Lecture 23: 28 April, 2022

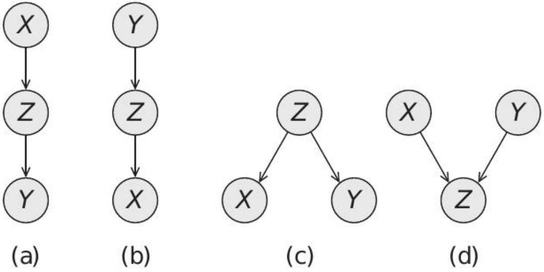
Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
January–May 2022

D-Separation

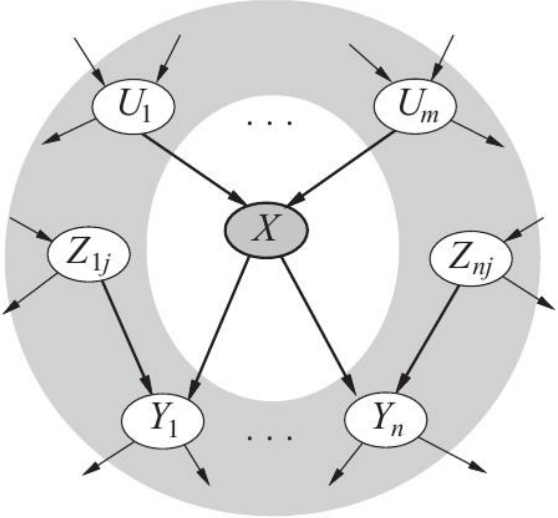
- Check if $X \perp Y \mid Z$
- Dependence should be blocked on every trail from X to Y
  - Each undirected path from X to Y is a sequence of basic trails
  - For (a), (b), (c), need Z present
  - For (d), need Z absent
  - In general, V-structure includes descendants of the bottom node



- x and y are D-separated given z if all trails are blocked
- Variation of breadth first search (BFS) to check if y is reachable from x through some trail
- Extends to sets — each $x \in X$ is D-separated from each $y \in Y$

