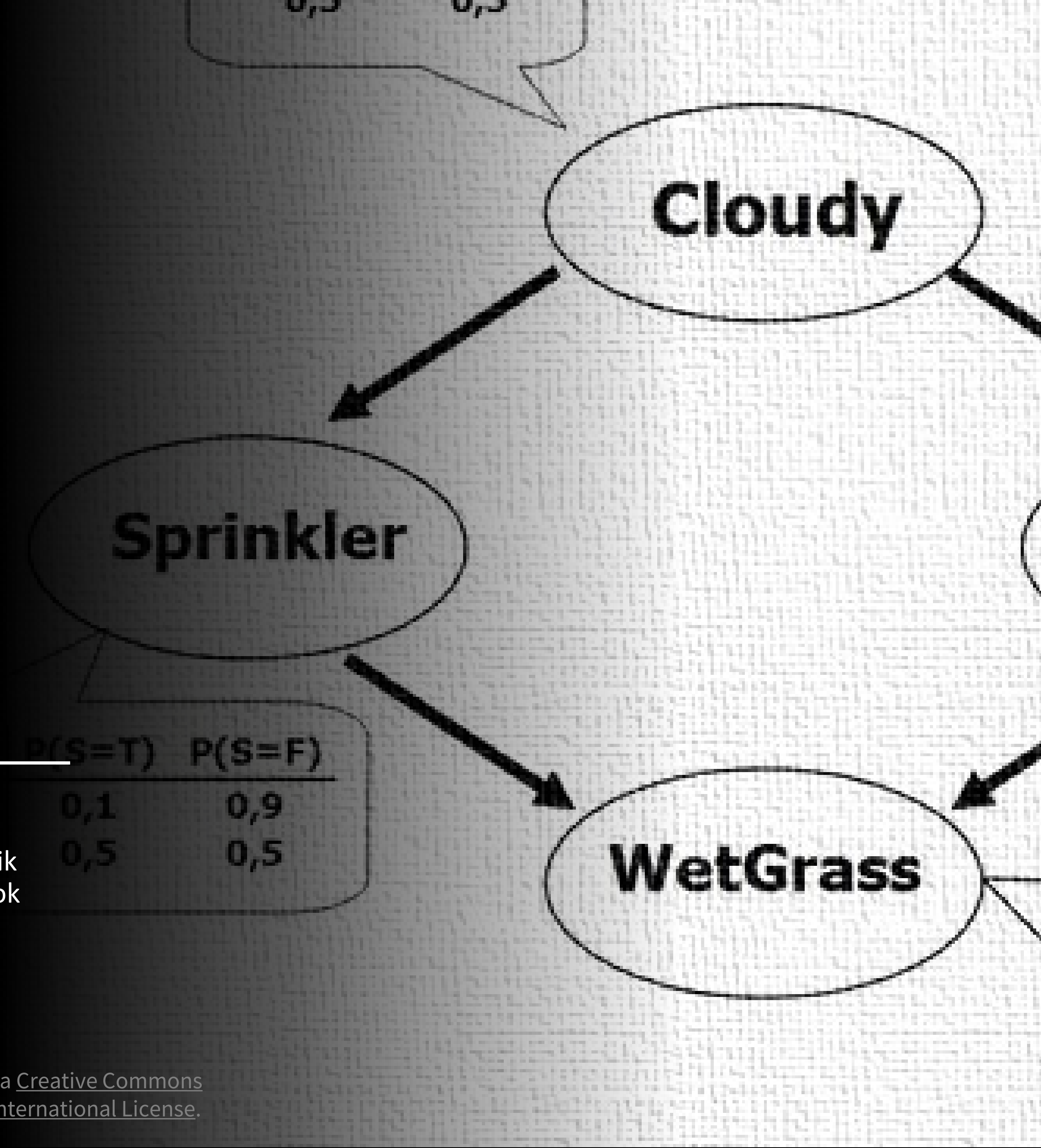
CS 5/7320 Artificial Intelligence

Probabilistic Reasoning (Bayesian networks)

AIMA Chapter 13



Slides by Michael Hahsler
based on slides by Svetlana Lazepnik
with figures from the AIMA textbook

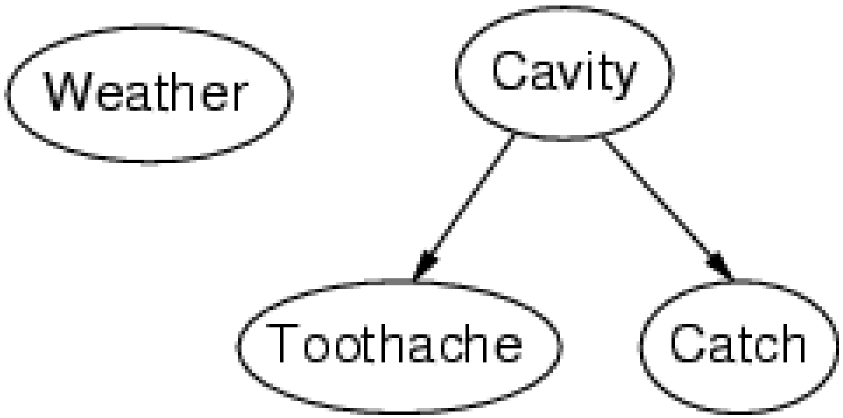
Probability Recap

- Conditional probability $$P(x|y) = \frac{P(x,y)}{P(y)} = \alpha P(x,y)$$
- Product rule $$P(x,y) = P(x|y)P(y)$$
- Chain rule $$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1,\ldots,X_{i-1}) \end{aligned}$$
- X, Y independent if and only if: $\forall x, y : P(x,y) = P(x)P(y)$
- X and Y are conditionally independent given Z if and only if:

$$\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$$ Written as $X \perp\!\!\!\perp Y | Z$

Notation: $P(X = x) = P(x)$

Bayesian networks (aka Belief Networks)

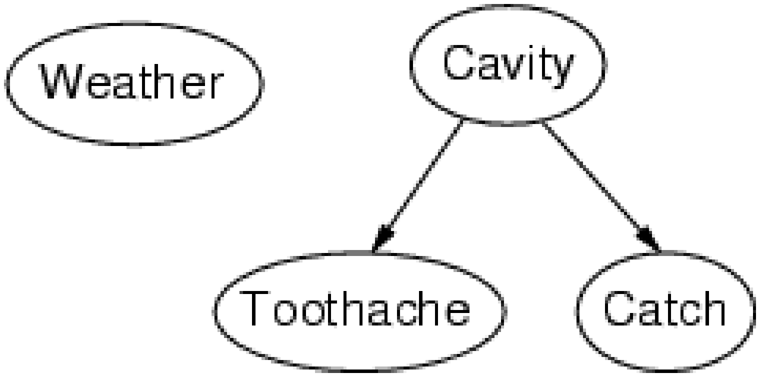


A type of graphical model.

A way to specify dependence between random variables.

A compact specification of a full joint distributions.

Structure of Bayesian Networks



Nodes: Random variables

- Can be assigned (observed) or unassigned (unobserved)

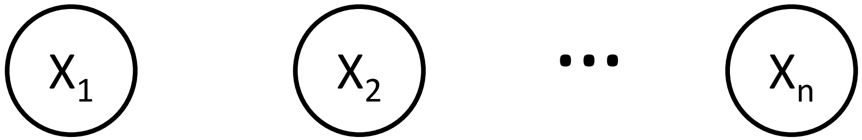
Arcs: Dependencies

- An arrow from one variable to another indicates direct influence.
- Show independence
  - *Weather* is independent of the other variables (no connection).
  - *Toothache* and *Catch* are conditionally independent given *Cavity* (directed arc).
- Must form a directed *acyclic* graph (DAG)

A network with all random variables assigned represents a **state of the system**.

