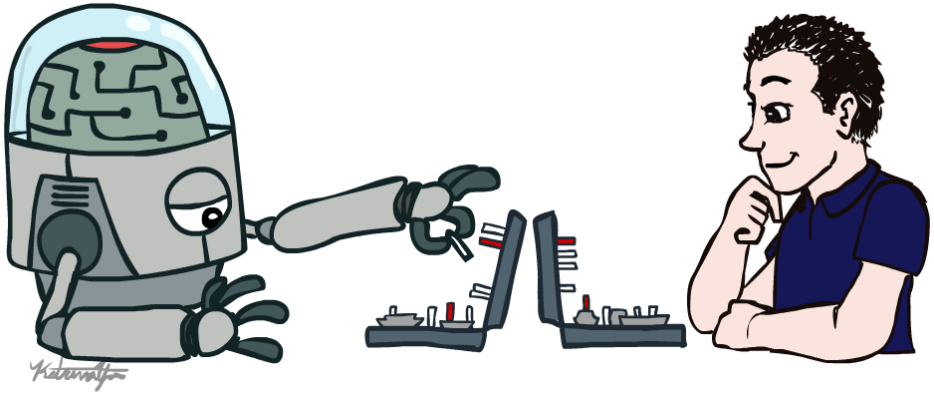
CSCI 446: Artificial Intelligence

Exam 2 Review



Instructor: Michele Van Dyne

Montana Tech

Main Topics

- Probability
- Bayes Nets
- Decision Networks and Value of Information
- Hidden Markov Models
- Naïve Bayes

Probability

- Random Variables
- Joint and Marginal Distributions
- Conditional Distributions
- “Rules”
  - Product Rule
  - Chain Rule
  - Bayes’ Rule
- Inference
- Independence
  - Absolute
  - Conditional

Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$ specifies a real number for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

  - Must obey: $P(x_1, x_2, \ldots x_n) \geq 0$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

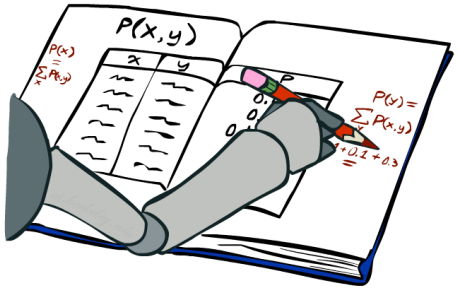
$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Size of distribution if n variables with domain sizes d?
  - For all but the smallest distributions, impractical to write out!

Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding



$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t, s)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$$P(s) = \sum_t P(t, s)$$

$P(W)$

W	P
sun	0.6
rain	0.4

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

The Chain Rule

- Really a generalization of the Product Rule:
  - Definition of conditional probability: $P(x|y) = P(x,y)/P(y)$
  - Product Rule: $P(x,y) = P(x|y)P(y)$ OR
  - $P(x,y) = P(y|x)P(x)$
  - Chain Rule: $P(x_1, x_2, \ldots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \ldots P(x_n|x_1 \ldots x_{n-1})$
  - There are n! ways to order the above conditionals
    - But when we build a Bayes net, we eliminate some of the conditional combinations by the topology of the net
    - Implies we can't get everything we could have gotten from a full joint distribution – but we do get what is important in the problem domain

The Chain Rule

- More generally, can always write any joint distribution as an incremental product of conditional distributions

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \dots x_n) = \prod_i P(x_i|x_1 \dots x_{i-1})$$

- Why is this always true?

