# Künstliche Intelligenz

## Hidden Markov Models

Dr.-Ing. Stefan Lüdtke

Universität Leipzig

Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)

# Motivation

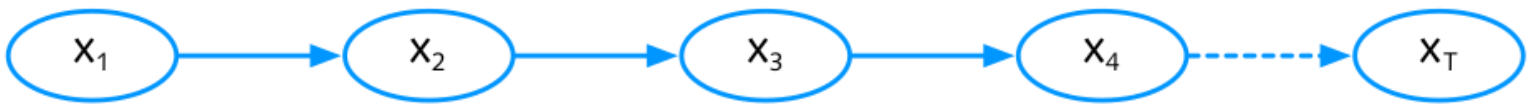
- Die Welt verändert sich über die Zeit
  - Wahrscheinlichkeitsverteilungen müssen für jeden Zeitschritt angepasst werden
  - Aber: Systemzustand ist abhängig von der Vergangenheit
- Generelle Idee
  - Großes Bayes'sches Netz, in dem es Knoten (= Zufallsvariablen) für jeden Zeitpunkt gibt
  - Spezialisierte Inferenzalgorithmen, die die besondere Netzwerkstruktur ausnutzen

# Sequentielle Prozesse

- Gegeben: Zustandsraum $X$, zu jedem Zeitpunkt $t$ hat das System einen Zustand $x_t \in X$
- Auch gegeben: Beobachtungsraum $Y$, zu jedem Zeitpunkt $t$ machen wir eine Beobachtung $y_t \in Y$
- Was ist eine sinnvolle Struktur des Bayes'sches Netzes? Wie hängen Zufallsvariablen voneinander ab?

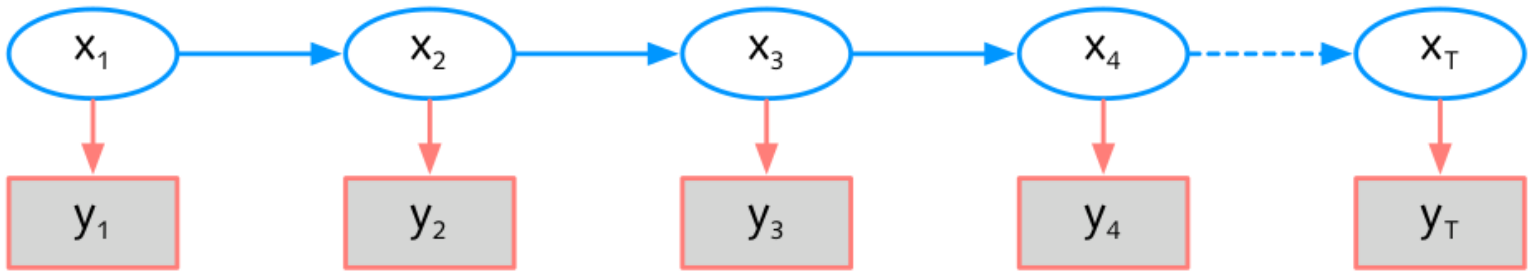
# Markov-Ketten

- Annahme: Zustände bilden eine *Markov-Kette ester Ordnung*: Zustand $X_t$ hängt nur vom Zustand $X_{t-1}$ ab
- Außerdem nehmen wir an, dass sich die Art der Abhängigkeit im Verlauf der Zeit nicht ändert, d.h. diese kann über eine bedingte Wahrscheinlichkeitsverteilung $P(X_t \mid X_{t-1})$ beschrieben werden
- Wir nennen $P(X_t \mid X_{t-1})$ *Transitionsmodell*



