Überblick

- Graphische Modelle: Syntax und Semantik
- Graphische Modelle im Maschinellen Lernen
- Inferenz in Graphischen Modellen (exakt, approximativ)
- Sequenzmodelle

Problemstellung Inferenz

- Gegeben graphisches Modell über Menge von ZV $\{X_1, \ldots, X_N\}$.

- Problemstellung Inferenz:
  - Variablen mit Evidenz $X_{i_1}, \ldots, X_{i_m}$ $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, N\}$
  - Anfrage-Variable X_a $a \in \{1, \ldots, N\} \setminus \{i_1, \ldots, i_m\}$

  - Berechne Verteilung über Anfrage-Variable gegeben Evidenz

Bedingte Verteilung über ZV X_a

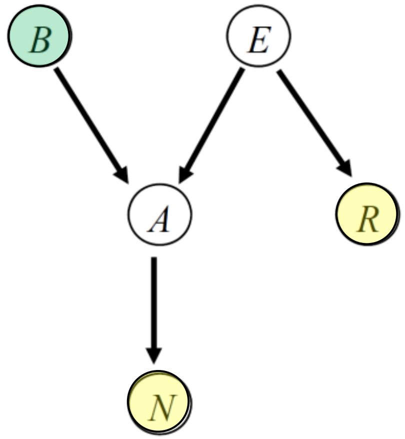
Evidenz: beobachtete Werte für ZV $X_{i_1}, \ldots, X_{i_m}$

Berechne $p(x_a \mid x_{i_1}, \ldots, x_{i_m})$

Allgemeiner auch $p(x_{a_1}, \ldots, x_{a_k} \mid x_{i_1}, \ldots, x_{i_m})$

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:

$$p(B=1 \mid N=1, R=0) = 0.7$$
$$p(B=0 \mid N=1, R=0) = 0.3$$

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

- Posterior über Parameter, Bayessche Vorhersage, …

Graphische Modelle: Inferenz

- Inferenz schwieriges Problem
  - Allgemeine graphische Modelle: exakte Inferenz NP-hart
  - Es gibt Algorithmen für exakte Inferenz in allgemeinen graphischen Modellen, deren Laufzeit von den Eigenschaften der Graphstruktur abhängt („Message-Passing“)
  - Es gibt verschiedene Techniken für approximative Inferenz (Sampling, Variational Inference, Expectation Propagation)

- Wir betrachten
  - Message-Passing Algorithmus: in Spezialfällen
  - Sampling-basierte approximative Inferenz

Inferenz: Diskrete vs. Kontinuierliche Variablen

- Wir diskutieren Inferenz nur für diskrete Variablen

- Betrachtete Inferenzalgorithmen sind auch auf kontinuierliche Variablen anwendbar
  - Summen ersetzen durch Integrale
  - Verteilungen müssen so gewählt sein, dass sich die entsprechenden Integrale in geschlossener Form ausrechnen lassen

Überblick

- Graphische Modelle: Syntax und Semantik
- Graphische Modelle im Maschinellen Lernen
- Inferenz in Graphischen Modellen
  - Exakte Inferenz
  - Approximative Inferenz
- Sequenzmodelle

Exakte Inferenz: Naiv

- Graphisches Modell: Repräsentation von $p(X_1, ..., X_N)$

- Naive Inferenz:

$$\text{Notation}: \{X_1, ..., X_N\} = \{ \underbrace{X_a}_{\substack{\text{Anfrage-}\\ \text{Variable}}}, \underbrace{X_{i_1}, ..., X_{i_m}}_{\text{Evidenz-Variablen}}, \underbrace{X_{j_1}, ..., X_{j_k}}_{\text{restliche Variablen}} \}$$

$$\text{Berechne für jeden Wert } x_a: \quad p(x_a \mid x_{i_1}, ..., x_{i_m}) = \frac{p(x_a, x_{i_1}, ..., x_{i_m})}{p(x_{i_1}, ..., x_{i_m})}$$

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

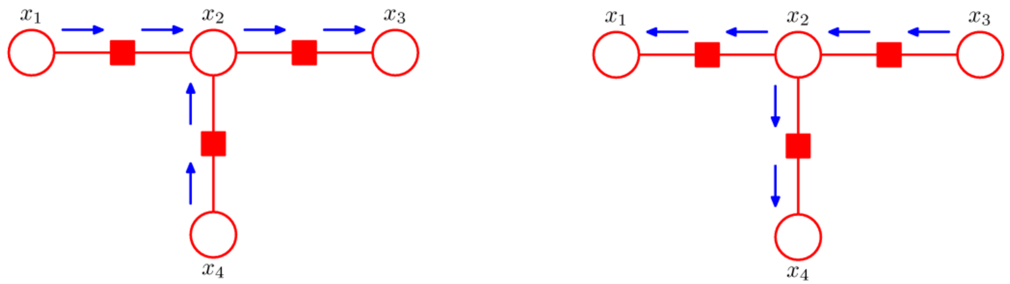
$$= \frac{1}{Z} \sum_{x_{j_1}} \sum_{x_{j_2}} \cdots \sum_{x_{j_k}} p(x_1, ..., x_N)$$