The Chain Rule

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})$$

- Why is this always true?
  - i = 1 $P(x_1)$
  - i = 2 $P(x_2|x_1)$
  - i = 3 $P(x_3|x_1, x_2)$
  - …
  - i = n $P(x_n|x_1, x_2, \ldots x_{n-1})$

- And how does we show it is equal to the full joint?
  - An example – next slide

The Chain Rule - Example

$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$ 3 variable chain

$= P(x_1) * \frac{P(x_2, x_1)}{P(x_1)} * \frac{P(x_3, x_2, x_1)}{P(x_2, x_1)}$ Expand conditionals

$= P(x_1) * \frac{P(x_2, x_1)}{P(x_1)} * \frac{P(x_3, x_2, x_1)}{P(x_2, x_1)}$ Cancel terms

$= P(x_3, x_2, x_1)$ The two are equal

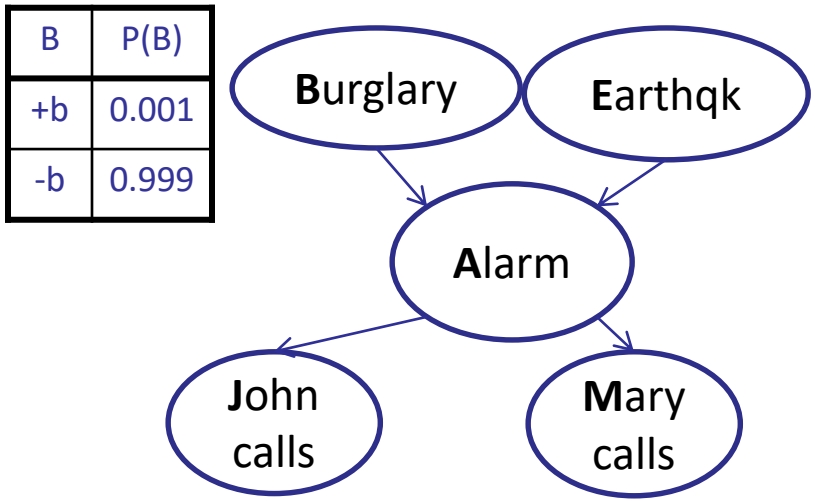
Bayes Nets

- Representation
  - Graphical Model Notation
  - Semantics
    - Conditional Probability Tables
- Independence
  - Bayes Net Independence Assumption
  - D-Separation
    - Causal Chains
    - Common Cause
    - Common Effect

Bayes Nets

- Inference
  - Enumeration
  - Variable Elimination
    - Factors
      - Selected Joint
      - Single Conditional
      - Family of Conditionals
      - Specified Family
    - Variable Ordering
  - Sampling
    - Prior Sampling
    - Rejection Sampling
    - Likelihood Weighting
    - Gibbs Sampling

Example: Alarm Network



B	P(B)
+b	0.001
-b	0.999

E	P(E)
+e	0.002
-e	0.998

A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

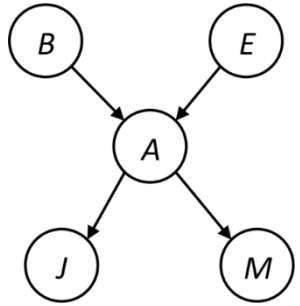
A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