# Observationsmodell

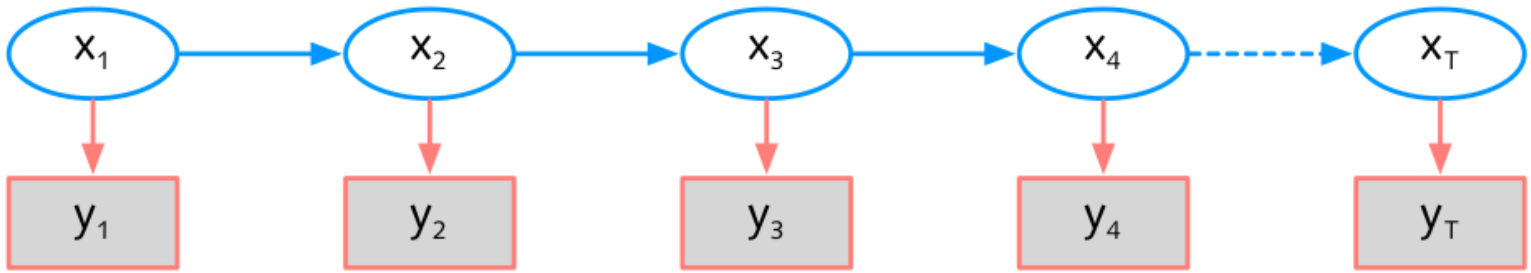
- Annahme: Beobachtung $Y_t$ hängt nur von Zustand $X_t$ ab
- Auch diese Abhängigkeit ändert sich nicht im Verlauf der Zeit, d.h. Repräsentation als bedingte Verteilung $P(Y_t \mid X_t)$
- Nennen $P(Y_t \mid X_t)$ *Observationsmodell*



# Hidden Markov Model

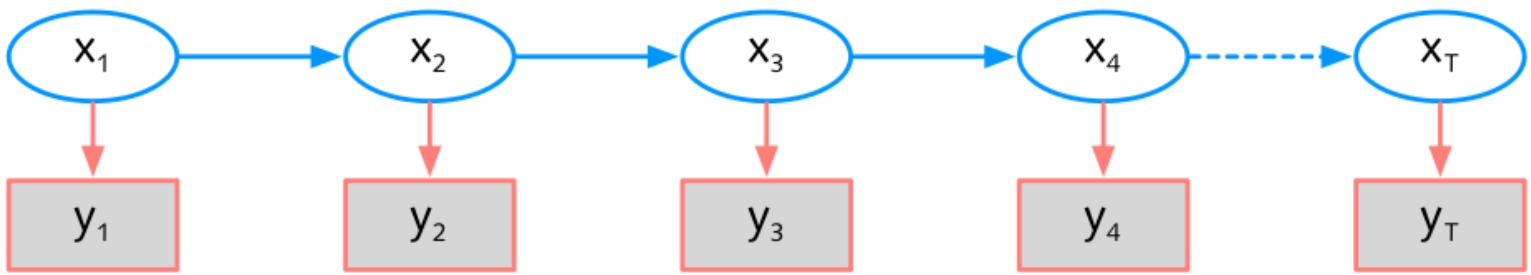
Ein *Hidden Markov Model* ist definiert durch

- Transitionsmodell $P(X_t \mid X_{t-1})$
- Observationsmodell $P(Y_t \mid X_t)$
- Initialverteilung (A-Priori-Verteilung) $P(X_1)$



# Inferenz in HMMs

- Query: Berechne $P(X_t \mid y_1, \ldots, y_t) = P(X_t \mid y_{1:t})$ für jedes $t$
  - Diese Inferenzaufgabe nennt sich *Filtering*, später werden wir noch andere typische Aufgaben kennenlernen
- Im Prinzip mit Standard-Inferenzalgorithmen für Bayes'sche Netze, aber das ist unnötig ineffizient
- Kann stattdessen die besondere Netzwerkstruktur ausnutzen, um effiziente, rekursive Formulierung zu erhalten



# Inferenz in HMMs: Filtering

- Angenommen, wir haben $P(X_{t-1} \mid y_{1:t-1})$ (d.h. Filtering-Query für $t-1$) schon ausgerechnet
- Gehen jetzt in 2 Schritten vor:
  - Berechne *Vorhersage* $P(X_t \mid y_{1:t-1})$ (d.h. ohne Einbeziehen von $y_t$)
  - Berechne *Korrektur* $P(X_t \mid y_{1:t-1}, y_t)$

# Filtering: Vorhersage

$$
\begin{aligned}
P(x_t \mid y_{1:t-1}) &= \sum_{x_{t-1}} P(x_t, x_{t-1} \mid y_{1:t-1}) \\
&= \sum_{x_{t-1}} P(x_t \mid x_{t-1} y_{1:t-1})\, P(x_{t-1} \mid y_{1:t-1}) \\
&= \sum_{x_{t-1}} P(x_t \mid x_{t-1})\, P(x_{t-1} \mid y_{1:t-1})
\end{aligned}
$$

