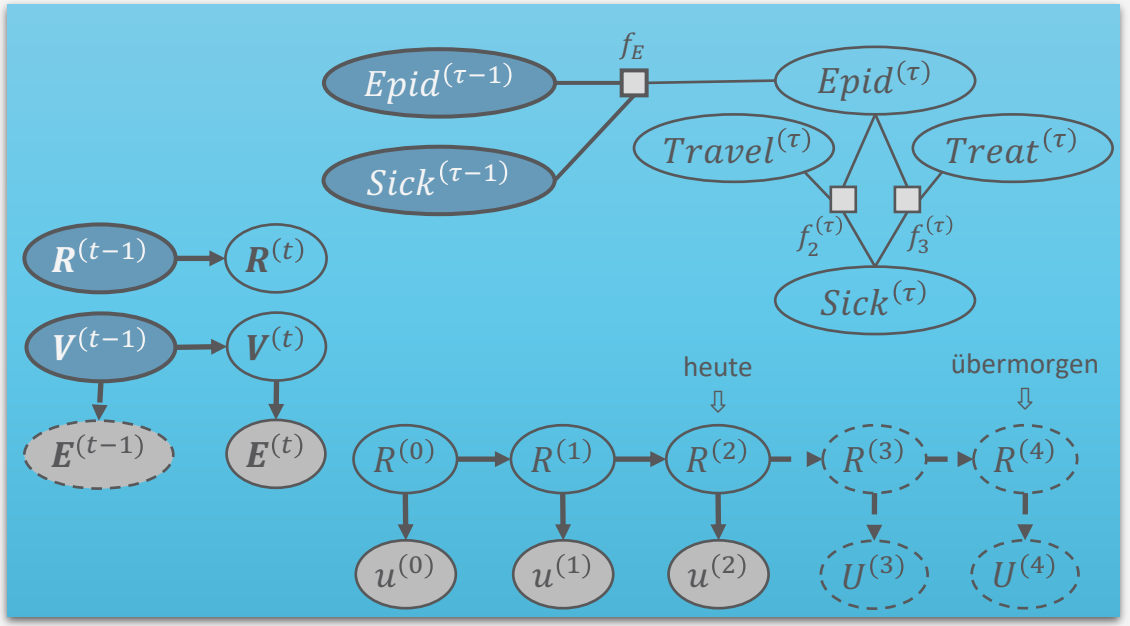
WWU
MÜNSTER

Sequentielle PGMs und Inferenz

Einführung in die Künstliche Intelligenz



wissen.leben

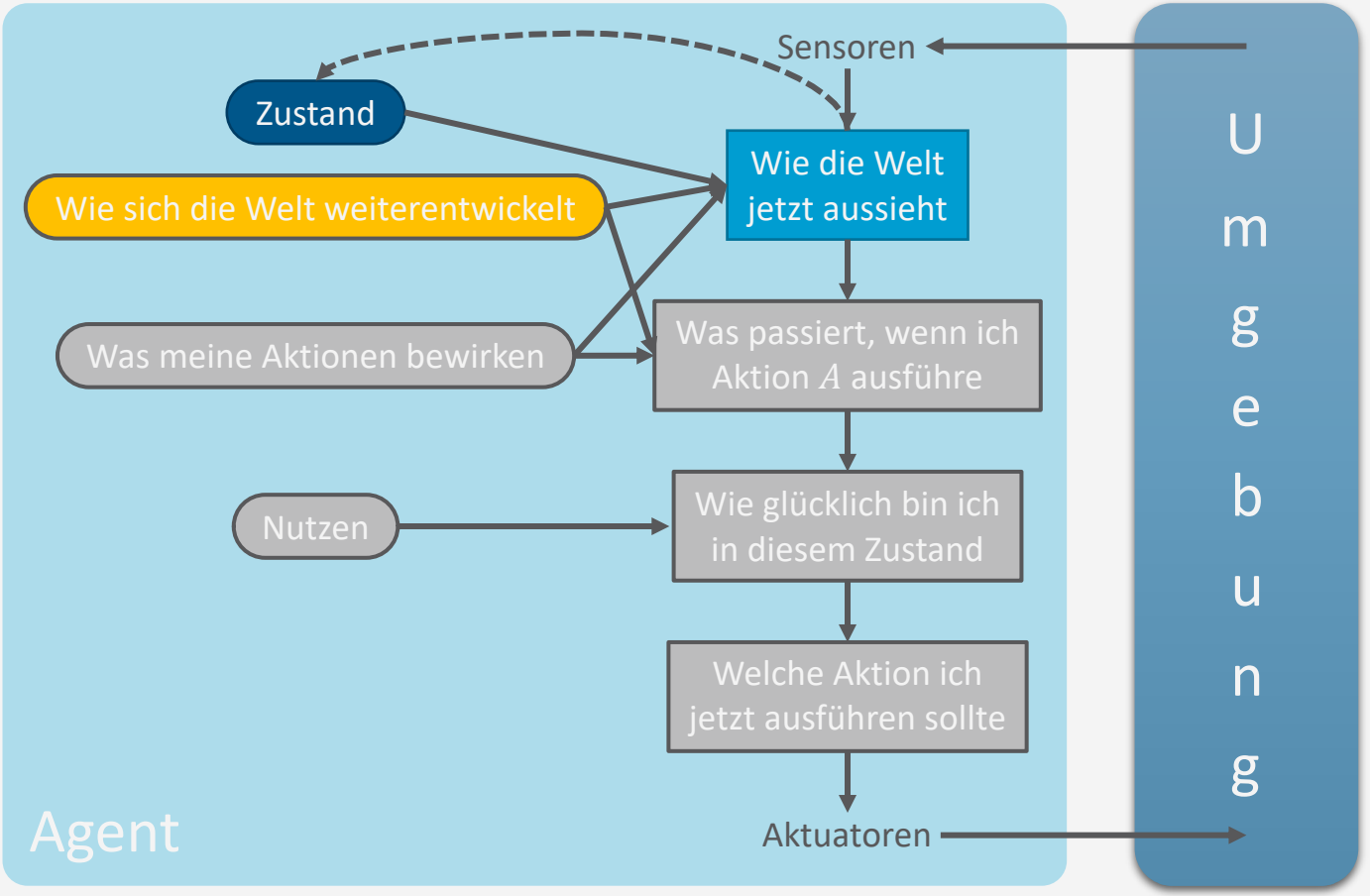
Tanya Braun
Arbeitsgruppe Data Science, Institut für Informatik

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*

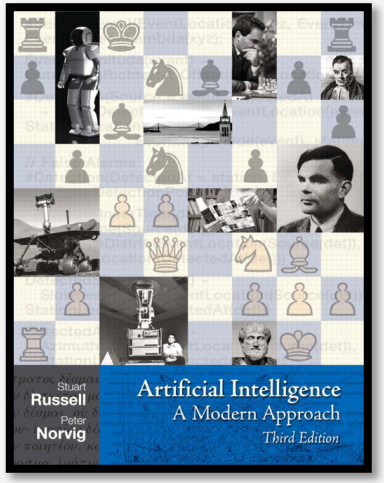
- Nachfolgende Themen der Vorlesung
  2. Episodische PGMs
  3. Exakte Inferenz in episodischen PGMs
  4. Approximative Inferenz in episodischen PGMs
  5. Lernalgorithmen für episodische PGMs
  6. Sequentielle PGMs und Inferenz
  7. Entscheidungstheoretische PGMs

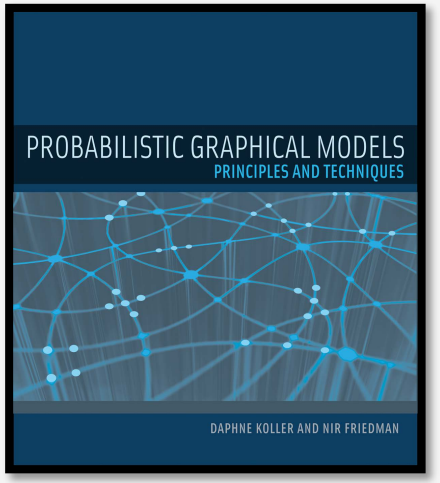


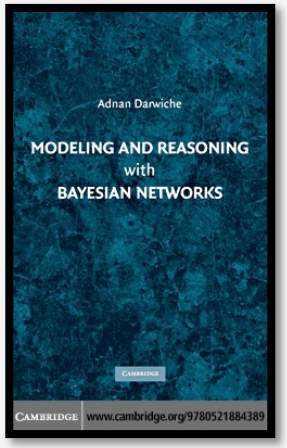
Literaturhinweise

Inhalte dieses Themenblocks werden in den folgenden Kapiteln der Vorlesungsbücher behandelt

- AIMA(de)
  - Kap. 15.2: Inferenz in temporalen Modellen
  - Kap. 15.3: Hidden-Markov-Modelle
  - Kap. 15.5: Dynamische Bayes Netze
- PGM
  - Kap. 6.2: Zeitliche Modelle
  - Kap. 15.2: Inferenz in zeitlichen Modellen







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

Bemerkung zu Namenskonventionen

- Gebräuchliche Namen für die gleiche Sache in PGMs
  - Dynamisch (*dynamic*)
    - ABER: *stationär* bezogen darauf, wie sich ein System verändert von einem Zustand zum nächsten
  - Zeitlich (*temporal*)
    - Änderungen zwischen Zuständen impliziert oft, dass die Zeit fortschreitet, i.e., eine zeitliche Zustandssequenz
      - Implizite Richtung der Kanten in Richtung der Zukunft
      - Vereinfachende Annahme: Diskrete Zeitschritte indiziert durch Integer (t)
  - Sequentiell (*sequential*)
    - Verallgemeinerung der Bezeichnung „zeitlich“, da eine Sequenz nicht nur durch fortschreitende Zeit, sondern auch durch etwas anderes entstehen kann
      - Beispiele: Räumliche Bewegung, Sequenz von Worten im Text
      - Meistens schreitet implizit dabei auch die Zeit voran

Überblick: 6. Sequentielle PGMs und Inferenz

A. Sequentielle PGMs

- Templates, dynamische BNs, dynamische Faktormodelle, Hidden Markov Modelle; Semantik
- Inferenzaufgaben: Wahrscheinlichkeitsanfragen (Filtering, Prediction, Hindsight), Zustandsanfragen (MPE, MAP)

B. Sequentielle Inferenz

- Naïve Inferenz mittels Ausrollen, Interface Algorithmus, Komplexität, Approximationen

C. Spezialfall Hidden-Markov-Modelle

- Viterbi-Algorithmus für MPEs
- Anfragebeantwortung durch Matrixoperationen
- Baum-Welch-Algorithmus zum Lernen

Zustand eines Systems über die Zeit

- Menge von Zufallsvariablen, um Zustand eines Systems zu beschreiben
  - $\boldsymbol{R} = \{R_1, \dots, R_n\}$
  - Template-Variablen (Schablone, Vorlage)
- Zustand des Systems zu einem Zeitpunkt t:
  - Charakterisiert durch Belegungen von $\boldsymbol{R}$ zum Zeitpunkt t
    - Instanziierung der Template-Variablen mit t, notiert als $\boldsymbol{R}^{(t)} = \left\{R_1^{(t)}, \dots, R_n^{(t)}\right\}$
    - Belegung mit Werten aus jeweils $\mathrm{Val}(R_i)$: $\boldsymbol{r}^{(t)} = \left\{r_1^{(t)}, \dots, r_n^{(t)}\right\}, r_i^{(t)} \in \mathrm{Val}(R_i)$
- Zustand des Systems über ein diskretes Intervall $[t_1, t_2], t_1 < t_2$
  - Menge der Variablen: $\boldsymbol{R}^{(t_1:t_2)} = \left\{\boldsymbol{R}^{(t)} \middle| t \in [t_1, t_2]\right\}$
  - Zustands-„Sequenz“: $\boldsymbol{r}^{(t_1:t_2)} = \left\{\boldsymbol{r}^{(t)} \middle| t \in [t_1, t_2]\right\}$
    - Als Sequenz: $\boldsymbol{r}^{(t_1:t_2)} = \left(\boldsymbol{r}^{(t)} \middle| t \in [t_1, t_2]\right)$

Zustand eines Systems über die Zeit

- Gegeben eine Menge von Template-Variablen $\boldsymbol{R} = \{R_1, \dots, R_n\}$ und ein Endpunkt T
  - Menge der Variablen: $\boldsymbol{R}^{(0:T)}$
- Eine mögliche Welt: Belegung $\boldsymbol{r}^{(0:T)}$ von $\boldsymbol{R}^{(0:T)}$ mit Wahrscheinlichkeit belegt
  - Genannt Trajektorie
- Vollständige gemeinsame Wahrscheinlichkeitsverteilung über alle möglichen Trajektorien

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{R}^{(0:T)}\right) = P\left(\boldsymbol{R}^{(0)}, \dots, \boldsymbol{R}^{(T)}\right) = P\left(R_1^{(0)}, \dots, R_n^{(0)}, \dots, R_1^{(T)}, \dots, R_n^{(T)}\right)$$

  - Kettenregel angewendet:

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{R}^{(0:T)}\right) = P\left(\boldsymbol{R}^{(0)}\right) \prod_{t=1}^{T} P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{(0:(t-1))}\right)$$

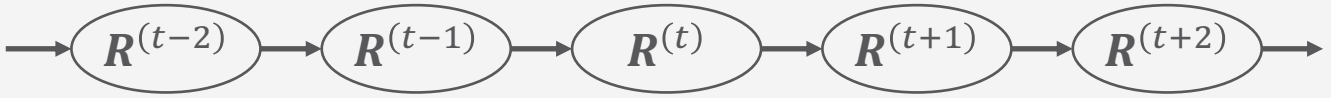
  - Sehr komplexe Verteilung ohne weitere vereinfachende Annahmen

Vereinfachende Annahme 1: Markov Annahme

- Markov Annahme: Nächster Zustand hängt nur vom jetzigen Zustand ab
  - Formal: $\left(\boldsymbol{R}^{(t+1)} \perp \boldsymbol{R}^{(0:(t-1))} \middle| \boldsymbol{R}^{(t)}\right)$
- Auswirkung auf $P\left(\boldsymbol{R}^{(0:T)}\right)$: Faktorisierung über t

