Introduction to AI

Chapter14.4-14.5

Inference in Bayesian Networks

Pengju Ren@IAIR

Outline

- Exact inference by enumeration
- Exact inference by variable elimination
- Approximate inference by stochastic simulation
- Approximate inference by Markov chain Monte Carlo

Basics

- **Query variables : X**
- **Evidence variable : E**
- **Hidden variable : Y (not evidence nor query)**
- **Posterior probability distribution : P(X|e)**

Inference tasks

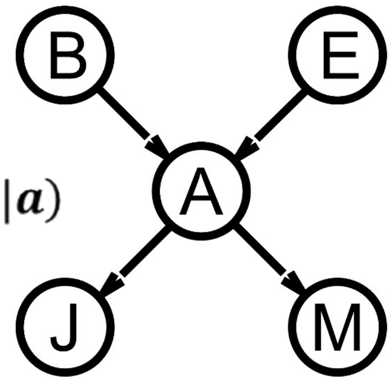
- **Simple queries:** compute posterior marginal $P(X_i|E=e)$
  e.g. $P(NoGas|Gauge = empty, Lights = on, Starts = false)$
- **Conjunctive queries:** $P(X_i, X_j|E=e) = P(X_i|E=e)P(X_j|X_i, E=e)$
- **Optimal decisions:** decision networks include utility information; probabilistic inference required for $P(outcome|action, evidence)$
- **Value of information:** which evidence to seek next?
- **Sensitivity analysis:** which probability values are most critical?
- **Explanation:** why do I need a new starter motor?

Inference by enumeration

- Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation.
- Simple query on the burglary network:

$$P(X|e) = \alpha P(X,e) = \alpha \sum_{y} P(X,e,y)$$

$$P(B|j,m) = P(B,j,m)|P(j,m) = \alpha P(B,j,m) = \alpha \sum_{e} \sum_{a} P(B,e,a,j,m)$$



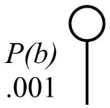
Rewrite full joint entries using product of CPT entries:

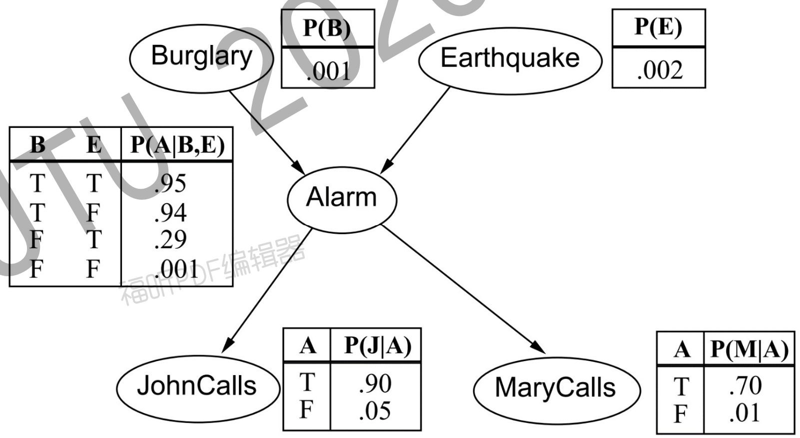
$$P(B|j,m) = \alpha \sum_{e} \sum_{a} P(B)P(e)P(a|B,e)P(j|a)P(m|a)$$

$$= \alpha P(B) \sum_{e} P(e) \sum_{a} P(a|B,e)P(j|a)P(m|a)$$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Enumeration tree





$$P(B|j,m) = \alpha \sum_e \sum_a P(B)P(e)P(a|B,e)P(j|a)P(m|a)$$

$$= \alpha P(B) \sum_e P(e) \sum_a P(a|B,e)P(j|a)P(m|a)$$

Enumeration is inefficient: repeated computation

e.g., computes *P(j|a)P(m|a) and P(j|¬a)P(m|¬a)* **for each value of e**

Enumeration algorithm

function ENUMERATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    extend $\mathbf{e}$ with value x_i for X
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(VARS[bn], $\mathbf{e}$)
  return NORMALIZE($\mathbf{Q}(X)$)

---

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in $\mathbf{e}$
    then return $P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\Sigma_y\ P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is $\mathbf{e}$ extended with $Y = y$

Inference by variable elimination

- **Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation**

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \quad \mathbf{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

$$\begin{aligned} \mathbf{f}_6(B, E) &= \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) \\ &= (\mathbf{f}_3(a, B, E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a, B, E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a)) \end{aligned}$$

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E)$$

Inference by variable elimination

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E)$$

Sum out e from the product of f2 and f6:

$$\begin{aligned} \mathbf{f}_7(B) &= \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) \\ &= \mathbf{f}_2(e) \times \mathbf{f}_6(B, e) + \mathbf{f}_2(\neg e) \times \mathbf{f}_6(B, \neg e) \end{aligned}$$

Therefore: $\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \mathbf{f}_7(B)$

Variable elimination: Basic operations

- **Summing out** a variable from a product of factors:
  move any constant factors outside the summation
  add up submatrices in pointwise product of remaining factors

$$\sum_x f_1 \times \ldots \times f_k = f_1 \times \ldots \times f_i \sum_x f_{i+1} \times \ldots \times f_k = f_1 \times \ldots \times f_{\bar{X}}$$

  Assume $f_1 \times \ldots \times f_i$ do not depend on X

- **Pointwise produce** of factors f_1 and f_2 :

$$\begin{aligned} & f_1(x_1,\ldots,x_j,y_1,\ldots,y_k) \times f_2(y_1,\ldots,y_k,z_1,\ldots,z_l) \\ & = f(x_1,\ldots,x_j,y_1,\ldots,y_k,z_1,\ldots,z_l) \end{aligned}$$

E.g., $f_1(a,b) \times f_2(b,c) = f(a,b,c)$

pointwise multiplication

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

$$\begin{aligned}\mathbf{f}(B,C) &= \sum_a \mathbf{f}_3(A,B,C) = \mathbf{f}_3(a,B,C) + \mathbf{f}_3(\neg a,B,C) \\ &= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix} = \begin{pmatrix} .24 & .96 \\ .48 & .32 \end{pmatrix}.\end{aligned}$$

Variable elimination algorithm

function ELIMINATION-ASK(X, $\mathbf{e}$, bn) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, evidence specified as an event
    bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

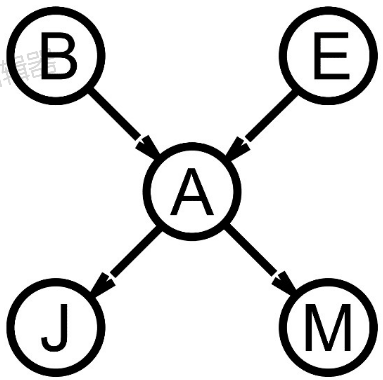
  $factors \leftarrow [\,]$; $vars \leftarrow$ REVERSE(VARS[bn])
  for each var **in** $vars$ **do**
    $factors \leftarrow$ [MAKE-FACTOR(var, $\mathbf{e}$)|$factors$]
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT(var, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Irrelevant variable

