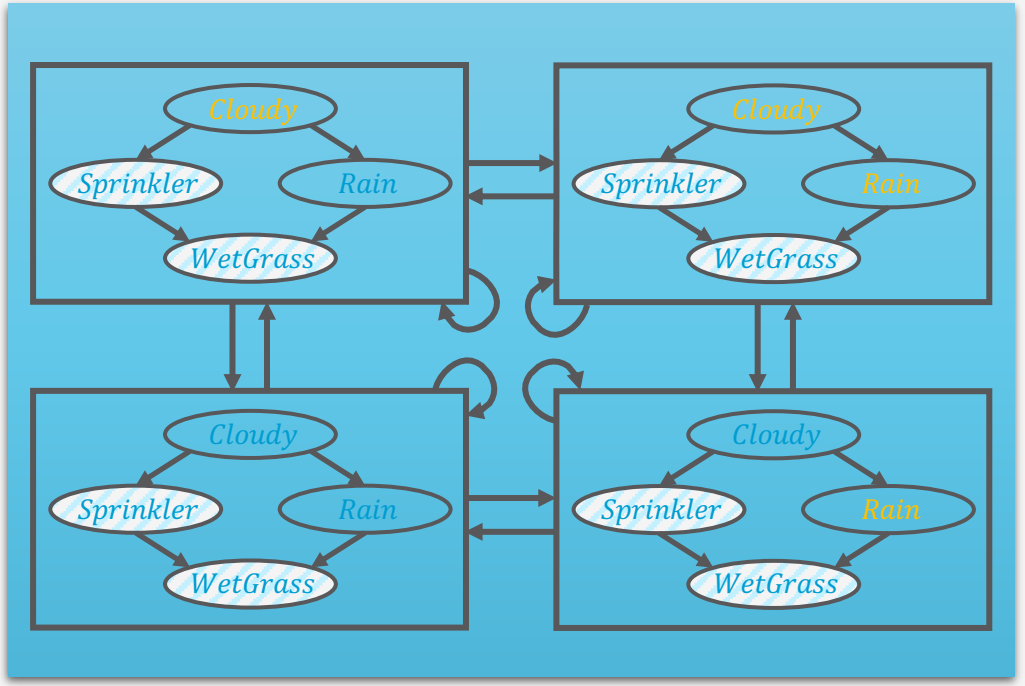
WWU
MÜNSTER

Approximative Inferenz in Episodischen PGMs

Einführung in die Künstliche Intelligenz



wissen.leben

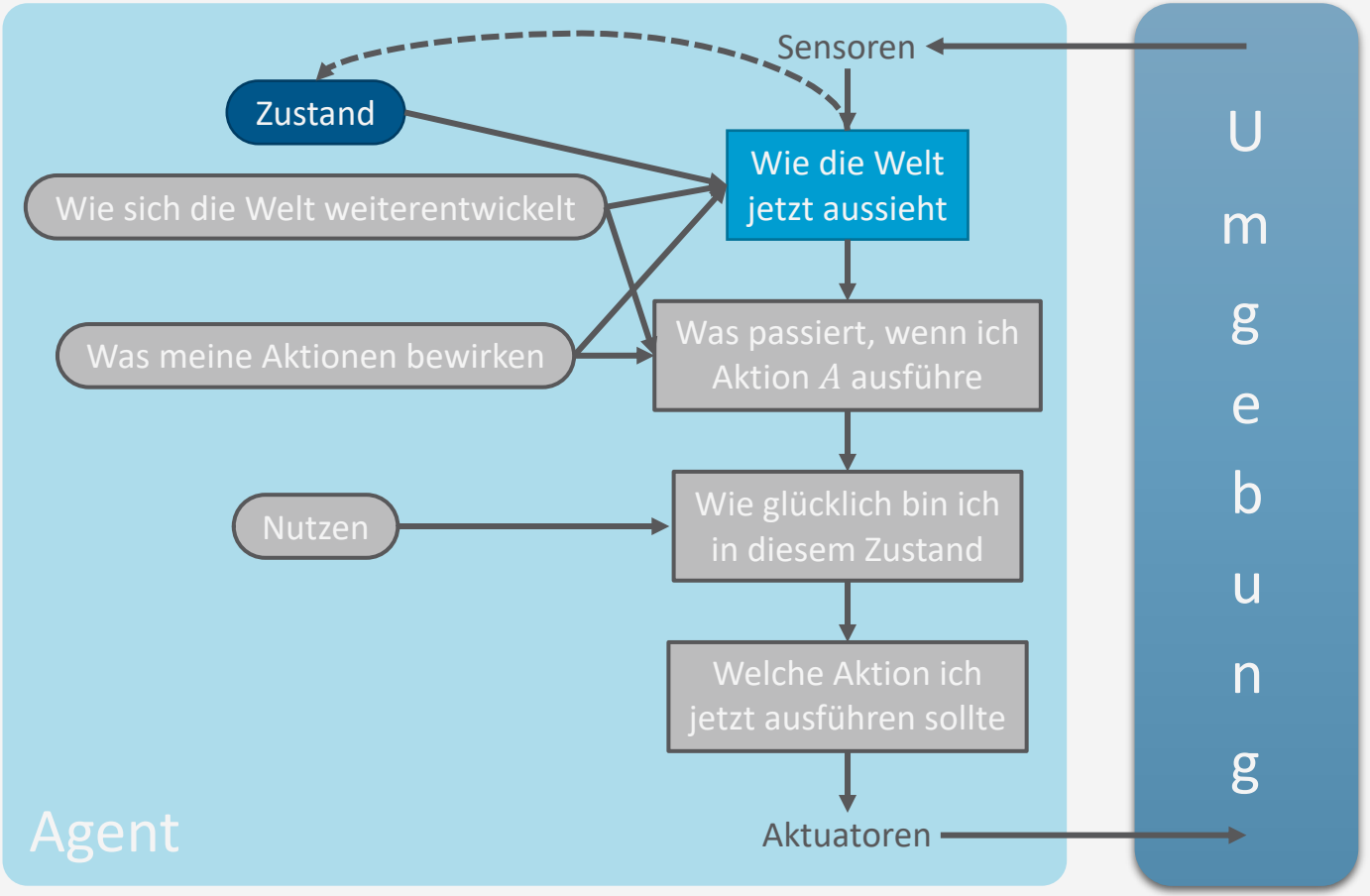
Tanya Braun
Arbeitsgruppe Data Science, Institut für Informatik

Inhalte

1. **Künstliche Intelligenz & Agenten**
   - Agentenabstraktion, Rationalität
   - Aufgabenumgebung
2. **Episodische PGMs**
   - Gerichtetes Modell: Bayes Netze (BNs)
   - Ungerichtete Modelle
3. **Exakte Inferenz in episodischen PGMs**
   - Wahrscheinlichkeits- und Zustandsanfragen
   - Direkt auf den Modellen, mittels Hilfsstrukturen
4. **Approximative Inferenz in episodischen PGMs**
   - Wahrscheinlichkeitsanfragen
   - Deterministische, stochastische Algorithmen
5. **Lernalgorithmen für episodische PGMs**
   - Bei (nicht) vollständigen Daten, (un)bekannter Struktur
6. **Sequentielle PGMs und Inferenz**
   - Dynamische BNs, Hidden-Markov-Modelle
   - filtering / prediction / hindsight Anfragen, wahrscheinlichste Zustandssequenz
   - Exakter, approximativer Algorithmus
7. **Entscheidungstheoretische PGMs**
   - Präferenzen, Nutzenprinzip
   - PGMs mit Entscheidungs- und Nutzenknoten
   - Berechnung der besten Aktion (Aktionssequenz)
8. **Abschlussbetrachtungen**

Einordnung der Vorlesung: *Modell- und nutzenbasierter Agent*

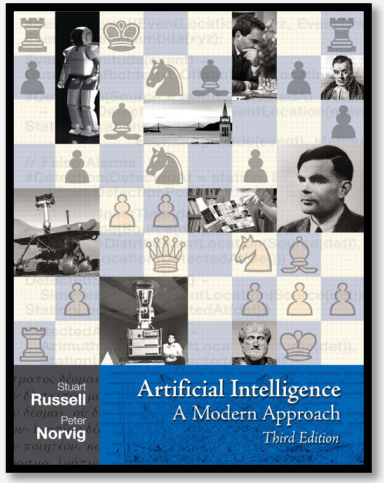
- Nachfolgende Themen der Vorlesung
  2. Episodische PGMs
  3. Exakte Inferenz in episodischen PGMs
  4. Approximative Inferenz in episodischen PGMs
  5. Lernalgorithmen für episodische PGMs
  6. Sequentielle PGMs und Inferenz
  7. Entscheidungstheoretische PGMs

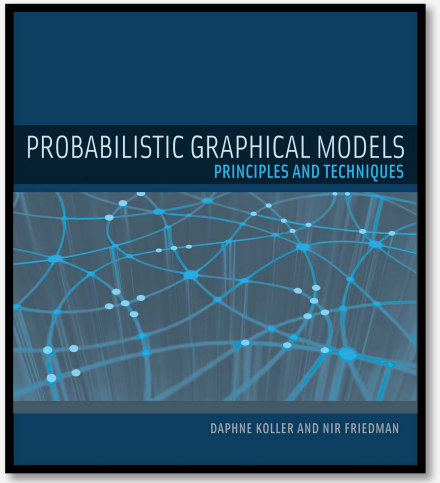


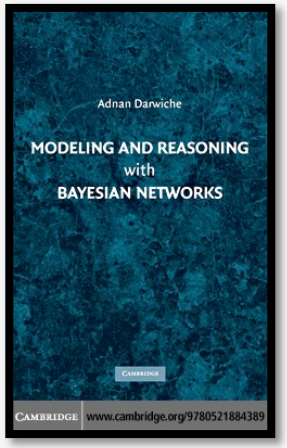
Literaturhinweise

Inhalte dieses Themenblocks werden in den folgenden Kapiteln der Vorlesungsbücher behandelt

- AIMA(de)
  - Kap. 14.5: Annähernde Inferenz in Bayes Netzen
- PGM
  - Kap. 12.1-3: Particle-based Approximate Inference



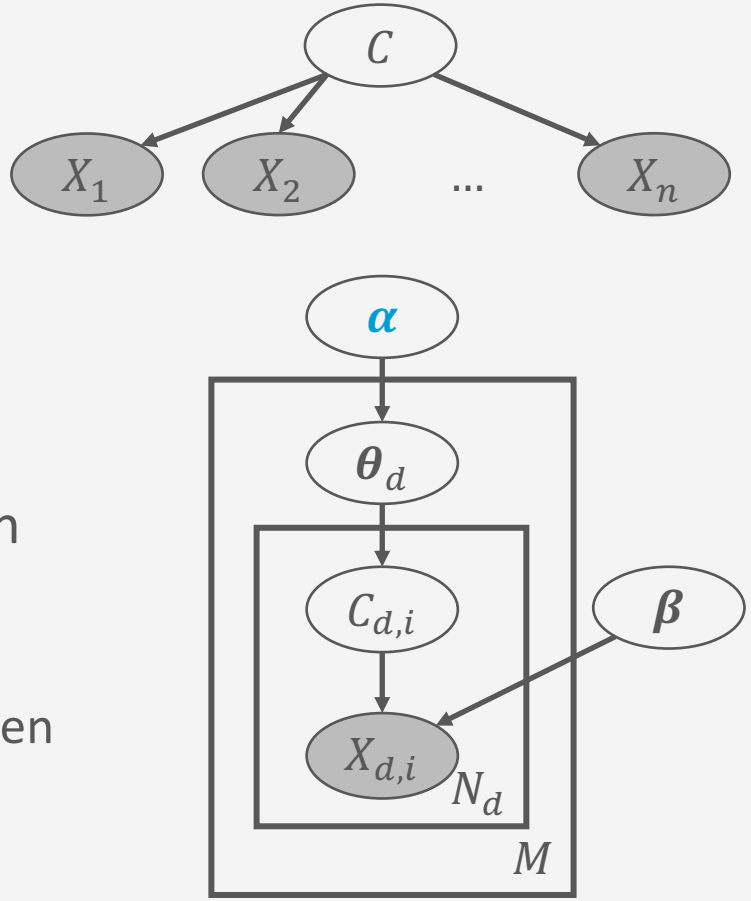




- Wer gerne ein anderes Buch ausprobieren möchte (Fokus auf BNs):
  - Adnan Darwiche, *Modelling and Reasoning with Bayesian Networks*, 2009.

Anwendungen

- Mixture of Topics
  - Mögliche Anfrage: Topic eines Wortes $P(C_{d,i} \mid \boldsymbol{x}_d)$
    - $\boldsymbol{S} = \{C_{d,i}\}, \boldsymbol{T} = \mathrm{rv}(\boldsymbol{x}_d)$ und damit
      $\boldsymbol{U} = \boldsymbol{R} \setminus \boldsymbol{S} \setminus \boldsymbol{T} = \{C_{d,j}\}_{i \neq j} \cup \{C_{d',k}\}_{d' \neq d}$
  - Unabhängigkeit zwischen den Dokumenten
    → Vereinfachung: nur d betrachten, damit $\boldsymbol{U} = \{C_{d,j}\}_{i \neq j}$
  - Problem: Immer noch (zu) viele Zufallsvariablen auszusummieren
    → Nächstes Thema:
    Approximative Inferenz durch Sampling (Thema 4)
    - Mögliche Anfrage: Fragmente eines Textes (mit Topic) vervollständigen $P(\boldsymbol{x}'_d \mid \boldsymbol{x}''_d)$ (bzw. $P(\boldsymbol{x}'_d \mid \boldsymbol{x}''_d, C_{d,i})$) mit $\boldsymbol{x}'_d \cup \boldsymbol{x}''_d = \boldsymbol{x}_d$
      - Werte (Worte) sampeln



Überblick: 4. Approximative Inferenz in episodischen PGMs

A. ***Überblick***
   - *PAC Theorie, deterministische vs. stochastische Approximation, Variational Inference*

B. *Direktes Sampling*
   - Sampling in einer Wahrscheinlichkeitsverteilung mit und ohne Evidenz
   - Forward Sampling (ohne Evidenz), Rejection Sampling (mit Evidenz) in BNs
   - Likelihood Weighting in BNs, Importance Sampling als Verallgemeinerung
   - Importance Sampling für Faktormodelle

C. *Inferenz durch Markov-Ketten-Simulation*
   - Markov-Chain Monte-Carlo (MCMC) Sampling: Gibbs Sampling, Metropolis-Hastings Sampling

D. *Sampling für die Datengenerierung*
   - Datensynthese

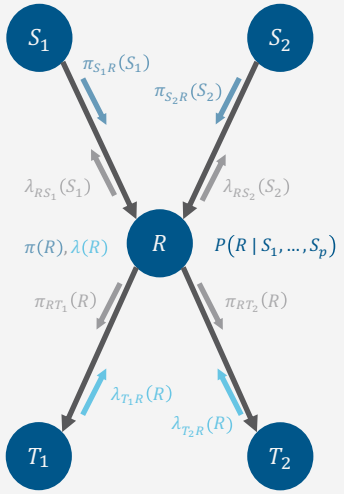
Approximationen (Annäherungen)

- Approximative (annähernde) Antworten zu Anfragen wie die Marginalverteilung $P(R)$ oder bedingte Wahrscheinlichkeitsverteilung $P(R|\boldsymbol{e})$
  - Soll Komplexität der exakten Inferenz überwinden
    - Problem der approximativen Inferenz aber auch $\mathcal{NP}$-schwer im Worst Case
- Annahme einer intuitiven Auffassung von Approximation:
  - Die Antwort mag bis zu einem gewissen Grad fehlerhaft sein
- Formale Betrachtung durch PAC Theorie (*Probably Approximately Correct*; wahrscheinlich annährend korrekt) mittels Parameter (δ, ε)
  - Konfidenz (quantifiziert durch δ) darin, dass die gefundene Lösung (Antwort) maximal um ε von der wahren Lösung (exakten Antwort) abweicht
    - Bzw. wie viele Samples (Proben) benötigt man, um δ, ε zu erfüllen?

Deterministische vs. stochastische Approximationen

Deterministische Approximation

- Ergibt jedes Mal das gleiche Ergebnis
- Beispiel
  - Loopy Belief Propagation
    - PP auf Nicht-Polytrees
    - Nachrichtenversand folgt gleichem Schema, Berechnungen werden gleich sein und auf das gleiche Ergebnis hinauslaufen



Stochastische Approximation

- Kann bei unterschiedlichen Läufen zu unterschiedlichen Ergebnissen führen
- Typischerweise Sampling-basiert
  - Kann z.B. von unterschiedlichen Seeds für Zufallszahlen abhängen
  - Sollte in der Regel gegen die richtigen Antwort konvergieren

Fokus dieses Vorlesungsteils

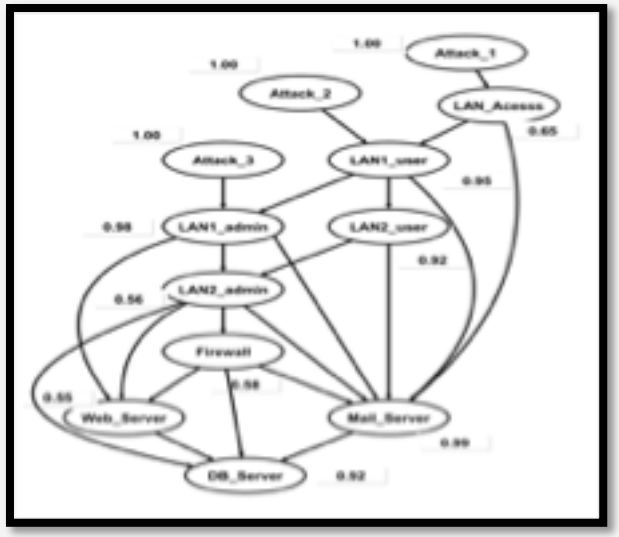
Anfragebeantwortung durch stochastische Simulation

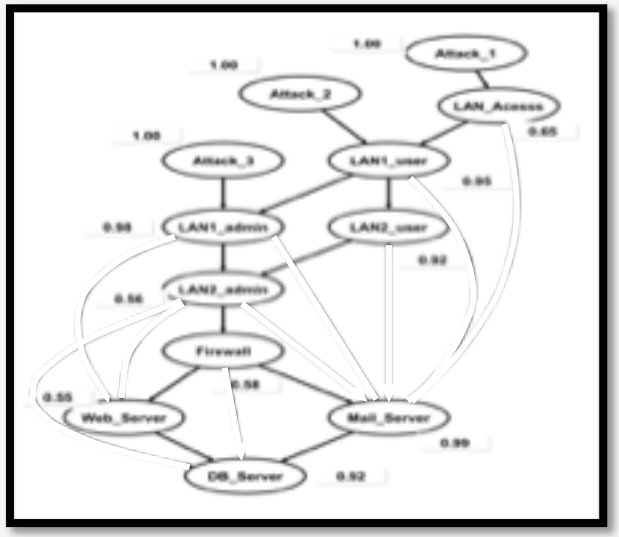
- Monte Carlo Methoden
  - Wiederholtes zufälliges Sampling um ein numerisches Ergebnis zu erhalten
- Grundidee
  1. Ziehe N Samples aus einer Sampling Verteilung S
  2. Berechne eine annähernde Verteilung $\hat{P}$
  3. Zeige, dass $\hat{P}$ zur wahren Verteilung P konvergiert
  - Güte der Approximation hängt von der Anzahl an Samples ab

