Bedingte Unabhängigkeit, Definition

Zwei Ereignisse a, b sind **bedingt unabhängig**, gegeben c, gdw.:

$P(a,b \mid c) = P(a \mid c)\, P(b \mid c)$

Zwei Variable A, B sind **bedingt unabhängig**, gegeben C, gdw.:

$$\mathbf{P}(A, B \mid C) = \mathbf{P}(A \mid C)\, \mathbf{P}(B \mid C)$$

Damit äquivalent sind die Formulierungen

$$\mathbf{P}(A \mid B,C) = \mathbf{P}(A \mid C) \text{ und } \mathbf{P}(B \mid A,C) = \mathbf{P}(B \mid C)$$

Bayes-Netze

… repräsentieren effizient gemeinsame W‘verteilungen und (bzw. unter Verwendung von) Aussagen zur bedingten Unabhängigkeit von Zufallsvariablen

Ein **Bayes-Netz** ist ein gerichteter azyklischer Graph, wobei

- Knoten entsprechen Zufallsvariablen (diskret, kontinuierlich)
- Kanten entsprechen *direkten* Abhängigkeiten zwischen Variablen; führt eine Kante von X nach Y, so heißt X ein Elternknoten von Y, die Menge aller Elternknoten ist $Parents(Y)$
- Jeder Knoten X trägt als Anschrift $\mathbf{P}(X \mid Parents(X))$

Interpretation: Sei $P(x_1, \ldots, x_n) := P(X_1{=}x_1, \ldots, X_n{=}x_n)$. Dann gilt:

$$P(x_1, \ldots, x_n) = \prod_i P(x_i \mid Parents(X_i))$$

vgl. „naives“ Bayessches Modell

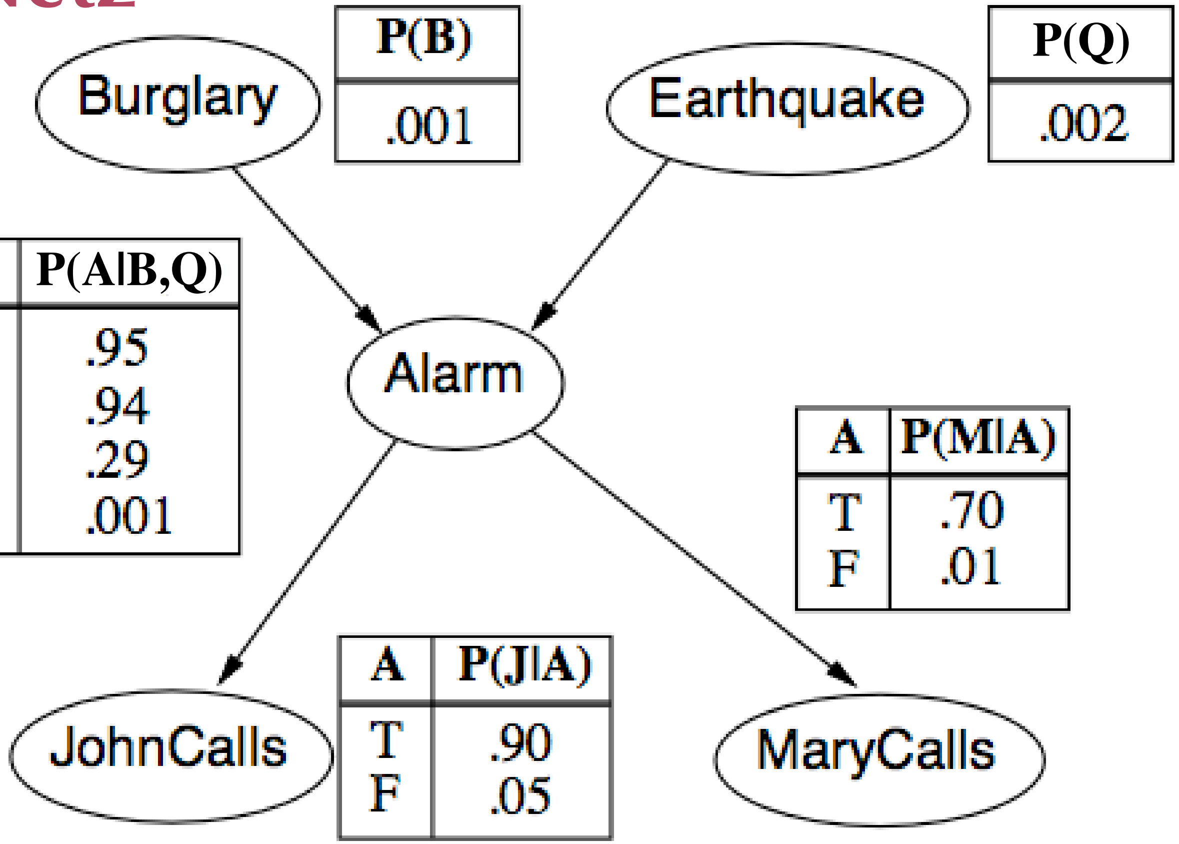
Beispiel für Kalifornische Wissenschaffende