- **Consider the query** *P(JohnCalls|Burglary = true)*

$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(J|a) \sum_m P(m|a)$$

Sum over *m* **is identically 1;** *M* **is irrelevant to the query**



Thm 1: *Y* **is irrelevant unless** *Y* $\in$ *Ancestors({X}* $\cup$ ***E****)*
Here, *X* = *JohnCalls*, **E**= *{Burglary}*, **and**
Ancestors({X} $\cup$ *E) = {Alarm, Earthquake}*
so *MaryCalls* **is irrelevant**
(Compare this to backward chaining from the query in Horn clause KBs)

Every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query

Inference by stochastic simulation

- **Basic idea:**
  1) Draw *N* samples from a sampling distribution S
  2) Compute an approximate posterior probability $\hat{P}$
  3) Show this converges to the true probability P

- **Outline:**
  – **Direct sampling:** Sampling from an empty network
  – **Rejection sampling:** reject samples disagreeing with evidence
  – **Likelihood weighting:** use evidence to weight samples
  – **Markov chain Monte Carlo** (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

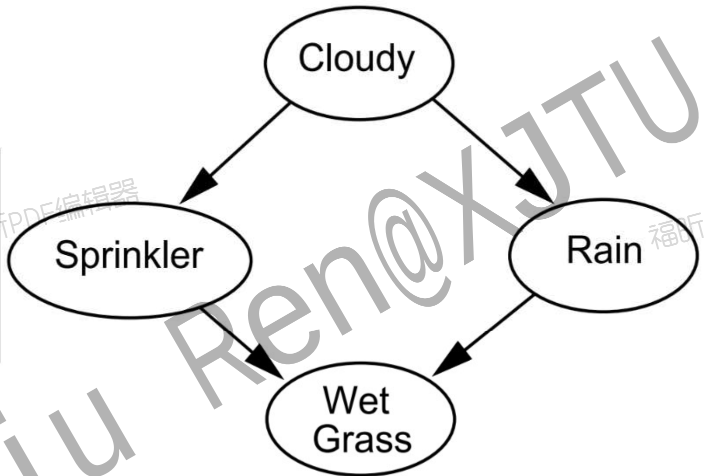
Sampling from an empty network

function PRIOR-SAMPLE(bn) **returns** an event sampled from bn
  inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for $i = 1$ **to** n **do**
    $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
      given the values of $Parents(X_i)$ in $\mathbf{x}$
  return $\mathbf{x}$

