CS 188: Artificial Intelligence
Fall 2011

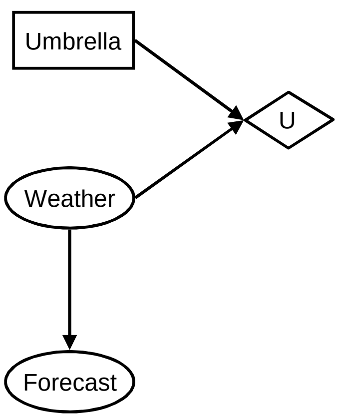
Lecture 17: Decision Diagrams

10/27/2011

Dan Klein – UC Berkeley

Decision Networks

- MEU: choose the action which maximizes the expected utility given the evidence
- Can directly operationalize this with decision networks
  - Bayes nets with nodes for utility and actions
  - Lets us calculate the expected utility for each action
- New node types:
  - Chance nodes (just like BNs)
  - Actions (rectangles, cannot have parents, act as observed evidence)
  - Utility node (diamond, depends on action and chance nodes)

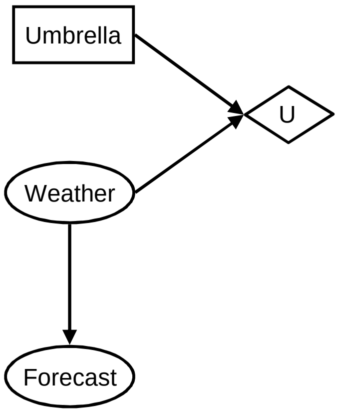


[DEMO: Ghostbusters]

4

Decision Networks

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



Example: Decision Networks

Umbrella = leave

$$\mathrm{EU}(\mathrm{leave}) = \sum_w P(w)U(\mathrm{leave}, w)$$

$$= 0.7 \cdot 100 + 0.3 \cdot 0 = 70$$

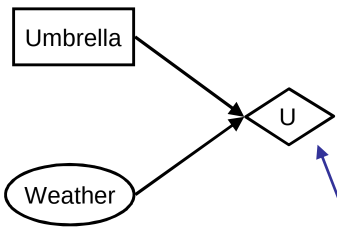
Umbrella = take

$$\mathrm{EU}(\mathrm{take}) = \sum_w P(w)U(\mathrm{take}, w)$$

$$= 0.7 \cdot 20 + 0.3 \cdot 70 = 35$$

Optimal decision = leave

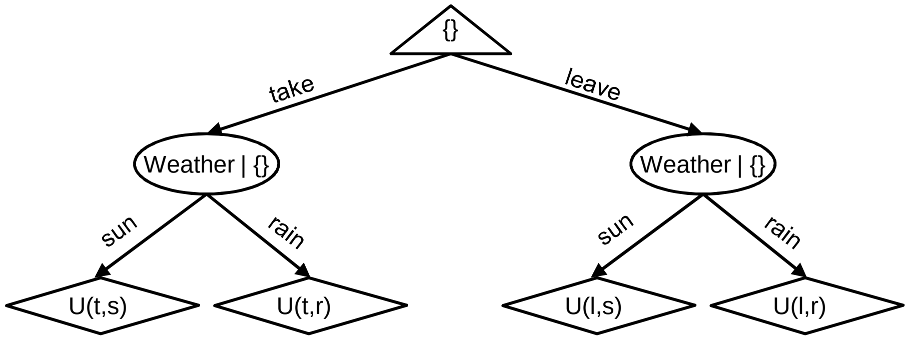
$$\mathrm{MEU}(\phi) = \max_a \mathrm{EU}(a) = 70$$



W	P(W)
sun	0.7
rain	0.3

A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Decisions as Outcome Trees



- Almost exactly like expectimax / MDPs
- What's changed?