- Hausalarm geht an bei Einbruch und manchmal bei Erdbeben
- Nachbarn John und Mary sollen im Büro anrufen, wenn sie tagsüber Alarm hören
- John ruft im Büro an, aber nicht Mary. Wie wahrscheinlich ist es, dass gerade eingebrochen wird?

Modellierung

- Variablen *Burglary, Earthquake, Alarm, JohnCalls, MaryCalls*
- Gesucht also $\mathbf{P}(B \mid j, {-m})$
- *Alarm* hängt direkt ab von *Burglary, Earthquake*;
  JohnCalls und *MaryCalls* hängen direkt je nur ab von *Alarm*
- W'keiten wie nachfolgend angegeben

Beispiel-Bayes-Netz



P(B)
.001

P(Q)
.002

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

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

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

Beispiel

$P(j,-m,a,b,-q)$

$= P(j|a)P(-m|a)P(a|b,-q)P(b)P(-q)$

$= 0.9\times 0.3\times 0.94\times 0.001\times 0.998$

$= 0.0002532924$

Doch eigentlich suchen wir ja **P**$(B \mid j,-m)$! (Fortsetzung folgt)

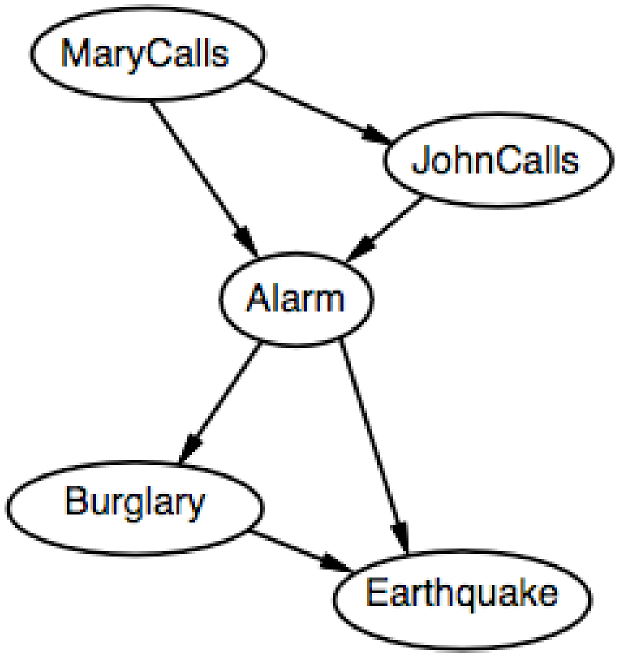
Konstruktion eines Bayes-Netzes

1. Wähle eine vollständige Ordnung $X_1, \ldots, X_n$ der Variablen
2. **for** i=1 **to** n

   füge X_i zum Netz hinzu;

   wähle X_is direkte Vorgänger aus $X_1, \ldots, X_{i-1}$ so, dass

   $$\mathbf{P}(X_i \mid Parents(X_i)) = \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$$



Bei ungeschickter Variablenordnung entstehen unnötig komplexe Netze!