Example

P(C)
.50

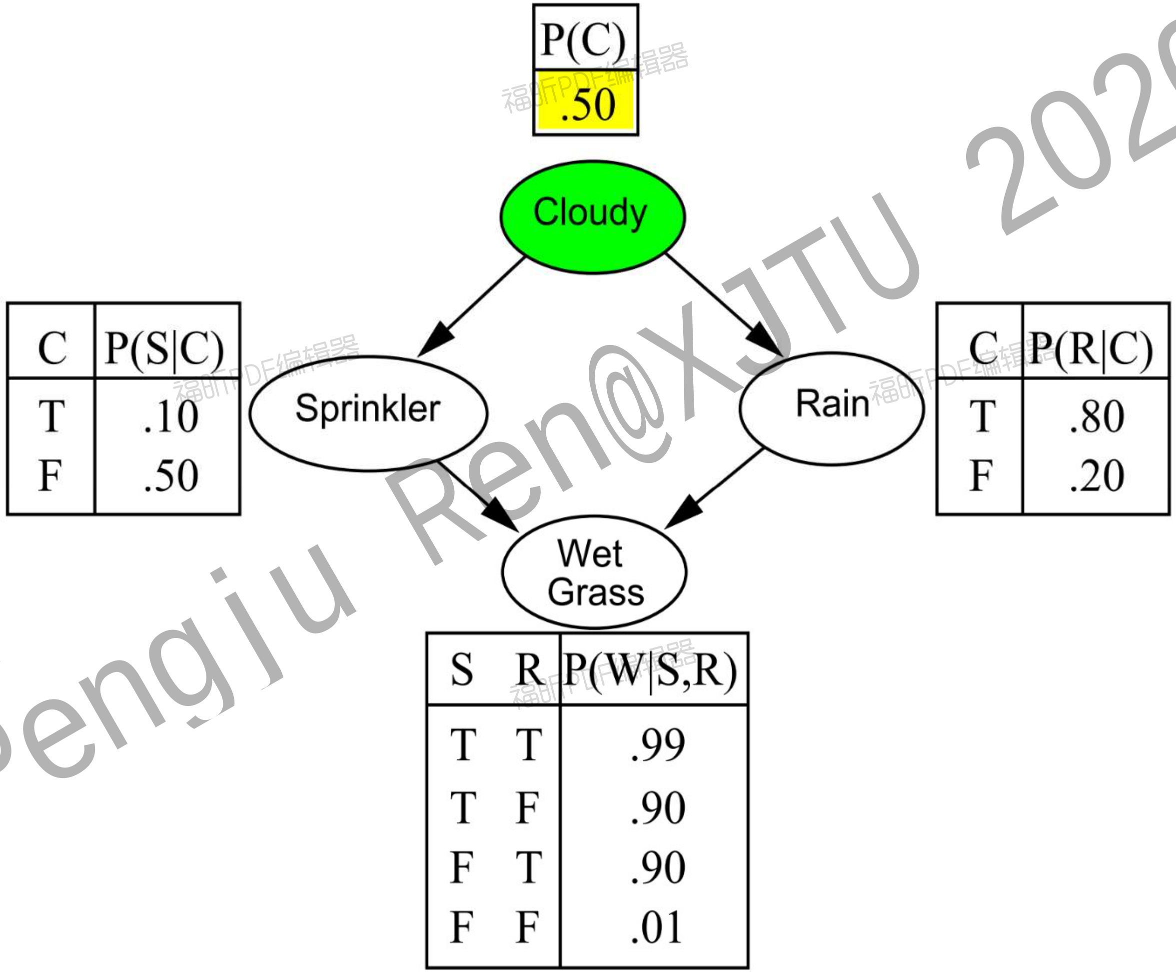


C	P(S|C)
T	.10
F	.50

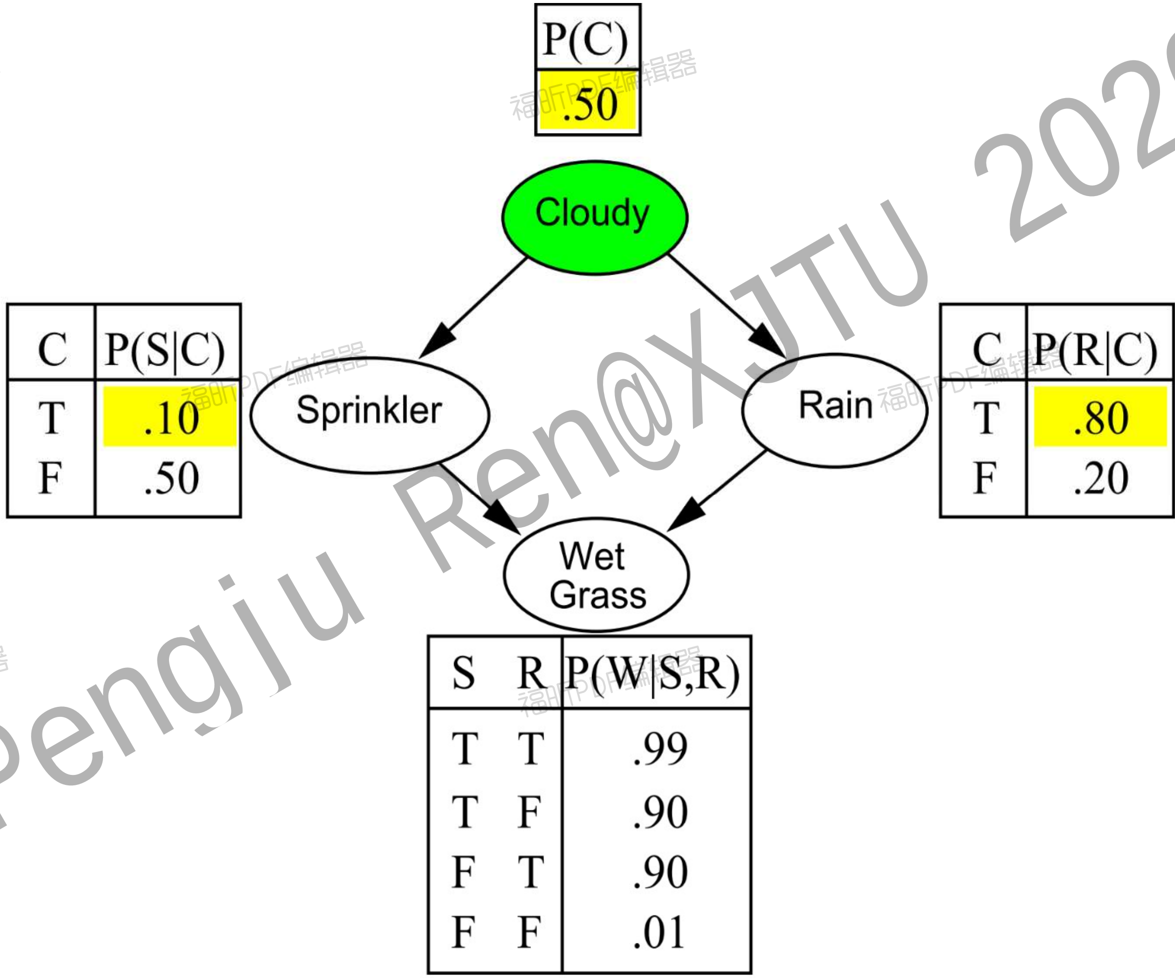
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