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{R}^{(0:T)}\right) = P\left(\boldsymbol{R}^{(0)}\right) \prod_{t=1}^{T} P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{(t-1)}\right)$$

  - Auch *Markov Prozess erster Ordnung* genannt
  - Aufteilung des Modells in Zeitscheiben (*time slices*)
  - Graphische Darstellung:

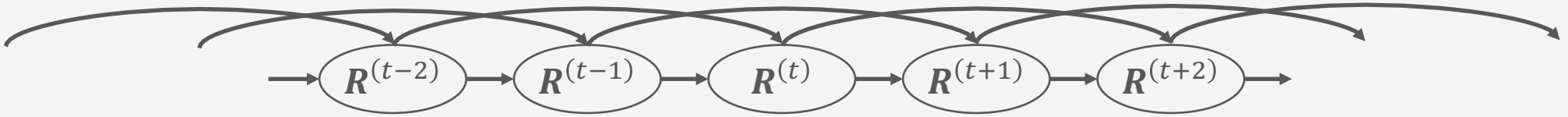


Vereinfachende Annahme 1: Markov Annahme

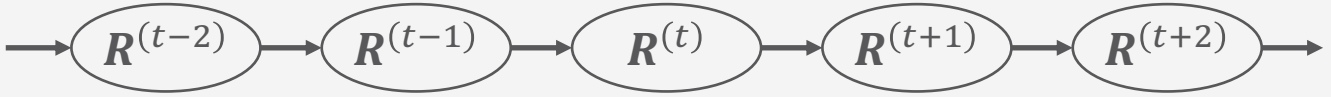
- Verallgemeinerung der Markov-Annahme: Markov-k
  - Der nächste Zustand hängt nur von den k vorherigen Zuständen ab

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{R}^{(0:T)}\right) = P\left(\boldsymbol{R}^{(0)}\right) \prod_{t=1}^{k} P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{(0:(t-1))}\right) \prod_{t=k+1}^{T} P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{((t-k):(t-1))}\right)$$

  - Graphische Darstellung für $k = 2$, ergibt *Markov Prozess zweiter Ordnung*



  - Markov-Annahme $\triangleq$ Markov-k mit $k = 1$



Vereinfachende Annahme 2: Stationäres System

- Gegeben eine Menge von Template-Variablen $\boldsymbol{R} = \{R_1, \dots, R_n\}$ und ein Endpunkt T
- Vollständige gemeinsame Wahrscheinlichkeitsverteilung über alle möglichen Trajektorien
  - Mit Markov Annahme: Faktorisierung über t

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{R}^{(0:T)}\right) = P\left(\boldsymbol{R}^{(0)}\right) \prod_{t=1}^{T} P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{(t-1)}\right)$$

- Annahme Stationarität: $P\left(\boldsymbol{R}^{(t)} \middle| \boldsymbol{R}^{(t-1)}\right)$ identisch für jedes t
  - Formal: Es gibt ein Übergangsmodell (*transition model*) $P(\boldsymbol{R}' | \boldsymbol{R})$, so dass für jedes $t \geq 1$ gilt

$$P\left(\boldsymbol{R}^{(t)} = \boldsymbol{r}' \middle| \boldsymbol{R}^{(t-1)} = \boldsymbol{r}\right) = P(\boldsymbol{R}' = \boldsymbol{r}' | \boldsymbol{R} = \boldsymbol{r})$$

Für ein stationäres System mit Markov Annahme reicht es, die folgenden Verteilungen zu spezifizieren:
$P\left(\boldsymbol{R}^{(0)}\right), P\left(\boldsymbol{R}^{(\tau)} \middle| \boldsymbol{R}^{(\tau-1)}\right)$

Was ist das Problem hiermit?

Beobachtbare und nicht beobachtbare Variablen

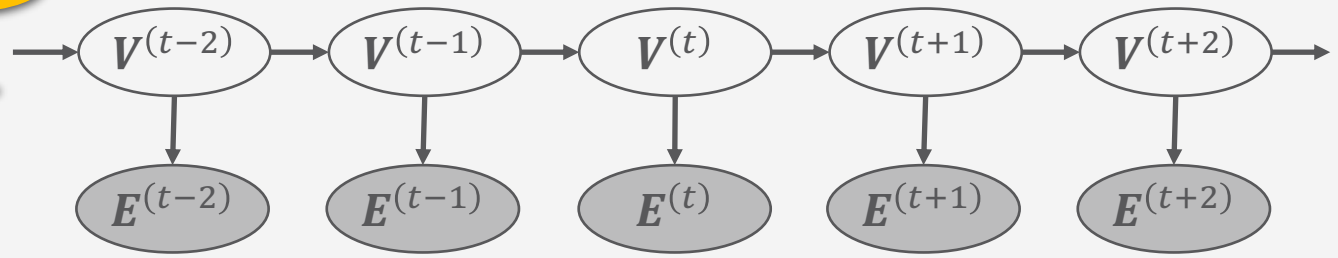
- Menge der Zufallsvariablen $\boldsymbol{R}$ lässt sich in der Regel in Evidenzvariablen $\boldsymbol{E}$ (jeden Zeitschritt mit Evidenz belegt) und latente Variablen $\boldsymbol{V}$ aufteilen
- Verteilung ist dann, wie folgt, faktorisierbar:

$$P_{\boldsymbol{R}}^{T} = P\left(\boldsymbol{V}^{(0:T)}, \boldsymbol{E}^{(0:T)}\right) = P\left(\boldsymbol{V}^{(0)}\right) \prod_{t=1}^{T} P\left(\boldsymbol{V}^{(t)} \middle| \boldsymbol{V}^{(t-1)}\right) P\left(\boldsymbol{E}^{(t)} \middle| \boldsymbol{V}^{(t)}\right)$$

Übergangsmodell

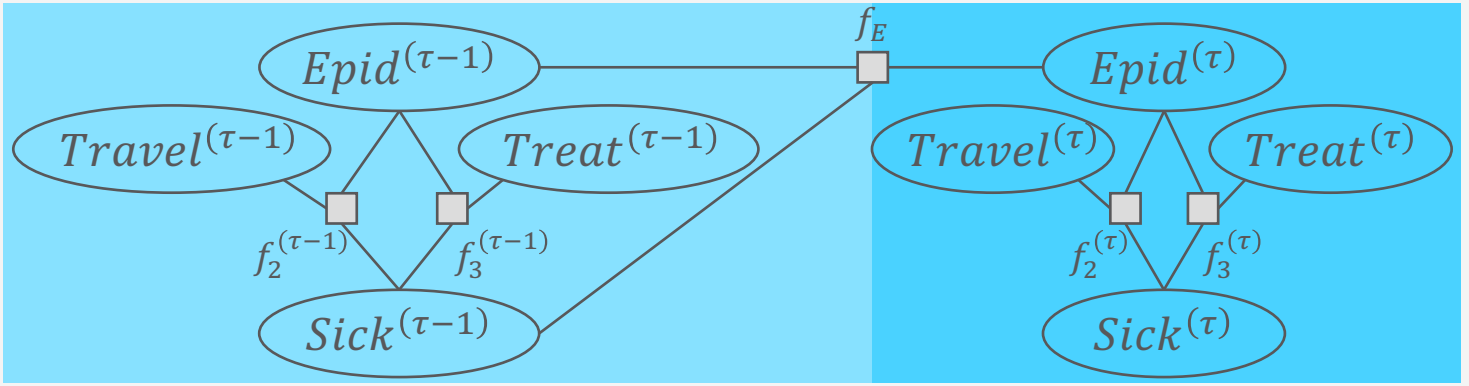
Sensormodell

Was für Unabhängigkeiten herrschen hier?



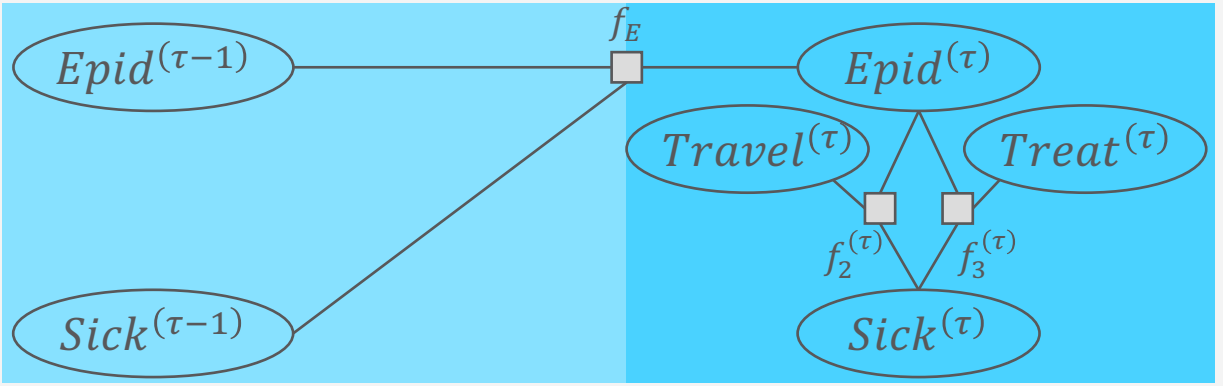
Allgemeine Faktorisierung für sequentielle Modelle

- Übergang zum nächsten Zeitschritt ausgelöst durch neue Beobachtungen / Ausführung von Aktionen
  - Markov Annahme: Effekt eines Aktion spätestens im nächsten Zeitschritt
- Modellierung
  - *Intra-Zeitscheiben*-Faktoren/CPDs: Beschreiben Vorgänge / Effekte innerhalb eines Zeitschrittes
    - Argumente nur aus Zeitschritt τ
      - Z.B. f_2: $Epid^{(\tau)}, Sick^{(\tau)}, Travel^{(\tau)}$
  - *Inter-Zeitscheiben*-Faktoren/CPDs: Beschreiben zeitliche Übergänge bzw. Vorgänge mit Effekten im nächsten Zeitschritt
    - Argumente aus $\tau - 1, \tau$
      - Z.B. f_E: $Epid^{(\tau-1)}, Sick^{(\tau-1)}, Epid^{(\tau)}$



2-Zeitscheiben-Modelle & 1.5-Zeitscheiben-Modelle

- 2-Zeitscheiben-Faktormodell (eigentlich nur noch ein 1.5-Zeitscheiben-Faktormodell)
  - Spezifizierung eines Übergangsmodells $P\left(\boldsymbol{R}^{(\tau)} \middle| \boldsymbol{R}^{(\tau-1)}\right)$ gemäß Faktorisierung
    - Partielle Spezifizierung eines sequentiellen Modells (ohne $P\left(\boldsymbol{R}^{(0)}\right)$)
    - Inter-Zeitscheiben-Effekte und Intra-Zeitscheiben-Effekte
  - Ausreichend: Intra-Zeitscheiben-Faktoren für τ, Inter-Zeitscheiben-Faktoren von $\tau - 1$ zu τ
  - Formal: $F^{\rightarrow} = \left\{f_i^{(\tau)}\right\}_{i=1}^{n} \cup \left\{f_j\right\}_{j=1}^{m}$ mit
    - Intra-Zeitscheiben-Faktoren
      $f_i^{(\tau)} = \phi_i^{(\tau)}\left(R_1^{(\tau)}, \ldots, R_{k_i}^{(\tau)}\right)$
    - Inter-Zeitscheiben-Faktoren
      $f_j = \phi_j\left(R_1^{(t)}, \ldots, R_{k_j}^{(t)}\right), t \in \{\tau, \tau - 1\}$

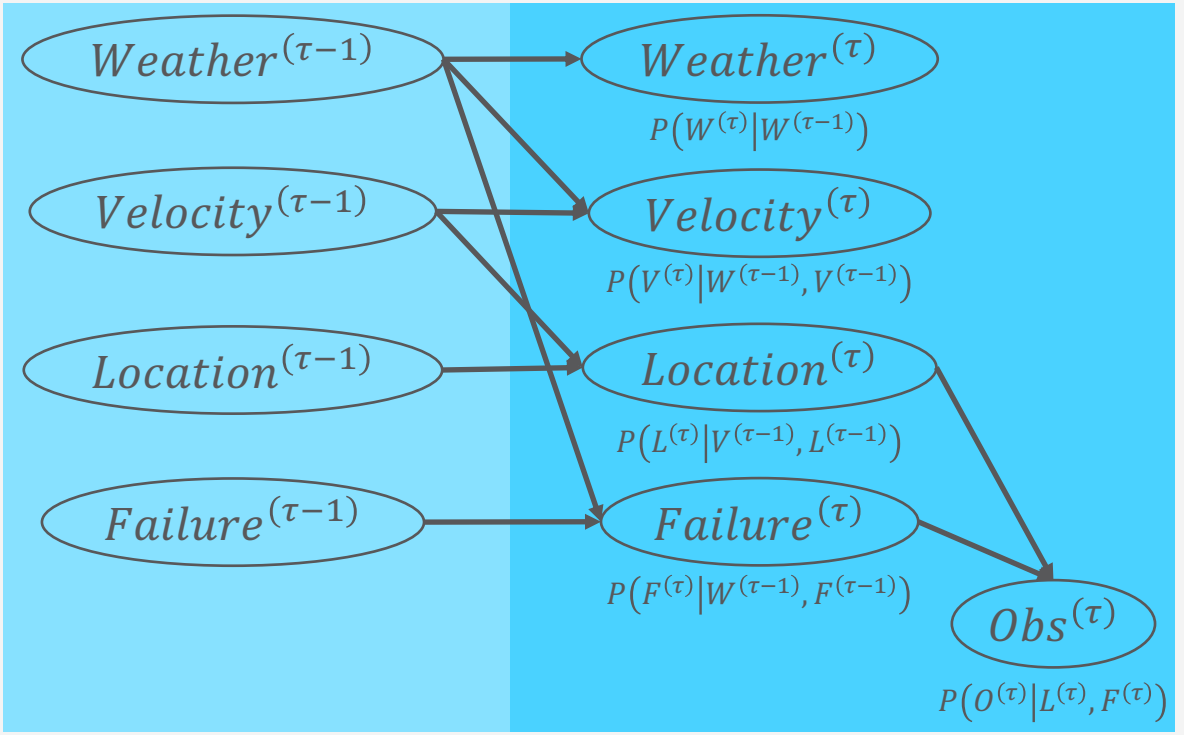


Was für Unabhängigkeiten herrschen hier?

