Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Graphische Modelle

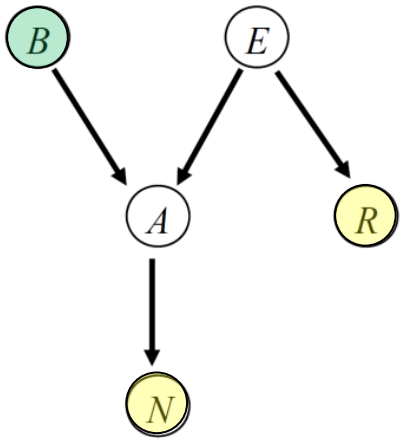
Christoph Sawade/Niels Landwehr/Tobias Scheffer

Überblick

- Gerichtete Graphische Modelle: Bayessche Netze
- Graphische Modelle im Maschinellen Lernen
- Inferenz in Graphischen Modellen
- Ungerichtete Graphische Modelle: Markov Netze

Graphische Modelle: Inferenz

- Beispiel „Alarm“ Domäne
  - Variablen mit Evidenz: N, R
  - Anfrage-Variablen: B



Wahrscheinlichkeit für Einbruch gegeben dass der Nachbar uns angerufen hat?

Zum Beispiel:

$$p(B=1 \mid N=1, R=0) = 0.6$$
$$p(B=0 \mid N=1, R=0) = 0.4$$

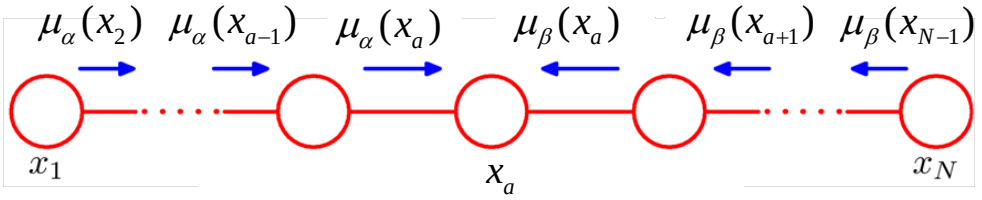
$$p(B=1 \mid N=1, R=1) = 0.2$$
$$p(B=0 \mid N=1, R=1) = 0.8$$

- Posterior über Parameter, Bayessche Vorhersage, …

Exakte Inferenz: Message-Passing

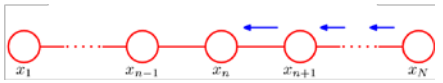
- Message Passing Algorithmus auf linearer Kette

$$p(\mathbf{x}) = \prod_{i=1}^{N} \psi_{i,i+1}(x_i, x_{i+1})$$



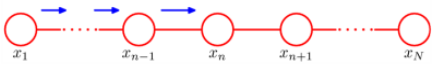
$$\mu_\beta(x_N) = \mathbf{1}$$

Für $k = N\text{-}1, \ldots, a$: $\mu_\beta(x_k) = \sum_{x_{k+1}} \psi_{k,k+1}(x_k, x_{k+1})\mu_\beta(x_{k+1})$



$$\mu_\alpha(x_1) = \mathbf{1}$$

Für $k = 2, \ldots, a$: $\mu_\alpha(x_k) = \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1}, x_k)\mu_\alpha(x_{k-1})$



$p(x_a) = \mu_\beta(x_a)\mu_\alpha(x_a)$ ⟵ Randverteilung über Anfragevariable x_a: Produkt der Nachrichten

Inferenz: Message-Passing

- Laufzeit:
  - Berechnung einer Nachricht:

$$\forall\, x_k: \quad \mu_\beta(x_k) = \sum_{x_{k+1}} \psi_{k,k+1}(x_k, x_{k+1})\mu_\beta(x_{k+1})$$

$\Rightarrow O(K^2)$ für Berechnung einer Nachricht (K diskrete Zustände)

  - N Nachrichten insgesamt

$\Rightarrow O(NK^2)$ Gesamtlaufzeit

  - Viel besser als naive Inferenz mit $O(K^N)$

Message-Passing mit Evidenz

- Bisher Randverteilung $p(x_a)$ ohne Evidenz bestimmt
- Was ist wenn wir Evidenz haben?

$$\text{Notation}: \{x_1, ..., x_N\} = \{ \underbrace{x_a}_{\substack{\text{Anfrage-}\\ \text{Variable}}} , \underbrace{x_{i_1}, ..., x_{i_m}}_{\text{Evidenz-Variablen}} , \underbrace{x_{j_1}, ..., x_{j_k}}_{\text{restliche Variablen}} \}$$

- Bedingte Verteilung

$$p(x_a \mid x_{i_1}, .., x_{i_m}) = \frac{p(x_a, x_{i_1}, ..., x_{i_m})}{p(x_{i_1}, ..., x_{i_m})}$$

$$= \frac{1}{Z} p(x_a, x_{i_1}, ..., x_{i_m})$$

Z einfach zu berechnen

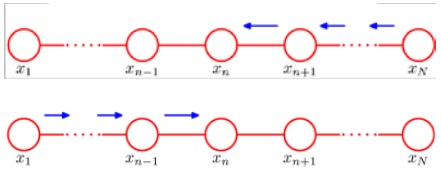
(Normalisierer univariate Verteilung)