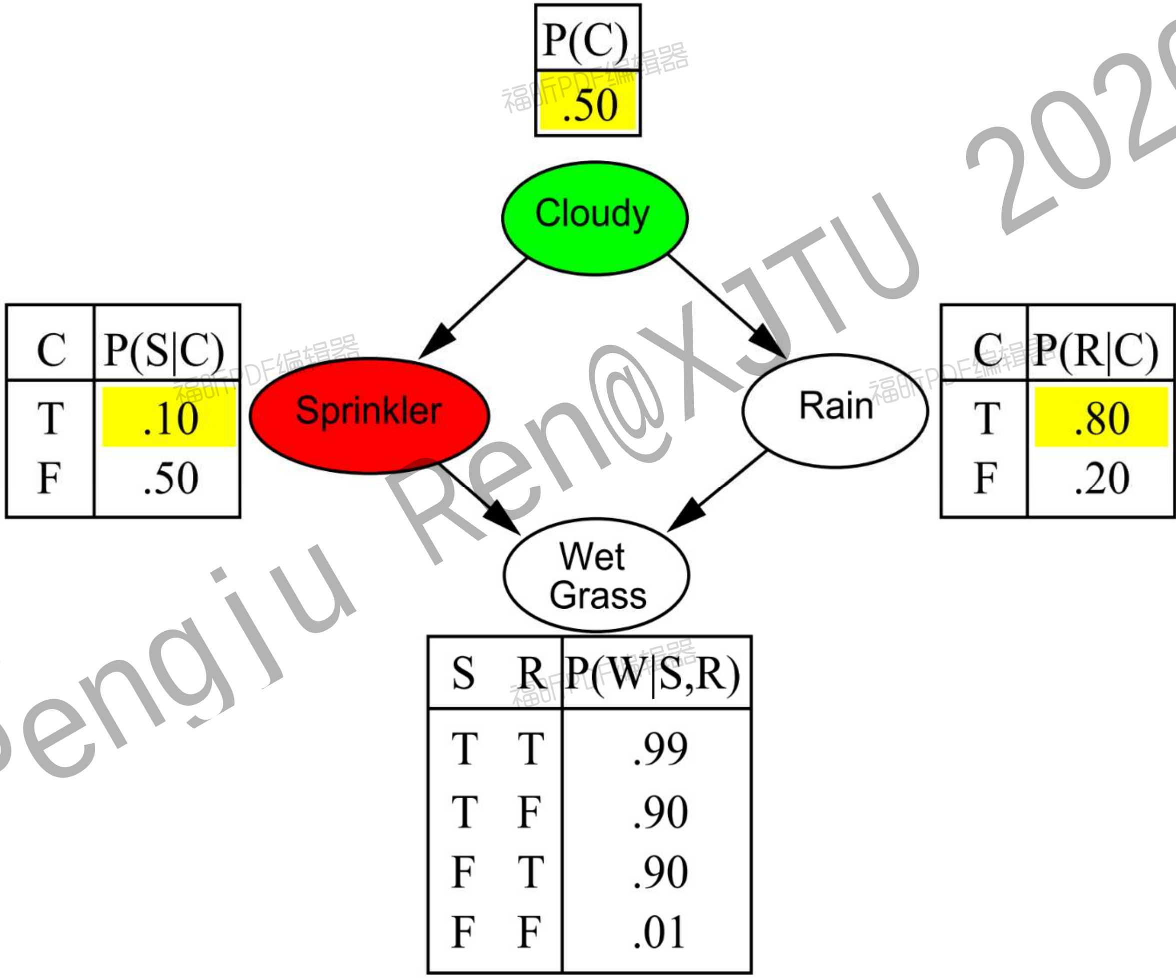
Inference tasks



Inference tasks



Inference tasks



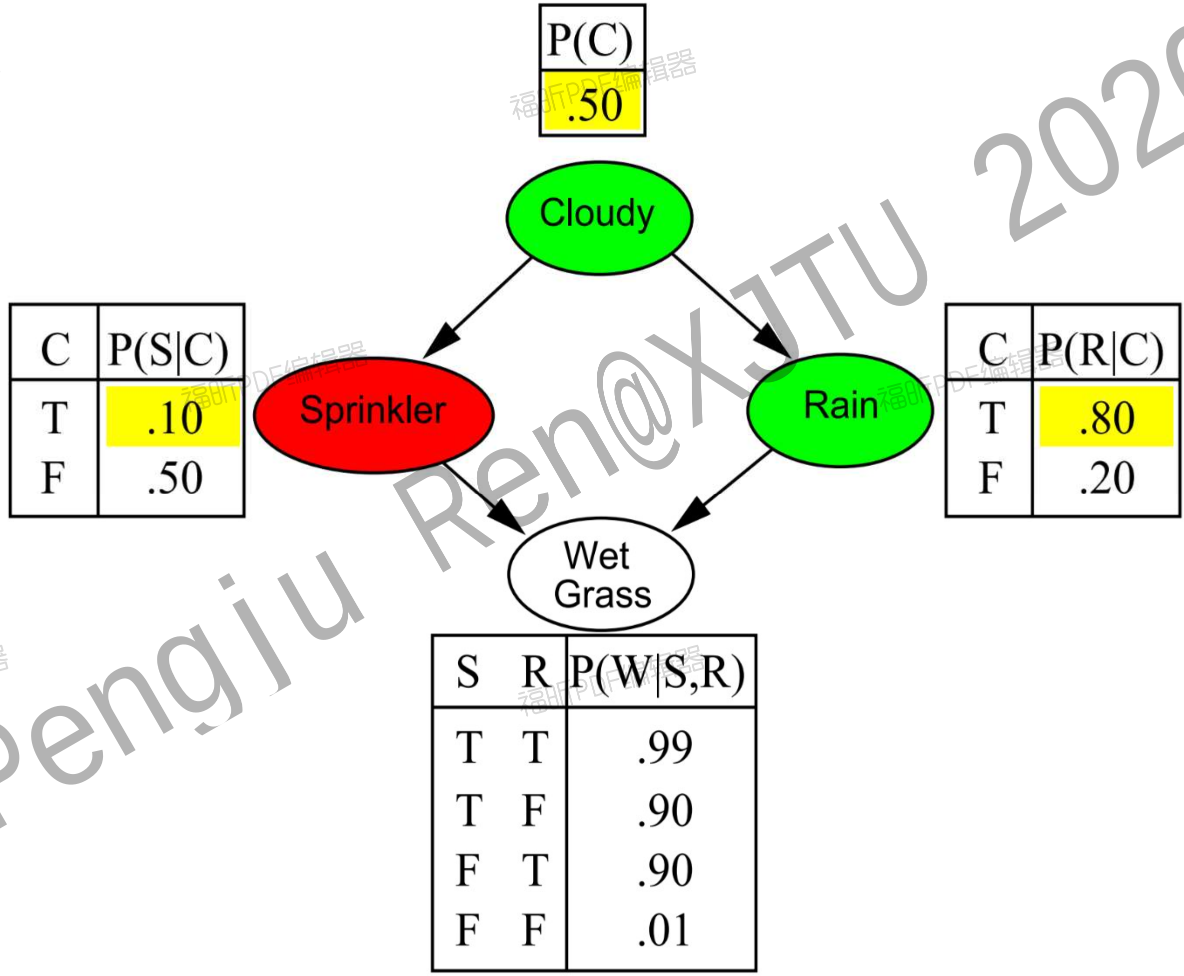
P(C)
.50

C	P(S|C)
T	.10
F	.50

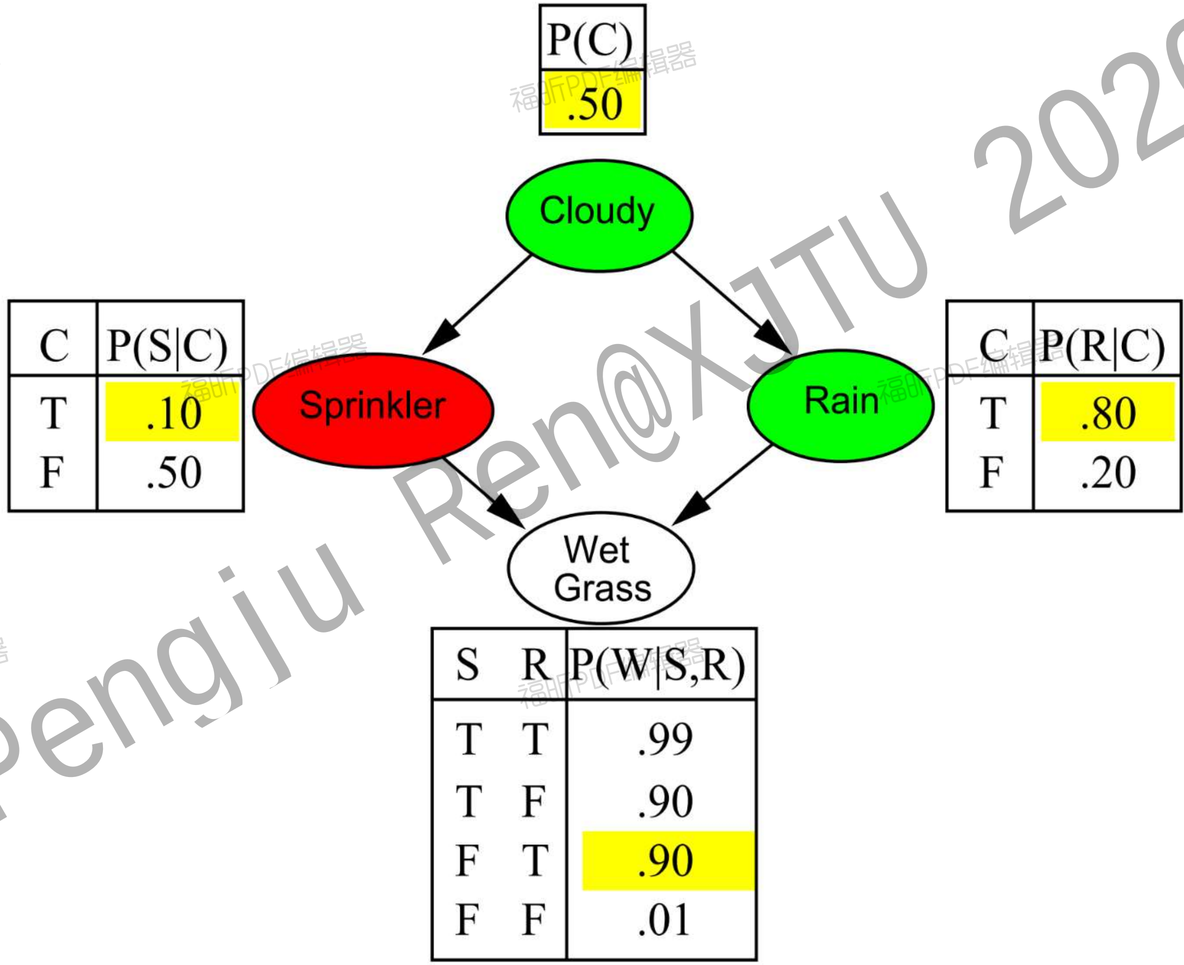
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