2-Zeitscheiben-Modelle & 1.5-Zeitscheiben-Modelle

- In derselben Art können wir 2-Zeitscheiben-BNs bauen
- 2-Zeitscheiben-BN (eigentlich nur noch ein 1.5-Zeitscheiben-BN)
  - $B^{\rightarrow} = \left\{P\left(R_i^{(\tau)} \middle| \mathrm{Pa}\left(R_i^{(\tau)}\right)\right)\right\}_{i=1}^{n}$ mit
    - Für alle $R_j^{(t)} \in \mathrm{Pa}\left(R_i^{(\tau)}\right)$ gilt $t \in \{\tau, \tau - 1\}$

Im Gegensatz zu der normalen Interpretation von BNs mit einer CPD pro Knoten haben die Zufallsvariablen aus $\tau - 1$ in $B^{\rightarrow}$ keine CPDs assoziiert.



Dynamische Modelle

- Annahmen: Markov-1, stationärer Prozess

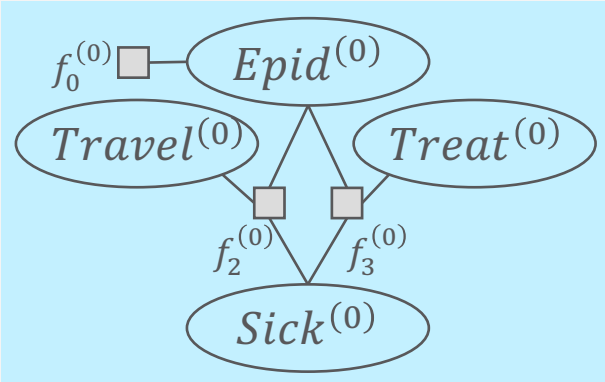
- Übliche Definition: Dynamische Modelle sind Tupel $(M^0, M^{\rightarrow})$, wobei
  - M^0 das Verhalten des ersten Zeitschritts (Intra-Zeitscheiben-Verhalten) beschreibt
  - $M^{\rightarrow}$ ein 2-Zeitscheiben-Modell ist, welches Intra- und Inter-Zeitscheiben-Verhalten beschreibt
    - Template-Modell, das instanziiert wird, indem τ mit einem Zeitpunkt t ersetzt wird
- Häufiger Zusammenhang zwischen $M^0, M^{\rightarrow}$
  - Intra-Zeitscheiben-Verhalten aus $M^{\rightarrow}$ zum Zeitpunkt τ gleich dem Verhalten des ersten Zeitschritts (Intra-Zeitscheiben-Verhalten aus $M^{\rightarrow}$ bei $\tau = 0$)
  - Inter-Zeitscheiben-Verhalten gibt es in M^0 nicht, da es keinen vorhergehenden Schritt gibt
    - Dafür wird M^0 manchmal ergänzt mit Apriori-Informationen kodiert in Verteilungen

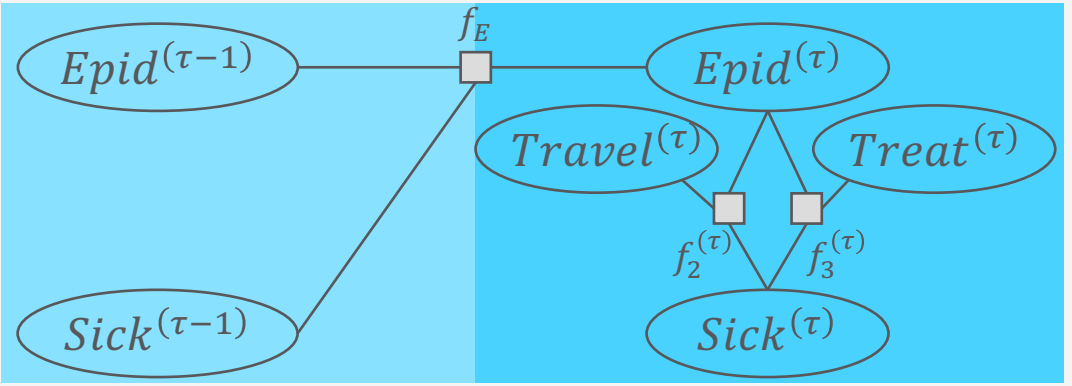
Dynamische Faktormodelle

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_i^{(0)}\right\}_{i=1}^{n_0}, f_i^{(0)} = \phi_i^{(0)}\left(R_1^{(0)}, \dots, R_{k_i}^{(0)}\right)$
  - $F^{\rightarrow}$ ein 2-Zeitscheiben-Faktormodell: $F^{\rightarrow} = \left\{f_i^{(\tau)}\right\}_{i=1}^{n} \cup \left\{f_j\right\}_{j=1}^{m}$ mit
    - $f_i^{(\tau)} = \phi_i^{(\tau)}\left(R_1^{(\tau)}, \dots, R_{k_i}^{(\tau)}\right)$ (Intra-Zeitscheiben-Faktoren)
    - $f_j = \phi_j\left(R_1^{(t)}, \dots, R_{k_j}^{(t)}\right), t \in \{\tau, \tau - 1\}$ (Inter-Zeitscheiben-Faktoren)
  - Beispiel:
    - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
    - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$
    - Keine Annahme zu Beobachtbarkeit

F_0 besteht i.d.R. aus den Intra-Zeitscheiben-Faktoren von $F^{\rightarrow}$ mit $\tau = 0$, evtl. ergänzt um Quasi-Apriori-Verteilungen, i.e.,

$$\left\{f_i^{(0)}\right\}_{i=1}^{n_0} = \left\{f_i^{(\tau)}\right\}_{i=1|\tau=0}^{n} \cup \left\{f_i^{(0)}\right\}_{i=1}^{n'}$$

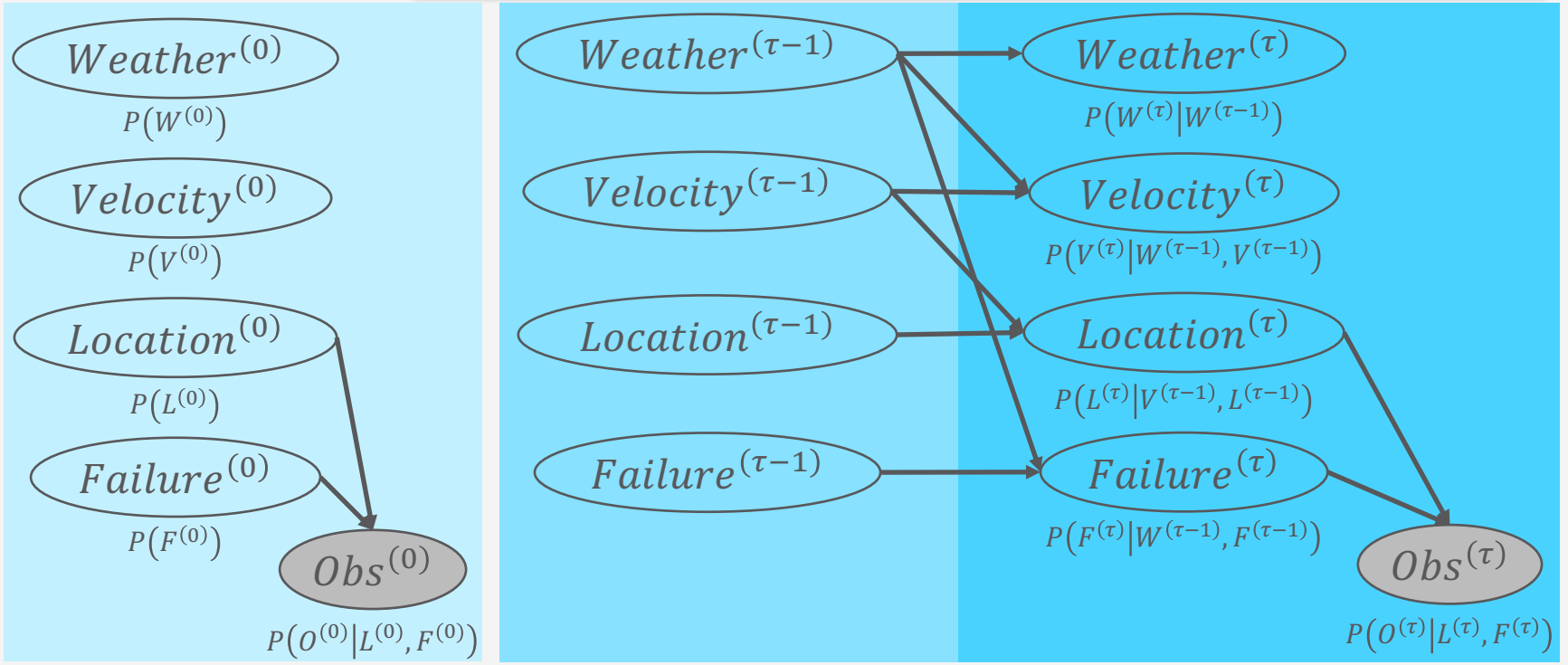




Dynamisches BN (DBN)

- Dynamisches BN $(B^0, B^{\rightarrow})$
  - $B^0 = \left\{P\left(R_i^{(0)} \middle| \mathrm{Pa}\left(R_i^{(0)}\right)\right)\right\}_{i=1}^{n_0}$
    - $\forall\, R_j^{(t)} \in \mathrm{Pa}\left(R_i^{(0)}\right): t = 0$
  - $B^{\rightarrow}$ ein 2-Zeitscheiben-BN: $B^{\rightarrow} = \left\{P\left(R_i^{(\tau)} \middle| \mathrm{Pa}\left(R_i^{(\tau)}\right)\right)\right\}_{i=1}^{n}$
    - $\forall\, R_j^{(t)} \in \mathrm{Pa}\left(R_i^{(\tau)}\right): t \in \{\tau, \tau-1\}$
  - Beispiel: Abbildung
    - Annahme: $\boldsymbol{E} = \{Obs\}$

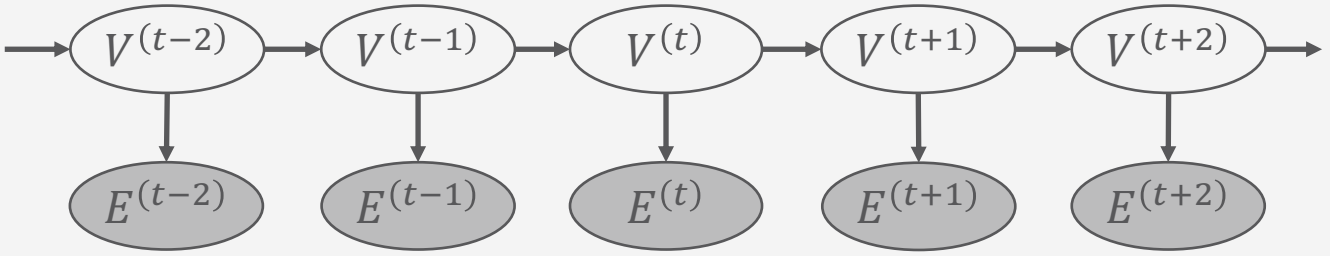
Bei $(B^0, B^{\rightarrow})$ besteht B^0 i.d.R. aus Intra-Zeitscheiben-CPDs, die auch in $B^{\rightarrow}$ gelten, und Apriori-Wahrscheinlichkeiten $P\left(R_{i'}^{(0)}\right)$ für die Wurzelknoten von B^0, was dann die Knoten in $B^{\rightarrow}$ mit Index $\tau - 1$ sind, welche keine CPD in $B^{\rightarrow}$ haben.



Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

Hidden Markov Modell (HMM)

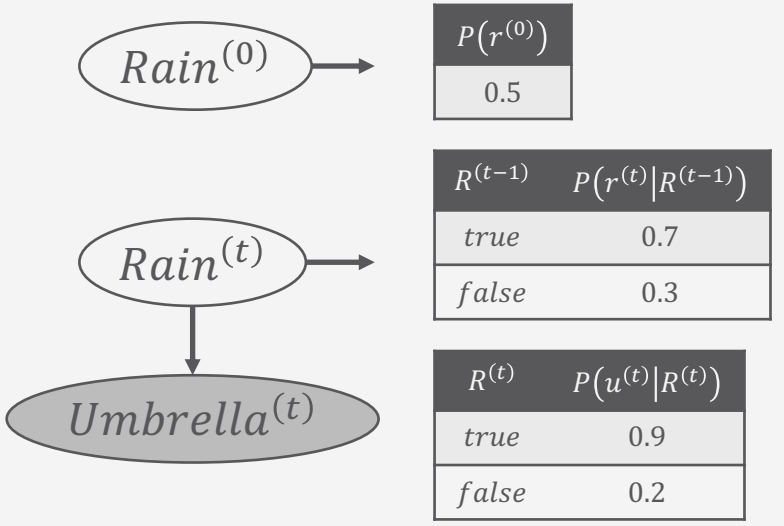
- Spezialfall des DBN mit $\boldsymbol{R} = \{V, E\}$ bzw. $\boldsymbol{V} = \{V\}, \boldsymbol{E} = \{E\}$
  - Latente Zufallsvariable V
  - Evidenzvariable E
  - Modell:
    - $B^0 = P\left(V^{(0)}\right)$
    - $B^{\rightarrow} = \left\{P\left(V^{(\tau)} \middle| V^{(\tau-1)}\right), P\left(E^{(\tau)} \middle| V^{(\tau)}\right)\right\}$
  - Repräsentierte Verteilung: $P_{V,E}^{T} = P\left(V^{(0)}\right) \prod_{t=1}^{T} P\left(V^{(t)} \middle| V^{(t-1)}\right) P\left(E^{(t)} \middle| V^{(t)}\right)$



Ein DBN mit $\boldsymbol{R} = \boldsymbol{V} \cup \boldsymbol{E}$ kann jederzeit in ein HMM umgewandelt werden:

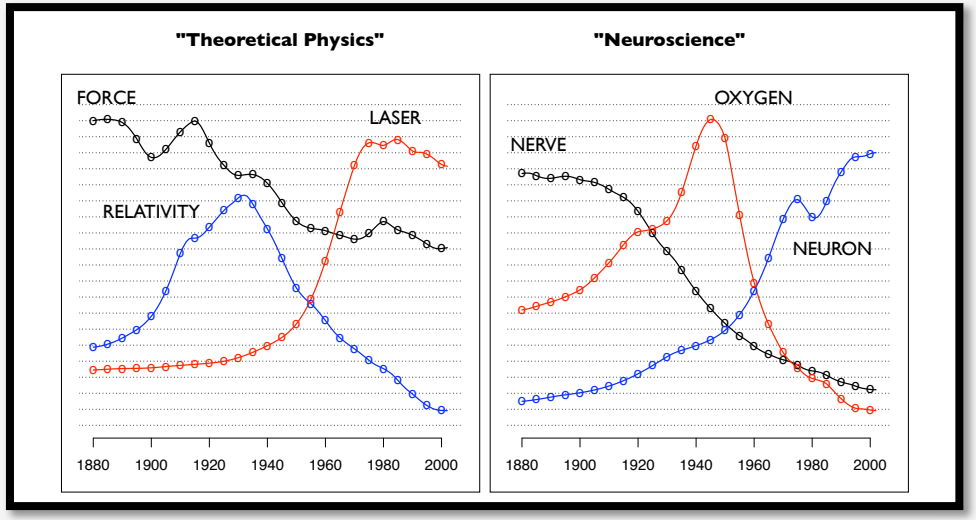
- Mit jedem $\boldsymbol{v} \in \mathrm{Val}(\boldsymbol{V})$ als abstrakten Wert v, $\mathrm{Val}(V) = \mathrm{Val}(\boldsymbol{V})$
- Mit jedem $\boldsymbol{e} \in \mathrm{Val}(\boldsymbol{E})$ als abstrakten Wert e, $\mathrm{Val}(E) = \mathrm{Val}(\boldsymbol{E})$

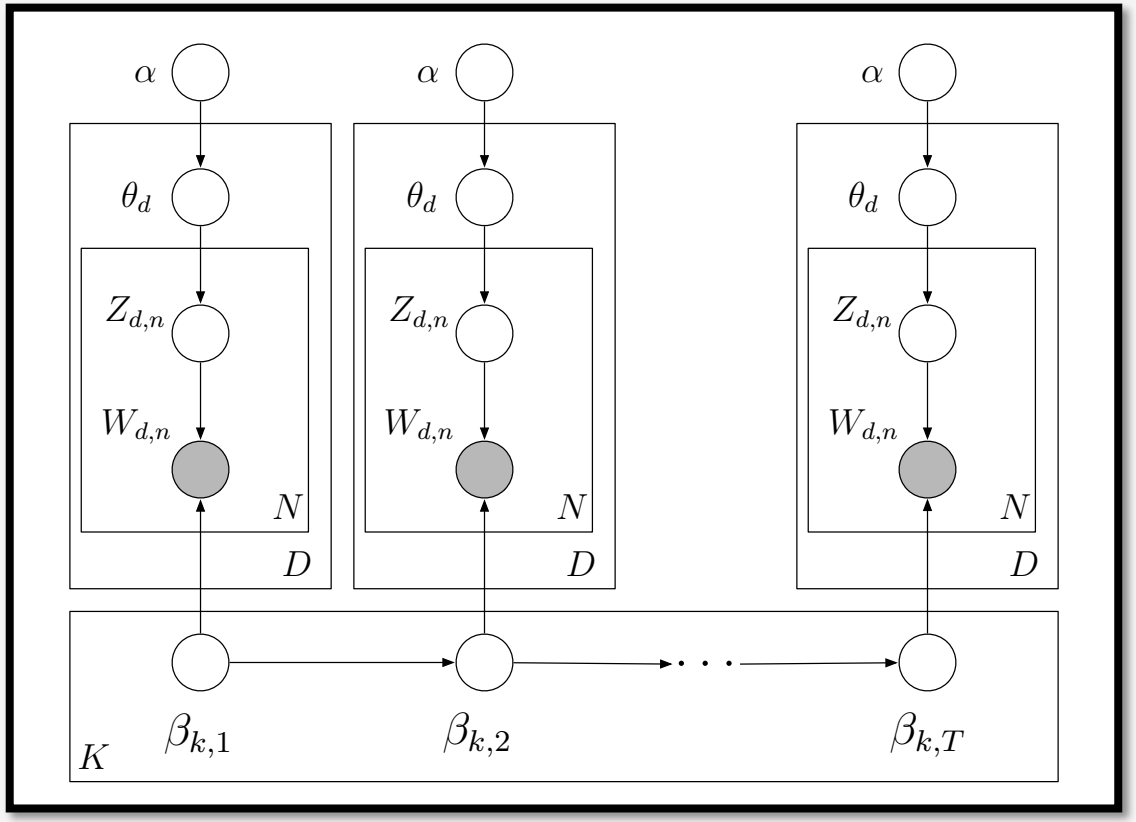
$\mathrm{Val}(V)$ und $\mathrm{Val}(E)$ entsprechend groß!



Topic Modellierung über die Zeit: Dynamic Topic Modell

- Corpus in Zeitscheiben einteilen (z.B. pro Jahr)
- Annahme: Innerhalb einer Zeitscheibe, aus einem LDA Modell generiert
- Erlaube, dass sich Topic-Verteilungen von einer Zeitscheibe zur nächsten verändern





Semantik

- Semantik definiert über die vollständige gemeinsame Wahrscheinlichkeitsverteilung gegeben ein Zeitpunkt $T > 0$
  - Dynamisches Modell $(M^0, M^{\rightarrow})$ für T Zeitschritte ausrollen

$$M = M^{(0:T)} = M^0 \cup \bigcup_{t=1}^{T} M^{\rightarrow|\tau=t}$$

    - $M^{\rightarrow|\tau=t}$: Ersetzte τ mit t in $M_{\rightarrow}$
  - Ergibt ein episodisches Modell mit bekannter Semantik, i.e., ein Modell, welches eine vollständige gemeinsame Wahrscheinlichkeitsverteilung als (normalisiertes) Produkt der Faktoren / CPDs repräsentiert

$$P_M^T = \frac{1}{Z} \prod_{\zeta \in M} \zeta$$

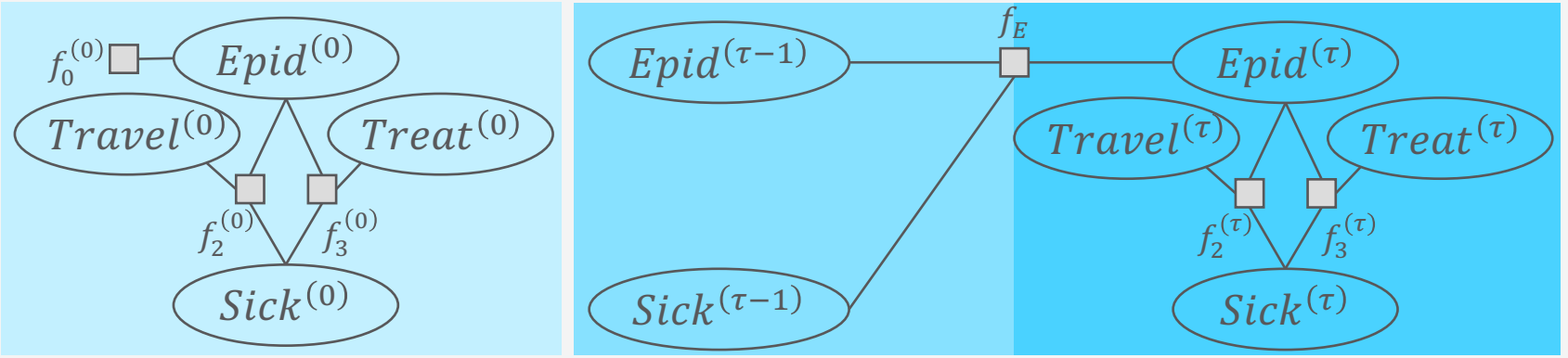
    - ζ ein Faktor, wenn $M^0, M^{\rightarrow}$ Faktormodelle, oder eine CPD, wenn $M^0, M^{\rightarrow}$ BNs (dann $Z = 1$)

Ausrollen: Beispiel – $T = 2$

- Ausgerolltes Modell

$$F = F^0 \cup \bigcup_{t=1}^{2} F^{\rightarrow|\tau=t}$$

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
  - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$



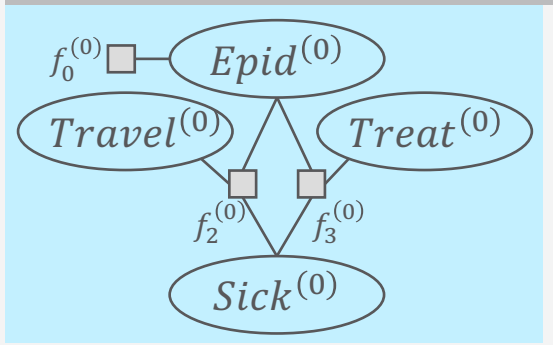
Ausrollen: Beispiel – $T = 2, t = 0$

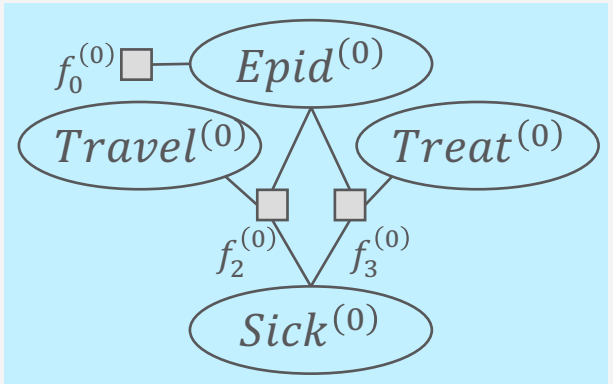
- Ausgerolltes Modell

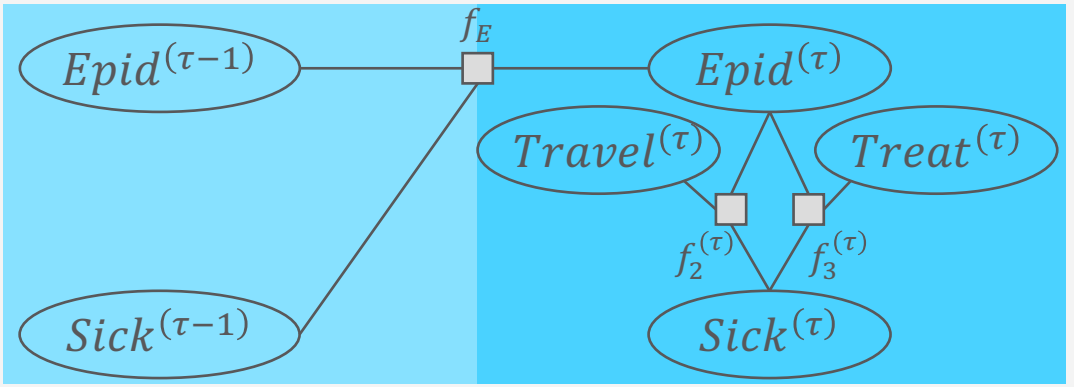
$$F = F^0 \cup \bigcup_{t=1}^{2} F^{\rightarrow|\tau=t}$$

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
  - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$

$F = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\} \quad \cup \cdots$







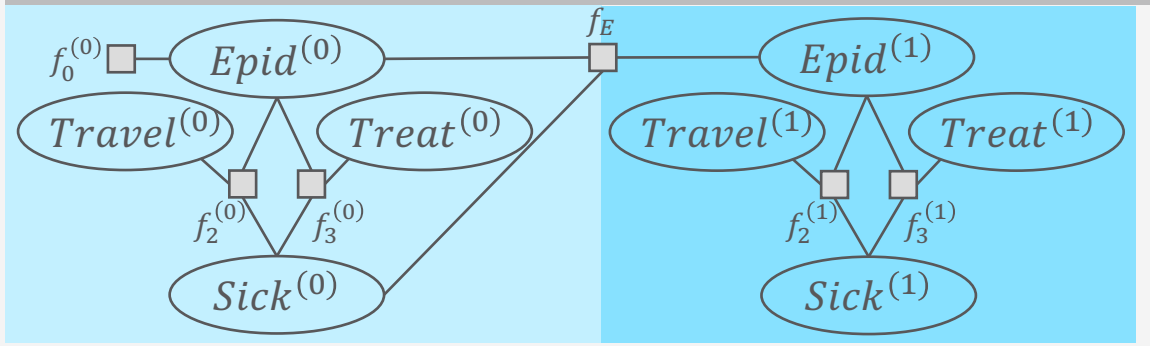
Ausrollen: Beispiel – $T = 2, t = 1$

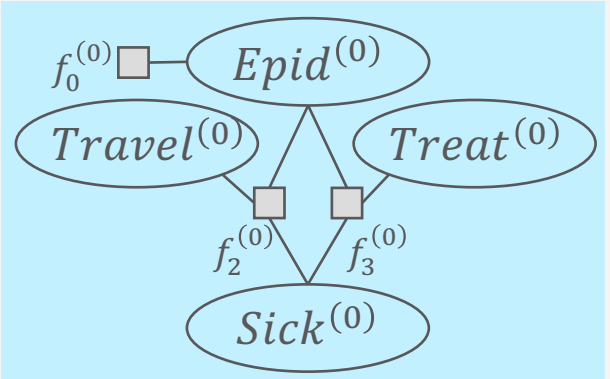
- Ausgerolltes Modell

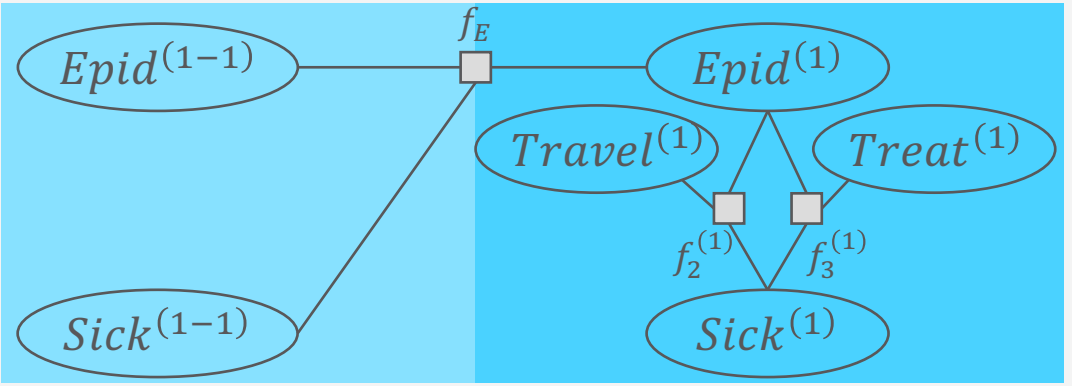
$$F = F^0 \cup \bigcup_{t=1}^{2} F^{\rightarrow|\tau=t}$$