Message-Passing mit Evidenz

- Ziel: $p(x_a, x_{i_1}, ..., x_{i_m}) = ?$
- Leichte Modifikation des Message-Passing Algorithmus
  - Wir berechnen wie bisher Nachrichten

$$\mu_\beta(x_{N-1}), ..., \mu_\beta(x_a)$$

$$\mu_\alpha(x_2), ..., \mu_\alpha(x_a)$$



  - Falls x_{k+1} unbeobachtet ist, summieren wir diesen Knoten aus

$$k+1 \notin \{i_1, ..., i_m\} \quad \Rightarrow \quad \mu_\beta(x_k) = \sum_{x_{k+1}} \psi_{k,k+1}(x_k, x_{k+1}) \mu_\beta(x_{k+1})$$

  - Falls x_{k+1} beobachtet ist, verwenden wir nur den entsprechenden Summanden

x_{k+1} beobachteter Wert (Evidenz)

$$k+1 \in \{i_1, ..., i_m\} \quad \Rightarrow \quad \mu_\beta(x_k) = \psi_{k,k+1}(x_k, x_{k+1}) \mu_\beta(x_{k+1})$$

Message-Passing mit Evidenz

- Ebenso für $\mu_\alpha(x_k)$

$$\mu_\alpha(x_k) = \begin{cases} \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1}, x_k)\mu_\alpha(x_{k-1}) : k-1 \notin \{i_1, ..., i_m\} & \text{(Knoten nicht beobachtet)} \\ \psi_{k-1,k}(x_{k-1}, x_k)\mu_\alpha(x_{k-1}) : k-1 \in \{i_1, ..., i_m\} & \text{(Knoten beobachtet)} \end{cases}$$

- Jetzt gilt

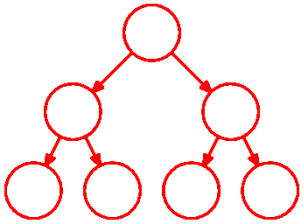
$$p(x_a, x_{i_1}, ..., x_{i_m}) = \mu_\alpha(x_a)\mu_\beta(x_a)$$

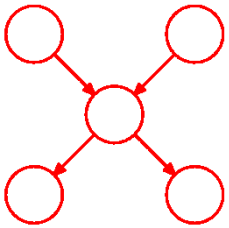
- Laufzeit für Inferenz mit Evidenz immer noch $O(NK^2)$

Inferenz in Allgemeinen Graphen

- Bisher nur Spezialfall: Inferenz auf linearer Kette
- Die Grundidee des Message-Passing funktioniert auch auf allgemeineren Graphen
- Erweiterung: Exakte Inferenz auf *Polytrees*
  - Polytree: Gerichteter Graph, in dem es zwischen zwei Knoten immer genau einen ungerichteten Pfad gibt
  - Etwas allgemeiner als gerichteter Baum

Gerichteter Baum



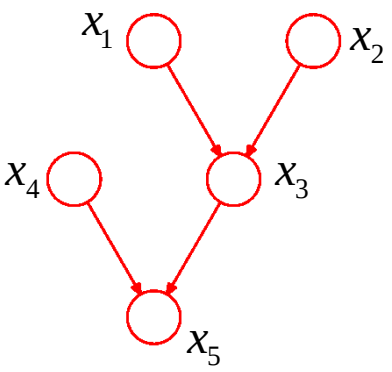


Polytree

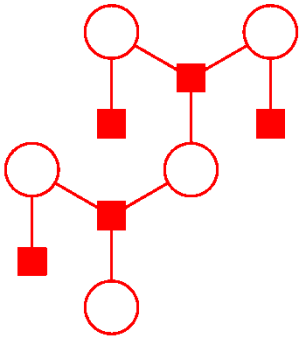
Inferenz in Allgemeinen Graphen

- Grundidee Message-Passing auf Polytrees:
  - Umwandlung in *Faktor-Graph* (ungerichteter Baum)

Ursprünglicher Graph



Faktor-Graph



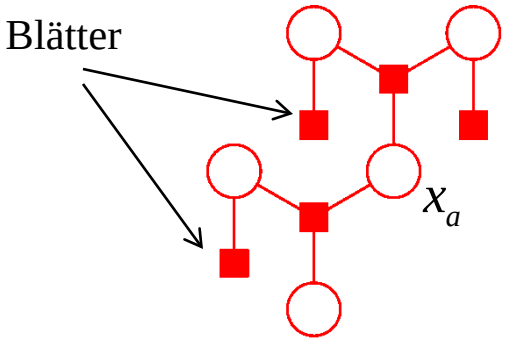
Gemeinsame Verteilung

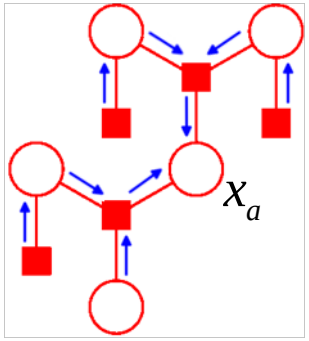
$$p(x_1, x_2, x_3, x_4, x_5) =$$
$$p(x_1)\,p(x_2)\,p(x_3 \mid x_1, x_2)\,p(x_4)\,\underbrace{p(x_5 \mid x_3, x_4)}_{\textit{Faktor}}$$

- Faktor-Knoten
  - Für jeden Faktor in der gemeinsamen Verteilung gibt es einen Faktor-Knoten
  - Ungerichtete Kanten von den Faktor-Knoten zu den im Faktor auftauchenden Variablen

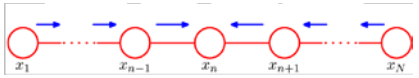
Inferenz in Allgemeinen Graphen (Skizze)

- Falls der ursprüngliche Graph ein Polytree war, ist der Faktor-Graph ein ungerichteter Baum (dh zykelfrei).



