Uncertainty and Utilities

23 September 2020
Uncertain outcomes
Worst-case vs average case

$max$

$min$

10 10 9 100
Worst-case vs average case
Worst-case vs average case
Worst-case vs average case
Worst-case vs average case

```
max

min

10  10  9  100
```
Worst-case vs average case

Consider uncertain outcomes determined by chance, not an adversary!
Why wouldn’t we know what the result of an action will be?

*Explicit randomness:*

Rolling dice

*Unpredictable opponents:*

The ghosts in Pac-Man respond randomly

*Actions can fail:*

When a robot tries to move, it might slip
Why wouldn’t we know what the result of an action will be?

*Explicit randomness:*
- Rolling dice

*Unpredictable opponents:*
- The ghosts in Pac-Man respond randomly

*Actions can fail:*
- When a robot tries to move, it might slip

The composition of the soils interfered with my traction.
Values should now reflect average-case outcomes, not worst-case outcomes like they do for minimax search.

This change is called expectimax search.
max

10 10 9 100
$\max$
max

10

54.5

chance

10

10

9

100
54.5 \text{ max} \begin{array}{c} 10 \end{array} \begin{array}{c} 54.5 \end{array} \begin{array}{c} \text{chance} \\ 10 \\ 10 \\ 9 \\ 100 \end{array}
**Expectimax search**: Compute the *average* score under optimal play.

- Max nodes are like in minimax search
- *Chance nodes* are like min nodes, but the outcome is uncertain
- Calculate their *expected utilities*, i.e., take weighted average (expectation) of children
**Expectimax search**: Compute the *average* score under optimal play.

Max nodes are like in minimax search

*Chance nodes* are like min nodes, but the outcome is uncertain

Calculate their *expected utilities*, i.e., take weighted average (expectation) of children

Next week, we’ll learn how to formalize the underlying uncertain-result problems as *Markov decision processes*. 
Minimax

SCORE: 0
Minimax

SCORE: 0
Expectimax

SCORE: 0
Expectimax, take 2

SCORE: 0
Expectimax pseudocode

```python
def value(state):
    if state is a terminal state:
        return state’s utility
    if next agent is MAX:
        return max-value(state)
    if next agent is EXP:
        return exp-value(state)
```
Expectimax pseudocode

```python
def max_value(state):
    v = -\infty
    for successor of state:
        v = max(v, value(successor))
    return v

def value(state):
    if state is a terminal state:
        return state's utility
    if next agent is MAX:
        return max-value(state)
    if next agent is EXP:
        return exp-value(state)
```

