Decision Networks
- Evaluating decision networks
  1. Set the evidence variables for the current state
  2. For each possible value of the decision node
     (a) Set the decision node to that value
     (b) Calculate the posterior probabilities for the parent nodes of the utility node, using a standard probabilistic inference algorithm.
     (c) Calculate the resulting utility for the action
  3. Return the action with the highest utility

The value of information
- Not all available information is provided to the agent before it makes its decision
- One of the most important parts of decision making is knowing what questions to ask.
  - To conduct expensive and critical tests or not depends on two factors:
    - Whether the different possible outcomes would make a significant difference to the optimal course of action
    - The likelihood of the various outcomes
- Information value theory enables an agent to choose what information to acquire.

Value of Information
- Solution
  1) block 3 contains oil → the company will buy this block
     profit: \( C - C/n = (n-1)C/n \)
  2) block 3 contains no oil → the company will buy a different block
     profit: \( C/(n-1) - C/n = C/(n(n-1)) \)

  expected profit: \( \frac{1}{n} \cdot \frac{(n-1)C}{n} + \frac{1}{n(n-1)} \cdot C = \frac{C}{n} \)

Value of Information
- General formula
  - Value of perfect information (VPI)
    \[
    EU(a|E) = \max \sum_i U(S_i)P(S_i|E)D(o,A)
    \]
    \[
    EU(a_\alpha|E,E_\beta) = \max \sum_i U(S_i)P(S_i|E,E_\beta)D(o,A,E_\beta)
    \]
  - VPI \( E_\alpha \) = \( \sum P(E_\alpha = E_\beta | E)EU(a_\alpha | E, E_\beta) - EU(a | E) \)

  \( E \): current evidence, \( \alpha \): current best choice
  \( E_\beta \): newly obtained evidence

Example: Finding Juliet
- A robot, Romeo, is in Charles’ office and must deliver a letter to Juliet
- Juliet is either in her office, or in the conference room.
- Without other prior knowledge, each possibility has probability 0.5

  - States: S0: Romeo in Charles’ office
  - S1: Romeo in Juliet’s office and Juliet here
  - S2: Romeo in Juliet’s office and Juliet not here
  - S3: Romeo in conference room and Juliet here
  - S4: Romeo in conference room and Juliet not here

  - Actions: GJO (go to Juliet’s office)
    GCR (go to conference room)

Utility Computation
**Decision Networks**
1. Set the evidence variables for the current state
2. For each possible value of the decision node
   a. Set the decision node to that value
   b. Calculate the posterior probabilities for the parent nodes of the utility node, using a standard probabilistic inference algorithm.
   c. Calculate the resulting utility for the action
3. Return the action with the highest utility

**The value of information**
- Not all available information is provided to the agent before it makes its decision
- One of the most important parts of decision making is knowing what questions to ask.
- To conduct expensive and critical tests or not depends on two factors:
  - Whether the different possible outcomes would make a significant difference to the optimal course of action
  - The likelihood of the various outcomes
- Information value theory enables an agent to choose what information to acquire.

**Value of Information**
- **Solution**
  1) block 3 contains oil → the company will buy this block
     profit: \( C - C/n = (n-1)C/n \)
  2) block 3 contains no oil → the company will buy a different block
     profit: \( C/(n-1) - C/n = C/n(n-1) \)
- **Expected profit:**
  \[
  \frac{1}{n} \frac{(n-1)C}{n} + \frac{n-1}{n(n-1)} C = \frac{C}{n} \]

**Value of Information**
- **General formula**
  - Value of perfect information (VPI)
    \[
    EU(\alpha | E) = \max \sum U(S_j) P(S_j | E, \alpha(A)) \\
    EU(\alpha | E_j, E) = \max \sum U(S_j) P(S_j | E, \alpha(A), E_j) \\
    VPI_j(E) = \sum P(E_j | \alpha_j, E) EU(\alpha_j | E_j, E) - EU(\alpha | E) \\
    \]
  - \( E \): current evidence, \( \alpha \): current best choice
  - \( E_j \): newly obtained evidence

**Ch 17 - Making Complex Decisions**
- **Outline**
  - Sequential Decision Problems
  - Markov Decision Processes
  - Optimal policy
  - Value Iteration

**Example: Finding Juliet**
- A robot, Romeo, is in Charles' office and must deliver a letter to Juliet
  - Juliet is either in her office, or in the conference room.
  - Without other prior knowledge, each possibility has probability 0.5
  - The robot's goal is to minimize the time spent in transit

**Utility Computation**
- **States are:**
  - S0: Romeo in Charles' office
  - S1: Romeo in Juliet's office and Juliet here
  - S2: Romeo in Juliet's office and Juliet not here
  - S3: Romeo in conference room and Juliet here
  - S4: Romeo in conference room and Juliet not here
- **Actions are:**
  - GJO (go to Juliet's office)
  - GCR (go to conference room)
Decision Networks

- Evaluating decision networks
  1. Set the evidence variables for the current state
  2. For each possible value of the decision node
     a. Set the decision node to that value
     b. Calculate the posterior probabilities for the parent nodes of the utility node, using a standard probabilistic inference algorithm.
     c. Calculate the resulting utility for the action
  3. Return the action with the highest utility

The value of information

- Not all available information is provided to the agent before it makes its decision
- One of the most important parts of decision making is knowing what questions to ask.
- To conduct expensive and critical tests or not depends on two factors:
  - Whether the different possible outcomes would make a significant difference to the optimal course of action
  - The likelihood of the various outcomes
- Information value theory enables an agent to choose what information to acquire.