Z Normalisierungsfaktor,
leicht explizit zu berechnen
bei univariaten Verteilungen

Zentrales Problem: Aussummieren aller restlichen Variablen (exponentiell, wenn naiv gelöst)

Effizientere Inferenzmethoden?

- Effizientere Methode als naive Inferenz?
  - Für allgemeine Graphen (vollständig verbunden) nicht möglich!
  - Aber wenn es Struktur im Modell gibt (Unabhängigkeiten), können wir diese unter Umständen ausnutzen
- Idee: Lokale Berechnungen, die entlang der Graphstruktur propagiert werden
  - Knoten schicken sich gegenseitig „Nachrichten“, die Ergebnisse von Teilberechnungen enthalten
  - „Message Passing“, „Belief Propagation“
  - Laufzeit der Verfahren hängt von Graphstruktur ab (exponentiell im worst-case)



Graphische Modelle: Inferenz auf linearer Kette

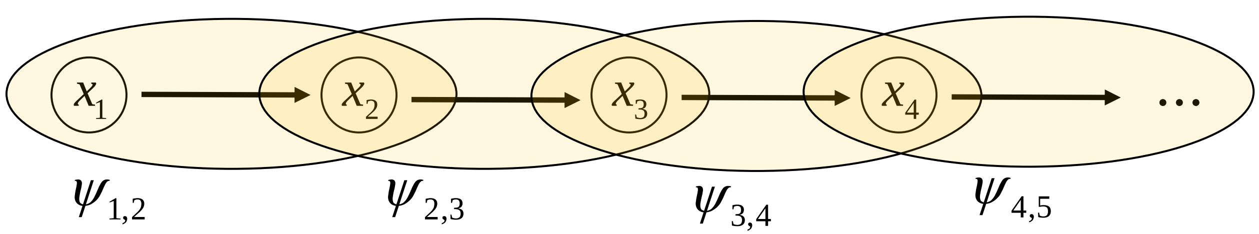
- Wir betrachten jetzt den Message-Passing Algorithmus in einem Spezialfall mit besonders einfacher Struktur: lineare Kette von Zufallsvariablen

$$x_1 \longrightarrow x_2 \longrightarrow x_3 \longrightarrow x_4 \longrightarrow \cdots$$

$$p(x_1,....,x_N) = p(x_1)\,p(x_2 \mid x_1)\,p(x_3 \mid x_2)\cdot ... \cdot p(x_N \mid x_{N-1})$$

- Notation: Darstellung der gemeinsamen Verteilung als Produkt von *Potenzialfunktionen* über Paare von ZV

$$p(x_1,...,x_N) = \overbrace{\psi_{1,2}(x_1,x_2)}^{p(x_1)p(x_2|x_1)}\,\overbrace{\psi_{2,3}(x_2,x_3)}^{p(x_3|x_2)}\cdot ... \cdot \overbrace{\psi_{N-1,N}(x_{N-1},x_N)}^{p(x_N|x_{N-1})}$$



Inferenz: lineare Kette von ZV

- Einführung des Message-Passing Algorithmus am Beispiel
- Lineare Kette von 5 Zufallsvariablen:

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5$$

- Randverteilung der 3. Zufallsvariable berechnen (keine Evidenz)

Anfrage-Variable

Restliche Variablen (aussummieren)

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} p(x_1, x_2, x_3, x_4, x_5)$$

$$= \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,4}(x_3, x_4)\psi_{4,5}(x_4, x_5)$$

- Naive Berechnung exponentiell (Mehrfachsumme)
- Idee: Struktur (lineare Kette) ausnutzen, um Berechnung effizienter durchzuführen

Inferenz: Message-Passing

- Nutze Faktorisierung der gemeinsamen Verteilung in Potenziale (Unabhängigkeiten)

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \psi_{4,5}(x_4, x_5)$$

$$= \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \underbrace{\sum_{x_5} \psi_{4,5}(x_4, x_5)}_{\text{Lokale Teilberechnung: "Nachricht" } \mu_\beta(x_4)}$$