- Sampling-Methoden in dieser Vorlesung
  - Direktes Sampling (im Graphen)
    - Forward Sampling: aus lokalen Verteilungen ohne Evidenz
    - Rejection Sampling: Verwerfe Samples, die der Evidenz widersprechen
    - Likelihood weighting, importance sampling: Gewichtete Samples
  - Markov Chain Monte Carlo (MCMC):
    - Sampling in einem stochastischen Prozess, dessen stationäre Verteilung der wahren Verteilung P entspricht
    - Methoden: Gibbs und Metropolis-Hastings

Hinweis: *Variational Inference (VI)*

- Idee: Zu schwieriges Optimierungs- bzw. Inferenzproblem durch ein einfacheres zu ersetzen, welches einem Garantien (untere / obere Schranken bzgl. der wahren Antwort) bietet
  - Beispiel: Im Faktormodell bestimmte weitere Unabhängigkeiten annehmen, so dass ein einfacheres Modell ergibt, in dem besser gerechnet werden kann
    - Unterer Bayesian Attack Graph erlaubt Baumweite 3 (oben: 7)
    - Schränkt quasi den Suchraum nach der richtigen Antwort ein auf den Bereich, der die weiteren Unabhängigkeiten erfüllt
    - Ergibt z.B. bei der Entropie-Maximierung eine untere Schranke
- Je nach angewendeter Methode kann VI eine deterministische oder stochastische Approximation liefern
  - Im Beispiel:
    - Im vereinfachten Modell mittels VE rechnen → deterministisch
    - Im vereinfachten Modell mittels Sampling rechnen → stochastisch





Überblick: 4. Approximative Inferenz in episodischen PGMs

A. Überblick
- *PAC Theorie, deterministische vs. stochastische Approximation, Variational Inference*

B. Direktes Sampling
- Sampling in einer Wahrscheinlichkeitsverteilung mit und ohne Evidenz
- Forward Sampling (ohne Evidenz), Rejection Sampling (mit Evidenz) in BNs
- Likelihood Weighting in BNs, Importance Sampling als Verallgemeinerung
- Importance Sampling für Faktormodelle

C. Inferenz durch Markov-Ketten-Simulation
- Markov-Chain Monte-Carlo (MCMC) Sampling: Gibbs Sampling, Metropolis-Hastings Sampling

D. Sampling für die Datengenerierung
- Datensynthese

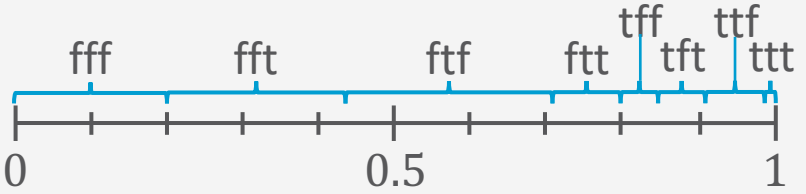
Sampling zur Anfragebeantwortung: Direktes Sampling ohne Evidenz

- Gegeben eine (vollständige gemeinsame) Wahrscheinlichkeitsverteilung $P_{\boldsymbol{R}}$ über Zufallsvariablen $\boldsymbol{R}$ und Anfragevariable $R \in \boldsymbol{R}$
- Sample: beinhaltet eine Beobachtung für jede Zufallsvariable
  - I.e., Sample = zusammengesetztes Event für alle $\boldsymbol{R}$
  - Manchmal auch Partikel genannt
- Vorgehen
  1. Generiere eine Menge von Samples
     - Pro Sample: Generiere ein zusammengesetztes Event für $\boldsymbol{R}$ gemäß der (vollständigen gemeinsamen) Verteilung $P_{\boldsymbol{R}}$
  2. Basierend auf der Menge von Samples, schätze $P(R)$ durch Zählen
     - Wie häufig kommt $R = r$ für jedes $r \in \mathrm{Val}(R)$ in den Samples vor?

Epid	*Travel*	*Sick*	*P*
false	*false*	*false*	0.20
false	*false*	*true*	0.24
false	*true*	*false*	0.28
false	*true*	*true*	0.08
true	*false*	*false*	0.05
true	*false*	*true*	0.06
true	*true*	*false*	0.07
true	*true*	*true*	0.02

Sampling zur Anfragebeantwortung: 1. Generierung von Samples

a. Teile Intervall $[0,1]$ gemäß $P_{\boldsymbol{R}}$ in Unter-Intervalle auf:
- Akkumuliert die Wahrscheinlichkeiten
- Formal, bei $\text{Val}(\boldsymbol{R}) = \{\boldsymbol{r}_1, \dots, \boldsymbol{r}_m\}$ und einer beliebigen, aber festen Reihenfolge $(\boldsymbol{r}_1, \dots, \boldsymbol{r}_m)$:
  - $[0, x_1]$
    - $x_1 = P(\boldsymbol{r}_1)$
  - $(x_{i-1}, x_i], i \in \{2, \dots, m\}$
    - $x_i = x_{i-1} + P(\boldsymbol{r}_i)$
    - Letzter Fall ($i = m$) ist damit: $(x_{m-1}, x_m] = (1 - P(\boldsymbol{r}_m), 1]$
- Beispiel
  - $\boldsymbol{R} = \{Epid, Sick, Travel\}$, $\boldsymbol{e} = \{sick\}$, $R = Travel$
  - $[0, 0.20], (0.20, 0.20 + 0.24], (0.44, 0.44 + 0.28], \dots$



Epid	*Travel*	*Sick*	*P*
false	*false*	*false*	0.20
false	*false*	*true*	0.24
false	*true*	*false*	0.28
false	*true*	*true*	0.08
true	*false*	*false*	0.05
true	*false*	*true*	0.06
true	*true*	*false*	0.07
true	*true*	*true*	0.02

Sampling zur Anfragebeantwortung: 1. Generierung von Samples

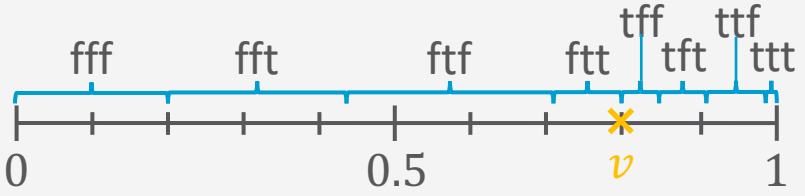
b. Für ein Sample, generiere eine Zufallszahl $v \in [0,1]$ und nehme das zusammengesetzte Event, welches zu dem Unter-Intervall gehört, in das v fällt

- Formal:

$$f(v) = \begin{cases} \boldsymbol{r}_1 & v \in [0, x_1], x_1 = P(\boldsymbol{r}_1) \\ \boldsymbol{r}_2 & v \in (x_1, x_2], x_2 = x_1 + P(\boldsymbol{r}_2) \\ \vdots & \vdots \\ \boldsymbol{r}_m & v \in (x_{m-1}, 1] \end{cases}$$

$$= \begin{cases} \boldsymbol{r}_1 & v \in [0, x_1], x_1 = P(\boldsymbol{r}_1) \\ \boldsymbol{r}_i & v \in (x_{i-1}, x_i], i \in \{2, \dots, m\}, x_i = x_{i-1} + P(\boldsymbol{r}_i) \end{cases}$$

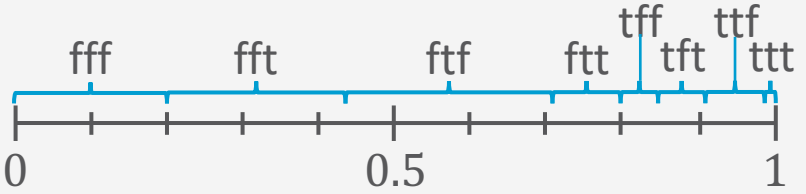
- Beispiel:
  - $v = 0.8 \rightarrow (false, true, true)$



Epid	Travel	Sick	P
false	false	false	0.20
false	false	true	0.24
false	true	false	0.28
false	true	true	0.08
true	false	false	0.05
true	false	true	0.06
true	true	false	0.07
true	true	true	0.02

Sampling zur Anfragebeantwortung: 2. Abschätzen von $P(R)$ (Zählen)

2. Gegeben die Menge an generierten Samples $\{\boldsymbol{r}_k\}_{k=1}^{N}$
   - Zähle, wie häufig in allen $\boldsymbol{r}_k$ die unterschiedlichen Werte von R vorkommen:
     - Für jedes $r \in \mathrm{Val}(R)$: $n_r = \sum_{k=1}^{N'} 1 \mid \pi_R(\boldsymbol{r}_k) = r$
   - Gebe aus: $\hat{P}(R) = \left(\frac{n_1}{n}, \ldots, \frac{n_l}{n}\right)$
     - $n = \sum_{r \in \mathrm{Val}(R)} n_r \,, l = |\mathrm{Val}(R)|$
     - $\hat{P}(R) \approx P(R)$
   - Beispiel:
     - $(f, f, t), (t, t, f), (f, t, t), (t, f, t), \ldots$
     - Annahme: $n_0 = 363, n_1 = 324, n = 687$



$Travel$	$\hat{P}(Travel)$
$false$	$\frac{363}{687} = 0.528$
$true$	$\frac{324}{687} = 0.472$

$P(Travel)$
0.55
0.45

$Epid$	$Travel$	$Sick$	P
$false$	$false$	$false$	0.20
$false$	$false$	$true$	0.24
$false$	$true$	$false$	0.28
$false$	$true$	$true$	0.08
$true$	$false$	$false$	0.05
$true$	$false$	$true$	0.06
$true$	$true$	$false$	0.07
$true$	$true$	$true$	0.02

Sampling: Generalisierung

- Schätze die Erwartung einer Funktion $f(\boldsymbol{R})$ relativ zu einer Verteilung $P(\boldsymbol{R})$

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} f(\boldsymbol{r}) P(\boldsymbol{r})$$

  - Generiere eine Menge von N Samples zur Schätzung des Wertes von f oder seiner Erwartung
  - Aggregiere die Ergebnisse:

$$E_P[f(\boldsymbol{r})] \approx \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{r}_i)$$

  - Man kann als f die Indikatorfunktion $\mathbf{1}$ wählen, die 1 ist, wenn $\boldsymbol{R} = \boldsymbol{r}$ und 0 sonst
    - Nutzen wir auf den folgenden Folien
- Genauigkeit hängt für gewöhnlich von der Anzahl an Samples N ab
  - Dann gilt das *Gesetz der großen Zahlen*

Sampling zur Anfragebeantwortung: Direktes Sampling mit Evidenz

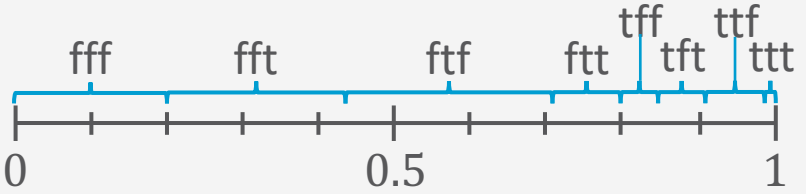
- Gegeben eine (vollständige gemeinsame) Wahrscheinlichkeitsverteilung $P_{\boldsymbol{R}}$ über Zufallsvariablen $\boldsymbol{R}$, Evidenz $\boldsymbol{e}$ und Anfragevariable $R \in \boldsymbol{R}$
- Vorgehen ähnlich zu vorher
  1. Generierung einer Menge von Samples über $\boldsymbol{R}$
     a. Aufteilung der Verteilung auf $[0,1]$
     b. Generierung von Samples
     - Unabhängig von $\boldsymbol{e}$
  2. Basierend auf der Menge von Samples, schätze $P(R|\boldsymbol{e})$ durch Zählen
     - Wie häufig kommt $R = r$ für jedes $r \in \mathrm{Val}(R)$ in den Samples, die mit $\boldsymbol{e}$ übereinstimmen, vor?
       - Verwirft (*reject*) die Samples, die nicht mit $\boldsymbol{e}$ übereinstimmen

Epid	*Travel*	*Sick*	*P*
false	*false*	*false*	0.20
false	*false*	*true*	0.24
false	*true*	*false*	0.28
false	*true*	*true*	0.08
true	*false*	*false*	0.05
true	*false*	*true*	0.06
true	*true*	*false*	0.07
true	*true*	*true*	0.02

Grundidee des *Rejection Samplings*

Sampling zur Anfragebeantwortung: 2. Abschätzen mit Evidenz

2. Gegeben die Menge an generierten Samples $\{\boldsymbol{r}_k\}_{k=1}^{N}$
   - Zähle, wie häufig in allen $\boldsymbol{r}_k$, die $\boldsymbol{e} = \{sick\}$ aufweisen, die unterschiedlichen Werte von R vorkommen:
     - $\{\boldsymbol{r}_k\}_{k=1}^{N'} = \left\{\boldsymbol{r}_k \mid \pi_{\mathrm{rv}(\boldsymbol{e})}(\boldsymbol{r}_k) = \boldsymbol{e}, k \in \{1, \dots, N\}\right\}$
     - Für jedes $r \in \mathrm{Val}(R)$: $n_r = \sum_{k=1}^{N'} 1 \mid \pi_R(\boldsymbol{r}_k) = r$
   - Gebe aus: $\hat{P}(R \mid \boldsymbol{e}) = \left(\frac{n_1}{n}, \dots, \frac{n_l}{n}\right)$
     - $n = \sum_{r \in \mathrm{Val}(R)} n_r$, $l = |\mathrm{Val}(R)|$
     - $\hat{P}(R \mid \boldsymbol{e}) \approx P(R \mid \boldsymbol{e})$
   - Beispiel: $P(Travel \mid sick)$
     - $(f, f, t), (t, t, f), (f, t, t), (t, f, t), \dots$
     - Annahme: $n_0 = 201, n_1 = 74, n = 275$



Man verwirft die Samples, die nicht der Evidenz entsprechen (*rejection*).

$Travel$	$\hat{P}(Travel \mid sick)$
$false$	$\frac{201}{275} = 0.731$
$true$	$\frac{74}{275} = 0.269$

$P(Travel \mid sick)$
0.75
0.25

$Epid$	$Travel$	$Sick$	P
$false$	$false$	$false$	0.20
$false$	$false$	$true$	0.24
$false$	$true$	$false$	0.28
$false$	$true$	$true$	0.08
$true$	$false$	$false$	0.05
$true$	$false$	$true$	0.06
$true$	$true$	$false$	0.07
$true$	$true$	$true$	0.02

Sampling zur Approximation in BNs: *Forward + Rejection Sampling*

- Im Vergleich zum Sampling aus einer vollständigen gemeinsamen Verteilung, Faktorisierung des BNs nutzen
- Verschränkte Ausführung: Gegeben ein BN B und Anfragevariable R, führe N mal aus:
  1. Generiere Sample $\boldsymbol{u}$, indem gegeben einer topologische Sortierung $\mathcal{U} = (U_1, \dots, U_n)$ Werte entlang $\mathcal{U}$ für jedes U_i gegeben der schon gesampelten Werte der Elternknoten $\mathrm{Pa}(U_i)$ gesampelt werden
  2. In einem Zähl-Vektor mit $|\mathrm{Val}(R)|$ Einträgen, erhöhe den Zähler um 1 bei $r = \pi_R(\boldsymbol{u})$
  - Ausgabe ist dann wieder $\hat{P}(R) = \left(\frac{n_1}{n}, \dots, \frac{n_l}{n}\right)$ nach Normalisierung der Vektoreinträge
  - Genannt *Forward Sampling*
- Zusätzlich gegeben Evidenz $\boldsymbol{e}$: nur Samples berücksichtigen, die zu $\boldsymbol{e}$ passen
  - Genannt *Rejection Sampling*

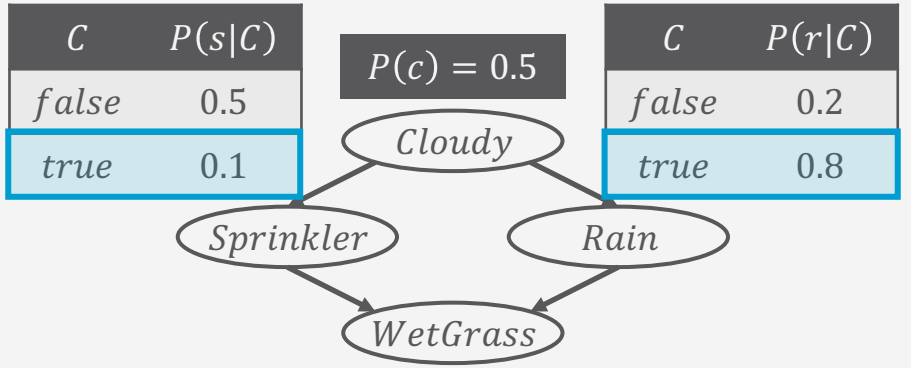
1. Samples in BNs generieren

- Gegeben eine topologische Sortierung $\mathcal{U} = (U_1, \dots, U_n)$
- Für jedes U_i, sample einen Wert aus seiner CPD $P(U_i \mid \mathrm{Pa}(U_i))$ gegeben der schon gesampelten Werte der Elternknoten $\mathrm{Pa}(U_i)$, i.e., aus

$$P\left(U_i \mid \pi_{\mathrm{Pa}(U_i)}(u_1, \dots, u_{i-1})\right)$$

  - Kann man wie auf den vorherigen Folien gezeigt sampeln
- Beispiel: $\mathcal{U} = (C, S, R, W)$
  - Sample aus $P(C) = (0.5, 0.5) \rightarrow true$ (true / false) $\rightarrow [c, \ , \ , \ ]$
  - Sample aus $P(S|c) = (0.1, 0.9) \rightarrow true$ $\rightarrow [c, s, \ , \ ]$
  - Sample aus $P(R|c) = (0.8, 0.2) \rightarrow true$ $\rightarrow [c, s, r, \ ]$
  - Sample aus $P(W|s, r) = (0.99, 0.01) \rightarrow true$ $\rightarrow [c, s, r, w]$