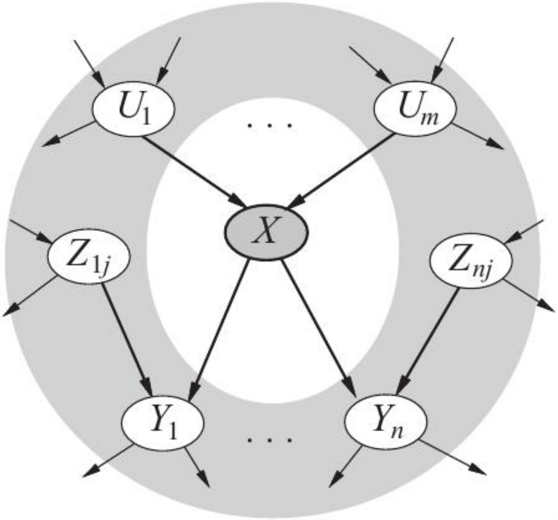
Markov blanket

- $MB(X)$ — Markov blanket of X



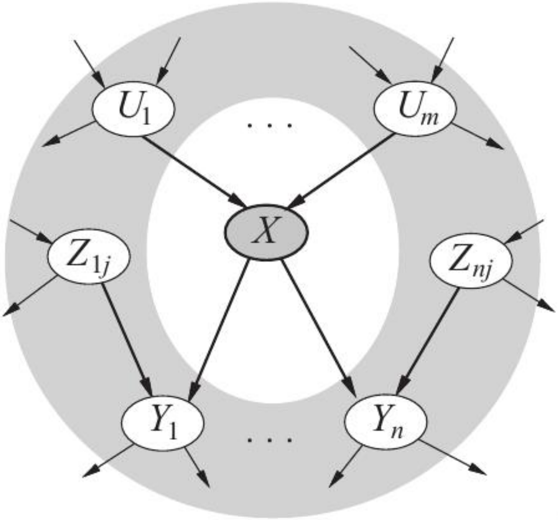
Markov blanket

- $MB(X)$ — Markov blanket of X
  - $Parents(X)$



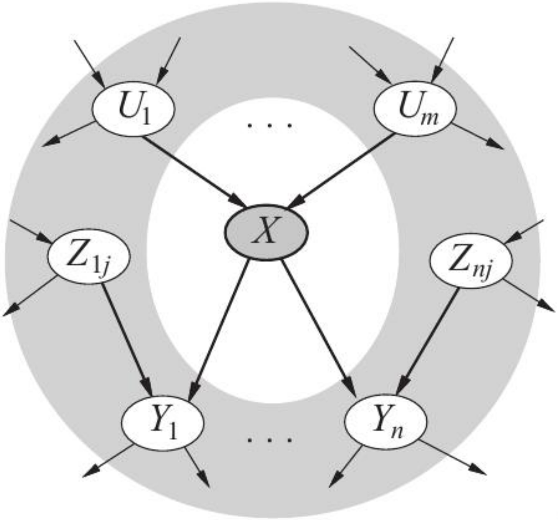
Markov blanket

- $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$



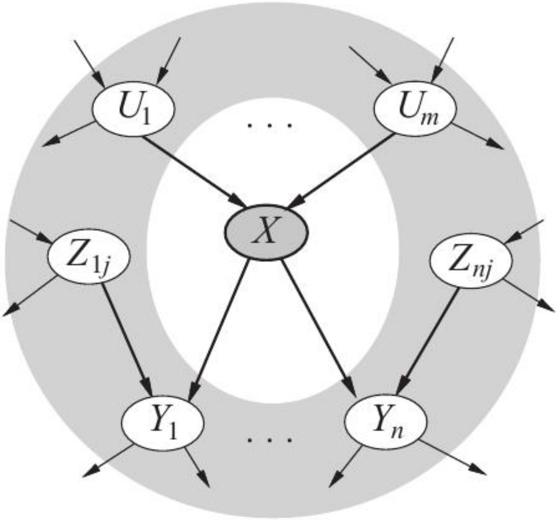
Markov blanket

- $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$



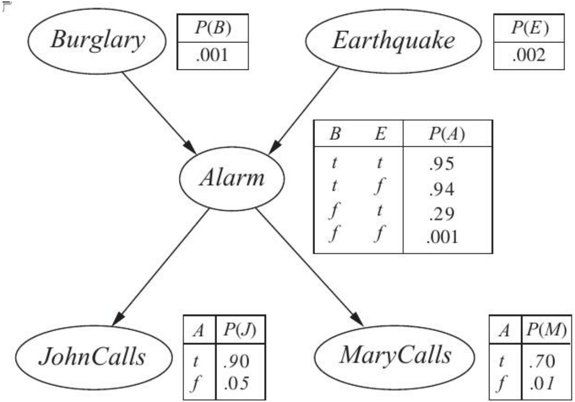
Markov blanket

- $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$



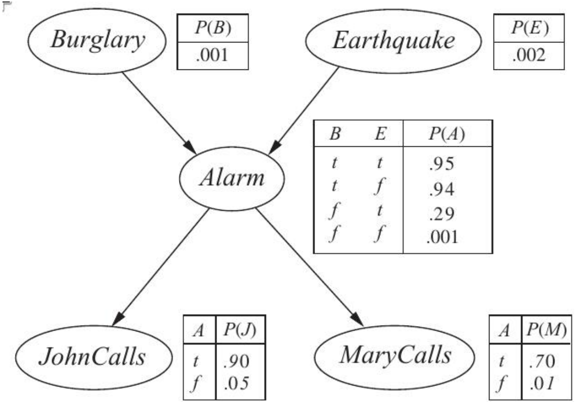
Computing with probabilistic graphical models

- John and Mary call Pearl. What is the probability that there has been a burglary?

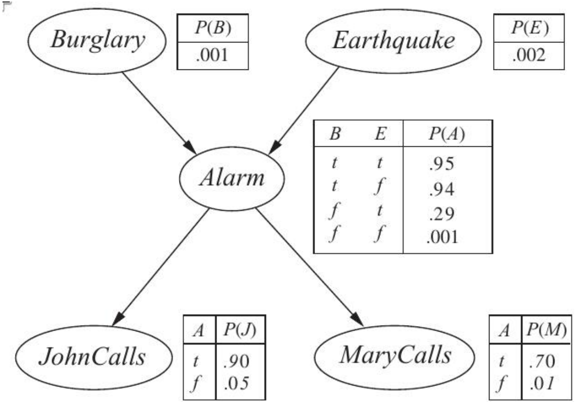


Computing with probabilistic graphical models

- John and Mary call Pearl. What is the probability that there has been a burglary?
- Want $P(b \mid m, j)$