Example

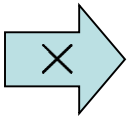
$$P(B|j,m) \propto P(B,j,m)$$

$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$

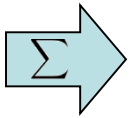


Choose A

$P(A|B,E)$
$P(j|A)$
$P(m|A)$



$P(j,m,A|B,E)$

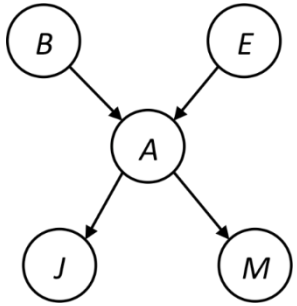


$P(j,m|B,E)$

$P(B)$	$P(E)$	$P(j,m\|B,E)$

Example

$P(B)$ $P(E)$ $P(j, m|B, E)$



Choose E

$P(E)$

$P(j, m|B, E)$ $\times$ $P(j, m, E|B)$ $\sum$ $P(j, m|B)$

$P(B)$ $P(j, m|B)$

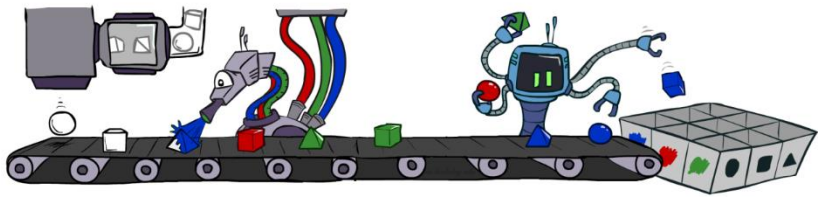
Finish with B

$P(B)$

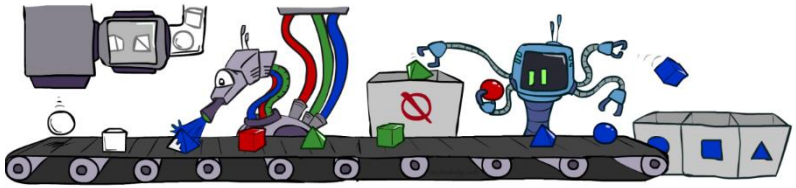
$P(j, m|B)$ $\times$ $P(j, m, B)$ Normalize $P(B|j, m)$

Bayes' Net Sampling Summary

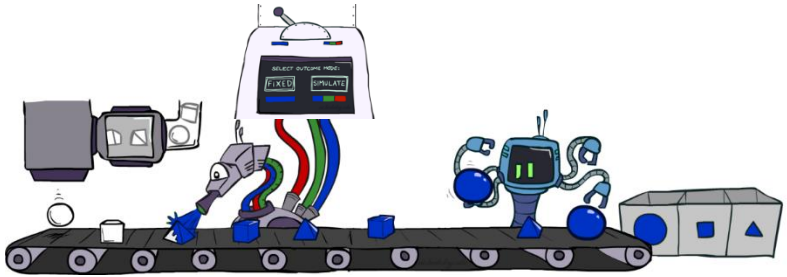
- Prior Sampling P



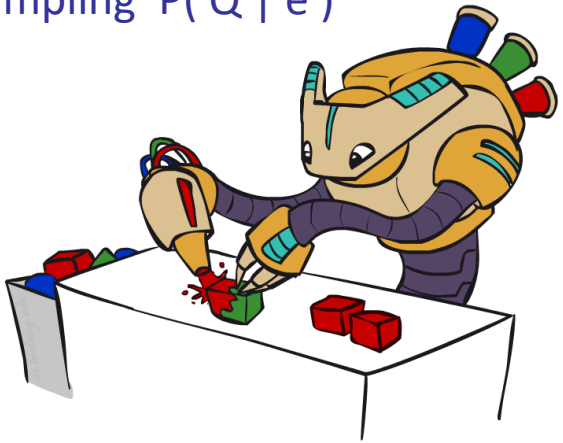
- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)



Decision Networks and Value of Information

- Decision Networks
  - Chance Nodes (Bayes Nets)
  - Action Nodes
  - Utility Nodes
- Value of Information
  - Maximum Expected Utility (MEU)
    - With and without evidence
  - Value of Obtaining Information
  - Properties
    - Non-negative
    - Non-additive
    - Order-independent
- POMDPs – Partially Observable Markov Decision Processes
  - Belief States

Decision Networks

Umbrella = leave

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

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

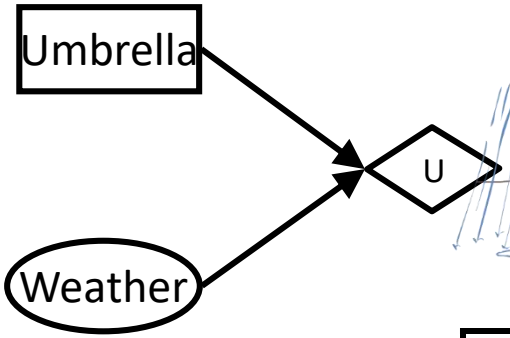
Umbrella = take

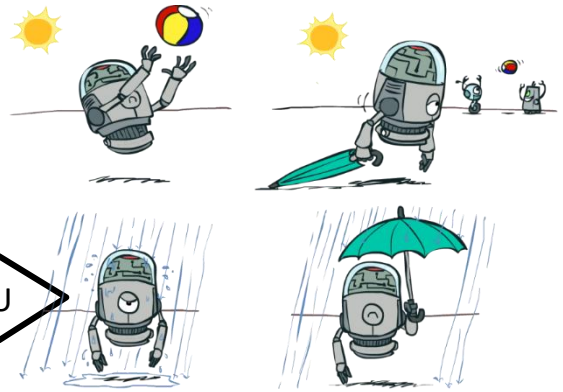
$$\mathrm{EU}(\text{take}) = \sum_w P(w)U(\text{take}, w)$$

