Decision Networks and Value of Perfect Information
Decision Networks and Value of Perfect Information
Decision Networks
Decision Networks
Decision Networks

Umbrella

Weather

Forecast
Decision Networks

- MEU: choose action which maximizes expected utility given evidence
- Weather Forecast

Directly operationalize with decision networks:
- Bayesian nets with nodes for utility and actions
- Can calculate expected utility for actions

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

MEU: choose action which maximizes expected utility given evidence
Decision Networks

MEU: choose action which maximizes expected utility given evidence

- Umbrella
- Weather
- Forecast

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks

Diagram:
- Umbrella
- Weather
- Forecast

U (Utility)

Bayes nets with nodes for utility and actions.
Can calculate expected utility for actions.

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions

Diagram:
- Umbrella
- Weather
- Forecast
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions
- Can calculate expected utility for actions
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions
- Can calculate expected utility for actions
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions
- Can calculate expected utility for actions

New node types:
- Chance nodes (just like BNs)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions
- Can calculate expected utility for actions

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

- Directly operationalize with decision networks
  - Bayes nets with nodes for utility and actions
  - Can calculate expected utility for actions

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

MEU: choose action which maximizes expected utility given evidence

Directly operationalize with decision networks
- Bayes nets with nodes for utility and actions
- Can calculate expected utility for actions

New node types:
- Chance nodes (just like BNs)
- Actions (rectangles, no parents, acts as observed evidence)
- Utility node (diamond, depends on action/chance nodes)
Decision Networks

- Umbrella
- Weather
- Forecast

Action selection:
1. Instantiate all evidence
2. Set action node(s) each possible way
3. Calculate posterior for parents of utility node, given evidence
4. Calculate expected utility for each action
5. Choose maximizing action
Decision Networks

Umbrella

Weather

Forecast

Action selection
Decision Networks

- Umbrella
- Weather
- Forecast

Action selection
- Instantiate all evidence

- Set action node(s) each possible way
- Calculate posterior for parents of utility node, given evidence
- Calculate expected utility for each action
- Choose maximizing action
Decision Networks

- Weather Forecast
- Umbrella

Action selection
- Instantiate all evidence
- Set action node(s) each possible way
Decision Networks

- Umbrella
- Weather
- Forecast

Action selection:
- Instantiate all evidence
- Set action node(s) each possible way
- Calculate posterior for parents of utility node, given evidence
Decision Networks

Action selection
- Instantiate all evidence
- Set action node(s) each possible way
- Calculate posterior for parents of utility node, given evidence
- Calculate expected utility for each action
Decision Networks

Umbrella

Weather

Forecast

Action selection
- Instantiate all evidence
- Set action node(s) each possible way
- Calculate posterior for parents of utility node, given evidence
- Calculate expected utility for each action
- Choose maximizing action
Decision Networks

Action selection
- Instantiate all evidence
- Set action node(s) each possible way
- Calculate posterior for parents of utility node, given evidence
- Calculate expected utility for each action
- Choose maximizing action
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

$\text{MEU}(\phi) = \max_a \text{EU}(a)$
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[ EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) \]
### Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[
EU(\text{leave}) = \sum_w P(w) U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0
\]
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[
EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70
\]
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[ EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70 \]

Umbrella = take

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[
EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70
\]

Umbrella = take

\[
EU(\text{take}) = \sum_w P(w)U(\text{take}, w)
\]
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[ EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70 \]

Umbrella = take

\[ EU(\text{take}) = \sum_w P(w)U(\text{take}, w) = 0.7 \times 20 + 0.3 \times 70 = 35 \]
Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[ EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70 \]

Umbrella = take

\[ EU(\text{take}) = \sum_w P(w)U(\text{take}, w) = 0.7 \times 20 + 0.3 \times 70 = 35 \]

Optimal decision = leave
## Decision Networks

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[
EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) = 0.7 \times 100 + 0.3 \times 0 = 70
\]

Umbrella = take

\[
EU(\text{take}) = \sum_w P(w)U(\text{take}, w) = 0.7 \times 20 + 0.3 \times 70 = 35
\]

Optimal decision = leave

\[
MEU(\phi) = \max_a EU(a).
\]
Decisions as Outcome Trees

Almost exactly like expectimax / MDPs
What's changed?
XXX: Complicated "path" states. One time reward.
Decisions as Outcome Trees

Almost exactly like expectimax / MDPs

What's changed?
XXX: Complicated "path" states. One time reward.
Decisions as Outcome Trees

Almost exactly like expectimax / MDPs
Decisions as Outcome Trees

Almost exactly like expectimax / MDPs

What’s changed?
Decisions as Outcome Trees

Almost exactly like expectimax / MDPs

What’s changed?