- Lokale Teilberechnung: „Nachricht“ $\mu_\beta(x_4)$
  - Berechne für alle Werte von x_4: $\mu_\beta(x_4) = \sum_{x_5} \psi_{4,5}(x_4, x_5)$
  - Nachricht ist Funktion in Abhängigkeit vom Zustand x_4 (z.B. kodiert als Vektor)
  - In der Nachricht ist der Knoten X_5 aussummiert

Inferenz: Message-Passing

- Nutze Faktorisierung der gemeinsamen Verteilung in Potenziale (Unabhängigkeiten)

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \psi_{4,5}(x_4, x_5)$$

$$= \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \underbrace{\sum_{x_5} \psi_{4,5}(x_4, x_5)}_{\text{Lokale Teilberechnung: "Nachricht" } \mu_\beta(x_4)}$$

Kodierung z.B. als Vektor: $\mu_\beta(x_4) = \begin{pmatrix} \sum_{x_5} \psi_{4,5}(0, x_5) \\ \sum_{x_5} \psi_{4,5}(1, x_5) \end{pmatrix}$

- Lokale Teilbe
  - Berechne fü

    x_5
  - Nachricht ist Funktion in Abhängigkeit vom Zustand x_4 (z.B. kodiert als Vektor)
  - In der Nachricht ist der Knoten X_5 aussummiert

Inferenz: Message-Passing

- Nutze Faktorisierung der gemeinsamen Verteilung in Potenziale (Unabhängigkeiten)

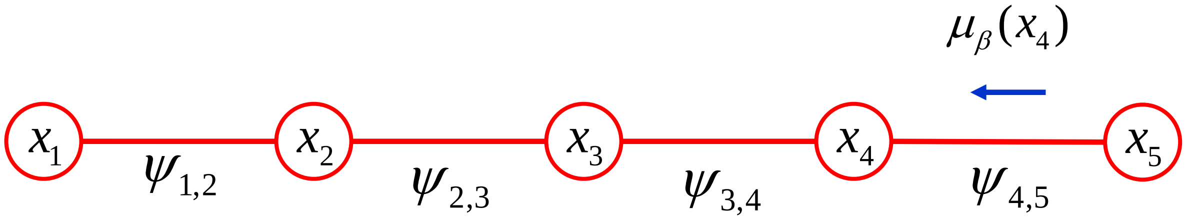
$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,4}(x_3, x_4)\psi_{4,5}(x_4, x_5)$$

$$= \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,4}(x_3, x_4)\underbrace{\sum_{x_5} \psi_{4,5}(x_4, x_5)}_{\text{Lokale Teilberechnung: "Nachricht" } \mu_\beta(x_4)}$$

- Lokale Teilberechnung: „Nachricht“ $\mu_\beta(x_4)$
  - Berechne für alle Werte von x_4: $\mu_\beta(x_4) = \sum_{x_5} \psi_{4,5}(x_4, x_5)$
  - Nachricht ist Funktion in Abhängigkeit vom Zustand x_4 (z.B. kodiert als Vektor)
  - In der Nachricht ist der Knoten X_5 aussummiert

Inferenz: Message-Passing

- **Anschauung**: Wir summieren den Knoten X_5 aus, und schicken das Ergebnis weiter an den Knoten X_4



Inferenz: Message-Passing

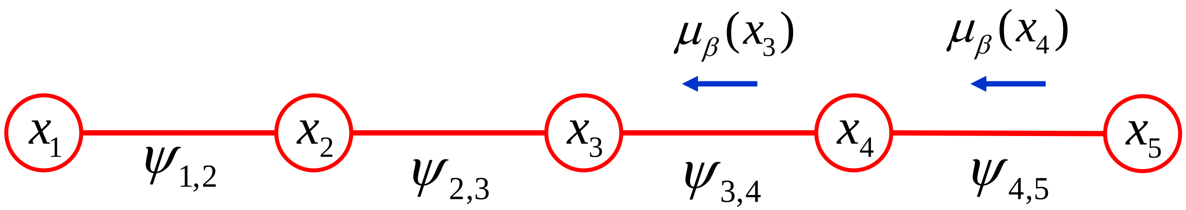
- Wir wenden dieselbe Idee auf die nächste auszusummierenden Variable an

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,4}(x_3, x_4)\mu_\beta(x_4)$$
$$= \sum_{x_1} \sum_{x_2} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\underbrace{\sum_{x_4} \psi_{3,4}(x_3, x_4)\mu_\beta(x_4)}_{\text{Lokale Teilberechnung: "Nachricht" } \mu_\beta(x_3)}$$

- Lokale Teilberechnung: „Nachricht“ $\mu_\beta(x_3)$
  - Berechne für alle Werte von x_3: $\mu_\beta(x_3) = \sum_{x_4} \psi_{3,4}(x_3, x_4)\mu_\beta(x_4)$
  - Nachricht ist Funktion in Abhängigkeit vom Zustand x_3
  - In der Nachricht sind die Knoten X_5, X_4 aussummiert