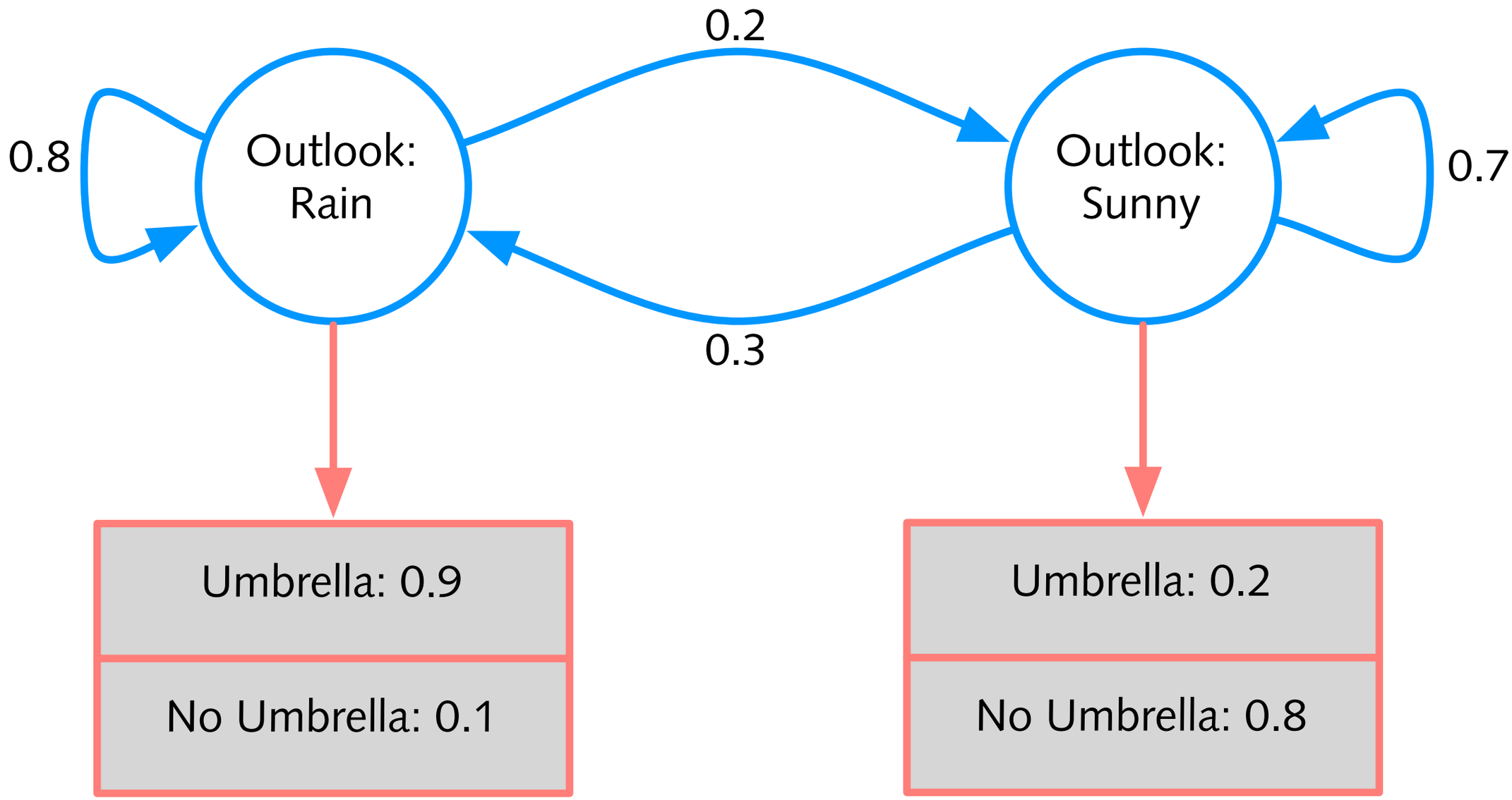
# Filtering: Korrektur

$$P(x_t \mid y_{1:t-1}, y_t) = \frac{P(y_t \mid x_t, y_{1:t-1})\, P(x_t \mid y_{1:t-1})}{P(y_t \mid y_{1:t-1})}$$
$$= \frac{P(y_t \mid x_t)\, P(x_t \mid y_{1:t-1})}{P(y_t)}$$