Value of Information

- Solution
  1) Block 3 contains oil → the company will buy this block
     profit: \( C - \frac{C}{n} = \frac{(n-1)C}{n} \)
  2) Block 3 contains no oil → the company will buy a different block
     profit: \( \frac{C}{n} - \frac{C}{n} = \frac{C}{n(n-1)} \)
     expected profit: \( \frac{1}{n} \left( \frac{(n-1)C}{n} - \frac{n-1}{n(n-1)} C \right) = \frac{C}{n} \)

Value of Information

- General formula
  - Value of perfect information (VPI)
    \[
    EU(a|E) = \max \sum_i U(S_j) P(S_j | E, Do(A))
    \]
    \[
    EU(a_e | E, E_j) = \max \sum_i U(S_j) P(S_j | E, Do(A), E_j)
    \]
    \[
    VPI_j(E_j) = \sum_i P(E_i = e_j | E) EU(a_e | E, E_i = e_j) - EU(a|E)
    \]
  - \( E \): current evidence, \( a \): current best choice
  - \( E_j \): newly obtained evidence

Example: Finding Juliet

- A robot, Romeo, is in Charles’ office and must deliver a letter to Juliet
- Juliet is either in her office, or in the conference room.
- Without other prior knowledge, each possibility has probability 0.5
- The robot’s goal is to minimize the time spent in transit

Utility Computation

- States are:
  - S0: Romeo in Charles’ office
  - S1: Romeo in Juliet’s office and Juliet here
  - S2: Romeo in Juliet’s office and Juliet not here
  - S3: Romeo in conference room and Juliet here
  - S4: Romeo in conference room and Juliet not here
- Actions are:
  - GJO (go to Juliet’s office)
  - GCR (go to conference room)
Utility Computation

0

1
GJO

2

3
GCR

4
GCR

3
GCR

1
GJO

5min Juliet’s off.

10min Conf. room

10min
Charles’ off.
Utility Computation

Probabilities
Utility Computation

Probabilities
Action costs
Utility Computation

Probabilities
Action costs
Final reward
Utility Computation

Probabilities
Action costs
Final reward
Expected utility
Another example

- The robot needs to recharge its batteries
- [4,3] provides power supply
- [4,2] is a sand area from which the robot cannot escape
- [4,3] or [4,2] are terminal states
- Reward of a terminal state: +1 or -1
- Reward of a non-terminal state: -1/25
- Utility of a history: sum of rewards of traversed states
- Goal: Maximize expected reward

Utility of an Action Sequence

- Consider the action sequence (U,R) from [3,2]
- A run produces one among 7 possible histories, each with some probability
- The utility of the sequence is the expected utility of the histories:
  \[ U = \sum_h U_h P(h) \]

Probability of Reaching the Goal

- \[ P([4,3] \mid (U,R),[3,2]) = P([4,3] \mid R,[3,3]) \times P([3,3] \mid U,[3,2]) \]
- \[ P([4,3] \mid R,[3,3]) = 0.8 \]
- \[ P([4,3] \mid R,[4,2]) = 0.1 \]
- \[ P([4,3] \mid (U,R),[3,2]) = 0.65 \]

Optimal Policy

- A policy \( P \) is a complete mapping from states to actions
- The optimal policy \( P^* \) is the one that always yields a history (ending at a terminal state) with maximal expected utility.
- Will study more with reinforcement learning

Summary

- Probability theory describes what an agent should believe based on evidence
- Utility theory describes what an agent wants
- Decision theory puts the two together to describe what an agent should do
- A rational agent should select actions that maximize its expected utility.
- Decision networks provide a simple formalism for expressing and solving decision problems.
- Making complex decisions will be studied with reinforcement learning
Another example

• The robot needs to recharge its batteries
• [4,3] provides power supply
• [4,2] is a sand area from which the robot cannot escape
• [4,3] or [4,2] are terminal states
• Reward of a terminal state: +1 or -1
• Reward of a non-terminal state: -1/25
• Utility of a history: sum of rewards of traversed states
• Goal: Maximize the utility of the history