Example: N independent coin flips

Complete independence: no interactions between coin flips

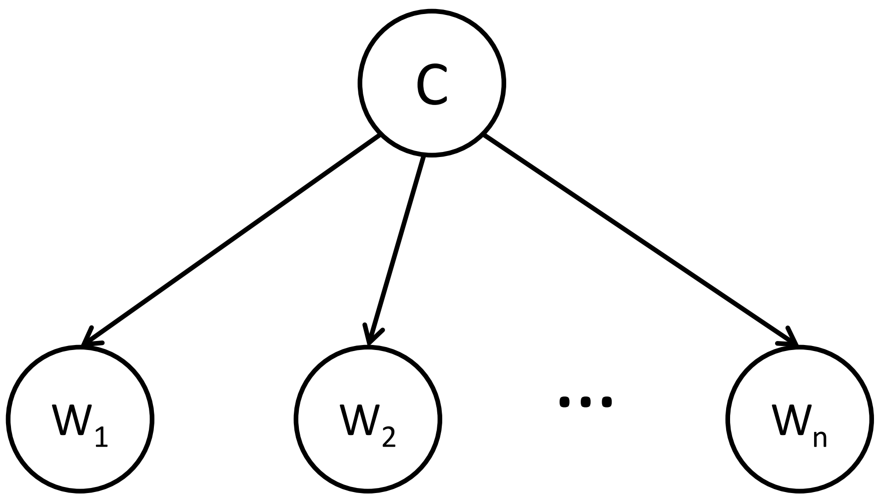


Example: Naïve Bayes spam filter

Random variables:

- C: message class (spam or not spam)
- $W_1, \ldots, W_n$: presence or absence of words comprising the message

Words depend on the class, but they are modeled conditional independent of each other given the class (= no direct connection between words).



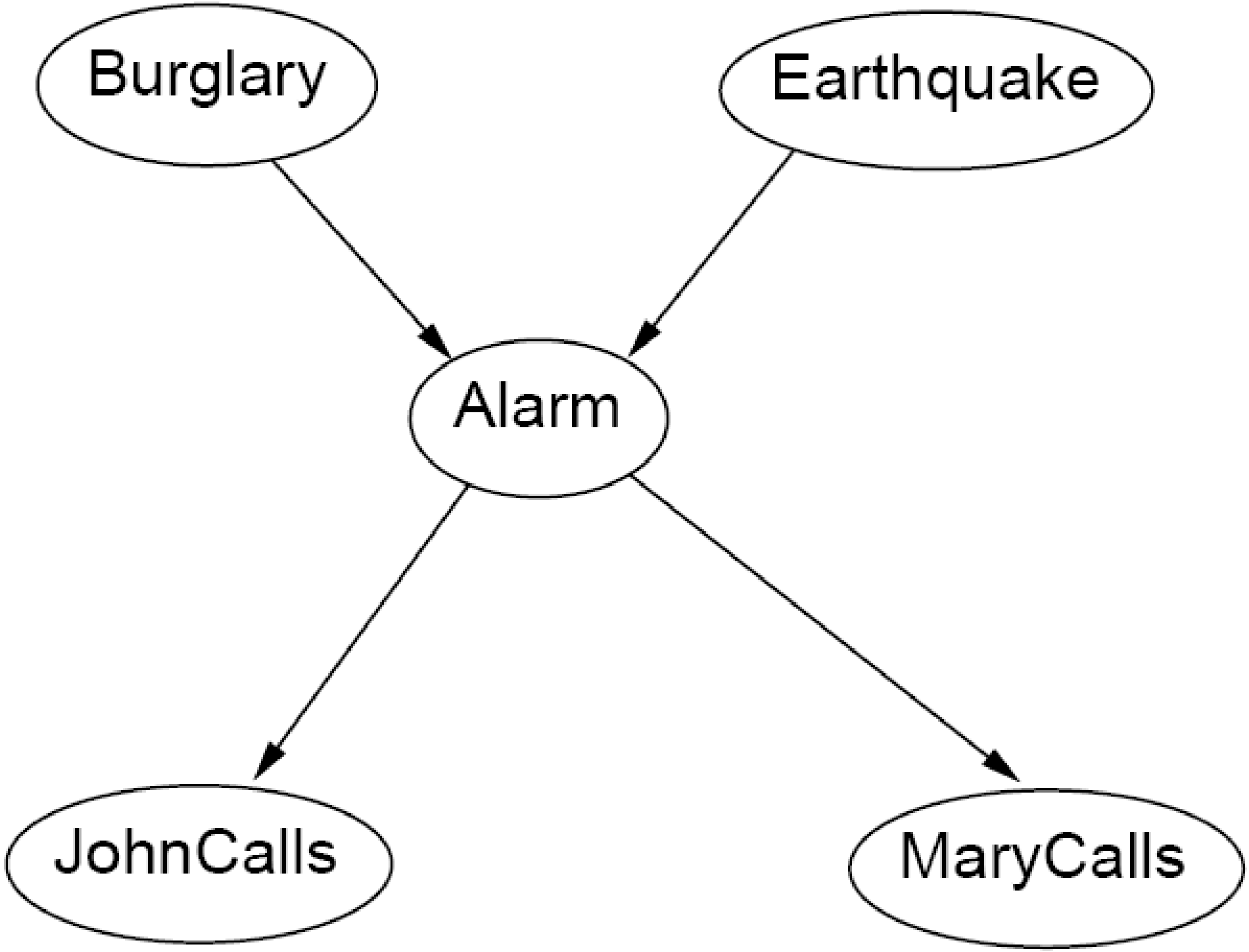
Example: Burglar Alarm

- **Description**: I have a burglar alarm that is sometimes set off by minor earthquakes. My two neighbors, John and Mary, promised to call me at work if they hear the alarm
- Example inference task: suppose Mary calls and John doesn't call. What is the probability of a burglary?

- What are the random variables?
  - Burglary, Earthquake, Alarm, JohnCalls, MaryCalls

- What are the direct influence relationships?
  - A burglar can set off the alarm
  - An earthquake can set off the alarm
  - The alarm can cause Mary to call
  - The alarm can cause John to call

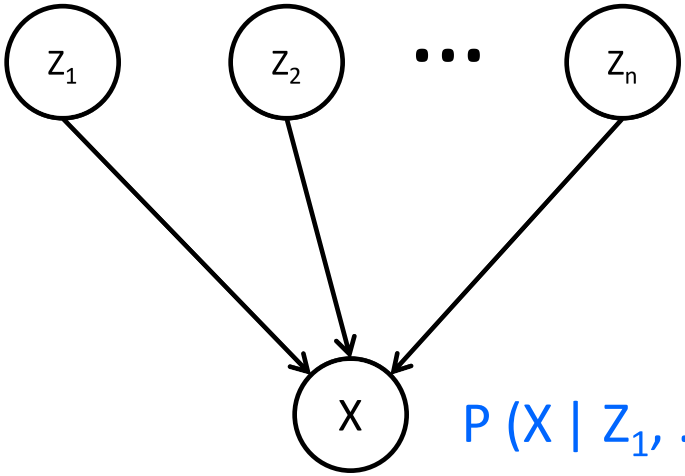
Example: Burglar Alarm



What are the model parameters?