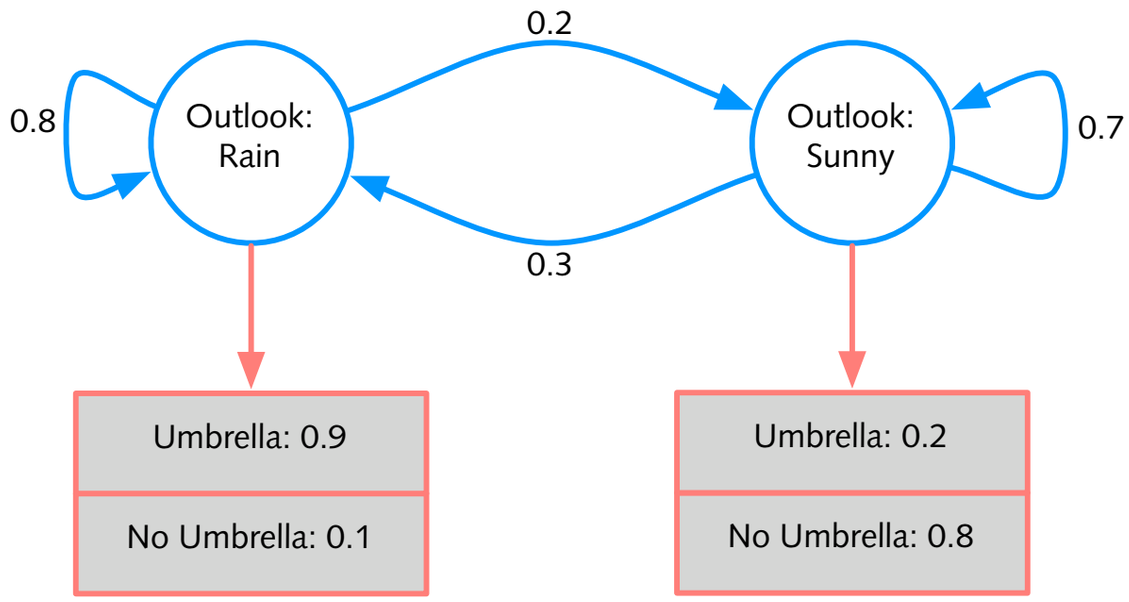
Der Term $P(y_t) = \sum_{x_t} P(y_t \mid x_t)\, P(x_t \mid y_{1:t-1})$ ist ein Normalisierungsfaktor und ergibt sich automatisch, wenn wir den Zähler für alle $x_t$ berechnen

# 1 Sequentielle Prozesse

Hier ein einfaches Modell:



# Sequentielle Prozesse

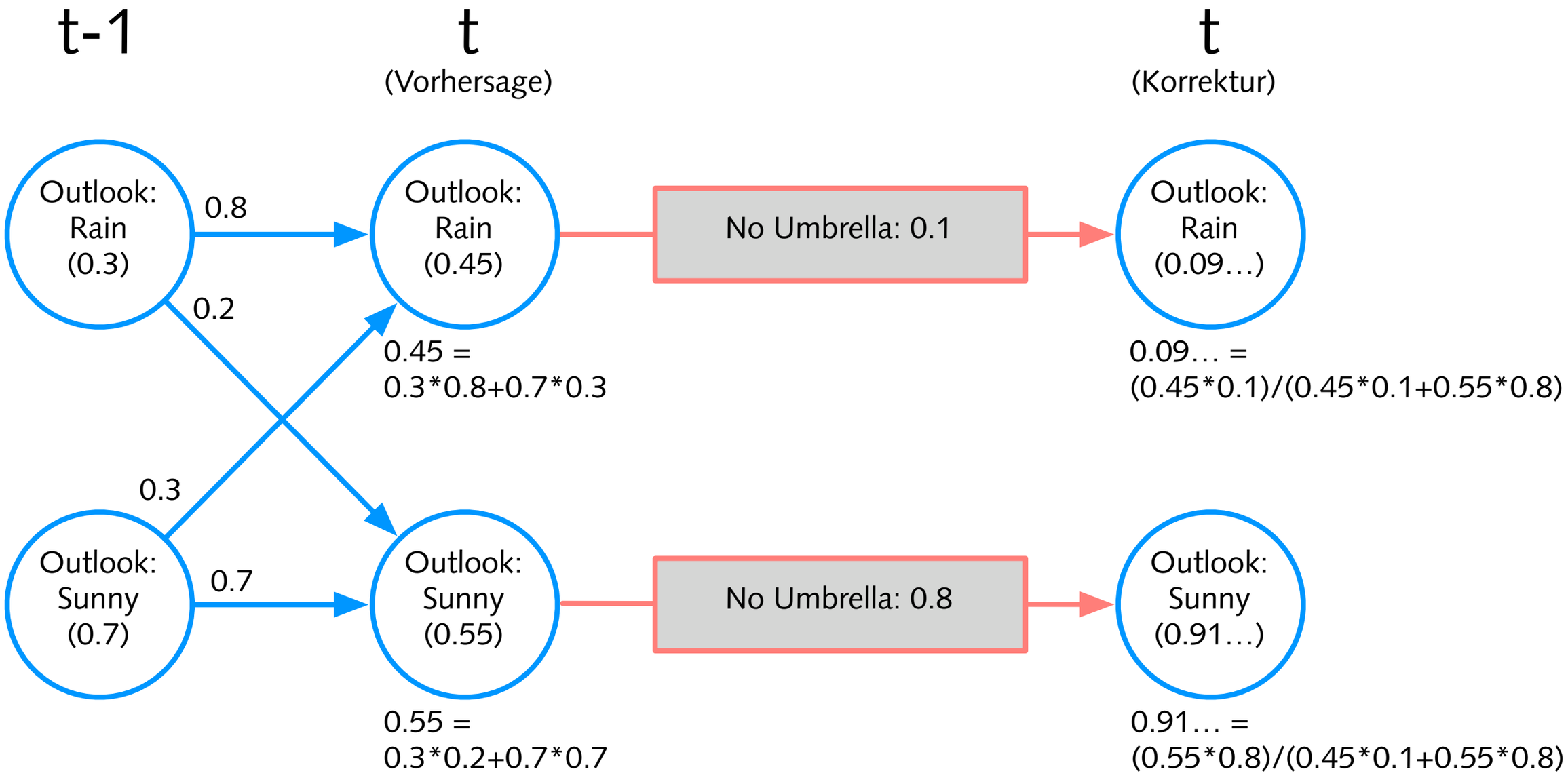



$$\mathbf{A} = \begin{array}{c|cc} x_t & x_{t-1} & \\ & Rain & Sunny \\ \hline Rain & 0.8 & 0.3 \\ Sunny & 0.2 & 0.7 \end{array} \qquad \mathbf{O} = \begin{array}{c|cc} x_t & y_t & \\ & NoUmbr. & Umbr. \\ \hline Rain & 0.1 & 0.9 \\ Sunny & 0.8 & 0.2 \end{array}$$

Nicht im Diagramm:

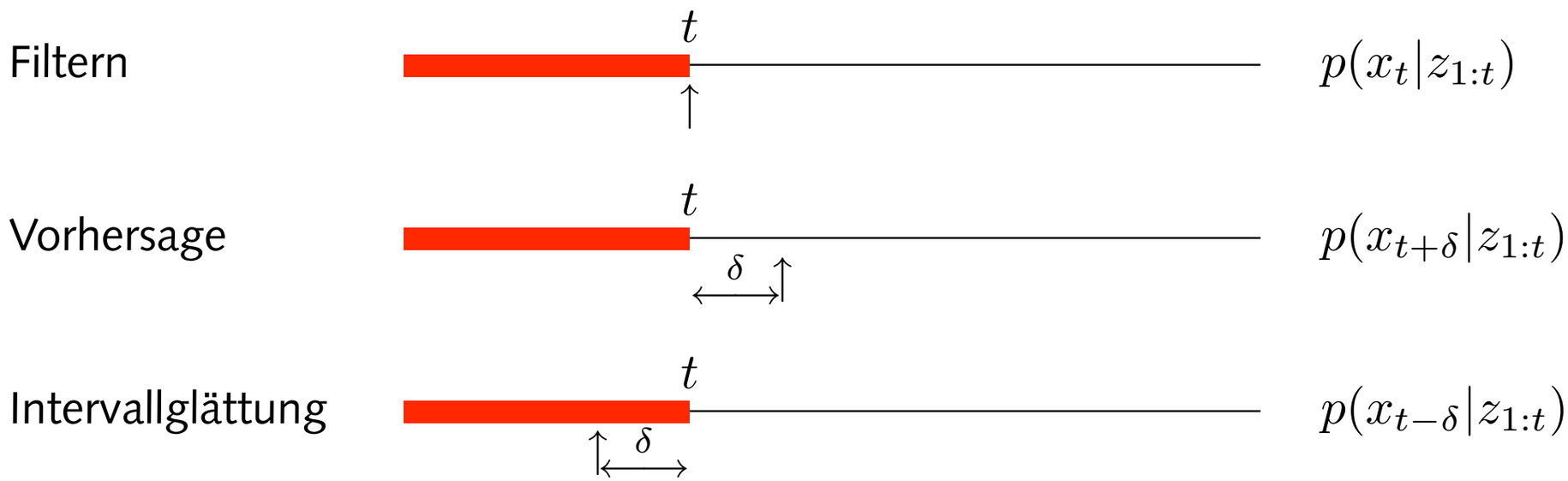
$$p(x_1) = \begin{array}{c|c} Rain & 0.3 \\ Sunny & 0.7 \end{array}$$