Utility of an Action Sequence

• Consider the action sequence (U,R) from [3,2]
• A run produces one among 7 possible histories, each with some probability
• The utility of the sequence is the expected utility of the histories:
  \[ U = \sum_{h} U_h P(h) \]

Probability of Reaching the Goal

• \( P(\{4,3\} | (U,R)\{3,2\}) = P(\{4,3\} | R\{3,3\}) \times P(\{3,3\} | U\{3,2\}) + P(\{4,3\} | R\{4,2\}) \times P(\{4,2\} | U\{3,2\}) \)
• \( P(\{4,3\} | R\{3,3\}) = 0.8 \quad P(4,3) | U\{3,2\}) = 0.1 \)
• \( P(\{4,3\} | (U,R)\{3,2\}) = 0.65 \)

Optimal Policy

A policy \( P \) is a complete mapping from states to actions
The optimal policy \( P^* \) is the one that has the greatest expected reward among all policies.
What if a terminal state is never reached?

Summary

• Probability theory describes what an agent should believe based on evidence
• Utility theory describes what an agent wants
• Decision theory puts the two together to describe what an agent should do
• A rational agent should select actions that maximize its expected utility.
• Decision networks provide a simple formalism for expressing and solving decision problems.
• Making complex decisions will be studied with reinforcement learning
A Markov Decision Process (MDP) model contains:
- A set of possible world states $S$
- A set of possible actions $A$
- A real valued reward function $R(s,a)$
- A description $T$ of each action’s effects in each state.

We assume the Markov Property: the effects of an action taken in a state depend only on that state and not on the prior history.
Stochastic Automata with Utilities

A *Markov Decision Process* (MDP) model contains:

- A set of possible world states \( S \)
- A set of possible actions \( A \)
- A real valued reward function \( R(s) \)
- A description \( T \) of each action’s effects in each state.

We assume the **Markov Property**: *the effects of an action taken in a state depend only on that state and not on the prior history.*
Representing Actions

Deterministic Actions:

- \( T : S \times A \rightarrow S \) For each state and action we specify a new state.

Stochastic Actions:

- \( T : S \times A \rightarrow Prob(S) \) For each state and action we specify a probability distribution over next states. Represents the distribution \( P(s' | s, a) \).
Representing Actions

Deterministic Actions:

• \( T : S \times A \rightarrow S \) For each state and action we specify a new state.

Stochastic Actions:

• \( T : S \times A \rightarrow \text{Prob}(S) \) For each state and action we specify a probability distribution over next states. Represents the distribution \( P(s' | s, a) \).
Representing Solutions

A *policy* \( \pi \) is a mapping from \( S \) to \( A \)
Following a Policy

Following a policy $\pi$:  
1. Determine the current state $s$  
2. Execute action $\pi(s)$  

Assumes full observability: the new state resulting from executing an action will be known to the system
Evaluating a Policy

How good is a policy $\pi$ in a state $s$?

For deterministic actions just total the rewards obtained... but result may be infinite.

For stochastic actions, instead expected total reward obtained–again typically yields infinite value.

How do we compare policies of infinite value?
Objective Functions

An objective function maps infinite sequences of rewards to single real numbers (representing utility)

Options:
1. Set a finite horizon and just total the reward
2. Discounting to prefer earlier rewards
3. Average reward rate in the limit

Discounting is perhaps the most analytically tractable and most widely studied approach
Discounting

A reward $n$ steps away is discounted by $\gamma^n$ for discount rate $0 < \gamma < 1$.

- models mortality: you may die at any moment
- models preference for shorter solutions
- a smoothed out version of limited horizon lookahead

We use *cumulative discounted reward* as our objective

$$(\text{Max value} \leq M + \gamma \cdot M + \gamma^2 \cdot M + \ldots = \frac{1}{1-\gamma} \cdot M)$$
Value Functions

A value function $V_\pi : S \rightarrow \mathbb{R}$ represents the expected objective value obtained following policy $\pi$ from each state in $S$.

Value functions partially order the policies,
- but at least one optimal policy exists, and
- all optimal policies have the same value function, $V^*$
Bellman Equations

- Bellman equations give a recursive definition of the optimal expected reward

\[ V^*(s) = R(s) + \max_a \gamma \sum_{s'} P(s' | s, a) V^*(s') \]

- If we can compute \( V^* \), we can easily find an optimal policy
  - Choose an action with maximum expected reward
Value Iteration

initialize $V(s)$ to all 0
repeat until (change in $V$ is small)
    for each state $s$ do
        $V'(s) := R(s) + \max_a \gamma \sum_{s'} P(s' | s, a)V(s')$
    end for
    $V := V'$
end repeat
Demo
Advanced Topics

- Explicit search through space of policies (policy iteration)
- Partial observations - POMDP
- Handling large state spaces
  - Factored state and value function representations
  - Approximate representations
    - Tile coding, neural networks, ...
- Simultaneously learning & solving an MDP or POMDP (reinforcement learning)