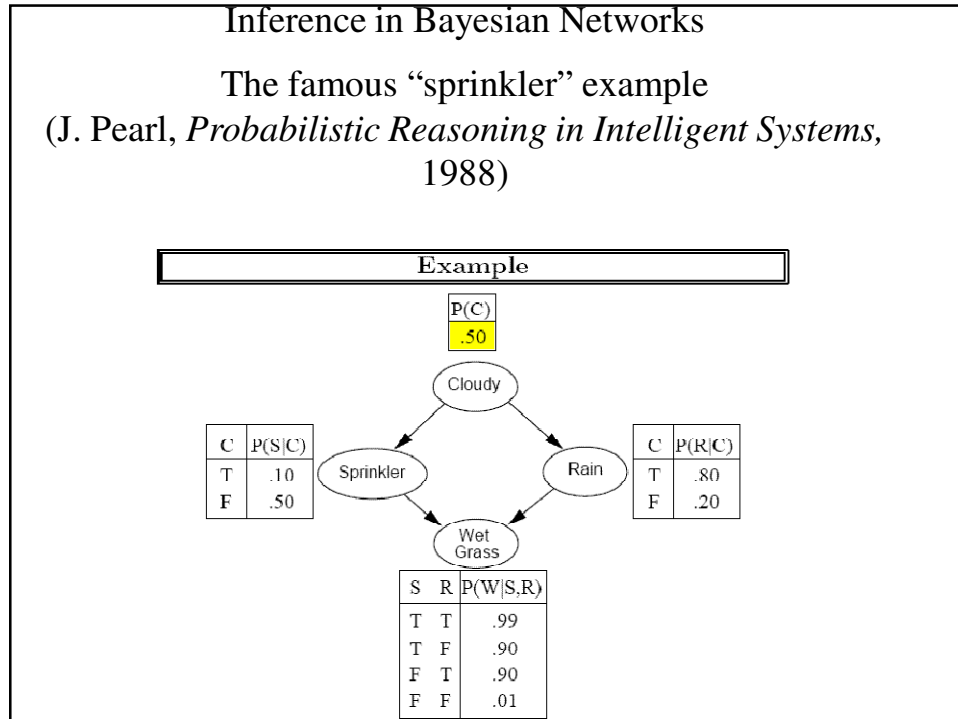


Inference in Bayesian Networks

The famous “sprinkler” example

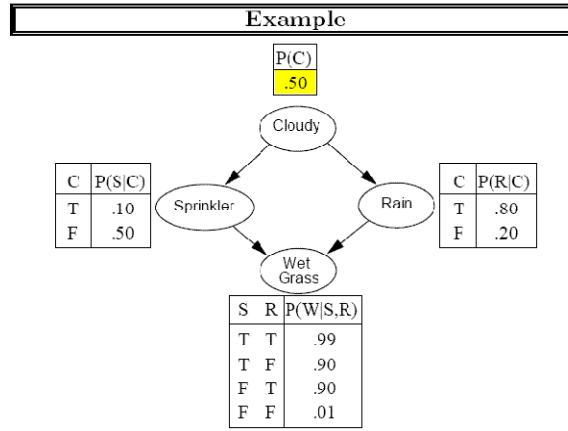
(J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, 1988)



Exercises

1. Suppose you observe it is cloudy and raining. What is the probability that the grass is wet?
2. Suppose you observe the sprinkler to be on and the grass is wet. What is the probability that it is raining?

Inference in Bayesian Networks



General question: What is $P(X|e)$?

Notation convention: upper-case letters refer to random variables;
lower-case letters refer to specific values of those variables

General question: Given query variable X and observed evidence variable values e , what is $P(X|e)$?

$$P(X|e) = \frac{P(X, e)}{P(e)} \quad (\text{definition of conditional probability})$$

$$= \alpha P(X, e) \quad \left(\alpha = \frac{1}{P(e)} \right)$$

$$= \alpha \sum_{\mathbf{y}} P(X, e, \mathbf{y}) \quad (\text{where } \mathbf{Y} \text{ are the non-evidence variables other than } X)$$

$$= \alpha \sum_{\mathbf{y}} \prod_{z \in \{X, e, \mathbf{y}\}} P(z | \text{parents}(Z)) \quad (\text{semantics of Bayesian networks})$$

Example: What is $P(c|r,w)$?