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

Optimal decision = leave

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

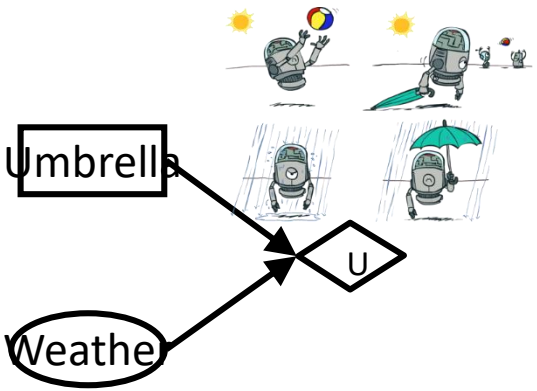


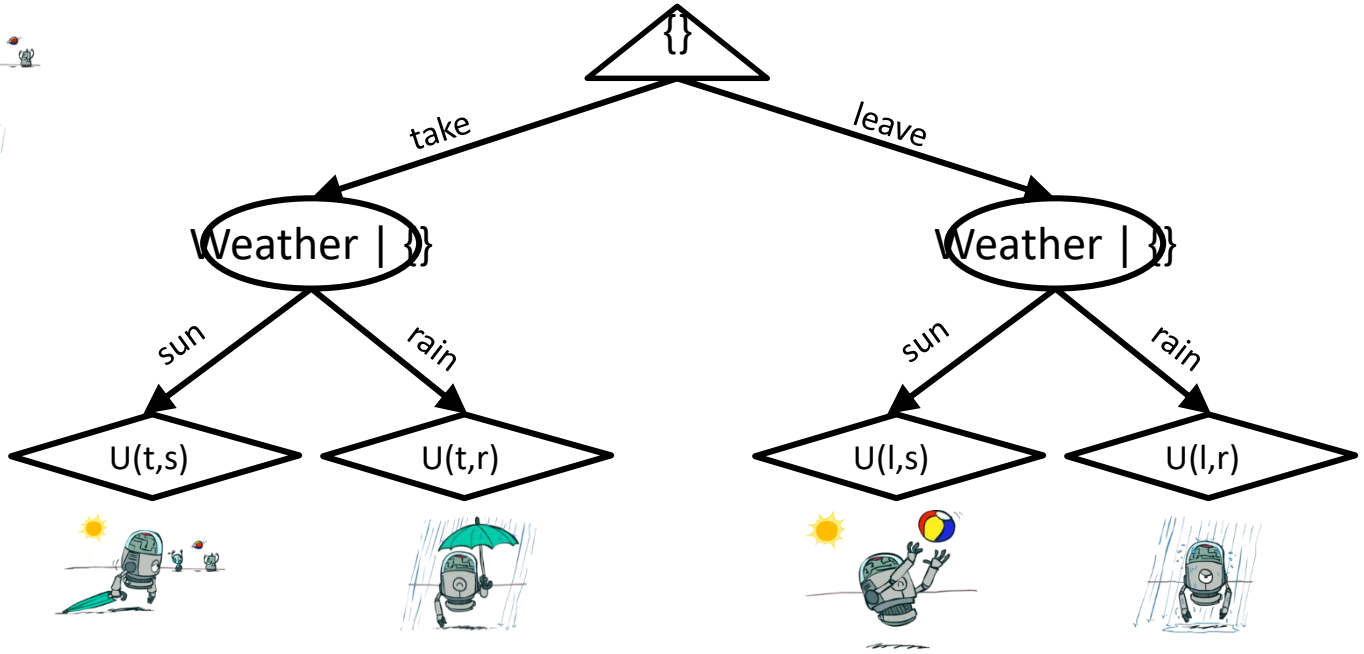


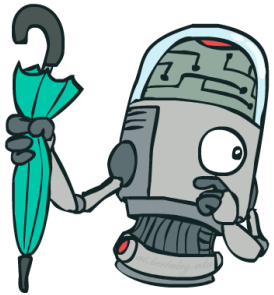
W	P(W)
sun	0.7
rain	0.3

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

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

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