Expectimax pseudocode

```python
def value(state):
    if state is a terminal state:
        return state’s utility
    if next agent is MAX:
        return max-value(state)
    if next agent is EXP:
        return exp-value(state)

def max-value(state):
    v = -∞
    for successor of state:
        v = max(v, value(successor))
    return v

def exp-value(state):
    v = 0
    for successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```
def exp-value(state):
    \( v = 0 \)
    for successor of state:
        \( p = \text{probability}(\text{successor}) \)
        \( v += p * \text{value}(\text{successor}) \)
    return \( v \)
Expectimax pseudocode

def exp-value(state):
    v = 0
    for successor in state:
        p = probability(successor)
        v += p * value(successor)
    return v

v = 1/2 \cdot 8 + 1/3 \cdot 24 + 1/6 \cdot -12
= 10
Expectimax example
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]

\[
\frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 6
\]
Expectimax example

\[ \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9 \]

\[ \frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 6 \]
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]

\[
\frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 6
\]

\[
\frac{1}{3} \cdot 15 + \frac{1}{3} \cdot 6 + \frac{1}{3} \cdot 0
\]
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]

\[
\frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 6
\]

\[
\frac{1}{3} \cdot 15 + \frac{1}{3} \cdot 6 + \frac{1}{3} \cdot 0
\]
Expectimax example

\[
\frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 12 + \frac{1}{3} \cdot 9
\]
\[
\frac{1}{3} \cdot 2 + \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 6
\]
\[
\frac{1}{3} \cdot 15 + \frac{1}{3} \cdot 6 + \frac{1}{3} \cdot 0
\]
Expectimax pruning?
Expectimax pruning?
Expectimax pruning?
Depth-limited expectimax
Depth-limited expectimax
Depth-limited expectimax

Estimate of true expectimax value (which would require a lot of work to compute)
Probabilities
A *random variable* represents an event whose outcome is unknown.

A *probability distribution* is an assignment of weights to outcomes.
**Example:** Traffic on highway

*Random variable:* $T =$ whether there’s traffic

*Outcomes:* $T$ in \{none, light, heavy\}

*Distribution:*

\[
\begin{align*}
P(T = \text{none}) &= 0.25 \\
P(T = \text{light}) &= 0.50 \\
P(T = \text{heavy}) &= 0.25
\end{align*}
\]

**Some laws of probability:**

- Probabilities are always non-negative
- Probabilities over all possible outcomes sum to one
The **expected value** of a function of a random variable is the average, weighted by the probability distribution over outcomes.
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

*Example*: How long to get to the airport?
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

**Example:** How long to get to the airport?
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

**Example**: How long to get to the airport?

<table>
<thead>
<tr>
<th>Time</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 min</td>
<td>0.25</td>
</tr>
<tr>
<td>30 min</td>
<td>0.50</td>
</tr>
<tr>
<td>60 min</td>
<td>0.25</td>
</tr>
</tbody>
</table>
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

**Example**: How long to get to the airport?

<table>
<thead>
<tr>
<th>Time</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 min</td>
<td>0.25</td>
</tr>
<tr>
<td>30 min</td>
<td>0.50</td>
</tr>
<tr>
<td>60 min</td>
<td>0.25</td>
</tr>
</tbody>
</table>
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

**Example:** How long to get to the airport?

<table>
<thead>
<tr>
<th>Time</th>
<th>Probability</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 min</td>
<td>0.25</td>
<td>$20 \times 0.25$</td>
</tr>
<tr>
<td>30 min</td>
<td>0.50</td>
<td>$30 \times 0.50$</td>
</tr>
<tr>
<td>60 min</td>
<td>0.25</td>
<td>$60 \times 0.25$</td>
</tr>
</tbody>
</table>
The *expected value* of a function of a random variable is the average, weighted by the probability distribution over outcomes.

**Example:** How long to get to the airport?

