Inferência em Redes Bayesianas

Paulo Santos

FEI - São Paulo

October 31, 2011

Outline I

1. Inferência exata
2. D-separation [Nilsson, 98]
3. Probabilistic inference in polytrees [Nilsson, 98]
4. Approximate inferences

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?

Inferência por enumeração

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Simple query on the burglary network: $P(B|j,m)$

$= P(B,j,m)/P(j,m)$

$= \alpha P(B,j,m)$

$= \alpha\ \Sigma_e\ \Sigma_a\ P(B,e,a,j,m)$

Rewrite full joint entries using product of CPT entries:

$P(B|j,m)$

$= \alpha\ \Sigma_e\ \Sigma_a\ P(B)P(e)P(a|B,e)P(j|a)P(m|a)$

$= \alpha P(B)\ \Sigma_e\ P(e)\ \Sigma_a\ P(a|B,e)P(j|a)P(m|a)$

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

Outro exemplo [Nilsson, 98]

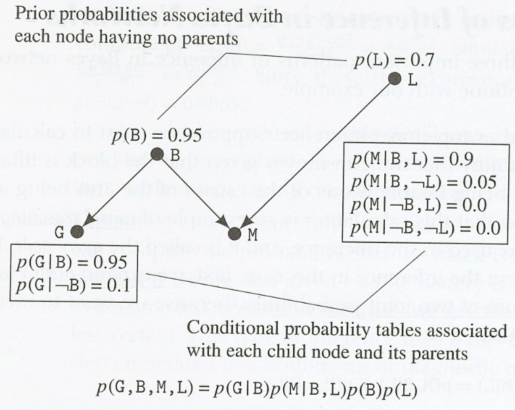


Figure 19.2

- B = battery charged; L = block is liftable; G = gauge indicates battery is charged; M = the arm moves

Patterns of inference

- causal (or top-down): e.g. $P(M|L)$
- diagnostic (or bottom-up): e.g. $P(\neg L|\neg M)$
- explaining away: e.g. $P(\neg L|\neg B, \neg M)$

Causal ($P(M|L)$)

- expand $P(M|L)$ into the sum of two joint probabilities (because we want to mention the other parent , B, of M)

$$P(M|L) = P(M, B|L) + P(M, \neg B|L)$$

- condition M on his other parent as well as on L, use chain rule

$$P(M|L) = P(M|B, L)P(B|L) + P(M|\neg B, L)P(\neg B|L)$$

- from the structure of the network: $P(B|L) = P(B)$ ($P(\neg B|L) = P(\neg B)$. Therefore:

$$P(M|L) = P(M|B, L)p(B) + P(M|\neg B, L)P(\neg B)$$

- from the TPC: $P(M|L) = 0.855$

Causal ($P(M|L)$)

- expand $P(M|L)$ into the sum of two joint probabilities (because we want to mention the other parent , B, of M)

$$P(M|L) = P(M, B|L) + P(M, \neg B|L)$$

- condition M on his other parent as well as on L, use chain rule

$$P(M|L) = P(M|B, L)P(B|L) + P(M|\neg B, L)P(\neg B|L)$$

- from the structure of the network: $P(B|L) = P(B)$ $(P(\neg B|L) = P(\neg B)$. Therefore:

$$P(M|L) = P(M|B, L)p(B) + P(M|\neg B, L)P(\neg B)$$

- from the TPC: $P(M|L) = 0.855$

Causal ($P(M|L)$)

- expand $P(M|L)$ into the sum of two joint probabilities (because we want to mention the other parent , B, of M)

$$P(M|L) = P(M, B|L) + P(M, \neg B|L)$$

- condition M on his other parent as well as on L, use chain rule

$$P(M|L) = P(M|B, L)P(B|L) + P(M|\neg B, L)P(\neg B|L)$$

- from the structure of the network: $P(B|L) = P(B)$ $(P(\neg B|L) = P(\neg B)$. Therefore:

$$P(M|L) = P(M|B, L)p(B) + P(M|\neg B, L)P(\neg B)$$

- from the TPC: $P(M|L) = 0.855$

Causal ($P(M|L)$)

- expand $P(M|L)$ into the sum of two joint probabilities (because we want to mention the other parent , B, of M)

$$P(M|L) = P(M, B|L) + P(M, \neg B|L)$$

- condition M on his other parent as well as on L, use chain rule

$$P(M|L) = P(M|B, L)P(B|L) + P(M|\neg B, L)P(\neg B|L)$$

- from the structure of the network: $P(B|L) = P(B)$ $(P(\neg B|L) = P(\neg B)$. Therefore:

$$P(M|L) = P(M|B, L)p(B) + P(M|\neg B, L)P(\neg B)$$

- from the TPC: $P(M|L) = 0.855$

Causal ($P(M|L)$): main operations

- rewrite the desired conditional probability of the query node, V, given the evidence, in terms of the joint probability of V and all of its parents (that are not evidence), given the evidence
- reexpress this joint probability back to the probability of V conditioned on all of the parents

Diagnostic ($P(\neg L|\neg M)$)

$P(\neg L|\neg M)$: effect (or symptom) to infer a cause

- use Bayes rule: $P(\neg L|\neg M)$:

$$P(\neg L|\neg M) = \frac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)}$$

- calculate $P(\neg M|\neg L)$ using causal reasoning: $P(\neg M|\neg L) = 0.145$
- compute $P(\neg L|\neg M) = \frac{0.145\times 0.3}{P(\neg M)} = 0.8863$
- **main step:** Bayes rule to convert the problem into a causal reasoning

Diagnostic ($P(\neg L|\neg M)$)

$P(\neg L|\neg M)$: effect (or symptom) to infer a cause

- use Bayes rule: $P(\neg L|\neg M)$:

$$P(\neg L|\neg M) = \frac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)}$$

- calculate $P(\neg M|\neg L)$ using causal reasoning: $P(\neg M|\neg L) = 0.145$
- compute $P(\neg L|\neg M) = \frac{0.145 \times 0.3}{P(\neg M)} = 0.8863$
- **main step:** Bayes rule to convert the problem into a causal reasoning

Diagnostic ($P(\neg L|\neg M)$)

$P(\neg L|\neg M)$: effect (or symptom) to infer a cause

- use Bayes rule: $P(\neg L|\neg M)$:

$$P(\neg L|\neg M) = \frac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)}$$

- calculate $P(\neg M|\neg L)$ using causal reasoning: $P(\neg M|\neg L) = 0.145$
- compute $P(\neg L|\neg M) = \frac{0.145 \times 0.3}{P(\neg M)} = 0.8863$
- **main step:** Bayes rule to convert the problem into a causal reasoning

Diagnostic ($P(\neg L|\neg M)$)

$P(\neg L|\neg M)$: effect (or symptom) to infer a cause

- use Bayes rule: $P(\neg L|\neg M)$:

$$P(\neg L|\neg M) = \frac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)}$$

- calculate $P(\neg M|\neg L)$ using causal reasoning: $P(\neg M|\neg L) = 0.145$
- compute $P(\neg L|\neg M) = \frac{0.145 \times 0.3}{P(\neg M)} = 0.8863$
- **main step:** Bayes rule to convert the problem into a causal reasoning

Explaining away ($P(\neg L|\neg B, \neg M)$)

$P(\neg L|\neg B, \neg M)$: "$\neg B$ explains $\neg M$ making $\neg L$ less certain"