Beispiel: Reihenfolge M, J, A, B, Q

$\mathbf{P}(J|M) \neq \mathbf{P}(J)$ ↪ $M \rightarrow J$

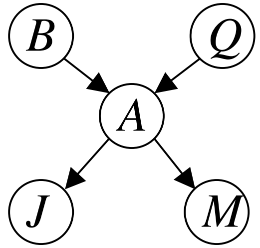
$\mathbf{P}(A|J,M) \neq \mathbf{P}(A|J)$ und $\mathbf{P}(A|J,M) \neq \mathbf{P}(A|M)$

↪ $M \rightarrow A,\ J \rightarrow A$

$\mathbf{P}(B|A,J,M) = \mathbf{P}(B|A)$ ↪ nicht $M \rightarrow B$ $J \rightarrow B$

…

Inferenz durch Ausrechnen



$\mathbf{P}(B \mid j,\text{-}m)$

$= \mathbf{P}(B, j,\text{-}m) \,/\, P(j,\text{-}m)$ Produktregel

$= \alpha\, \mathbf{P}(B, j,\text{-}m)$ Normalisierung mit Konstante

$= \alpha \sum_q \sum_a [\mathbf{P}(B, Q{=}q, A{=}a, j, \text{-}m)]$ Kolmogorow 3

Weiter mit $B{=}true$ (anderer Fall entsprechend)

$= \alpha \sum_q \sum_a P(b)P(Q{=}q)P(A{=}a|b,q)P(j|a)P(\text{-}m|a)$ bed. Unabhängigkeit

$= \alpha\, P(b) \sum_q P(Q{=}q)\, [\sum_a P(A{=}a|b,q)P(j|a)P(\text{-}m|a)]$

Durch Einsetzen der W'keiten gemäß Bayes-Netz:

$\approx \alpha\, \langle 0.0002568, 0.0497979 \rangle$ $\alpha \approx 20$

$\approx \langle 0.00513, 0.99487 \rangle$

➥ Die W'keit für einen Einbruch ist von a priori 0,1% auf 0,513% gestiegen

Inferenz durch vollständiges Ausrechnen

```
function ENUMERATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network with variables {X} ∪ E ∪ Y
   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
       extend e with value x_i for X
       Q(x_i) ← ENUMERATE-ALL(VARS[bn], e)
   return NORMALIZE(Q(X))
```

verstecke Variablen

```
function ENUMERATE-ALL(vars, e) returns a real number
   if EMPTY?(vars) then return 1.0
   Y ← FIRST(vars)
   if Y has value y in e
       then return P(y | Pa(Y)) × ENUMERATE-ALL(REST(vars), e)
       else return Σ_y P(y | Pa(Y)) × ENUMERATE-ALL(REST(vars), e_y)
           where e_y is e extended with Y = y
```

Parents, direkte Vorgänger von Y

Eigenschaften des vollständigen Ausrechnens

☺ Terminiert sicher mit korrektem Ergebnis

☺ Speicherbedarf: $O(n)$ („Tiefentraversierung“)

☹ Zeitbedarf: $O(m^n)$ für diskrete ZV max. m Werte (boolsche ZV: m=2)

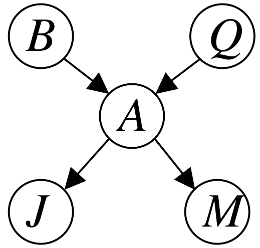
☹ **Satz**: Exakte Inferenz in Bayes-Netzen ist NP-vollständig
Beweisidee: Abbildung propositionaler Inferenz in Bayes-Netz-Inferenz

Optimierungsmöglichkeiten (s. nächste Folien)

- *caching* von mehrfach auftauchenden Teilergebnissen
- Variablenelimination
- Überführung von Bayes-Netzen in Polybäume