```
Time  | Probability | Outcome |
------|-------------|---------|
20 min| 0.25        | 20 min  |
30 min| 0.50        | 30 min  |
60 min| 0.25        | 60 min  |
```

\[
(20 \text{ min} \times 0.25) + (30 \text{ min} \times 0.50) + (60 \text{ min} \times 0.25) = 35 \text{ min}
\]
In expectimax search, we have a probabilistic model of how the opponent (or environment) will behave in any state.

- Model could be a simple uniform distribution (roll a die)
- Model could be sophisticated and require a great deal of computation
- We have a chance node for any outcome out of our control: opponent or environment
- The model might say that adversarial actions are likely!

For now, assume each chance node magically comes along with probabilities that specify the distribution over its outcomes.

**Having a probabilistic belief about another agent’s action does not mean that the agent is flipping any coins!**
What probabilities to use?

In expectimax search, we have a probabilistic model of how the opponent (or environment) will behave in any state:

- Model could be a simple uniform distribution (roll a die)
- Model could be sophisticated and require a great deal of computation
- We have a chance node for any outcome out of our control: opponent or environment
- The model might say that adversarial actions are likely!

For now, assume each chance node magically comes along with probabilities that specify the distribution over its outcomes.

Having a probabilistic belief about another agent’s action does not mean that the agent is flipping any coins!
What probabilities to use?

In expectimax search, we have a probabilistic model of how the opponent (or environment) will behave in any state.

- Model could be a simple uniform distribution (roll a die)
- Model could be sophisticated and require a great deal of computation
- We have a chance node for any outcome out of our control: opponent or environment
- The model might say that adversarial actions are likely!

For now, assume each chance node magically comes along with probabilities that specify the distribution over its outcomes.

*Having a probabilistic belief about another agent’s action does not mean that the agent is flipping any coins!
Modeling assumptions
The dangers of optimism and pessimism
The dangers of optimism and pessimism

*Dangerous optimism*

Assuming chance when the world is adversarial
The dangers of optimism and pessimism

*Dangerous optimism*

Assuming chance when the world is adversarial
The dangers of optimism and pessimism

Dangerous optimism
Assuming chance when the world is adversarial

Dangerous pessimism
Assuming the worst case when it’s not likely
The dangers of optimism and pessimism

**Dangerous optimism**
Assuming chance when the world is adversarial

**Dangerous pessimism**
Assuming the worst case when it’s not likely
Assumptions vs reality

Adversarial Ghost   Random Ghost

Minimax
Pac-Man

Expectimax
Pac-Man
Random ghost
Expectimax Pac-Man

SCORE: 0
Random ghost
Expectimax Pac-Man

SCORE: 0
Adversarial ghost
Minimax Pac-Man
Adversarial ghost
Minimax Pac-Man
Adversarial ghost
Expectimax Pac-Man
Adversarial ghost
Expectimax Pac-Man
Random ghost
Minimax Pac-Man

SCORE: -1
Random ghost
Minimax Pac-Man

SCORE:  -1
Assumptions vs reality

<table>
<thead>
<tr>
<th></th>
<th>Adversarial Ghost</th>
<th>Random Ghost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimax</strong></td>
<td>Won 5/5</td>
<td>Won 5/5</td>
</tr>
<tr>
<td><strong>Pac-Man</strong></td>
<td>Avg. Score: 483</td>
<td>Avg. Score: 493</td>
</tr>
<tr>
<td><strong>Expectimax</strong></td>
<td>Won 1/5</td>
<td>Won 5/5</td>
</tr>
<tr>
<td><strong>Pac-Man</strong></td>
<td>Avg. Score: −303</td>
<td>Avg. Score: 503</td>
</tr>
</tbody>
</table>

Results from playing five games

Pac-Man used depth-4 search with an eval. function that avoids trouble.
Ghost used depth-2 search with an eval. function that seeks Pac-Man.
Other game types
Mixed layer types

E.g., backgammon

*Expectiminimax*

Environment is an extra “random agent” player that moves after each min/max agent

Each node computes the appropriate combination of its children
Example: Backgammon

Dice rolls increase $b$: 21 possible rolls with two dice

Backgammon $\approx$ 20 legal moves
Depth $2 = 20 \times (21 \times 20)^3 = 1.2 \times 10^9$

As depth increases, probability of reaching a given search node shrinks

So usefulness of search is diminished
So limiting depth is less damaging
But pruning is trickier…

Historic AI:

TDGammon uses depth-2 search + very good evaluation function + reinforcement learning.
First AI world champion in any game!
Multi-agent utilities

What if the game is not zero-sum, or has multiple players?

Generalization of minimax:

- Terminals have utility tuples
- Node values are also utility tuples
- Each player maximizes its own component
- Can give rise to cooperation and competition dynamically…