Inferenz: Message-Passing

- **Anschauung**: Wir summieren den Knoten X_4 aus, und schicken das Ergebnis weiter an den Knoten X_3



- X_3 ist Anfrageknoten, also nicht aussummieren

Inferenz: Message-Passing

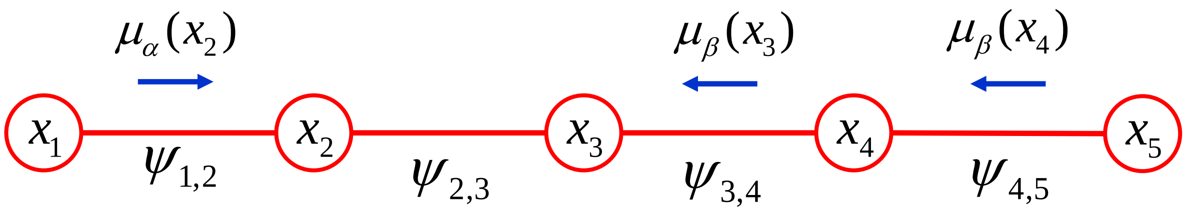
- Gleiches Prinzip anwenden auf die Variablen links vom Anfrageknoten

$$\begin{aligned} p(x_3) &= \sum_{x_1} \sum_{x_2} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \mu_\beta(x_3) \\ &= \mu_\beta(x_3) \sum_{x_2} \sum_{x_1} \psi_{2,3}(x_2, x_3) \psi_{1,2}(x_1, x_2) \\ &= \mu_\beta(x_3) \sum_{x_2} \psi_{2,3}(x_2, x_3) \underbrace{\sum_{x_1} \psi_{1,2}(x_1, x_2)}_{\text{"Nachricht" } \mu_\alpha(x_2)} \end{aligned}$$

- Teilberechnung: „Nachricht“ $\mu_\alpha(x_2)$
  - Berechne für alle Werte von x_2: $\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2)$
  - Nachricht ist Funktion in Abhängigkeit vom Zustand x_2
  - In der Nachricht ist der Knoten X_1 aussummiert

Inferenz: Message-Passing

- **Anschauung**: Wir summieren den Knoten X_1 aus, und schicken das Ergebnis weiter an den Knoten X_2



Inferenz: Message-Passing

- Letzte Variable X_2 aussummieren

$$p(x_3) = \mu_\beta(x_3) \underbrace{\sum_{x_2} \psi_{2,3}(x_2, x_3)\mu_\alpha(x_2)}_{\text{"Nachricht" } \mu_\alpha(x_3)}$$

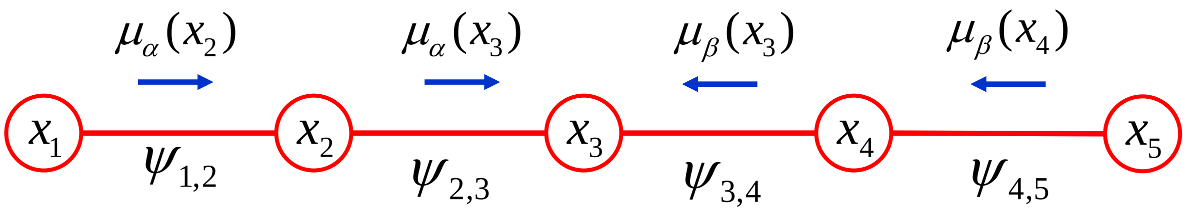
$$= \mu_\beta(x_3)\mu_\alpha(x_3)$$

  - Gesuchte Randverteilung ist Produkt der Nachrichten im Anfrageknoten:

$$p(x_3) = \mu_\beta(x_3)\mu_\alpha(x_3)$$

Inferenz: Message-Passing

- Nachrichten-Austausch Schema:



$$p(x_3) = \mu_\beta(x_3)\mu_\alpha(x_3)$$

Endergebnis: gesuchte Randverteilung ist Produkt der Nachrichten

Inferenz: Message-Passing

- Laufzeit:
  - Berechnung einer Nachricht:

    Berechne für alle Werte von x_3: $\mu_\beta(x_3) = \sum_{x_4} \psi_{3,4}(x_3, x_4)\mu_\beta(x_4)$

    $\Rightarrow O(M^2)$ für Berechnung einer Nachricht (Variablen mit M diskreten Zuständen)
  - $N\text{-}1$ Nachrichten insgesamt

    $\Rightarrow O(NM^2)$ Gesamtlaufzeit
  - Viel besser als naive Inferenz mit $O(M^N)$