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
  - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$

$F = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\} \quad \cup \quad \left\{f_2^{(1)}, f_3^{(1)}\right\} \cup \{f_E\} \quad \cup \cdots$





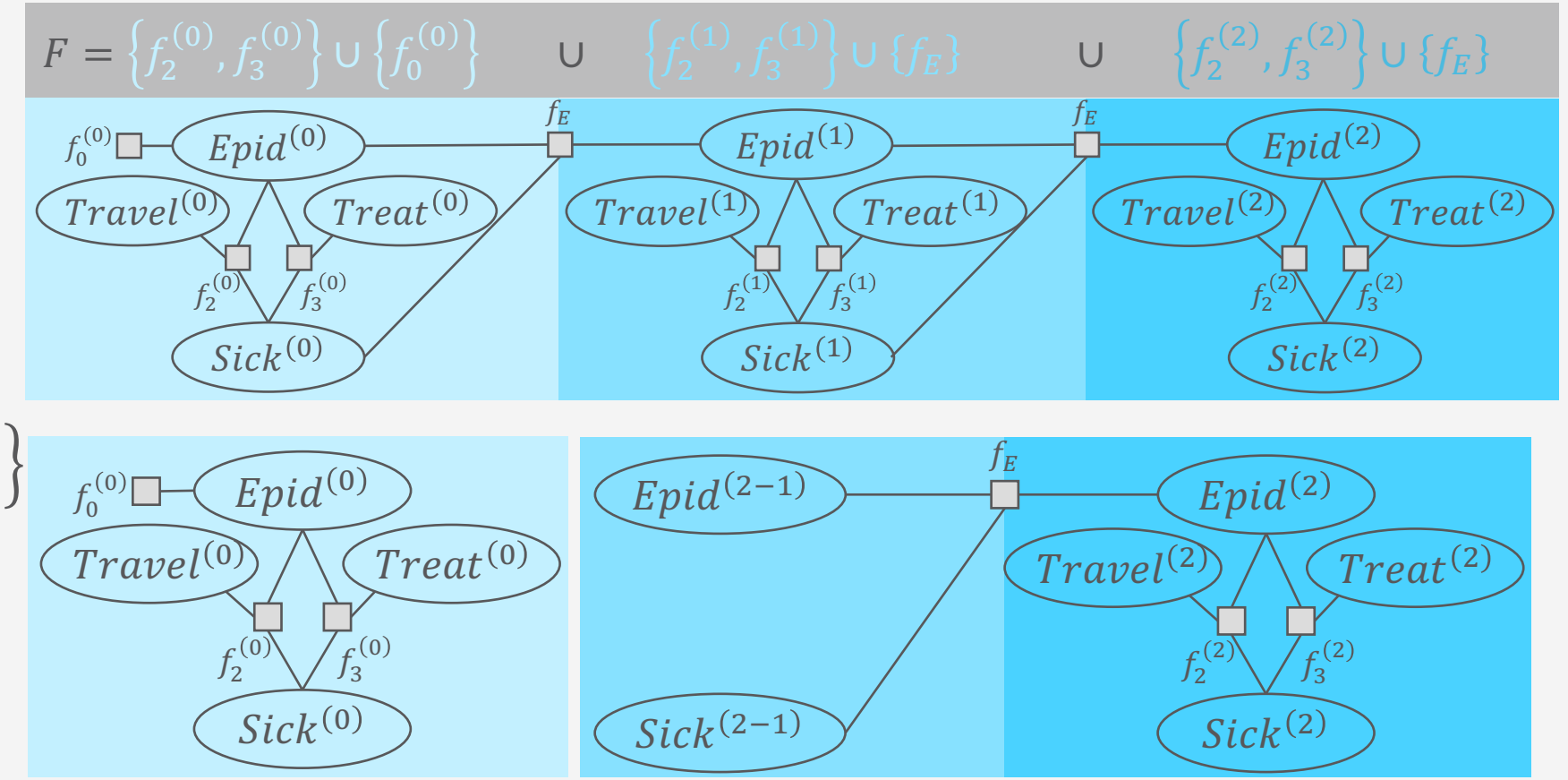


Ausrollen: Beispiel – $T = 2, t = 2$

- Ausgerolltes Modell

$$F = F^0 \cup \bigcup_{t=1}^{2} F^{\rightarrow|\tau=t}$$

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
  - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$



Ausrollen: Beispiel – $T = 2$, Semantik

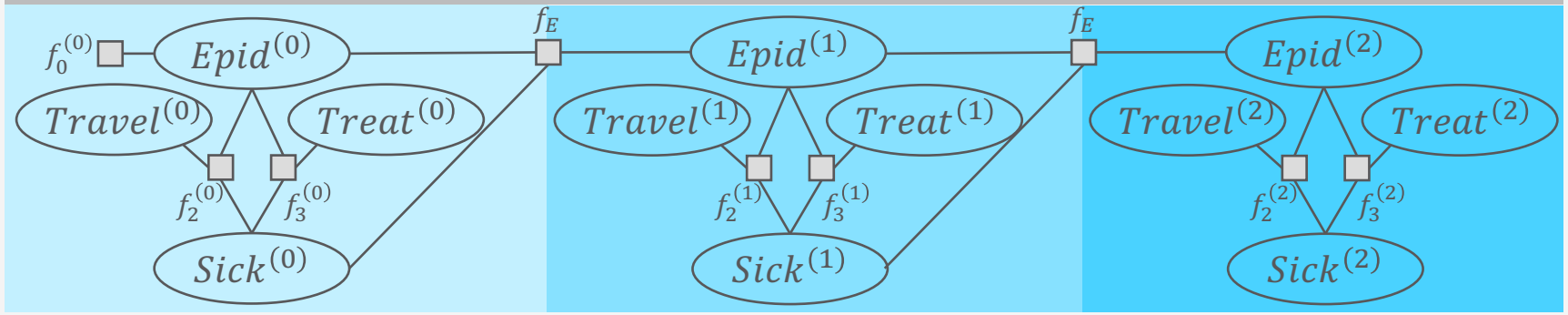
Ein Blick auf die Unabhängigkeiten

- Ausgerolltes Modell

$$F = F^0 \cup \bigcup_{t=1}^{2} F^{\rightarrow|\tau=t}$$

- Dynamisches Faktormodell $(F^0, F^{\rightarrow})$ mit
  - $F^0 = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\}$
  - $F^{\rightarrow} = \left\{f_2^{(\tau)}, f_3^{(\tau)}\right\} \cup \{f_E\}$

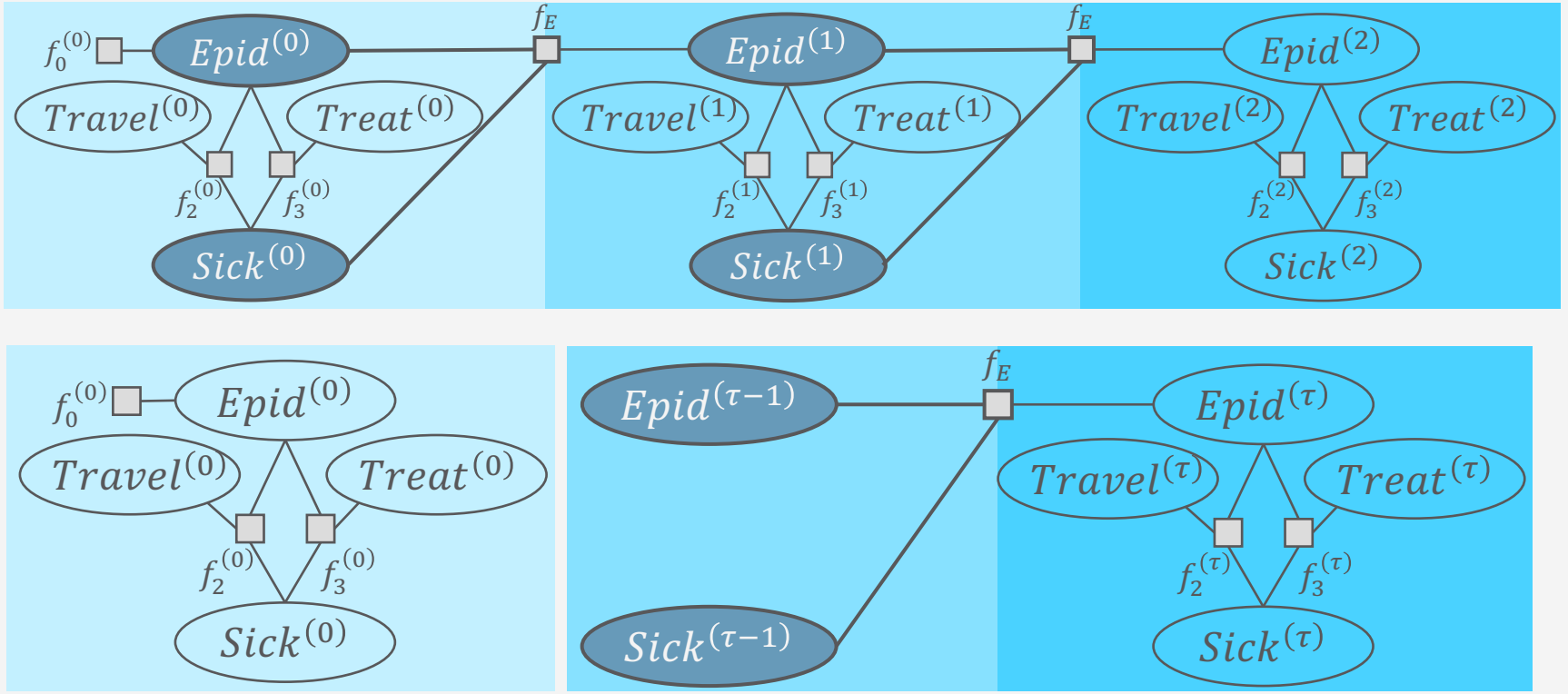
$$F = \left\{f_2^{(0)}, f_3^{(0)}\right\} \cup \left\{f_0^{(0)}\right\} \quad \cup \quad \left\{f_2^{(1)}, f_3^{(1)}\right\} \cup \{f_E\} \quad \cup \quad \left\{f_2^{(2)}, f_3^{(2)}\right\} \cup \{f_E\}$$



$$P_F^T = \frac{1}{Z}\prod_{f\in F} f = \frac{1}{Z}\phi_2^{(0)}\left(E^{(0)}, S^{(0)}, Tl^{(0)}\right)\cdot\phi_3^{(0)}\left(E^{(0)}, S^{(0)}, Tt^{(0)}\right)\cdot\phi_0^{(0)}\left(E^{(0)}\right)$$
$$\cdot\phi_2^{(1)}\left(E^{(1)}, S^{(1)}, Tl^{(1)}\right)\cdot\phi_3^{(1)}\left(E^{(1)}, S^{(1)}, Tt^{(1)}\right)\cdot\phi_E\left(E^{(1)}\right)$$
$$\cdot\phi_2^{(2)}\left(E^{(2)}, S^{(2)}, Tl^{(2)}\right)\cdot\phi_3^{(2)}\left(E^{(2)}, S^{(2)}, Tt^{(2)}\right)\cdot\phi_E\left(E^{(2)}\right)$$

Interfaces

- Interface: Separierende Untermenge an Zufallsvariablen, durch die die Zeitscheiben voneinander unabhängig sind
  - Beispiel:
    - $Epid^{(\tau-1)}$
    - $Sick^{(\tau-1)}$
  - D.h., Variablen der Inter-Zeitscheiben-Faktoren f_i mit $\tau - 1$