```
1, 6, 6
7, 1, 2
6, 1, 2
7, 2, 1
5, 1, 7
1, 5, 2
7, 7, 1
5, 2, 5
```
Multi-agent utilities

What if the game is not zero-sum, or has multiple players?

Generalization of minimax:

Terminals have utility tuples
Node values are also utility tuples
Each player maximizes its own component
Can give rise to cooperation and competition dynamically...
Multi-agent utilities

What if the game is not zero-sum, or has multiple players?

Generalization of minimax:

- Terminals have utility tuples
- Node values are also utility tuples
- Each player maximizes its own component
- Can give rise to cooperation and competition dynamically…
Utilities
Why should we average utilities?
Why not just use minimax?
Why should we average utilities?

Why not just use minimax?

Worse-case reasoning can be dominated by very unlikely possibilities.
Why should we average utilities?

Why not just use minimax?

  Worse-case reasoning can be dominated by very unlikely possibilities. E.g., you shouldn’t go to class because you might get run over by a golf cart and die.
The principle of maximum (expected) utility:

A rational agent should choose the action that maximizes its expected utility, given its knowledge.
Utilities are functions from outcomes (states of the world) to real numbers, which describe an agent’s preferences
Where do utilities come from?

Utilities summarize the agent’s goals
In a game, may be simple: win (+1) or lose (−1)
But any “rational” preferences can be summarized as a utility function

We hard-wire utilities and let behaviors emerge.

Why don’t we let agents pick utilities?
Why don’t we prescribe behaviors?
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*. 

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For worst-case minimax reasoning, terminal function scale doesn’t matter.

We call this *insensitivity to monotonic transformations*.

We just want better states to have higher evaluations (get the ordering right)
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
What utilities to use?

For average-case expectimax reasoning, we need magnitudes to be meaningful.
Utilities: Uncertain outcomes

Getting ice cream

Get single  Get double
Utilities: Uncertain outcomes

Getting ice cream

Get single

Get double
Utilities: Uncertain outcomes

Getting ice cream

Get single

Oops

Get double
Utilities: Uncertain outcomes

Getting ice cream

Get single

Get double

Oops

Whew!
An agent must have *preferences* among:

*Prizes*: $A, B$, etc.

*Lotteries*: situations with uncertain prizes

$L = [p, A; (1 - p), B]$

**Notation:**

*Preference*: $A > B$

*Indifference*: $A \sim B$
Rationality

PLAN A

PLAN B

☐ ☑

☒ ☑

☒ ☑

☒ ☑

☒ ☑
Rational preferences

We want some constraints on preferences before we call them *rational*, such as:

\[
\text{Axiom of Transitivity: } (A > B) \land (B > C) \Rightarrow (A > C)
\]

For example: an agent with *intransitive preferences* can be induced to give away all of its money

- If \( B > C \), then an agent with \( C \) would pay (say) 1¢ to get \( B \)
- If \( A > B \), then an agent with \( B \) would pay (say) 1¢ to get \( A \)
Rational preferences

We want some constraints on preferences before we call them *rational*, such as:

\[
\text{Axiom of Transitivity: } (A > B) \land (B > C) \Rightarrow (A > C)
\]

For example: an agent with *intransitive preferences* can be induced to give away all of its money:

- If \( B > C \), then an agent with \( C \) would pay (say) 1¢ to get \( B \);
- If \( A > B \), then an agent with \( B \) would pay (say) 1¢ to get \( A \);
- If \( C > A \), then an agent with \( A \) would pay (say) 1¢ to get \( C \);
Rational preferences

*The Axioms of Rationality*