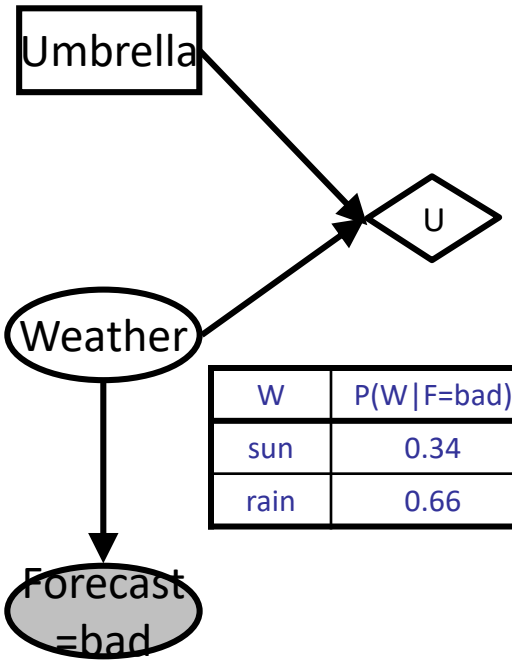
Umbrella = take

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

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

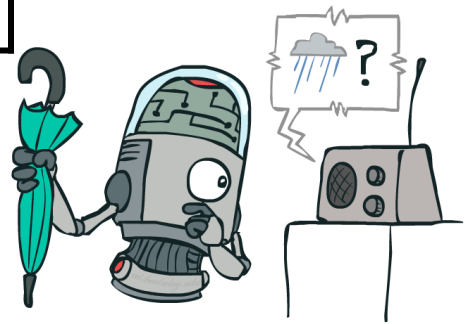
Optimal decision = take

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

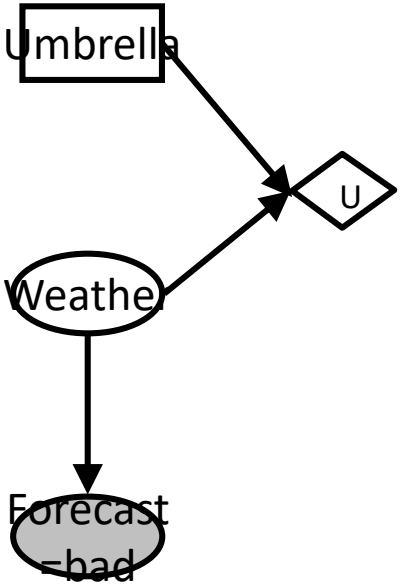


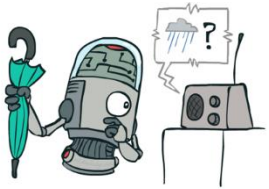
A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