XXX: Complicated “path” states. One time reward.
Example: Decision Networks

Weather
Forecast=bad

Umbrella

| W     | P(W|F=bad) |
|-------|-----------|
| sun   | 0.34      |
| rain  | 0.66      |

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

Optimal decision = take

MEU(F=bad) = max_a EU(a|bad) = 53.
Example: Decision Networks

Weather

Forecast=bad

Umbrella

| W   | \( P(W|F=\text{bad}) \) |
|-----|--------------------------|
| sun | 0.34                     |
| rain| 0.66                     |

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

Umbrella = leave

Optimal decision = take

\( \text{MEU}(F=\text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53. \)
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave} | \text{bad}) = \sum_w Pr(w | \text{bad}) U(\text{leave}, w) \]
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave}|\text{bad}) = \sum_w Pr(w|\text{bad}) U(\text{leave}, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 \]

| W  | \(P(W|F=\text{bad})\) |
|----|------------------------|
| sun| 0.34                   |
| rain| 0.66                  |

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>(U(A,W))</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave}|\text{bad}) = \sum_w \text{Pr}(w|\text{bad}) U(\text{leave}, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave} | \text{bad}) = \sum_{w} Pr(w | \text{bad}) U(\text{leave}, w) \]

\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]

Umbrella = take
Example: Decision Networks

Umbrella = leave

\[ EU(leave | bad) = \sum_w Pr(w | bad) U(leave, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]

Umbrella = take

\[ EU(take | bad) = \sum_w Pr(w | bad) U(take, w) \]
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave}|\text{bad}) = \sum_w Pr(w|\text{bad}) U(\text{leave}, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]

Umbrella = take

\[ EU(\text{take}|\text{bad}) = \sum_w Pr(w|\text{bad}) U(\text{take}, w) = 0.34 \cdot 20 + 0.66 \cdot 70 \]
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave} | \text{bad}) = \sum_w Pr(w | \text{bad}) U(\text{leave}, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]

Umbrella = take

\[ EU(\text{take} | \text{bad}) = \sum_w Pr(w | \text{bad}) U(\text{take}, w) = 0.34 \cdot 20 + 0.66 \cdot 70 = 53. \]
Example: Decision Networks

Umbrella = leave

$$EU(leave|bad) = \sum_w Pr(w|bad) U(leave, w)$$
$$= 0.34 \cdot 100 + 0.66 \cdot 0 = 34.$$ 

Umbrella = take

$$EU(take|bad) = \sum_w Pr(w|bad) U(take, w) = 0.34 \cdot 20 + 0.66 \cdot 70 = 53.$$ 

Optimal decision = take
Example: Decision Networks

Umbrella = leave

\[ EU(\text{leave}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{leave}, w) \]
\[ = 0.34 \cdot 100 + 0.66 \cdot 0 = 34. \]

Umbrella = take

\[ EU(\text{take}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{take}, w) = 0.34 \cdot 20 + 0.66 \cdot 70 = 53. \]

Optimal decision = take

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53. \]
Decisions as Outcome Trees

Weather

U

Umbrella

Take

Leave

Sun

Rain

Same as before, except now we have evidence!
Decisions as Outcome Trees

Weather

Umbrella
U(t,s)
U(t,r)
U(l,s)
U(l,r)

Weather

{b}

take

leave

sun
rain

sun
rain

Same as before, except now we have evidence!
Decisions as Outcome Trees

Same as before, except now we have evidence!
Ghostbusters Decision Network

Ghost Location

Sensor(1,1)  Sensor(1,2)  Sensor(1,3)  ...  Sensor(1,n)

Sensor(2,1)  Sensor(2,2)  Sensor(2,3)  ...  Sensor(2,n)

Bust
Ghostbusters Decision Network

Demo: Ghostbusters with probability
Video of Demo Ghostbusters with Probability
Value of Information
Value of Information

```
<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
```