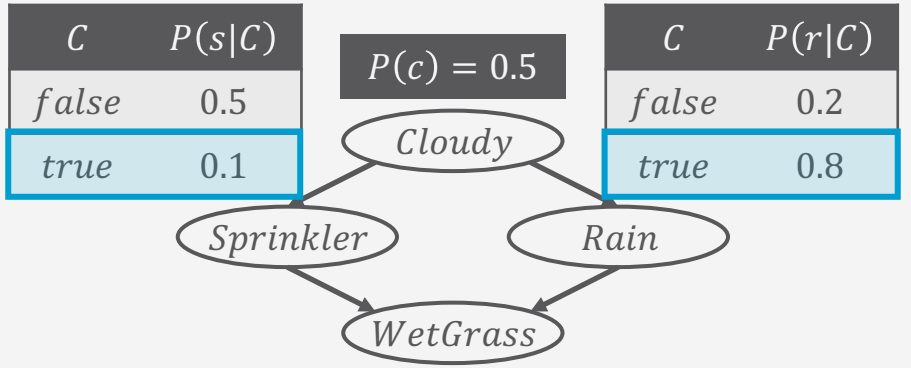
Sample



S	R	P(w|S,R)
false	false	0.9
false	true	0.9
true	false	0.9
true	true	0.99

2. Zählen und Ausgabe

- Gegeben Sample $\boldsymbol{u}$ und Anfragevariable R
- In einem Zähl-Vektor N mit $|\mathrm{Val}(R)|$ Einträgen, erhöhe den Zähler um 1 bei $r = \pi_R(\boldsymbol{u})$
- Am Ende die Einträge in N normalisieren, indem jeder Eintrag durch die Summe der Einträge geteilt wird
- Beispiel: Anfragevariable $Rain$
  - Sample ist $[c, s, r, w]$
  - Erhöhe Zähler für r
  - Bei $\mathrm{N} = [321,310]$, Ergebnis ist $\mathrm{N} = [0.509,0.491]$
  - Bei zusätzlicher Evidenz $WetGrass = false$, i.e.,$\neg w$
    - Verwerfe Sample, da $w \neq \neg w$



S	R	$P(w\|S,R)$
$false$	$false$	0.9
$false$	$true$	0.9
$true$	$false$	0.9
$true$	$true$	0.99

Rejection Sampling

RejectionSampling$(R, \boldsymbol{e}, B, N, \mathcal{U})$ ▷ Forward Sampling bei $\boldsymbol{e} = \emptyset$

Vektor N der Länge $|\mathrm{Val}(R)|$, anfangs 0 ▷ Speichert Zählerstände für alle $r \in \mathrm{Val}(R)$

for $t = 1, \ldots, N$ **do**

Sample $\boldsymbol{r} \leftarrow \mathrm{PriorSample}(B, \mathcal{U})$

if $\pi_{\mathrm{rv}(\boldsymbol{e})}(\boldsymbol{r}) = \boldsymbol{e}$ **then** ▷ Immer wahr bei $\boldsymbol{e} = \emptyset$

$\mathrm{N}[r] \leftarrow \mathrm{N}[r] + 1$ mit $r = \pi_{\mathrm{rv}(R)}(\boldsymbol{r})$

return Normalise(N) ▷ Jeden Eintrag durch die Summe der Einträge teilen

---

PriorSample$(B, \mathcal{U})$

Leeres Sample $\boldsymbol{u} \leftarrow \left(\perp_1, \ldots, \perp_{\mathrm{len}(\mathcal{U})}\right)$

for $i = 1, \ldots, \mathrm{len}(\mathcal{U})$ **do**

$\boldsymbol{u}[i] \leftarrow$ Sample Wert aus $P\left(U_i \,|\pi_{\mathrm{Pa}(U_i)}(\boldsymbol{u}[1\!:i-1])\right)$

return $\boldsymbol{u}$

Rejection Sampling

M. Henrion: “Propagating uncertainty in Bayesian networks by probabilistic logic sampling”. In: *Uncertainty in AI*, 1988.

Forward Sampling: Analyse

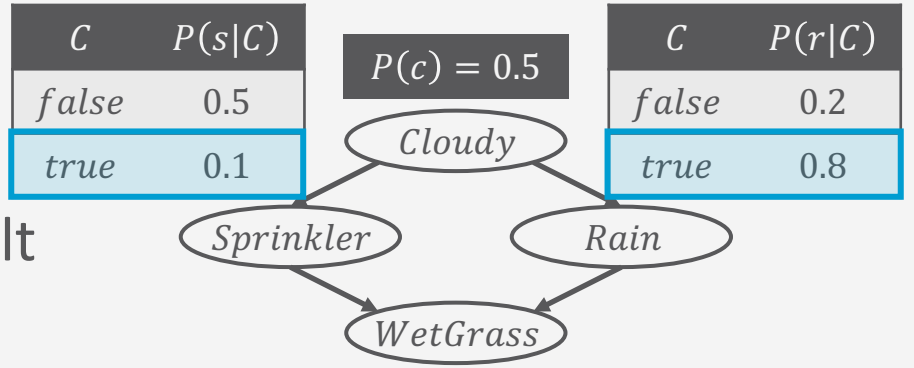
- Wahrscheinlichkeit eines Samples aus **PriorSample**:

$$S_{PS}(u_1, \dots, u_n) = \prod_{i=1}^{n} P\big(u_i \mid \mathrm{Pa}(U_i)\big) = P(u_1, \dots, u_n)$$

  - Wahre Wahrscheinlichkeit
  - Beispiel: $S_{PS}(c, s, r, w) = 0.5 \cdot 0.1 \cdot 0.8 \cdot 0.99 = 0.0396$
- Mit $N_{PS}(u_1, \dots, u_n)$ die Anzahl an Samples für $u_1, \dots, u_n$ gilt

$$\begin{aligned}\lim_{N\to\infty} \hat{P}(u_1, \dots, u_n) &= \lim_{N\to\infty} \frac{N_{PS}(u_1, \dots, u_n)}{N} \\ &= S_{PS}(u_1, \dots, u_n) \\ &= P(u_1, \dots, u_n)\end{aligned}$$

- Damit ist Schätzung mittels **PriorSample** konsistent
  - I.e., $\hat{P}(u_1, \dots, u_n) \approx P(u_1, \dots, u_n)$



S	R	P(w|S,R)
false	false	0.9
false	true	0.9
true	false	0.9
true	true	0.99

Kurzer Einblick in PAC

- Güte des Ergebnis hängt von der Anzahl N an Samples ab
- PAC : probably approximately correct
  - Mit Wahrscheinlichkeit $1 - \delta$...
  - ... ist der Fehler beschränkt durch ε
- Benötigte Anzahl N um eine Schätzung mit (ε, δ)-Zuverlässigkeit für eine Anfrage $P(\boldsymbol{e})$ zu erhalten, dessen Fehler durch ε mit einer Wahrscheinlichkeit von $1 - \delta$ beschränkt ist:

$$N \geq \frac{\ln\left(\frac{2}{\delta}\right)}{2\varepsilon^2}$$

[Nutzt die so genannte Hoeffding-Schranke]

- Benötigte Anzahl N um gegeben ε eine gewisse Fehlerwahrscheinlichkeit δ zu garantieren:

$$N \geq 3\frac{\ln\left(\frac{2}{\delta}\right)}{P(\boldsymbol{e})\varepsilon^2}$$

[Nutzt die so genannte Chernov-Schranke]

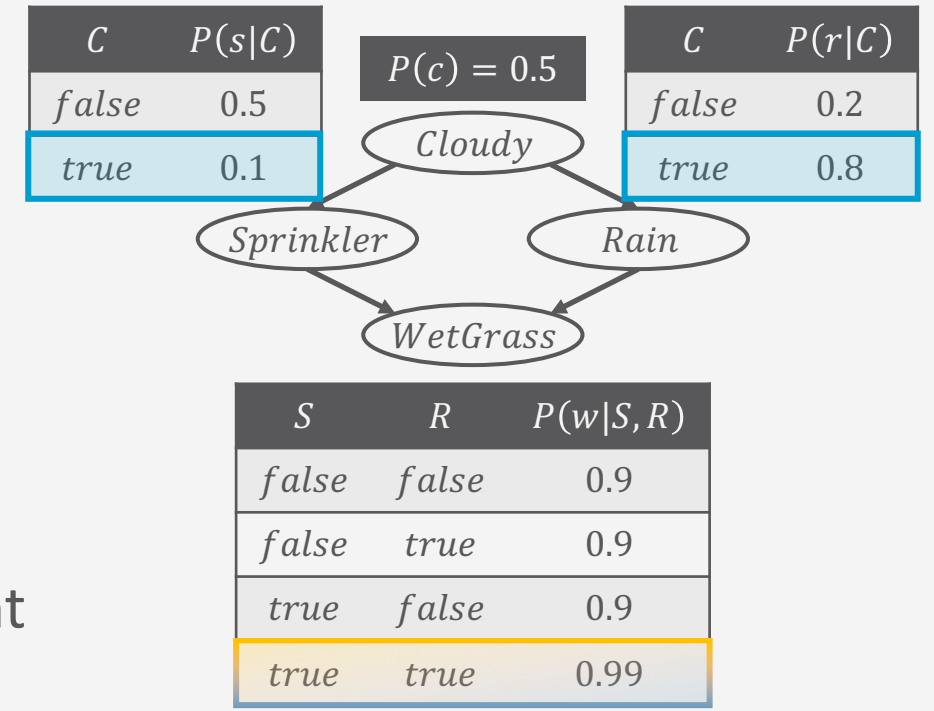
Rejection Sampling: Analyse

- Mit $N_{PS}(r, \boldsymbol{e})$ die Anzahl an Samples, die $r, \boldsymbol{e}$ erfüllen, gilt:

$$\hat{P}(r \mid \boldsymbol{e}) = \frac{1}{Z} N_{PS}(r, \boldsymbol{e})$$
$$= \frac{N_{PS}(r, \boldsymbol{e})}{N_{PS}(\boldsymbol{e})}$$
$$\approx \frac{P(r \mid \boldsymbol{e})}{P(\boldsymbol{e})}$$
$$= P(r \mid \boldsymbol{e})$$

- Damit ist Schätzung mittels **RejectionSampling** konsistent

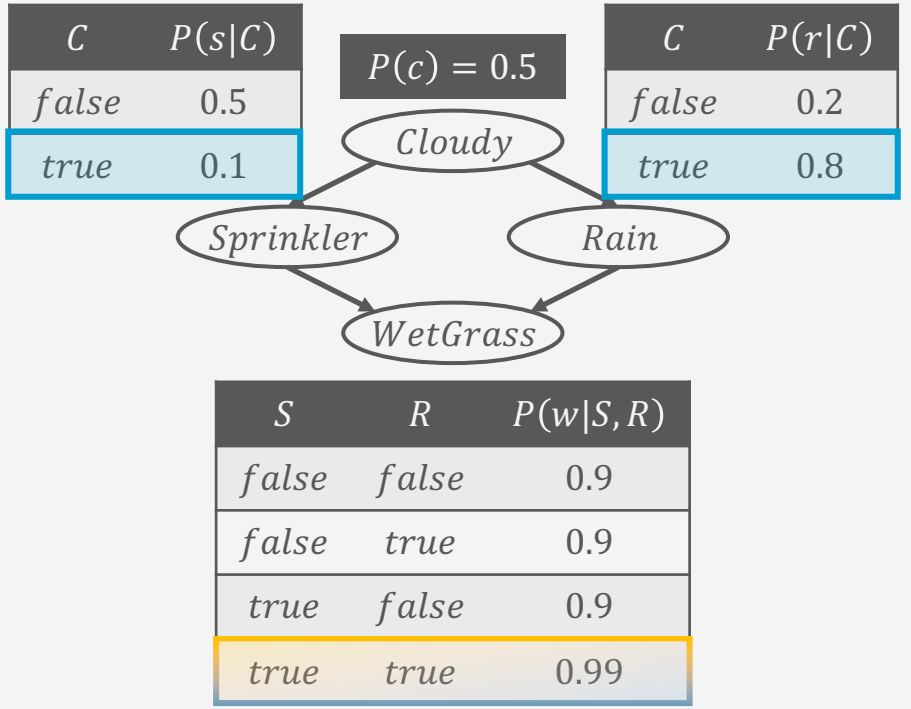
Was ist das Problem mit dem Vorgehen?



Rejection Sampling: Problem

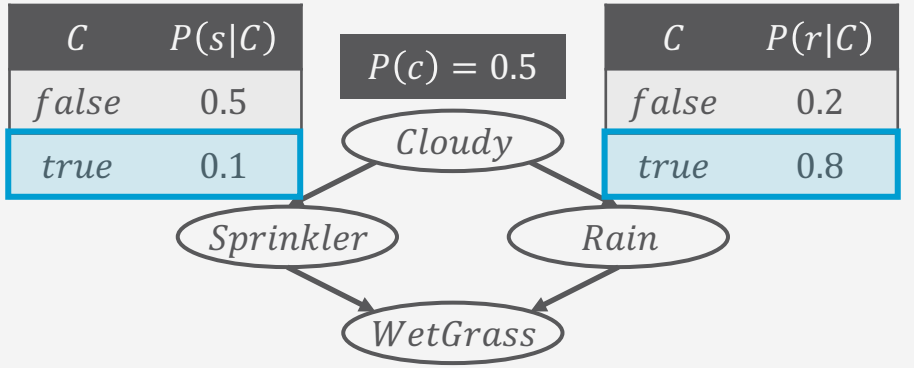
- Beispiel
  - Schätzung von $P(Rain \mid Sprinkler = true)$ mit 100 Samples:
    - 73 Samples → $Sprinkler = false$
    - 27 Samples → $Sprinkler = true$
      - 8 Samples → $Rain = true$
      - 19 Samples → $Rain = false$
  - $P(Rain \mid Sprinkler = true)$
    $= \text{Normalise}\big((8,19)\big)$
    $= (0.296, 0.704)$

- Zu viele Samples werden verworfen



Rejection Sampling: Problem

- Sampling hoffnungslos teuer, wenn $P(\boldsymbol{e})$ klein
  - Da Wahrscheinlichkeit Samples zu generieren, in denen $\boldsymbol{e}$ gilt, klein
    - $P(\boldsymbol{e}) = 0.001, N = 10000 \rightarrow M \approx 10$ nicht verworfene Samples
    - Gilt z.B. für jede Menge von Symptomen in medizinischen Diagnosesystemen
  - $P(\boldsymbol{e})$ wird exponentiell kleiner mit steigender Anzahl an $\boldsymbol{e}$
- Wenn M^* die Anzahl an nicht verworfenen Samples ist, die wir haben wollen, dann müssen wir rund $N = \frac{M^*}{P(\boldsymbol{e})}$ Samples generieren
  - $M^* = 10000, P(\boldsymbol{e}) = 0.001 \rightarrow N = 10{,}000{,}000$



C	$P(s\|C)$
$false$	0.5
$true$	0.1

C	$P(r\|C)$
$false$	0.2
$true$	0.8

S	R	$P(w\|S,R)$
$false$	$false$	0.9
$false$	$true$	0.9
$true$	$false$	0.9
$true$	$true$	0.99

Likelihood Weighting

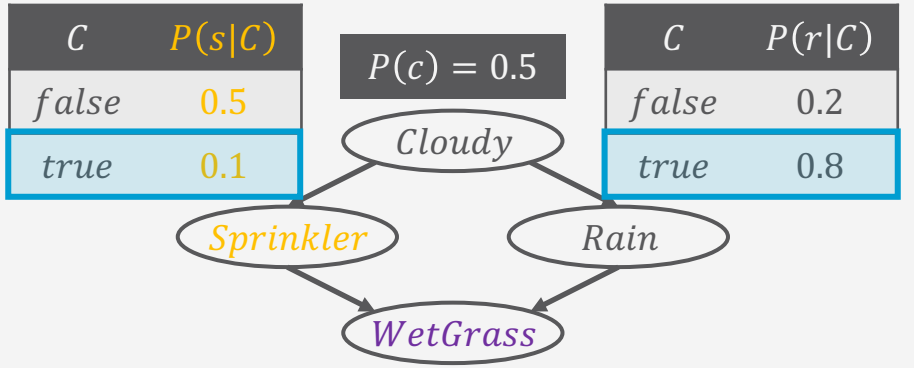
- Ziel
  - Ineffizienz von Rejection Sampling vermeiden

- Idee
  - Nur Samples (zusammengesetzte Ereignisse) generieren, die zu der Evidenz $\boldsymbol{e}$ passen
  - Jedes Ereignis gewichtet mit der Wahrscheinlichkeit (*likelihood*), dass das Ereignis zur Evidenz passt

R. Fung and K.-C. Chang. Weighing and integrating evidence for stochastic simulation in Bayesian networks. In *Uncertainty in AI*, 1989.
R. D. Shachter and M. A. Peot. Simulation approaches to general probabilistic inference on belief networks. In *Uncertainty in AI*, 1989.