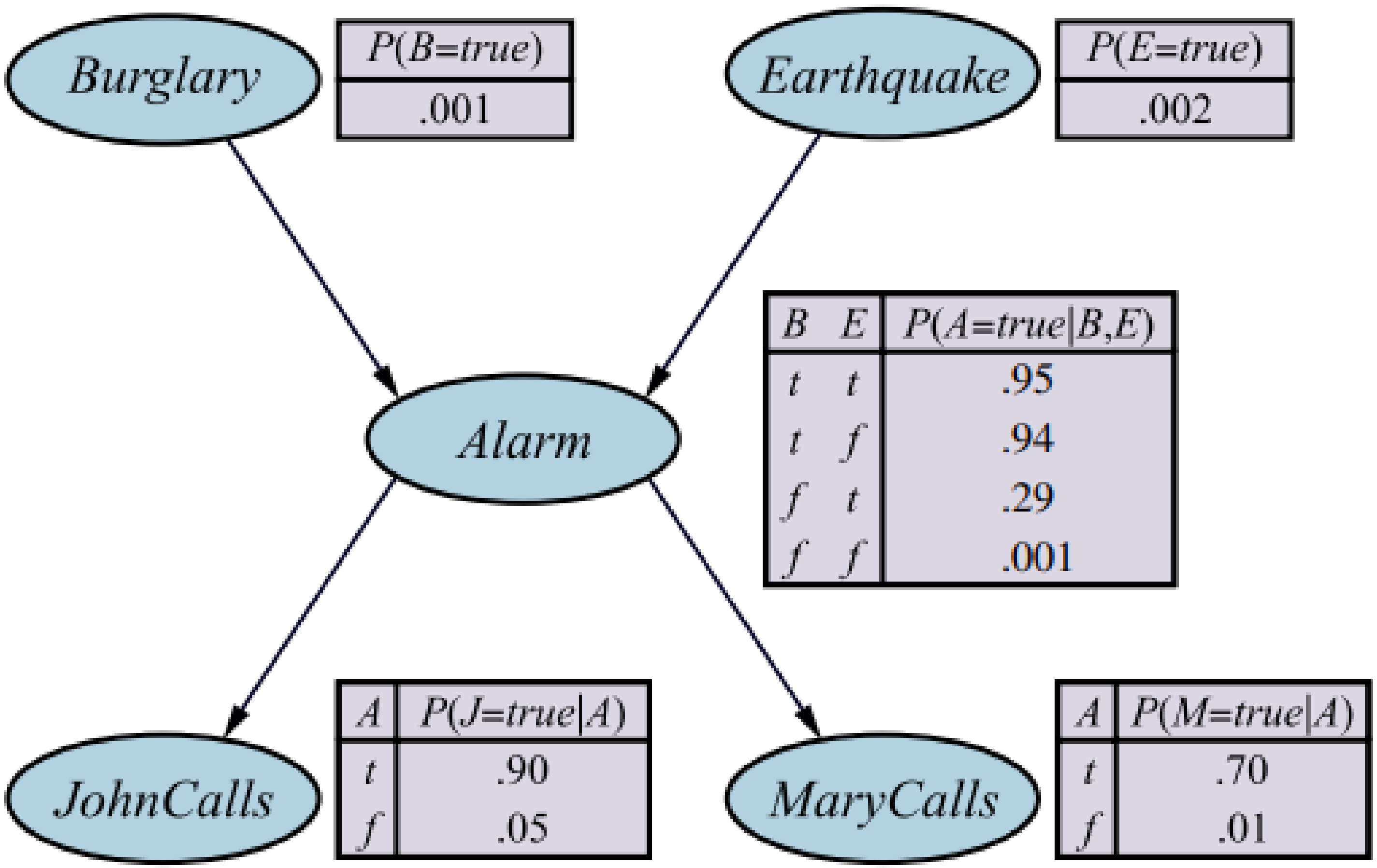
Parameters: Conditional probability tables

To specify the full joint distribution, we need to specify a *conditional* distribution for each node given its parents as a conditional probability table (CPT): P (X | Parents(X))



$P(X \mid Z_1, \ldots, Z_n)$

Example: Burglar Alarm with CPTs



$P(B=true)$
.001

$P(E=true)$
.002

No parents

B	E	$P(A=true \mid B,E)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

2 parents

A	$P(J=true \mid A)$
t	.90
f	.05

A	$P(M=true \mid A)$
t	.70
f	.01

1 parent

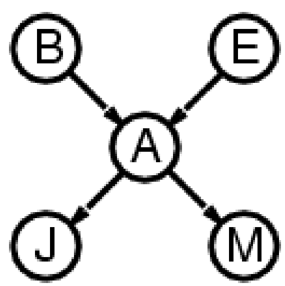
The joint probability distribution

- For each node X_i, we know $P(X_i \mid Parents(X_i))$
- How do we get the full joint distribution $P(X_1, \ldots, X_n)$?

- Using chain rule:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid X_1,\ldots,X_{i-1}\right) = \prod_{i=1}^{n} P\left(X_i \mid Parents(X_i)\right)$$

- Example:



Construct following arrows

$$P(j, m, a, b, e) \;=\; P(b)\, P(e)\, P(a \mid b, e)\, P(j \mid a)\, P(m \mid a)$$

Dependence

- Example: *causal chain*

X: Low pressure

Y: Rain

Z: Traffic

- Are X and Z independent?

$$P(X, Y, Z) = P(X)P(Y|X)P(Z|Y)$$

Conditioning

$$P(X, Z) = \sum_y P(X)P(y|X)P(Z|y)$$

Marginalize over Y

$$= P(X)\sum_y P(Z|y)P(y|X) \neq P(X)P(Z)$$

X and Z are **not** independent!

Conditional independence

- Example: *causal chain*

X: Low pressure

Y: Rain

Z: Traffic

- Is Z independent of X given Y?

$$P(X,Z|Y) = \frac{P(X,Y,Z)}{P(Y)} = \frac{\mathrm{P(X)P(Y|X)P(Z|Y)}}{P(Y)}$$ Conditioning

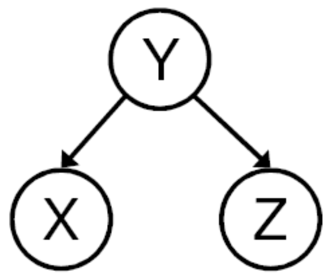
$$= \frac{P(X)\frac{P(X|Y)P(Y)}{P(X)}P(Z|Y)}{P(Y)}$$ Bayes' rule

$= \mathrm{P(X|Y)P(Z|Y)}$ = Definition of conditional indepedence

X and Z are conditionally independent given Y

Conditional independence

- Common cause



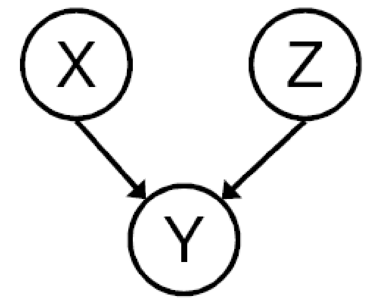
Y: Project due

X: Newsgroup busy

Z: Lab full

- Are X and Z independent?
  - No
- Are they conditionally independent given Y?
  - Yes

- Common effect



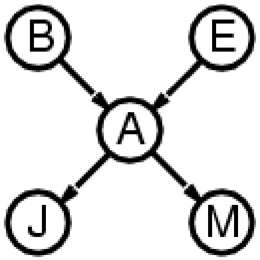
X: Raining

Z: Ballgame

Y: Traffic

- Are X and Z independent?
  - Yes
- Are they conditionally independent given Y?
  - No

Compactness



- Suppose we have a Boolean variable X_i with k Boolean parents. How many rows does its conditional probability table have?
  - 2^k rows for all the combinations of parent values, each row requires one number p for X_i = true
- If each variable has no more than k parents, how many numbers does the complete network require?
  - $O(n \cdot 2^k)$ numbers – vs. $O(2^n)$ for the full joint distribution

- Example: How many nodes for the burglary network?