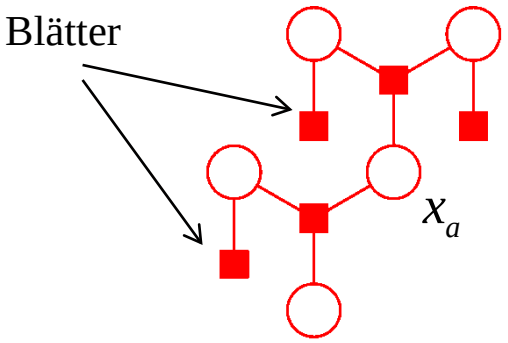


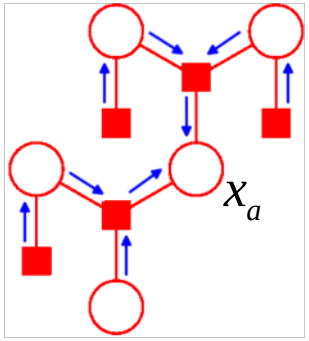
Spezialfall
lineare Kette



- Betrachten Anfragevariable x_a als Wurzel des Baumes
- Nachrichten von den Blättern zur Wurzel schicken (immer eindeutiger Pfad, weil Baum)
- Es gibt zwei Typen von Nachrichten: Faktor-Nachrichten und Variablen-Nachrichten

Inferenz in Allgemeinen Graphen (Skizze)

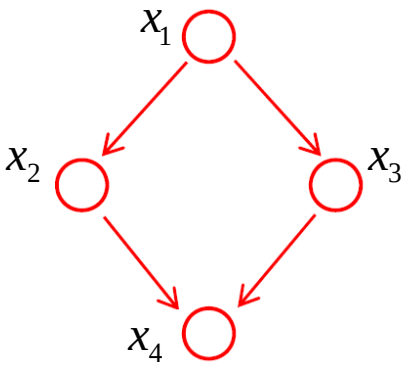


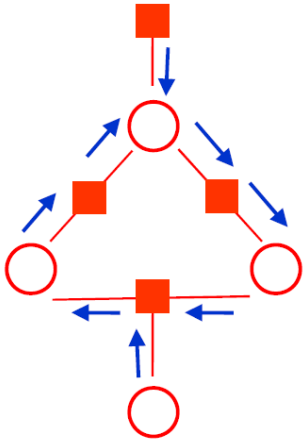


- Nachrichten werden „verschmolzen“, dabei müssen wir über mehrere Variablen summieren
- Grundidee dieselbe wie bei Inferenz auf der linearen Kette: geschicktes Aussummieren
- Laufzeit abhängig von Graphstruktur, exponentiell im worst-case
- Details im Bishop-Textbuch („Sum-Product“ Algorithmus)

Inferenz in Allgemeinen Graphen

- Inferenz in Graphen, die keine Polytrees sind?
- Approximativer Ansatz: Iteratives Message-Passing Schema, wegen Zyklen im Graph nicht exakt





„Loopy Belief Propagation“

$$p(\mathbf{x}) = p(x_1)\,p(x_2 \mid x_1)\,p(x_3 \mid x_1)\,p(x_4 \mid x_2, x_3)$$

- Alternative für exakte Inferenz in allgemeinen Graphen:
  - Graph in einen äquivalenten azyklischen Graphen umwandeln
  - „Junction Tree“ Algorithmus, (i.A. exponentielle Laufzeit)

Überblick

- Gerichtete Graphische Modelle: Bayessche Netze
- Graphische Modelle im Maschinellen Lernen
- Inferenz in Graphischen Modellen
  - Exakte Inferenz: Message-Passing
  - Approximative Inferenz: Sampling

Approximative Inferenz

- Exakte Inferenz NP-hart: In der Praxis spielen *approximative* Inferenzverfahren wichtige Rolle

- Wir betrachten Sampling-basierte Verfahren
  - Relativ einfach zu verstehen/implementieren
  - Anytime-Algorithmen (je länger die Laufzeit, desto genauer)

Inferenz: Sampling-basiert

- Grundidee Sampling:
  - Wir interessieren uns für eine Verteilung $p(\mathbf{z})$, $\mathbf{z}$ ist eine Menge von Zufallsvariablen (z.B. bedingte Verteilung über Anfragevariablen in graphischem Modell)
  - Es ist schwierig, $p(\mathbf{z})$ direkt auszurechnen
  - Stattdessen ziehen wir „Samples“ (Stichproben)

$$\mathbf{z}^{(k)} \sim p(\mathbf{z}) \qquad \text{i.i.d.,} \quad k = 1, \ldots, K,$$