Top-down embedded in a bottom-up reasoning

- use bayes rule:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, \neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- definition of conditional prob.:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- use the structure of Bayes Net:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B)P(\neg L)}{P(\neg B, \neg M)}$$

- using the probabilities in the TPC and solving $P(\neg B, \neg M)$, we get $P(\neg L|\neg B, \neg M) = 0.3$

Explaining away ($P(\neg L|\neg B, \neg M)$)

$P(\neg L|\neg B, \neg M)$: "$\neg B$ explains $\neg M$ making $\neg L$ less certain"

Top-down embedded in a bottom-up reasoning

- use bayes rule:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, \neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- definition of conditional prob.:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- use the structure of Bayes Net:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B)P(\neg L)}{P(\neg B, \neg M)}$$

- using the probabilities in the TPC and solving $P(\neg B, \neg M)$, we get $P(\neg L|\neg B, \neg M) = 0.3$

Explaining away ($P(\neg L|\neg B, \neg M)$)

$P(\neg L|\neg B, \neg M)$: "$\neg B$ explains $\neg M$ making $\neg L$ less certain"
Top-down embedded in a bottom-up reasoning

- use bayes rule:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, \neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- definition of conditional prob.:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- use the structure of Bayes Net:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B)P(\neg L)}{P(\neg B, \neg M)}$$

- using the probabilities in the TPC and solving $P(\neg B, \neg M)$, we get $P(\neg L|\neg B, \neg M) = 0.3$

Explaining away ($P(\neg L|\neg B, \neg M)$)

$P(\neg L|\neg B, \neg M)$: "$\neg B$ explains $\neg M$ making $\neg L$ less certain"
Top-down embedded in a bottom-up reasoning

- use bayes rule:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, \neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- definition of conditional prob.:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B, \neg M)}$$

- use the structure of Bayes Net:

$$P(\neg L|\neg B, \neg M) = \frac{P(\neg M, |\neg B \neg L)P(\neg B)P(\neg L)}{P(\neg B, \neg M)}$$

- using the probabilities in the TPC and solving $P(\neg B, \neg M)$, we get $P(\neg L|\neg B, \neg M) = 0.3$

D-separation

- a Bayes net implies more conditional probabilities than those implied by the parent nodes
- E.g. $P(M|G,B) = P(M|B)$: M is conditionally independent of G given B (even though we're not given both of M's parents)
- Knowledge of the effect G can influence knowledge about the cause B, which influences knowledge about another effect M.
- but if we are given the cause B there is nothing more that G can tell us about M. In this case B d-separates G and M

- a Bayes net implies more conditional probabilities than those implied by the parent nodes
- E.g. $P(M|G, B) = P(M|B)$: M is conditionally independent of G given B (even though we're not given both of M's parents)
- Knowledge of the effect G can influence knowledge about the cause B, which influences knowledge about another effect M.
- but if we are given the cause B there is nothing more that G can tell us about M. In this case B d-separates G and M

D-separation

Two node V_i and V_j are conditionally independent given a set of nodes E ($I(V_i, V_j|E)$) if for every undirected path in the Bayes net between V_i and V_j, there is some node, V_b, on the path having one of the following three properties:

1. V_b is in E, and both arcs on the path lead out of V_b;
2. V_b is in E, and one arc on the path leads in to V_b and one arc leads out;
3. Neither V_b or any descendant of V_b is in E, and both arcs on the path lead in to V_b

D-separation

- V_b blocks the path given E if any of the above holds.
- if all paths between V_i and V_j are blocked, we say that E d-separates V_i and V_j.
- therefore, V_i and V_j are conditionally independent given E

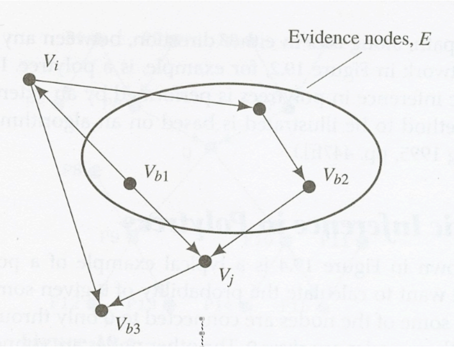
D-separation

- V_b blocks the path given E if any of the above holds.
- if all paths between V_i and V_j are blocked, we say that E d-separates V_i and V_j.
- therefore, V_i and V_j are conditionally independent given E

D-separation

- V_b blocks the path given E if any of the above holds.
- if all paths between V_i and V_j are blocked, we say that E d-separates V_i and V_j.
- therefore, V_i and V_j are conditionally independent given E

D-separation



- V_i is independent of V_j given the evidence, since all paths between them are blocked:
  - V_{b1} is evidence, and both arcs lead out of it
  - V_{b2} is evidence, and one arc leads in and one arc leads out
  - V_{b3} is not evidence, nor are any of its descendants, and both arcs lead into it.

Patterns of inference

- all evidence nodes are above the query
- all evidence nodes are below the query
- there are evidence above and below.

Probabilistic inference in polytrees

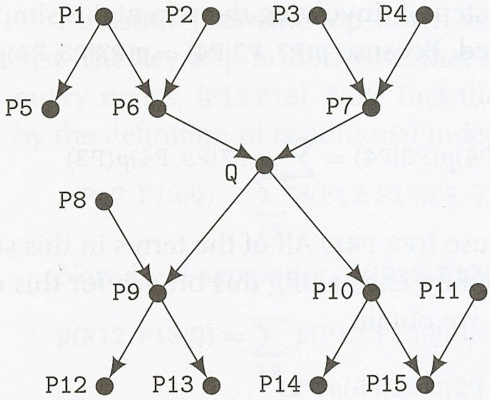


Figure 19.4

Evidence above: $P(Q|P5, P4)$

Bottom-up recursive algorithm that calculates the probability of each of the ancestors of Q, given the evidence, until the evidence is reached or is below that ancestor.

- invove the parents of Q:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q, P6, P7|P5, P4)$
- make the parents part of the evidence using the definition of conditional probability:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7, P5, P4)P(P6, P7|P5, P4)$
- since a node is conditionally independent of non-descendants given their parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6, P7|P5, P4)$

Evidence above: $P(Q|P5, P4)$

Bottom-up recursive algorithm that calculates the probability of each of the ancestors of Q, given the evidence, until the evidence is reached or is below that ancestor.

- invove the parents of Q:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q, P6, P7|P5, P4)$
- make the parents part of the evidence using the definition of conditional probability:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7, P5, P4) P(P6, P7|P5, P4)$
- since a node is conditionally independent of non-descendants given their parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6, P7|P5, P4)$

Evidence above: $P(Q|P5, P4)$

Bottom-up recursive algorithm that calculates the probability of each of the ancestors of Q, given the evidence, until the evidence is reached or is below that ancestor.