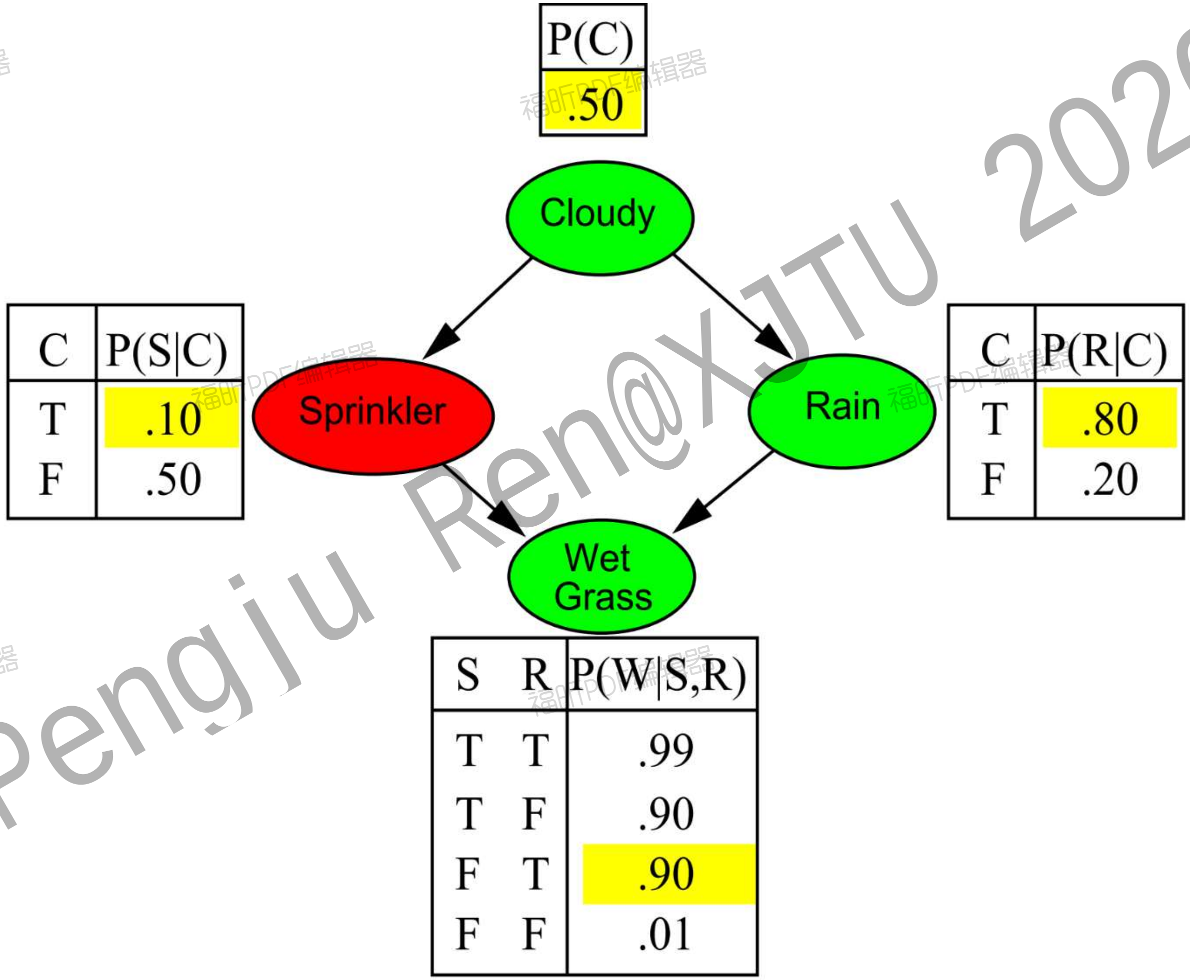
Inference tasks



Inference tasks



Inference tasks



Sampling from an empty network contd.

- Probability that *PRIORSAMPLE* generates a particular event

$$S_{PS}(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i)) = P(x_1,\ldots,x_n)$$

i.e., the true prior probability

E.g., $S_{PS}(t,f,t,t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t,f,t,t)$

- Let $N_{PS}(x_1,\ldots,x_n)$ be the number of samples generated for event $x_1,\ldots,x_n$ Then we have

$$\lim_{N\to\infty} \hat{P}(x_1,\ldots,x_n) = \lim_{N\to\infty} N_{PS}(x_1,\ldots,x_n)/N$$
$$= S_{PS}(x_1,\ldots,x_n) = P(x_1,\ldots,x_n)$$

- That is, estimates derived from *PRIORSAMPLE* are consistent

Shorthand: $\hat{P}(x_1,\ldots,x_n) \approx P(x_1,\ldots,x_n)$

Rejection sampling

estimated from samples agreeing with e

```
function REJECTION-SAMPLING(X, e, bn, N) returns an estimate of P(X|e)
   local variables: N, a vector of counts over X, initially zero

   for j = 1 to N do
       x ← PRIOR-SAMPLE(bn)
       if x is consistent with e then
           N[x] ← N[x]+1 where x is the value of X in x
   return NORMALIZE(N[X])
```

E.g., estimate *P(Rain|Sprinkler = true)* **using 100 samples**

27 samples have *Sprinkler = true*

Of these, 8 have *Rain = true* **and 19 have** *Rain = false*.

$$\hat{P}(Rain|Spinkler = true) = NORMALIZE(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$$