Inferenz: Message-Passing Algorithmus

- Algorithmus: Message-Passing auf linearer Kette
  - Eingabe:

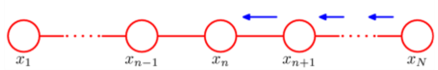
$$p(x_1,\dots,x_N)=\psi_{1,2}(x_1,x_2),\dots,\psi_{N-1,N}(x_{N-1},x_N)$$

Gesucht: $p(x_a)=?$

  - Berechne Nachrichten (rekursiv):

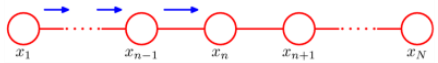
$$\mu_\beta(x_N)=\mathbf{1}$$

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



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

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



  - Ausgabe:

$p(x_a)=\mu_\alpha(x_a)\mu_\beta(x_a)$ (Verteilung)

Message-Passing mit Evidenz

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

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

- Bedingte Verteilung

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

Z einfach zu berechnen

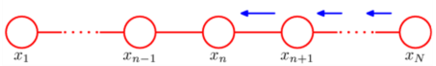
(Normalisierer univariate Verteilung)

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

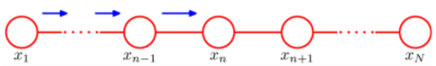
Message-Passing mit Evidenz

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

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



$$\mu_\alpha(x_2), \ldots, \; \mu_\alpha(x_a)$$



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

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

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

x_{k+1} beobachteter Wert (Evidenz)

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

Message-Passing mit Evidenz

- Ebenso für $\mu_\alpha(x_k)$

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

- Jetzt gilt

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

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

Beispiel: Markov Modelle

- Beispiel für Inferenz auf linearer Kette: Markov Modelle

- Markov Modelle: einfache Modelle für dynamische probabilistische Prozesse
  - Prozess, der verschiede Zustände annehmen kann
  - Zufallsvariable X_t repräsentiert den Zustand zur Zeit t
  - Diskrete Zeitpunkte $t=1,\ldots,T$

- Beispiel: Wetter
  - Zufallsvariable X_t = Wetter am Tag t
  - Zwei mögliche Zustände, Regen und Sonne

Beispiel: Markov Modelle

- Modellierung:
  - Prozess wird in einem zufällig gewählten Zustand gestartet:

    Verteilung über Startzustände $p(x_1)$

  - In jedem Schritt geht der Prozess in einen neuen Zustand über, abhängig nur vom gegenwärtigen Zustand (vereinfachende Annahme!)

    Verteilung über nächsten Zustand $p(x_{t+1} \mid x_t)$

- Unabhängigkeitsannahme

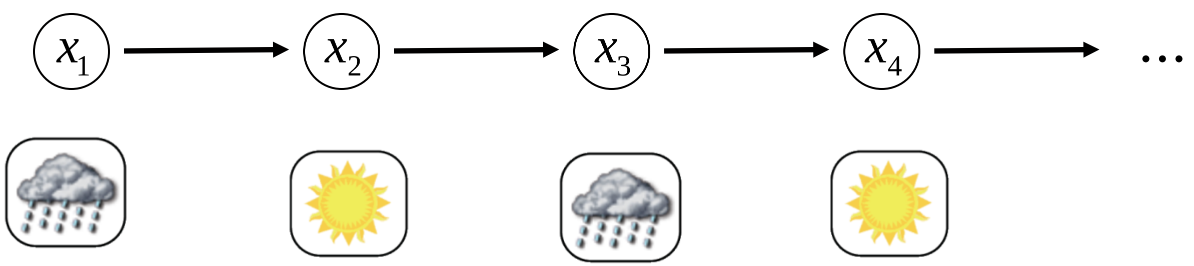
  $\forall t: \; p(x_{t+1} \mid x_1, \ldots, x_t) = p(x_{t+1} \mid x_t)$ "Markov" Eigenschaft

- Übergangswahrscheinlichkeiten hängen nicht von t ab:

  $\forall t: \; p(x_{t+1} \mid x_t) = p(x_t \mid x_{t-1})$ "Stationärer" Prozess

Beispiel: Markov Modelle

- Beispiel Markov Modell:
  - Zustand x_t = Wetter am Tag t
  - Zwei mögliche Zustände, Regen und Sonne