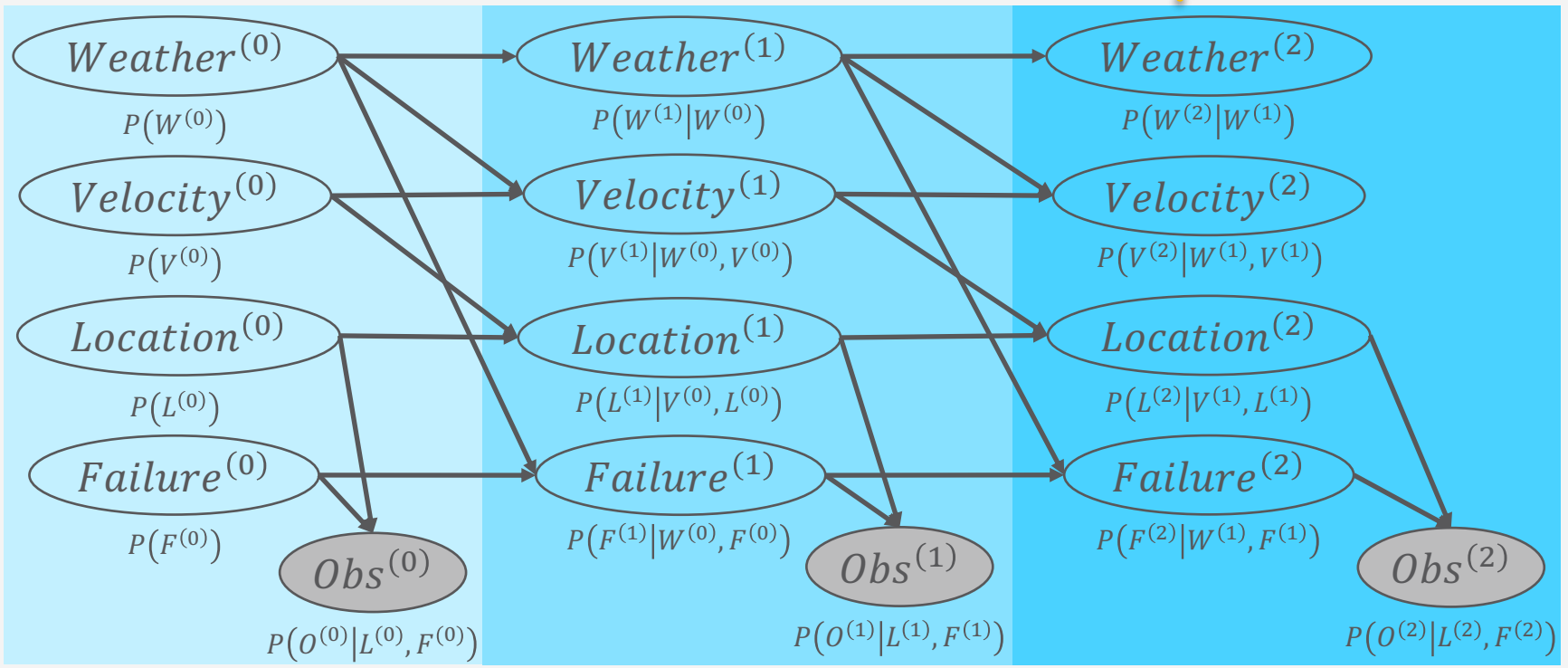
Wie sieht das bei DBNs aus? DBN ausrollen: Beispiel – $T = 2$

- Ausgerolltes Modell

$$B = B^0 \cup \bigcup_{t=1}^{2} B^{\rightarrow|\tau=t}$$

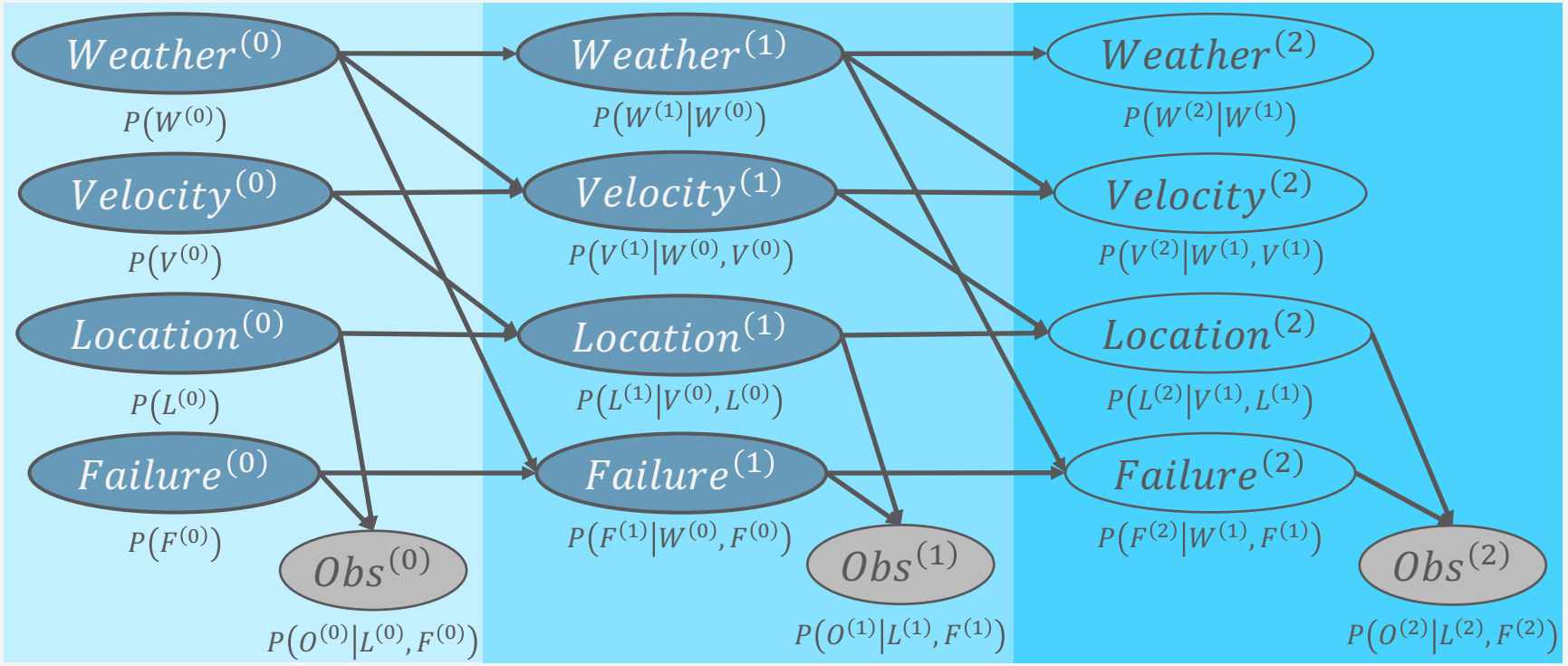
- DBN $(B^0, B^{\rightarrow})$ mit
  - $B^0 = \{P(O^{(0)}|L^{(0)}, F^{(0)})\} \cup \{P(W^{(0)}), P(V^{(0)}), P(L^{(0)}), P(F^{(0)})\}$
  - $B^{\rightarrow} = \{P(O^{(\tau)}|L^{(\tau)}, F^{(\tau)})\} \cup \{P(W^{(\tau)}|W^{(\tau-1)}), P(V^{(\tau)}|W^{(\tau-1)}, V^{(\tau-1)}), P(L^{(\tau)}|V^{(\tau-1)}, L^{(\tau-1)}), P(F^{(\tau)}|W^{(\tau-1)}, F^{(\tau-1)})\}$

Und der Blick auf die Unabhängigkeiten



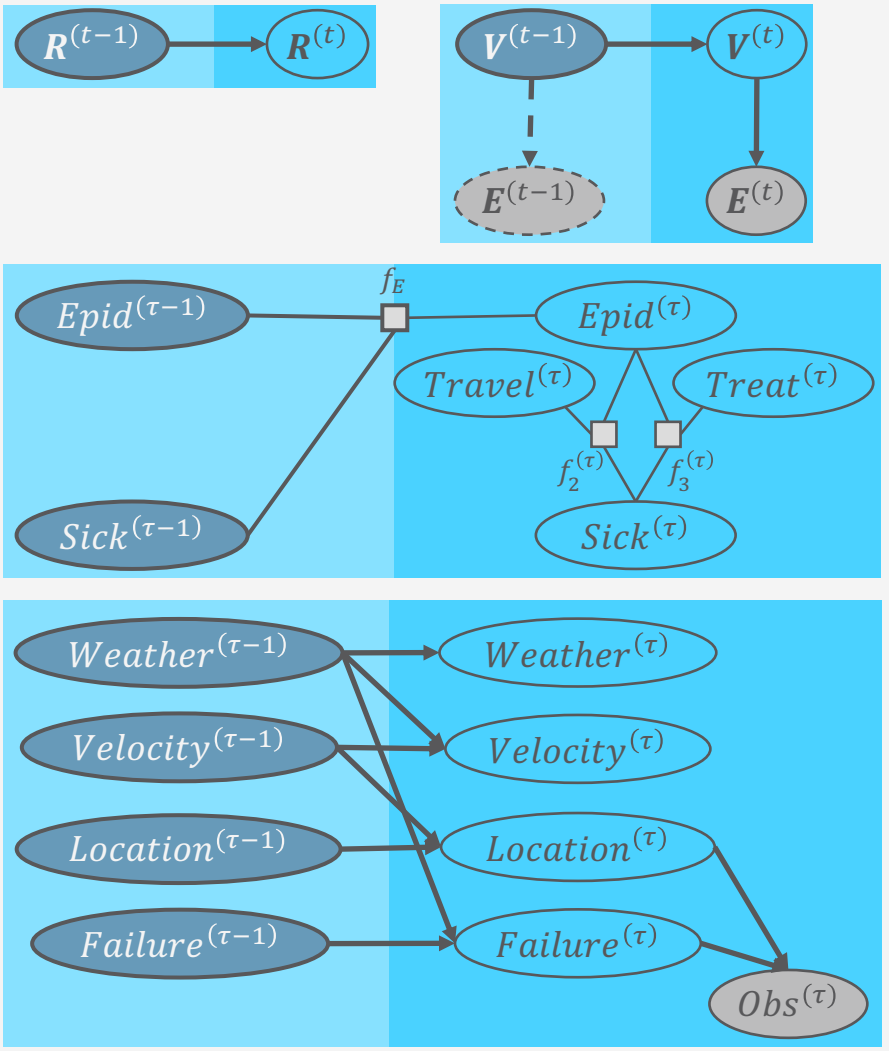
Interfaces

- Interface: Separierende Untermenge an Zufallsvariablen, durch die die Zeitscheiben voneinander unabhängig sind
  - Beispiel:
    - $Wheater^{(\tau-1)}$
    - $Velocity^{(\tau-1)}$
    - $Location^{(\tau-1)}$
    - $Failure^{(\tau-1)}$
  - D.h., Variablen des Übergangsmodells mit $\tau - 1$



Interfaces

- Vorwärts-Interface $\boldsymbol{I}^{(\tau-1)}$: Separatoren der Zeitscheiben bei fortschreitender Zeit (von $M^{(\tau-1)}$ nach $M^{(\tau)}$)
  - Triviale Separatoren, in der Regel nicht minimal
    - Maximales Interface: $\boldsymbol{R}$
    - Bei $\boldsymbol{R} = \boldsymbol{V} \cup \boldsymbol{E}$: maximales Interface $\boldsymbol{V}$
  - Dynamisches Faktormodell:
    Menge der Variablen mit Index $\tau - 1$ in $F^{\rightarrow}$
    - $\boldsymbol{I} = \left\{ R^{(\tau-1)} \middle| R^{(\tau-1)} \in \mathrm{rv}(F^{\rightarrow}) \right\}$
    - Kommen nur in Inter-Zeitscheiben-Faktoren vor
  - DBN: Menge der Variablen mit Index $\tau - 1$ in $B^{\rightarrow}$
    - $\boldsymbol{I} = \left\{ R^{(\tau-1)} \middle| R^{(\tau-1)} \in \mathrm{rv}(B^{\rightarrow}) \right\}$
    - Sind Elternknoten von τ-Variablen in $B^{\rightarrow}$



$\boldsymbol{I}$ trennt auch von $M^{(\tau)}$ nach $M^{(\tau-1)}$; es gibt aber auch Rückwärts-Interfaces, die aus der Richtung τ nach $\tau - 1$ bestimmt werden [Näheres dazu in Kevin Murphys Dissertation, 2002]

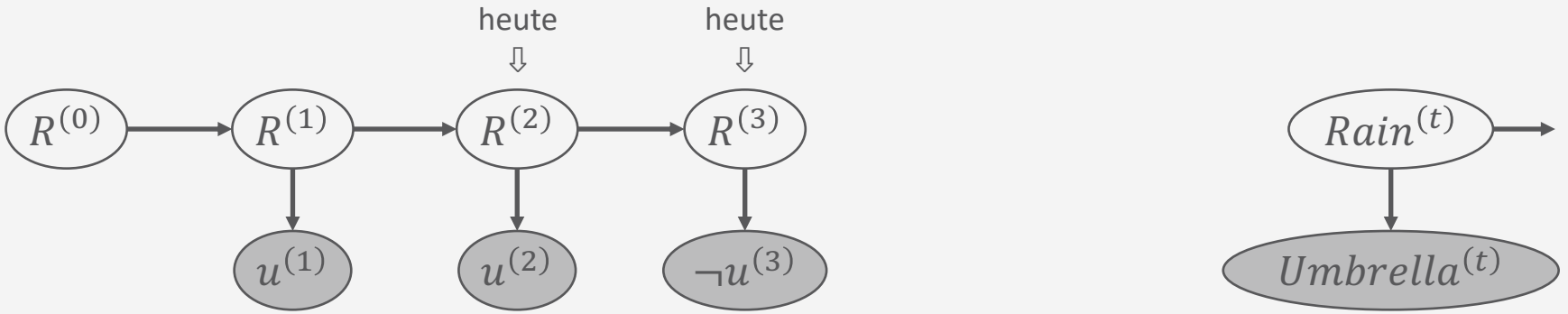
Anfragebeantwortungsproblem & Anfragetypen

- Filtern / überwachen (*filtering*)
  - Was ist die Wahrscheinlichkeit, dass es heute ($t = 2$) regnet gegeben alle Beobachtungen zu Regenschirmen bis einschließlich heute?

$$P\left(R^{(2)} \middle| u^{(1)}, u^{(2)}\right)$$

  - Nächster Tag: Was ist die Wahrscheinlichkeit, dass es heute ($t = 3$) regnet gegeben alle Beobachtungen zu Regenschirmen bis einschließlich heute?