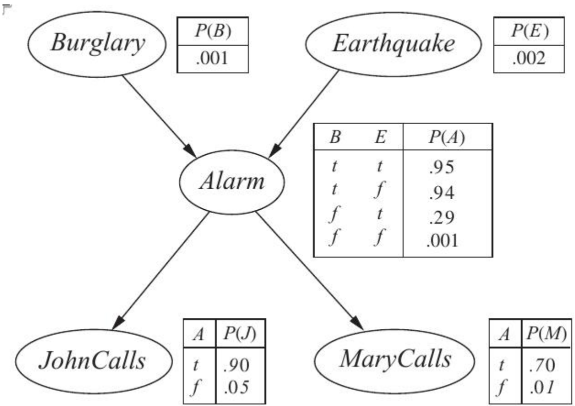


- John and Mary call Pearl. What is the probability that there has been a burglary?
- Want $P(b \mid m, j)$
- $\dfrac{P(b, m, j)}{P(m, j)}$



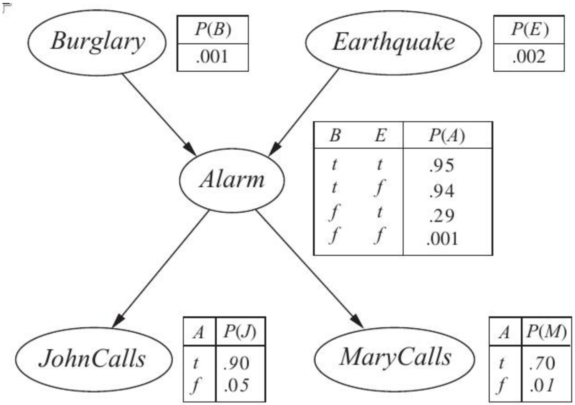
Computing with probabilistic graphical models

- John and Mary call Pearl. What is the probability that there has been a burglary?
- Want $P(b \mid m, j)$
- $\dfrac{P(b, m, j)}{P(m, j)}$
- Use chain rule to evaluate joint probabilities



Computing with probabilistic graphical models

- John and Mary call Pearl. What is the probability that there has been a burglary?
- Want $P(b \mid m, j)$
- $\dfrac{P(b, m, j)}{P(m, j)}$
- Use chain rule to evaluate joint probabilities
- Reorder variables appropriately, topological order of graph



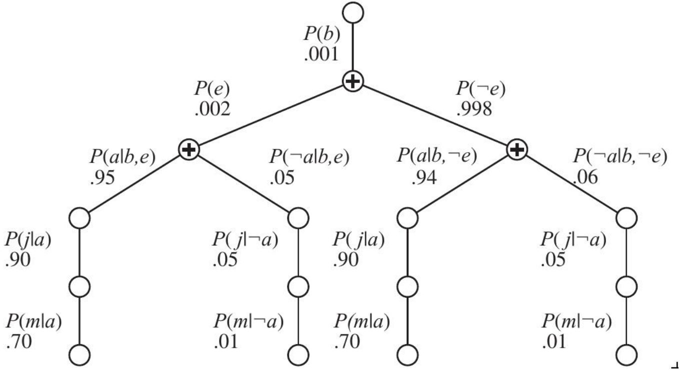
Computing with probabilistic graphical models

- $P(m, j, b) = P(b) \sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(a \mid b, e) P(m \mid a) P(j \mid a)$

Computing with probabilistic graphical models

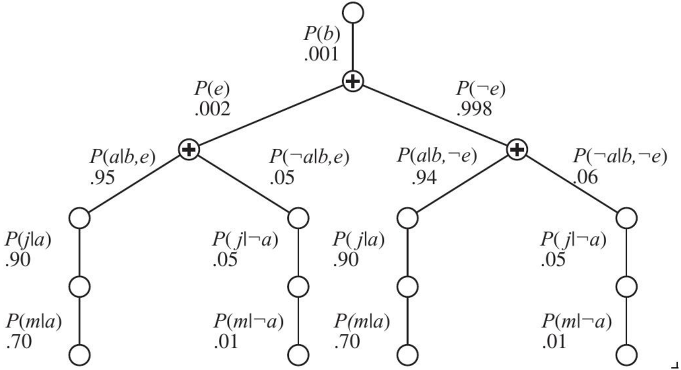
- $P(m, j, b) = P(b) \sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(a \mid b, e) P(m \mid a) P(j \mid a)$