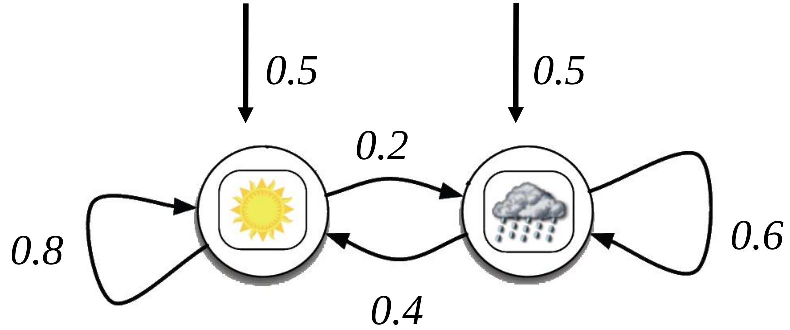
  - Verteilungen

$$p(x_1 = s) = 0.5$$
$$p(x_1 = r) = 0.5$$

$$p(x_t = s \mid x_{t-1} = s) = 0.8$$
$$p(x_t = r \mid x_{t-1} = s) = 0.2$$
$$p(x_t = s \mid x_{t-1} = r) = 0.4$$
$$p(x_t = r \mid x_{t-1} = r) = 0.6$$

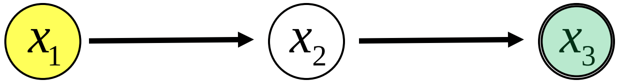
Beispiel: Markov Modelle

- Markov Modelle entsprechen probabilistischen endlichen Automaten



- Beispiel Inferenzproblem:
  - Wie ist das Wetter übermorgen, gegeben dass es heute regnet?

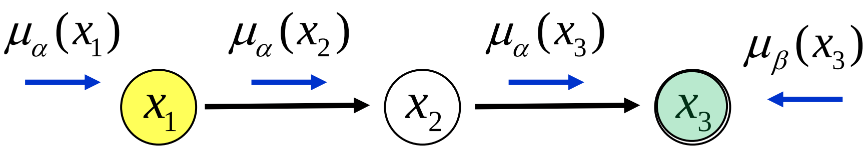
$$p(x_3 \mid x_1 = r) = ?$$



Beispiel: Markov Modelle

- Berechnung mit Message-Passing Algorithmus

- Nachrichten: $\mu_\alpha(x_1), \mu_\alpha(x_2), \mu_\alpha(x_3); \quad \mu_\beta(x_3)$



- Initialisierung $\mu_\alpha(x_1)$:

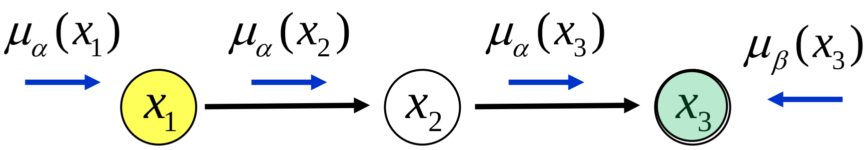
$$\mu_\alpha(x_1 = s) = 1$$

$$\mu_\alpha(x_1 = r) = 1$$

Beispiel: Markov Modelle

- Berechnung mit Message-Passing Algorithmus

- Nachrichten: $\mu_\alpha(x_1), \mu_\alpha(x_2), \mu_\alpha(x_3); \quad \mu_\beta(x_3)$



- Berechnung $\mu_\alpha(x_2)$: Linker Knoten $x_1 = r$ beobachtet

$$\mu_\alpha(x_2) = \Psi_{1,2}(x_1, x_2)\mu_\alpha(x_1) = p(x_1)p(x_2 \mid x_1)\mu_\alpha(x_1)$$

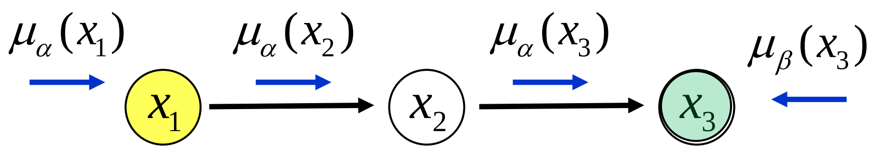
$$\mu_\alpha(x_2 = s) = p(x_1 = r)p(x_2 = s \mid x_1 = r)\mu_\alpha(x_1 = r) = 0.5 \cdot 0.4 \cdot 1 = 0.2$$

$$\mu_\alpha(x_2 = r) = p(x_1 = r)p(x_2 = r \mid x_1 = r)\mu_\alpha(x_1 = r) = 0.5 \cdot 0.6 \cdot 1 = 0.3$$

Beispiel: Markov Modelle

- Berechnung mit Message-Passing Algorithmus

- Nachrichten: $\mu_\alpha(x_1), \mu_\alpha(x_2), \mu_\alpha(x_3); \quad \mu_\beta(x_3)$



- Berechnung $\mu_\alpha(x_3)$: Linker Knoten x_2 nicht beobachtet.