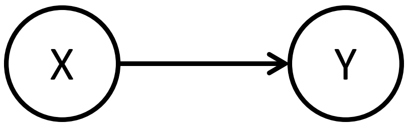
  1 + 1 + 4 + 2 + 2 = 10 numbers

  (vs. specification of the complete joint probability 2^5-1 = 31)

Constructing Bayesian networks

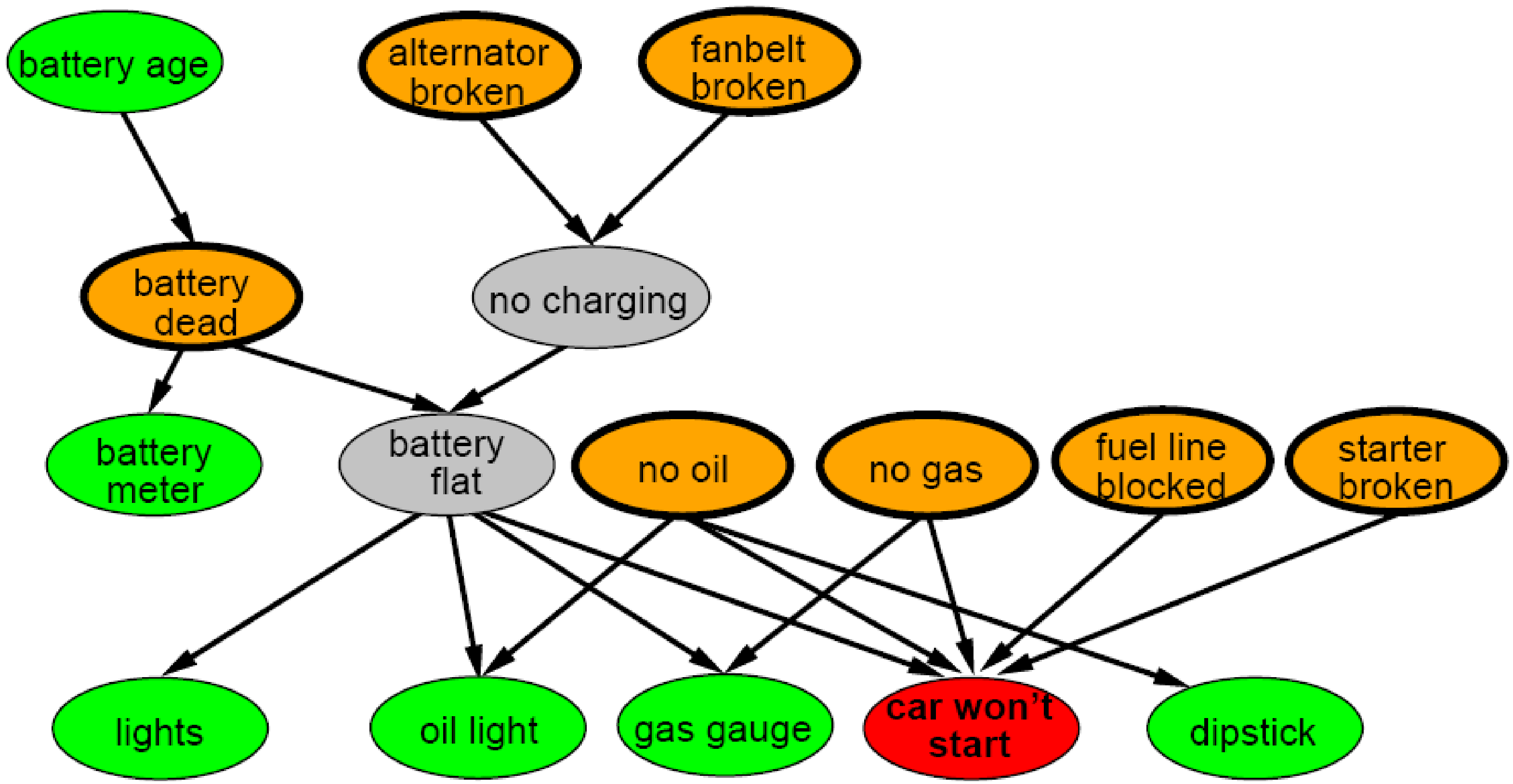
1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
   - add X_i to the network
   - select parents from $X_1, \ldots, X_{i-1}$ such that
     $P(X_i \mid Parents(X_i)) = P(X_i \mid X_1, \ldots X_{i-1})$

Note: Networks are typically constructed by domain experts with causality in mind. E.g., X causes Y:



A more realistic Bayes Network: Car diagnosis

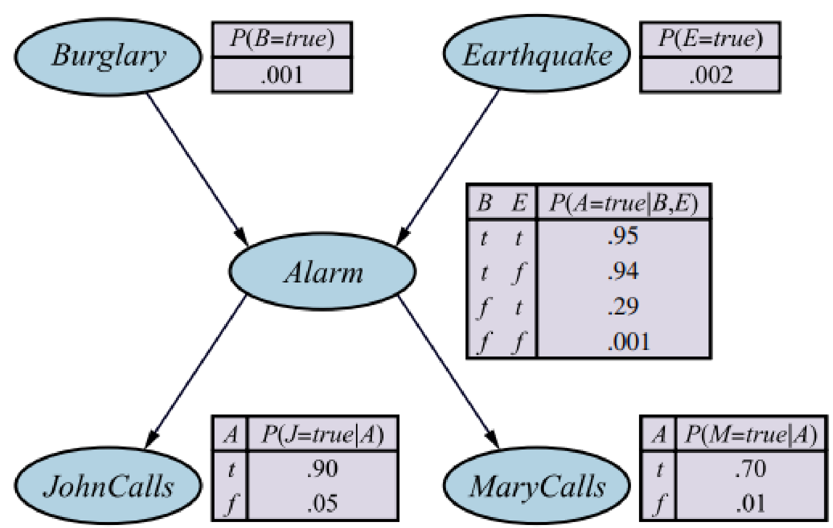
- Initial observation: car won't start
- Green: testable evidence
- Orange: "broken, so fix it" nodes
- Gray: "hidden variables" to ensure sparse structure, reduce parameters