$$P(c|r,w) = \alpha \sum_s P(c,r,w,s)$$

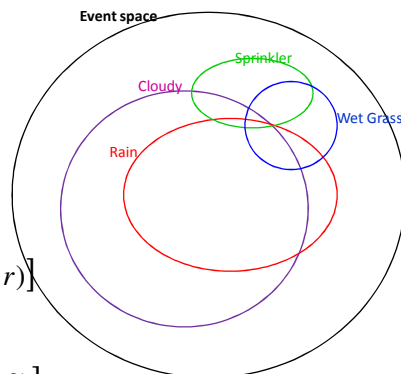
$$= \alpha \sum_s \prod_{z \in \{c,r,w,s\}} P(z | \text{parents}(Z))$$

$$= \alpha \sum_s [P(c)P(r|c)P(s|c)P(w|s,r)]$$

$$= \alpha [(.5 \times .8 \times .1 \times .99) + (.5 \times .8 \times .9 \times .9)]$$

$$= \alpha (.3636)$$

$$P(\neg c | r, w) = \alpha (.0945)$$



$$P(C | r, w) = \alpha \langle .3636, .0945 \rangle$$

$$= \left\langle \frac{.3636}{.3636 + .0945}, \frac{.0945}{.3636 + .0945} \right\rangle$$

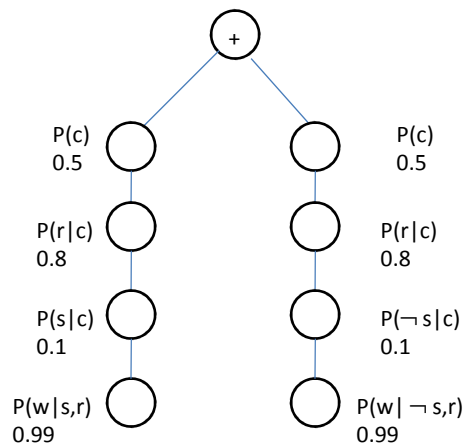
$$= \langle .794, .206 \rangle$$

- Worst-case complexity is exponential in n (number of nodes)
- Problem is having to enumerate all possibilities for many variables.

$$\alpha \sum_s [P(c)P(r|c)P(s|c)P(w|s,r)]$$

- Can solve by building tree

$$\alpha \sum_s [P(c)P(r|c)P(s|c)P(w|s,r)]$$

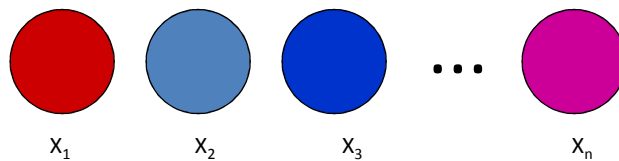


Can reduce computation by computing terms only once and storing for future use.
See “variable elimination algorithm” in reading.

- In general, however, exact inference in Bayesian networks is too expensive.

Approximate inference in Bayesian networks

Instead of enumerating all possibilities, sample to estimate probabilities.



Direct Sampling

- Suppose we have no evidence, but we want to determine $P(c,s,r,w)$ for all c,s,r,w .
- Direct sampling:
 - Sample each variable in topological order, conditioned on values of parents.
 - I.e., always sample from $P(X_i | \text{parents}(X_i))$

Example

1. Sample from $P(\text{Cloudy})$. Suppose returns *true*.
2. Sample from $P(\text{Sprinkler} | \text{Cloudy} = \text{true})$. Suppose returns *false*.
3. Sample from $P(\text{Rain} | \text{Cloudy} = \text{true})$. Suppose returns *true*.
4. Sample from $P(\text{WetGrass} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true})$. Suppose returns *true*.

Here is the sampled event: [*true, false, true, true*]

- Suppose there are N total samples, and let $N_S(x_1, \dots, x_n)$ be the observed frequency of the specific event x_1, \dots, x_n .

$$\lim_{N \rightarrow \infty} \frac{N_S(x_1, \dots, x_n)}{N} = P(x_1, \dots, x_n)$$

$$\frac{N_S(x_1, \dots, x_n)}{N} \approx P(x_1, \dots, x_n)$$

- Suppose N samples, n nodes. Complexity $O(Nn)$.
- Problem 1: Need *lots* of samples to get good probability estimates.
- Problem 2: Many samples are not realistic; low likelihood.

Likelihood weighting

- Now suppose we have evidence e . Thus values for the evidence variables E are fixed.
- We want to estimate $P(X | e)$
- Need to sample X and Y , where Y is the set of non-evidence variables.
- Each event sampled is weighted by the likelihood that that event accords to the evidence.
- I.e., events in which the actual evidence appears unlikely should be given less weight.