Similar to a basic real-world empirical estimation procedure

Analysis of rejection sampling

$$\hat{P}(X|e) = \alpha N_{PS}(X,e)$$ **(algorithm defn.)**

$$= N_{PS}(X,e)/N_{PS}(e)$$ **(normalized by $N_{PS}(e)$)**

$$\approx P(X,e)/P(e)$$ **(property of *PRIORSAMPLE*)**

$$= P(X|e)$$ **(defn. of conditional probability)**

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if *P(e)* is small

***P(e)* drops off exponentially with number of evidence variables**

Likelihood weighting

Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

function LIKELIHOOD-WEIGHTING(X, $\mathbf{e}$, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{W}$, a vector of weighted counts over X, initially zero

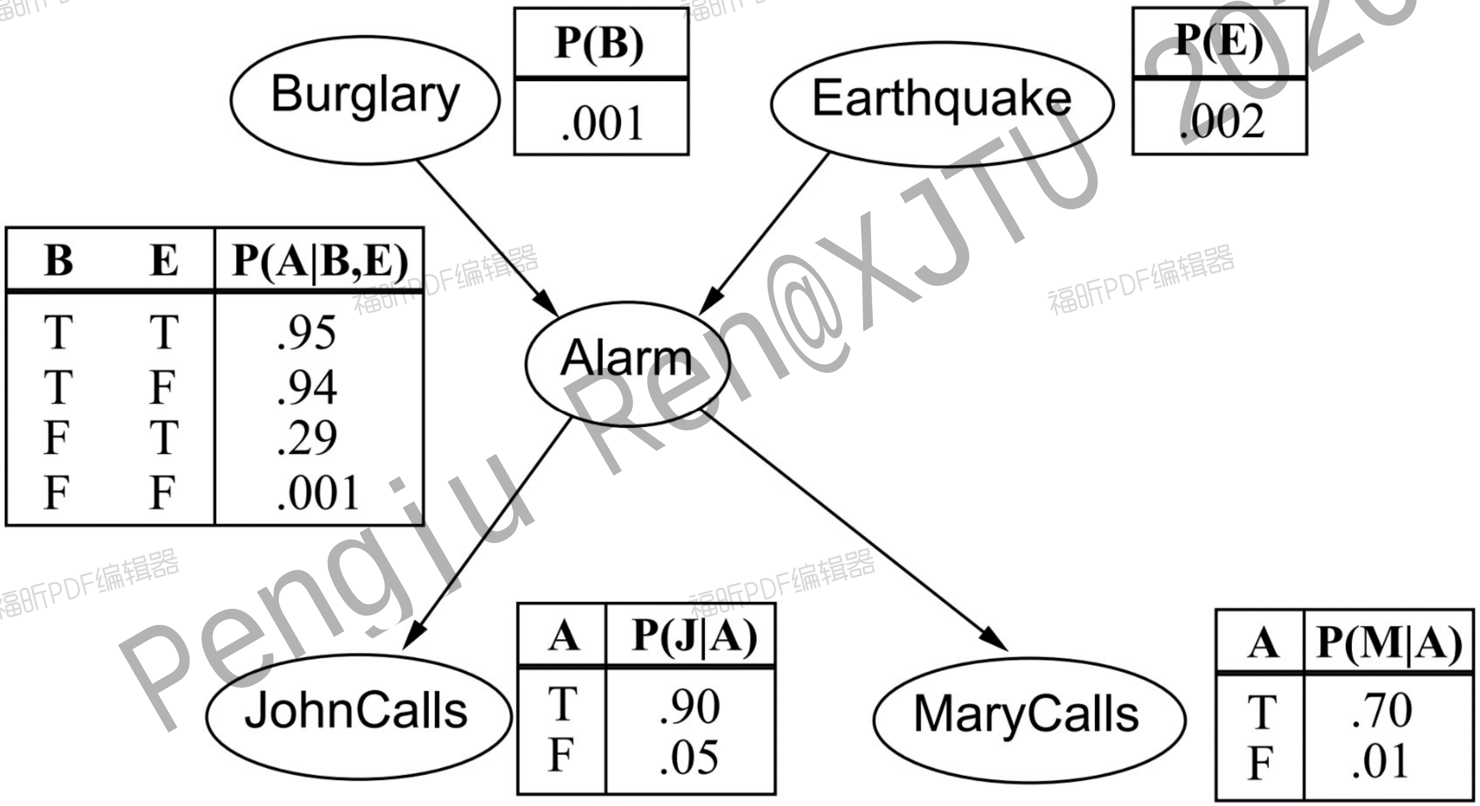
  for $j = 1$ to N **do**
    $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE(bn)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}[X]$)

---

function WEIGHTED-SAMPLE(bn, $\mathbf{e}$) **returns** an event and a weight

  $\mathbf{x} \leftarrow$ an event with n elements; $w \leftarrow 1$
  for $i = 1$ **to** n **do**
    if X_i has a value x_i in $\mathbf{e}$
      then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

Likelihood weighting example



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Likelihood weighting example

Sample 1

Evidence is *Burglary=false* and *Earthquake=false*. We will now query the remaining nodes in the network to determine their state.

We now set the weight *w* is set to 1.0 and *x* to empty.

Burglary is an evidence variable with value *false*. Therefore, we set

$w = wp(Burglary{=}False) = (1.0)(0.999) = 0.999$

$x = (\sim b)$.

Earthquake is an evidence variable with value *false*. Therefore, we set

$w = wp(Earthquake{=}False) = (0.999)(0.998) = 0.997$

$x = (\sim b, \sim e)$.

We sample from p(*Alarm*|*Burglary=false*, *Earthquake=false*) = <0.001, 0.999>; suppose this returns *false*.

$x = (\sim b, \sim e, \sim a)$.

We sample from p(*JohnCalls*|*Alarm=false*) = <0.05, 0.95>; suppose this returns *false*.

$x = (\sim b, \sim e, \sim a, \sim j)$.

We sample from p(*MaryCalls*|*Alarm=false*) = <0.01, 0.99>; suppose this returns *false*.

$x = (\sim b, \sim e, \sim a, \sim j, \sim m)$.

Likelihood weighting example

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997

Likelihood weighting example

Sample 2

Evidence is *Alarm=false* and *JohnCalls=true*. We will now query the remaining nodes in the network to determine their state.

We now set the weight *w* is set to 1.0 and *x* to empty.

Burglary is not an evidence variable so we sample it; suppose it return *false*.

x = (~b).

Earthquake is not an evidence variable so we sample it; suppose it return *false*.

x = (~b,~e).

Alarm is an evidence variable with value *false*. Therefore, we set

w = wp(*Alarm=false* | *Burglary=false*, *Earthquake=false*) = (1.0)(0.999) = 0.999

x = (~b,~e,~a).

JohnCalls is an evidence variable with value *true*. Therefore, we set

w = wp(*JohnCalls=true* | *Alarm=false*) = (0.999)(0.05) = 0.05

x = (~b,~e,~a,j).

MaryCalls is not an evidence variable so we sample it; suppose it return *false*.

x = (~b,~e,~a,j,~m).

