ROB311 Quiz 2

Hanhee Lee

February 17, 2025

Contents

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:

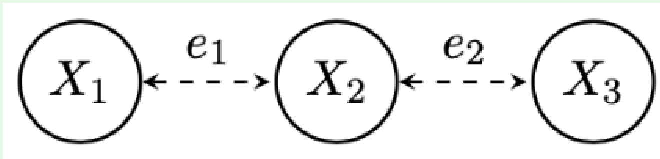

Figure 1

- 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.

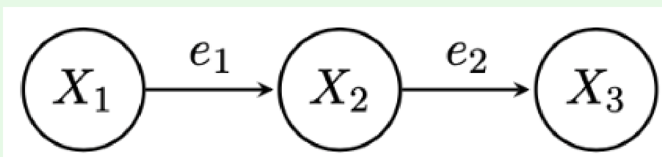

Figure 2

- 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 influeincing 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.

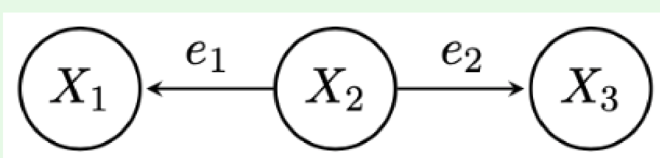

Figure 3

- 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.

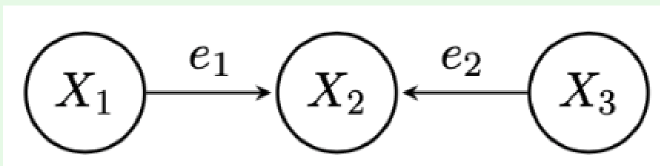

Figure 4

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

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,

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

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, $\mathcal{V} \setminus \mathrm{des}(X)$, given X's parents, $\mathrm{pts}(X)$.

Notes:

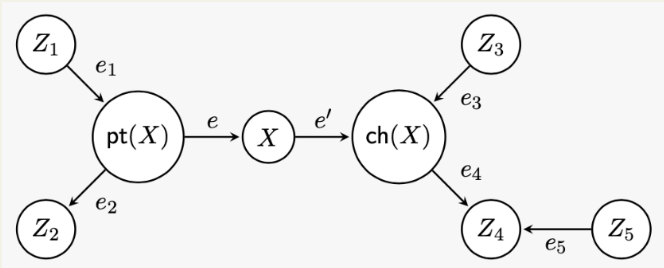

Figure 5

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:

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

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

- $\mathbf{Q} = \{Q_1, \dots, Q_{|\mathbf{Q}|}\}$: 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 $\mathrm{mbk}(X)$, 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

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

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

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

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 $\to \infty$, 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:**
   - $W = \{\text{Sun}, \text{Rainy}\}$: 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 varaibles, then all paths need to be blocked.

Example:

2.4.2 Independence

Process:
1.

Example:
1. **Given:** Bayesian network.

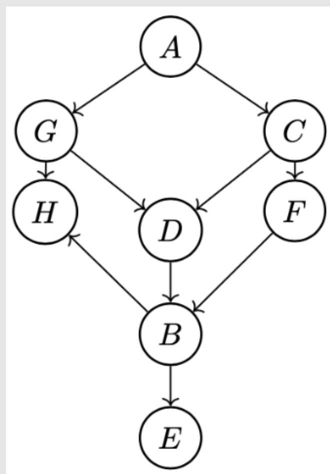

Figure 6

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:**

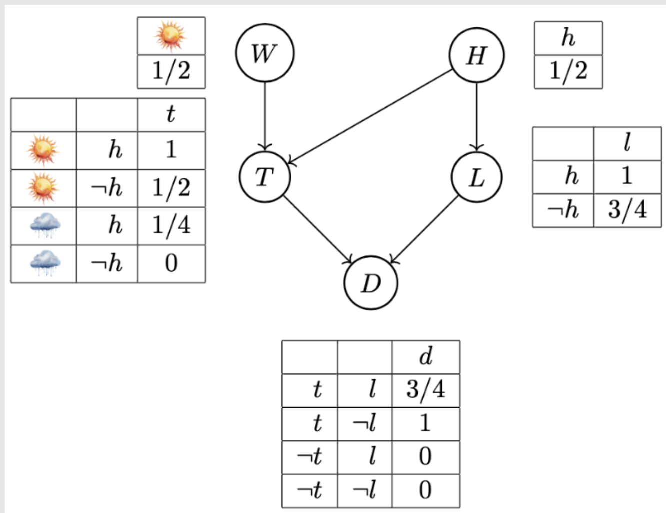

Figure 7

2. **Problem:**

2.4.5 Inference via Sampling

Process:

1.