Example: Decision Networks

Umbrella = leave

$$\mathrm{EU}(\mathrm{leave}|\mathrm{bad}) = \sum_w P(w|\mathrm{bad})U(\mathrm{leave}, w)$$

$$= 0.34 \cdot 100 + 0.66 \cdot 0 = 34$$

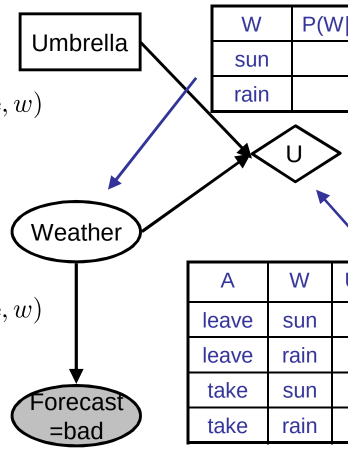
Umbrella = take

$$\mathrm{EU}(\mathrm{take}|\mathrm{bad}) = \sum_w P(w|\mathrm{bad})U(\mathrm{take}, w)$$

$$= 0.34 \cdot 20 + 0.66 \cdot 70 = 53$$

Optimal decision = take

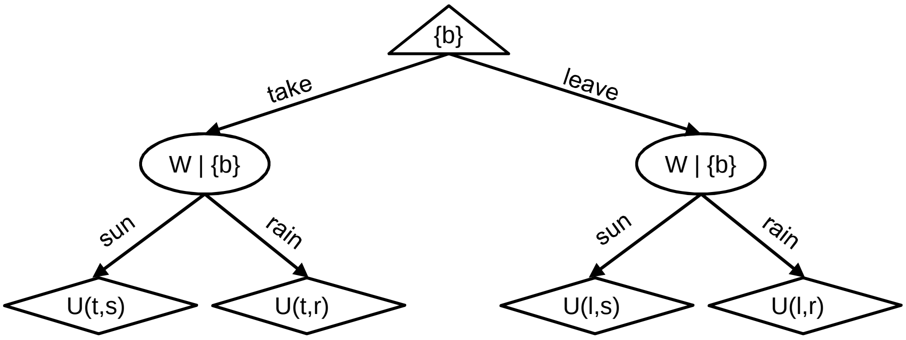
$$\mathrm{MEU}(F = \mathrm{bad}) = \max_a \mathrm{EU}(a|\mathrm{bad}) = 53$$



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

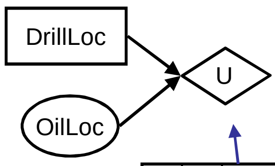
A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Decisions as Outcome Trees



Value of Information

- Idea: compute value of acquiring evidence
  - Can be done directly from decision network
- Example: buying oil drilling rights
  - Two blocks A and B, exactly one has oil, worth k
  - You can drill in one location
  - Prior probabilities 0.5 each, & mutually exclusive
  - Drilling in either A or B has EU = k/2, MEU = k/2
- Question: what's the value of information of O?
  - Value of knowing which of A or B has oil
  - Value is expected gain in MEU from new info
  - Survey may say "oil in a" or "oil in b," prob 0.5 each
  - If we know OilLoc, MEU is k (either way)
  - Gain in MEU from knowing OilLoc?
  - VPI(OilLoc) = k/2
  - Fair price of information: k/2



O	P
a	1/2
b	1/2

D	O	U
a	a	k
a	b	0
b	a	0
b	b	k

VPI Example: Weather

MEU with no evidence

$$\text{MEU}(\emptyset) = \max_a \text{EU}(a) = 70$$

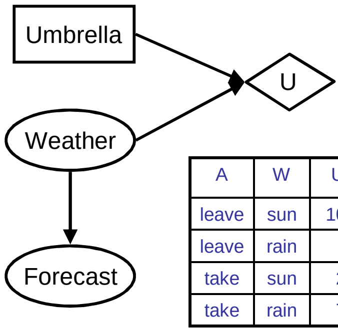
MEU if forecast is bad

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

MEU if forecast is good

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

Forecast distribution



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

F	P(F)
good	0.59
bad	0.41

$$0.59 \cdot (95) + 0.41 \cdot (53) - 70$$
$$77.8 - 70 = 7.8$$

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

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:

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

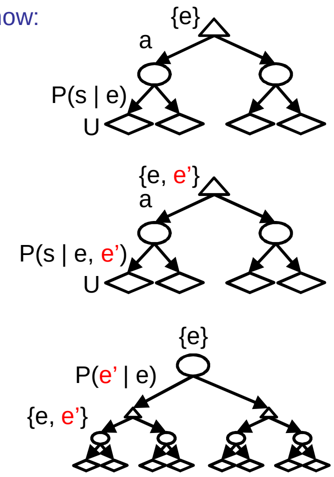
- BUT E' is a random variable whose value is unknown, so we don't know what e' will be

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

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

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

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



VPI Properties

- Nonnegative

$$\forall E', e : \mathrm{VPI}(E'|e) \geq 0$$

- Nonadditive ---consider, e.g., obtaining E_j twice

$$\mathrm{VPI}(E_j, E_k|e) \neq \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e)$$

- Order-independent

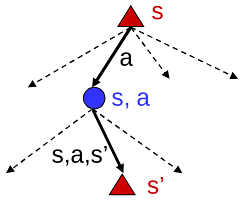
$$\begin{aligned}\mathrm{VPI}(E_j, E_k|e) &= \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e, E_j) \\ &= \mathrm{VPI}(E_k|e) + \mathrm{VPI}(E_j|e, E_k)\end{aligned}$$

Quick VPI Questions

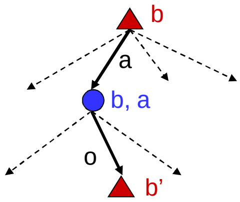
- The soup of the day is either clam chowder or split pea, but you wouldn't order either one. What's the value of knowing which it is?
- There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What's the value of knowing which?
- You're playing the lottery. The prize will be $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?

POMDPs

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



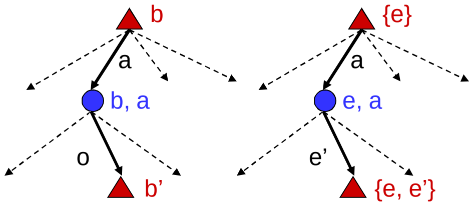
- POMDPs add:
  - Observations O
  - Observation function P(o|s) (or O(s,o))
- POMDPs are MDPs over belief states b (distributions over S)



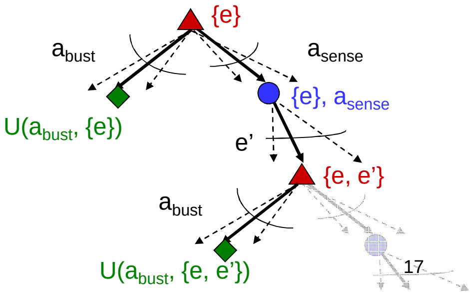
- We'll be able to say more in a few lectures

Example: Ghostbusters

- In (static) Ghostbusters:
  - Belief state determined by evidence to date {e}
  - Tree really over evidence sets
  - Probabilistic reasoning needed to predict new evidence given past evidence

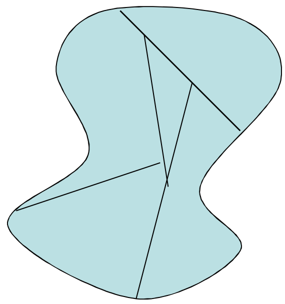


- Solving POMDPs
  - One way: use truncated expectimax to compute approximate value of actions
  - What if you only considered busting or one sense followed by a bust?
  - You get a VPI-based agent!



More Generally

- General solutions map belief functions to actions
  - Can 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 then in general!