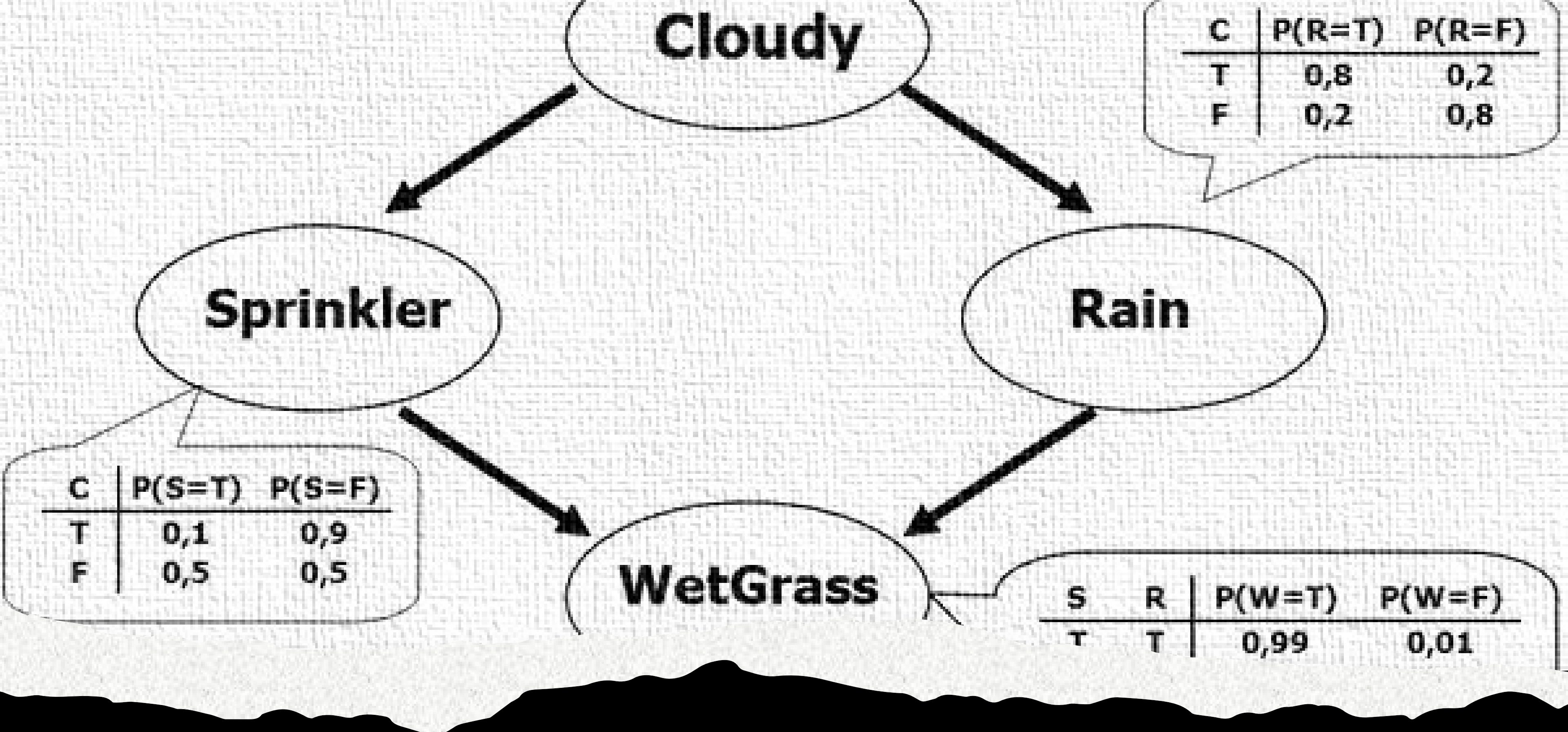


Summary

- Bayesian networks provide a natural representation for joint probabilities used to calculated conditional probabilities used in inference.
- Conditional independence (induced by causality) reduces the number of needed parameters.
- Representation
  - Topology
  - Conditional probability tables
  - Generally easy for domain experts to construct

$P(B, E, A, J, M)$ is defined by





Inference

Calculate the posterior probability given evidence

Inference

Goal

- Query *variables:* **X**
- *Evidence* (*observed*) variables: **E = e**
- *Set of unobserved* variables: **Y**
- Calculate the probability of X given e.

If we know the full joint distribution **P(X, E, Y)**, we can infer **X** by:

$$\boldsymbol{P}(X|E = e) = \frac{\boldsymbol{P}(X, e)}{P(e)} \propto \sum_{y} \boldsymbol{P}(X, e, y)$$

Sum over values of unobservable variables = marginalizing them out.

Inference: Bayesian network

$$\boldsymbol{P}(X|E=e) = \frac{\boldsymbol{P}(X,e)}{P(e)} \propto \boxed{\sum_y} \boxed{\boldsymbol{P}(X,e,y)}$$

Problems

1. **Full joint distributions are too large** to store.

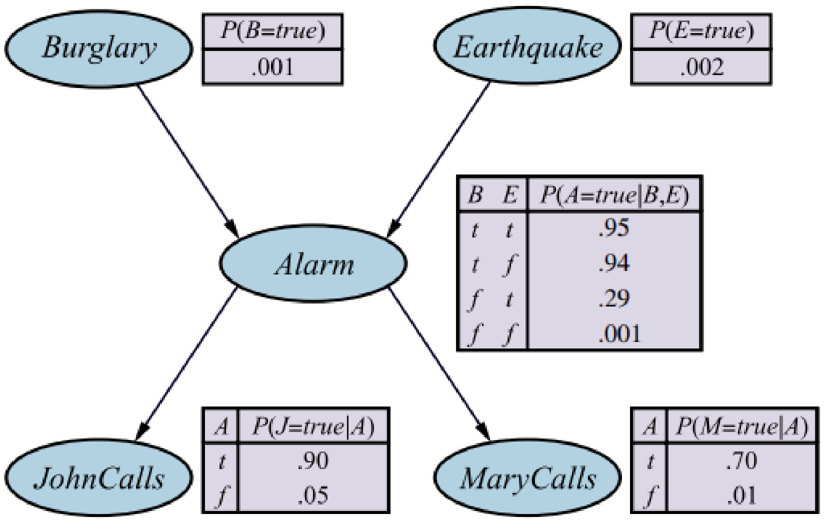
   Bayes nets provide significant savings for representing the conditional probability structure.

2. Marginalizing out many unobservable variables Y may involve **too many summation terms**.

   This summation is called **exact inference by enumeration**. Unfortunately, it does not scale well (#p-hard).

In praxis, **approximate inference by sampling** is used.

Exact inference: Example



Assume we can observe being called. And want to know the probability of a burglary.
Query: **P**(B | j, m) with unobservable variables: Earthquake, Alarm

$$P(b|j,m) = \frac{P(b,j,m)}{P(j,m)} \propto \sum_{E=e} \sum_{A=a} P(b,e,a,j,m)$$

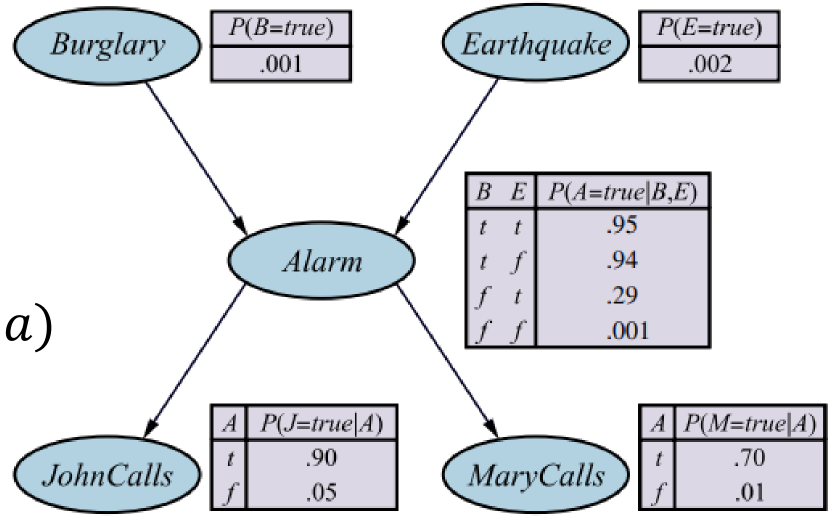
$$= \sum_{E=e} \sum_{A=a} P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$

$$= P(b) \sum_{E=e} P(e) \sum_{A=a} P(a|b,e)\, P(j|a)P(m|a)$$

Full joint probability and marginalize over E and A

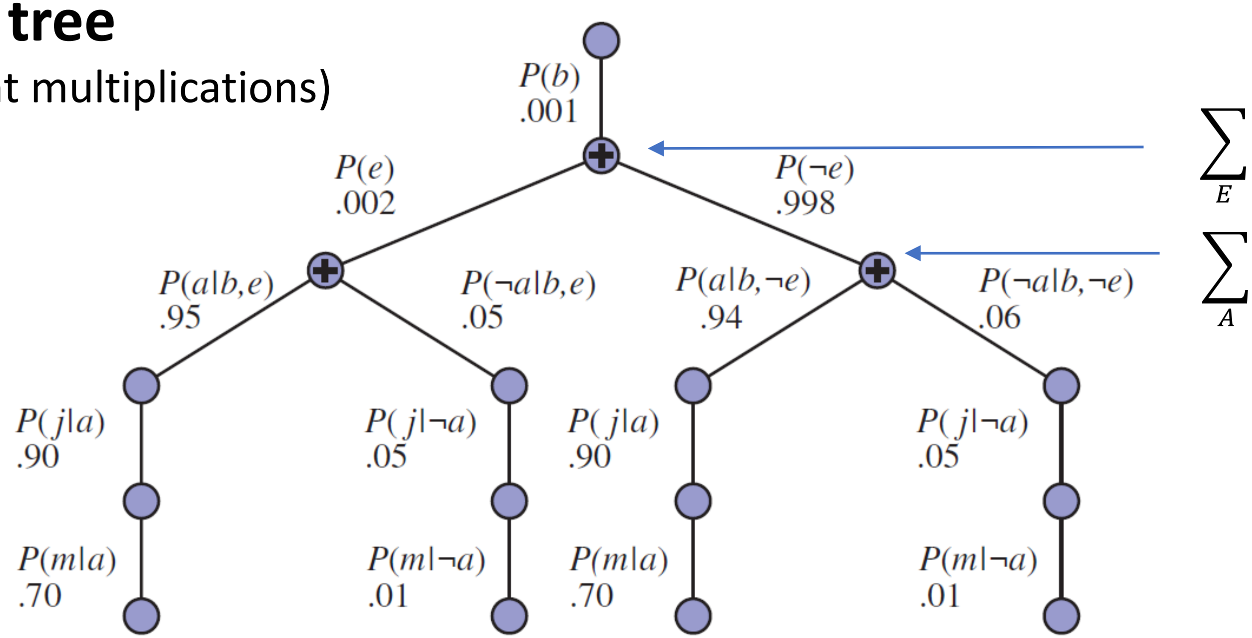
Exact inference: Example

$$P(b|j,m)$$
$$\propto P(b)\sum_{E=e} P(e)\sum_{A=a} P(a|b,e)\,P(j|a)P(m|a)$$



Evaluation tree

(lines represent multiplications)



Approximate inference: Sampling

Bayesian networks can be used as ***generative models.***

Allows us to efficiently generate samples from the joint distribution.