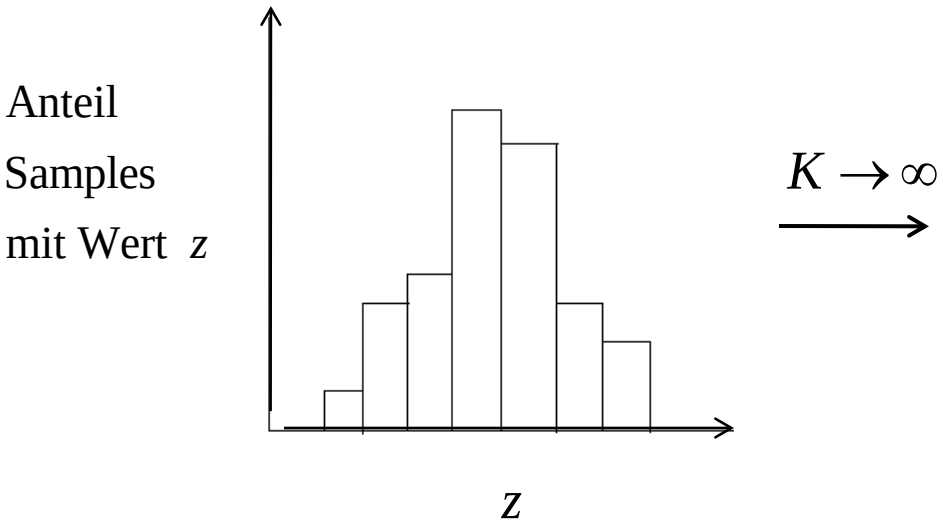
    jedes Sample $\mathbf{z}^{(k)}$ ist eine vollständige Belegung der Zufallsvariablen in $\mathbf{z}$

- Die Samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(K)}$ approximieren die Verteilung $p(\mathbf{z})$

Inferenz: Sampling-basiert

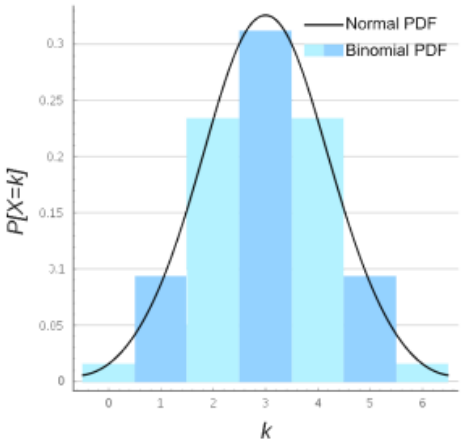
- Beispiel:
  - Eindimensionale Verteilung, $\mathbf{z} = \{z\}$
  - Diskrete Variable mit Zuständen {0,…,6}: Anzahl „Kopf“ bei 6 Münzwürfen
  - K=100 Experimente, in denen wir jeweils Münze 6x werfen

Sample-Histogramm



$K \to \infty$

Echte Verteilung (Binomial)



Sampling-Inferenz für Graphische Modelle

- Gegeben graphisches Modell, repräsentiert Verteilung durch

$$p(x_1,...,x_N) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

- Etwas allgemeinere Inferenz-Problemstellung: Menge von Anfragevariablen

$$p(\mathbf{x}_I \mid \mathbf{x}_D) \approx ?$$

$\mathbf{x}_I \subseteq \mathbf{x} = \{x_1,...,x_N\}$ Menge von Anfragevariablen

$\mathbf{x}_D \subseteq \mathbf{x} = \{x_1,...,x_N\}$ Menge von Evidenzvariablen

- Wir betrachten zunächst den Fall ohne Evidenz:

$$p(\mathbf{x}_I) \approx ? \qquad \mathbf{x}_I = \{x_{i_1},...,x_{i_m}\} \subseteq \{x_1,...,x_N\}$$

Sampling-Inferenz für Graphische Modelle

- Ziel: Samples aus der Randverteilung $p(\mathbf{x}_I) = p(x_{i_1}, ..., x_{i_m})$

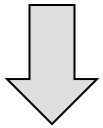
$$\mathbf{x}_I^{(k)} \sim p(\mathbf{x}_I) \qquad k = 1, ..., K$$

- Es genügt, Samples aus der Gesamtverteilung $p(\mathbf{x}) = p(x_1, ..., x_N)$ zu ziehen:

$$\mathbf{x}^{(k)} = (x_1^{(k)}, ..., x_N^{(k)}) \sim p(x_1, ..., x_N) \qquad k = 1, ..., K$$

- Samples aus der Randverteilung $p(x_{i_1}, ..., x_{i_m})$ erhalten wir einfach durch Projektion der Samples auf die $\{x_{i_1}, ..., x_{i_m}\}$

$$\mathbf{x}^{(k)} = (x_1^{(k)}, ..., x_N^{(k)}) \sim p(x_1, ..., x_N) \qquad k = 1, ..., K$$



Projektion

$$\mathbf{x}_I^{(k)} = (x_{i_1}^{(k)}, ..., x_{i_m}^{(k)}) \sim p(x_{i_1}, ..., x_{i_m}) \qquad k = 1, ..., K$$

Inferenz: Ancestral Sampling

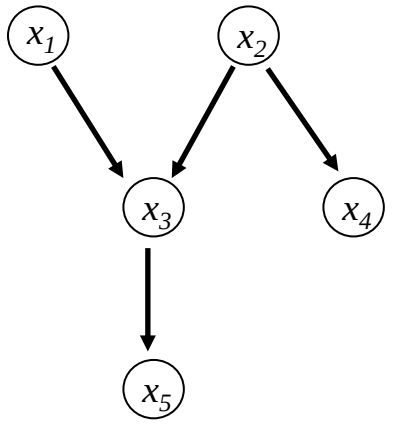
- Wie ziehen wir Samples $\mathbf{x}^{(k)} \sim p(\mathbf{x})$?