Likelihood weighting example

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.05

Likelihood weighting example

Sample 3

Evidence is *JohnCalls=true* and *MaryCalls=true*. We will now query the remaining nodes in the network to determine their state.

We now set the weight *w* is set to 1.0 and *x* to empty.

Burglary is not an evidence variable so we sample it; suppose it return *false*.

x = (~b).

Earthquake is not an evidence variable so we sample it; suppose it return *false*.

x = (~b,~e).

Alarm is not an evidence variable so we sample it; suppose it return *true*.

x = (~b,~e,a).

JohnCalls is an evidence variable with value *true*. Therefore, we set

w = wp(*JohnCalls=true* | *Alarm=true*) = (1.0)(0.90) = 0.90

x = (~b,~e,a,j).

MaryCalls is an evidence variable with value *true*. Therefore, we set

w = wp(*MaryCalls=true* | *Alarm=true*) = (0.90)(0.70) = 0.63

x = (~b,~e,a,j,m).

Likelihood weighting example

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.05
3	~b,~e,a,j,m	0.63

Likelihood weighting example

Sample 4

Evidence is *Burglary=false*, *Earthquake=false*, and *JohnCalls=true*. We will now query the remaining nodes in the network to determine their state.

We now set the weight w is set to 1.0 and x to empty.

Burglary is an evidence variable with value *false*. Therefore, we set

w = wp(*Burglary=False*) = (1.0)(0.999) = 0.999

x = (~b).

Earthquake is an evidence variable with value *false*. Therefore, we set

w = wp(*Earthquake=False*) = (0.999)(0.998) = 0.997

x = (~b,~e).

Alarm is not an evidence variable so we sample it; suppose it return *false*.

x = (~b,~e,~a).

JohnCalls is an evidence variable with value *true*. Therefore, we set

w = wp(*JohnCalls=true* | *Alarm=false*) = (0.997)(0.05) = 0.05

x = (~b,~e,~a,j).

MaryCalls is not an evidence variable so we sample it; suppose it return *false*.

x = (~b,~e,~a,j,~m).

Likelihood weighting example

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.05
3	~b,~e,a,j,m	0.63

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.05+0.05=0.1
3	~b,~e,a,j,m	0.63

Likelihood weighting example

Sample 5

Evidence is *Burglary=true* and *Earthquake=false*. We will now query the remaining nodes in the network to determine their state.

We now set the weight *w* is set to 1.0 and *x* to empty.

Burglary is an evidence variable with value *true*. Therefore, we set

w = wp(*Burglary=True*) = (1.0)(0.001) = 0.001

x = (b).

Earthquake is an evidence variable with value *false*. Therefore, we set

w = wp(*Earthquake=False*) = (.001)(0.998) = 0.001

x = (b,~e).

Alarm is not an evidence variable so we sample it; suppose it return *false*.

x = (b,~e,~a).

JohnCalls is not an evidence variable so we sample it; suppose it return *false*.

x = (b,~e,~a,~j,~m).

MaryCalls is not an evidence variable so we sample it; suppose it return *false*.

x = (b,~e,~a,~j,~m).

Likelihood weighting example

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.1
3	~b,~e,a,j,m	0.63
4	b,~e,~a,~j,~m	0.001

Using Likelihood Weights

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.1
3	~b,~e,a,j,m	0.63
4	b,~e,~a,~j,~m	0.001

In order to compute the probability of an event that is independent, such as P(*Burglary=true*), we sum the weight for every sample where *Burglary=true* and divide by the sum of all of the weights. For example, in the above data, the only sample where *Burglary=true* is sample 4, with weight 0.001. Therefore,

$$P(Burglary = true) = \frac{0.001}{0.997 + 0.1 + 0.63 + 0.001} = \frac{0.001}{1.728} = 0.00058$$

Using Likelihood Weights

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.1
3	~b,~e,a,j,m	0.63
4	b,~e,~a,~j,~m	0.001

In order to compute the probability of an event, *X=true*, that is dependent on another event, *Y=true*, we sum the weights of all samples where *X=true* **and** *Y=true* and divide it by the sum of the weights of all samples where *Y=true*. For example, if we want to compute p(a | j), we need to sum the weights of all samples where we have both a and j (meaning *Alarm=True* and *JohnCalls=True*). We find that only sample 3 meets this criteria with a weight of 0.63. We now sum the weights of all samples that have j. Only samples 2 and 3 meet this criteria with weights 0.10 and 0.63, respectively. Putting this all together, we have

$$P(a|j) = \frac{0.63}{0.1 + 0.63} = \frac{0.63}{0.73} = 0.863$$

Using Likelihood Weights

Sample	Key	Weight
1	~b,~e,~a,~j,~m	0.997
2	~b,~e,~a,j,~m	0.1
3	~b,~e,a,j,m	0.63
4	b,~e,~a,~j,~m	0.001

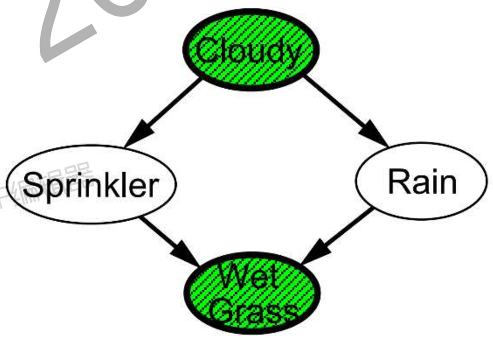
In the above data, the probability of an event that has never been observed is zero. This is because we have information about every node in the alarm network in every sample. For example, if we want to compute p(b | a), we need to sum the weights for all samples where we have both b and a. There are no such samples. Therefore, the sum is zero and the probability is zero.

Likelihood weighting analysis

Sampling probability for *WEIGHTSAMPLE* is

$$S_{WS}(z,e) = \prod_{i=1}^{l} P(z_i | parents(Z_i))$$

Note: pays attention to evidence in ancestors only
⇒ somewhere "in between" prior and posterior distribution



Weight for a given sample z, e is

$$w(z,e) = \prod_{i=1}^{m} P(e_i | parents(E_i))$$

Weighted sampling probability is

$$S_{WS}(z,e)w(z,e) = \prod_{i=1}^{l} P(z_i \mid parents(Z_i)) \prod_{i=1}^{m} P(e_i \mid parents(E_i)) = P(z,e)$$

(by standard global semantics of network)

