



# Artificial Intelligence Programming

## *Probabilistic Inference*

Chris Brooks

Department of Computer Science

University of San Francisco



# Probability Review

- Probability allows us to represent a belief about a statement, or a likelihood that a statement is true.
  - $P(\text{rain}) = 0.6$  means that we believe it is 60% likely that it is currently raining.
- Axioms:
  - $0 \leq P(a) \leq 1$
  - The probability of  $(A \vee B)$  is  $P(A) + P(B) - P(A \wedge B)$
  - Tautologies have  $P = 1$
  - Contradictions have  $P = 0$

# Conditional Probability

- Once we begin to make observations about the value of certain variables, our belief in other variables changes.
  - Once we notice that it's cloudy,  $P(Rain)$  goes up.
- this is called *conditional probability*
- Written as:  $P(Rain|Cloudy)$
- $P(a|b) = \frac{P(a \wedge b)}{P(b)}$
- or  $P(a \wedge b) = P(a|b)P(b)$ 
  - This is called the *product rule*.

# Conditional Probability

- Example:  $P(\textit{Cloudy}) = 0.3$
- $P(\textit{Rain}) = 0.2$
- $P(\textit{cloudy} \wedge \textit{rain}) = 0.15$
- $P(\textit{cloudy} \wedge \neg \textit{Rain}) = 0.1$
- $P(\neg \textit{cloudy} \wedge \textit{Rain}) = 0.1$
- $P(\neg \textit{Cloudy} \wedge \neg \textit{Rain}) = 0.65$
- Initially,  $P(\textit{Rain}) = 0.2$ . Once we see that it's cloudy,  
$$P(\textit{Rain}|\textit{Cloudy}) = P\frac{(\textit{Rain}\wedge\textit{Cloudy})}{P(\textit{Cloudy})} = \frac{0.15}{0.3} = 0.5$$

# Combinations of events

- The probability of  $(A \wedge B)$  is  $P(A|B)P(B)$
- What if  $A$  and  $B$  are independent?
- Then  $P(A|B)$  is  $P(A)$ , and  $P(A \wedge B)$  is  $P(A)P(B)$ .
- Example:
  - What is the probability of “heads” five times in a row?
  - What is the probability of at least one “head”?

# Bayes' Rule

- Often, we want to know how a probability changes as a result of an observation.
- Recall the Product Rule:
  - $P(a \wedge b) = P(a|b)P(b)$
  - $P(a \wedge b) = P(b|a)P(a)$
- We can set these equal to each other
  - $P(a|b)P(b) = P(b|a)P(a)$
- And then divide by  $P(a)$ 
  - $P(b|a) = \frac{P(a|b)P(b)}{P(a)}$
- This equality is known as Bayes' theorem (or rule or law).

# Monty Hall Problem

From the game show “Let’s make a Deal”

- Pick one of three doors. Fabulous prize behind one door, goats behind other 2 doors.
- Monty opens one of the doors you did not pick, shows a goat
- Monty then offers you the chance to switch doors, to the other unopened door
- Should you switch?

# Monty Hall Problem

## Problem Clarification:

- Prize location selected randomly
- Monty always opens a door, allows contestants to switch
- When Monty has a choice about which door to open, he chooses randomly.

Variables    Prize:  $P = p_A, p_B, p_C$   
                  Choose:  $C = c_A, c_B, c_C$   
                  Monty:  $M = m_A, m_B, m_C$

# Monty Hall Problem

Without loss of generality, assume:

- Choose door A
- Monty opens door B

$$P(p_A | c_A, m_B) = ?$$

# Monty Hall Problem

Without loss of generality, assume:

- Choose door A
- Monty opens door B

$$P(p_A | c_A, m_B) = P(m_B | c_A, p_A) \frac{P(p_A | c_A)}{P(m_B | c_A)}$$

# Monty Hall Problem

$$P(p_A|c_A, m_B) = P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)}$$

- $P(m_B|c_A, p_A) = ?$

# Monty Hall Problem

$$P(p_A|c_A, m_B) = P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)}$$

- $P(m_B|c_A, p_A) = 1/2$
- $P(p_A|c_A) = ?$

# Monty Hall Problem

$$P(p_A|c_A, m_B) = P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)}$$

- $P(m_B|c_A, p_A) = 1/2$

- $P(p_A|c_A) = 1/3$

- $P(m_B|c_A) = ?$

# Monty Hall Problem

$$P(p_A|c_A, m_B) = P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)}$$

- $P(m_B|c_A, p_A) = 1/2$

- $P(p_A|c_A) = 1/3$

- $$P(m_B|c_A) = P(m_b|c_A, p_A)P(p_A) +$$
$$P(m_b|c_A, p_B)P(p_B) +$$
$$P(m_b|c_A, p_C)P(p_C)$$

# Monty Hall Problem

$$P(p_A|c_A, m_B) = P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)}$$

- $P(m_B|c_A, p_A) = 1/2$

- $P(p_A|c_A) = 1/3$

- $$P(m_B|c_A) = P(m_b|c_A, p_A)P(p_A) +$$
$$P(m_b|c_A, p_B)P(p_B) +$$
$$P(m_b|c_A, p_C)P(p_C)$$

- $P(p_A) = P(p_B) = P(p_C) = 1/3$

- $P(m_b|c_A, p_A) = 1/2$

- $P(m_b|c_A, p_B) = 0$

Won't open prize door

- $P(m_b|c_A, p_C) = 1$

Monty has no choice

# Monty Hall Problem

$$\begin{aligned}P(p_A|c_A, m_B) &= P(m_B|c_A, p_A) \frac{P(p_A|c_A)}{P(m_B|c_A)} \\ &= 1/3\end{aligned}$$

- $P(m_B|c_A, p_A) = 1/2$

- $P(p_A|c_A) = 1/3$

- $$\begin{aligned}P(m_B|c_A) &= P(m_b|c_A, p_A)P(p_A) + \\ &\quad P(m_b|c_A, p_B)P(p_B) + \\ &\quad P(m_b|c_A, p_C)P(p_C) = 1/2\end{aligned}$$

- $P(p_A) = P(p_B) = P(p_C) = 1/3$

- $P(m_b|c_A, p_A) = 1/2$

- $P(m_b|c_A, p_B) = 0$

Won't open prize door

- $P(m_b|c_A, p_C) = 1$

Monty has no choice

# Monty Hall Problem

$$\begin{aligned}P(p_C|c_A, m_B) &= P(m_B|c_A, p_C) \frac{P(p_C|c_A)}{P(m_B|c_A)} \\ &= 2/3\end{aligned}$$

- $P(m_B|c_A, p_C) = 1$

- $P(p_C|c_A) = 1/3$

- $$\begin{aligned}P(m_B|c_A) &= P(m_b|c_A, p_A)P(p_A) + \\ &\quad P(m_b|c_A, p_B)P(p_B) + \\ &\quad P(m_b|c_A, p_C)P(p_C) = 1/2\end{aligned}$$

- $P(p_A) = P(p_B) = P(p_C) = 1/3$

- $P(m_b|c_A, p_A) = 1/2$

- $P(m_b|c_A, p_B) = 0$

Won't open prize door

- $P(m_b|c_A, p_C) = 1$

Monty has no choice

# Inference with the JPD

- Remember that we can list all combinations of events and use this to do inference.
- This is the joint probability distribution (JPD).

	Hum = High Sky = Overcast	Hum = High Sky = Sunny	Hum = Normal Sky = Overcast	Hum = Normal Sky = Sunny
Rain	0.1	0.05	0.15	0.05
$\neg$ Rain	0.2	0.15	0.1	0.2

- $$P(\text{Rain}) = P(\text{Rain} \wedge \text{Hum} = \text{High} \wedge \text{Sky} = \text{Overcast}) \vee P(\text{Rain} \wedge \text{Hum} = \text{Normal} \wedge \text{Sky} = \text{Overcast}) \vee P(\text{Rain} \wedge \text{Hum} = \text{High} \wedge \text{Sky} = \text{Sunny}) \vee P(\text{Rain} \wedge \text{Hum} = \text{Normal} \wedge \text{Sky} = \text{Sunny})$$
- $$P(\text{Rain}) = 0.1 + 0.05 + 0.15 + 0.05 = 0.35$$