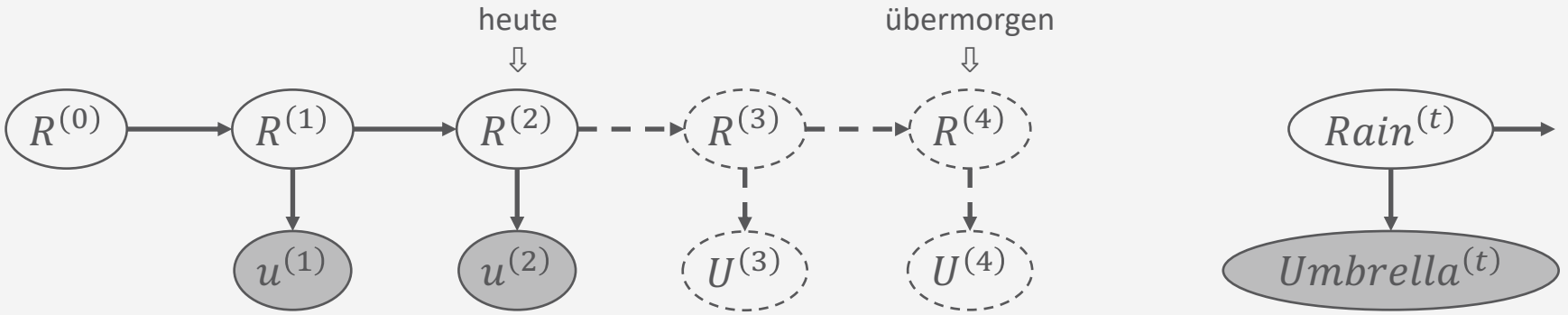
$$P\left(R^{(3)} \middle| u^{(1)}, u^{(2)}, \neg u^{(3)}\right)$$



Anfragebeantwortungsproblem & Anfragetypen

- Vorhersage (*prediction*)
  - Was ist die Wahrscheinlichkeit, dass es übermorgen regnet gegeben alle Beobachtungen zu Regenschirmen bis einschließlich heute ($t = 2$)?

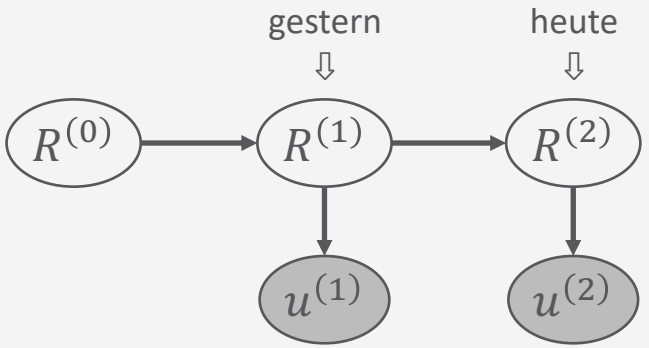
$$P\left(R^{(4)} \middle| u^{(1)}, u^{(2)}\right)$$

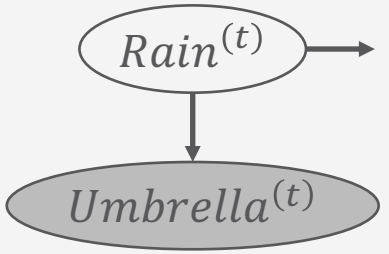


Anfragebeantwortungsproblem & Anfragetypen

- Rückschau (*hindsight*)
  - Auch Glättung (*smoothing*) genannt, bezieht sich aber eher auf das, was man berechnungstechnisch tut, als auf die Anfrage
  - Was ist die Wahrscheinlichkeit, dass es gestern geregnet hat gegeben alle Beobachtungen zu Regenschirmen bis einschließlich heute ($t = 2$)?

$$P\left(R^{(1)} \middle| u^{(1)}, u^{(2)}\right)$$



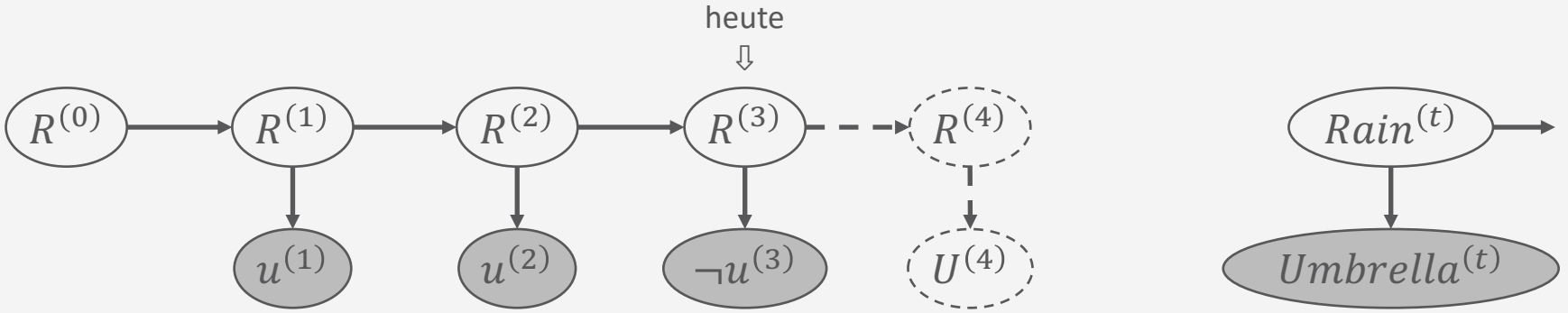


Anfragebeantwortungsproblem & Anfragetypen

- Wahrscheinlichste Erklärung (*MPE*)
  - Wenn der Regenschirm an den ersten beiden Tagen, aber nicht am dritten zu beobachten war, was ist die wahrscheinlichste Wettersequenz, die diese Regenschirm-Beobachtungen produziert hat?

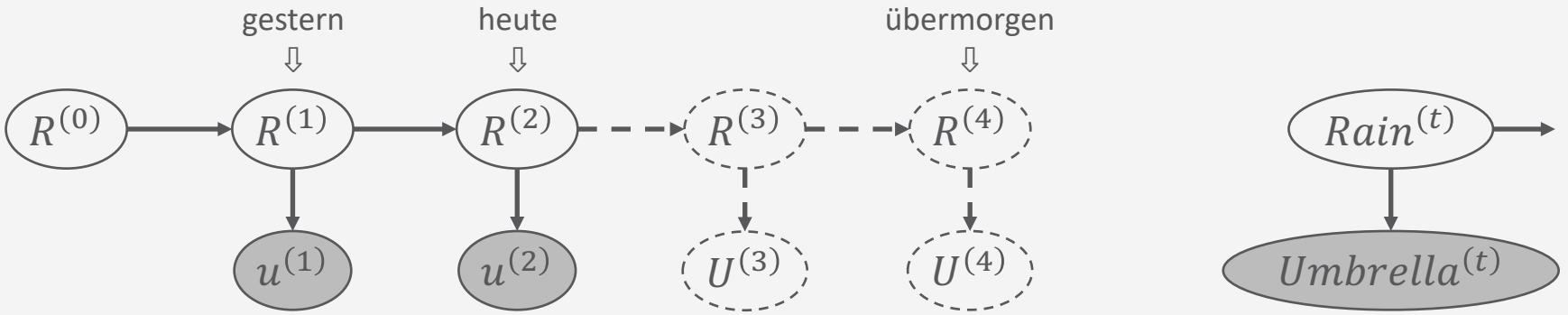
$$MPE\left(\boldsymbol{e}^{(1:3)}\right) = \arg\max_{\boldsymbol{r}^{(0:3)}} P\left(\boldsymbol{r}^{(0:3)} \middle| \boldsymbol{e}^{(1:3)}\right)$$

  - Geht natürlich auch als *MAP*, wenn es mehr Variablen ohne Evidenz gibt und
    - Man nur an einer Untermenge von denen interessiert ist: $\boldsymbol{U} \subseteq \boldsymbol{V}$, $\boldsymbol{V}$ Variablen ohne Evidenz
    - Man nur an einigen (letzten k) Zeitschritten interessiert ist: $\boldsymbol{V}^{(t-k:t)}$ bei $\boldsymbol{e}^{(0:t)}$



Anfragen: Formal

- Wahrscheinlichkeitsanfrage $P\left(\boldsymbol{S}^{(\pi)} \middle| \boldsymbol{e}^{(0:t)}\right)$, t der aktuelle Zeitschritt, an ein Modell M mit Template-Zufallsvariablen $\boldsymbol{R}$
  - Filtern / Überwachen (*filtering*): $\pi = t$
  - Vorhersage (*prediction*): $\pi > t$
  - Rückschau (*hindsight*): $\pi < t$



Anfragen: Formal

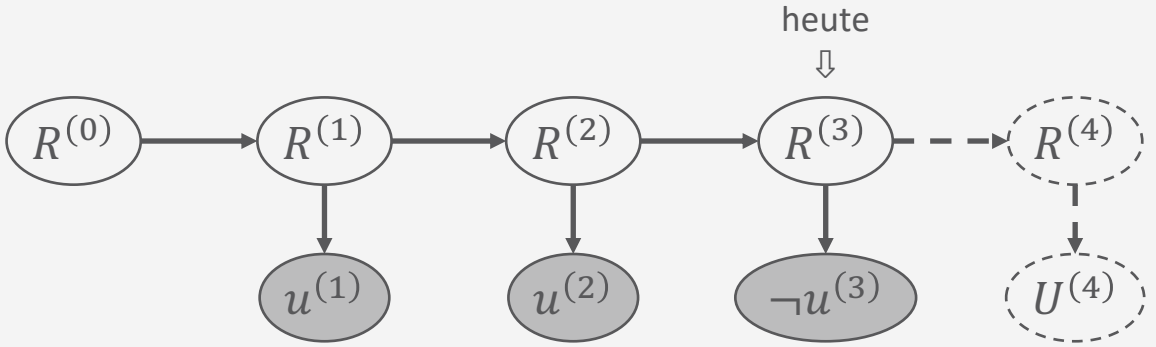
- Zustandsanfragen

$$MPE_M\left(\boldsymbol{e}^{(0:T)}\right) = \underset{\boldsymbol{v}^{(0:t)} \in \mathrm{Val}\left(\boldsymbol{V}^{(0:T)}\right)}{\arg\max} P\left(\boldsymbol{v}^{(0:T)} \middle| \boldsymbol{e}^{(0:T)}\right)$$

  - $\boldsymbol{V}^{(0:T)} = \boldsymbol{R}^{(0:T)} \setminus \mathrm{rv}\left(\boldsymbol{e}^{(0:T)}\right)$

$$MAP_M\left(\boldsymbol{U}^{(t_1:t_2)} \middle| \boldsymbol{e}^{(0:t)}\right) = \underset{\boldsymbol{u}^{(t_1:t_2)} \in \mathrm{Val}\left(\boldsymbol{U}^{(t_1:t_2)}\right)}{\arg\max} \sum_{\boldsymbol{v}^{(0:T)} \in \mathrm{Val}\left(\boldsymbol{V}^{(0:T)}\right)} P\left(\boldsymbol{u}^{(t_1:t_2)}, \boldsymbol{v}^{(0:T)} \middle| \boldsymbol{e}^{(0:t)}\right)$$

  - $\boldsymbol{V}^{(0:T)} = \boldsymbol{R}^{(0:T)} \setminus \boldsymbol{U}^{(t_1:t_2)} \setminus \mathrm{rv}\left(\boldsymbol{e}^{(0:t)}\right), T = \max\{t, t_1, t_2\}$



Zwischenzusammenfassung

- Sequentielle Daten modellieren
  - Annahmen: Markov-1, stationärer Prozess
  - Dynamische Modelle bestehen aus zwei episodischen Modellen
    - Eins um den ersten Schritt zu beschreiben
    - Eins um die Vorgänge innerhalb eines Schritts und den Übergang von einem Schritt zum nächsten zu beschreiben
      - Kopiervorlage
  - Semantik: Modell für T Schritte ausrollen, ergibt ein episodisches Modell mit bekannter Semantik (vollständige gemeinsame Verteilung)
- Dynamische Modelle: dynamische Faktormodelle, DBNs, HMMs
  - Folgen alle den gleichen Annahmen und der gleichen Struktur / Idee
- Inferenzaufgaben: Filtering, Prediction, Hindsight; MPE, MAP

Überblick: 6. Sequentielle PGMs und Inferenz

A. Sequentielle PGMs

- Templates, dynamische BNs, dynamische Faktormodelle, Hidden-Markov-Modelle; Semantik
- Inferenzaufgaben: Wahrscheinlichkeitsanfragen (Filtering, Prediction, Hindsight), Zustandsanfragen (MPE, MAP)

B. Sequentielle Inferenz

- Naïve Inferenz mittels Ausrollen, Interface Algorithmus, Komplexität, Approximationen

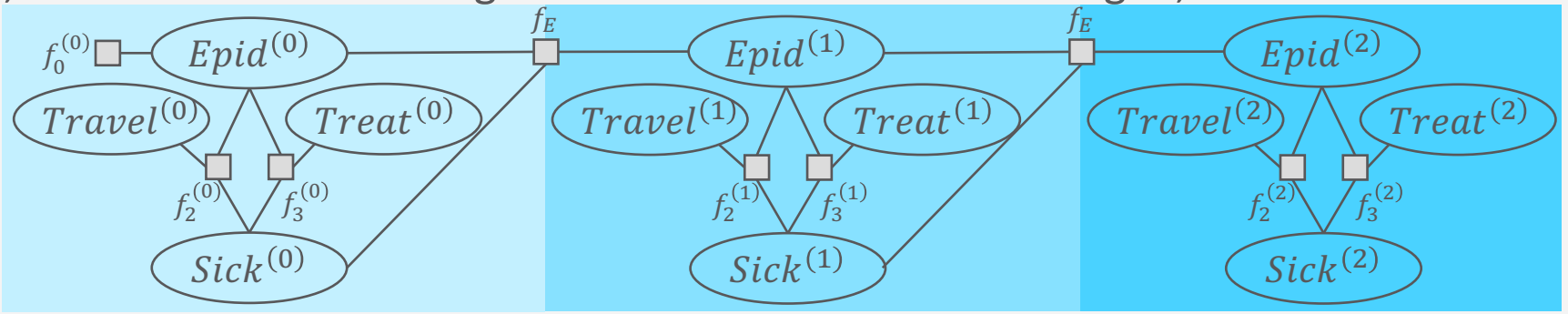
C. Spezialfall Hidden-Markov-Modelle

- Viterbi-Algorithmus für MPEs
- Anfragebeantwortung durch Matrixoperationen
- Baum-Welch-Algorithmus zum Lernen

