum_{s'} p(s' \mid s, a) \left( r(s, a, s') + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s', a', T-1) \right)$ <ul style="list-style-type: none"> <li>Quality of move <math>(s, a)</math> under policy <math>\pi</math> with <math>T</math> transitions remaining.</li> <li><math>q_\pi(s, a, 0) = 0</math> for all <math>s, a</math>: Base case.</li> </ul>
Optimal Q function	$q^*(s, a, T) = \sum_{s'} p(s' \mid s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T-1) \right)$ <ul style="list-style-type: none"> <li><math>q^*(s, a, 0) = 0</math> for all <math>s, a</math>: Base case.</li> </ul>
IDK Expected Return	$\mathbb{E}_\pi[U_{T_{\max}}] = \sum_s \mathbb{E}_\pi[G_0 \mid S_0 = s] p_0(s)$ $= \sum_s v_\pi(s, 0) p_0(s)$ <ul style="list-style-type: none"> <li><math>G_0 = U_{T_{\max}}</math></li> </ul>
IDK Optimal Expected Return	$\max_\pi \mathbb{E}[U_{T_{\max}}] = \sum_s v^*(s, 0) p_0(s)$

### 3.4.2 Bayesian Network

Notes:  $S_0, A_1, R_1, S_1, A_2, R_2, S_2, \dots$  form a **Bayesian Network**:

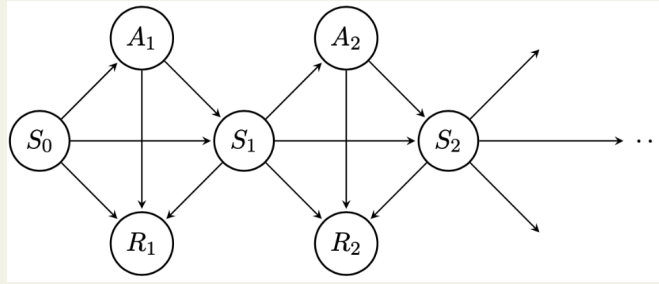


Figure 10

### 3.4.3 Intuition on Formulae

Notes:

$$\mathbb{E}_\pi[R_{\tau+1} \mid S_\tau = s] = \sum_{a, s'} \pi(a \mid s) p(s' \mid a, s) r(s, a, s')$$

- $\pi(a \mid s) p(s' \mid a, s)$ : Prob. of getting to  $s'$  from  $s$  w/ action  $a$
- $r(s, a, s')$ : Reward of getting to  $s'$  from  $s$  w/ action  $a$

$$\mathbb{E}_\pi[G_{\tau+1} \mid S_\tau = s] = \sum_{a, s'} \pi(a \mid s) p(s' \mid a, s) \mathbb{E}_\pi[G_{\tau+1} \mid S_{\tau+1} = s']$$

- $\pi(a \mid s) p(s' \mid a, s)$ : Prob. of getting to  $s'$  from  $s$  w/ action  $a$
- $\mathbb{E}_\pi[G_{\tau+1} \mid S_{\tau+1} = s']$ : Expected future return at  $\tau + 1$  from  $s'$  at  $\tau + 1$ .
- $\sum_{a, s'}$ : Sum over all possible future states and current actions to get expected future return at  $\tau + 1$  from  $s$  at  $\tau$ .



### 3.5 Canonical Examples

#### 3.5.1 Markov Chains

**Example:**

1. **Given:** Caveman needs to predict the weather,  $W$ , which is either sunny or rainy. Suppose the weather tomorrow depends on the weather today:

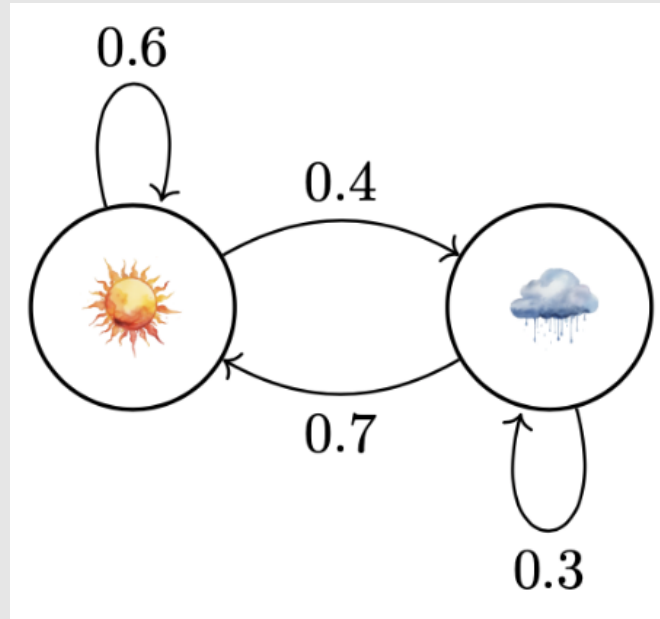


Figure 11

2. **Problem:** Caveman wants to predict the weather on a given day.

#### 3.5.2 Markov Reward Processes

**Example:**

1. **Given:** Caveman needs to predict the weather,  $W$ , which is either sunny or rainy. Suppose the weather tomorrow depends on the weather today:

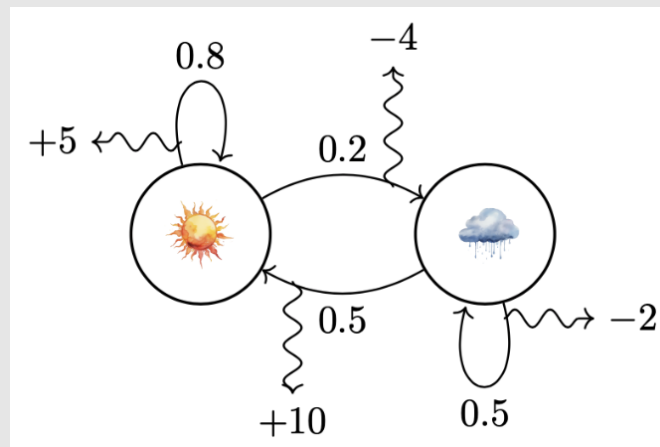


Figure 12

- Depending on the transition, caveman may feel happier/sadder. This is quantified w/ the rewards.
2. **Problem:** Caveman wants to predict the weather on a given day that maximizes his happiness.

## 3.5.3 Markov Decision Processes

**Example:**

1. **Given:**

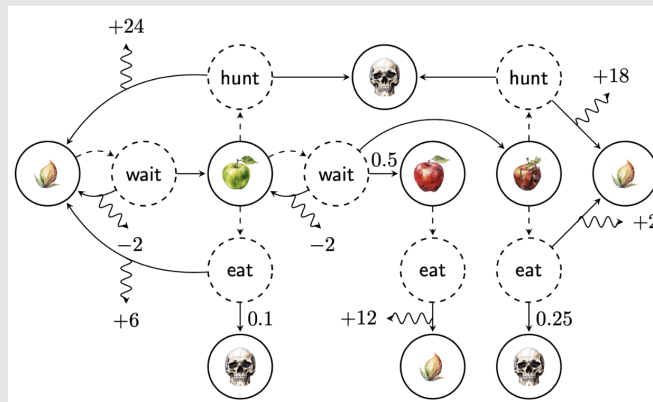


Figure 13

- Solid straight line: Outcome of action  $a$  from state  $s$ .
- Dotted straight line: Choice of action (policy) from state  $s$ .
  - If policy known, then reduced to MRP.
- Squiggly line: Reward for action  $a$  from state  $s$  to state  $s'$ .
- Assume uniform probability.
  - Since  $\sum p = 1$ , therefore count # of arrows going out of  $s$  and divide by 1 to get  $p$ .
- Same states have the same connections (i.e. all can use them just to hard to draw)

2. **Problem:** Find the optimal policy for  $\gamma = 1$  and  $T_{\max} = 5$ .