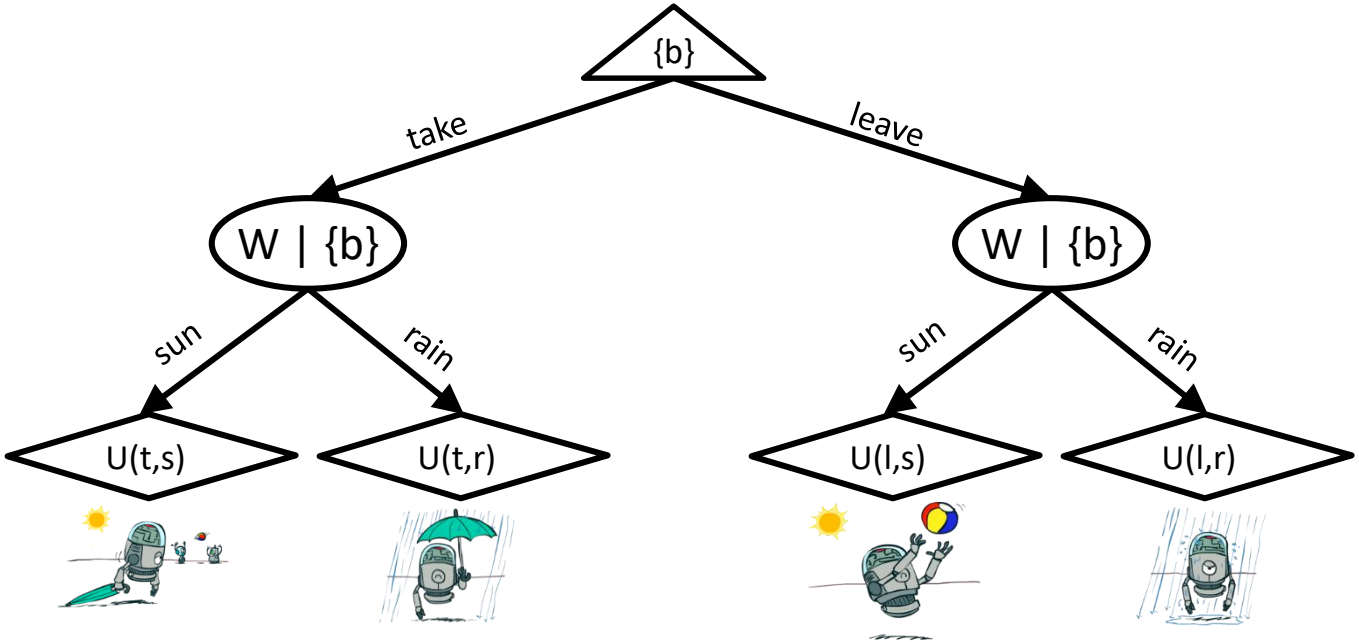
W	P(W|F=bad)
sun	0.34
rain	0.66



Decisions as Outcome Trees

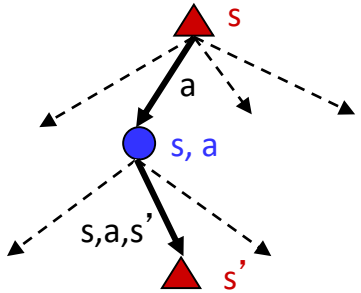




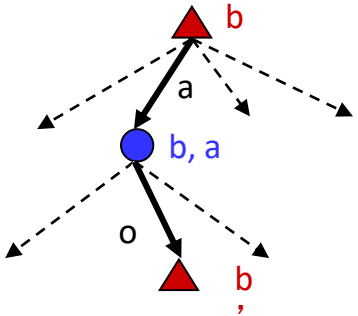


POMDPs

- 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))



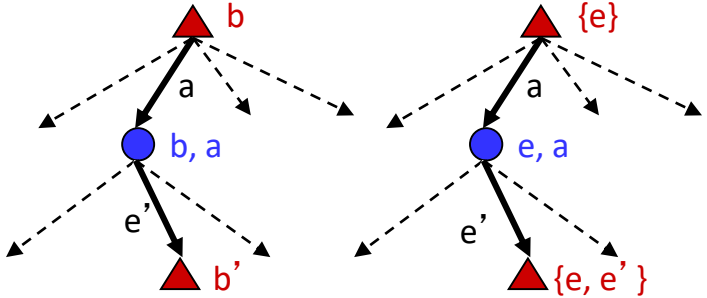
- POMDPs are MDPs over belief states b (distributions over S)

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

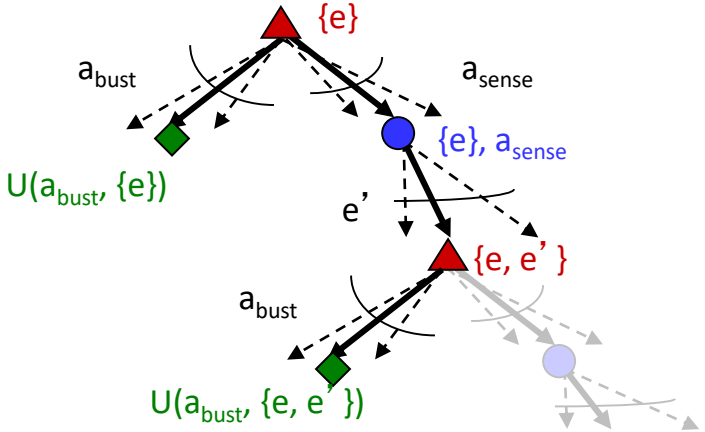
Demo: Ghostbusters with VPI

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!