- invove the parents of Q:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q, P6, P7|P5, P4)$
- make the parents part of the evidence using the definition of conditional probability:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7, P5, P4) P(P6, P7|P5, P4)$
- since a node is conditionally independent of non-descendants given their parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6, P7|P5, P4)$

Evidence above: $P(Q|P5, P4)$

... continued

- d-separation allows us to split the parents: $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5, P4)P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent: $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5)P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2)P(P1|P5)P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule: $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

Evidence above: $P(Q|P5, P4)$

... continued

- d-separation allows us to split the parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5, P4) P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5) P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2) P(P1|P5) P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule:
    $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

Evidence above: $P(Q|P5, P4)$

... continued

- d-separation allows us to split the parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5, P4) P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5) P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2) P(P1|P5) P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule:
    $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

Evidence above: $P(Q|P5, P4)$

... continued

- d-separation allows us to split the parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5, P4)P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5)P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2)P(P1|P5)P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule:
    $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

Evidence above: $P(Q|P5, P4)$

... continued

- d-separation allows us to split the parents:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5, P4)P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent:
  $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7)P(P6|P5)P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3|P4) = \Sigma_{P3} P(P7|P3, P4)P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2)P(P1|P5)P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule:
    $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

... continued

- d-separation allows us to split the parents: $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5, P4) P(P7|P5, P4)$
- d-separation allows us to ignore the evidence above one parent: $P(Q|P5, P4) = \Sigma_{P6,P7} P(Q|P6, P7) P(P6|P5) P(P7|P4)$
  - $P(P7|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3|P4) = \Sigma_{P3} P(P7|P3, P4) P(P3)$
  - $P(P6|P5) = \Sigma_{P1,P2} P(P6|P1, P2) P(P1|P5) P(P2)$
  - $P(P1|P5)$... the evidence is now below. Simply use Bayes rule: $P(P1|P5) = \frac{P(P5|P1)P(P1)}{P(P5)}$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$
$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation: $P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$

$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation: $P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$
$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation: $P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$

$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation: $P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$
$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation: $P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- use Bayes rule:

$$P(Q|P12, P13, P14, P11) = \frac{P(P12, P13, P14, P11|Q)P(Q)}{P(P12, P13, P14, P11)}$$

- by d-separation: $I(\{P12, P13\}, \{P14, P11\})$:

$$P(Q|P12, P13, P14, P11) = kP(P12, P13|Q)P(P14, P11|Q)P(Q)$$

- $P(P12, P13|Q)$:
  - include a single child of Q:

$$P(P12, P13|Q) = \Sigma_{P9} P(P12, P13, P9|Q)$$

$$= \Sigma_{P9} P(P12, P13|P9, Q)P(P9|Q)$$

  - by d-separation:$P(P12, P13, P9|Q) = \Sigma_{P9} P(P12, P13|P9)P(P9|Q)$
  - similarly: $P(P9|Q) = \Sigma_{P8} P(P9|P8, Q)P(P8)$

Evidence below: $P(Q|P12, P13, P14, P11)$

EXERCISE: do a similar procedure to obtain:
$P(P12, P13|P9) = P(P12|P9)P(P12|P9)$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10) P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10) P(P11|P10) P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10) P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11) P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11) P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11) P(P10|P11) = P(P15|P10, P11$$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10)P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10)P(P11|P10)P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10)P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11)P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11)P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11)P(P10|P11) = P(P15|P10, P11$$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10) P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10) P(P11|P10) P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10) P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11) P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11) P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11) P(P10|P11) = P(P15|P10, P11$$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10)P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10)P(P11|P10)P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10)P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11)P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11)P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11)P(P10|P11) = P(P15|P10, P11$$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10)P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10)P(P11|P10)P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10)P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11)P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11)P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11)P(P10|P11) = P(P15|P10, P11$$

Evidence below: $P(Q|P12, P13, P14, P11)$

- $P(P14, P11|Q) = \Sigma_{P10} P(P14, P11|P10)P(P10|Q)$
- $P(P14, P11|Q) = \Sigma_{P10} P(P14|P10)P(P11|P10)P(P10|Q)$
  - $P(P11|P10) = \Sigma_{P15} P(P11|P15, P10)P(P15|P10)$
  - where $P(P15|P10) = \Sigma P(P15|P10, P11)P(P11))$
  - but in $P(P11|P15, P10)$ the query node is above the evidence, so we apply the Bayes Rule:

$$P(P11|P15, P10) = \frac{P(P15, P10|P11)P(P11)}{P(P15, P10)}$$

  - the algorithm terminates with:

$$P(P15, P10|P11) = P(P15|P10, P11)P(P10|P11) = P(P15|P10, P11$$

Evidence above and below: $P(Q|\{P5, P4\}, \{P14, P11\})$

- Separate tehe evidence into above E^+ and below E^- and use a version of the Bayes rule to write:

$$P(Q|E^+, E^-) = \frac{P(E^-|Q, E^+)P(Q|E^+)}{P(E^-|E^+)}$$

- Q d-separates E^- from E^+, so

$$P(Q|E^+, E^-) = k_2 P(E^-|Q)P(Q|E^+)$$

- we use the appropriate procedures in each care.

Evidence above and below: $P(Q|\{P5, P4\}, \{P14, P11\})$

- Separate tehe evidence into above E^+ and below E^- and use a version of the Bayes rule to write:

$$P(Q|E^+, E^-) = \frac{P(E^-|Q, E^+)P(Q|E^+)}{P(E^-|E^+)}$$

- Q d-separates E^- from E^+, so

$$P(Q|E^+, E^-) = k_2 P(E^-|Q)P(Q|E^+)$$

- we use the appropriate procedures in each care.

Evidence above and below: $P(Q|\{P5, P4\}, \{P14, P11\})$

- Separate tehe evidence into above E^+ and below E^- and use a version of the Bayes rule to write:

$$P(Q|E^+, E^-) = \frac{P(E^-|Q, E^+)P(Q|E^+)}{P(E^-|E^+)}$$

- Q d-separates E^- from E^+, so

$$P(Q|E^+, E^-) = k_2 P(E^-|Q)P(Q|E^+)$$

- we use the appropriate procedures in each care.

Inference by stochastic simulation

Basic Idea

- Draw N samples from a sampling distribution S
- Compute an approximate posterior probability $\hat{P}$
- Show this converges to the true probability P

Outline: randomized sampling (Monte Carlo Alg.)

- 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

Inference by stochastic simulation

Basic Idea

- Draw N samples from a sampling distribution S
- Compute an approximate posterior probability $\hat{P}$
- Show this converges to the true probability P

Outline: randomized sampling (Monte Carlo Alg.)

- 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

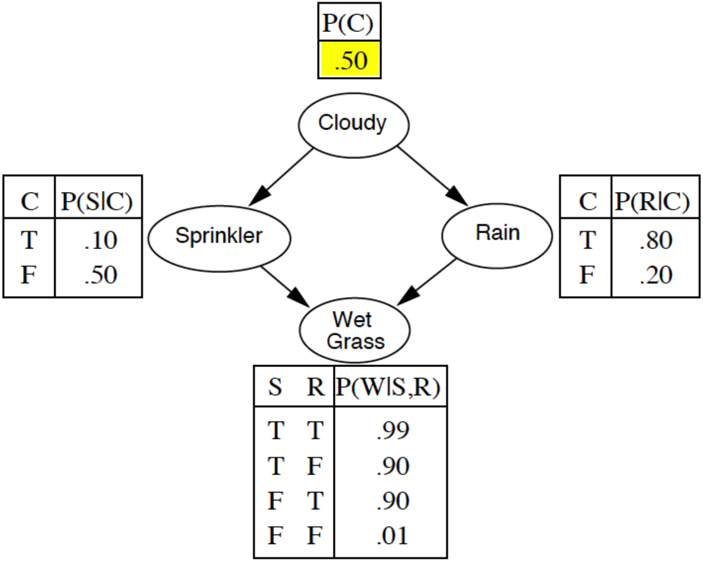
function PRIOR-SAMPLE(bn) **returns** an event sampled from bn
    inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