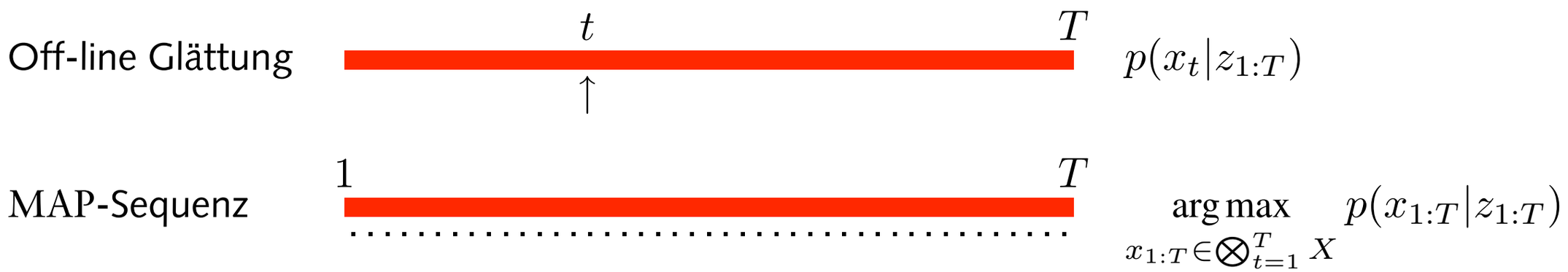
# 2 Inferenz in sequentiellen Prozessen



# 2 Inferenz in sequentiellen Prozessen

## Filtern, Vorhersage, Glätten und MAP-Sequenz im Vergleich



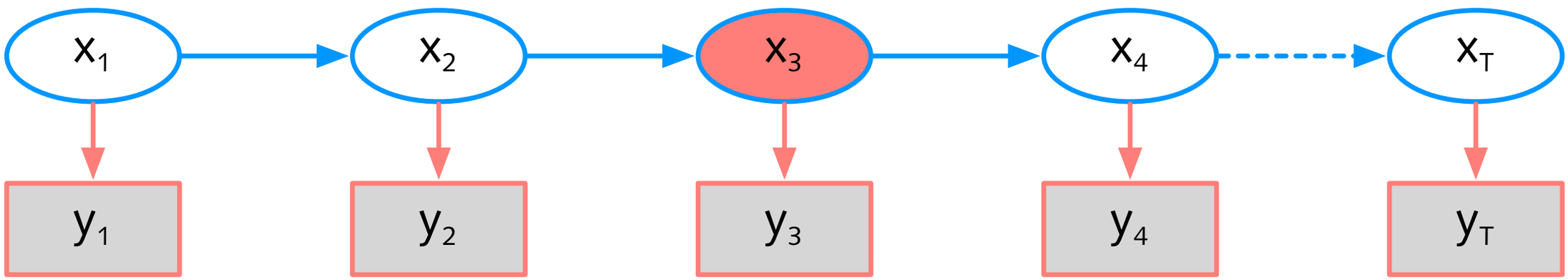



# 2 Inferenz in sequentiellen Prozessen

Die Bestimmung von $p(x_t \mid y_{1:t})$ wird auch als *Filterung* bezeichnet.

Ein weiteres Verfahren von Interesse nennt sich *Glättung*.

D.h., gesucht: $p(x_t \mid y_{1:T})$




Die Idee ist, für die Schätzung zum Zeitpunkt $t$ (hier: $t = 3$) die *gesamte* Sequenz von Beobachtungen $y_{1:T}$ zu verwenden.

Analog zum Lesen eines Kriminalromans: Die Vermutung über den Täter auf Seite 1 ist oft eine andere, als die nach dem Lesen der letzten Seite.

# 2 Inferenz in sequentiellen Prozessen

$$
\begin{aligned}
p(x_t \mid y_{1:T}) &= \sum_{x_{t+1}} p(x_t, x_{t+1} \mid y_{1:T}) && \text{Marginalisierung} \\
&= \sum_{x_{t+1}} p(x_t \mid x_{t+1}, y_{1:T}) p(x_{t+1} \mid y_{1:T}) && \text{Kettenregel} \\
&= \sum_{x_{t+1}} p(x_t \mid x_{t+1}, y_{1:t}) p_t(x_{t+1} \mid y_{1:T}) && \text{Unabh. in HMM} \\
&= \sum_{x_{t+1}} \frac{p(x_{t+1} \mid x_t, y_{1:t})\, p(x_t \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})} p(x_{t+1} \mid y_{1:T}) && \text{Bayes} \\
&= \sum_{x_{t+1}} \frac{p(x_{t+1} \mid x_t)\, p(x_t \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})} p(x_{t+1} \mid y_{1:T}) && \text{Unabh. in HMM} \\
&= p(x_t \mid y_{1:t}) \sum_{x_{t+1}} p(x_{t+1} \mid x_t) \frac{p(x_{t+1} \mid y_{1:T})}{p(x_{t+1} \mid y_{1:t})}
\end{aligned}
$$

Die Glättung arbeitet von hinten nach vorne und nutzt dabei Vorhersage und Ergebnis der Filterung.

# Sequentielle Prozesse

Annahme:

- Systemmatrix $\mathbf{A}$ gegeben
- Observierungsmatrix $\mathbf{O}$ gegeben
- Zustandswahrscheinlichkeiten zum Zeitpunkt $t-1$ als Vektor $\mathbf{p}_{t-1}$ gegeben
- Beobachtung $y_t$ gegeben.

Dann erhalten wir

- den Vektor der Vorhersagewahrscheinlichkeiten $\mathbf{p}_{t|t-1}$ durch:

$$\mathbf{p}_{t|t-1} = \mathbf{A}\mathbf{p}_{t-1}$$

- und die Zustandswahrscheinlichkeiten für den Zeitpunkt $t$ durch:

$$\mathbf{p}_t = \frac{1}{\|\tilde{\mathbf{p}}_t\|_1}\tilde{\mathbf{p}}_t \qquad \text{wobei } \tilde{\mathbf{p}}_t = \mathbf{p}_{t|t-1} \circ \mathbf{O}_{[\cdot,y_t]}$$

# Sequentielle Prozesse

Dabei ist:

- $\mathbf{a} \circ \mathbf{b}$ das Hadamard-Produkt (auch Schur-Produkt genannt)
- $\mathbf{O}_{[\cdot, y_t]}$ der zu $y_t$ korrespondierende Spaltenvektor der Matrix $\mathbf{O}$.
- $\|\mathbf{a}\|_1$ die $L_1$-Norm des Vektors $\mathbf{a}$, also einfach die Summe der Absolutbeträge der Vektorelemente.