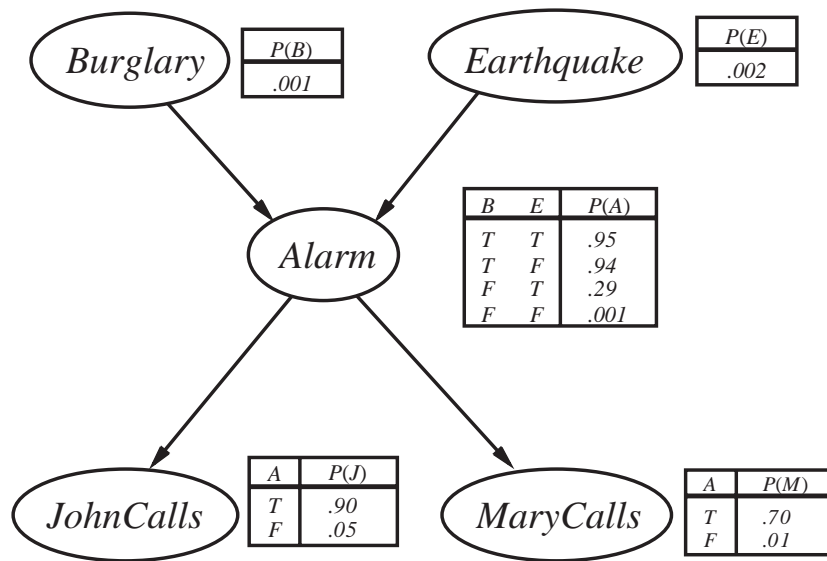
# Probabilistic Inference

- Problems in working with the joint probability distribution:
  - Exponentially large table.
- We need a data structure that captures the fact that many variables do not influence each other.
  - For example, the color of Bart's hat does not influence whether Homer is hungry.
- We call this structure a *Bayesian network* (or a *belief network*)

# Bayesian Network

- A Bayesian network is a directed graph in which each node is annotated with probability information. A network has:
  - A set of random variables that correspond to the nodes in the network.
  - A set of directed edges or arrows connecting nodes. These represent influence. In there is an arrow from  $X$  to  $Y$ , then  $X$  is the parent of  $Y$ .
  - Each node keeps a conditional probability distribution indicating the probability of each value it can take, conditional on its parents values.
  - No cycles. (it is a directed acyclic graph)
- The topology of the network specifies what variables directly influence other variables. (conditional independence relationships).

# Burglary example



- Two neighbors will call when they hear your alarm.
  - John sometimes overreacts
  - Mary sometimes misses the alarm.
- Two things can set off the alarm
  - Earthquake
  - Burglary
- Given who has called, what's the probability of a burglary?

# Network structure

- Each node has a conditional probability table.
- This gives the probability of each value of that node, given its parents' values.
- These sum to 1.
- Nodes with no parents just contain priors.

# Summarizing uncertainty

- Notice that we don't need to have nodes for all the reasons why Mary might not call.
  - A probabilistic approach lets us summarize this information in  $\neg M$
- This allows a small agent to deal with large worlds that have a large number of possibly uncertain outcomes.
- How would we handle this with logic?

# Implicitly representing the full JPD

- Recall that the full joint distribution allows us to calculate the probability of any variable, given all the others.
- Independent events can be separated into separate tables.
- These are the CPTs seen in the Bayesian network.
- Therefore, we can use this info to perform computations.

- $P(x_1, x_2, \dots, x_n) = \prod P(x_i | \text{parents}(x_i))$

- $$P(A \wedge \neg E \wedge \neg B \wedge J \wedge M) =$$
$$P(J|A)P(M|A)P(A|\neg B \wedge \neg E)P(\neg B)P(\neg E) =$$
$$0.90 * 0.70 * 0.001 * 0.999 * 0.998 = 0.00062$$

# Some examples

- What is the probability that Both Mary and John call, given that the alarm sounded?
  - $P(M|A) * P(J|A) = .90 * .70 = 0.63$
- What is the probability of a breakin, given that we hear an alarm?
  - $P(B|A) = 0.95 + 0.001$
- What is the probability of a breakin given that both John and Mary called?
  - $P(B|J, M) = P(B \wedge J \wedge M \wedge A \wedge \neg E) \vee P(B \wedge J \wedge M \wedge A \wedge E) \vee P(B \wedge J \wedge M \wedge \neg A \wedge \neg E) \vee P(B \wedge J \wedge M \wedge \neg A \wedge E)$
- This last example shows a form of *inference*.