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Orderability</strong></td>
<td>((A \succ B) \lor (B \succ A) \lor (A \sim B))</td>
</tr>
<tr>
<td><strong>Transitivity</strong></td>
<td>((A \succ B) \land (B \succ C) \Rightarrow (A \succ C))</td>
</tr>
<tr>
<td><strong>Continuity</strong></td>
<td>(A \succ B \succ C \Rightarrow \exists p \ [p, A; 1 - p, C] \sim B)</td>
</tr>
<tr>
<td><strong>Substitutability</strong></td>
<td>(A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C])</td>
</tr>
<tr>
<td><strong>Monotonicity</strong></td>
<td>(A \succ B \Rightarrow (p \geq q \iff [p, A; 1 - p, B] \succeq [q, A; 1 - q, B]))</td>
</tr>
</tbody>
</table>
Rational preferences

The Axioms of Rationality

Orderability
\[(A \succ B) \lor (B \succ A) \lor (A \sim B)\]

Transitivity
\[(A \succ B) \land (B \succ C) \Rightarrow (A \succ C)\]

Continuity
\[A \succ B \succ C \Rightarrow \exists p \ [p, A; 1 - p, C] \sim B\]

Substitutability
\[A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]\]

Monotonicity
\[A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1 - p, B] \succeq [q, A; 1 - q, B])\]

THEOREM  Rational preferences imply behavior describable as maximization of expected utility.
Rational preferences

The Axioms of Rationality

**Orderability**

\[(A \succ B) \lor (B \succ A) \lor (A \sim B)\]

**Transitivity**

\[(A \succ B) \land (B \succ C) \Rightarrow (A \succ C)\]

**Continuity**

\[A \succ B \succ C \Rightarrow \exists p [p, A; 1 - p, C] \sim B\]

**Substitutability**

\[A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]\]

**Monotonicity**

\[A \succ B \Rightarrow\]

\[(p \geq q \Leftrightarrow [p, A; 1 - p, B] \succeq [q, A; 1 - q, B])\]

**THEOREM**  Rational preferences imply behavior describable as maximization of expected utility.
THEOREM [Ramsey, 1931; von Neumann & Morgenstern, 1944]

Given any preferences satisfying these constraints, there exists a real-valued function $U$ such that:

\[
U(A) \geq U(B) \iff A \succeq B
\]

\[
U([p_1, S_1; \ldots; p_n, S_n]) = \sum_i p_i U(S_i)
\]

i.e., values assigned by $U$ preserve preferences of both prizes and lotteries!

**Maximum expected utility (MEU) principle:**
THEOREM [Ramsey, 1931; von Neumann & Morgenstern, 1944]

Given any preferences satisfying these constraints, there exists a real-valued function $U$ such that:

$$U(A) \geq U(B) \iff A \succeq B$$

$$U([p_1, S_1; \ldots ; p_n, S_n]) = \sum_i p_i U(S_i)$$

i.e., values assigned by $U$ preserve preferences of both prizes and lotteries!

**Maximum expected utility (MEU) principle:**

Choose the action that maximizes expected utility.
Note: An agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities, e.g.,

- a lookup table for perfect tic-tac-toe
- a reflex vacuum cleaner
Human utilities

Spin the wheel or pay $ to pass
Utility scales

*Normalized utilities*: $u_+ = 1.0$, $u_- = 0.0$

*Micromorts*: one-millionth chance of death, useful for paying to reduce product risks, etc.
Utility scales

*Normalized utilities*: \( u_+ = 1.0, \ u_- = 0.0 \)

*Micromorts*: one-millionth chance of death, useful for paying to reduce product risks, etc.

*QALYs*: quality-adjusted life years, useful for medical decisions involving substantial risk.

We can have these different scales because behavior is invariant under positive linear transformation:
Utility scales

Normalized utilities: \( u_+ = 1.0, \ u_- = 0.0 \)

Micromorts: one-millionth chance of death, useful for paying to reduce product risks, etc.

QALYs: quality-adjusted life years, useful for medical decisions involving substantial risk.

We can have these different scales because behavior is invariant under positive linear transformation:

\[ U'(x) = k_1 U(x) + k_2 \quad \text{where} \ k_1 > 0 \]