$$\mu_\alpha(x_3) = \sum_{x_2} \Psi_{2,3}(x_2, x_3)\mu_\alpha(x_2) = \sum_{x_2} p(x_3 \mid x_2)\mu_\alpha(x_2)$$

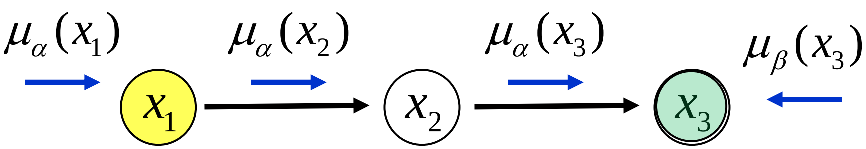
$$\mu_\alpha(x_3 = s) = p(x_3 = s \mid x_2 = s)\mu_\alpha(x_2 = s) + p(x_3 = s \mid x_2 = r)\mu_\alpha(x_2 = r)$$
$$= 0.8 \cdot 0.2 + 0.4 \cdot 0.3 = 0.28$$

$$\mu_\alpha(x_3 = r) = p(x_3 = r \mid x_2 = s)\mu_\alpha(x_2 = s) + p(x_3 = r \mid x_2 = r)\mu_\alpha(x_2 = r)$$
$$= 0.2 \cdot 0.2 + 0.6 \cdot 0.3 = 0.22$$

Beispiel: Markov Modelle

- Berechnung mit Message-Passing Algorithmus

- Nachrichten: $\mu_\alpha(x_1), \mu_\alpha(x_2), \mu_\alpha(x_3); \quad \mu_\beta(x_3)$



- Initialisierung:

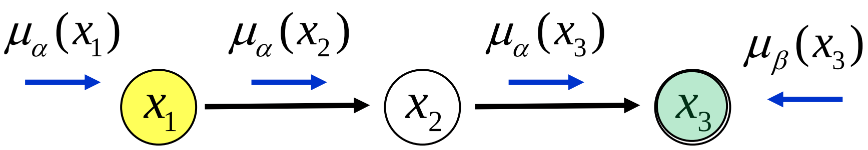
$$\mu_\beta(x_3 = s) = 1$$

$$\mu_\beta(x_3 = r) = 1$$

Beispiel: Markov Modelle

- Berechnung mit Message-Passing Algorithmus

- Nachrichten: $\mu_\alpha(x_1), \mu_\alpha(x_2), \mu_\alpha(x_3); \quad \mu_\beta(x_3)$



- Ergebnis: Nachrichten im Anfrageknoten multiplizieren

$$p(x_3 = s \mid x_1 = r) = \frac{1}{Z}\mu_\alpha(x_3 = s)\mu_\beta(x_3 = s) = \frac{1}{Z}0.28$$

$$Z = 0.28 + 0.22 = 0.5$$

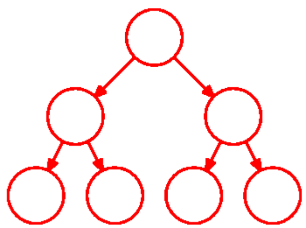
$$p(x_3 = r \mid x_1 = r) = \frac{1}{Z}\mu_\alpha(x_3 = r)\mu_\beta(x_3 = r) = \frac{1}{Z}0.22$$

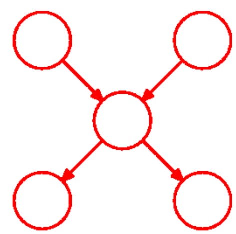
$$p(x_3 = s \mid x_1 = r) = 0.56$$
$$p(x_3 = r \mid x_1 = r) = 0.44$$