Example:

Estimate $P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$.

WeightedSample algorithm:

1. Set weight $w = 1.0$
2. Sample from *Cloudy*. Suppose it returns *true*.
3. *Sprinkler* is an evidence variable with value *true*. Update likelihood weighting:

$$w \leftarrow w \times P(\text{Sprinkler} = \text{true} \mid \text{Cloudy} = \text{true}) = 0.1$$

Low likelihood for sprinkler if cloudy is true, so this sample gets lower weight.

4. Sample from $P(\text{Rain} \mid \text{Cloudy} = \text{true})$. Suppose this returns *true*.
5. *WetGrass* is an evidence variable with value *true*. Update likelihood weighting:

$$w \leftarrow w \times P(\text{WetGrass} = \text{true} \mid \text{Sprinkler} = \text{true}, \text{Rain} = \text{true}) = 0.099$$

6. Return event $[\text{true}, \text{true}, \text{true}, \text{true}]$ with weight 0.099.

Weight is low because *cloudy* = *true*, so *sprinkler* is unlikely to be true.

Problem with likelihood sampling

- As number of evidence variables increases, performance degrades. This is because most samples will have very low weights, so weighted estimate will be dominated by fraction of samples that accord more than an infinitesimal likelihood to the evidence.

Markov Chain Monte Carlo Sampling

- One of most common methods used in real applications.
- Uses idea of “Markov blanket” of a variable X_i :
 - parents, children, children’s parents
- **Recall that:** By construction of Bayesian network, a node is conditionally independent of its non-descendants, given its parents.

- **Proposition:** A node X_i is conditionally independent of all other nodes in the network, given its Markov blanket.
 - Example.
 - Need to show that X_i is conditionally independent of nodes outside its Markov blanket.
 - Need to show that X_i can be conditionally dependent on children's parents.

Proof.

Markov Chain Monte Carlo Sampling Algorithm

- Start with random sample from variables: (x_1, \dots, x_n) . This is the current “state” of the algorithm.
- Next state: Randomly sample value for one non-evidence variable X_i , conditioned on current values in “Markov Blanket” of X_i .

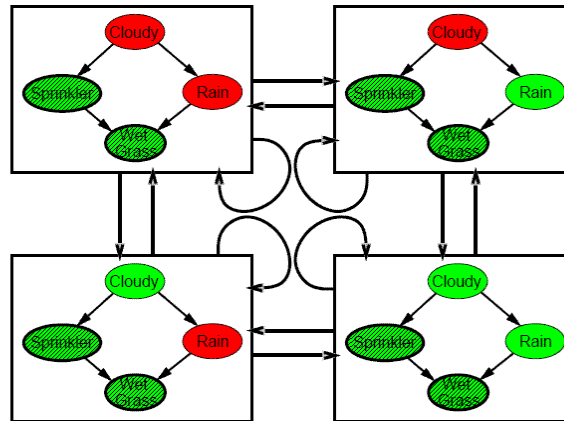
Example

- Query: What is $P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$?
- MCMC:
 - Random sample, with evidence variables fixed:
[true, true, false, true]
 - Repeat:
 1. Sample *Cloudy*, given current values of its Markov blanket:
Sprinkler = true, Rain = false. Suppose result is *false*. New state:
[false, true, false, true]
 2. Sample *Rain*, given current values of its Markov blanket:
Cloudy = false, Sprinkler = true, WetGrass = true. Suppose result is *true*. New state: [false, true, true, true].

- Each sample contributes to estimate for query
 $P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$
- Suppose we perform 100 such samples, 20 with *Rain = true* and 80 with *Rain = false*.
- Then answer to the query is
 $\text{Normalize}(\langle 20, 80 \rangle) = \langle .20, .80 \rangle$
- Claim: “The sampling process settles into a dynamic equilibrium in which the long-run fraction of time spent in each state is exactly proportional to its posterior probability, given the evidence.”
- Proof of claim is on pp. 517-518.

The Markov chain

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



Wander about for a while, average what you see

Claim (again)

- Claim: MCMC settles into behavior in which each state is sampled exactly according to its posterior probability, given the evidence.
- That is: for all variables X_i , the probability of the value x_i of X_i appearing in a sample is equal to $P(x_i | \mathbf{e})$.

Next time: Learning in Bayesian
Networks