3. **Soln:**

**Example:**

$T$	$s$	$a$	$q^*(s, a, T) = \sum_{s'} p(s'   s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T - 1) \right)$
0	-	-	0
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(s, 0) = \text{NA}</math></li> </ul>			
1	seed	wait	$q^*(\text{seed}, \text{wait}, 1) = \underbrace{0.5(-2 + 0)}_{s' = \text{seed}} + \underbrace{0.5(0 + 0)}_{s' = \text{ga}} = -1$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{seed}, 1) = \text{wait}</math></li> </ul>			
1	ga	wait	$q^*(\text{ga}, \text{wait}, 1) = \underbrace{0.25(-2 + 0)}_{s' = \text{ga}} + \underbrace{0.5(0 + 0)}_{s' = \text{rea}} + \underbrace{0.25(0 + 0)}_{s' = \text{roa}} = -0.5$
1	ga	eat	$q^*(\text{ga}, \text{eat}, 1) = \underbrace{0.1(0 + 0)}_{s' = \text{dead}} + \underbrace{0.9(6 + 0)}_{s' = \text{seed}} = 5.4$
1	ga	hunt	$q^*(\text{ga}, \text{hunt}, 1) = \underbrace{0.5(24 + 0)}_{s' = \text{seed}} + \underbrace{0.5(0 + 0)}_{s' = \text{dead}} = 12$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{ga}, 1) = \text{hunt}</math></li> </ul>			
1	rea	eat	$q^*(\text{rea}, \text{eat}, 1) = \underbrace{1(12 + 0)}_{s' = \text{seed}} = 12$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{rea}, 1) = \text{eat}</math></li> </ul>			
1	roa	eat	$q^*(\text{roa}, \text{eat}, 1) = \underbrace{0.25(0 + 0)}_{s' = \text{dead}} + \underbrace{0.75(2 + 0)}_{s' = \text{seed}} = 1.5$
1	roa	hunt	$q^*(\text{roa}, \text{hunt}, 1) = \underbrace{0.5(0 + 0)}_{s' = \text{dead}} + \underbrace{0.5(18 + 0)}_{s' = \text{seed}} = 9$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{roa}, 1) = \text{hunt}</math></li> </ul>			
1	dead	-	$q^*(\text{dead}, -, 1) = \underbrace{1(0 + 0)}_{s' = \text{end}} = 0$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(s, 1) = -</math></li> </ul>			
<ul style="list-style-type: none"> <li>Optimal Policy w/ 1 Transition Remaining: <math>\pi^*(a   s, 1) = \begin{cases} 1 &amp; \text{if } a = a^*(s, 1) \\ 0 &amp; \text{otherwise} \end{cases}</math></li> </ul>			

**Example:**

$T$	$s$	$a$	$q^*(s, a, T) = \sum_{s'} p(s'   s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T - 1) \right)$
2	seed	wait	$q^*(\text{seed}, \text{wait}, 2) = \underbrace{0.5(-2 - 1)}_{s'=\text{seed}} + \underbrace{0.5(0 + 12)}_{s'=\text{ga}} = 4.5$
• Best Action: $a^*(\text{seed}, 2) = \text{wait}$			
2	ga	wait	$q^*(\text{ga}, \text{wait}, 2) = \underbrace{0.25(-2 + 12)}_{s'=\text{ga}} + \underbrace{0.5(0 + 12)}_{s'=\text{rea}} + \underbrace{0.25(0 + 9)}_{s'=\text{roa}} = 10.75$
2	ga	eat	$q^*(\text{ga}, \text{eat}, 2) = \underbrace{0.1(0 + 0)}_{s'=\text{dead}} + \underbrace{0.9(6 - 1)}_{s'=\text{seed}} = 4.5$
2	ga	hunt	$q^*(\text{ga}, \text{hunt}, 2) = \underbrace{0.5(24 - 1)}_{s'=\text{seed}} + \underbrace{0.5(0 + 0)}_{s'=\text{dead}} = 11.5$
• Best Action: $a^*(\text{ga}, 2) = \text{hunt}$			
2	rea	eat	$q^*(\text{rea}, \text{eat}, 2) = \underbrace{1(12 - 1)}_{s'=\text{seed}} = 11$
• Best Action: $a^*(\text{rea}, 2) = \text{eat}$			
2	roa	eat	$q^*(\text{roa}, \text{eat}, 2) = \underbrace{0.25(0 + 0)}_{s'=\text{dead}} + \underbrace{0.75(2 - 1)}_{s'=\text{seed}} = 0.5$
2	roa	hunt	$q^*(\text{roa}, \text{hunt}, 2) = \underbrace{0.5(0 + 0)}_{s'=\text{dead}} + \underbrace{0.5(18 - 1)}_{s'=\text{seed}} = 8.5$
• Best Action: $a^*(\text{roa}, 2) = \text{hunt}$			
2	dead	-	$q^*(\text{dead}, -, 2) = \underbrace{1(0 + 0)}_{s'=\text{end}} = 0$
• Best Action: $a^*(s, 2) = -$			
• Optimal Policy w/ 2 Transitions Remaining: $\pi^*(a   s, 2) = \begin{cases} 1 & \text{if } a = a^*(s, 2) \\ 0 & \text{otherwise} \end{cases}$			