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



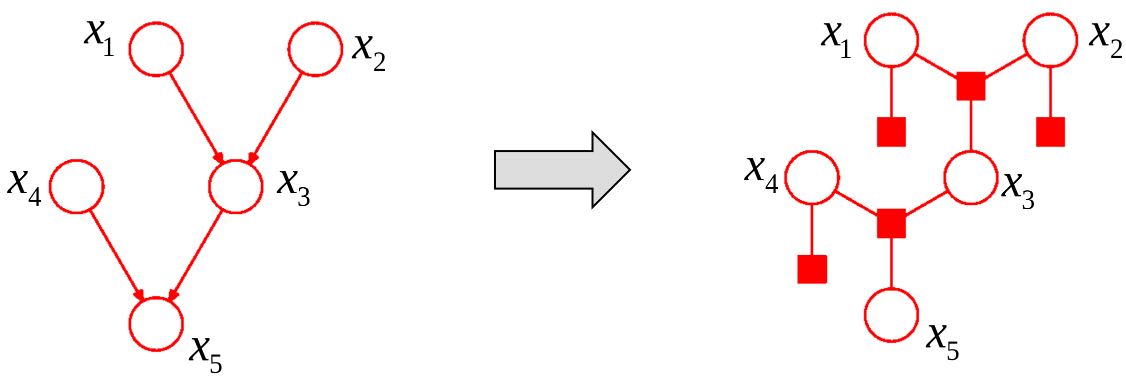


Polytree

Inferenz in Allgemeinen Graphen

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

Ursprünglicher Graph / Faktor-Graph



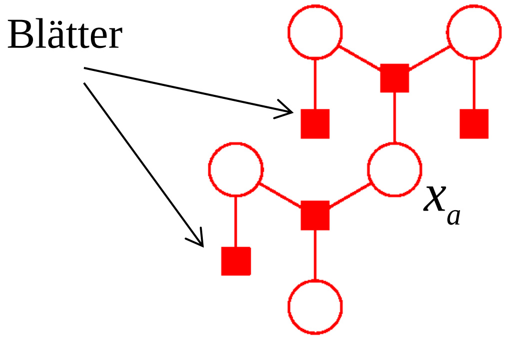
Gemeinsame Verteilung

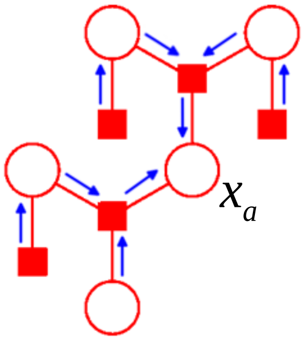
$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)\,p(x_2)\,p(x_3 \mid x_1, x_2)\,p(x_4)\,\underbrace{p(x_5 \mid x_3, x_4)}_{Faktor}$$

■ 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

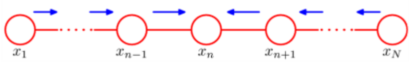
Inferenz in Allgemeinen Graphen (Skizze)

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



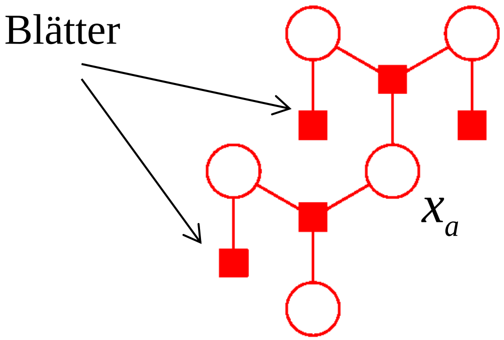


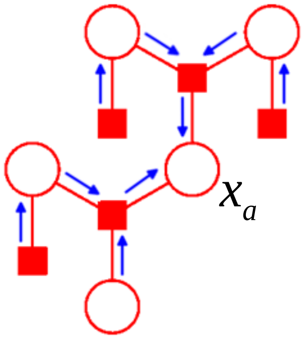
Spezialfall lineare Kette



- 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)

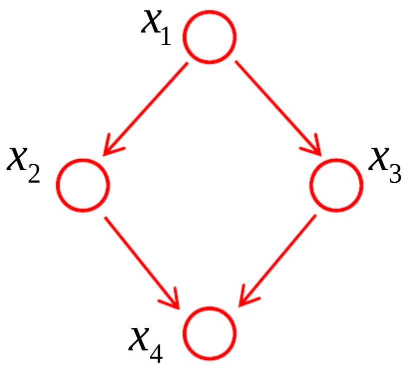




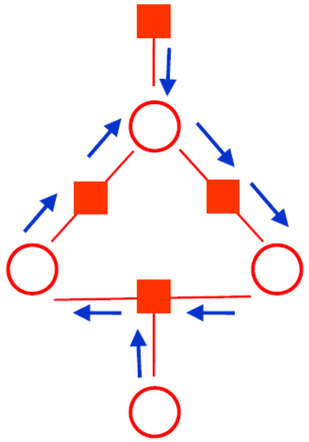
- Nachrichten werden „verschmolzen", dabei müssen wir über mehrere Variablen summieren
- Grundidee dieselbe wie bei Inferenz auf der linearen Kette: geschicktes Aussummieren
- Details im Bishop-Textbuch („Sum-Product" Algorithmus)

Inferenz in Allgemeinen Graphen

- Inferenz in Graphen, die keine Polytrees sind?
- Iteratives Message-Passing Schema, wegen Zyklen im Graph nicht exakt (=approximativer Algorithmus)



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



„Loopy Belief Propagation"

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

Überblick

- Graphische Modelle: Syntax und Semantik
- Graphische Modelle im Maschinellen Lernen
- Exakte Inferenz in Graphischen Modellen
- Approximative Inferenz in Graphischen Modellen
- Sequenzmodelle