- Construct the computation tree



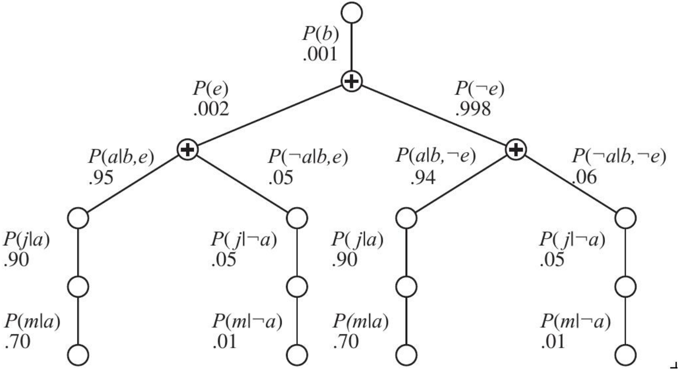
Computing with probabilistic graphical models

- $P(m, j, b) = P(b) \sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(a \mid b, e) P(m \mid a) P(j \mid a)$
- Construct the computation tree
- Use dynamic programming to avoid duplicated computations



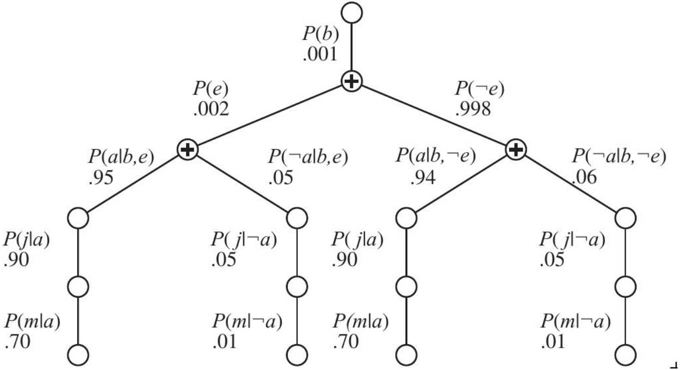
Computing with probabilistic graphical models

- $P(m,j,b) = P(b)\sum_{e=0}^{1} P(e)\sum_{a=0}^{1} P(a \mid b,e)P(m \mid a)P(j \mid a)$
- Construct the computation tree
- Use dynamic programming to avoid duplicated computations
- However, exact inference is NP-complete, in general



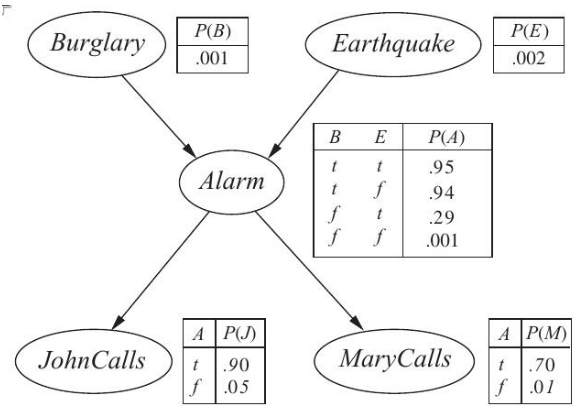
Computing with probabilistic graphical models

- $P(m, j, b) = P(b) \sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(a \mid b, e) P(m \mid a) P(j \mid a)$
- Construct the computation tree
- Use dynamic programming to avoid duplicated computations
- However, exact inference is NP-complete, in general
- Instead, approximate inference through sampling