    $\mathbf{x} \leftarrow$ 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 $\mathbf{x}$

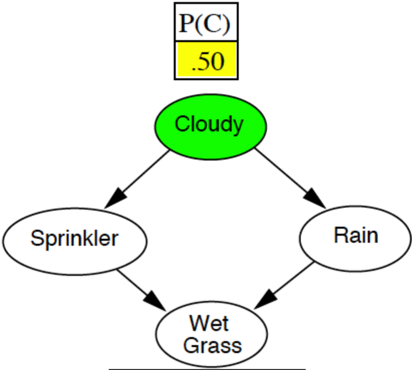
Sampling from an empty network

- generates events from a network that has no evidence associated to it
- sample each variable in turn, in topological order
- the probability distribution from which the value is sampled is conditioned on the values already assigned to the variable's parents

Sampling from an empty network



Sampling from an empty network

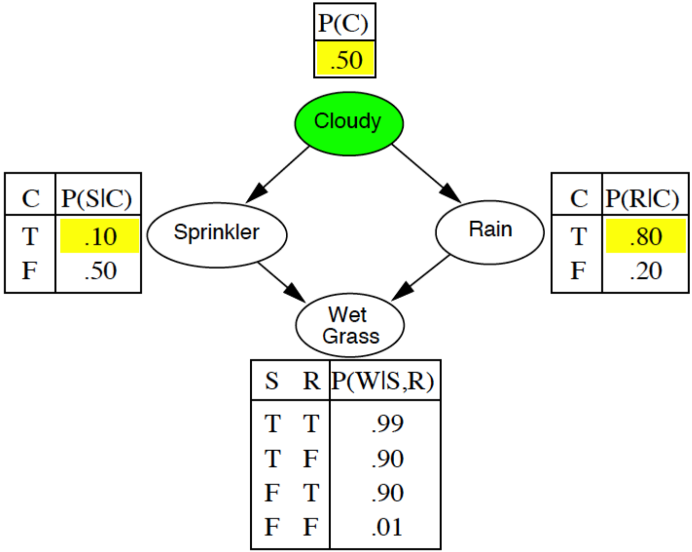


C	P(S|C)
T	.10
F	.50

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

Sampling from an empty network



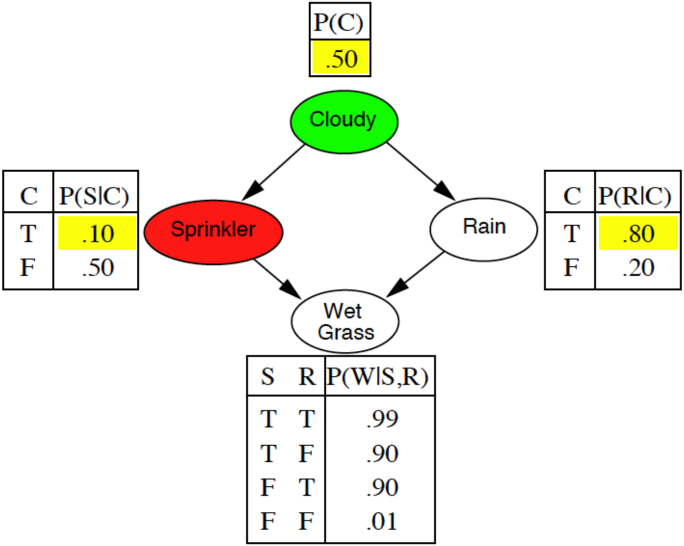
P(C)
.50

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

Sampling from an empty network



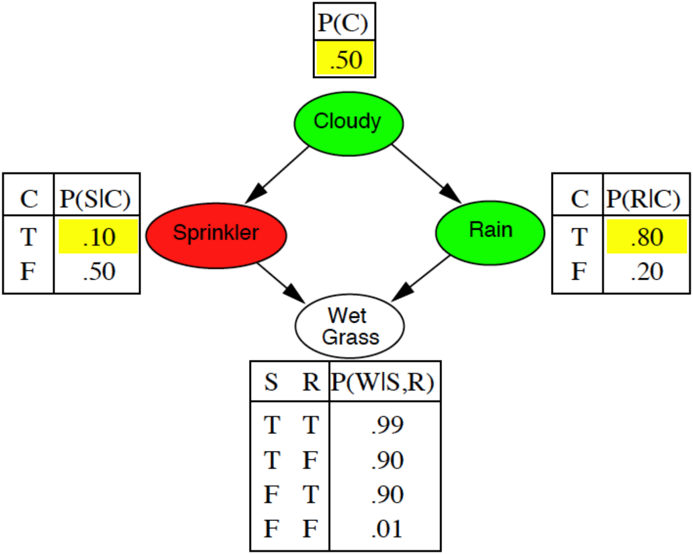
P(C)
.50

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

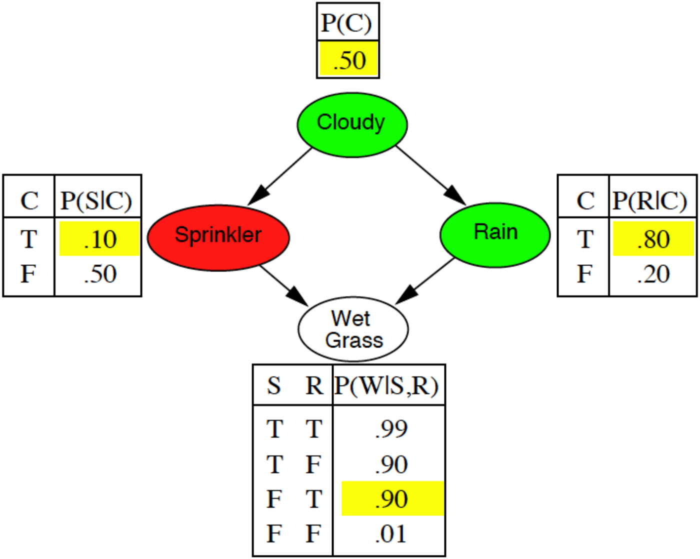
C	P(R|C)
T	.80
F	.20

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

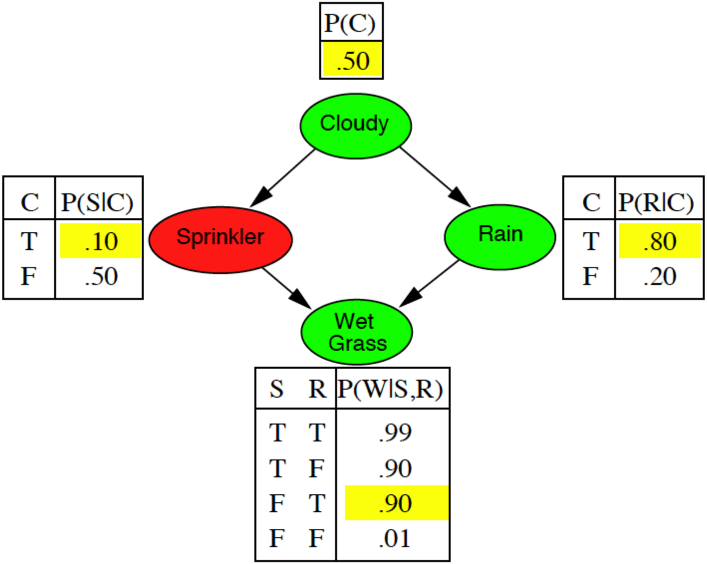
Sampling from an empty network



Sampling from an empty network



Sampling from an empty network



Sampling from an empty network

- from the example, Prior-sample returns the event $[C = true, S = false, R = true, W = true]$
- let be the probability that a specific event is generated by Prior-sample. Looking at the sampling process:

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

- but by construction of the Bayes net, this is the probability of the event: $S_{PS}(x_1, \ldots x_n) = P(x_1, \ldots x_n)$

Sampling from an empty network

- from the example, Prior-sample returns the event $[C = true, S = false, R = true, W = true]$
- let be the probability that a specific event is generated by Prior-sample. Looking at the sampling process:

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

- but by construction of the Bayes net, this is the probability of the event: $S_{PS}(x_1, \ldots x_n) = P(x_1, \ldots x_n)$

Sampling from an empty network

- from the example, Prior-sample returns the event $[C = \mathit{true}, S = \mathit{false}, R = \mathit{true}, W = \mathit{true}]$
- let be the probability that a specific event is generated by Prior-sample. Looking at the sampling process:

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

- but by construction of the Bayes net, this is the probability of the event: $S_{PS}(x_1, \ldots x_n) = P(x_1, \ldots x_n)$

Sampling from an empty network

- in any sampling alg., the answers are computed by counting the actual samples generated
- suppose there are N total samples, and let $N_{PS}(x_1, \ldots, x_n)$ be the number of times the specific event $x_1, \ldots, x_n$ occurs in the set of samples
- we expect this number to converge in the limit to its expected value according to the sampling probability:

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

- in the example, the sample is generated with probability

$$S_{PS}(true, false, true, true) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324$$

- **so, in the limit of large N, we expect 32.4% of the samples to be of this event**

Sampling from an empty network

- in any sampling alg., the answers are computed by counting the actual samples generated
- suppose there are N total samples, and let $N_{PS}(x_1, \ldots, x_n)$ be the number of times the specific event $x_1, \ldots, x_n$ occurs in the set of samples
- we expect this number to converge in the limit to its expected value according to the sampling probability:

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

- in the example, the sample is generated with probability

$$S_{PS}(true, false, true, true) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324$$

- **so, in the limit of large N, we expect 32.4% of the samples to be of this event**

Sampling from an empty network

- in any sampling alg., the answers are computed by counting the actual samples generated
- suppose there are N total samples, and let $N_{PS}(x_1, \ldots, x_n)$ be the number of times the specific event $x_1, \ldots, x_n$ occurs in the set of samples
- we expect this number to converge in the limit to its expected value according to the sampling probability:

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

- in the example, the sample is generated with probability

$$S_{PS}(true, false, true, true) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324$$

- **so, in the limit of large N, we expect 32.4% of the samples to be of this event**

Sampling from an empty network

- in any sampling alg., the answers are computed by counting the actual samples generated
- suppose there are N total samples, and let $N_{PS}(x_1, \ldots, x_n)$ be the number of times the specific event $x_1, \ldots, x_n$ occurs in the set of samples
- we expect this number to converge in the limit to its expected value according to the sampling probability:

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

- in the example, the sample is generated with probability

$$S_{PS}(true, false, true, true) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324$$

- **so, in the limit of large N, we expect 32.4% of the samples to be of this event**

Sampling from an empty network

- **key point:** the probability of any event can be estimated as a fraction of *all complete events]* generated by the sampling process that where the event happens
- E.g. if we generate 1000 samples from the sprinkler network, and 511 of them have rain = true, then the estimated probability of rain ($\hat{P}(Rain = true)$) is 0.511

Sampling from an empty network

- **key point:** the probability of any event can be estimated as a fraction of *all complete events]* generated by the sampling process that where the event happens
- E.g. if we generate 1000 samples from the sprinkler network, and 511 of them have rain = true, then the estimated probability of rain ($\hat{P}(Rain = true)$) is 0.511

Rejection Sampling

- used to compute conditional probabilities: $P(X|e)$
- first, it generates samples from the prior distribution on the network
- second: rejects all those that do not match the evidence
- finally: $\hat{P}(X = x|e)$ is obtained be counting how often $X = x$ occurs in the sample set.

Rejection Sampling

- used to compute conditional probabilities: $P(X|e)$
- first, it generates samples from the prior distribution on the network
- second: rejects all those that do not match the evidence
- finally: $\hat{P}(X = x|e)$ is obtained be counting how often $X = x$ occurs in the sample set.

Rejection Sampling

- used to compute conditional probabilities: $P(X|e)$
- first, it generates samples from the prior distribution on the network
- second: rejects all those that do not match the evidence
- finally: $\hat{P}(X = x|e)$ is obtained be counting how often $X = x$ occurs in the sample set.

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 $\mathbf{x}$

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

- $P(Rain|Sprinkler = true) =?$
- estimate using 100 samples
- of the 100 generated, 73 have $Sprinkler = false$ and are rejected, while 27 have e $Sprinkler = true$
- of the 27, 8 have $Rain = true$, and 19 have $Rain = false$
- $P(Rain|Sprinkler = true) = NORMALISE(< 8, 19 >) =< 0.296, 0.704 >$
- the true answer is $< 0.3, 0.7 >$
- standard deviation of the error $\approx 1/\sqrt{n}$, where n is the number of samples used in the estimate.

Rejection Sampling

Problems:

- rejects too many samples: expensive if $P(e)$ is small
- fraction of samples consistent with $P(e)$ drops exponentially as the number of evidence variables grows: unusable for complex problems

Rejection Sampling

Problems:

- rejects too many samples: expensive if $P(e)$ is small
- fraction of samples consistent with $P(e)$ drops exponentially as the number of evidence variables grows: unusable for complex problems

Rejection Sampling

Problems:

- rejects too many samples: expensive if $P(e)$ is small
- fraction of samples consistent with $P(e)$ drops exponentially as the number of evidence variables grows: unusable for complex problems

Likelihood weighting

- avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence
- particular instance of **importance sampling**
- idea: fix evidence variables, sample only nonevidence variables
- weight each sample by the likelyhood it accords to the evidence
  - as measured by the product of the conditional probabilities for each evidence variable, given its parents

Likelihood weighting

- avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence
- particular instance of **importance sampling**
- idea: fix evidence variables, sample only nonevidence variables
- weight each sample by the likelyhood it accords to the evidence
  - as measured by the product of the conditional probabilities for each evidence variable, given its parents

Likelihood weighting

- avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence
- particular instance of **importance sampling**
- idea: fix evidence variables, sample only nonevidence variables
- weight each sample by the likelyhood it accords to the evidence
  - as measured by the product of the conditional probabilities for each evidence variable, given its parents

Likelihood weighting

- avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence
- particular instance of **importance sampling**
- idea: fix evidence variables, sample only nonevidence variables
- weight each sample by the likelyhood it accords to the evidence
  - as measured by the product of the conditional probabilities for each evidence variable, given its parents

Likelihood weighting

- avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence
- particular instance of **importance sampling**
- idea: fix evidence variables, sample only nonevidence variables
- weight each sample by the likelyhood it accords to the evidence
  - as measured by the product of the conditional probabilities for each evidence variable, given its parents