Likelihood Weighting: Beispiel

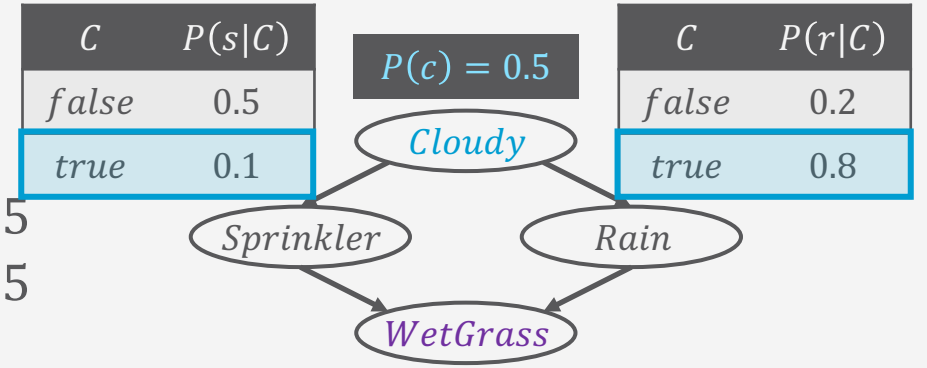
- $P(R \mid s, w)$?
  - Topologische Sortierung: (C, S, R, W)
- Sample generieren
  - Gewicht w auf 1.0 setzen $\rightarrow [\ ,\ ,\ ,\ ], w = 1.0$
  - Sample aus $P(C) = (0.5, 0.5) \rightarrow true$ $\rightarrow [c,\ ,\ ,\ ], w = 1.0$
  - S ist eine Evidenzvariable mit Wert $true$
    $w \leftarrow w \cdot P(s \mid c) = 0.1$ $\rightarrow [c, s,\ ,\ ], w = 0.1$
  - Sample aus $P(R \mid c) = (0.8, 0.2) \rightarrow true$ $\rightarrow [c, s, r,\ ], w = 0.1$
  - W ist eine Evidenzvariable mit Wert $true$
    $w \leftarrow w \cdot P(w \mid s, r) = 0.099$ $\rightarrow [c, s, r, w], w = 0.099$
- Schätzung
  - Gewichte akkumulieren zu $R = true$ oder $R = false$
    - Obiges Sample beinhaltet r → Geht an $R = true$ mit Gewicht 0.099
  - Normalisieren (durch die Summe der akkumulierten Gewichte teilen)



S	R	$P(w \mid S, R)$
$false$	$false$	0.9
$false$	$true$	0.9
$true$	$false$	0.9
$true$	$true$	0.99

Likelihood Weighting: Beispiel

- $P(R \mid c, w)$?
  - Topologische Sortierung: (C, S, R, W)
- Sample generieren
  - Gewicht w auf 1.0 setzen $\rightarrow [\ ,\ ,\ ,\ ], w = 1.0$
  - C ist eine Evidenzvariable mit Wert $true$
    $w \leftarrow w \cdot P(c) = 0.5$ $\rightarrow [c,\ ,\ ,\ ], w = 0.5$
  - Sample aus $P(S \mid c) = (0.1, 0.9) \rightarrow false$ $\rightarrow [c, \neg s,\ ,\ ], w = 0.5$
  - Sample aus $P(R \mid c) = (0.8, 0.2) \rightarrow true$ $\rightarrow [c, \neg s, r,\ ], w = 0.5$
  - W ist eine Evidenzvariable mit Wert $true$
    $w \leftarrow w \cdot P(w \mid \neg s, r) = 0.45$ $\rightarrow [c, \neg s, r, w], w = 0.45$
- Schätzung
  - Gewichte akkumulieren zu $R = true$ oder $R = false$
    - Obiges Sample beinhaltet $r \rightarrow$ Geht an $R = true$ mit Gewicht 0.45
  - Normalisieren (durch die Summe der akkumulierten Gewichte teilen)



S	R	$P(w \mid S, R)$
$false$	$false$	0.9
$false$	$true$	0.9
$true$	$false$	0.9
$true$	$true$	0.99

Liklihood Weighting

LikelihoodWeighting$(R, \boldsymbol{e}, B, N, \mathcal{U})$
  Vektor W der Länge $|\mathrm{Val}(R)|$, anfangs 0 ▷ *Gewichtete* Zählerstände für alle $r \in \mathrm{Val}(R)$
  for $t = 1, \ldots, N$ **do**
    $\boldsymbol{r}, w \leftarrow \mathrm{WeightedSample}(B, \mathcal{U}, \boldsymbol{e})$
    $W[r] \leftarrow W[r] + w$ mit $r = \pi_{\mathrm{rv}(R)}(\boldsymbol{r})$
  return Normalise(W) ▷ Jeden Eintrag durch die Summe der Einträge teilen

WeightedSample$(B, \mathcal{U}, \boldsymbol{e})$
  Leeres Sample $\boldsymbol{u} \leftarrow (\bot_1, \ldots, \bot_{\mathrm{len}(\mathcal{U})})$; Gewicht $w \leftarrow 1.0$
  for $i = 1, \ldots, \mathrm{len}(\mathcal{U})$ **do**
    if $U_i \in \mathrm{rv}(\boldsymbol{e})$ **then**
      $\boldsymbol{u}[i] \leftarrow \pi_{\mathrm{rv}(U_i)}(\boldsymbol{e}); w \leftarrow w \cdot P\left(\pi_{\mathrm{rv}(U_i)}(\boldsymbol{e}) \mid \pi_{\mathrm{Pa}(U_i)}(\boldsymbol{u}[1:i-1])\right)$
    else
      $\boldsymbol{u}[i] \leftarrow$ Sample Wert aus $P\left(U_i \mid \pi_{\mathrm{Pa}(U_i)}(\boldsymbol{u}[1:i-1])\right)$
  return $\boldsymbol{u}, w$

Likelihood Weighting

Likelihood Weighting: Analyse

- Sampling Wahrscheinlichkeit für WeightedSample $\boldsymbol{u}$ bestehend aus Evidenz $\boldsymbol{e}$ und gesampelten Werten $\boldsymbol{r'}$, i.e., $\mathrm{rv}(\boldsymbol{r'}) = \mathrm{rv}(B) \setminus \mathrm{rv}(\boldsymbol{e})$

$$S_{WS}(\boldsymbol{r'}, \boldsymbol{e}) = \prod_{r' \in \boldsymbol{r'}} P\left(r' \mid \pi_{\mathrm{Pa}(R')}(\boldsymbol{u})\right)$$

- Gewicht für ein gegebenes Sample $\boldsymbol{u}$ mit $\boldsymbol{r'}, \boldsymbol{e}$

$$w(\boldsymbol{r'}, \boldsymbol{e}) = \prod_{e \in \boldsymbol{e}} P\left(e \mid \pi_{\mathrm{Pa}(E)}(\boldsymbol{u})\right)$$

- Gewichtete Sampling Wahrscheinlichkeit ist

$$S_{WS}(\boldsymbol{r'}, \boldsymbol{e}) \cdot w(\boldsymbol{r'}, \boldsymbol{e}) = \prod_{r' \in \boldsymbol{r'}} P\left(r' \mid \pi_{\mathrm{Pa}(R')}(\boldsymbol{u})\right) \cdot \prod_{e \in \boldsymbol{e}} P\left(e \mid \pi_{\mathrm{Pa}(E)}(\boldsymbol{u})\right) = P(\boldsymbol{r'}, \boldsymbol{e})$$

  - Letzter Schritt durch die Semantik von BNs
- Damit ist Schätzung mittels **LikelihoodWeighting** konsistent

Importance Sampling

- Likelihood Weighting ein Spezialfall des so genannten Importance Samplings für BNs
- Erinnerung: Schätze die Erwartung einer Funktion $f(\boldsymbol{R})$ relativ zu einer Verteilung $P(\boldsymbol{R})$

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} f(\boldsymbol{r}) P(\boldsymbol{r})$$

  - $P(\boldsymbol{R})$ typischerweise Zielverteilung (*target distribution*) genannt
  - Generiere eine Menge von N Samples und aggregiere die Ergebnisse: $E_P[f(\boldsymbol{r})] \approx \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{r}_i)$
  - Was wir bisher gemacht haben
- Wenn Generierung von Samples aus P aufwendig (oder schwer möglich), benutze (einfachere) Vorschlagsverteilung (*proposal distribution*) Q
  - Aus einem BN sampeln recht einfach, aber zum Beispiel so nicht möglich bei Faktormodellen
    - Keine topologische Sortierung, keine lokalen *Wahrscheinlichkeit*sverteilungen

Herman Kahn. Random Sampling (Monte Carlo) Techniques in Neutron Attenuation Problems. Nucleonics, 1950.

Vorschlagsverteilung (*proposal distribution*) nutzen

- Bedingung: Vorschlagsverteilung Q dominiert P
  - I.e., $Q(\boldsymbol{r}) > 0$, wann immer $P(\boldsymbol{r}) > 0$
    - Q darf keine Zustände ignorieren, die Wahrscheinlichkeiten ungleich null mit P haben
    - Genauer: Träger von Q muss Träger von P beinhalten
      - Träger einer Verteilung S sind alle Punkte $\boldsymbol{r}$, so dass gilt $S(\boldsymbol{r}) > 0$
  - Im Allgemeinen Q willkürlich, aber Performanz hängt in hohem Grade davon ab, wie ähnlich Q zu P ist
    - Beispiel: Wahrscheinlichkeiten nahe null in $Q(\boldsymbol{r})$ nur dann, wenn $f(\boldsymbol{r})P(\boldsymbol{r})$ auch sehr klein
      - Varianz klein halten

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] = \sum_{\boldsymbol{r} \in \mathcal{R}(\boldsymbol{R})} f(\boldsymbol{r})P(\boldsymbol{r})$$

- Generiere Samples aus Q anstatt aus P
  - Man kann nicht direkt die f-Werte der generierten Samples mitteln
    → Schätzer *anpassen*, um die falsche Sampling-Verteilung zu kompensieren

Unnormalisiertes Importance Sampling

- Anpassung:

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] = \sum_{\boldsymbol{r}\in \mathrm{Val}(\boldsymbol{R})} f(\boldsymbol{r})P(\boldsymbol{r}) = \sum_{\boldsymbol{r}\in \mathrm{Val}(\boldsymbol{R})} Q(\boldsymbol{r})f(\boldsymbol{r})\frac{P(\boldsymbol{r})}{Q(\boldsymbol{r})} = E_{Q(\boldsymbol{R})}\left[f(\boldsymbol{R})\frac{P(\boldsymbol{R})}{Q(\boldsymbol{R})}\right]$$

  - Korrektur: $\frac{P(\boldsymbol{R})}{Q(\boldsymbol{R})}$
- Generiere eine Menge von Samples aus Q und dann schätze:

Annahme: P ist bekannt

$$E_P[f(\boldsymbol{r})] \approx \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{r}_i)\frac{P(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)} = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{r}_i)w(\boldsymbol{r}_i)$$

$$w(\boldsymbol{r}_i) \stackrel{\text{def}}{=} \frac{P(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$$

Wenn P bekannt ist: Likelihood Weighting als Importance Sampling

$$w(\boldsymbol{r}_i) \stackrel{\text{def}}{=} \frac{P(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$$

- BN B ohne Evidenz $\boldsymbol{e}$ ist P
- BN B mit absorbierter Evidenz $\boldsymbol{e}$ $(= B')$ ist Q
- Samples werden gemäß der Wahrscheinlichkeiten in den CPDs gewichtet
  - Wahrscheinlichkeit eines Samples $\boldsymbol{r}' \cup \boldsymbol{e}$ in B passend zu $\boldsymbol{e}$, i.e., P:

    $$P(\boldsymbol{r}' \mid \boldsymbol{e}) = \frac{P(\boldsymbol{r}', \boldsymbol{e})}{P(\boldsymbol{e})} = \frac{\prod_{r \in (\boldsymbol{r}' \cup \boldsymbol{e})} P\left(r \mid \pi_{\mathrm{Pa}(R)}(\boldsymbol{r}' \cup \boldsymbol{e})\right)}{P(\boldsymbol{e})}$$

  - Wahrscheinlichkeit eines Samples $\boldsymbol{r}'$ in B', i.e., Q: $Q(\boldsymbol{r}') = \prod_{r' \in \boldsymbol{r}'} P\left(r' \mid \pi_{\mathrm{Pa}(R')}(\boldsymbol{r}')\right)$
  - Gewicht

$$w(\boldsymbol{r}', \boldsymbol{e}) = \frac{P(\boldsymbol{r}' \mid \boldsymbol{e})}{Q(\boldsymbol{r}')} = \frac{\prod_{r \in (\boldsymbol{r}' \cup \boldsymbol{e})} P\left(r \mid \pi_{\mathrm{Pa}(R)}(\boldsymbol{r}' \cup \boldsymbol{e})\right)}{P(\boldsymbol{e}) \prod_{r' \in \boldsymbol{r}'} P\left(r' \mid \pi_{\mathrm{Pa}(R')}(\boldsymbol{r}')\right)} = \frac{\prod_{e \in \boldsymbol{e}} P\left(e \mid \pi_{\mathrm{Pa}(E)}(\boldsymbol{r})\right)}{P(\boldsymbol{e})} \Rightarrow w(\boldsymbol{r}', \boldsymbol{e}) = \prod_{e \in \boldsymbol{e}} P\left(e \mid \mathrm{Pa}(E)\right)$$

  - $P(\boldsymbol{e})$ identisch für alle Samples → weglassen okay

Likelihood Weighting: Analyse

$$w(\boldsymbol{r}', \boldsymbol{e}) = \prod_{e \in \boldsymbol{e}} P\left(e \mid \pi_{\mathrm{Pa}(E)}(\boldsymbol{u})\right)$$

$$w(\boldsymbol{r}_i) \stackrel{\text{def}}{=} \frac{P(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$$

Wenn P nicht bekannt ist

- Der am häufigsten vorkommende Grund aus Q zu sampeln:
  P nur bekannt bis auf die Normalisierungskonstante Z
  - Zugriff auf Funktion $\tilde{P}$, welche keine normalisierte Verteilung ist, sondern $\tilde{P}(\boldsymbol{R}) = P(\boldsymbol{R}) \cdot Z$
- Normalisiertes Importance Sampling
  - Definiere $w(\boldsymbol{r}_i)$ über $\tilde{P}$: $w(\boldsymbol{r}_i) \stackrel{\text{def}}{=} \frac{\tilde{P}(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$
  - Schätzung funktioniert nicht mehr (keine Wahrscheinlichkeitsverteilung):

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] \neq \sum_{\boldsymbol{r} \in \text{Val}(\boldsymbol{R})} f(\boldsymbol{r})\tilde{P}(\boldsymbol{r}) = \sum_{\boldsymbol{r} \in \text{Val}(\boldsymbol{R})} Q(\boldsymbol{r}) f(\boldsymbol{r}) \frac{\tilde{P}(\boldsymbol{r})}{Q(\boldsymbol{r})} \neq E_{Q(\boldsymbol{R})}\left[f(\boldsymbol{R}) \frac{\tilde{P}(\boldsymbol{r})}{Q(\boldsymbol{R})}\right]$$

$$E_P[f(\boldsymbol{r})] \approx \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{r}_i) \frac{P(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$$

Normalisiertes Importance Sampling

$$w(\boldsymbol{r}_i) \stackrel{\text{def}}{=} \frac{\tilde{P}(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$$

- Trick: $w(\boldsymbol{R})$ als Zufallsvariable auffassen mit Erwartungswert Z:

$$E_{Q(\boldsymbol{R})}[w(\boldsymbol{R})] = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} Q(\boldsymbol{r}) w(\boldsymbol{r}) = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} Q(\boldsymbol{r}) \frac{\tilde{P}(\boldsymbol{r})}{Q(\boldsymbol{r})} = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} \tilde{P}(\boldsymbol{r}) = Z$$

- Gegeben $E_{Q(\boldsymbol{R})}[w(\boldsymbol{R})] = Z$

$$E_{P(\boldsymbol{R})}[f(\boldsymbol{R})] = \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} f(\boldsymbol{r}) P(\boldsymbol{r}) = \frac{1}{Z} \sum_{\boldsymbol{r} \in \mathrm{Val}(\boldsymbol{R})} Q(\boldsymbol{r}) f(\boldsymbol{r}) \frac{\tilde{P}(\boldsymbol{r})}{Q(\boldsymbol{r})} = \frac{1}{Z} E_{Q(\boldsymbol{R})}[f(\boldsymbol{R}) w(\boldsymbol{R})] = \frac{E_{Q(\boldsymbol{R})}[f(\boldsymbol{R}) w(\boldsymbol{R})]}{E_{Q(\boldsymbol{R})}[w(\boldsymbol{R})]}$$

- Schätzer für Zähler und Nenner verwenden: $E_P[f(\boldsymbol{r})] \approx \frac{\sum_{i=1}^{N} f(\boldsymbol{r}_i) w(\boldsymbol{r}_i)}{\sum_{i=1}^{N} w(\boldsymbol{r}_i)}$

Sampling in Faktormodellen

- Angenommen es gibt eine Vorschlagsverteilung $Q(\boldsymbol{R})$, die einfaches Sampling ermöglicht
- Dann können wir Samples $\boldsymbol{r}_i$ generieren und entsprechend gewichten durch $w(\boldsymbol{r}_i) = \frac{\tilde{P}(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}$
  - wobei $\tilde{P}(\boldsymbol{r}_i) = \prod_{f \in F} \phi_f \left(\pi_{\mathrm{rv}(f)}(\boldsymbol{r}_i)\right)$
    - Produkt der Potentiale, auf die die Argumente der Faktoren mit den Werten in $\boldsymbol{r}_i$ abbilden
- Wenn wir das für alle Samples machen, schätzen wir Z (bzw. $P(\boldsymbol{e})$):

$$E_{P(\boldsymbol{R})}[w(\boldsymbol{R})] \approx \sum_{\boldsymbol{r} \in \mathcal{R}(\boldsymbol{R})} \frac{\tilde{P}(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)} = Z$$

- Wenn wir in $P(r|\boldsymbol{e})$ interessiert sind: $E_P[f(r)] \approx \frac{\sum_{i=1}^{N} f(r) \frac{\tilde{P}(r)}{Q(r)}}{\sum_{i=1}^{N} \frac{\tilde{P}(\boldsymbol{r}_i)}{Q(\boldsymbol{r}_i)}}$

Woher bekommen wir Q? – Eine Idee

- Gegeben ein Faktormodell F

1. Transformiere F in ein Modell F', so dass die Faktoren über maximale Cliquen gehen
   - Im Jtree: Multipliziere alle Faktoren in jedem lokalen Modell, so dass es einen lokalen Faktor $f_i = \phi_i(R_1, \dots, R_{k_i})$ über die Clustervariablen gibt
2. Transformiere F' in ein „quasi“ BN F^{BN}:
   - Wähle ein Cluster als Wurzel und richte alle Kanten weg von dem Cluster aus → „gerichteter Jtree“
   - Normalisiere wie folgt:
     - Wurzel-Cluster: marginal über die Clustervariablen (Potentiale addieren sich auf zu 1)
     - Alle anderen Cluster: bedingt über die Separatorvariablen des „Elternknotens“ im gerichteten Jtree

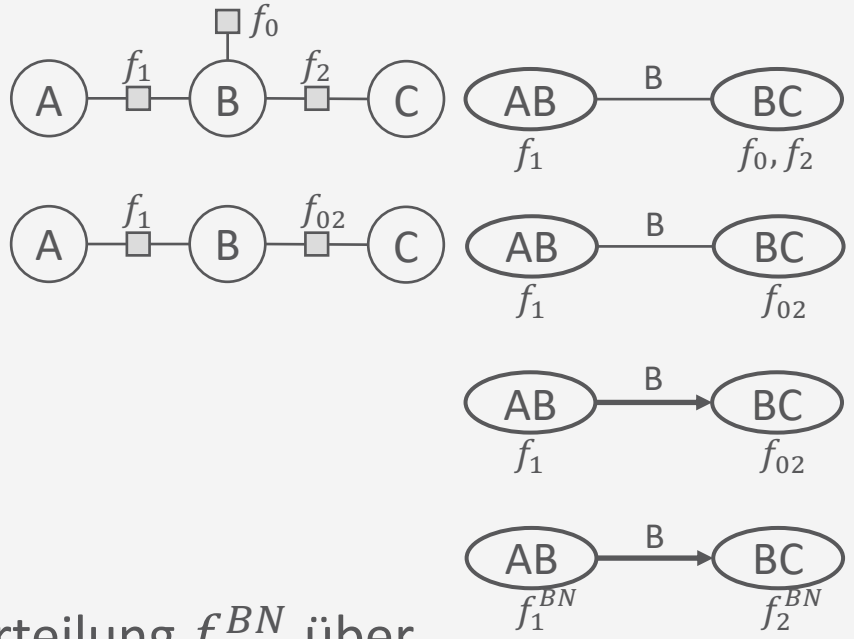
       → Erzwingt eine Faktorisierung in (bedingte) Wahrscheinlichkeitsverteilungen mit $Z = 1$

       → Legt eine Art topologische Sortierung für die Zufallsvariablen in F fest
       - Wurzel-Clustervariablen zuerst, gefolgt von den Variablen in den Nachfolger-Clusters im gerichteten Jtree

- Sample aus $Q = F^{BN}$ z.B. mittels Likelihood Weighting

Beispiel

- Faktormodell F mit Jtree J
- Faktormodell F', max. Cliquen
  - $f_{02} = f_2 \cdot f_0$
- Wähle ein Cluster als Wurzel
  - $\boldsymbol{C}_1 = \{\text{AB}\}$
- Normalisiere
  - $\boldsymbol{C}_1$, so dass f_1 eine Marginalverteilung f_1^{BN} über $\boldsymbol{C}_1$ ist
  - $\boldsymbol{C}_2$, so dass f_{02} eine Verteilung f_{02}^{BN} bedingt auf $\boldsymbol{S}_{12} = \boldsymbol{C}_1 \cap \boldsymbol{C}_2 = \{\text{B}\}$ über $\boldsymbol{C}_2 \backslash \boldsymbol{S}_{12} = \{\text{C}\}$ist