↪ **Satz**: Inferenz in Bayes-Polybäumen ist linear in Speicher- und Zeitbedarf (hier ohne Beweis)

Wiederverwendung von Teilergebnissen



wir hatten vorher:

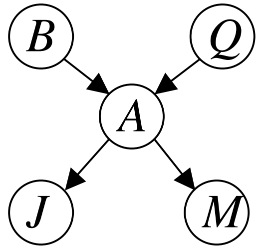
$\mathbf{P}(B \mid j,-m)$

$$= \alpha\, P(b) \sum_q P(Q=q)\, \left[\sum_a P(A=a|b,q)P(j|a)P(-m|a)\right]$$

Die Terme $P(j|a)P(-m|a)$ (für alle Werte von a) tauchen in mehreren Summanden auf!

↪ Stell die Berechnung so um, dass die Verteilung „von innen nach außen“ berechnet wird (Verfahren: Russell/Norvig S. 507ff)

Variablenelimination



Beispielanfrage: $\mathbf{P}(J \mid b)$
(wie wahrscheinlich ruft John an, wenn ein Einbruch vorliegt?)

$$\mathbf{P}(J \mid b) = \alpha\, P(b) \sum_q P(Q{=}q) \left[\sum_a P(A{=}a|b,q) P(J|a) \sum_m P(m|a)\right]$$

Beobachtung: $\sum_m P(m|a)$ ist immer =1!
M hängt nur ab von A, für das immer alle Werte untersucht werden:
M ist **irrelevant** für die Anfrage: M-Terme können wegfallen

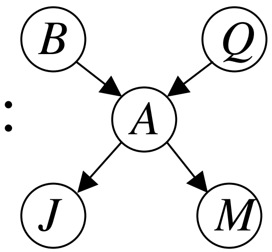
Satz:
Variable $Y \neq X$ ist irrelevant für die Anfrage $\mathbf{P}(X \mid \boldsymbol{E})$, falls
$M \notin \mathit{Vorgänger}(\{X\} \cup \boldsymbol{E})$ (*Vorgänger* ist trans. Hülle von *Parents*)
(Beweis: Nach Konstruktion eines Bayes-Netzes)

Im **Beispiel**: $X{=}J$, $\boldsymbol{E}{=}\{B\}$, $\mathit{Vorgänger}(\{X\} \cup \boldsymbol{E}) = \{A, Q\}$
↪ M ist irrelevant

Netze zu Polybäumen

Polybaum: Graph, in dem es zwischen zwei Knoten max. 1 Pfad gibt.

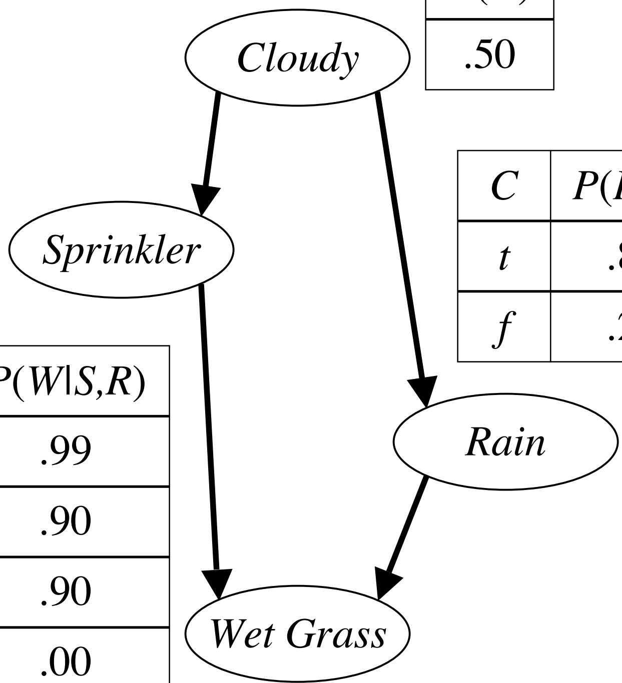
Beispiel:



Gegenbeispiel:

$P(C)$
.50

C	$P(S\|C)$
t	.10
f	.50



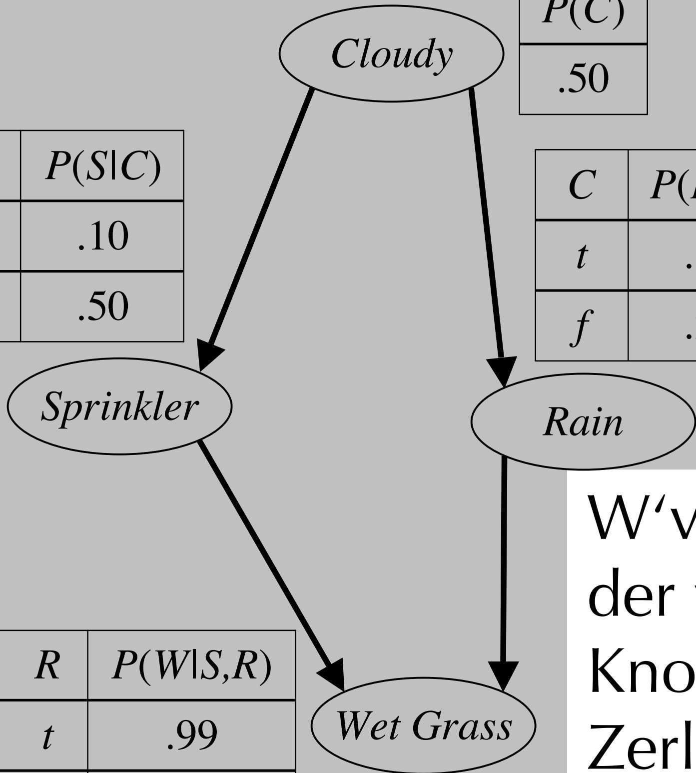
C	$P(R\|C)$
t	.80
f	.20

S	R	$P(W\|S,R)$
t	t	.99
t	f	.90
f	t	.90
f	f	.00

Umformungsansatz:

- Verbinde Knoten, bis das Resultat ein Polybaum ist!
- Errechne Verteilungen der verbundenen Knoten aus Verteilungen der Einzelknoten

Rasensprengen mit Polybäumen



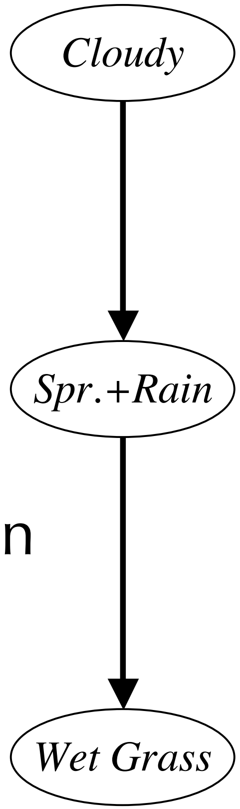
$P(C)$
.50

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	.00

W‘verteilungen der verbundenen Knoten nehmen Zerlegung der W‘verteilungen im Bayes-Netz teils zurück!



$P(C)$
.50

C	$P(S{+}R \mid C)$			
	$t\,t$	$t\,f$	$f\,t$	$f\,f$
t	.08	.02	.72	.18
f	.10	.40	.10	.40

$S{+}R$	$P(W\|S{+}R)$
$t\ t$	.99
$t\ f$	.90
$f\ t$	.90
$f\ f$	.00

Bayes-Netz-Inferenz durch Approximation

Rechne (bedingte) Verteilungen nicht aus,
sondern ermittle sie durch Stichproben aus dem Netz!