- Einfach bei gerichteten graphischen Modellen: „Ancestral Sampling“
  - Nutze Faktorisierung der gemeinsamen Verteilung

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1, \ldots, x_N)$$
$$= \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

  - „Ziehen entlang der Kanten“

„Ziehen entlang der Kanten“



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

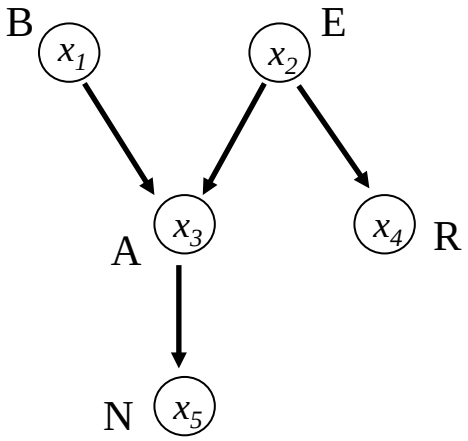
$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$...$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

Schon gezogene Werte

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,.,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel

$$x_1^{(k)} \sim p(x_1) \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$...$$
$$x_N^{(k)} \sim p(x_N \mid p$$

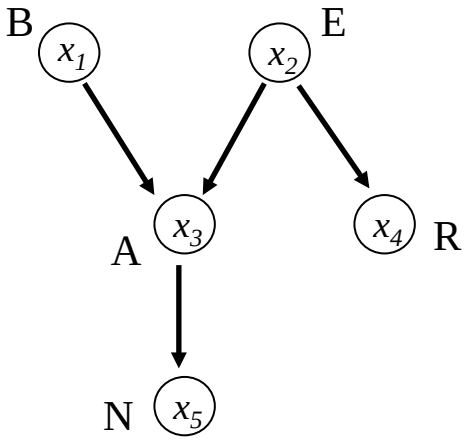
Schon gezogene Werte

P(B=1)
0.1

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,.,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel

$$x_1^{(k)} \sim p(x_1) \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

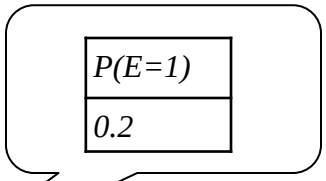
- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$...$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

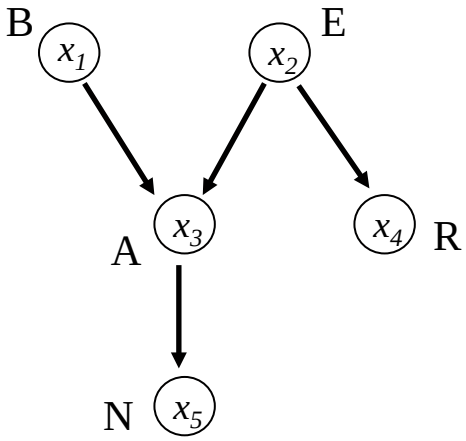
Schon gezogene Werte

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,..,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel



$$x_1^{(k)} \sim p(x_1) \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

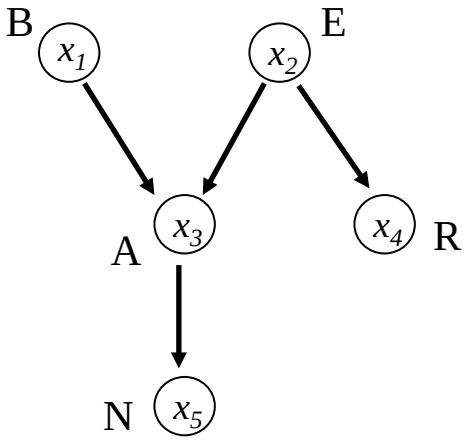
$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa$$

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,..,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

B	E	P(A=1|B,E)
0	0	0.01
0	1	0.5
1	0	0.9
1	1	0.95

- Beispiel

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2) \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$...$$
$$x_N^{(k)} \sim p(x_N \mid pa($$

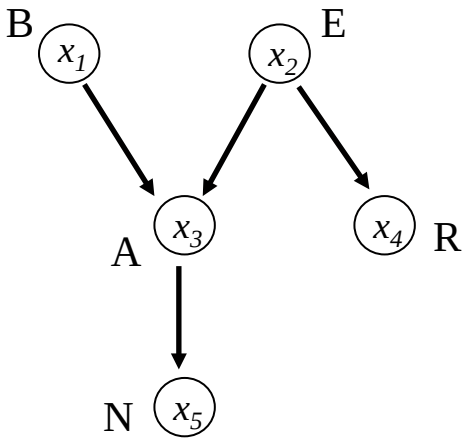
Schon gezogene Werte

E	P(R=1|E)
0	0.01
1	0.5

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,.,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2)$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