18

19

Reasoning over Time

- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring

- Need to introduce time into our models
- Basic approach: hidden Markov models (HMMs)
- More general: dynamic Bayes' nets

20

Markov Models

- A Markov model is a chain-structured BN
  - Each node is identically distributed (stationary)
  - Value of X at a given time is called the state
  - As a BN:

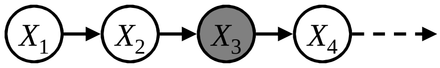
$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$

$$P(X_1) \qquad P(X|X_{-1})$$

  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

[DEMO: Ghostbusters]

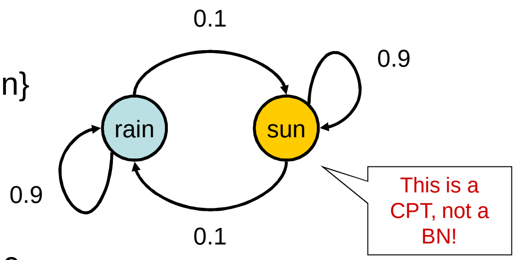
Conditional Independence



- Basic conditional independence:
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property
- Note that the chain is just a (growing) BN
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length

Example: Markov Chain

- Weather:
  - States: X = {rain, sun}
  - Transitions:



  - Initial distribution: 1.0 sun
  - What's the probability distribution after one step?

$$P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9$$

Mini-Forward Algorithm

- Question: probability of being in state x at time t?
- Slow answer:
  - Enumerate all sequences of length t which end in s
  - Add up their probabilities

$$P(X_t = sun) = \sum_{x_1 \ldots x_{t-1}} P(x_1, \ldots x_{t-1}, sun)$$

$$P(X_1 = sun)P(X_2 = sun|X_1 = sun)P(X_3 = sun|X_2 = sun)P(X_4 = sun|X_3 = sun)$$

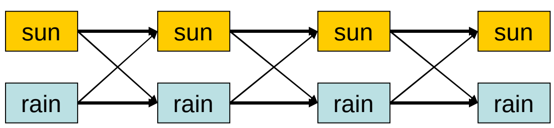
$$P(X_1 = sun)P(X_2 = rain|X_1 = sun)P(X_3 = sun|X_2 = rain)P(X_4 = sun|X_3 = sun)$$

$$\vdots$$

24

Mini-Forward Algorithm

- Better way: cached incremental belief updates
  - An instance of variable elimination!



$$P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$$

$$P(x_1) = \text{known}$$

Forward simulation

Example

- From initial observation of sun

$$\left\langle \begin{matrix} 1.0 \\ 0.0 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.9 \\ 0.1 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.82 \\ 0.18 \end{matrix} \right\rangle \quad \Rightarrow \quad \left\langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \right\rangle$$

$$\mathrm{P}(X_1) \quad \mathrm{P}(X_2) \quad \mathrm{P}(X_3) \quad \quad \mathrm{P}(X_\infty)$$

- From initial observation of rain

$$\left\langle \begin{matrix} 0.0 \\ 1.0 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.1 \\ 0.9 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.18 \\ 0.82 \end{matrix} \right\rangle \quad \Rightarrow \quad \left\langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \right\rangle$$

$$\mathrm{P}(X_1) \quad \mathrm{P}(X_2) \quad \mathrm{P}(X_3) \quad \quad \mathrm{P}(X_\infty)$$

26

Stationary Distributions

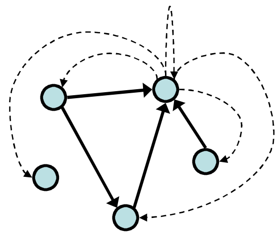
- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!

- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the stationary distribution of the chain
  - Usually, can only predict a short time out

[DEMO: Ghostbusters]

Web Link Analysis

- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (dotted lines)
    - With prob. 1-c, follow a random outlink (solid lines)



- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page!
  - Somewhat robust to link spam
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors

28