Idea: compute value of acquiring evidence
can be done directly from decision network
Example: buying oil drilling rights
two blocks, oil in one of A or B.
equally likely.
can drill in one location
Drilling in A or B: has EU = k/2, MEU = k/2
question: what's value of information of O?
value of knowing whether A or B has oil.
value is expected gain in MEU from info.
survey may say "oil in a" or "oil in b."
if know OilLoc, MEU is k (either way).
gain in MEU from knowing OilLoc?
VPI(OilLoc) = k/2
fair price of information: k/2
Idea: compute value of acquiring evidence
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Idea: compute value of acquiring evidence

- Can be done directly from decision network

Example: buying oil drilling rights

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of $A$ or $B$. Equally likely.
- Can drill in one location

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network
Example: buying oil drilling rights
- Two blocks, oil in one of \( A \) or \( B \). Equally likely.
- Can drill in one location
- Drilling in \( A \) or \( B \): has \( EU = k/2 \), \( MEU = k/2 \)

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of \( A \) or \( B \). Equally likely.
- Can drill in one location
- Drilling in \( A \) or \( B \): has \( EU = k/2, MEU = k/2 \)

Question: what’s value of information of \( O \)?
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of O?
- Value of knowing whether A or B has oil.
Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of $A$ or $B$. Equally likely.
- Can drill in one location
- Drilling in $A$ or $B$: has $EU = k/2$, $MEU = k/2$

Question: what's value of information of $O$?
- Value of knowing whether $A$ or $B$ has oil.
- Value is expected gain in MEU from info.
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of O?
- Value of knowing whether A or B has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”

<table>
<thead>
<tr>
<th>D</th>
<th>O</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>k</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/2</td>
</tr>
<tr>
<td>b</td>
<td>1/2</td>
</tr>
</tbody>
</table>
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of O?
- Value of knowing whether A or B has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”
- If know OilLoc, MEU is $k$ (either way).
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of $A$ or $B$. Equally likely.
- Can drill in one location
- Drilling in $A$ or $B$: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of $O$?
- Value of knowing whether $A$ or $B$ has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”
- If know OilLoc, MEU is $k$ (either way).
- Gain in MEU from knowing OilLoc?
Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B.
  - Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what's value of information of O?
- Value of knowing whether A or B has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”
- If know OilLoc, MEU is $k$ (either way).
- Gain in MEU from knowing OilLoc?
- $VPI(OilLoc) = k/2$
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of O?
- Value of knowing whether A or B has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”
- If know OilLoc, MEU is k (either way).
- Gain in MEU from knowing OilLoc?
- $VPI(OilLoc) = k/2$
- Fair price of information: $k/2$
Value of Information

Idea: compute value of acquiring evidence
- Can be done directly from decision network

Example: buying oil drilling rights
- Two blocks, oil in one of A or B. Equally likely.
- Can drill in one location
- Drilling in A or B: has $EU = k/2$, $MEU = k/2$

Question: what’s value of information of O?
- Value of knowing whether A or B has oil.
- Value is expected gain in MEU from info.
- Survey may say “oil in a” or “oil in b.”
- If know OilLoc, MEU is $k$ (either way).
- Gain in MEU from knowing OilLoc?
- $VPI(OilLoc) = k/2$
- Fair price of information: $k/2$
VPI Example: Weather

MEU with no evidence: 70

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
VPI Example: Weather

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

MEU with no evidence: 70

MEU if forecast is bad:

Forecast distribution:

\[
P(F) = \begin{cases} 
0.59 & \text{good} \\
0.41 & \text{bad} 
\end{cases}
\]

VPI \( (E' \mid e) = \sum e' P(e' \mid e) \text{MEU}(e, e') - \text{MEU}(e) \)

VPI = 0.59 \times (95) + 0.41 \times (53) - 70 = 77.8 - 70 = 7.8
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ MEU(F = \text{bad}) = \max_a EU(a | \text{bad}) = 53 \]

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:

\[
\begin{array}{|c|c|c|}
\hline
A & W & U(A,W) \\
\hline
\text{leave} & \text{sun} & 100 \\
\text{leave} & \text{rain} & 0 \\
\text{take} & \text{sun} & 20 \\
\text{take} & \text{rain} & 70 \\
\hline
\end{array}
\]

\[ VPI = 0.59 \cdot (95) + 0.41 \cdot (53) - 70 = 77.8 - 70 = 7.8 \]
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:

\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:

\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F</th>
<th>P(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>0.59</td>
</tr>
<tr>
<td>bad</td>
<td>0.41</td>
</tr>
</tbody>
</table>
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:
\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:
\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th>F</th>
<th>P(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>0.59</td>
</tr>
<tr>
<td>bad</td>
<td>0.41</td>
</tr>
</tbody>
</table>
VPI Example: Weather

**MEU with no evidence:** 70

**MEU if forecast is bad:**

\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

**MEU if forecast is good:**

\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

\[
VPI(E' | e) = (\sum_{e'} P(e' | e) MEU(e, e')) - MEU(e)
\]
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:
\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:
\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A, W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

\[ VPI(E'|e) = (\sum_{e'} P(e' | e) MEU(e, e')) - MEU(e) \]

\[ VPI = 0.59 \cdot (95) + 0.41 \cdot (53) - 70 \]
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:

\[ \text{MEU}(F = \text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53 \]

MEU if forecast is good:

\[ \text{MEU}(F = \text{good}) = \max_a \text{EU}(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th>F</th>
<th>P(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>0.59</td>
</tr>
<tr>
<td>bad</td>
<td>0.41</td>
</tr>
</tbody>
</table>

\[ VPI(E'|e) = (\sum_{e'} P(e'|e) \text{MEU}(e,e')) - \text{MEU}(e) \]

\[ VPI = 0.59 \cdot (95) + 0.41 \cdot (53) - 70 = 77.8 - 70 \]
VPI Example: Weather

MEU with no evidence: 70

MEU if forecast is bad:
\[ MEU(F = \text{bad}) = \max_a EU(a|\text{bad}) = 53 \]

MEU if forecast is good:
\[ MEU(F = \text{good}) = \max_a EU(a|\text{good}) = 95 \]

Forecast distribution:

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>P(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>0.41</td>
<td></td>
</tr>
</tbody>
</table>

VPI Example:
\[
VPI(e', e) = (\sum_{e'} P(e' | e) MEU(e, e')) - MEU(e)
\]

\[
VPI = 0.59 \cdot (95) + 0.41 \cdot (53) - 70 = 77.8 - 70 = 7.8
\]
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$\text{MEU}(e) = \max_a \sum_s P(s|e) U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$\text{MEU}(e, e') = \max_a \sum_s P(s|e, e') U(s, a)$$

BUT $E'$ is a random variable whose value is unknown, so we don't know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:

$$\text{MEU}(e, E') = \sum_{e'} e' P(e'|e) \text{MEU}(e, e')$$

Value of information: how much MEU goes up by revealing $E'$ first then acting over acting now:

$$VPI(E'|e) = \text{MEU}(e, E') - \text{MEU}(e)$$
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e)U(s,a)$$
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e) U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$\text{BUT} E' \text{ is a random variable whose value is unknown, so we don't know what } e' \text{ will be.}$$

Expected value if $E'$ is revealed and then we act:

$$MEU(e, E') = \sum_{e'} P(e'|e) MEU(e, e')$$

Value of information: how much MEU goes up by revealing $E'$ first then acting, over acting now:

$$VPI(E' | e) = MEU(e, E') - MEU(e)$$
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_{a} \sum_{s} P(s|e)U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e, e') = \max_{a} \sum_{s} P(s|e, e')U(s, a)$$

BUT $E'$ is a random variable whose value is unknown, so we don't know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:

$$MEU(e, E') = \sum_{e'} P(e'|e) MEU(e, e')$$

Value of information: how much MEU goes up by revealing $E'$ first then acting, over acting now:

$$VPI(E'|e) = MEU(e, E') - MEU(e)$$
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e) U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e, e') = \max_a \sum_s P(s|e, e') U(s, a)$$

BUT $E'$ is a random variable whose value is unknown, so we don’t know what $e'$ will be.
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e) U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e, e') = \max_a \sum_s P(s|e, e') U(s, a)$$

BUT $E'$ is a random variable whose value is unknown, so we don’t know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e)U(s,a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e,e') = \max_a \sum_s P(s|e,e')U(s,a)$$

BUT $E'$ is a random variable whose value is unknown, so we don’t know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:

$$MEU(e,E') = \sum' e P(e'|e)MEU(e,e')$$
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e)U(s, a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e, e') = \max_a \sum_s P(s|e, e')U(s, a)$$