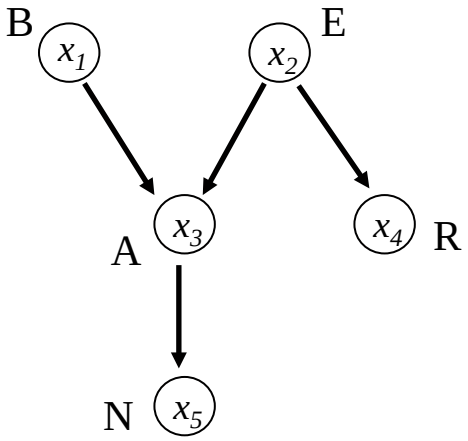
Schon gezogene Werte

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,..,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel

A	P(N=1|A)
0	0.1
1	0.7

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2)$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = \ldots$$
$$x_4^{(k)} \sim p(x_4 \mid x_3 = \ldots) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$



Inferenz: Ancestral Sampling

- Wir ziehen ein Sample $(x_1,...,x_N)^{(k)}$, indem wir nacheinander die einzelnen x_i ziehen

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$...$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

Schon gezogene Werte

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1,.,x_N) = \prod_{n=1}^{N} p(x_n \mid pa(x_n))$$

- Beispiel

$$x_1^{(k)} \sim p(x_1) \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \rightarrow x_5 = 1$$

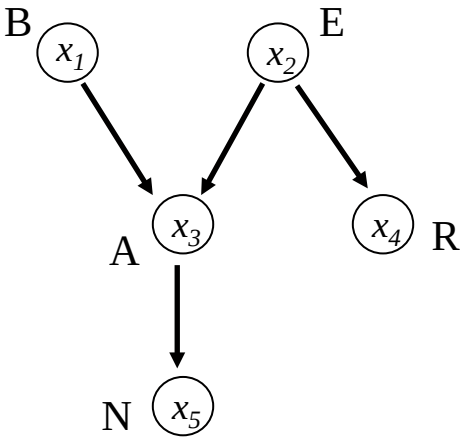
$$\Rightarrow \quad \mathbf{x}^{(k)} = (1,0,1,0,1)$$

Inferenz: Ancestral Sampling

- Beispiel für Schätzung der Randverteilungen aus Samples:

$$\mathbf{x}^{(1)} = (1,0,1,0,1)$$
$$\mathbf{x}^{(2)} = (0,0,0,0,0)$$
$$\mathbf{x}^{(3)} = (0,1,0,1,0)$$
$$\mathbf{x}^{(4)} = (0,1,1,0,1)$$
$$\mathbf{x}^{(5)} = (0,0,0,0,0)$$

$$p(x_3 = 1) \approx 0.4$$
$$p(x_4 = 1) \approx 0.2$$
$$p(x_5 = 1) \approx 0.4$$



- Analyse Ancentral Sampling
  - + Zieht direkt aus der korrekten Verteilung
  - + Effizient
  - - Funktioniert nur ohne Evidenz

Inferenz: Logic Sampling

- Wie erhalten wir Samples unter Evidenz?

Beobachtete Variablen

$$\mathbf{x}_I^{(k)} \sim p(\mathbf{x}_I \mid \mathbf{x}_D) = p(x_{i_1}, ..., x_{i_m} \mid x_{j_1}, ..., x_{j_l})$$

- Logic Sampling: Ancestral Sampling + Zurückweisung von Samples, die nicht mit der Beobachtung konsistent sind
  - Ancestral Sampling: vollständige Samples

$$\mathbf{x}^{(k)} = (x_1^{(k)}, ..., x_N^{(k)}) \sim p(\mathbf{x})$$

  - Fallunterscheidung:

$(x_{j_1}^{(k)}, ..., x_{j_l}^{(k)}) = (x_{j_1}, ..., x_{j_l})$ [Sample konsistent mit Beobachtung]: akzeptiere $\mathbf{x}^{(k)}$

$(x_{j_1}^{(k)}, ..., x_{j_l}^{(k)}) \neq (x_{j_1}, ..., x_{j_l})$ [Sample ìnkonsistent mit Beobachtung]: weise $\mathbf{x}^{(k)}$ zurück

Inferenz: Logic Sampling

- Die im Logic Sampling akzeptierten Samples $\mathbf{x}^{(k)}$ repräsentieren die bedingte Verteilung gegeben Evidenz:

$$\mathbf{x}^{(k)} \sim p(\mathbf{x} \mid \mathbf{x}_D)$$

- Marginale Samples

$$\mathbf{x}_I^{(k)} \sim p(\mathbf{x}_I \mid \mathbf{x}_D)$$

  wieder durch Projektion auf Anfragevariablen

- Problem: Oft werden fast alle Samples verworfen
  - Wahrscheinlichkeit, Sample zu generieren, das mit $\mathbf{x}_D$ konsistent ist, sinkt meist exponentiell schnell mit Größe von D
  - Entsprechend exponentielle Laufzeit, um ausreichende Menge von Samples zu erhalten
  - In der Praxis selten anwendbar

Inferenz: MCMC

- Alternative Strategie zum Erzeugen von Samples: Markov Chain Monte Carlo („MCMC“)
- Idee:
  - Schwierig, direkt Samples aus $p(\mathbf{z})$ zu ziehen
  - Alternativstrategie: Konstruiere Folge von Samples

$$\mathbf{z}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \mathbf{z}^{(3)} \rightarrow \mathbf{z}^{(4)} \rightarrow \mathbf{z}^{(5)} \rightarrow$$

$\mathbf{z}^{(0)}$ zufällig initialisiert $\qquad$ $\mathbf{z}^{(t+1)} \sim p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{t})$

durch mehrfache probabilistische Update-Schritte $\mathbf{z}^{(t+1)} \sim p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{t})$.