With deterministic prizes only (no lottery choices), only ordinal utility can be determined, i.e., total order on prizes.
Human utilities

Utilities map states to real numbers. Which numbers?
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a standard lottery $L_p$ between

“best possible prize” $u_+$ with probability $p$

“worst possible catastrophe” $u_-$ with probability $1-p$
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a standard lottery $L_p$ between

“best possible prize” $u_+$ with probability $p$

“worst possible catastrophe” $u_-$ with probability $1-p$
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a standard lottery $L_p$ between

- “best possible prize” $u_+$ with probability $p$
- “worst possible catastrophe” $u_-$ with probability $1-p$

Pay $30
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a \textit{standard lottery} $L_p$ between

“best possible prize” $u_+$ with probability $p$

“worst possible catastrophe” $u_-$ with probability $1-p$

\begin{center}
\begin{tikzpicture}
\node[draw,shape=circle,fill=green!20] (A) at (0,0) {No change};
\node[draw,shape=circle] (B) at (-2,-2) {Pay $30$};
\node[draw,shape=circle] (C) at (2,-2) {No change};
\draw (B) -- (A);
\draw (A) -- (C);
\end{tikzpicture}
\end{center}
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a standard lottery $L_p$ between

- “best possible prize” $u_+$ with probability $p$
- “worst possible catastrophe” $u_-$ with probability $1-p$

Pay $30$  
$\sim$

No change  Instant death
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a \textit{standard lottery} $L_p$ between

- “best possible prize” $u_+$ with probability $p$
- “worst possible catastrophe” $u_-$ with probability $1-p$

Adjust lottery probability $p$ until indifference: $A \sim L_p$

Resulting $p$ is a utility in $[0, 1]$

\[\begin{align*}
\text{Pay } $30 & \sim \text{ No change} \quad \text{Instant death}
\end{align*}\]
Human utilities

Utilities map states to real numbers. Which numbers?

Standard approach to assessment (elicitation) of human utilities:

Compare a prize $A$ to a standard lottery $L_p$ between

“best possible prize” $u_+$ with probability $p$

“worst possible catastrophe” $u_-$ with probability $1-p$

Adjust lottery probability $p$ until indifference: $A \sim L_p$

Resulting $p$ is a utility in $[0, 1]$

Pay $30$ $\sim$

0.999999 $\rightarrow$ 0.000001

No change Instant death
Money

Money *does not* behave as a utility function, but we can talk about the utility of having money (or being in debt)

Given a lottery $L = [p, X; (1 - p), Y]$
Money

Money *does not* behave as a utility function, but we can talk about the utility of having money (or being in debt)

**Given a lottery** $L = [p, X; (1 - p), Y]$  

The expected monetary value, $EMV(L)$, is $p \cdot X + (1 - p) \cdot Y$  
Utility $U(L) = p \cdot U(X) + (1 - p) \cdot U(Y)$  
Typically, $U(L) < U(EMV(L))$
Money

Money *does not* behave as a utility function, but we can talk about the utility of having money (or being in debt)

Given a lottery $L = [p, \$X; (1 - p), \$Y]$

The expected monetary value, $EMV(L)$, is $p \cdot X + (1 - p) \cdot Y$

Utility $U(L) = p \cdot U(\$X) + (1 - p) \cdot U(\$Y)$

Typically, $U(L) < U(EMV(L))$

In this sense, people are *risk-averse*.

But, when deep in debt, people are *risk-prone*.
Money

Money *does not* behave as a utility function, but we can talk about the utility of having money (or being in debt)

Given a lottery \( L = [p, \, \$X; \, (1 - p), \, \$Y] \)

The expected monetary value, \( EMV(L) \), is \( p \cdot X + (1 - p) \cdot Y \)

Utility \( U(L) = p \cdot U(\$X) + (1 - p) \cdot U(\$Y) \)

Typically, \( U(L) < U(EMV(L)) \)

In this sense, people are *risk-averse*.