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 $\mathbf{P}(X_i \mid Parents(X_i))$
  return x

W‘keit, dass Funktion die Stichprobe $(x_1, \ldots, x_n)$ erzeugt:

$$S_{PS}(x_1, \ldots, x_n) = \prod_i P(x_i \mid Parents(X_i)) = P(x_1, \ldots, x_n)$$

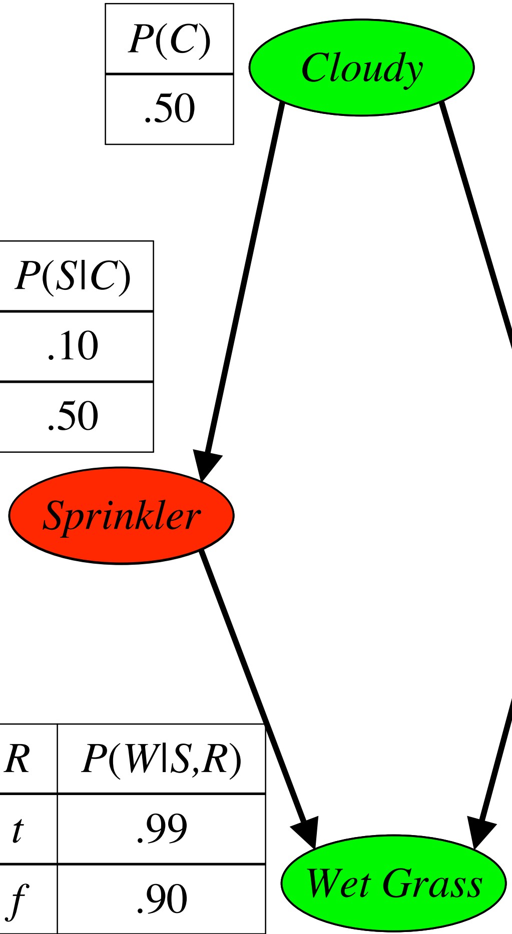
d.h.: hinreichend viele Stichproben approximieren Verteilung!

Beispiel: Approximatives Rasenwässern

$P(C)$
.50

C	$P(S \mid C)$
t	.10
f	.50

C	$P(R \mid C)$
t	.80
f	.20



S	R	$P(W \mid S,R)$
t	t	.99
t	f	.90
f	t	.90
f	f	.00

1. Ziehe C nach $\mathbf{P}(C)=\langle 0.5, 0.5\rangle$
   ↪ Ergebnis sei *Cloudy=true*
2. Ziehe S nach $\mathbf{P}(S \mid c)=\langle 0.1, 0.9\rangle$
   ↪ Ergebnis sei *Sprinkler=false*
3. Ziehe S nach $\mathbf{P}(R \mid c)=\langle 0.8, 0.2\rangle$
   ↪ Ergebnis sei *Rain=true*
4. Ziehe S nach $\mathbf{P}(W \mid \neg s, r)=\langle 0.9, 0.1\rangle$
   ↪ Ergebnis sei *Wet_Grass=true*

Sei $N_{PS}(x_1, \ldots, x_n)$ die Häufigkeit, mit der $(x_1, \ldots, x_n)$ gezogen wurde.

Nach N (hinreichend groß) Stichproben:

$$P(x_1, \ldots, x_n) = N_{PS}(x_1, \ldots, x_n) / N$$

Grenzen von PRIOR-SAMPLE

- Vorhandene Evidenz wird nicht verwendet:
  $\mathbf{P}(\boldsymbol{X})$ wird approximiert, aber nicht ohne Weiteres $\mathbf{P}(\boldsymbol{X}|\mathbf{e})$
- Bei vielen Variablen oder unwahrscheinlicher Evidenz $\mathbf{e}$ (kleinem $P(\mathbf{e})$) werden viele „Nieten“ gezogen
  ↪ N muss groß sein
- Je mehr Evidenzvariable, desto kleiner $P(\mathbf{e})$
  ↪ Algorithmus wird relativ schlechter bei mehr Information!