# Constructing a Bayesian network

- There are often several ways to construct a Bayesian network.
- The knowledge engineer needs to discover *conditional independence* relationships.
- Parents of a node should be those variables that directly influence its value.
  - JohnCalls is influenced by Earthquake, but not directly.
  - John and Mary calling don't influence each other.
- Formally, we believe that:

$$P(\text{MaryCalls} | \text{JohnCalls}, \text{Alarm}, \text{Earthquake}, \text{Burglary}) = P(\text{MaryCalls} | \text{Alarm})$$

# Compactness and Node Ordering

- Bayesian networks allow for a more compact representation of the domain
  - Redundant information is removed.
- Example: Say we have 30 nodes, each with 5 parents.
- Each CPT will contain  $2^5 = 32$  numbers Total: 960.
- Joint:  $2^{30}$  entries, nearly all redundant.

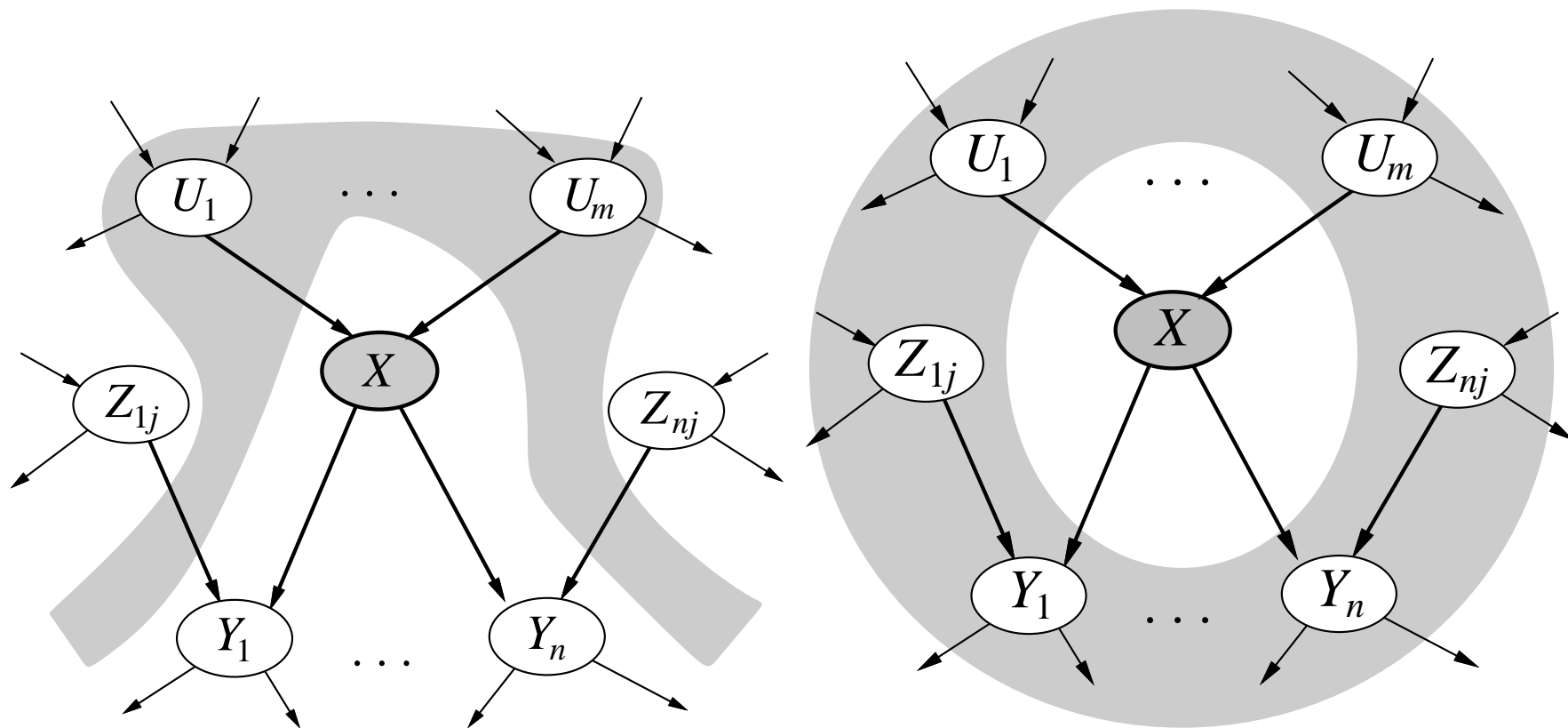
# Building a network

- Begin with root causes
- Then add the variables they directly influence.
- Then add their direct children.
- This is called a *causal model*
  - Reasoning in terms of cause and effect
- Estimates are much easier to come up with this way.
- We could try to build from effect to cause, but the network would be more complex, and the CPTs hard to estimate.  $P(E|B) = ?$

# Conditional Independence

- Recall that conditional independence means that two variables are independent of each other, given the observation of a third variable.
- $P(a \wedge b|c) = P(a|c)P(b|c)$
- A node is conditionally independent of its nondescendants, given its parents.
  - Given Alarm, JohnCalls is independent of Burglary and Earthquake.
- A node is conditionally independent of all other nodes, given its parents, children, and siblings (the children's other parents).