Likelihood weighting analysis

$$\begin{aligned}
\hat{P}(x \mid \mathbf{e}) &= \alpha \sum_{\mathbf{y}} N_{WS}(x, \mathbf{y}, \mathbf{e}) w(x, \mathbf{y}, \mathbf{e}) && \text{from LIKELIHOOD-WEIGHTING} \\
&\approx \alpha' \sum_{\mathbf{y}} S_{WS}(x, \mathbf{y}, \mathbf{e}) w(x, \mathbf{y}, \mathbf{e}) && \text{for large } N \\
&= \alpha' \sum_{\mathbf{y}} P(x, \mathbf{y}, \mathbf{e}) && \text{by Equation (14.9)} \\
&= \alpha' P(x, \mathbf{e}) = P(x \mid \mathbf{e}) .
\end{aligned}$$

$$S_{WS}(z,e)w(z,e) = \prod_{i=1}^{l} P(z_i \mid parents(Z_i)) \prod_{i=1}^{m} P(e_i \mid parents(E_i)) = P(z,e)$$

Hence likelihood weighting returns consistent estimates
but performance still degrades with many evidence variables
because a few samples have nearly all the total weight

Approximate inference using MCMC

"State" of network = current assignment to all variables.
Generate next state by sampling one variable given Markov blanket
Sample each variable in turn, keeping evidence fixed

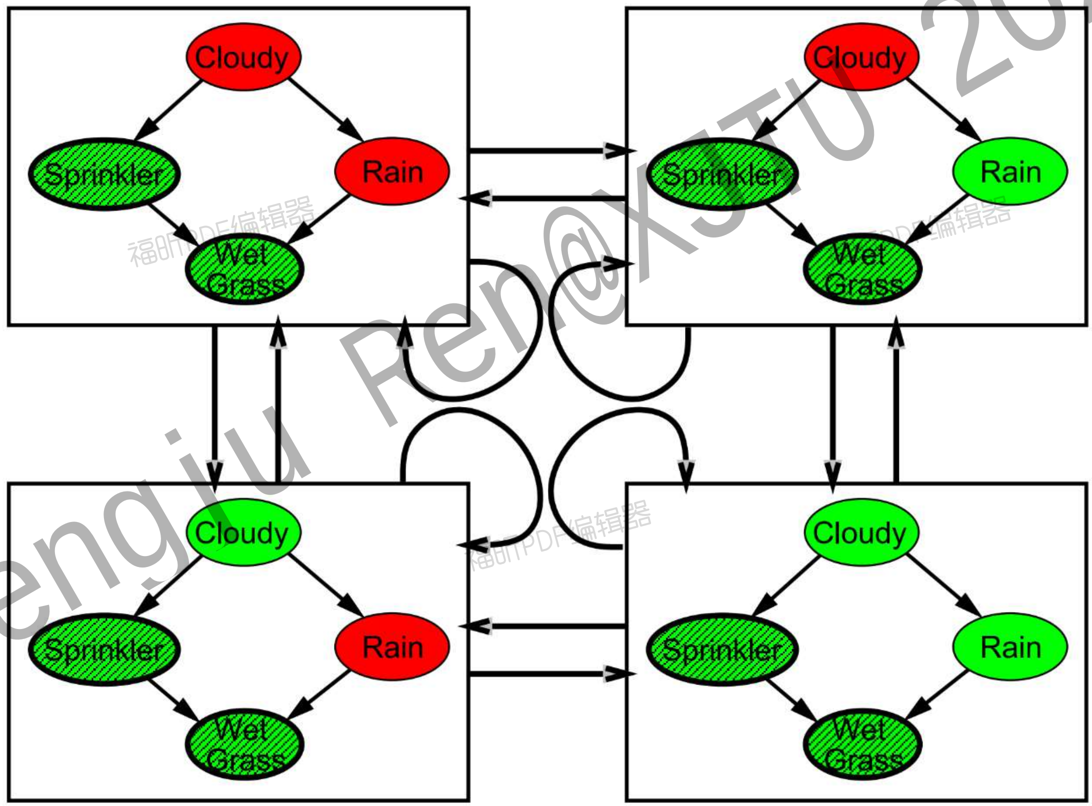
function MCMC-ASK(X, $\mathbf{e}$, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Y}$
  for $j = 1$ to N **do**
    for each Z_i in $\mathbf{Z}$ **do**
      sample the value of Z_i in $\mathbf{x}$ from $\mathbf{P}(Z_i|mb(Z_i))$
        given the values of $MB(Z_i)$ in $\mathbf{x}$
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{N}[X]$)

Can also choose a variable to sample at random each time

The Markov chain

With *Sprinkler = true, WetGrass = true*, **there are four states:**



Wander about for a while, average what you see

MCMC example contd.

Estimate *P(Rain|Sprinkler= true, WetGrass= true)*

Sample *Cloudy* **or** *Rain* **given its Markov blanket, repeat.**

Count number of times *Rain* **is true and false in the samples.**

E.g., visit 100 states

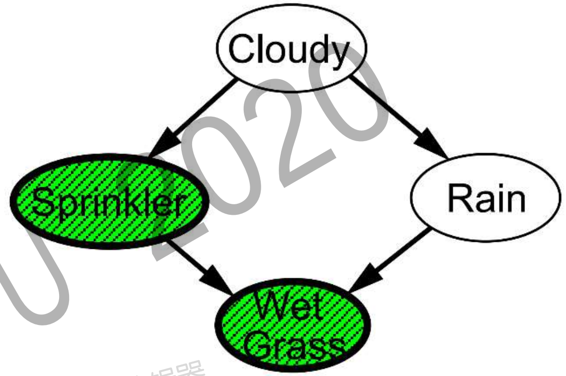
31 have *Rain= true*, **69 have** *Rain= false*

$$\hat{P}(Rain|Sprinkler = true, WetGrass = true)$$
$$= NORMALIZE(\langle 31, 60\rangle) = \langle 0.31, 0.69\rangle$$

Theorem: chain approaches **stationary distribution:**

long-run fraction of time spent in each state is exactly proportional to its posterior probability

Markov blanket sampling



Markov blanket of ***Cloudy*** **is**

Sprinkler **and** *Rain*

Markov blanket of ***Rain*** **is**

Cloudy, *Sprinkler*, **and** *WetGrass*

Probability given the Markov blanket is calculated as follows:

$$P(x_i'|mb(X_i)) = P(x_i'|parents(X_i)) \prod_{Z_j \in Children(X_i)} P(z_j|parents(Z_j))$$

Easily implemented in message-passing parallel systems, brains

Main computational problems:

1) **Difficult to tell if convergence has been achieved**
2) **Can be wasteful if Markov blanket is large:**

$P(x_i|mb(X_i))$ **won't change much (law of large numbers)**

Summary

- Exact inference by variable elimination:

– polytime on polytrees, NP-hard on general graphs

– space = time, very sensitive to topology

- Approximate inference by LW, MCMC:
  - – LW does poorly when there is lots of (downstream) evidence
  - – LW, MCMC generally insensitive to topology
  - – Convergence can be very slow with probabilities close to 1 or 0
  - – Can handle arbitrary combinations of discrete and continuous variables