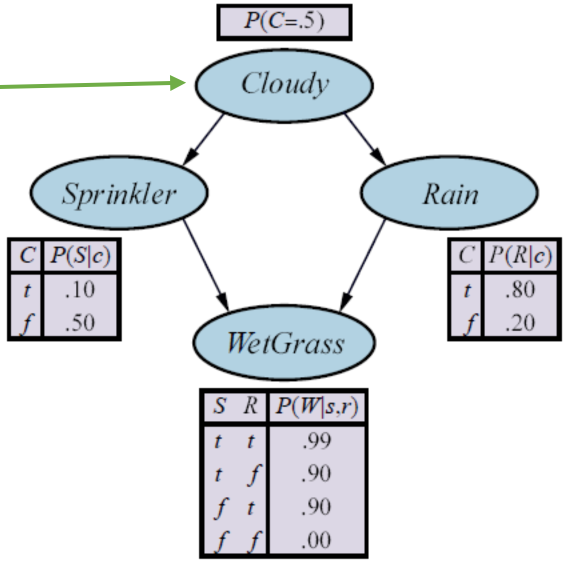
Idea: Generate samples from the network to estimate joint and conditional probability distributions.

Prior-Sample Algorithm to Create a Sample (Event)

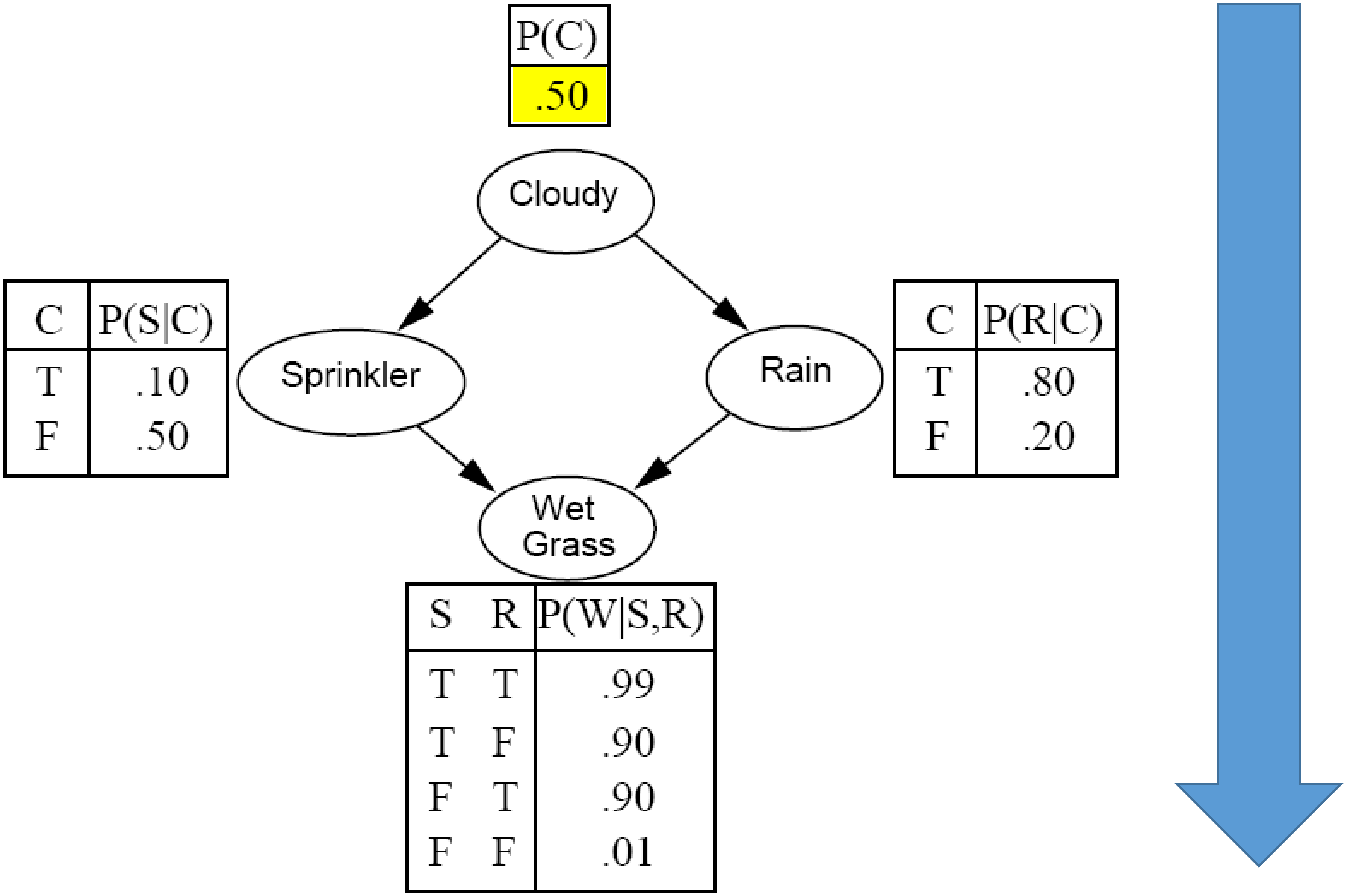
function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for each variable X_i **in** $X_1, \ldots, X_n$ **do**
    $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return x

We need to start with the random variables that have no parents.



Example: Sampling from a Bayesian Network



P(C)
.50

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

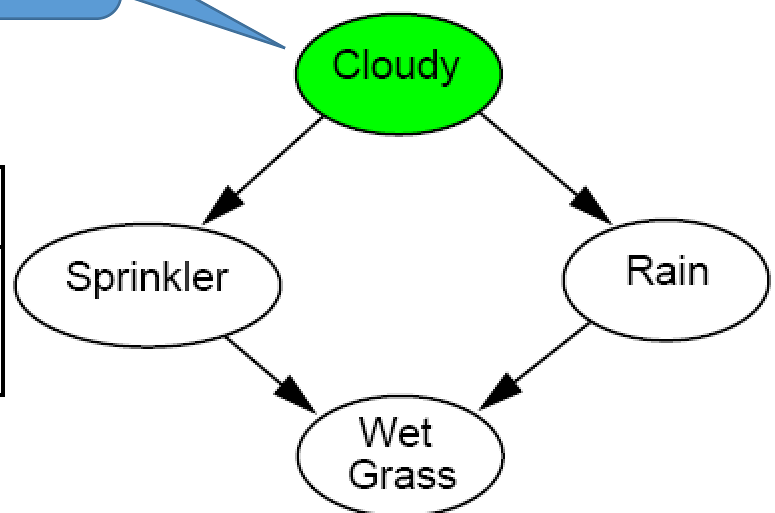
C	P(R|C)
T	.80
F	.20

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

Example: Sampling from a Bayesian Network

Draw a random value using the probability.