In R:

```
fstep <- function(A,O,yt,pt1) {
  tilde.pt <- O[,yt] * (A %*% pt1)
  tilde.pt / sum(tilde.pt)
}
```

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")
  - Angenommen, nur Sequenzen $y_1, \ldots, y_T$ vorhanden (*unüberwachtes Lernen*): Expectation Maximization-Algorithmus ($\rightarrow$ *Statistische Signalverarbeitung und Inferenz*)

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")
  - Angenommen, nur Sequenzen $y_1, \ldots, y_T$ vorhanden (*unüberwachtes Lernen*): Expectation Maximization-Algorithmus ($\rightarrow$ *Statistische Signalverarbeitung und Inferenz*)
- Inferenzalgorithmen (mindestens) linear in $|X|$ – was, wenn $|X|$ sehr groß ist?

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")
  - Angenommen, nur Sequenzen $y_1, \ldots, y_T$ vorhanden (*unüberwachtes Lernen*): Expectation Maximization-Algorithmus ($\rightarrow$ *Statistische Signalverarbeitung und Inferenz*)
- Inferenzalgorithmen (mindestens) linear in $|X|$ – was, wenn $|X|$ sehr groß ist?
  - Approximative Inferenzalgorithmen, z.B. Sampling-basiert (Partikelfilter)

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")
  - Angenommen, nur Sequenzen $y_1, \ldots, y_T$ vorhanden (*unüberwachtes Lernen*): Expectation Maximization-Algorithmus ($\rightarrow$ *Statistische Signalverarbeitung und Inferenz*)
- Inferenzalgorithmen (mindestens) linear in $|X|$ – was, wenn $|X|$ sehr groß ist?
  - Approximative Inferenzalgorithmen, z.B. Sampling-basiert (Partikelfilter)
- Variante für *kontinuierlichen* Zustandsraum $X$ (z.B. 2D-Position im Raum): Kalman-Filter (aber: Normalverteilungsannahmen)

# Weitere Aspekte

- Woher kommen die Verteilungen $P(X_t \mid X_{t-1})$, $P(Y_t \mid X_t)$ und $P(X_1)$? Ideen?
  - Explizite Konstruktion aus Domänenwissen
  - Angenommen, Sequenzen $x_1, \ldots, x_T$ und $y_1, \ldots, y_T$ vorhanden (sog. *überwachtes Lernen*): Relative Häufigkeiten ("Auszählen")
  - Angenommen, nur Sequenzen $y_1, \ldots, y_T$ vorhanden (*unüberwachtes Lernen*): Expectation Maximization-Algorithmus ($\rightarrow$ *Statistische Signalverarbeitung und Inferenz*)
- Inferenzalgorithmen (mindestens) linear in $|X|$ – was, wenn $|X|$ sehr groß ist?
  - Approximative Inferenzalgorithmen, z.B. Sampling-basiert (Partikelfilter)
- Variante für *kontinuierlichen* Zustandsraum $X$ (z.B. 2D-Position im Raum): Kalman-Filter (aber: Normalverteilungsannahmen)
- Variante für Nicht-atomaren Zustandsraum: Dynamische Bayes'sche Netze ($P(\mathbf{X}_t \mid y_{1:t})$ wird als Bayes'sches Netz repräsentiert)

# HMM Anwendungen

- Verarbeitung natürlicher Sprache, z.B. Part-Of-Speech-Tagging
- Audiodatenvararbeitung: Spracherkennung
- Bioinformatik: Analyse von Proteinsequenzen
- Sensorbasierte Aktivitätserkennung

# Zusammenfassung

- Sequentielle Modelle sind ein wichtiger Spezialfall Bayes'scher Netze
- Markov- und Stationaritäts-Annahmen, sodass ein Modell durch
  - Transitionsmodell $P(X_t \mid X_{t-1})$
  - Observationsmodell $P(Y_t \mid X_t)$
  - Initialverteilung $P(X_1)$

  vollständig beschrieben ist
- Wenn $X$ eine endliche Menge ist, nennen wir das entstehende Modell *Hidden Markov Model*
- Das Filtering-, Smoothing- und MAP-Problem kann durch Ausnutzung der Modelleigenschaften effizient gelöst werden