Lösung: **Gewichtete Stichproben** (WEIGHTED-SAMPLE)

- Belege Evidenz-Variable wie durch $\mathbf{e}$ vorgegeben
- Wichte $\boldsymbol{X}$ nach $\mathbf{P}(\mathbf{e}|\boldsymbol{X}-\mathbf{e})$

Gewichtete Stichproben

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

   for j = 1 to N do
       x, w ← WEIGHTED-SAMPLE(bn)
       W[x] ← W[x] + w where x is the value of X in x
   return NORMALIZE(W[X])


function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

   x ← an event with n elements; w ← 1
   for i = 1 to n do
       if X_i has a value x_i in e
           then w ← w × P(X_i = x_i | Parents(X_i))
           else x_i ← a random sample from P(X_i | Parents(X_i))
   return x, w
```

Korrektheit von LIKELIHOOD-WEIGHTING

- Stichprobenw'keit in WEIGHTED-SAMPLE:
  $S_{WS}(\mathbf{z},\mathbf{e}) = \prod^{l}_{i} P(z_i \mid Parents(Z_i))$
- Gewicht für Stichprobe $(\mathbf{z},\mathbf{e})$:
  $w(\mathbf{z},\mathbf{e}) = \prod^{m}_{i} P(e_i \mid Parents(E_i))$
- Gewichtete Stichprobenw'keit:
  $S_{WS}(\mathbf{z},\mathbf{e})w(\mathbf{z},\mathbf{e}) = \prod^{l}_{i} P(z_i \mid Parents(Z_i)) \prod^{m}_{i} P(e_i \mid Parents(E_i)) = P(\mathbf{z},\mathbf{e})$

Es bleibt das Problem, dass bei kleinem $P(\mathbf{e})$ (geringe W'keit des Evidenzereignisses, viele Evidenzvariablen) große Stichproben erforderlich sind!

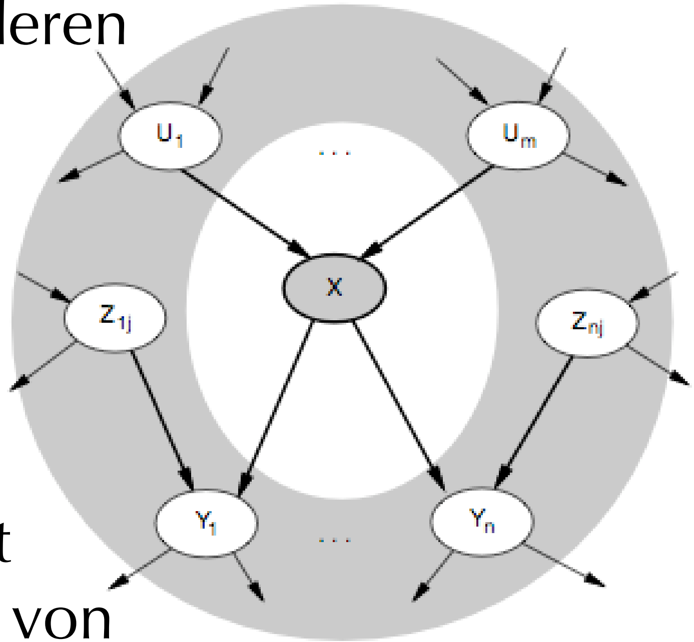
vgl. lokale Suchverfahren!

Stichprobenziehen per *random walk*

Idee:

- Starte mit zufälligem Ereignis, das die Evidenz spiegelt