P(C)
.50

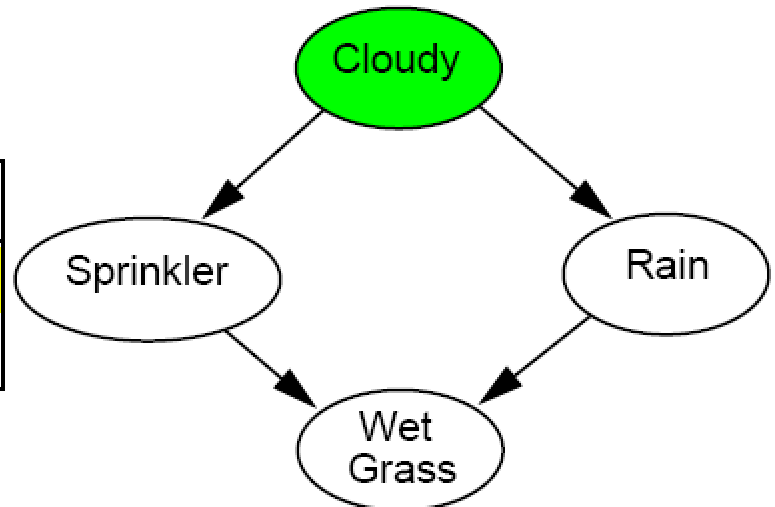


C	P(S|C)
T	.10
F	.50

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

Example: Sampling from a Bayesian Network



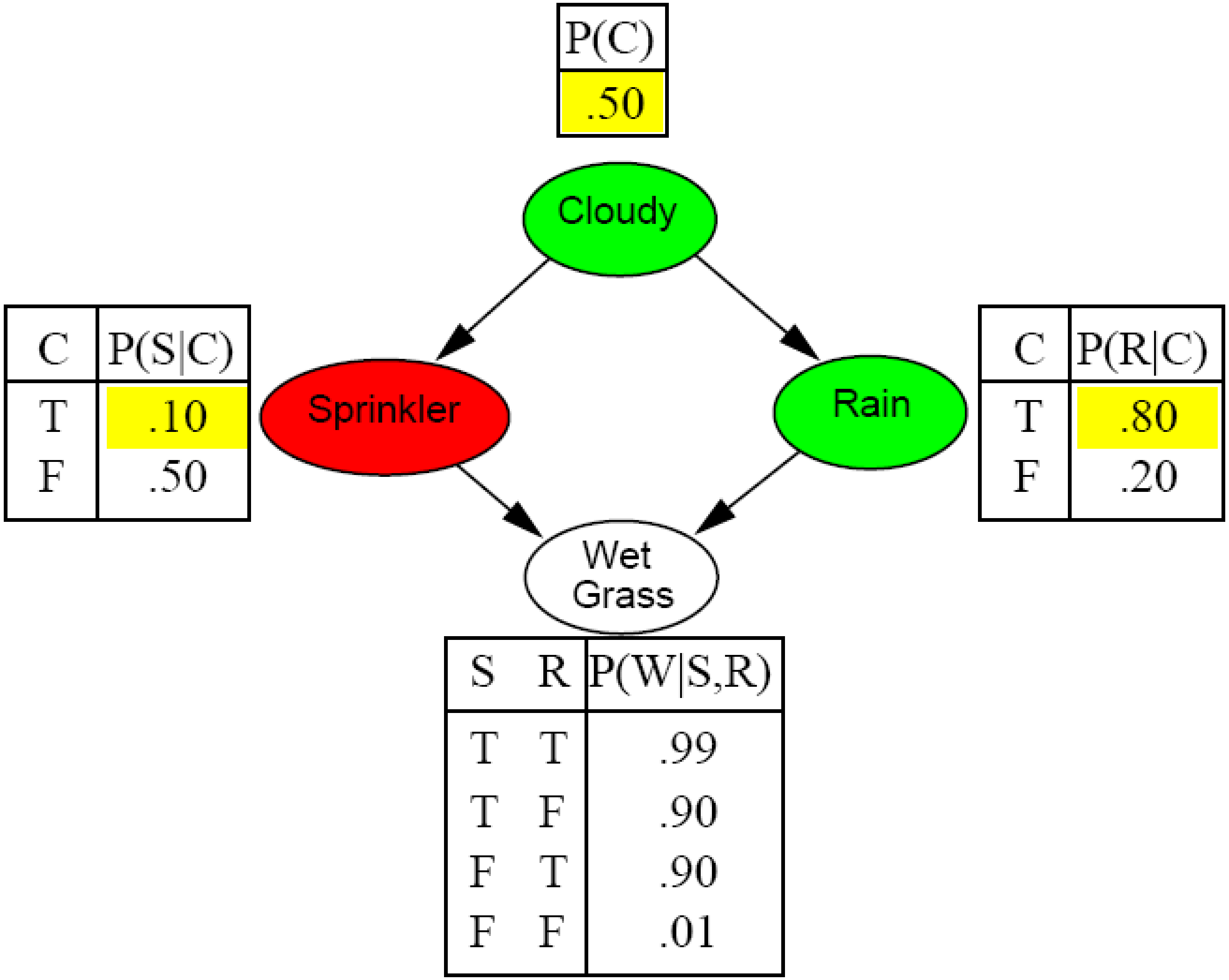
P(C)
.50

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

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

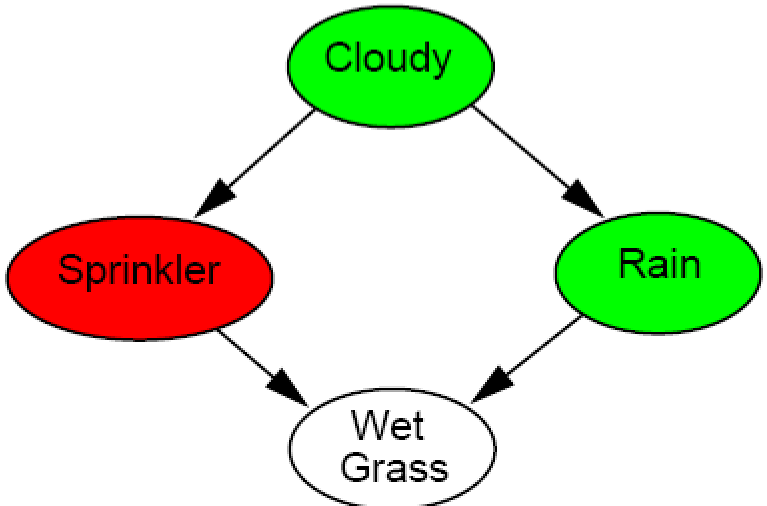
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: Sampling from a Bayesian Network



Example: Sampling from a Bayesian Network

P(C)
.50

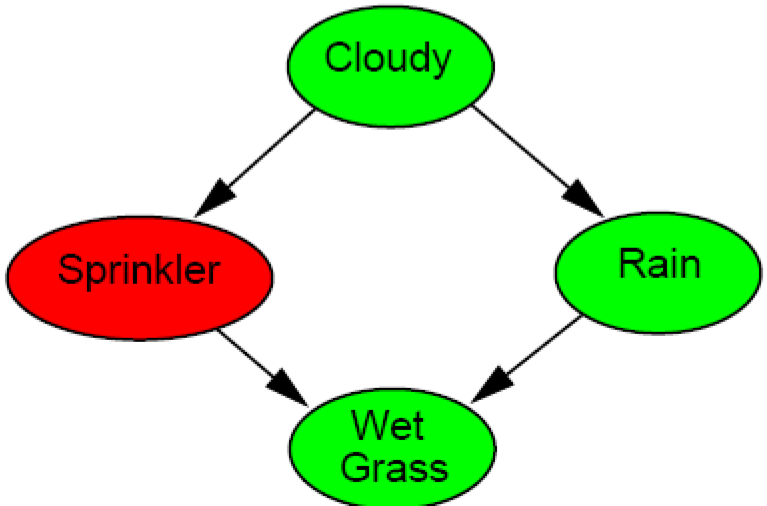


C	P(S|C)
T	.10
F	.50

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

Example: Sampling from a Bayesian Network



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

Prior Sample returns the event:

[C = True, S = False, R = True, W = True]

Estimating the joint probability distribution

Sample *N* times and determine $N_{PS}(x_1, x_2, \ldots, x_n)$, the count of how many times Prior-Sample produces event $(x_1, x_2, \ldots, x_n)$.

$$\hat{P}(x_1, x_2, \ldots, x_n) = \frac{N_{PS}(x_1, x_2, \ldots, x_n)}{N}$$

The marginal probability of partially specified event (some x values are known) can also be calculates. E.g.,

$$\hat{P}(x_1) = \frac{N_{PS}(x_1)}{N}$$

Estimating conditional probabilities: **Rejection sampling**

Sample *N* times and **ignore the samples that are not consistent with the evidence e.**

$$\hat{P}(X|e) = \alpha N_{PS}(X, e) = \frac{N_{PS}(X, e)}{N_{PS}(e)}$$

Issue: What if e is a rare event?