Example:

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

3 Markov

3.0.1 Random Process

Definition: Time-varying random variables $S_0, S_1, S_2, \ldots$.

3.0.2 Markov Process

Definition: Random process + depends on previous time step only (memoryless)
- w.l.o.g. states can contain history of previous states.

3.1 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:

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

  - 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	$p_T(s) := \mathbb{P}[S_T = s] = \sum_{s'} p_{T-1}(s')p(s\|s')$

- $p_{T-1}(s')$: Prob. s' at T-1 (given)
  - $p_0(s)$: Base case
- $p(s|s')$: Prob. s given s' (from graph)

3.1.1 Bayesian Network

Definition: $S_0, S_1, S_2, \ldots$ form a **Bayesian Network**:

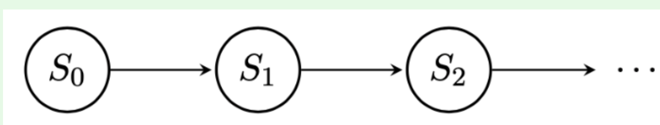

Figure 8

3.2 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., $(S_{t-1}, \varnothing, S_t)$

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	$r(s, s') :=$ reward for transition $(s, \varnothing, s')$
Discount factor	$\gamma \in [0, 1]$
Return after T transitions	$U_T = \sum_{t=1}^{T} \gamma^{t-1} R_t = U_{T-1} + \gamma^{T-1} R_T$

- i.e. The (possibly discounted) sum of the rewards after T transitions (sequence of rewards)
- **Why?**
  - Future rewards are less valuable than immediate rewards.
  - Won't converge if sum goes to ∞ if $\gamma = 1$.

Expected return after T transitions	$\mathbb{E}[U_T] = \mathbb{E}[U_{T-1}] + \gamma^{T-1}\mathbb{E}[R_t] = \mathbb{E}[U_{T-1}] + \gamma^{T-1} \sum_{s,s'} p_{T-1}(s) p(s'\|s) r(s, s')$

- $p_{T-1}(s)p(s'|s)$: Prob. $s \to s'$
- $r(s, s')$: rwd $s \to s'$
- $\mathbb{E}[U_0] := 0$: Base case

3.2.1 Bayesian Network

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

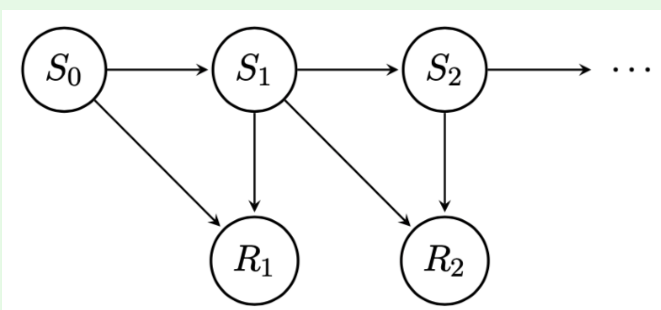

Figure 9

3.3 Markov Decision Processes (MDPs)

3.3.1 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:

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

  - 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	$r(s,a,s') :=$ 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 • A Markov Decision Process can be either: – **Finite**: $T_{\max}$ is finite – **Infinite**: $T_{\max}$ is infinite ∗ For infinite MDPs, we must have $\gamma < 1$.	$T_{\max}$
Prob. that state of the env. after T transitions is s • $p_{T-1}(s)$: Prob. s' at T-1 • $\pi(a\|s')$: Action a from s' • $p(s\|s',a)$: Prob. s given s', a	$p_T(s) = \sum_{a,s'} p_{T-1}(s)\pi(a\|s')p(s\|s',a)$
Expected return after T transitions • $\mathbb{E}_\pi[R_t] = \sum_{s,a,s'} p_{T-1}(s)\pi(a \mid s)p(s' \mid s,a)r(s,a,s')$ • $\mathbb{E}_\pi[U_0] = 0$: Base case.	$\mathbb{E}_\pi[U_T]=\mathbb{E}_\pi[U_{T-1}] + \gamma^{T-1}\mathbb{E}_\pi[R_t]$
Future return after τ transitions • Starting at $\tau + 1$ for the future return.	$G_\tau = \sum_{t=\tau+1}^{T} \gamma^{t-(\tau+1)}R_t = R_{\tau+1} + \gamma G_{\tau+1}$
Expected future return after τ transitions given $S_\tau{=}s$ • $\mathbb{E}_\pi[G_{T_{\max}} \mid S_{T_{\max}} = s] = 0$: Base case. • $\mathbb{E}_\pi[R_{\tau+1} \mid S_\tau = s] = \sum_{a,s'} \pi(a \mid s)p(s' \mid a,s)r(s,a,s')$ – $\pi(a \mid s)p(s' \mid a,s)$: 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 τ.	$\mathbb{E}_\pi[G_\tau \mid S_\tau{=}s]=\mathbb{E}_\pi[R_{\tau+1} \mid S_\tau{=}s] + \gamma\mathbb{E}_\pi[G_{\tau+1} \mid S_\tau{=}s] = \sum_{a,s'} \pi(a \mid s)p(s' \mid s,a)\left(r(s,a,s') + \gamma\mathbb{E}_\pi[G_{\tau+1} \mid S_{\tau+1}{=}s']\right)$

Summary:

Name	**Function:**
Value function	$v_\pi(s,T) := \mathbb{E}_\pi[G_{T_{\max}-T} \mid S_{T_{\max}-T} = s] = \sum_{a,s'} \pi(a \mid s)p(s' \mid s,a)\left(r(s,a,s') + \gamma v_\pi(s', T-1)\right)$ • Value of state s under the policy π with T transitions remaining. – i.e. How good the state is at time T (e.g. If $v(s,T) = 5$, then the expected future return at T is 5). • $v(s,0) = 0$ for all s: Base case
Optimal action	$a^*(s,T) = \arg\max_{a\in\mathcal{A}(s)} \sum_{s'} p(s' \mid s,a)\left(r(s,a,s') + \gamma v_{\pi^*}(s', T-1)\right) = \arg\max_{a\in\mathcal{A}(s)} q^*(s,a,T)$
Optimal policy	$\pi^*(a \mid s,T) = \arg\max_{\pi(a\mid s,T)} \mathbb{E}_\pi[G_\tau \mid S_\tau = s] = \begin{cases} 1 & \text{if } a = a^*(s,T) \\ 0 & \text{otherwise} \end{cases}$ • Choose $\pi(\cdot \mid s)$ to maximize the expected future return after τ transitions given $S_\tau = s$. • **Note:** Policy always depends on transitions remaining so may omit.
Optimal value function	$v^*(s,T) = \max_a \sum_{s'} p(s' \mid a,s)\left(r(s,a,s') + \gamma v^*(s', \tau+1)\right)$ • Assume we use an optimal policy π^*. • $v^*(s,0) = 0$ for all s: Base case.
Q function (quality)	$q_\pi(s,a,T) := \mathbb{E}_\pi[G_{T_{\max}-T} \mid S_{T_{\max}-T} = s, A_{T_{\max}-(T-1)} = a] = \sum_{s'} p(s' \mid s,a)\left(r(s,a,s') + \gamma \sum_{a'} \pi(a' \mid s')q_\pi(s',a',T-1)\right)$ • Quality of move (s,a) under policy π with T transitions remaining. • $q_\pi(s,a,0) = 0$ for all s,a: Base case.
Optimal Q function	$q^*(s,a,T) = \sum_{s'} p(s' \mid s,a)\left(r(s,a,s') + \gamma \max_{a'} q^*(s',a',T-1)\right)$ • $q^*(s,a,0) = 0$ for all s,a: Base case.
IDK Expected Return	$\mathbb{E}_\pi[U_{T_{\max}}] = \sum_s \mathbb{E}_\pi[G_0 \mid S_0 = s]p_0(s) = \sum_s v_\pi(s,0)p_0(s)$ • $G_0 = U_{T_{\max}}$
IDK Optimal Expected Return	$\max_\pi \mathbb{E}[U_{T_{\max}}] = \sum_s v^*(s,0)p_0(s)$