**Example:**

$T$	$s$	$a$	$q^*(s, a, T) = \sum_{s'} p(s'   s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T - 1) \right)$
3	seed	wait	$q^*(\text{seed}, \text{wait}, 3) = \underbrace{0.5(-2 + 4.5)}_{s'=\text{seed}} + \underbrace{0.5(0 + 11.5)}_{s'=\text{ga}} = 7$
• Best Action: $a^*(\text{seed}, 3) = \text{wait}$			
3	ga	wait	$q^*(\text{ga}, \text{wait}, 3) = \underbrace{0.25(-2 + 11.5)}_{s'=\text{ga}} + \underbrace{0.5(0 + 11)}_{s'=\text{rea}} + \underbrace{0.25(0 + 8.5)}_{s'=\text{roa}} = 10$
3	ga	eat	$q^*(\text{ga}, \text{eat}, 3) = \underbrace{0.1(0 + 0)}_{s'=\text{dead}} + \underbrace{0.9(6 + 4.5)}_{s'=\text{seed}} = 9.45$
3	ga	hunt	$q^*(\text{ga}, \text{hunt}, 3) = \underbrace{0.5(24 + 4.5)}_{s'=\text{seed}} + \underbrace{0.5(0 + 0)}_{s'=\text{dead}} = 14.25$
• Best Action: $a^*(\text{ga}, 3) = \text{hunt}$			
3	rea	eat	$q^*(\text{rea}, \text{eat}, 3) = \underbrace{1(12 + 4.5)}_{s'=\text{seed}} = 16.5$
• Best Action: $a^*(\text{rea}, 3) = \text{eat}$			
3	roa	eat	$q^*(\text{roa}, \text{eat}, 3) = \underbrace{0.25(0 + 0)}_{s'=\text{dead}} + \underbrace{0.75(2 + 4.5)}_{s'=\text{seed}} = 4.875$
3	roa	hunt	$q^*(\text{roa}, \text{hunt}, 3) = \underbrace{0.5(0 + 0)}_{s'=\text{dead}} + \underbrace{0.5(18 + 4.5)}_{s'=\text{seed}} = 11.25$
• Best Action: $a^*(\text{roa}, 3) = \text{hunt}$			
3	dead	-	$q^*(\text{dead}, -, 3) = \underbrace{1(0 + 0)}_{s'=\text{end}} = 0$
• Best Action: $a^*(s, 3) = -$			
• Optimal Policy w/ 3 Transitions Remaining: $\pi^*(a   s, 3) = \begin{cases} 1 & \text{if } a = a^*(s, 3) \\ 0 & \text{otherwise} \end{cases}$			