  - Wenn Updates geeignet gewählt, gilt asymptotisch

$\mathbf{z}^{(T)} \sim p(\mathbf{z})$ ungefähr, für sehr grosse T

ZV: T-te Variablenbelegung

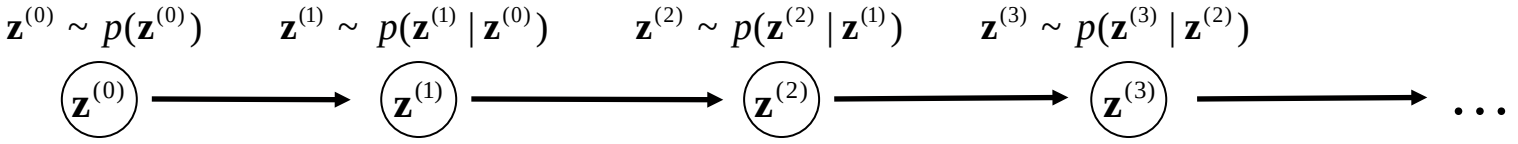
Markov-Ketten

- Betrachte Folge der Samples

$$\mathbf{z}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \mathbf{z}^{(3)} \rightarrow \mathbf{z}^{(4)} \rightarrow \mathbf{z}^{(5)} \rightarrow$$

  als Zufallsvariablen, $\mathbf{z}^{(t)}$ heisst Zustand der Kette zum Zeitpunkt *t*

- Diese Zufallsvariablen bilden Markov-Kette:



- Homogene Kette: beschrieben durch von *t* unabhängige Transitionswahrscheinlichkeiten, $p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)}) = p(\mathbf{z}^{(t)} \mid \mathbf{z}^{(t-1)})$

$$\mathrm{T}(\mathbf{z}^{(t)}, \mathbf{z}^{(t+1)}) = p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)})$$

Wahrscheinlichkeit Übergang $\mathbf{z}^{(t)} \rightarrow \mathbf{z}^{(t+1)}$

Neuer Zustand

Aktueller Zustand

Markov-Ketten

- Verteilung über Folgezustand berechnen aus Verteilung über aktuellem Zustand

Folgezustand ↘ ↙ Aktueller Zustand

$$p(\mathbf{z}^{(t+1)}) = \sum_{\mathbf{z}_t} p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)})\, p(\mathbf{z}^{(t)})$$

$$= \sum_{\mathbf{z}_t} T(\mathbf{z}^{(t)}, \mathbf{z}^{(t+1)})\, p(\mathbf{z}^{(t)})$$

- Stationäre Verteilung
  - Eine Verteilung $p_*(\mathbf{z})$ über die Zustände der Kette heisst „stationär“, falls sie bei einem Schritt der Markov-Kette nicht verändert wird:

$$\mathbf{z}^{(t)} \sim p_*(\mathbf{z}) \quad \Rightarrow \quad \mathbf{z}^{(t+1)} \sim p_*(\mathbf{z})$$

$$p_*(\mathbf{z}') = \sum_{\mathbf{z}} T(\mathbf{z}, \mathbf{z}')\, p_*(\mathbf{z})$$

Markov-Ketten: Stationäre Verteilung

- Falls die Kette eine stationäre Verteilung erreicht, dh. $p(\mathbf{z}^{(T)}) = p_*(\mathbf{z}^{(T)})$ für ein geeignetes T, so bleibt diese Verteilung erhalten, dh. $p(\mathbf{z}^{(T+N)}) = p_*(\mathbf{z}^{(T+N)})$ für alle N.
  - Kette ist „konvergiert“ zur Verteilung p_*

- Unter bestimmten Bedingungen („Ergodische Ketten“) konvergiert eine Markov-Kette für $t \to \infty$ gegen eine eindeutige stationäre Verteilung, diese heisst dann „Gleichgewichtsverteilung“

Inferenz: MCMC

- Gegeben graphisches Modell über ZV $\mathbf{x} = \{x_1,\ldots,x_N\}$, definiert Verteilung $p(\mathbf{x})$
- Annahme zunächst: keine Evidenz
- „Markov Chain Monte Carlo“ Methoden
  - Konstruiere aus dem graphischen Modell eine Folge von Samples durch iterative probabilistische Updates

$$\mathbf{x}^{(0)} \rightarrow \mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)} \rightarrow \mathbf{x}^{(3)} \rightarrow \mathbf{x}^{(4)} \rightarrow \mathbf{x}^{(5)} \rightarrow$$