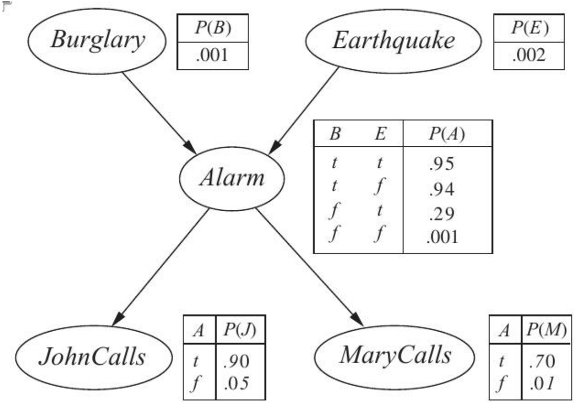
Approximate inference

- Generate random samples (b, e, a, m, j), count to estimate probabilities



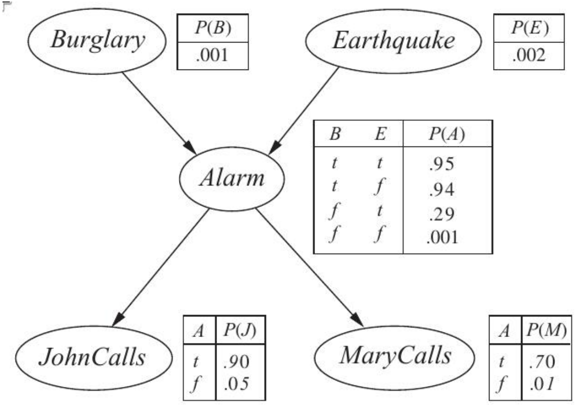
Approximate inference

- Generate random samples (b, e, a, m, j), count to estimate probabilities
- Random samples should respect conditional probabilities



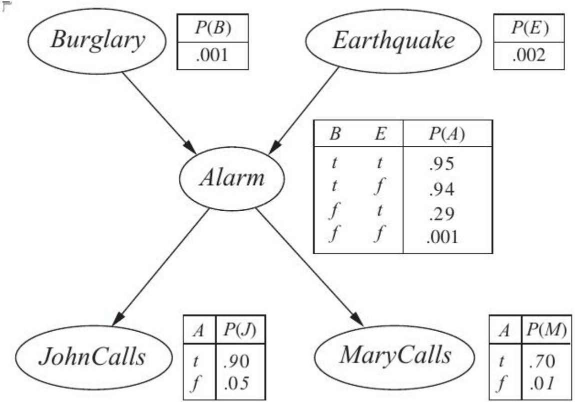
Approximate inference

- Generate random samples (b, e, a, m, j), count to estimate probabilities
- Random samples should respect conditional probabilities
- Fix $MB(x)$ before generating x

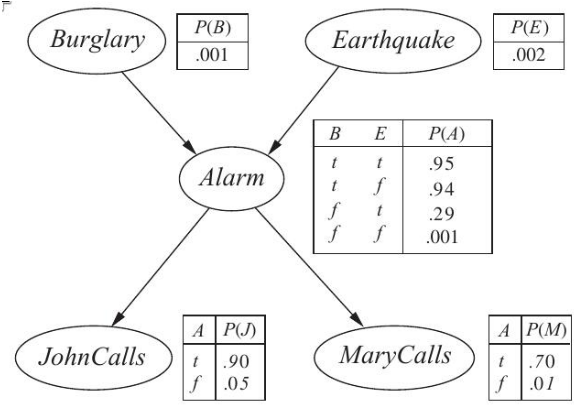


Approximate inference

- Generate random samples (b, e, a, m, j), count to estimate probabilities
- Random samples should respect conditional probabilities
- Fix $MB(x)$ before generating x
- Generate in topological order
  - Generate b, e with probabilities $P(b)$ and $P(e)$
  - Generate a with probability $P(a \mid b, e)$
  - Generate j, m with probabilities $P(j \mid a)$, $P(m \mid a)$

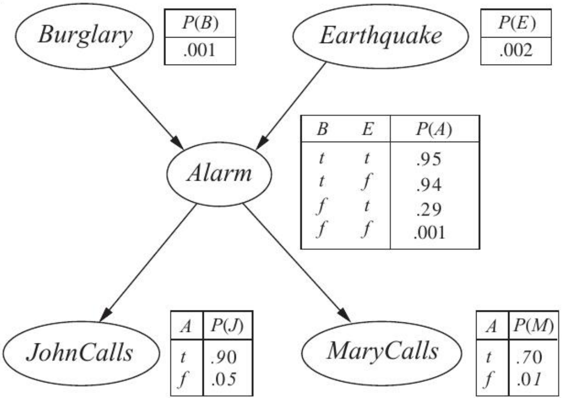


- We are interested in $P(b \mid j, m)$



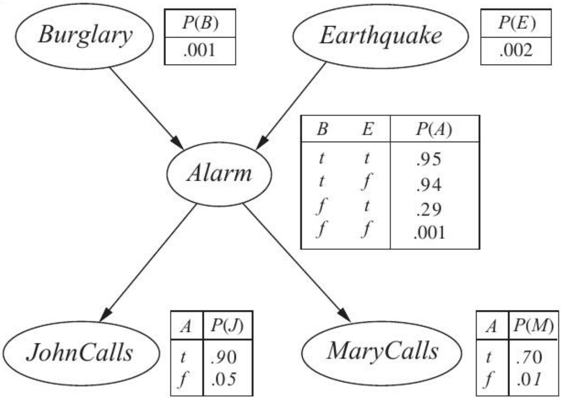
Approximate inference

- We are interested in $P(b \mid j, m)$
- Samples with $\neg j$ or $\neg m$ are useless