A	B	f_1
0	0	1
0	1	2
1	0	3
1	1	4

B	f_0
0	2
1	1

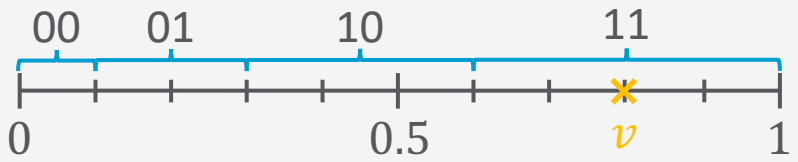
B	C	f_2
0	0	3
0	1	2
1	0	4
1	1	1

A	B	f_1^{BN}
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

B	C	f_{02}	f_{02}^{BN}
0	0	6	0.6
0	1	4	0.4
1	0	4	0.8
1	1	1	0.2

Beispiel

- Sortierung: AB, C
- Sample Werte für AB aus f_1^{BN}
  - Da es eine Wahrscheinlichkeitsverteilung ist
    - Generiere eine Zufallszahl $v \in [0,1]$ und nehme die Zuweisungen für AB, in deren Intervall v fällt



    - E.g., $v = 0.8 \rightarrow 11$
  - Bilde die Zuweisung auf die Separatorvariablen ab: → B = 1

B	f_0
0	2
1	1

A	B	f_1
0	0	1
0	1	2
1	0	3
1	1	4

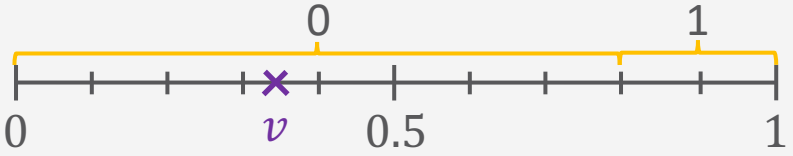
B	C	f_2
0	0	3
0	1	2
1	0	4
1	1	1

A	B	f_1^{BN}
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

B	C	f_{02}^{BN}
0	0	0.6
0	1	0.4
1	0	0.8
1	1	0.2

Beispiel

- Sortierung: AB, C
- Sample Werte für BC\B bedingt auf B = 1 aus f_{02}^{BN}
  - f_{02}^{BN} bedingt auf B = 1 ergibt eine Verteilung



  - E.g., $v = 0.35 \rightarrow 0$

- Sample: $[\overset{A}{1},\overset{B}{1},\overset{C}{0}]$
  - Q Gewicht: $Q([1,1,0]) = f_1^{BN}(1,1) \cdot f_{02}^{BN}(1,0) = 0.4 \cdot 0.8 = 0.32$
  - $\tilde{P}$ Gewicht: $\tilde{P}([1,1,0]) = f_0(1) \cdot f_1(1,1) \cdot f_2(1,0) = 1 \cdot 4 \cdot 4 = 16$
  - $w([1,1,0]) = \frac{16}{0.32} = 50$

B	f_0
0	2
1	1

A	B	f_1
0	0	1
0	1	2
1	0	3
1	1	4

B	C	f_2
0	0	3
0	1	2
1	0	4
1	1	1

A	B	f_1^{BN}
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

B	C	f_{02}^{BN}
0	0	0.6
0	1	0.4
1	0	0.8
1	1	0.2

Beispiel

- Sortierung: AB, C
- Menge von N Samples $\boldsymbol{r}_i$ mit Gewichten $w(\boldsymbol{r}_i)$
  - Beispiel: Sample $[\overset{A}{1},\overset{B}{1},\overset{C}{0}]$
    - $w([1,1,0]) = \frac{16}{0.32} = 50$
- Angenommen: Anfrage $P(C = 1)$
- Schätze

$$P(C = 1) \approx \frac{\sum_{i=1}^{N} \mathbf{1}(\boldsymbol{r}_i, C = 1) w(\boldsymbol{r}_i)}{\sum_{i=1}^{N} w(\boldsymbol{r}_i)}$$

$$\mathbf{1}(\boldsymbol{r}_i, C = 1) = \begin{cases} 1 & \pi_C(\boldsymbol{r}_i) = 1 \\ 0 & sonst \end{cases}$$

B	f_0
0	2
1	1

A	B	f_1
0	0	1
0	1	2
1	0	3
1	1	4

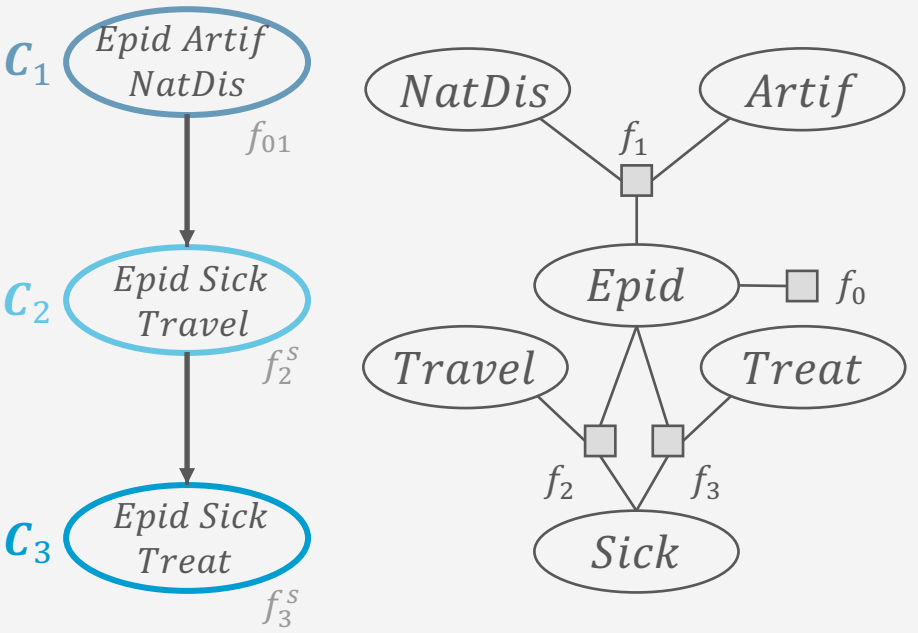
B	C	f_2
0	0	3
0	1	2
1	0	4
1	1	1

A	B	f_1^{BN}
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

B	C	f_{02}^{BN}
0	0	0.6
0	1	0.4
1	0	0.8
1	1	0.2

Beispiel

- Laufendes Beispiel aus den vorherigen Vorlesungen:
  - Jtree mit lokalen Modellen in einen Faktor multipliziert und Evidenz absorbiert (Foliensatz 3b, Folie 38)
  - Fehlt noch Ausrichtung der Kanten + Normalisierung
    - Wähle C_1: Kantenausrichtung $C_1 \rightarrow C_2$, $C_2 \rightarrow C_3$
    - Normalisierung: f_{01} durch Summe der Potentiale teilen, f_2^s auf $Epid$, f_3^s auf $Epid$ und $Sick$ (ist Evidenz) bedingen
  - Zuweisung für $Epid, Artif, NatDis$ aus f_{01}^{BN} sampeln: $[e, a, n] \rightarrow$ auf $Epid$ abbilden und an C_2 senden: e
  - Zuweisung für $Travel$ aus f_2^s gegeben e sampeln: $[e, s, tl] \rightarrow$ auf $Epid$ abbilden und an C_3 senden: e
  - Zuweisung für $Treat$ aus f_3^s gegeben e sampeln: $[e, s, tt]$
  - Gesamtsample: $[e, a, n, s, tl, tt]$



Ohne Evidenz:
- Zuweisung für $Travel, Sick$ aus f_2 gegeben $Epid = e$ sampeln: $[e, s, tl] \rightarrow$ auf $Epid, Sick$ abbilden und an C_3 senden: e, s
- Zuweisung für $Treat$ aus f_3 gegeben e, s sampeln: $[e, s, tt]$

Likelihood Weighting bzw. Importance Sampling: Problem

- Beheben das Problem der Ineffizienz des Rejection Samplings, da alle Samples verwendet werden können bzw. nur Samples erzeugt werden, die zur Evidenz passen
  - Evidenz steuert, aus welchen Verteilungen gesampelt wird
- Aber: Importance Sampling benötigt eine halbwegs passende Vorschlagsverteilung Q
  - Kann schwer zu bauen / finden sein, wenn etwas anderes als gerichtete Modelle betrachtet
    - BN: Einfach → Likelihood Weighting als Spezialfall von Importance Sampling
- Problem: Performanz lässt bei vielen Evidenzvariablen auch hier stark nach
  - Die meisten Samples haben verschwindend geringes Gewicht
  - Nur kleiner Bruchteil von Samples ordnet Evidenz mehr an Gewicht zu
    - Tragen quasi den Großteil des Gesamtgewichts
  - Damit gewichtete Schätzung von diesen Samples dominiert
  - Evidenz-Problem verstärkt, wenn Evidenzvariablen eher hinten in der topologischen Sortierung
    - Nichtevidenzvariablen ohne Evidenz in ihren Eltern und Vorfahren, um Erzeugen der Samples zu steuern

Zwischenzusammenfassung

- *Forward Sampling*: Sampeln aus einem BN ohne Evidenz, Abschätzen über Zählen
  - Entlang einer topologischen Sortierung, gegeben der schon gesampelten Elternwerte, neuen Wert pro Zufallsvariable sampeln
- *Rejection Sampling*: Sampeln aus einem BN mit Evidenz und dann beim Abschätzen verwerfen der Samples, die nicht zur Evidenz passen
  - Problem: Ineffizient wegen vielen verworfener Samples
- *Importance Sampling*
  - Aus Vorschlagsverteilung sampeln, falls man aus der Originalverteilung schwer sampeln kann, + Anpassung der Samples über Gewichtung
  - Ermöglicht sampeln aus ungerichteten Modellen: Quasi-BN z.B. über den Jtree erstellen
  - Spezialfall *Likelihood Weighting*: Sampeln aus einem BN mit Evidenz (Vorschlagsverteilung), so dass nur Samples passend zur Evidenz entstehen, gewichtet mit der Wahrscheinlichkeit, dass es zur Evidenz passt
  - Problem: viel Evidenz, Evidenz in den hinteren Teilen der topologischen Sortierung

Überblick: 4. Approximative Inferenz in episodischen PGMs

A. Überblick
- *PAC Theorie, deterministische vs. stochastische Approximation, Variational Inference*

B. Direktes Sampling
- Sampling in einer Wahrscheinlichkeitsverteilung mit und ohne Evidenz
- Forward Sampling (ohne Evidenz), Rejection Sampling (mit Evidenz) in BNs
- Likelihood Weighting in BNs, Importance Sampling als Verallgemeinerung
- Importance Sampling für Faktormodelle

C. Inferenz durch Markov-Ketten-Simulation
- Markov-Chain Monte-Carlo (MCMC) Sampling: Gibbs Sampling, Metropolis-Hastings Sampling

D. Sampling für die Datengenerierung
- Datensynthese

Markov Chain Monte Carlo (MCMC): Markov-Ketten-Simulation

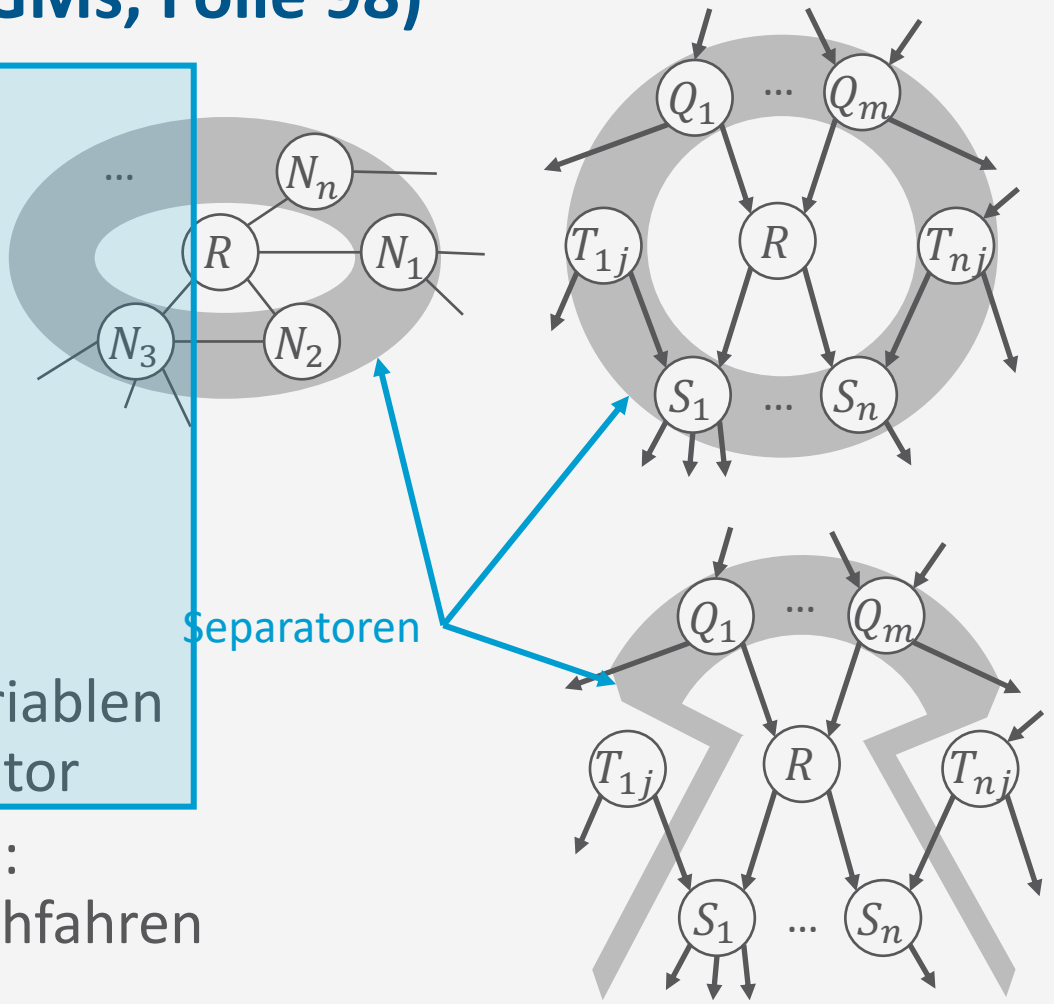
- Monte Carlo Methoden
  - Wiederholtes zufälliges Sampling um ein numerisches Ergebnis zu erhalten

- Stellen wir uns vor, dass Modell in einem bestimmten momentanen Zustand, welcher für jede Zufallsvariable einen Wert angibt (zusammengesetztes Ereignis für alle Zufallsvariablen)
- MCMC generiert ein nächstes Sample (zusammengesetztes Ereignis), indem es eine zufällige Änderung am vorherigen Sample (zusammengesetzten Ereignis) vornimmt
  - Nächster Zustand generiert durch zufälliges Sampeln eines Wertes für eine Nicht-Evidenzvariable R_i bedingt auf den momentanen Werten der Zufallsvariablen im Markov Blanket von R_i
  - Einfachste Form des MCMC Samplings: Gibbs Sampling

Markov Blanket (Foliensatz 2: Episodische PGMs, Folie 98)

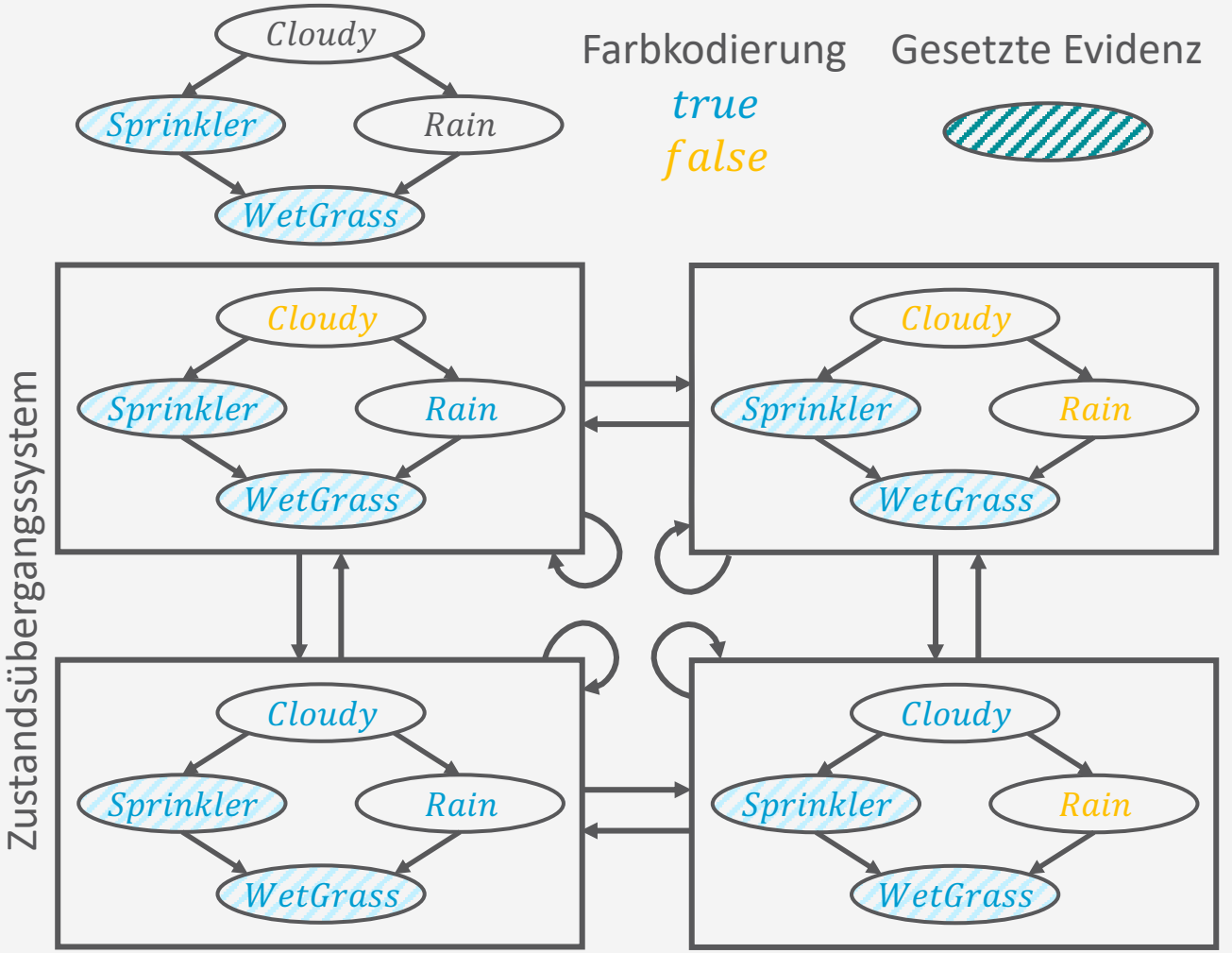
- Markov Blanket einer Zufallsvariable R
  - in einem Faktormodell:
    - Menge der direkten Nachbarn N_i von R im MN
  - in einem BN:
    - Vereinigung der Mengen
      - Eltern Q_k von R
      - Kinder S_i von R
      - Eltern T_{ij} der Kinder S_i, die nicht R sind