Likelihood weighting

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

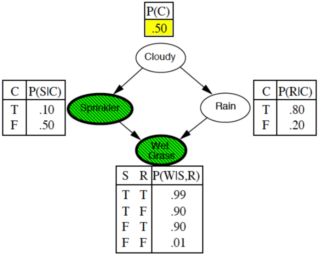
  for $j = 1$ to N **do**
    x, $w \leftarrow$ WEIGHTED-SAMPLE(bn)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in **x**
  return NORMALIZE($\mathbf{W}[X]$)

---

function WEIGHTED-SAMPLE(bn, **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 **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 **x**, w

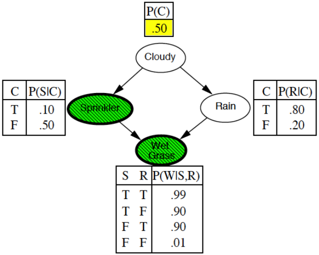
Likelihood weighting



$w = 1.0$

- **query:** $P(Rain|Cloudy = true, WetGrass = true)$
- the weight w is set to 1;
- $Cloudy$ is evidence:

$$w \leftarrow w \times P(Cloudy = true) = 0.5$$

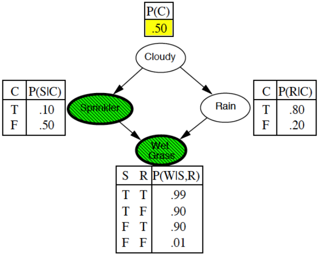


$w = 1.0$

- **query:** $P(Rain|Cloudy = true, WetGrass = true)$
- the weight w is set to 1;
- *Cloudy* is evidence:

$$w \leftarrow w \times P(Cloudy = true) = 0.5$$

Likelihood weighting

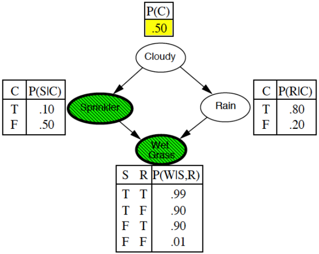


$w = 1.0$

- **query:** $P(Rain|Cloudy = true, WetGrass = true)$
- the weight w is set to 1;
- *Cloudy* is evidence:

$$w \leftarrow w \times P(Cloudy = true) = 0.5$$

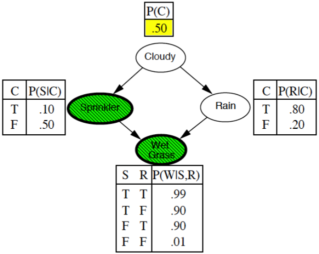
Likelihood weighting



$w = 1.0$

- *Sprinkler* is not evidence, so sample from $P(\mathit{Sprinkler}|\mathit{Cloudy} = \mathit{true}) =< 0.1, 0.9 >$suppose this returns *false*
- sample from $P(\mathit{Rain}|\mathit{Cloudy} = \mathit{true}) =< 0.8, 0.2 >$; suppose this returns *true*.

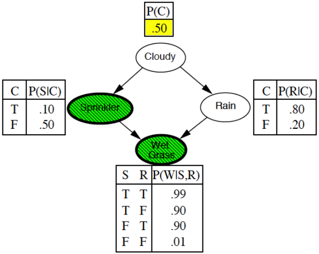
Likelihood weighting



$w = 1.0$

- *Sprinkler* is not evidence, so sample from $P(Sprinkler|Cloudy = true) =< 0.1, 0.9 >$suppose this returns *false*
- sample from $P(Rain|Cloudy = true) =< 0.8, 0.2 >$; suppose this returns *true*.

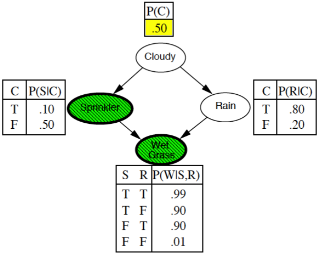
Likelihood weighting



$w = 1.0$

- *Sprinkler* is not evidence, so sample from $P(Sprinkler|Cloudy = true) =< 0.1, 0.9 >$suppose this returns *false*
- sample from $P(Rain|Cloudy = true) =< 0.8, 0.2 >$; suppose this returns *true*.

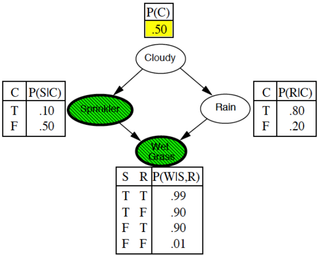
Likelihood weighting



$w = 1.0$

- *Sprinkler* is not evidence, so sample from $P(Sprinkler|Cloudy = true) =< 0.1, 0.9 >$suppose this returns *false*
- sample from $P(Rain|Cloudy = true) =< 0.8, 0.2 >$; suppose this returns *true*.

Likelihood weighting



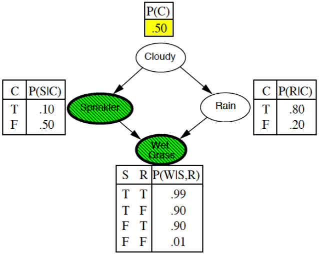
$w = 1.0$

- *WetGrass* is an evidence variable with value *true*:

  $w \leftarrow w \times P(WetGrass = true | Sprinkler = false, Rain = true) = 0.45$

- Here Weighted-Sample returns the event [*true*, *false*, *true*, *true*] with weight 0.45, and is associated to $Rain = true$

Likelihood weighting



$w = 1.0$

- *WetGrass* is an evidence variable with value *true*:

  $$w \leftarrow w \times P(WetGrass = true | Sprinkler = false, Rain = true) = 0.45$$

- Here Weighted-Sample returns the event [*true*, *false*, *true*, *true*] with weight 0.45, and is associated to $Rain = true$

Likelihood weighting: example1

consider two urns A and B, one with 2 black balls and one white, and the other with 3 black and 1 white [Neapolitan, 2004]

- $P_A(black)$ is the probability of getting a black ball on urn A
- $P_B(black)$ is the probability of getting a black ball on urn B
- estimate $P_A(black)$ by sampling with replacement **from urn B**, using Likelyhood weighting

Likelihood weighting: example1

consider two urns A and B, one with 2 black balls and one white, and the other with 3 black and 1 white [Neapolitan, 2004]

- $P_A(black)$ is the probability of getting a black ball on urn A
- $P_B(black)$ is the probability of getting a black ball on urn B
- estimate $P_A(black)$ by sampling with replacement **from urn B**, using Likelyhood weighting

Likelihood weighting: example1

Suppose we create a sample of of size m by sampling from A, and k of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m\to\infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m\to\infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

Suppose we create a sample of of size *m* by sampling from A, and *k* of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m \to \infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m \to \infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

Suppose we create a sample of of size m by sampling from A, and k of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m \to \infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m \to \infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

Suppose we create a sample of of size m by sampling from A, and k of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m \to \infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m \to \infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

Suppose we create a sample of of size m by sampling from A, and k of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m \to \infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m \to \infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

Suppose we create a sample of of size m by sampling from A, and k of the balls are black. Instead of adding 1 each time we sample a black ball, we add $score(black)$, where

- $score(black) = \frac{P_A(black)}{P_B(black)}$
- by adding the score(balck) each time a black ball is sampled: $\frac{k \times score(black)}{m}$
- in the limit we have:
  - $$\lim_{m\to\infty} \frac{k \times score(black)}{m} =$$
  - $$= \frac{P_A(black)}{P_B(black)} \lim_{m\to\infty} \frac{k}{m}$$
  - $$= \frac{P_A(black)}{P_B(black)} P_B(black)$$

Likelihood weighting: example1

- Therefore, if we use $\frac{k \times score(black)}{m}$ as an estimate of $P(black)$ we have convergence
- however, for any finite sample:

$$\frac{k \times score(black)}{m} + \frac{(m-k) \times score(white)}{m} = 1$$

- i.e., results should be normalised.