BUT $E'$ is a random variable whose value is unknown, so we don’t know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:

$$MEU(e, E') = \sum_{e'} P(e'|e)MEU(e, e')$$

Value of information: how much MEU goes up by revealing $E'$ first then acting, over acting now:
Value of Information

Assume we have evidence $E = e$. Value if we act now:

$$MEU(e) = \max_a \sum_s P(s|e)U(s,a)$$

Assume we see that $E' = e'$. Value if we act then:

$$MEU(e, e') = \max_a \sum_s P(s|e,e')U(s,a)$$

BUT $E'$ is a random variable whose value is unknown, so we don’t know what $e'$ will be.

Expected value if $E'$ is revealed and then we act:

$$MEU(e, E') = \sum_{e'} P(e'|e)MEU(e, e')$$

Value of information: how much MEU goes up by revealing $E'$ first then acting, over acting now:

$$VPI(E'|e) = MEU(e, E') - MEU(e).$$
VPI Properties

Nonnegative:
\[ \forall E', e : \text{VPI}(E' | e) \geq 0 \]

Nonadditive (think of observing \(E_j\) twice)
\[ \text{VPI}(E_j, E_k | e) \neq \text{VPI}(E_j | e) + \text{VPI}(E_k | e) \]

Order-independent
\[ \text{VPI}(E_j, E_k | e) = \text{VPI}(E_j | e) + \text{VPI}(E_k | e, E_j) = \text{VPI}(E_k | e) + \text{VPI}(E_j | e, E_k) \]
VPI Properties

Nonnegative:

Order-independent
VPI Properties

Nonnegative:

$$\forall E', e : VPI(E' \mid e) \geq 0$$
VPI Properties

Nonnegative:

\[ \forall E', e : \text{VPI}(E' \mid e) \geq 0 \]
VPI Properties

Nonnegative:
\[ \forall E', e : VPI(E' | e) \geq 0 \]

Nonadditive (think of observing Ej twice)
Nonnegative:
\[ \forall E', e : \text{VPI}(E' \mid e) \geq 0 \]

Nonadditive (think of observing Ej twice)
\[ \text{VPI}(E_j, E_k \mid e) \neq \text{VPI}(E_j \mid e) + \text{VPI}(E_k \mid e) \]
VPI Properties

Nonnegative:
\[ \forall E', e : VPI(E' | e) \geq 0 \]

Nonadditive (think of observing Ej twice)
\[ VPI(E_j, E_k | e) \neq VPI(E_j | e) + VPI(E_k | e) \]

Order-independent
VPI Properties

Nonnegative:
\[ \forall E', e : VPI(E' \mid e) \geq 0 \]

Nonadditive (think of observing $E_j$ twice)
\[ VPI(E_j, E_k \mid e) \neq VPI(E_j \mid e) + VPI(E_k \mid e) \]

Order-independent
\[ VPI(E_j, E_k \mid e) = VPI(E_j \mid e) + VPI(E_k \mid e, E_j) \]
VPI Properties

Nonnegative:
\[ \forall E', e : VPI(E' | e) \geq 0 \]

Nonadditive (think of observing \( E_j \) twice)
\[ VPI(E_j, E_k | e) \neq VPI(E_j | e) + VPI(E_k | e) \]

Order-independent
\[ VPI(E_j, E_k | e) = VPI(E_j | e) + VPI(E_k | e, E_j) \]
VPI Properties

Nonnegative:
\[ \forall E', e : VPI(E' | e) \geq 0 \]

Nonadditive (think of observing Ej twice)
\[ VPI(E_j, E_k | e) \neq VPI(E_j | e) + VPI(E_k | e) \]

Order-independent
\[ VPI(E_j, E_k | e) = VPI(E_j | e) + VPI(E_k | e, E_j) \]
\[ = VPI(E_k | e) + VPI(E_j | e, E_k) \]
Quick VPI Questions

The soup of the day is either clam chowder or split pea, but you wouldn’t order either one. What’s the value of knowing which it is?

There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What’s the value of knowing which?

You’re playing the lottery. The prize will be $0 or $100. You can play any number between 1 and 100 (chance of winning is 1%). What is the value of knowing the winning number?
Quick VPI Questions

The soup of the day is either clam chowder or split pea, but you wouldn’t order either one. What’s the value of knowing which it is?