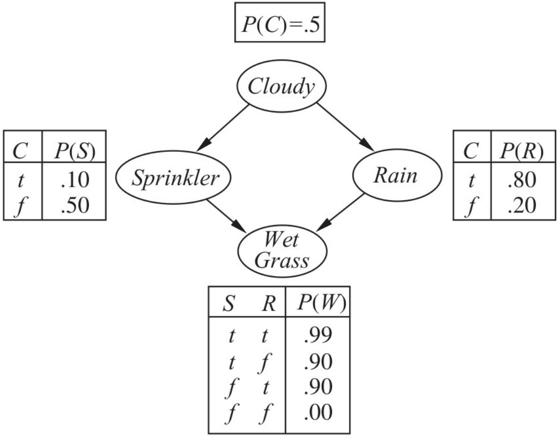
Approximate inference

- We are interested in $P(b \mid j, m)$
- Samples with $\neg j$ or $\neg m$ are useless
- Can we sample more efficiently?



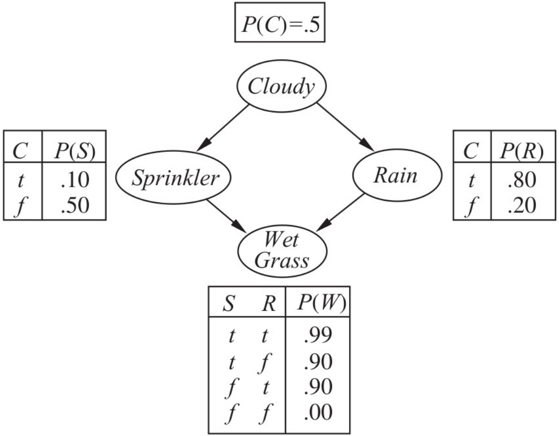
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$



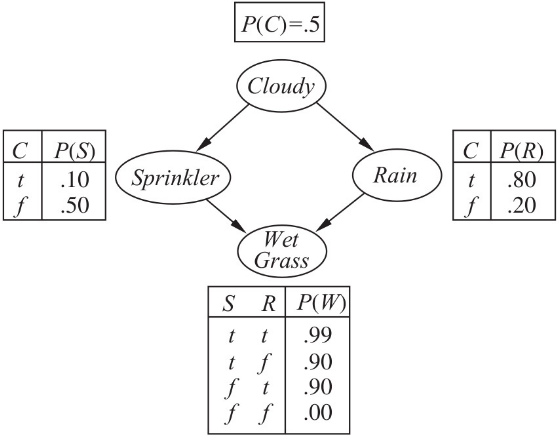
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Topological order
  - Generate *Cloudy*
  - Generate *Sprinkler*, *Rain*
  - Generate *Wet Grass*



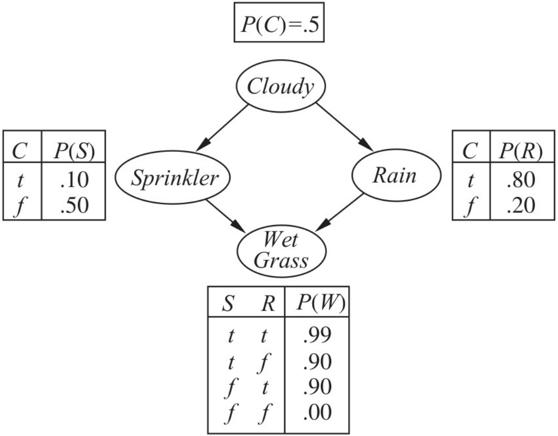
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Topological order
  - Generate *Cloudy*
  - Generate *Sprinkler*, *Rain*
  - Generate *Wet Grass*
- If we start with $\neg Cloudy$, sample is useless