Likelihood weighting: example1

- Therefore, if we use $\frac{k \times score(black)}{m}$ as an estimate of $P(black)$ we have convergence
- however, for any finite sample:

$$\frac{k \times score(black)}{m} + \frac{(m-k) \times score(white)}{m} = 1$$

- i.e., results should be normalised.

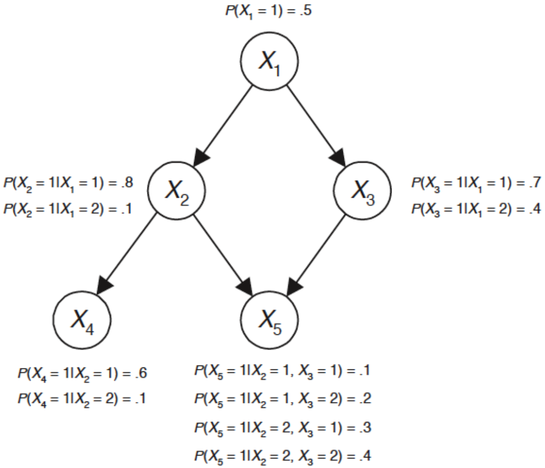
Likelihood weighting: example1

- Therefore, if we use $\frac{k \times score(black)}{m}$ as an estimate of $P(black)$ we have convergence
- however, for any finite sample:

$$\frac{k \times score(black)}{m} + \frac{(m-k) \times score(white)}{m} = 1$$

- i.e., results should be normalised.

Likelihood weighting: example2



- Task: estimate $P(X_j = x_j | X_3 = 1, X_4 = 2)$ for j = 1, 2, and 5; using likelihood sampling.

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- first, fix X_3 to 1 and X_4 to 2;
- then, generate values of the other var, according to the distributions in the network.
- Initially, generate a value for X_1 according to $P(X_1) = 0.5$ ... let's say we get $X_1 = 2$;
- then, generate a value for X_2 according to $P(X_2 = 1|X_1 = 2) = 0.1$ ... let's say we get $X_2 = 2$;
- finally, generate a value for X_5 according to $P(X_5 = 1|X_2 = 2, X_3 = 1) = 0.3$... let's say we get $X_5 = 1$;

Likelihood weighting: example2

- now calculate the score of each case. For instance:

$$score(Case1) = score(X_1 = 2, X_2 = 2, X_5 = 1) =$$

$$= P(X_4 = 2|X_2 = 2)P(X_3 = 1|X_1 = 2) = 0.9 \times 0.4 = 0.36$$

$Case$	X_1	X_2	X_3	X_4	X_5	$score'$
1	2	2	1	2	1	.36
2	1	1	1	2	2	.28
3	2	1	1	2	2	.16
4	1	1	1	2	1	.28

Likelihood weighting: example2

- now calculate the score of each case. For instance:

$$score(Case1) = score(X_1 = 2, X_2 = 2, X_5 = 1) =$$

$$= P(X_4 = 2|X_2 = 2)P(X_3 = 1|X_1 = 2) = 0.9 \times 0.4 = 0.36$$

Case	X_1	X_2	X_3	X_4	X_5	$score'$
1	2	2	1	2	1	.36
2	1	1	1	2	2	.28
3	2	1	1	2	2	.16
4	1	1	1	2	1	.28

Likelihood weighting: example2

- now calculate the score of each case. For instance:

$$score(Case1) = score(X_1 = 2, X_2 = 2, X_5 = 1) =$$

$$= P(X_4 = 2 | X_2 = 2) P(X_3 = 1 | X_1 = 2) = 0.9 \times 0.4 = 0.36$$

Case	X_1	X_2	X_3	X_4	X_5	*score'*
1	2	2	1	2	1	.36
2	1	1	1	2	2	.28
3	2	1	1	2	2	.16
4	1	1	1	2	1	.28

Likelihood weighting: example2

Finally, we estimate the conditional probability of the value of any particular variable by normalising the sum of the scores for the cases containing that variable. For example,

- $\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) \propto [score(Case2) + score(Case4)] = .28 + .28 = .56$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto [score(Case1) + score(Case3)] = .36 + .16 = .52$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto \frac{.56}{.56+.52} = .52$
- *avette capito adesso?*

Likelihood weighting: example2

Finally, we estimate the conditional probability of the value of any particular variable by normalising the sum of the scores for the cases containing that variable. For example,

- $\hat{P}(X_1 = 1|X_3 = 1, X_4 = 2) \propto [score(Case2) + score(Case4)] = .28 + .28 = .56$
- $\hat{P}(X_1 = 2|X_3 = 1, X_4 = 2) \propto [score(Case1) + score(Case3)] = .36 + .16 = .52$
- $\hat{P}(X_1 = 2|X_3 = 1, X_4 = 2) \propto \frac{.56}{.56+.52} = .52$
- *avette capito adesso?*

Likelihood weighting: example2

Finally, we estimate the conditional probability of the value of any particular variable by normalising the sum of the scores for the cases containing that variable. For example,

- $\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) \propto [score(Case2) + score(Case4)] = .28 + .28 = .56$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto [score(Case1) + score(Case3)] = .36 + .16 = .52$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto \frac{.56}{.56 + .52} = .52$
- *avette capito adesso?*

Likelihood weighting: example2

Finally, we estimate the conditional probability of the value of any particular variable by normalising the sum of the scores for the cases containing that variable. For example,

- $\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) \propto [score(Case2) + score(Case4)] = .28 + .28 = .56$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto [score(Case1) + score(Case3)] = .36 + .16 = .52$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto \frac{.56}{.56 + .52} = .52$
- *avette capito adesso?*

Likelihood weighting: example2

Finally, we estimate the conditional probability of the value of any particular variable by normalising the sum of the scores for the cases containing that variable. For example,

- $\hat{P}(X_1 = 1 | X_3 = 1, X_4 = 2) \propto [score(Case2) + score(Case4)] = .28 + .28 = .56$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto [score(Case1) + score(Case3)] = .36 + .16 = .52$
- $\hat{P}(X_1 = 2 | X_3 = 1, X_4 = 2) \propto \frac{.56}{.56 + .52} = .52$
- *avette capito adesso?*

Likelihood weighting

- suffers degradation in performance as the number of evidence variables increases

Markov Chain Monte Carlo (MCMC)

- generate each sample by making random change to the preceding sample
- helpful to think MCMC alg. as being in a particular current state specifying a value for every variable and generating a next state by making random changes to the current state.
- Gibbs sampling (special case of MCMC) is applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known and is easy (or at least, easier) to sample from

Markov Chain Monte Carlo (MCMC)

- generate each sample by making random change to the preceding sample
- helpful to think MCMC alg. as being in a particular current state specifying a value for every variable and generating a next state by making random changes to the current state.
- Gibbs sampling (special case of MCMC) is applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known and is easy (or at least, easier) to sample from