There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What’s the value of knowing which?

You’re playing the lottery. The prize will be $0 or $100. You can play any number between 1 and 100 (chance of winning is 1%). What is the value of knowing the winning number?
Quick VPI Questions

The soup of the day is either clam chowder or split pea, but you wouldn’t order either one. What’s the value of knowing which it is?

There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What’s the value of knowing which?
Quick VPI Questions

The soup of the day is either clam chowder or split pea, but you wouldn’t order either one. What’s the value of knowing which it is?

There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What’s the value of knowing which?

You’re playing the lottery. The prize will be $0 or $100. You can play any number between 1 and 100 (chance of winning is 1%). What is the value of knowing the winning number?
Value of Imperfect Information?
Value of Imperfect Information?

No such thing
Value of Imperfect Information?

No such thing

Information corresponds to the observation of a node in the decision network
Value of Imperfect Information?

No such thing

Information corresponds to the observation of a node in the decision network

“Noisy” means don’t observe another variable: noisy version of original variable
VPI Question

Generally:

\[ \text{if Parents}(U) \perp \perp Z \mid \text{Current Evidence} \]

Then \[ VPI(Z \mid \text{Current Evidence}) = 0 \]
VPI Question

VPI(OilLoc) ?

Generally:
If Parents(U) \perp \perp Z | CurrentEvidence
Then VPI(Z | CurrentEvidence) = 0
VPI Question

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI Question

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI(Scout) ?

Generally:
$t$ If Parents(U) ⊥ ⊥ $Z$ | CurrentEvidence
$t$ Then VPI($Z$ | CurrentEvidence) = 0
VPI Question

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI(Scout) ?
VPI(Scout | ScoutingReport) ?
**VPI Question**

Generally:

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI(Scout) ?
VPI(Scout | ScoutingReport) ?
VPI Question

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI(Scout) ?
VPI(Scout | ScoutingReport) ?

Generally:
- If Parents(U) ⊥⊥ Z | CurrentEvidence
VPI Question

- VPI(OilLoc) ?
- VPI(ScoutingReport) ?
- VPI(Scout) ?
- VPI(Scout | ScoutingReport) ?

Generally:

- If $\text{Parents}(U) \perp \parallel Z \mid \text{CurrentEvidence}$
- Then $\text{VPI}(Z \mid \text{CurrentEvidence}) = 0$
VPI Question

VPI(OilLoc) ?
VPI(ScoutingReport) ?
VPI(Scout) ?
VPI(Scout | ScoutingReport) ?

Generally:
- If Parents(U) ⊥⊥ Z | CurrentEvidence
- Then VPI( Z | CurrentEvidence) = 0
POMDPs
POMDPs

MDPs have:
- States $S_t$
- Actions $A_t$
- Transition function $P(s'|s, a)$ (or $T(s, a, s')$)
- Rewards $R(s, a, s')$

POMDPs add:
- Observations $O_t$
- Observation function $P(o|s)$ (or $O(s, o)$)

POMDPs are MDPs over belief states $b_t$ (distributions over $S$)

This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:

- States $S$
- Actions $A$
- Transition function $P(s'|s,a)$ (or $T(s,a,s')$)
- Rewards $R(s,a,s')$

POMDPs add:

- Observations $O$
- Observation function $P(o|s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States $S$

POMDPs add:
- Observations $O$
- Observation function $P(o|s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
MDPs have:

- States $S$
- Actions $A$

POMDPs add:

- Observations $O$
- Observation function $P(o|s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' \mid s, a)$ (or $T(s, a, s')$)

POMDPs add:
- Observations $O$
- Observation function $\mathbb{P}(o \mid s)$ (or $O(s, o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' \mid s,a)$ (or $T(s,a,s')$)
- Rewards $R(s,a,s')$

POMDPs add:
- Observations $O$
- Observation function $P(o \mid s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$)
This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States S
- Actions A
- Transition function $P(s' \mid s, a)$ (or $T(s, a, s')$)
- Rewards $R(s, a, s')$

POMDPs add:
- Observations $O$
- Observation function $P(o \mid s)$ (or $O(s, o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' | s, a)$ (or $T(s, a, s')$)
- Rewards $R(s, a, s')$