  - Burglary is independent of JohnCalls given Alarm and Earthquake.

# Conditional Independence



# Conditional Independence

- General rules:
  - Two nodes are conditionally independent, given their parents. (JohnCalls and MaryCalls, given Alarm)
  - A node is conditionally independent of non-descendants, given its parents. (JohnCalls is conditionally independent of Burglary given Alarm).
  - A node is conditionally independent of all other nodes, given its parents, children, and its children's other parents.

# Inference in Bayesian networks

- In most cases, we'll want to use a Bayesian network to tell us posterior probabilities.
- We observe a variable - how do other variables change?
- We can distinguish between *query variables* (things we want to know about) and *evidence variables* (things we can observe).
- There are also *hidden variables*, which influence query variables but are not directly observable.
- JohnCalls and MaryCalls are evidence, Earthquake and Burglary are queries, and Alarm is hidden.
- Updating in Bayesian networks can be quite complex - we'll skip over this.
  - For more information, see R & N

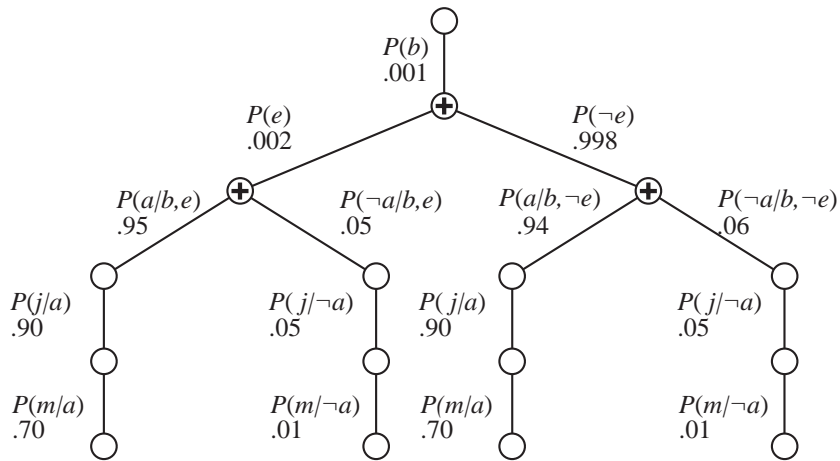
# Enumeration

- In some cases, reasoning in a Bayesian network is still very expensive.
- For example, we can rewrite the probability  $P(X|e) = \alpha P(X \wedge e) = \alpha \sum P(X \wedge e \wedge y)$  where  $y$  are hidden variables.
- This is equivalent to summing within the joint probability distribution.
- Consider  $P(\text{Burglary} | \text{JohnCalls}, \text{MaryCalls}) = \alpha P(\text{Burglary} \wedge \text{JohnCalls} \wedge \text{MaryCalls})$
- This is  $\alpha \sum_E \sum_A P(B, E, A, J, M)$  For  $B = \text{true}$ , we must calculate:  $P(B|J, M) = \alpha \sum_E \sum_A P(B)P(E)P(A|B, E)P(J|A)P(M|A)$