- Erzeuge neue Ereignisse durch „lokale Änderung“ in Nicht-Evidenz-Variablen
- Die lokalen Änderungen entsprechen W‘verteilung abhängig von den unveränderten Variablen

Markow-Hülle von X: die Eltern von X, die direkten Nachfolger von X und deren (andere) Eltern



Satz:
X ist bedingt unabhängig von allen anderen Knoten, gegeben seine Markow-Hülle

Markov Chain Monte Carlo Sampling

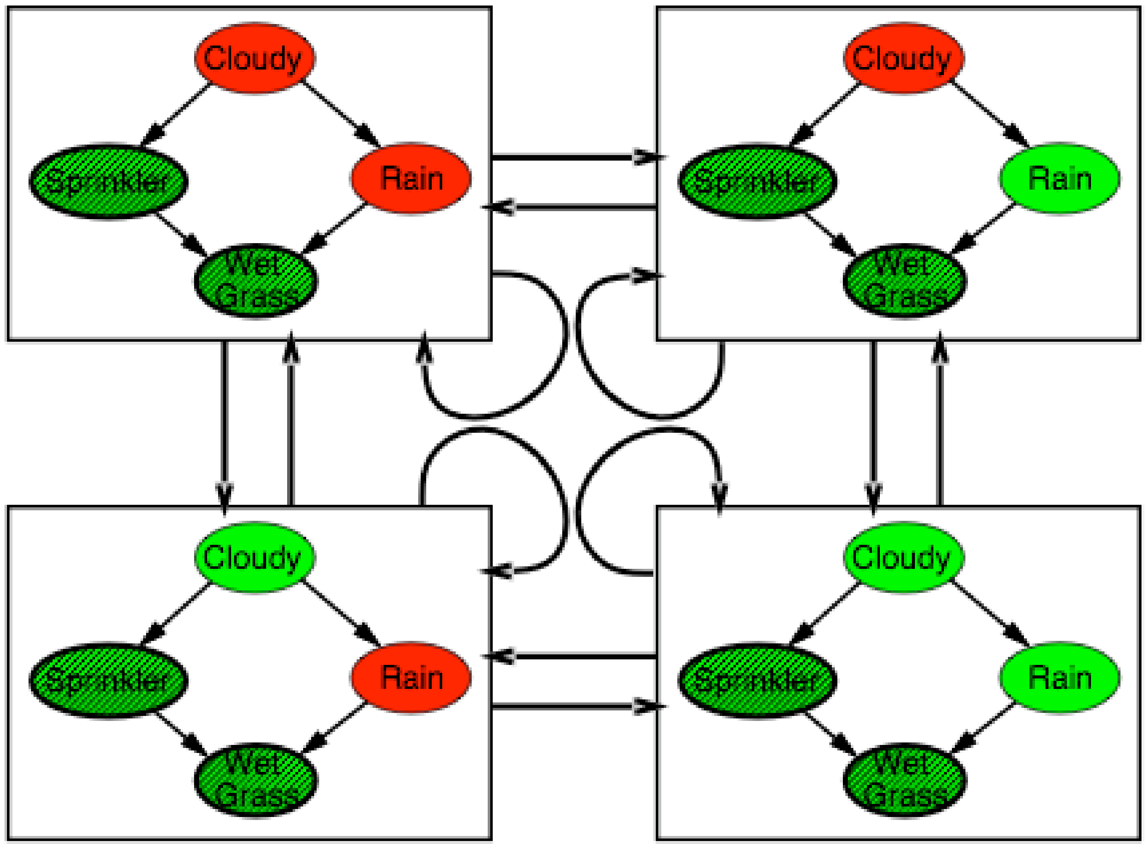
function MCMC-ASK(X, **e**, bn, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initially copied from **e**

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for $j = 1$ to N **do**
    $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
    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 $\mathbf{x}$
  return NORMALIZE($\mathbf{N}[X]$)

Markow-Hülle (*Markov blanket*)

Qualitatives Verhalten von MCMC (Beispiel)

Sei **e**={*Sprinkler*=*true*, *Wet_Grass*=*true*}, **Z**={*Cloudy*, *Rain*}



Stichproben entsprechen einer **Markow-Kette**

- Für $\mathbf{P}(r|e)$ setze z.B. N=100
- Zähle Stichproben aus nach r, –r; haben z.B. 31 r, 69 –r
- $\mathbf{P}(r|e)$ = NORMALIZE(⟨31,69⟩) = ⟨0.31, 0.69⟩

Eigenschaften von MCMC

- ☺ **Satz**: Die Markow-Kette approximiert eine stationäre Verteilung: Im Limes ist sie in einer Häufigkeit in jedem Zustand, die der bedingten W'keit des entsprechenden Ereignisses entspricht.
- ☹ *N* muss man raten – wann ist Markow-Kette konvergiert?
- ☹ Bei sehr großen Markow-Hüllen lange Zeit bis Konvergenz