**Example:**

$T$	$s$	$a$	$q^*(s, a, T) = \sum_{s'} p(s'   s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T - 1) \right)$
4	seed	wait	$q^*(\text{seed}, \text{wait}, 4) = \underbrace{0.5(-2 + 7)}_{s'=\text{seed}} + \underbrace{0.5(0 + 14.25)}_{s'=\text{ga}} = 9.625$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{seed}, 4) = \text{wait}</math></li> </ul>			
4	ga	wait	$q^*(\text{ga}, \text{wait}, 4) = \underbrace{0.25(-2 + 14.25)}_{s'=\text{ga}} + \underbrace{0.5(0 + 16.5)}_{s'=\text{rea}} + \underbrace{0.25(0 + 11.25)}_{s'=\text{roa}} = 14.125$
4	ga	eat	$q^*(\text{ga}, \text{eat}, 4) = \underbrace{0.1(0 + 0)}_{s'=\text{dead}} + \underbrace{0.9(6 + 7)}_{s'=\text{seed}} = 11.7$
4	ga	hunt	$q^*(\text{ga}, \text{hunt}, 4) = \underbrace{0.5(24 + 7)}_{s'=\text{seed}} + \underbrace{0.5(0 + 0)}_{s'=\text{dead}} = 15.5$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{ga}, 4) = \text{hunt}</math></li> </ul>			
4	rea	eat	$q^*(\text{rea}, \text{eat}, 4) = \underbrace{1(12 + 7)}_{s'=\text{seed}} = 19$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{rea}, 4) = \text{eat}</math></li> </ul>			
4	roa	eat	$q^*(\text{roa}, \text{eat}, 4) = \underbrace{0.25(0 + 0)}_{s'=\text{dead}} + \underbrace{0.75(2 + 7)}_{s'=\text{seed}} = 6.75$
4	roa	hunt	$q^*(\text{roa}, \text{hunt}, 4) = \underbrace{0.5(0 + 0)}_{s'=\text{dead}} + \underbrace{0.5(18 + 7)}_{s'=\text{seed}} = 12.5$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(\text{roa}, 4) = \text{hunt}</math></li> </ul>			
4	dead	-	$q^*(\text{dead}, -, 4) = \underbrace{1(0 + 0)}_{s'=\text{end}} = 0$
<ul style="list-style-type: none"> <li>Best Action: <math>a^*(s, 4) = -</math></li> </ul>			
<ul style="list-style-type: none"> <li>Optimal Policy w/ 4 Transitions Remaining: <math>\pi^*(a   s, 4) = \begin{cases} 1 &amp; \text{if } a = a^*(s, 4) \\ 0 &amp; \text{otherwise} \end{cases}</math></li> </ul>			

**Example:**

$T$	$s$	$a$	$q^*(s, a, T) = \sum_{s'} p(s'   s, a) \left( r(s, a, s') + \gamma \max_{a'} q^*(s', a', T-1) \right)$
5	seed	wait	$q^*(\text{seed}, \text{wait}, 5) = \underbrace{0.5(-2 + 9.625)}_{s'=\text{seed}} + \underbrace{0.5(0 + 15.5)}_{s'=\text{ga}} = 11.5625$
• Best Action: $a^*(\text{seed}, 5) = \text{wait}$			
5	ga	wait	$q^*(\text{ga}, \text{wait}, 5) = \underbrace{0.25(-2 + 15.5)}_{s'=\text{ga}} + \underbrace{0.5(0 + 19)}_{s'=\text{rea}} + \underbrace{0.25(0 + 12.5)}_{s'=\text{roa}} = 16$
5	ga	eat	$q^*(\text{ga}, \text{eat}, 5) = \underbrace{0.1(0 + 0)}_{s'=\text{dead}} + \underbrace{0.9(6 + 9.625)}_{s'=\text{seed}} = 14.0625$
5	ga	hunt	$q^*(\text{ga}, \text{hunt}, 5) = \underbrace{0.5(24 + 9.625)}_{s'=\text{seed}} + \underbrace{0.5(0 + 0)}_{s'=\text{dead}} = 16.8125$
• Best Action: $a^*(\text{ga}, 5) = \text{hunt}$			
5	rea	eat	$q^*(\text{rea}, \text{eat}, 5) = \underbrace{1(12 + 9.625)}_{s'=\text{seed}} = 21.625$
• Best Action: $a^*(\text{rea}, 5) = \text{eat}$			
5	roa	eat	$q^*(\text{roa}, \text{eat}, 5) = \underbrace{0.25(0 + 0)}_{s'=\text{dead}} + \underbrace{0.75(2 + 9.625)}_{s'=\text{seed}} = 8.71875$
5	roa	hunt	$q^*(\text{roa}, \text{hunt}, 5) = \underbrace{0.5(0 + 0)}_{s'=\text{dead}} + \underbrace{0.5(18 + 9.625)}_{s'=\text{seed}} = 13.8125$
• Best Action: $a^*(\text{roa}, 5) = \text{hunt}$			
5	dead	-	$q^*(\text{dead}, -, 5) = \underbrace{1(0 + 0)}_{s'=\text{end}} = 0$
• Best Action: $a^*(s, 5) = -$			
• Optimal Policy w/ 5 Transitions Remaining: $\pi^*(a   s, 5) = \begin{cases} 1 & \text{if } a = a^*(s, 5) \\ 0 & \text{otherwise} \end{cases}$			