    } Nachbarn
- R bedingt unabhängig von allen anderen Zufallsvariablen im Modell gegeben sein Markov Blanket als Separator
- Vorherige Definition lokaler Unabhängigkeit in BNs: Gegeben die Eltern als Separatoren: Alle Nicht-Nachfahren



MCMC: Beispiel

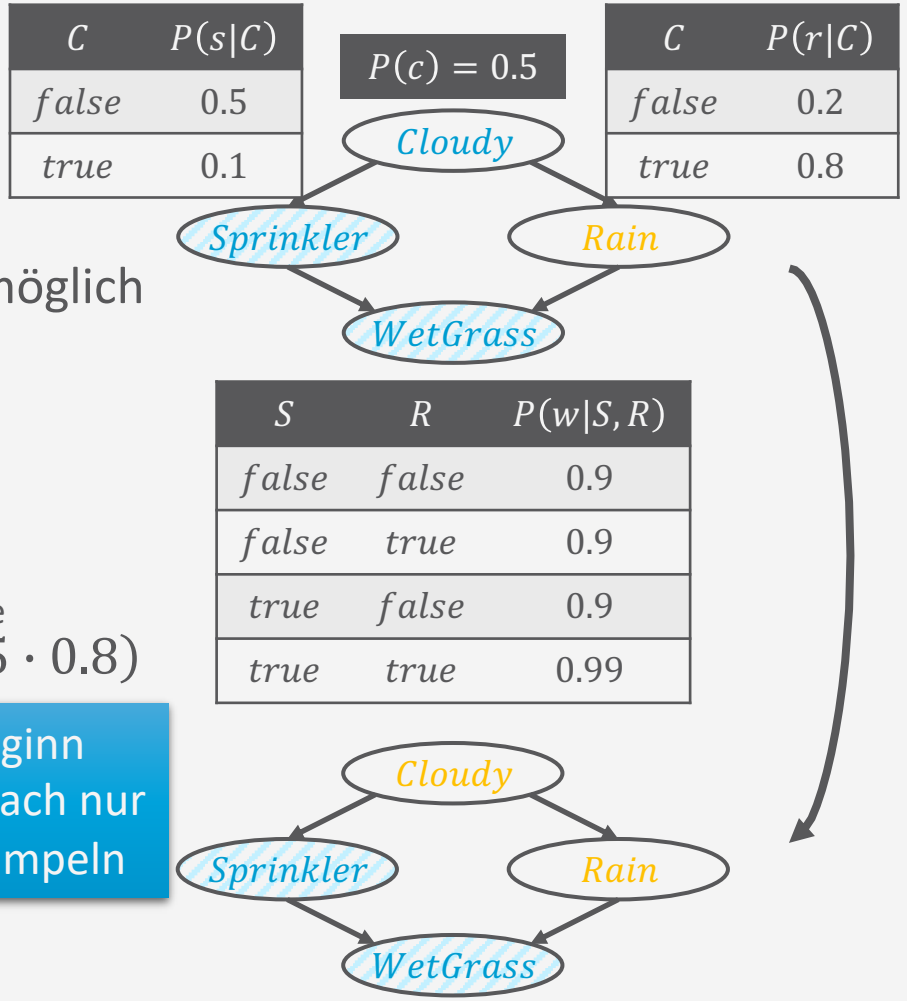
- Gegeben $S = true, W = true$, vier Zustände im Zustandsübergangssystem
  - Vier mögliche Kombinationen von Werten der verbleibenden Zufallsvariablen C, R
  - Kanten zwischen Zuständen beschreiben mögliche Übergänge
    - Kann man mit Wahrscheinlichten belegen (gemäß Wahrscheinlichkeiten aus BN)
    - Führt zu einer Markov-Kette von Zuständen
- Vorgehen:
  Eine Weile im System „umherwandern“ und mitteln, was man sieht



MCMC: Beispiel

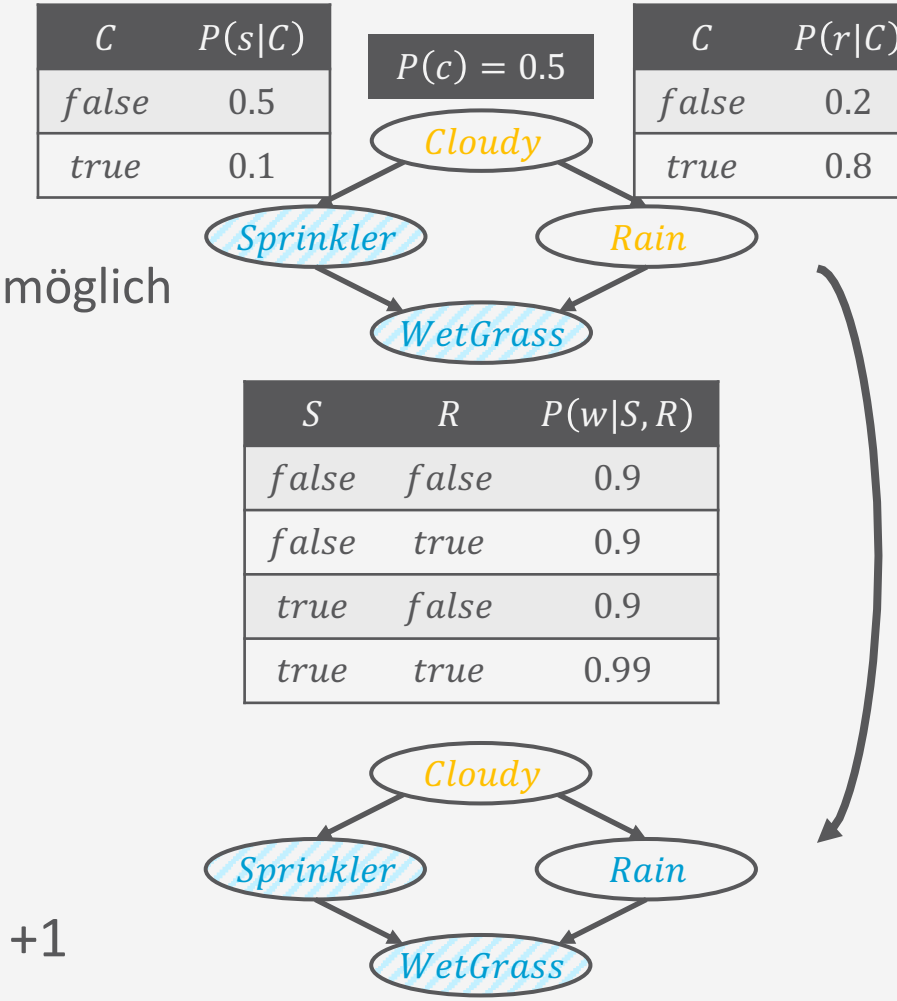
- $P(R|s,w)$?
  - Sampling Reihenfolge (C,R) ohne Evidenzvariablen
    - Randomisierte Auswahl der nächsten zu sampelnde Variable möglich
- Sample generieren (N mal):
  - Zufälliger initialer Zustand: $[c, s, \neg r, w]$
  - C sampeln gegeben den momentanen Werten von $MB(C) = \{S, R\}$, i.e., aus $P(C|s, \neg r)$
    - $P(C|s,\neg r) = P(C)P(\neg r|C)P(s|C) = (0.5 \cdot 0.1 \cdot 0.2, 0.5 \cdot 0.5 \cdot 0.8)$ $= (0.01,0.2) = (0.05,0.95)$ (true, false)
    - Annahme: Ergebnis ist $\neg c$
      - Neuer momentaner Zustand: $[\neg c, s, \neg r, w]$
      - Zähler aktualisieren: $R = false \rightarrow +1$

Einmal zu Beginn berechnen; danach nur noch daraus sampeln



MCMC: Beispiel

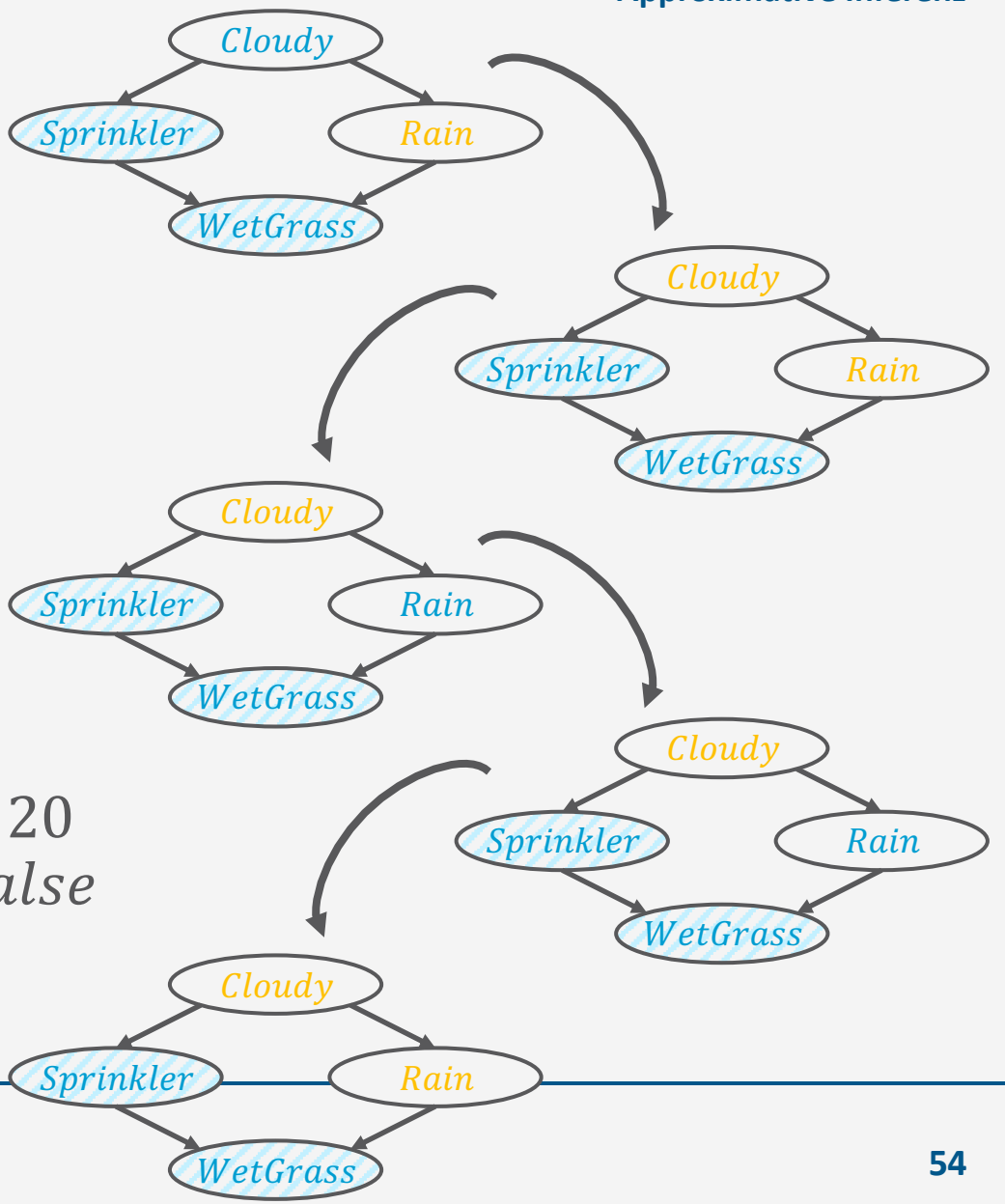
- $P(R|s,w)$?
  - Sampling Reihenfolge (C,R) ohne Evidenzvariablen
    - Randomisierte Auswahl der nächsten zu sampelnde Variable möglich
- Sample generieren (N mal)
  - Momentaner Zustand: $[\neg c, s, \neg r, w]$
  - R sampeln gegeben den momentanen Werten von $MB(R) = \{C,S,W\}$, i.e., aus $P(R|\neg c, s, w)$
    - Annahme: Ergebnis ist r
      - Neuer momentaner Zustand: $[\neg c, s, \underline{r}, w]$
      - Zähler aktualisieren: $R = true \rightarrow$ +1
  - C sampeln aus $P(C|s,r) \rightarrow [\underline{\neg c}, s, r, w]$, $R = true \rightarrow$ +1
  - R sampeln $P(R|\neg c, s, w) \rightarrow [\neg c, s, \underline{\neg r}, w]$, $R = false \rightarrow$ +1



MCMC: Beispiel

- $P(R|s,w)$?
  - Topologische Sortierung: (C,S,R,W) bzw. (C,R) ohne Evidenzvariablen
- Zufälliger initialer Zustand: $[c,s,\neg r,w]$
  - Folgezustand: $[\underline{\neg c},s,\neg r,w]$
  - Folgezustand: $[\neg c,s,\underline{\ \ r},w]$
  - Folgezustand: $[\underline{\neg c},s,\ \ r,w]$
  - Folgezustand: $[\neg c,s,\underline{\neg r},w]$
- Annahme: Nach $N = 80$ Iterationen hat der Prozess 20 Zustände mit $R = true$ und 60 Zustände mit $R = false$ besucht; Ergebnis der Anfrage:

$$\text{Normalise}((20,60)) = (0.25,0.75)$$



Gibbs Sampling

- Gegeben ein BN B, Evidenz $\boldsymbol{e}$ und Anfragevariable R
- Zustand des Netzwerks = momentane Zuweisung $\boldsymbol{r}$ an alle Zufallsvariablen $\boldsymbol{R} = \mathrm{rv}(B)$
  - Initial bestehend aus $\boldsymbol{e}$ für $\mathrm{rv}(\boldsymbol{e})$ und zufälligen Werten $\boldsymbol{r}'$ für alle Nichtevidenz-Variablen $\boldsymbol{R}' = \mathrm{rv}(B) \setminus \mathrm{rv}(\boldsymbol{e})$
- Erzeuge den nächsten Zustand durch Sampeln einer Nichtevidenz-Variable U gegeben seines Markov Blankets $\mathrm{MB}(U)$ mit den Werten aus dem momentanen Zustand
  - Sample Wert u für U aus $P\left(U \mid \pi_{\mathrm{MB}(U)}(\boldsymbol{r})\right)$
  - Ersetze den Wert von U in $\boldsymbol{r}$ mit u
  - Zähler inkrementieren für den Wert r von R, der in $\boldsymbol{r}$ vorkommt
  - Jede Variable der Reihe nach sampeln, während Evidenz gesetzt bleibt
    - Auch möglich eine Variable zum Sampeln jedes Mal zufällig zu wählen

Gibbs Sampling: Algorithm

Gibbs$(R, \boldsymbol{e}, B, N)$

Vektor N der Länge $|\mathrm{Val}(R)|$, anfangs 0 ▷ Speichert Zählerstände für alle $r \in \mathrm{Val}(R)$

Momentaner Zustand $\boldsymbol{r}$ bestehend aus $\boldsymbol{e}$ und zufälligen Werten $\boldsymbol{r}'$ für $\mathrm{rv}(B) \setminus \mathrm{rv}(\boldsymbol{e})$

for $i = 1 \ldots N$ **do**

  for $U \in \mathrm{rv}(B) \setminus \mathrm{rv}(\boldsymbol{e})$ **do**

    $u \leftarrow$ Sample Wert für U aus $P\left(U \mid \pi_{\mathrm{MB}(U)}(\boldsymbol{r})\right)$

    $\boldsymbol{r}[U] \leftarrow u$ ▷ Wert von U in $\boldsymbol{r}$ durch u ersetzen

    $\mathrm{N}[r] \leftarrow \mathrm{N}[r] + 1$ mit $r = \pi_{\mathrm{rv}(R)}(\boldsymbol{r})$

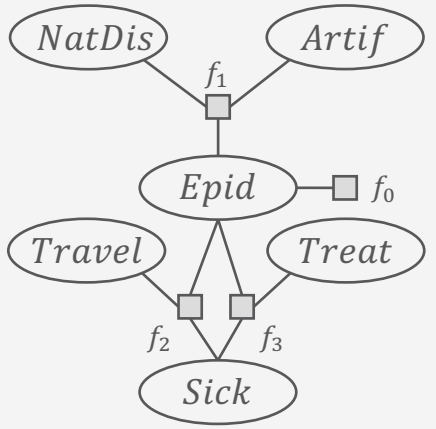
return $\mathrm{Normalise}(\mathrm{N})$

Gibbs Sampling

Gibbs Sampling und Faktormodelle

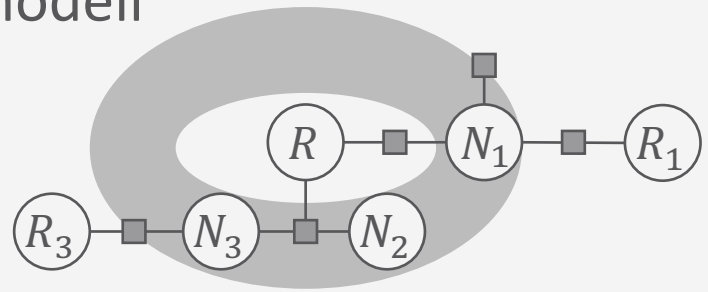
- Sampeln aus $P(U \mid \pi_{\mathrm{MB}(U)}(\boldsymbol{r}))$ in F für Anfragevariable R
  - Zuweisungen für jede Zufallsvariable in F vorhanden
- Da U gegeben $\mathrm{MB}(U)$ unabhängig von allen anderen Zufallsvariablen in F (lokale Unabhängigkeit) und Werte für $\mathrm{MB}(U)$ verfügbar: normalisiertes Produkt $P(U, \pi_{\mathrm{MB}(U)}(\boldsymbol{r}))$ der Faktoren F_U zwischen U und $\mathrm{MB}(U)$ mit Werten $\pi_{\mathrm{MB}(U)}(\boldsymbol{r})$ betrachten
  - $F_U = \{f \mid f \in F, U \in \mathrm{rv}(f)\}$, $P\big(U, \mathrm{MB}(U)\big) = \frac{1}{Z} \prod_{f \in F_U} f$
- Beispiel: Zustand $[e, \neg n, \neg a, s, \neg tl, tt]$, Anfragevariable Tl
  - Sample neuen Wert für z.B. Tl aus $P(Tl \mid \pi_{\mathrm{MB}(Tl)}(\boldsymbol{r})) = P(Tl \mid e, s)$
  - $F_{Tl} = \{\phi_2(Tl, e, s)\}$: ϕ_2 normalisieren → $P(Tl\,, e, s) = (0.25, 0.75)$
  - Wert für Tl sampeln: tl, neuer Zustand $[e, \neg n, \neg a, s, tl, tt]$, $Tl = true$: +1

$Travel$	$Epid$	$Sick$	ϕ_2	P
$false$	$false$	$false$	20	
$false$	$false$	$true$	24	
$false$	$true$	$false$	5	
$false$	$true$	$true$	6	0.75
$true$	$false$	$false$	28	
$true$	$false$	$true$	8	
$true$	$true$	$false$	7	
$true$	$true$	$true$	2	0.25



Beispiel mit zwei Faktoren

- Faktormodell



- Sample aus
  $P(R \mid \pi_{\mathrm{MB}(R)}(\boldsymbol{r})) = P(R \mid n_1, n_2, n_3)$
- $\phi(R, n_1) \cdot \phi(R, n_2, n_3)$ normalisieren
  $\rightarrow P(R, n_1, n_2, n_3)$
- Bei $N_1 = 1, N_2 = 1, N_3 = 0$
  - $f_{12} = \phi(R, 1) \cdot \phi(R, 1, 0) = \phi(R)$
  - Aus $f'_{12} = \frac{1}{Z} f_{12}$ neuen R Wert sampeln

R	N_1	f_1
~~0~~	~~0~~	~~1~~
0	1	2
~~1~~	~~0~~	~~3~~
1	1	4

R	N_2	N_3	f_2
~~0~~	~~0~~	~~0~~	~~3~~
~~0~~	~~0~~	~~1~~	~~2~~
0	1	0	4
~~0~~	~~1~~	~~1~~	~~1~~
~~1~~	~~0~~	~~0~~	~~5~~
~~1~~	~~0~~	~~1~~	~~6~~
1	1	0	8
~~1~~	~~1~~	~~1~~	~~1~~

(MB Werte absorbiert)

R	f_1
0	2
1	4

R	f_2
0	4
1	8

Produkt

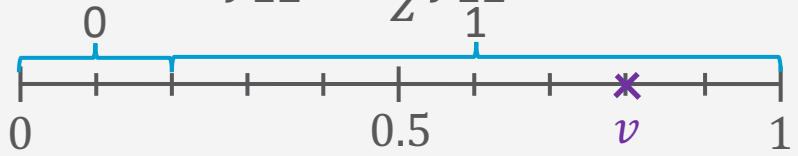
R	f_{12}
0	8
1	32

Norm.