POMDPs add:
- Observations $O$
- Observation function $P(o | s)$ (or $O(s, o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$). This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' \mid s, a)$ (or $T(s, a, s')$)
- Rewards $R(s, a, s')$

POMDPs add:
- Observations $O$
POMDPs

MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' | s,a)$ (or $T(s,a,s')$)
- Rewards $R(s,a,s')$

POMDPs add:
- Observations $O$
- Observation function $P(o | s)$ (or $O(s,o)$)

This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States S
- Actions A
- Transition function $P(s' | s, a)$ (or $T(s, a, s')$)
- Rewards $R(s, a, s')$

POMDPs add:
- Observations $O$
- Observation function $P(o | s)$ (or $O(s, o)$)

This is like belief states in Hidden Markov Models!
POMDPs

MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' | s,a)$ (or $T(s,a,s')$)
- Rewards $R(s,a,s')$

POMDPs add:
- Observations $O$
- Observation function $P(o | s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$)
POMDPs

MDPs have:
- States $S$
- Actions $A$
- Transition function $P(s' \mid s,a)$ (or $T(s,a,s')$)
- Rewards $R(s,a,s')$

POMDPs add:
- Observations $O$
- Observation function $P(o \mid s)$ (or $O(s,o)$)

POMDPs are MDPs over belief states $b$ (distributions over $S$)
This is like belief states in Hidden Markov Models!
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs:

- One way: use truncated expectimax to compute approximate value of actions.
- What if you only considered busting or one sense followed by a bust?
- You get a VPI-based agent!

Demo: Ghostbusters with VPI
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs:

- One way: use truncated expectimax to compute approximate value of actions.

What if you only considered busting or one sense followed by a bust?

You get a VPI-based agent!

Demo: Ghostbusters with VPI
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence e.
Example: Ghostbusters

In (static) Ghostbusters:
- Belief state determined by evidence \( e \).
- Tree really over evidence sets.

\[
\begin{align*}
\text{Example: Ghostbusters} & \\
\text{In (static) Ghostbusters:} & \\
& \quad \text{Belief state determined by evidence } e. \\
& \quad \text{Tree really over evidence sets.}
\end{align*}
\]
In (static) Ghostbusters:

- Belief state determined by evidence e.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Demo: Ghostbusters with VPI
In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs:

- One way: use truncated expectimax to compute approximate value of actions.
- What if you only considered busting or one sense followed by a bust?

Demo: Ghostbusters with VPI.
In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs

- One way:
Example: Ghostbusters

In (static) Ghostbusters:
- Belief state determined by evidence e.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs
- One way: use truncated expectimax to compute approximate value of actions.
Example: Ghostbusters

In (static) Ghostbusters:
- Belief state determined by evidence e.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence

Solving POMDPs
- One way: use truncated expectimax to compute approximate value of actions
- What if you only considered busting or one sense followed by a bust?
In (static) Ghostbusters:

- Belief state determined by evidence $e$.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence.

Solving POMDPs

- One way: use truncated expectimax to compute approximate value of actions.
- What if you only considered busting or one sense followed by a bust?
- You get a VPI-based agent!
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence e.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence

Solving POMDPs

- One way: use truncated expectimax to compute approximate value of actions
- What if you only considered busting or one sense followed by a bust?
- You get a VPI-based agent!
Example: Ghostbusters

In (static) Ghostbusters:

- Belief state determined by evidence e.
- Tree really over evidence sets.
- Probabilistic reasoning predicts new evidence given past evidence

Solving POMDPs

- One way: use truncated expectimax to compute approximate value of actions
- What if you only considered busting or one sense followed by a bust?
- You get a VPI-based agent!

Demo: Ghostbusters with VPI
Video of Demo Ghostbusters with VPI
POMDPs as Decision Networks

- **MDP:**
  - **States:** $S$
  - **Actions:** $A$
  - **Rewards:** $R(s,a,s')$

- **POMCPs add:**
  - **Observations:** $O$
  - **Observation:** $P(o|s)$
POMDPs as Decision Networks

**MDP:**
- States: S
- Actions: A
- Rewards: $R(s,a,s')$

**POMCPs add:**
- Observations: $O$
- Observation: $P(o|s)$
POMDPs More Generally*

How can we solve POMDPs?