Hidden Markov Models

- Exact Filtering
  - Base Cases
    - Observation
    - Passage of Time
  - Forward Algorithm
- Particle Filtering
  - Process
    - Generate Particles
    - Elapse Time (Simulate Change)
    - “Observe” Evidence – Weight according to probability
    - Resample
  - Dynamic Bayes Networks
  - Most Likely Explanation (MLE)

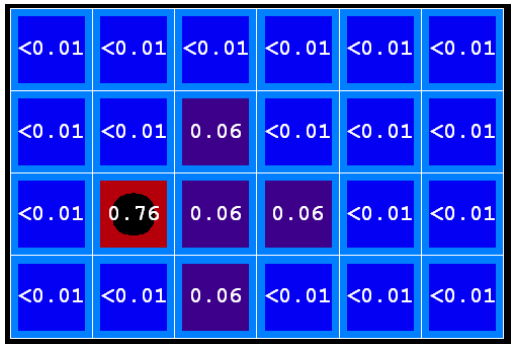
Recap: Filtering

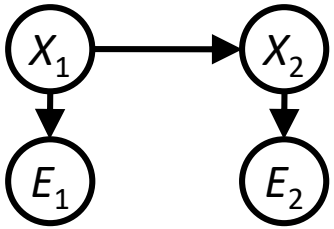
Elapse time: compute P(X_t | $e_{1:t-1}$)

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

Observe: compute P(X_t | $e_{1:t}$)

$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$





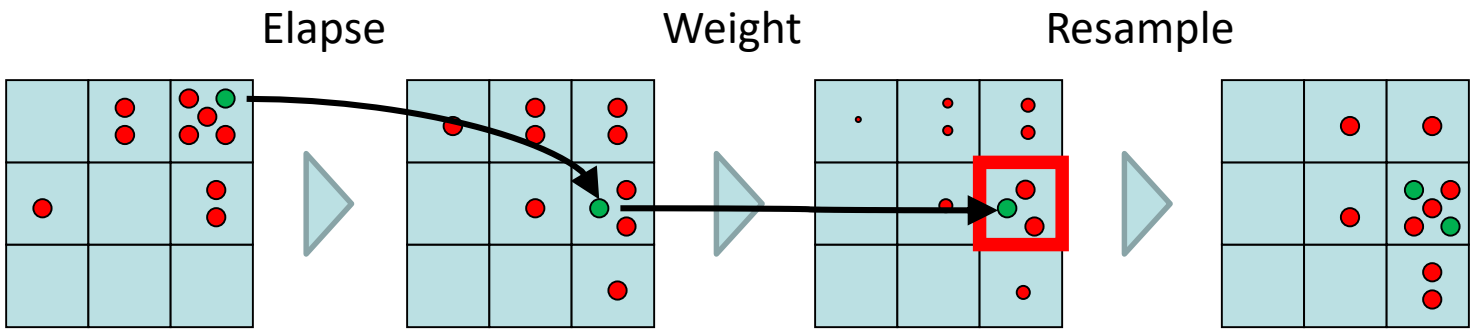
Belief: <P(rain), P(sun)>

$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

[Demo: Ghostbusters Exact

Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution



Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

(New) Particles:
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)

[Demos: ghostbusters particle filtering (L15D3,4,5)]

Naïve Bayes

- Classification
  - Model-Based Classification
- Training and Testing
  - Generalization and Overfitting
  - Parameter Estimation
  - Smoothing
  - Unseen Events
  - Tuning
  - Features

Questions