But, when deep in debt, people are *risk-prone*.
Example: Insurance

Consider the lottery $[0.5, \$1000;\ 0.5, \$0]$
Example: Insurance

Consider the lottery \[0.5, \$1000; \ 0.5, \$0\]

What is its *expected monetary value*? ($500)
Example: Insurance

Consider the lottery $[0.5, \$1000; \ 0.5, \$0]$

What is its *expected monetary value*? ($500$)

What is its *certainty equivalent*?
Example: Insurance

Consider the lottery \([0.5, \$1000; 0.5, \$0]\)

What is its *expected monetary value*? ($500)

What is its *certainty equivalent*?

Monetary value acceptable in lieu of lottery
Example: Insurance

Consider the lottery [0.5, $1000; 0.5, $0]

What is its expected monetary value? ($500)

What is its certainty equivalent?

Monetary value acceptable in lieu of lottery
$400 for most people
Example: Insurance

Consider the lottery \([0.5, \$1000; \ 0.5, \$0]\)

- What is its *expected monetary value*? ($500)
- What is its *certainty equivalent*?
  - Monetary value acceptable in lieu of lottery
  - $400 for most people
- Difference of $100 is the *insurance premium*
  - There’s an insurance industry because people will pay to reduce their risk
  - If everyone were risk-neutral, no insurance needed!
Example: Insurance

Consider the lottery $[0.5, $1000; 0.5, $0]$

What is its *expected monetary value*? ($500)

What is its *certainty equivalent*?

Monetary value acceptable in lieu of lottery

$400 for most people

Difference of $100 is the *insurance premium*

There’s an insurance industry because people will pay to reduce their risk

If everyone were risk-neutral, no insurance needed!

It’s win–win: you’d rather have the $400 and the insurance company would rather have the lottery (their utility curve is flat and they have many lotteries)
Example: Human rationality?

Famous example of Allais (1953)

A: [0.8, $4k; 0.2, $0]
Example: Human rationality?

Famous example of Allais (1953)

A: [0.8, $4k; 0.2, $0]
B: [1.0, $3k; 0.0, $0]

C: [0.2, $4k; 0.8, $0]
Example: Human rationality?

Famous example of Allais (1953)

A: [0.8, $4k; 0.2, $0]
B: [1.0, $3k; 0.0, $0]

C: [0.2, $4k; 0.8, $0]
D: [0.25, $3k; 0.75, $0]

Most people prefer $B \succ A$, $C \succ D$

But if $U($0$) = 0$, then
Example: Human rationality?

Famous example of Allais (1953)

\[ \begin{align*}
    A &: [0.8, \$4k; \ 0.2, \$0] \\
    B &: [1.0, \$3k; \ 0.0, \$0] \\
    C &: [0.2, \$4k; \ 0.8, \$0] \\
    D &: [0.25, \$3k; \ 0.75, \$0] \\
\end{align*} \]

Most people prefer \( B > A, \ C > D \)

But if \( U(\$0) = 0 \), then

\[ \begin{align*}
    B > A \Rightarrow U(\$3k) > 0.8 \ U(\$4k) \\
    C > D \Rightarrow 0.8 \ U(\$4k) > U(\$3k) \\
\end{align*} \]
Example: Human rationality?

Famous example of Allais (1953)

A: [0.8, $4k; 0.2, $0]
B: [1.0, $3k; 0.0, $0]
C: [0.2, $4k; 0.8, $0]
D: [0.25, $3k; 0.75, $0]

Most people prefer \( B \succ A, \ C \succ D \)

But if \( U(0) = 0 \), then

\[
B \succ A \Rightarrow U(3k) > 0.8 \ U(4k)
\]
\[
C \succ D \Rightarrow 0.8 \ U(4k) > U(3k)
\]
Summary

Minimax

Used when the opponent behaves optimally and can be optimized using α–β pruning. Minimax favors conservative actions, so it also tends to yield favorable results when the opponent is unknown.

Expectimax

Used when facing a suboptimal opponent, using a probability distribution over the moves we believe they will make to compute the expected value of states.

Real game trees are big, so use evaluation functions for early termination.

Define utility functions for agents such that they make rational decisions.
Next time: Markov decision processes!
Acknowledgments

The lecture incorporates material from:

Pieter Abbeel, Dan Klein, et al., University of California, Berkeley; ai.berkeley.edu