3.3.2 Bayesian Network

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

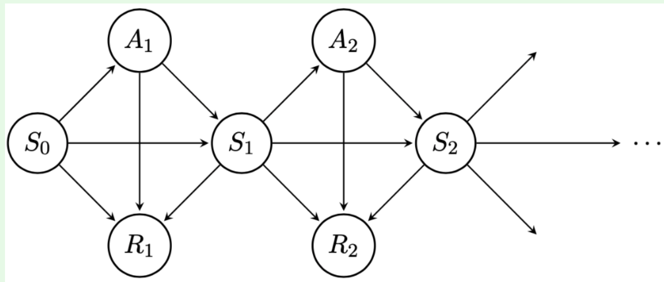

Figure 10

3.3.3 Intuition on Formula

Notes:

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

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

3.4 Canonical Examples

3.4.1 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:

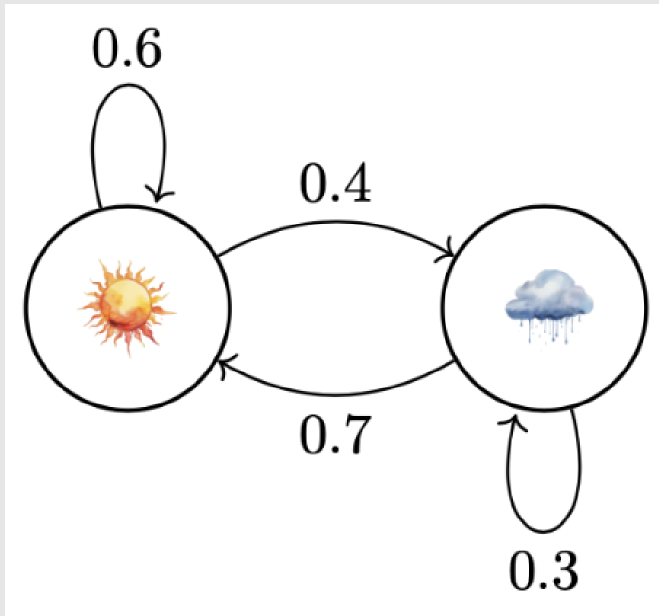

Figure 11

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

3.4.2 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:

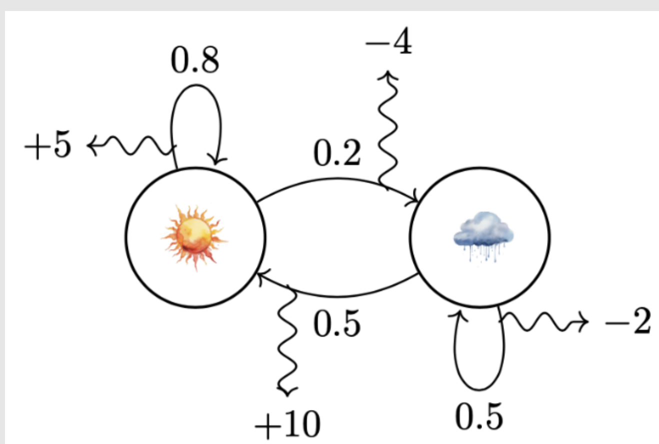

Figure 12

   - 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.

3.4.3 Markov Decision Processes

Example:

1. **Given:**

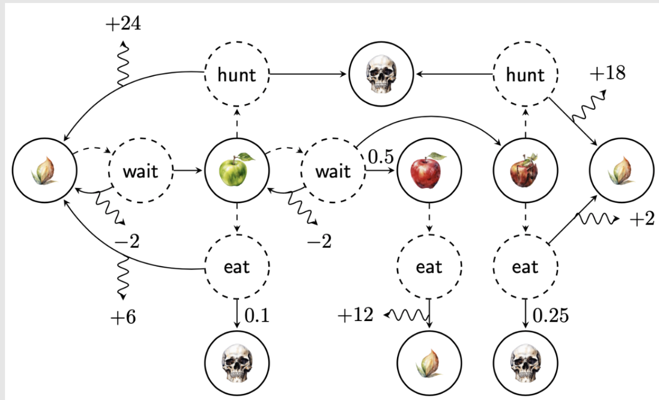

Figure 13

- 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)