R	f'_{12}
0	0.2
1	0.8

Beispiel mit zwei Faktoren

- Sample aus $f'_{12} = \frac{1}{Z} f_{12}$ neuen R Wert:



  - Neuer Zustand: $N_1 = 1, N_2 = 1, N_3 = 0$, $R_3 = r_3, R_1 = r_1, R = 1$
- Angenommen R_3 ist Anfragevariable
  - Inkrementiere den Zähler für r_3
- Nächste Variable sampeln
  - Z.B. N_1 mit $\mathrm{MB}(N_1) = \{R = 1, R_1 = r_1\}$
    - $\phi(R = 1, N_1), \phi(N_1), \phi(N_1, R_1 = r_1)$
      - Multiplizieren und normalisieren: ein ϕ'
    - Neuen Wert für N_1 aus ϕ' sampeln

R	N_1	f_1
~~0~~	~~0~~	~~1~~
0	1	2
~~1~~	~~0~~	~~3~~
1	1	4

R	N_2	N_3	f_2
~~0~~	~~0~~	~~0~~	~~3~~
~~0~~	~~0~~	~~1~~	~~2~~
0	1	0	4
~~0~~	~~1~~	~~1~~	~~1~~
~~1~~	~~0~~	~~0~~	~~5~~
~~1~~	~~0~~	~~1~~	~~6~~
1	1	0	8
~~1~~	~~1~~	~~1~~	~~1~~

(MB Werte absorbiert)

R	f_1
0	2
1	4

R	f_2
0	4
1	8

Produkt

R	f_{12}
0	8
1	32

Norm.

R	f'_{12}
0	0.2
1	0.8

Ein paar Grundlagen für MCMC

- Eine Markov-Kette besteht aus n *Zuständen* und einer $n \times n$ *Transitionsmatrix* $\mathfrak{T}$
  - In jedem Schritt, sind wir in genau einem Zustand
  - Matrixeintrag $\mathfrak{T}_{ij}$, $1 \leq i, j \leq n$, gibt die relative Häufigkeit an, dass j der nächste Zustand ist gegeben, dass wir momentan in Zustand i sind
    - $\mathfrak{T}_{ij} \triangleq \mathfrak{T}(i \rightarrow j)$

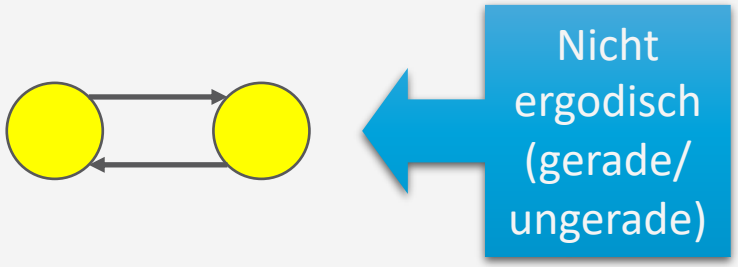
    - Wahrscheinlichkeitsverteilung, d.h.:

$$\sum_{j=1}^{n} \mathfrak{T}_{ij} = 1$$

$\mathcal{T}_{ii} > 0$ möglich (Schleife)

Ein paar Grundlagen für MCMC

- Markov-Kette muss ergodisch sein, damit MCMC funktioniert
- Markov-Kette ist ergodisch, wenn
  - es einen Pfad von jedem Zustand zu jedem anderen Zustand gibt (Irreduzibilität, *irreducibility*)
    - Kein Teil des Systems zieht davon
  - sie zu Zuständen in unregelmäßigen Abständen zurückkehrt (Aperiodizität, *aperiodicity*)
    - Periodizität: Zurückkehr zu einem Zustand nur alle $c > 1$ Schritte möglich



  - für jeden Startzustand gilt, dass nach einer endlich Anlaufzeit T_0 die Wahrscheinlichkeit, in einem Zustand zu einer gegebenen Zeit $T > T_0$ zu sein, nicht null ist (positive Rekurrenz, *positive recurrence*)
    - Gegeben einem endlichen Zustandsraum: Positive Rekurrenz folgt aus der Irreduzibilität

Ein paar Grundlagen für MCMC

- Ergodentheorie: über dynamische Systeme, die ergodisch sind
  - System muss maßerhaltend sein
    - Maß einer Menge: Jeder passenden Untermenge eine Zahl zuordnen
    - Axiome der Wahrscheinlichkeitstheorie (*Kolmogorov Axiome*) korrespondieren zu den Axiomen der Maßtheorie
      - Einige ergodische Theoreme im probabilistischen Rahmen einsetzbar
- Ein paar Unterschiede
  - In Ergodentheorie
    - Irreduzierbar + positiv recurrent = ergodisch
    - irreduzierbar + positiv recurrent + aperiodisch = mischend
  - Während in der Wahrscheinlichkeitstheorie
    - irreduzierbar + positiv recurrent + aperiodisch = ergodisch

Kolmogorov Axiome

1. Wahrscheinlichkeit eines Ereignisses ist eine nicht-negative reelle Zahl
2. Wahrscheinlichkeiten addieren sich zu 1 auf
3. Wahrscheinlichkeit einer Menge von disjunkten Ereignissen ist gleich der Summe über die Einzelwahrscheinlichkeiten (Unabhängigkeit; σ-Additivität)

Ein paar Grundlagen für MCMC

- Für eine ergodische Markov-Kette mit einer endlichen Menge an Zuständen gibt eine eindeutige long-term visit rate für jeden Zustand
  - Auch Steady-State- oder *stationäre* Verteilung genannt
    - Stationarität: Übergangswahrscheinlichkeiten zwischen Zuständen ändern sich nicht im Verlauf der Zeit
  - Über einen langen Zeitraum wird jeder Zustand proportional zu dieser Rate besucht
    - Unabhängig davon, wo die Kette startet

- Grund, warum Sampling funktioniert, wenn N (Sampling Größe) groß genug ist

Ein paar Grundlagen für MCMC

- Für eine ergodische Markov-Kette mit einer endlichen Menge an Zuständen gibt eine eindeutige long-term visit rate für jeden Zustand
  - Bekannte Anwendung: *PageRank*, ursprüngliches Ranking-Prinzip von Google
    - Ordne eine Menge von relevanten Webseiten für eine Anfrage gemäß der Wahrscheinlichkeiten, die sie in der Steady-State-Verteilung haben (Ranking ist unabhängig von der Anfrage)
    - Markov-Kette:
      - Webseiten = Zustände (i.e., auf einer bestimmten Webseite sein und damit nicht auf anderen)
      - Pfeile von einem Zustand zum nächsten (einer Webseite zur nächsten), wenn es ausgehende Links von der Webseite zur nächsten gibt
      - Übergangsmodell $\mathfrak{T}$: für jeden Zustand, gleich verteilt über alle ausgehenden Links
    - Berechne Steady-State-Verteilung λ (als Vektor): λ muss $\lambda^T \mathfrak{T} = \lambda^T$ erfüllen
      - Passender Eigenvektor zum Eigenwert 1

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd: The PageRank Citation Ranking: Bringing Order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

Stationäre Verteilung formal

- Eine Markov-Kette ist regulär, wenn es eine Zahl k gibt, so dass für jedes Paar $\boldsymbol{r}, \boldsymbol{r}' \in \mathrm{Val}(\boldsymbol{R})$ die Wahrscheinlichkeit von $\boldsymbol{r}$ nach $\boldsymbol{r}'$ in genau k Schritten zu kommen > 0 ist
  - Für endliche Zustandsräume: Bedingung für Regularität äquivalent zur Bedingung für Ergodizität
    - Manchmal einfacher zu verifizieren
  - In Faktormmodelle:
    Wenn alle Potentiale strikt positiv sind, dann ist die Gibbs Sampling Markov-Kette regulär

Stationäre Verteilung formal

- Eine Markov-Kette mit Übergangsmodell $\mathfrak{T}$ ist reversibel, wenn es eine eindeutige Verteilung λ gibt, so dass für alle $\boldsymbol{r}, \boldsymbol{r}' \in \mathrm{Val}(\boldsymbol{R})$ gilt:

$$\lambda(\boldsymbol{r})\mathcal{T}(\boldsymbol{r} \to \boldsymbol{r}') = \lambda(\boldsymbol{r}')\mathcal{T}(\boldsymbol{r}' \to \boldsymbol{r})$$

  - Gleichung wird detailliertes Gleichgewicht (*detailed balance*) genannt
    - Wähle einen Startzustand zufällig gemäß λ
    - Nehme einen zufälligen Übergang aus dem gewählten Zustand gemäß $\mathfrak{T}$
  - Stellt sicher, dass mittels dieses Prozesses die Wahrscheinlichkeit eines Übergangs $\boldsymbol{r} \to \boldsymbol{r}'$ gleich der Wahrscheinlichkeit eines Übergangs $\boldsymbol{r}' \to \boldsymbol{r}$ ist

Wenn $\mathcal{T}$ regulär ist und die detailed balance Gleichung relativ zu λ erfüllt, dann ist λ die eindeutige stationäre Verteilung von $\mathcal{T}$.

Parallelisierung

- Gibbs Sampling auf einer Menge von Kopien des Modells unabhängig von einander laufen lassen
  - Mehr Samples in der gleichen Zeit

  oder
  - Gleiche Menge an Samples in weniger Zeit
- Individuelle Zähler zu einem aggregieren

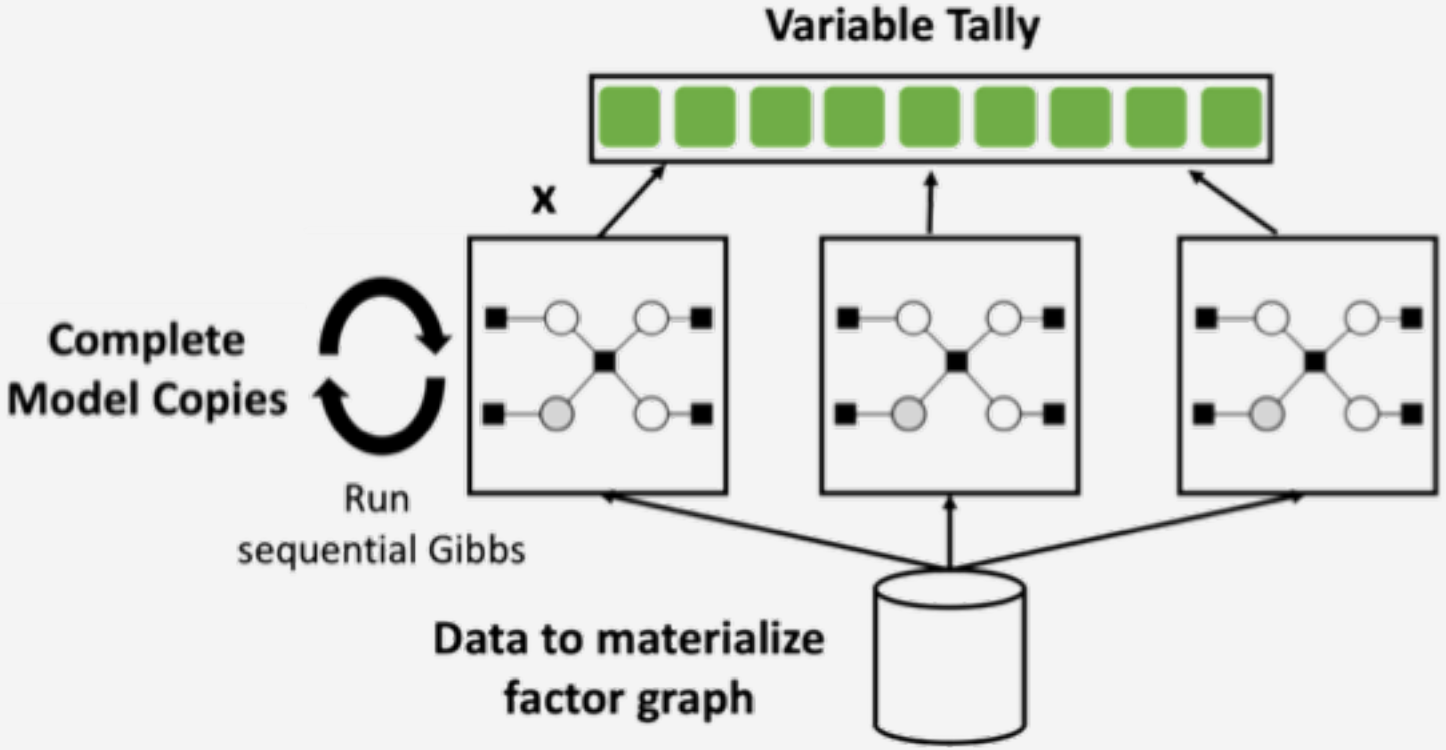
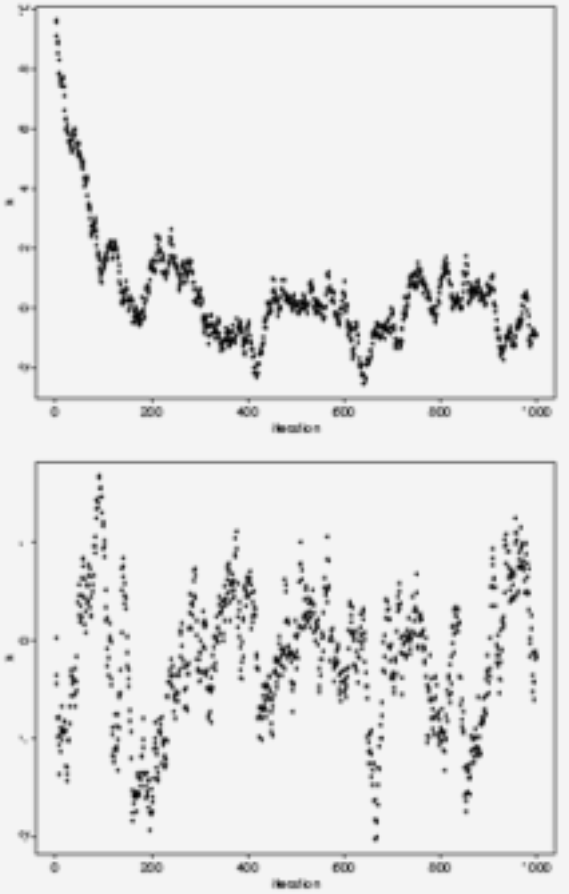


Abbildung: Theo Rekatsinas, https://thodrek.github.io/CS839_fall18/lectures/lecture_14/Lecture_14.pdf

Burn-in & Thinning

- Kontroverse Techniken, um jeweils ein Problem zu lösen
- Problem 1: Samples beginnen an einem zufälligen Zustand, der sehr unwahrscheinlich sein kann, was die Verteilung verzerren kann
  - *Burn-in*/warm-up: die ersten $N' < N$ Samples verwerfen
  - Alternativen
    - In einem sehr wahrscheinlichen Zustand starten, wenn bekannt
    - In einem Zustand starten, in dem ein vorheriger Lauf geendet hat
- Problem 2: Da der nächste Zustand vom vorherigen Zustand abhängt, sind die Samples nicht mehr unabhängig (Autokorrelation)
  - *Thinning*/subsampling: nur jedes k'te Sample nehmen
    - Löst nicht wirklilch das Problem, da auch die immer noch abhängig von einander sind



Eine Menge von Zufallsvariablen, die einer Normalverteilung mit Mittelwert 0 folgen; beginnend bei $x = 10$ and $x = 0$

Abbildungen: Charles Geyer: Burn-in Is Unnecessary. http://users.stat.umn.edu/~geyer/mcmc/burn.html
William A. Link and Mitchell J. Eaton: On Thinning of Chains in MCMC. In Methods in Ecology and Evolution, 2011. https://besjournals.onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2011.00131.x

Weitere Probleme mit Gibbs Sampling

- Nur sehr lokale Schritte im Zustandsraum
  - Änderung einer Zufallsvariable pro Schritt
- In Modellen mit eng korrelierten Zufallsvariablen können solche Schritte von sehr wahrscheinlichen Zuständen zu welchen mit sehr niedriger Wahrscheinlichkeit führen
  - Mit hoher Wahrscheinlichkeit einfach wieder in den hoch-wahrscheinlichen Zustand zurückzukehren
  - Unwahrscheinlich, dass Kette von so einem Zustand wegkommt
    - Kette mischt sich nur langsam
- Ketten nutzen, welche eine größere Menge von Schritten inklusive weiteren Schritten (über mehrere Zustände hinweg) ermöglicht
  - Passend bauen, dass die Markov-Kette trotzdem die gleiche / gewünschte stationäre Verteilung hat

Metropolis-Hastings Algorithmus (MH)

- Baut eine reversible Markov-Kette mit einer gewünschten stationären Verteilung λ
  - Unter der Annahme, dass man den nächsten Zustand, i.e., das nächste Sample, aus einer gewollten Zielverteilung nicht generieren kann
  - Nutzt deshalb eine Vorschlagsvorteilung
  - Vergleich: Importance Sampling mit einer Vorschlagsverteilung
    - Zielverteilung: Sampling Verteilung bzgl. des nächsten Zustand an einem gewünschten Zustand
    - Sample aus einer Vorschlagsverteilung und korrigiere für den Fehler
    - Aber: keine Gewichte
      - Verfallen mit der Anzahl an Übergängen
  - Stattdessen: wählt, ob man den vorgeschlagenen Übergang nimmt, mit einer Wahrscheinlichkeit, welche den Unterschied zwischen der Vorschlags- und Zielverteilung korrigiert

W.K. Hastings: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. In *Biometrika*, 1970.