$\mathbf{x}^{(t)}$ jeweils Belegung aller Knoten im Netz

$\mathbf{x}^{(0)}$ zufällig initialisiert

$$\mathbf{x}^{(t+1)} \sim p(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{t})$$

  - Ziel: Updates so wählen, dass sich ergodische Markov-Kette mit Gleichgewichtsverteilung $p(\mathbf{x})$ ergibt
  - Einfachste Methode: lokales Ziehen einer Variable, gegeben Zustand der anderen Variablen („Gibbs-Sampling“)

Inferenz: Gibbs Sampling

- Gibbs Sampling: Eine Version von MCMC
- Übergangswahrscheinlichkeiten bestimmt durch wiederholtes lokales Ziehen einer ZV, gegeben den Zustand aller anderen ZV
  - Gegeben alter Zustand $\mathbf{x} = (x_1, ..., x_N)$
  - Ziehen des neuen Zustands $\mathbf{x}' = (x_1', ..., x_N')$:

$$x_1' \sim p(x_1 \mid \overbrace{x_2, ..., x_N}^{\text{Beobachtete (alte) Werte}})$$

$$x_2' \sim p(x_2 \mid x_1', x_3, ..., x_N)$$

$$x_3' \sim p(x_3 \mid x_1', x_2', x_4, ..., x_N)$$

$$...$$

$$x_N' \sim p(x_N \mid x_1', x_2', ..., x_{N-1}')$$

Anfangs zufällige Initialisierung

Inferenz: Gibbs Sampling

- Satz: Falls $p(x_n \mid x_1, x_2, ..., x_{n-1}, x_{n+1}, ..., x_{N-1}) \neq 0$ für alle n und alle möglichen Zustände x_i, so ist die resultierende Markov-Kette ergodisch mit Gleichgewichtsverteilung $p(\mathbf{x})$.

- Einzelner Gibbs-Schritt einfach, bedingte Verteilung über eine Variable gegeben Evidenz auf **allen** anderen Variablen direkt auszurechnen:

Berechnung von $p(x_1 \mid \overbrace{x_2, ..., x_N}^{\text{Beobachtete (alte) Werte}})$:

Für $x_1 = 1$ berechne $p_1 = \mathrm{p}(x_1, x_2, ..., x_N)$

Für $x_1 = 0$ berechne $p_0 = \mathrm{p}(x_1, x_2, ..., x_N)$

$$p(x_1 = 1 \mid x_2, ..., x_N) = \frac{p_1}{p_1 + p_0}$$

Gibbs-Sampling mit Evidenz

- Bisher haben wir Inferenz ohne Evidenz betrachtet
- Wie erhalten wir Samples aus der bedingten Verteilung?

Ziel: $\mathbf{x}^{(T)} \sim p(\mathbf{x} \mid \mathbf{x}_D)$ ungefähr, für sehr grosse T

- Leichte Modifikation der Gibbs-Sampling Methode:
  - Gibbs-Sampling zieht immer eine Variable x_i neu, gegeben Zustand der anderen Variablen
  - Mit Evidenz: Nur die unbeobachteten Variablen werden jeweils neu gezogen, die beobachteten Variablen werden fest auf den beobachteten Wert gesetzt

Inferenz: Gibbs Sampling

- Zusammenfassung Gibbs Sampling Algorithmus:
  - $\mathbf{x}^{(0)} =$ zufällige Initialisierung aller ZV, konsistent mit Evidenz $\mathbf{x}_D$
  - Für $t = 1, ..., T$: $\mathbf{x}^{(t)} = \text{Gibbs-update}(\mathbf{x}^{(t-1)})$
  - Die Samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, ...$ sind asymptotisch verteilt nach $p(\mathbf{x} \mid \mathbf{x}_D)$
- Gibbs Sampling in vielen praktischen Anwendungen brauchbar
  - Einzelne Update-Schritte effizient
  - Garantierte Konvergenz (für $t \to \infty$)
  - Erlaubt, Samples aus $p(\mathbf{x} \mid \mathbf{x}_D)$ zu ziehen, ohne dass Laufzeit explodiert wenn Evidenzmenge groß (im Gegensatz zu Logic Sampling)

Inferenz: Gibbs Sampling

- Gibbs-Sampling: Konvergenz
  - Konvergenz der Markov-Kette $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots$ ist nur garantiert für $t \to \infty$.
  - In der Praxis: „Burn-In“ Iterationen, bevor Samples verwendet werden (verwerfe Samples $\mathbf{x}^{(t)}$ für $t \leq T_{Burn-in}$)
  - Es gibt auch Konvergenztests, um Anzahl der Burn-In Iterationen zu bestimmen

Inferenz: Zusammenfassung

- Exakte Inferenz
  - Message-Passing Algorithmen
  - Exakte Inferenz auf Polytrees (mit Junction-Tree Erweiterung auf allgemeinen Graphen)
  - Laufzeit abhängig von Graphstruktur, exponentiell im worst-case

- Approximative Inferenz
  - Sampling-Methoden: Approximation durch Menge von „Samples“, exakte Ergebnisse für $t \rightarrow \infty$.
    - Ancestral Sampling: einfach, schnell, keine Evidenz
    - Logic Sampling: mit Evidenz, aber selten praktikabel
    - MCMC/Gibbs-Sampling: Effizientes approximatives Ziehen von Samples unter Evidenz