# Enumeration

- Problem- a network with  $n$  nodes will require  $O(n2^n)$  computations.
- We can simplify by bringing the priors outside the summation.
- $\alpha P(B) \sum_E P(E) \sum_A P(A|B, E) P(J|A) P(M|A)$
- Still requires  $O(2^n)$  calculations

# Enumeration



- This tree shows the computations needed to determine  $P(B|J, M)$
- Notice that there are lots of repeated computations in here.
- By eliminating repeated computations, we can speed things up.

# Variable Elimination

- We can reduce the number of calculations by caching results and reusing them.
- Evaluate the expression from right-to-left.
- Summations are done only for nodes influenced by the variable being summed.

- Consider

$$P(B|j, m) = \alpha \underbrace{P(B)}_B \sum_e \underbrace{P(e)}_E \sum_a \underbrace{P(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M$$

- The equation has been separated into factors.
- Once we compute  $P(M|a)$  and  $P(M|\neg a)$ , we cache those results in a matrix  $f_M$ .
- Similarly for  $P(J|a)$  and  $P(J|\neg a)$  stored in  $f_J$

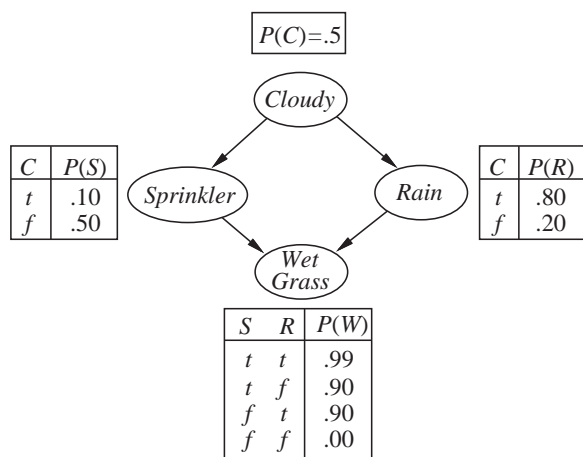
# Variable Elimination

- $P(A|B, E)$  will result in a  $2 \times 2 \times 2$  matrix, stored in  $F_A$ .
- We then need to compute the sum over the different possible values of  $A$
- This sum is  $\sum_a f_A(A, B, E) f_J(A) f_M(A)$
- We can process  $E$  the same way.
- In essence, we're doing dynamic programming here, and exploiting the same memoization process.
- Complexity
  - Polytrees: (one undirected path between any two nodes) - linear in the number of nodes.
  - Multiply-connected nodes: Exponential.

# Scaling Up

- Many techniques have been developed to allow Bayesian networks to scale to hundreds of nodes.
- Clustering
  - Nodes are joined together to make the network into a polytree.
  - CPT within the node grows, but network structure is simplified.
- Approximating inference
  - Monte Carlo sampling is used to estimate conditional probabilities.

# Monte Carlo sampling example



- Start at the top of the network and select a random sample. (say it's *true*).
- Draw a random sample from its children. conditioned on *true*.
  - $P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle$ . Say we select *False*
  - $P(\text{Rain} | \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle$ . Say we select *True*
  - $P(\text{WetGrass} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = \langle 0.9, 0.1 \rangle$ . Say we select *true*.
- This gives us a sample for  $\langle \text{cloudy}, \neg \text{Sprinkler}, \text{Rain}, \text{WetGrass} \rangle$
- As we increase the number of samples, this provides an estimate of  $P(\text{cloudy}, \neg \text{Sprinkler}, \text{Rain}, \text{WetGrass})$ .
- We choose the query we are interested in and then sample the network “enough” times to determine the probability of that event occurring,

# Applications of Bayesian Networks

- Diagnosis (widely used in Microsoft's products)
- Medical diagnosis
- Spam filtering
- Expert systems applications (plant control, monitoring)
- Robotic control