- Example: burglary $\wedge$ earthquake
- Rejection sampling ends up throwing away most of the samples. This is very inefficient!

Estimating conditional probabilities: **Rejection sampling**

function REJECTION-SAMPLING(X, $\mathbf{e}$, bn, N) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  inputs: X, the query variable
        $\mathbf{e}$, observed values for variables $\mathbf{E}$
        bn, a Bayesian network
        N, the total number of samples to be generated
  local variables: $\mathbf{C}$, a vector of counts for each value of X, initially zero

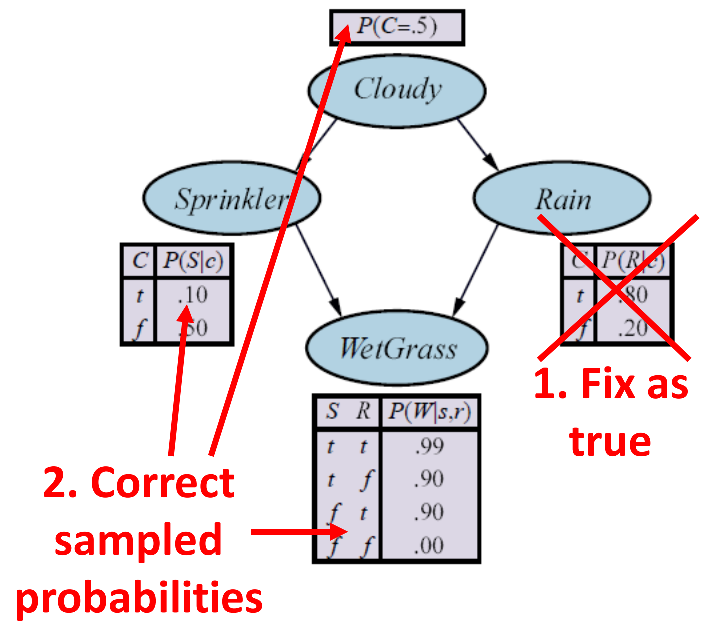
  for $j = 1$ **to** N **do**
    $\mathbf{x} \leftarrow$ PRIOR-SAMPLE(bn)
    if $\mathbf{x}$ is consistent with $\mathbf{e}$ **then**
      $\mathbf{C}[j] \leftarrow \mathbf{C}[j]+1$ where x_j is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{C}$)

We throw away many samples if e is rare!

Estimating conditional probabilities: **Importance sampling** (likelihood weighting)

Avoid the need to throw out samples.

Example: Evidence = it rains



1. Fix the evidence $\boldsymbol{E = e}$ for sampling and estimate the probably for the non-evidence variables

$$Q_{WS}(x)$$

2. Correct the probabilities using weights

$$P(x|e) = w(x) Q_{WS}(x)$$

Turns out the weights in this case can be easily calculated

$$w(x) = \alpha \prod_{i=1}^{m} P(e_i | parents(E_i))$$

Estimating conditional probabilities: Markov Chain Monte Carlo Sampling (MCMC)

- **Generates a sequence of samples** instead of creating each sample individually from scratch.
- Create a state by making random changes to the current state. The sequence of states forms a random process called a **Markov Chain** (MC).
- The MCs stationary distribution turns out to be the posterior distribution of the non-evidence variables.
- Estimate the stationary distribution using **Monte Carlo** simulation by counting how often each state is reached and normalize to obtain probability estimates.
- Algorithms:
  1. Gibbs sampling (works well for BNs)
  2. Metropolis-Hastings sampling

Note: Simulated annealing belongs to the family of MCMC algorithms.

Gibbs sampling in Bayes Networks

function GIBBS-ASK(X, $\mathbf{e}$, bn, N) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  local variables: $\mathbf{C}$, a vector of counts for each value of X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initialized from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for $k = 1$ **to** N **do**
    choose any variable Z_i from $\mathbf{Z}$ according to any distribution $\rho(i)$
    set the value of Z_i in $\mathbf{x}$ by sampling from $\mathbf{P}(Z_i \mid mb(Z_i))$
    $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where x_j is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{C}$)

Random State

Change one variable

Count

- $mb(Z_i)$ is the Markov blanket of random variable Z_i (all variables it can be dependent of, i.e., parents, children and parents of children).

$$P(z_i|mb(Z_i)) = \alpha P(z_i|parents(Z_i)) \prod_{Y_i \in children(X_i)} P(y_i|parents(Y_j))$$

Gibbs Sampling: Example

Find
$P(Rain \,|Sprinkler \;= true, WetGrass \;= true)$.

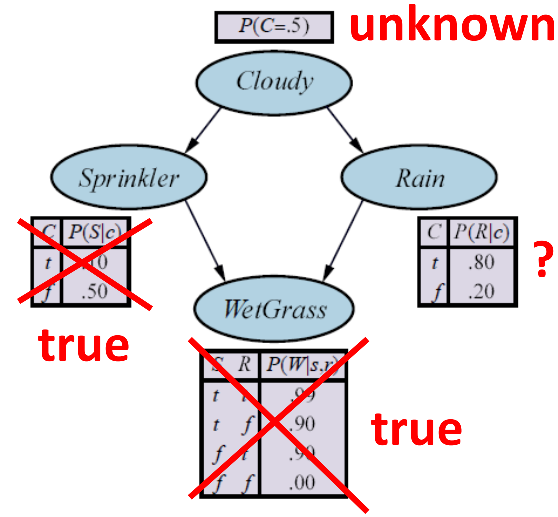
Determine states and calculate transition probabilities of the Markov chain for the query using $P(z_i|mb(Z_i))$.

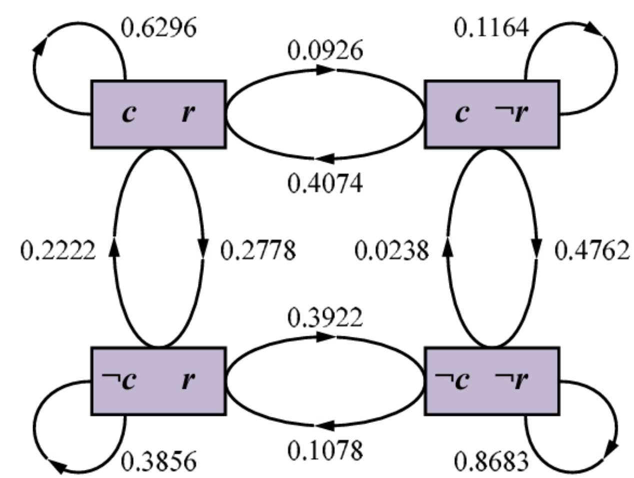
The algorithm wanders around in this graph using the stated transition probabilities.

Assume that we observe 20 states with $Rain = true$ and 60 with $rain = false$:

$$NORMALIZE(\langle 20,60\rangle) = \langle 0.25,0.75\rangle$$

$P(Rain \,|Sprinkler \;= true, WetGrass \;= true) \approx 0.75$





Note the self-loops: the state stays the same when either variable is chosen and then resamples the same value it already has.

Conclusion

- Bayesian networks provide an efficient way to store a probabilistic model by exploiting conditional independence.
- Inference (estimating conditional probabilities) is still difficult, for all but tiny models.
- State of the art is sampling from the model.