- POMDP is an MDP over belief $b$.
- $s \in \{\text{cool, warm, overheated}\}$.
- $b \in [0, 1]^3$ ← vector of three continuous numbers.
- Use: Value iteration, policy iteration, etc. over $b$.
- $b$ is continuous (discretize or use functions.)
- $b$ is very big (one number for each state.)
POMDPs More Generally*

How can we solve POMDPs?

POMDP is MDP over belief $b$. $s \in \{\text{cool, warm, overheated}\}$. $b \in [0, 1]^3 \leftarrow$ vector of three continuous numbers. $b$ is very big (one number for each state.)

POMDP is MDP over belief $b$. 
POMDPs More Generally*

How can we solve POMDPs?

POMDP is MDP over belief $b$. $s \in \{\text{cool, warm, overheated}\}$. 

POMDP is MDP over belief $b$. $s \in \{\text{cool, warm, overheated}\}$. 

*POMDPs, in general, refer to Partially Observable Markov Decision Processes, a type of decision-making problem where the state of the system is not directly observable. They are used in various applications, including robotics, game theory, and artificial intelligence. The MDP (Markov Decision Process) is a mathematical framework for modeling decision-making situations.
POMDPs More Generally*

How can we solve POMDPs?

POMDP is MDP over belief $b$.

$s \in \{\text{cool, warm, overheated}\}$.

$b \in [0, 1]^3 \leftarrow \text{vector of three continuous numbers!}$
POMDPs More Generally*

How can we solve POMDPs?

POMDP is MDP over *belief* $b$.

$s \in \{\text{cool, warm, overheated}\}$.

$b \in [0, 1]^3 \leftarrow$ vector of three continuous numbers!

Use: Value iteration, policy iteration, etc. over $b$. 
POMDPs More Generally*
How can we solve POMDPs?

POMDP is MDP over belief $b$.
$s \in \{\text{cool, warm, overheated}\}$.
$b \in [0, 1]^3 \leftarrow$ vector of three continuous numbers!
Use: Value iteration, policy iteration, etc. over $b$.
$b$ is continuous (discretize or use functions.)
POMDPs More Generally*

How can we solve POMDPs?

POMDP is MDP over belief $b$.
$s \in \{\text{cool, warm, overheated}\}$.
$b \in [0, 1]^3 \leftarrow \text{vector of three continuous numbers!}$
Use: Value iteration, policy iteration, etc. over $b$.
$b$ is continuous (discretize or use functions.)
$b$ is very big (one number for each state.)
POMDPs More Generally*

General solutions map belief functions to actions.

- Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly).
- Can build approximate policies using discretization methods.
- Can factor belief functions in various ways.

Overall, POMDPs are very (actually PSACE-) hard.

Most real problems are POMDPs, but we can rarely solve them in general!

E.g. depth limited in ghostbusters.
POMDPs More Generally*

General solutions map belief functions to actions
POMDPs More Generally*

General solutions map belief functions to actions
- Divide regions of belief space (set of belief functions)

E.g. depth limited in ghostbusters.
POMDPs More Generally*

General solutions map belief functions to actions

- Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)
POMDPs More Generally*

General solutions map belief functions to actions

- Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)

- Can build approximate policies using discretization methods
POMDPs More Generally*

- General solutions map belief functions to actions
  - Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)
- Can build approximate policies using discretization methods
- Can factor belief functions in various ways

*PSACE- hardness is an approximation hierarchy, not a complexity class.
POMDPs More Generally*

- General solutions map belief functions to actions
  - Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)

- Can build approximate policies using discretization methods
- Can factor belief functions in various ways
POMDPs More Generally*

- General solutions map belief functions to actions
  - Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)

- Can build approximate policies using discretization methods
- Can factor belief functions in various ways

Overall, POMDPs are very (actually PSACE-) hard
POMDPs More Generally*

General solutions map belief functions to actions

- Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)

- Can build approximate policies using discretization methods
- Can factor belief functions in various ways

Overall, POMDPs are very (actually PSACE-) hard

Most real problems are POMDPs, but we can rarely solve them in general!
POMDPs More Generally*

General solutions map belief functions to actions
- Divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)

- Can build approximate policies using discretization methods
- Can factor belief functions in various ways

Overall, POMDPs are very (actually PSACE-) hard

Most real problems are POMDPs, but we can rarely solve them in general! E.g. depth limited in ghostbusters.