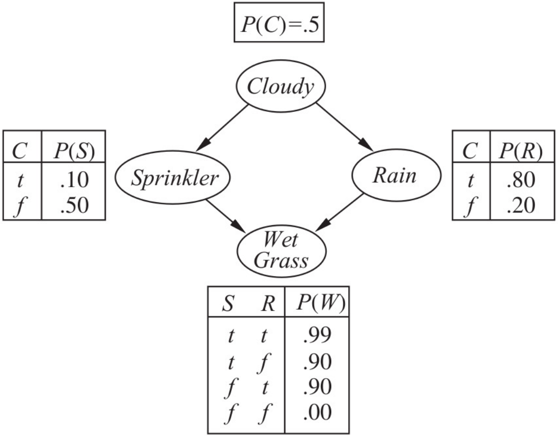
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Topological order
  - Generate *Cloudy*
  - Generate *Sprinkler*, *Rain*
  - Generate *Wet Grass*
- If we start with $\neg Cloudy$, sample is useless
- Immediately stop and reject this sample — rejection sampling



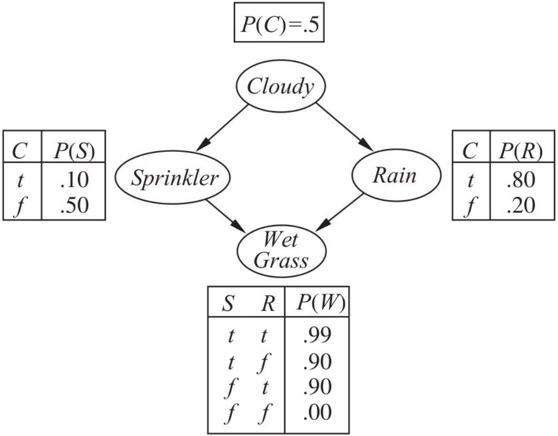
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Topological order
  - Generate *Cloudy*
  - Generate *Sprinkler*, *Rain*
  - Generate *Wet Grass*
- If we start with $\neg Cloudy$, sample is useless
- Immediately stop and reject this sample — rejection sampling
- General problem with low probability situation — lots of samples



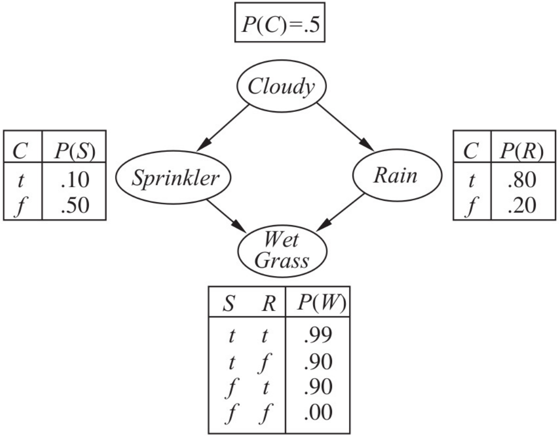
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$



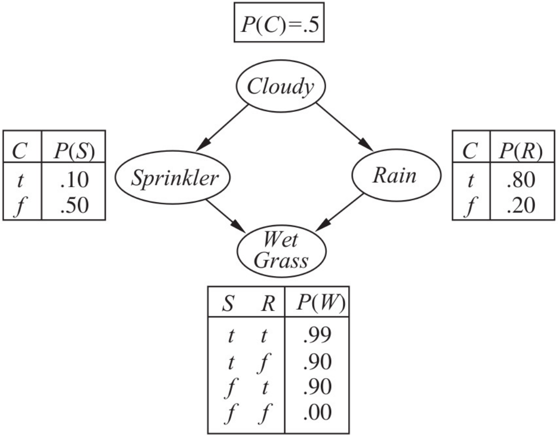
Likelihood weighted sampling

- *P*(*Rain* | *Cloudy*, *Wet Grass*)
- Fix evidence *Cloudy*, *Wet Grass* true



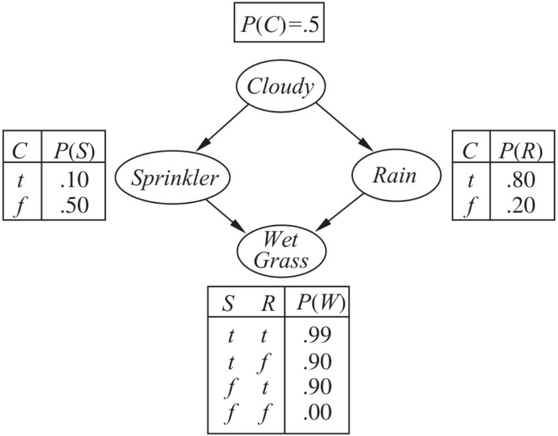
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables



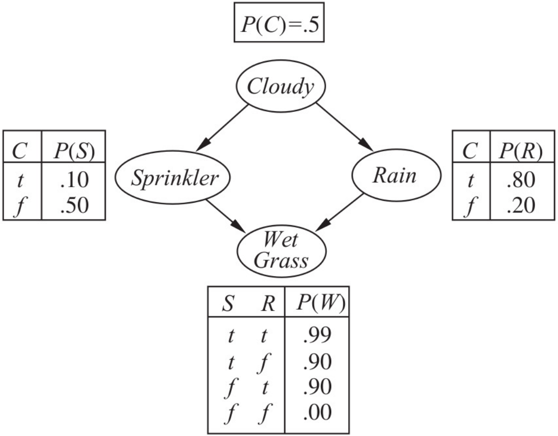
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Suppose we generate $c, \neg s, r, w$



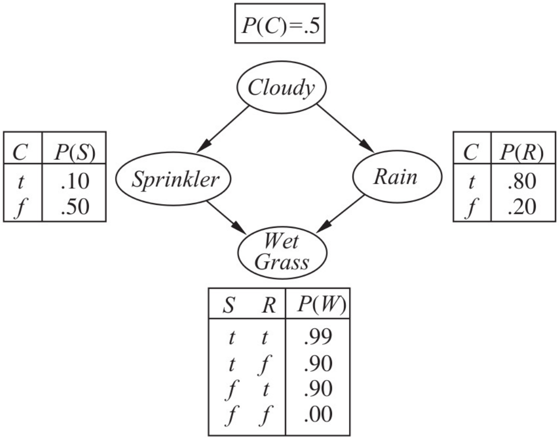
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Suppose we generate $c, \neg s, r, w$
- Compute likelihood of evidence:
  $0.5 \times 0.9 = 0.45$



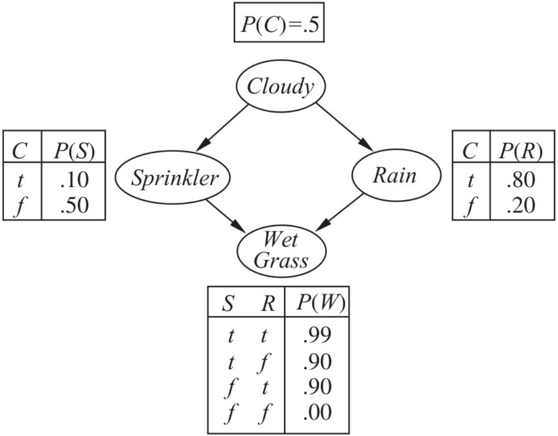
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Suppose we generate $c, \neg s, r, w$
- Compute likelihood of evidence:
  $0.5 \times 0.9 = 0.45$
- 0.45 is likelihood weight of sample



Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Suppose we generate $c, \neg s, r, w$
- Compute likelihood of evidence:
  $0.5 \times 0.9 = 0.45$
- 0.45 is likelihood weight of sample
- Samples $s_1, s_2, \ldots, s_N$ with weights
  $w_1, w_2, \ldots w_N$



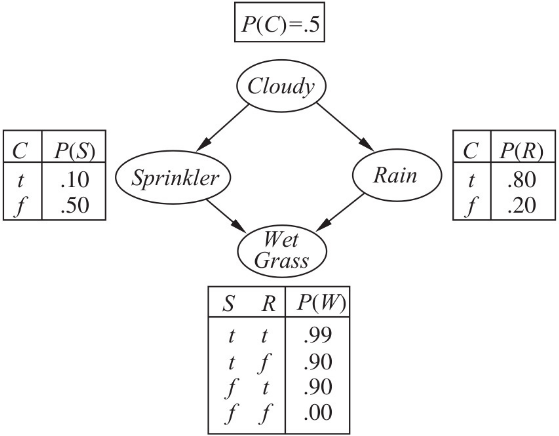
C	P(S)
t	.10
f	.50

C	P(R)
t	.80
f	.20

S	R	P(W)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Suppose we generate $c, \neg s, r, w$
- Compute likelihood of evidence:
  $0.5 \times 0.9 = 0.45$
- 0.45 is likelihood weight of sample
- Samples $s_1, s_2, \ldots, s_N$ with weights
  $w_1, w_2, \ldots w_N$
- $P(r \mid c, w) = \dfrac{\sum_{s_i\ has\ rain} w_i}{\sum_{1 \leq j \leq N} w_j}$



Markov chains