Markov Chain Monte Carlo (MCMC)

- generate each sample by making random change to the preceding sample
- helpful to think MCMC alg. as being in a particular current state specifying a value for every variable and generating a next state by making random changes to the current state.
- Gibbs sampling (special case of MCMC) is applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known and is easy (or at least, easier) to sample from

Gibbs sampling

- starts at an arbitrary state (with the evidence variables fixed)
- generates a next state by randomly sampling a value for one of the non-evidence variables X_i
- the sampling for X_i is done conditioned on the current values of the variables in the Markov blanket of X_i
- the algorithm wanders randomly around the state space (the space of possible complete assigments) flipping one variable at a time

Gibbs sampling

- starts at an arbitrary state (with the evidence variables fixed)
- generates a next state by randomly sampling a value for one of the non-evidence variables X_i
- the sampling for X_i is done conditioned on the current values of the variables in the Markov blanket of X_i
- the algorithm wanders randomly around the state space (the space of possible complete assigments) flipping one variable at a time

Gibbs sampling

- starts at an arbitrary state (with the evidence variables fixed)
- generates a next state by randomly sampling a value for one of the non-evidence variables X_i
- the sampling for X_i is done conditioned on the current values of the variables in the Markov blanket of X_i
- the algorithm wanders randomly around the state space (the space of possible complete assigments) flipping one variable at a time

Gibbs sampling

- starts at an arbitrary state (with the evidence variables fixed)
- generates a next state by randomly sampling a value for one of the non-evidence variables X_i
- the sampling for X_i is done conditioned on the current values of the variables in the Markov blanket of X_i
- the algorithm wanders randomly around the state space (the space of possible complete assigments) flipping one variable at a time

Gibbs sampling

- $P(Rain|Sprinkler = true, WetGrass = true) =?$
- evidence *Sprinker* and *WetGrass* are fixed
- nonevidence *Cloudy* and *Rain* initialized randomly (let's say to *true* and *false*, resp.)
- initial state: $[true, true, false, true]$
- the nonevidence vars are sampled repeatedly in an arbitrary order:

Gibbs sampling

- $P(Rain|Sprinkler = true, WetGrass = true) = ?$
- evidence *Sprinker* and *WetGrass* are fixed
- nonevidence *Cloudy* and *Rain* initialized randomly (let's say to *true* and *false*, resp.)
- initial state: $[true, true, false, true]$
- the nonevidence vars are sampled repeatedly in an arbitrary order:

Gibbs sampling

- $P(Rain | Sprinkler = true, WetGrass = true) =?$
- evidence *Sprinker* and *WetGrass* are fixed
- nonevidence *Cloudy* and *Rain* initialized randomly (let's say to *true* and *false*, resp.)
- initial state: $[true, true, false, true]$
- the nonevidence vars are sampled repeatedly in an arbitrary order:

Gibbs sampling

- $P(Rain|Sprinkler = true, WetGrass = true) = ?$
- evidence *Sprinker* and *WetGrass* are fixed
- nonevidence *Cloudy* and *Rain* initialized randomly (let's say to *true* and *false*, resp.)
- initial state: $[true, true, false, true]$
- the nonevidence vars are sampled repeatedly in an arbitrary order:

Gibbs sampling

The nonevidence vars are sampled repeatedly in an arbitrary order:

- *Cloudy* is sampled given the current values of its Markov Blanket variables
  - sample from $P(Cloudy|Sprinkler = true, Rain = false)$; suppose the value is false.
  - then, the new current state is $[false, true, false, true]$
- *Rain* is sampled given its Markov Blanket
  - sample from $P(Rain|Cloudy = false, Sprinkler = true)$; suppose the value is $Rain = true$.
  - then, the new current state is $[false, true, true, true]$

Gibbs sampling

The nonevidence vars are sampled repeatedly in an arbitrary order:

- *Cloudy* is sampled given the current values of its Markov Blanket variables
  - sample from $P(Cloudy|Sprinkler = true, Rain = false)$; suppose the value is false.
  - then, the new current state is $[false, true, false, true]$
- *Rain* is sampled given its Markov Blanket
  - sample from $P(Rain|Cloudy = false, Sprinkler = true)$; suppose the value is $Rain = true$.
  - then, the new current state is $[false, true, true, true]$

Gibbs sampling

The nonevidence vars are sampled repeatedly in an arbitrary order:

- *Cloudy* is sampled given the current values of its Markov Blanket variables
    - sample from $P(Cloudy|Sprinkler = true, Rain = false)$; suppose the value is false.
    - then, the new current state is $[false, true, false, true]$
- *Rain* is sampled given its Markov Blanket
    - sample from $P(Rain|Cloudy = false, Sprinkler = true)$; suppose the value is $Rain = true$.
    - then, the new current state is $[false, true, true, true]$

Gibbs sampling

The nonevidence vars are sampled repeatedly in an arbitrary order:

- *Cloudy* is sampled given the current values of its Markov Blanket variables
  - sample from $P(Cloudy|Sprinkler = true, Rain = false)$; suppose the value is false.
  - then, the new current state is $[false, true, false, true]$
- *Rain* is sampled given its Markov Blanket
  - sample from $P(Rain|Cloudy = false, Sprinkler = true)$; suppose the value is $Rain = true$.
  - then, the new current state is $[false, true, true, true]$

Gibbs sampling

- Each state visited during this process is a sample that contributes to the estimate for the query variable *Rain*
- If a state has probability p then we want to visit it $p \times 100\%$ of the time
- if the process visits 20 states where Rain is true and 60 states where Rain is false, the the answer to the query is $NORMALIZE(< 20, 60 >) =< 0.25, 0.75 >$

Gibbs sampling

- Each state visited during this process is a sample that contributes to the estimate for the query variable *Rain*
- If a state has probability p then we want to visit it $p \times 100\%$ of the time
- if the process visits 20 states where Rain is true and 60 states where Rain is false, the the answer to the query is $NORMALIZE(< 20, 60 >) =< 0.25, 0.75 >$

Gibbs sampling

- Each state visited during this process is a sample that contributes to the estimate for the query variable *Rain*
- If a state has probability p then we want to visit it $p \times 100\%$ of the time
- if the process visits 20 states where Rain is true and 60 states where Rain is false, the the answer to the query is $NORMALIZE(< 20, 60 >) =< 0.25, 0.75 >$

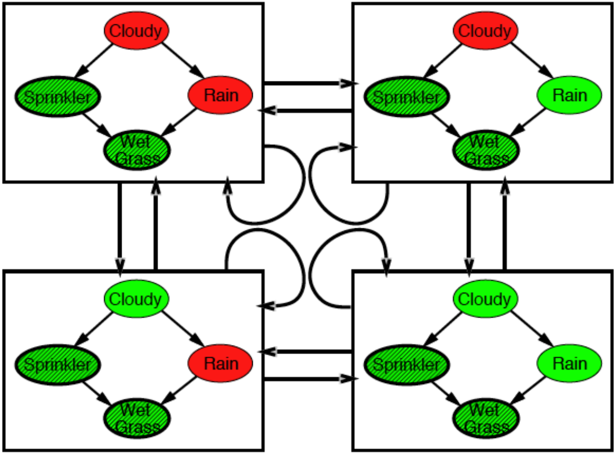
MCMC - Gibbs sampling

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($\mathbf{N}[X]$)

MCMC

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



Wander about for a while, average what you see