Vorschlagsverteilung in MH

- Vorschlagsverteilung $\mathfrak{T}^Q$ definiert ein Übergangsmodell über den Zustandsraum $\mathrm{Val}(\boldsymbol{R})$
  - Für jeden Zustand $\boldsymbol{r}$ definiert $\mathfrak{T}^Q$ eine Verteilung über mögliche Nachfolge-Zustände in $\mathrm{Val}(\boldsymbol{R})$, aus der man zufällig einen nächsten Zustand $\boldsymbol{r}'$ als Kandidat wählt
    - Entweder Vorschlag annehmen und nach $\boldsymbol{r}'$ übergehen
    - Oder Vorschlag ablehnen und in $\boldsymbol{r}$ bleiben
  - Für jedes Paar von Zuständen $\boldsymbol{r}, \boldsymbol{r}'$ gibt es eine Annahme-Wahrscheinlichkeit (*acceptance probability*) $\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}')$
  - Tatsächliche Übergangsmodell der Markov-Kette:

$$\mathfrak{T}(\boldsymbol{r} \to \boldsymbol{r}') = \begin{cases} \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}') & \boldsymbol{r} \neq \boldsymbol{r}' \\ \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}) + \sum_{\boldsymbol{r}' \neq \boldsymbol{r}} \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')\big(1 - \mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}')\big) & \text{sonst} \end{cases}$$

  - Vorschlagsverteilung frei wählbar, solange die Verteilung eine reguläre Kette induziert

Annahme-Wahrscheinlichkeit

- Gegeben Vorschlagsverteilung $\mathfrak{T}^Q$, wähle Annahme-Wahrscheinlichkeiten $\mathfrak{A}$ so, dass die gewünschte stationäre Verteilung λ herauskommt
  - Detailliertes Gleichgewicht muss gelten:

$$\lambda(\boldsymbol{r})\mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}') = \lambda(\boldsymbol{r}')\mathfrak{T}^Q(\boldsymbol{r}' \to \boldsymbol{r})\mathfrak{A}(\boldsymbol{r}' \to \boldsymbol{r})$$

  - Setze $\mathfrak{A}$ auf

$$\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}') = \min\left[1, \frac{\lambda(\boldsymbol{r}')\mathfrak{T}^Q(\boldsymbol{r}' \to \boldsymbol{r})}{\lambda(\boldsymbol{r})\mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')}\right]$$

Mit $\mathfrak{T}^Q$ eine Vorschlagsverteilung, betrachte die Markov-Kette $\mathfrak{T}$ definiert durch

$$\mathfrak{T}(\boldsymbol{r} \to \boldsymbol{r}') = \begin{cases} \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}') & \boldsymbol{r} \neq \boldsymbol{r}' \\ \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}) + \sum_{\boldsymbol{r}' \neq \boldsymbol{r}} \mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')\left(1 - \mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}')\right) & \text{sonst} \end{cases}$$

mit

$$\mathfrak{A}(\boldsymbol{r} \to \boldsymbol{r}') = \min\left[1, \frac{\lambda(\boldsymbol{r}')\mathfrak{T}^Q(\boldsymbol{r}' \to \boldsymbol{r})}{\lambda(\boldsymbol{r})\mathfrak{T}^Q(\boldsymbol{r} \to \boldsymbol{r}')}\right].$$

Wenn $\mathfrak{T}$ regulär ist, dann hat sie die stationäre Verteilung λ.

MH: Algorithmus

- Folgt der gleichen Vorgehen wie Gibbs Sampling außer, dass
  - der nächste Zustand $\boldsymbol{r}_i$ aus der Vorschlagsverteilung $\mathfrak{T}^Q$ anstatt der Zielverteilung $\mathfrak{T}$ generiert wird und
  - $\boldsymbol{r}_i$ basierend auf der Annahme-Wahrscheinlichkeit $\mathfrak{A}$ angenommen oder verworfen wird

Gibbs$(R, \boldsymbol{e}, B, N)$
  Vektor N der Länge $|\text{Val}(R)|$, anfangs 0 ▸ Speichert Zählerstände für alle $r \in \text{Val}(R)$
  Momentaner Zustand $\boldsymbol{r}$ bestehend aus $\boldsymbol{e}$ und zufälligen Werten $\boldsymbol{r}'$ für $\text{rv}(B) \setminus \text{rv}(\boldsymbol{e})$
  for $i = 1 \ldots N$ **do**
    for $U \in \text{rv}(B) \setminus \text{rv}(\boldsymbol{e})$ **do**
      $u \leftarrow$ Sample Wert für U aus $P\left(U \mid \pi_{\text{MB}(U)}(\boldsymbol{r})\right)$
      $\boldsymbol{r}[U] \leftarrow u$ ▸ Wert von U in $\boldsymbol{r}$ durch u ersetzen
      $\text{N}[r] \leftarrow \text{N}[r] + 1$ mit $r = \pi_{\text{rv}(R)}(\boldsymbol{r})$
  return Normalise(N)

Gibbs Sampling

Zwischenzusammenfassung

- MCMC Methoden: Baue eine reguläre Markov-Kette mit stationärer Verteilung, die der Anfrageverteilung entspricht
  - Sample nächste Zuständen der Kette
  - Wie beim direkten Sampling, zähle die Vorkommen
- *Gibbs Sampling*
  - Baue Zustandsübergangssystem aus den möglichen Zuständen der Nicht-Evidenzvariablen
  - Sample den nächsten Zustand durch Sampeln eines neuen Wertes für eine Zufallsvariable gegeben ihr Markov Blanket
- *Metropolis-Hastings Sampling*
  - Nutze eine Vorschlagsverteilung zum Sampeln, wenn aus der Zielverteilung nicht gesampelt werden kann
  - Korrigiere den Unterschied zwischen der Verteilungen über eine Annahme-Wahrscheinlichkeit

Überblick: 4. Approximative Inferenz in episodischen PGMs

A. Überblick
- *PAC Theorie, deterministische vs. stochastische Approximation, Variational Inference*

B. Direktes Sampling
- Sampling in einer Wahrscheinlichkeitsverteilung mit und ohne Evidenz
- Forward Sampling (ohne Evidenz), Rejection Sampling (mit Evidenz) in BNs
- Likelihood Weighting in BNs, Importance Sampling als Verallgemeinerung
- Importance Sampling für Faktormodelle

C. Inferenz durch Markov-Ketten-Simulation
- Markov-Chain Monte-Carlo (MCMC) Sampling: Gibbs Sampling, Metropolis-Hastings Sampling

D. Sampling für die Datengenerierung
- Datensynthese

Datensynthese

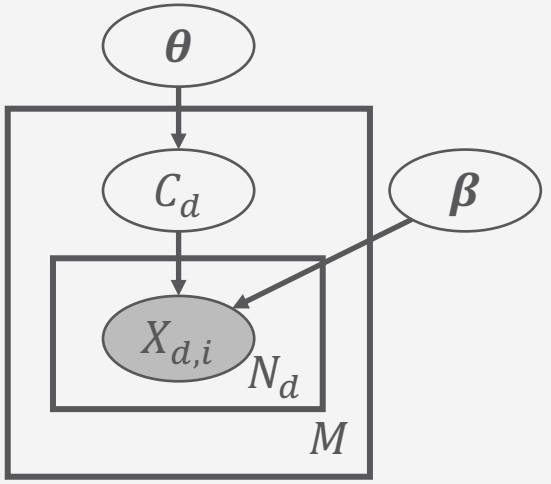
- Gegeben ein PGM
- Generiere neue Daten mittels Sampling

- Nutzen
  - Neuer Input für Lernalgorithmen
    - Iteratives Lernen
  - Veröffentlichung von generierten anstatt originalen Daten
    - Ursprüngliche sensitive Daten durch generierte Daten ersetzen

- Beispiel: Topic-Modellierung → Generierung von Dokumenten
  - Erinnerung: Bag-of-Words Annahme (keine grammatikalisch richtigen Sätze erzeugbar)

Mixture of Unigrams

- Jedes Dokument hat genau ein Topic k
  - M Dokumente, N_d Worte im Dokument d
  - K mögliche Topics
  - Multinomial-Parameter $\boldsymbol{\theta} \in [0,1]^K$ für die A-priori-Verteilung $P(C)$ über $k \in \text{Val}(C)$: $C \sim Mul(\boldsymbol{\theta})$
  - Multinomial-Parameter $\boldsymbol{\beta} \in [0,1]^N$ für die CPD $P(X|C)$ über das Lexikon $\mathcal{D} = \text{Val}(X)$ für jedes $k \in \text{Val}(C)$: $X \sim Mul(\boldsymbol{\beta}[k])$
- Sample ein neues Dokument
  - Sample ein Topic aus $P(C), \text{Val}(C) = \{1,2,3,4\}$
  - Wähle N (Länge des Dokuments; z.B. sample aus Poisson-Vert.)
  - Sample N mal aus $P(X|C)$
    - Worte mit hoher Wahrscheinlichkeit pro Topic rechts abgebildet

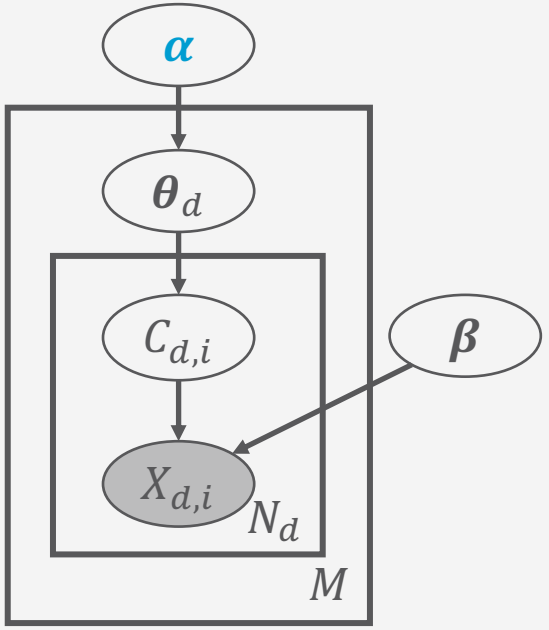


1 (Arts)	2 (Budgets)	3 (Children)	4 (Education)
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Mixture of Topics

- Dokumente bestehen aus einem Mix von Topics
  - M Dokumente, N_d Worte im Dokument d
  - K mögliche Topics, pro Dokument d Topic-Verteilung $\boldsymbol{\theta}_d$
- Sample ein neues Dokument
  - Sample eine Topic-*Verteilung* $\boldsymbol{\theta}_d$ über K Topics aus $\boldsymbol{\alpha}$
  - Wähle N (Länge des Dokuments; z.B. sample aus Poisson-Vert.)
  - Wiederhole N mal
    - Sample Topic $c_{d,i}$ aus $P(C_{d,i}) = \boldsymbol{\theta}_d$
    - Sample Wort $x_{d,i}$ aus $P(X_{d,i}|c_{d,i}) = \boldsymbol{\beta}[c_{d,i}]$
      - Häufige Schreibweise in der Forschungsgemeinde: $c_{d,i} \sim Mul(\cdot \mid \boldsymbol{\theta}_d)$ und $x_{d,i} \sim Mul(\cdot \mid \boldsymbol{\beta}_c)$



$$\boldsymbol{\theta}_d = (0.32, 0.34, 0.25, 0.09)$$
$$N = 68$$

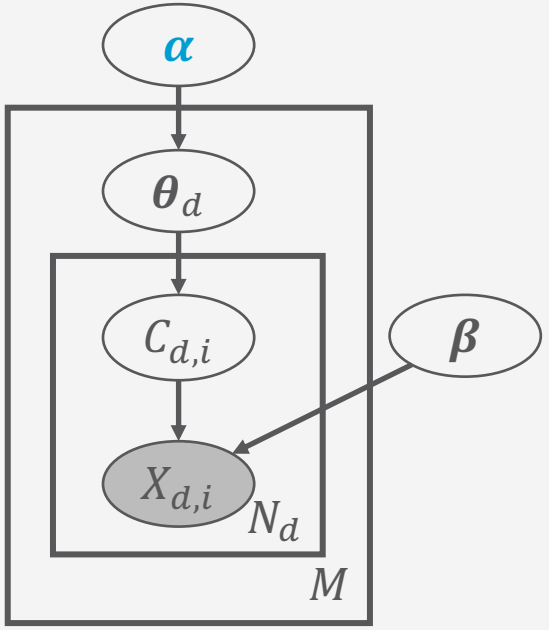
David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Mixture of Topics

1 (Arts)	2 (Budgets)	3 (Children)	4 (Education)
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

$$\boldsymbol{\theta}_d = (0.32, 0.34, 0.25, 0.09)$$
$$N = 68$$

William Randolph Hearst Foundation $1.25 million Lincoln Center Opera New York Philharmonic School board real opportunity make mark future performing grants act bit important traditional support research education social services Hearst Foundation President Randolph Hearst Monday announcing Lincoln share $200.000 new building house young provide new public facilities Opera New York Philharmonic receive $400.000 School music performing taught $250.000 Hearst Foundation leading supporter Lincoln Center Fund make annual $100.000



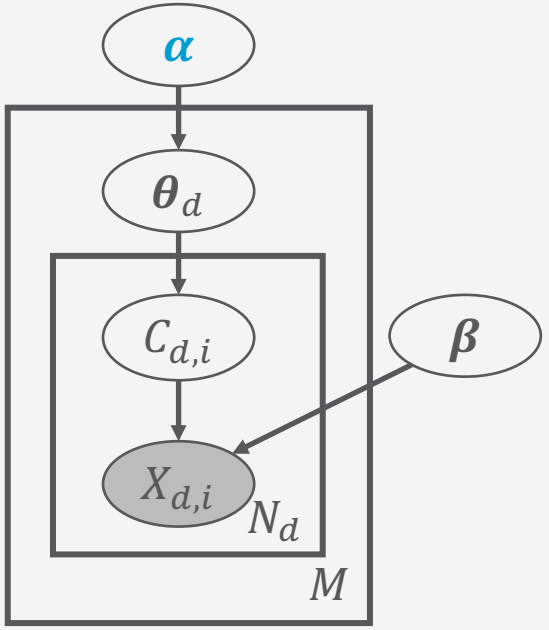
David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Mixture of Topics: Hyperparameter α

- Sampling einer Topic-*Verteilung* $\boldsymbol{\theta}_d$ über K Topics aus $\boldsymbol{\alpha}$
  - Erlaubt Vorwissen zu kodieren, ob Dokumente eher aus wenigen Topics mit hoher Wahrscheinlichkeit oder aus vielen Topics mit ähnlicher Wahrscheinlichkeit bestehen
  - Genutzte Verteilung zur Kodierung: Dirichlet-Verteilung

$$p(\theta|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

  - Im Allgemeinen $\boldsymbol{\alpha}$ ein Vektor aus K nicht-negative Einträgen
    - In der Regel für Topics alle gleich, also ein α ausreichend:
      Hohe Werte: eher Gleichverteilung zwischen den Topics
      Niedriger Werte: eher weniger Einträge mit hohen Wahrscheinlichkeiten
    - Unterschiedliche Einträge sagen aus, dass einige Topics eher allein Vorkommen, andere eher zusammen

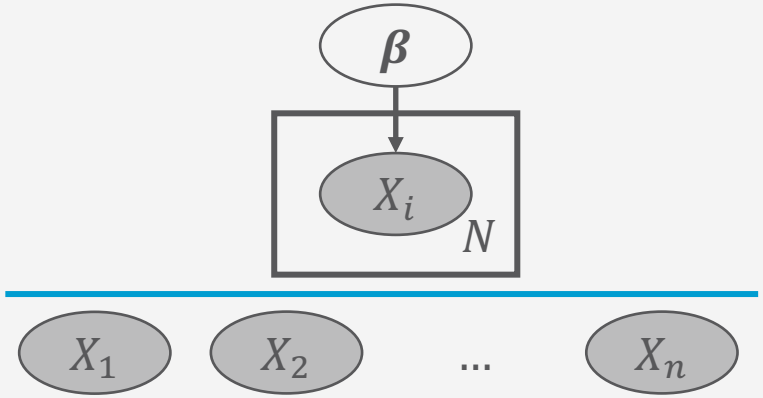


$\alpha = 1$:
$\boldsymbol{\theta}_d = (0.32, 0.34, 0.25, 0.09)$
$\alpha = 10$:
$\boldsymbol{\theta}_d = (0.27, 0.23, 0.25, 0.25)$
$\alpha = 0.1$:
$\boldsymbol{\theta}_d = (0.68, 0.08, 0.10, 0.14)$

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Der Vollständigkeit halber: Dokumentenmodellierung ohne Topics

- Unigram: A-priori-Verteilung über Worte in Lexikon
  - X_i gibt an, ob ein Wort w_i aus Lexikon $\mathcal{D}$ vorkommt
  - Nur Bag-of-Words Annahme; *keine* Topics vorgesehen



fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the

Automatisch generierte Sätze aus einem Unigramm-Modell ohne Vorverarbeitung

Mittels direktem Sampling aus gegebenen Verteilungen ($\boldsymbol{\beta}$) gesampelt

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: *Latent Dirichlet Allocation*. Journal of Machine Learning Research, 2003.

Überblick: 4. Approximative Inferenz in episodischen PGMs

A. *Überblick*
   - *PAC Theorie, deterministische vs. stochastische Approximation, Variational Inference*

B. *Direktes Sampling*
   - Sampling in einer Wahrscheinlichkeitsverteilung mit und ohne Evidenz
   - Forward Sampling (ohne Evidenz), Rejection Sampling (mit Evidenz) in BNs
   - Likelihood Weighting in BNs, Importance Sampling als Verallgemeinerung
   - Importance Sampling für Faktormodelle

C. *Inferenz durch Markov-Ketten-Simulation*
   - Markov-Chain Monte-Carlo (MCMC) Sampling: Gibbs Sampling, Metropolis-Hastings Sampling

D. *Sampling für die Datengenerierung*
   - Datensynthese

→ Lernalgorithmen für episodische PGMs

Einordnung der Vorlesung: *Modell- und nutzenbasierter Agent*

- Nachfolgende Themen der Vorlesung
  2. Episodische PGMs
  3. Exakte Inferenz in episodischen PGMs
  4. Approximative Inferenz in episodischen PGMs
  5. Lernalgorithmen für episodische PGMs
  6. Sequentielle PGMs und Inferenz
  7. Entscheidungstheoretische PGMs