Naïve Inferenz über Ausrollen des Modells

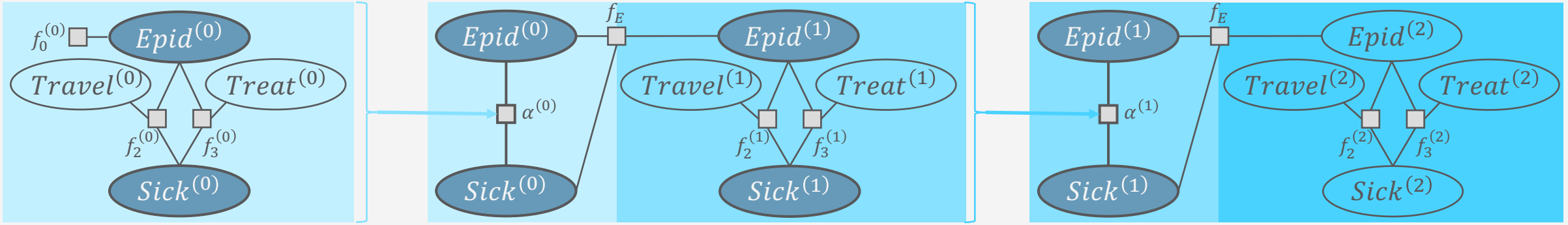
- Gegeben ein Modell $M = (M^0, M^{\rightarrow})$ und eine Anfrage $P\left(\boldsymbol{R}^{(\pi)} \middle| \boldsymbol{e}^{(1:t)}\right)$
  1. Modell für $T = \max\{t, \pi\}$ Schritte ausrollen → episodisches Modell
  2. $P\left(\boldsymbol{R}^{(\pi)} \middle| \boldsymbol{e}^{(1:t)}\right)$ mit Algorithmus für episodische Modelle beantworten
- Probleme
  - Ausgerolltes Modell sehr groß
  - Neu ausrollen (oder anpassen), wenn sich T oder $\boldsymbol{e}^{(1:t)}$ ändern
    - Zeit schreitet voran: T++, $\boldsymbol{e}^{(1:t)}$ um $\boldsymbol{e}^{(t+1)}$ verlängert → $M^{\rightarrow}$ für $\tau = t + 1$ anhängen, Evidenz absorbieren
    - Vor allem bei mehreren Anfragen mit sich ändernden π pro t aufwendig

Ziel: Nur mit einem aktuellen Modell $M^{(t)}$ arbeiten, welches mehrere Anfragen und fortschreitende Zeit (t++) effizient verarbeiten kann → Interfaces nutzen



Interfaces um mit der Zeit zu gehen

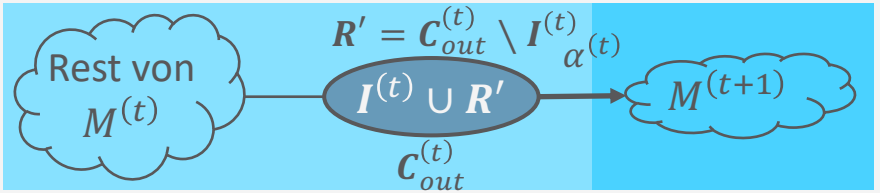
- Wenn Inferenz für Zeitschritt t fertig
  - Anfrage für $\boldsymbol{I}^{(t)}$ an aktuelles Modell $M^{(t)}$ stellen
    - Quasi wie im Junction Tree Algorithmus (JT): Nachricht $\alpha^{(t)}$ über Separator $\boldsymbol{I}^{(t)}$ berechnen
  - Resultat der Anfrage zu Modell $M^{(t+1)}$ hinzufügen
    - $M^{(t+1)} = M^{\rightarrow|\tau=t+1} \cup \alpha^{(t)}$
      - Ohne Anbindung an $M^{(0:t)}$, da Information über $M^{(0:t)}$ in $\alpha^{(t)}$ vorhanden
- Jtrees und JT dafür nutzen



Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

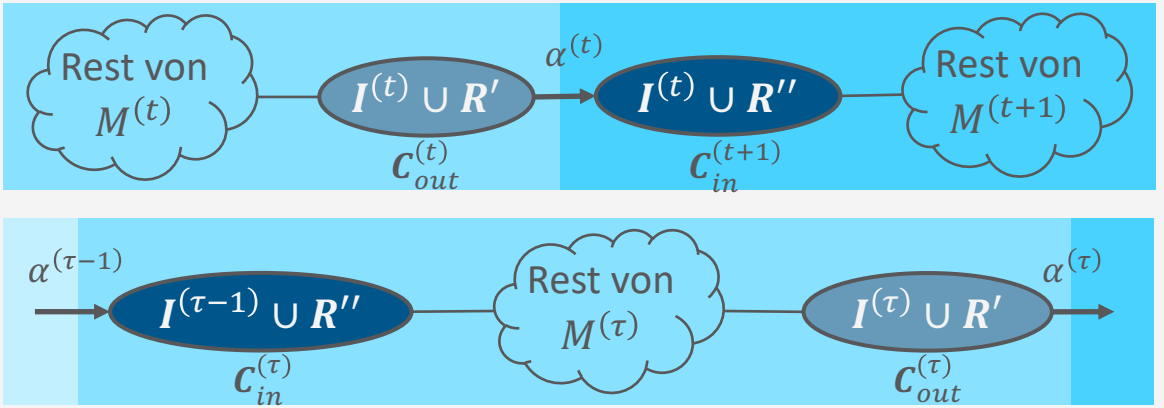
Jtrees und JT um mit der Zeit zu gehen

- Ermöglicht auch effiziente Beantwortung von mehreren Anfragen pro Zeitscheibe
- Anpassung für die Dynamik: $\boldsymbol{I}^{(t)}$ in einem Cluster vorkommen lassen
  - Vermeidet wiederkehrende große Subgraph-Anfrage
  - Im Faktormodell: Faktor $\phi\left(\boldsymbol{I}^{(\tau)}\right)$ vor dem Bau des Jtrees zum Modell hinzufügen
    - Potentiale in ϕ irrelevant für Bau, können also unspezifiziert gelassen werden
    - ϕ kann nach dem Bau entfernt werden
      - Wenn nicht entfernt, Potentiale in ϕ uniform verteilt
  - Im BN: Beim Moralisieren des BNs, $\boldsymbol{I}^{(\tau)}$ vollständig miteinander verbinden
  - Stellt sicher, dass ein Cluster $\boldsymbol{C}$ $\boldsymbol{I}^{(\tau)}$ enthält (Jtree Eigenschaft 2), i.e., $\boldsymbol{I}^{(\tau)} \subseteq \boldsymbol{C}$
    - Dieses Cluster trennt die Zeitscheiben voneinander
    - So genanntes Outcluster, da hier Nachricht $\alpha^{(t)}$ über $\boldsymbol{I}^{(\tau)}$ berechnet und losgeschickt wird (*outgoing*)



Jtrees und JT um mit der Zeit zu gehen

- Da $\alpha^{(t)}$ über $\boldsymbol{I}^{(t)}$ zu einem Cluster im Jtree für $t+1$ hinzugefügt werden muss, braucht man auch ein Cluster $\boldsymbol{C}$, welches das Interface zur vorherigen Zeitscheibe enthält
  - So genanntes Incluster, da hier Nachricht $\alpha^{(t)}$ über $\boldsymbol{I}^{(t)}$ ankommt (*incoming*)
  - Im Faktormodell: Auch ein Faktor $\phi\left(\boldsymbol{I}^{(\tau-1)}\right)$ vor dem Bau des Jtrees zum Modell hinzufügen
    - Es gilt wieder, Potentiale in ϕ irrelevant für Bau, können also unspezifiziert gelassen werden
    - ϕ kann nach dem Bau entfernt werden; wenn nicht entfernt, Potentiale in ϕ uniform verteilt
  - Im BN: Beim Moralisieren des BNs, $\boldsymbol{I}^{(\tau-1)}$ vollständig miteinander verbinden
  - Stellt sicher, dass ein Cluster $\boldsymbol{C}$ $\boldsymbol{I}^{(\tau-1)}$ enthält (Jtree Eigenschaft 2), i.e., $\boldsymbol{I}^{(\tau-1)} \subseteq \boldsymbol{C}$
  - In jeder $t > 0$ Zeitscheibe, In- und Outcluster vorhanden



Incluster & Outcluster: Gegenüberstellung

- Cluster, welches $\boldsymbol{I}^{(\tau-1)}$ enthält→ Incluster $\boldsymbol{C}_{in}^{(\tau)}$
  - Aus der Perspektive vonτ: separiert Vergangenheit von der Gegenwart
  - Erhält eingehende Informationen vom Outcluster der vorherigen Zeitscheibe
- Cluster, welches $\boldsymbol{I}^{(\tau)}$ enthält→ Outcluster $\boldsymbol{C}_{out}^{(\tau)}$
  - Aus der Perspektive von τ: separiert Gegenwart von der Zukunft
  - Schickt Informationen raus zum Incluster der nächsten Zeitscheibe
  - Gegenwart wird zur Vergangenheit im nächster Schritt

Dynamische Jtrees

- Gegeben ein Interface $\boldsymbol{I}^{(\tau)}$ für ein dynamisches Modell $M = (M^0, M^{\rightarrow})$
- Baue zwei Jtrees $(J^0, J^{\rightarrow})$ für M
  - J^0 für $M^0 \cup \{\phi(\boldsymbol{I}^{(0)})\}$
    - $\boldsymbol{I}^{(0)} = \boldsymbol{I}^{(\tau)|\tau=0}$ in einem Cluster (Outcluster)
  - $J^{\rightarrow}$ für $M^{\rightarrow} \cup \{\phi(\boldsymbol{I}^{(\tau-1)}), \phi(\boldsymbol{I}^{(\tau)})\}$
    - $\boldsymbol{I}^{(\tau-1)} = \boldsymbol{I}^{(\tau)|\tau=\tau-1}$ in einem Cluster (Incluster)
    - $\boldsymbol{I}^{(\tau)}$ in einem Cluster (Outcluster)
    - Template-Jtree, der instanziiert wird, indem τ mit einem Zeitpunkt t ersetzt wird
  - Standardverfahren zum Bau der Jtrees verwenden
    - Z.B. Dtree bauen, Cluster bestimmen, Cluster in Jtree umwandeln, minimieren

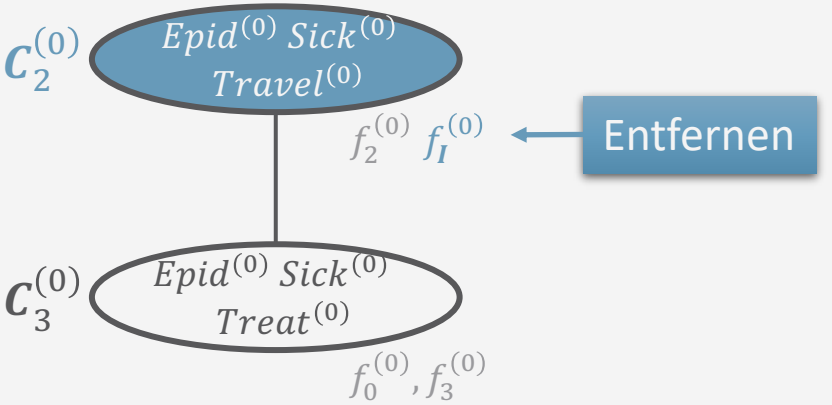
Jtree $J = (V, E)$ ein ungerichteter, azyklischer Graph mit V eine Menge von Clustern und E eine Menge von Kanten für ein Modell M; muss drei Eigenschaften erfüllen:

1. Cluster bestehen aus Variablen aus M
2. Variablen jedes Faktors / CPD in einem Cluster enthalten
3. *Running intersection property*

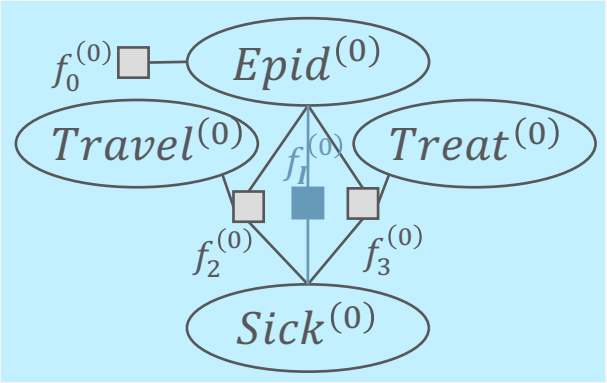
Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

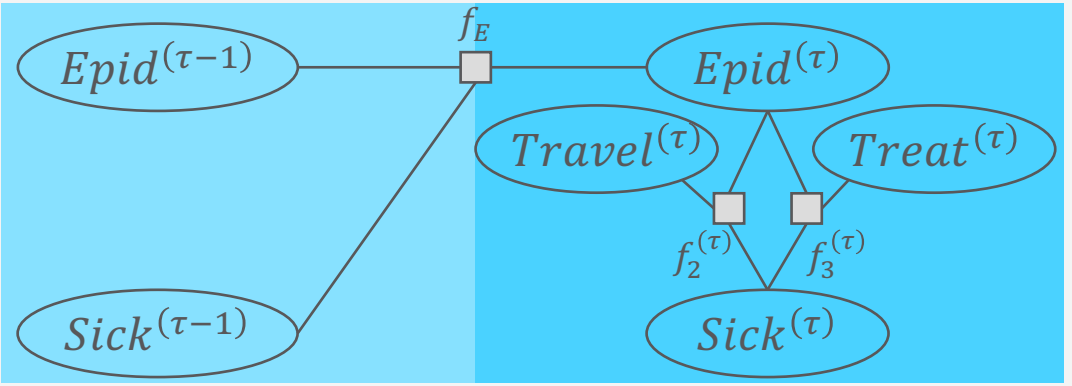
Dynamische Jtrees: Beispiel

- Dynamisches Faktormodell $F = (F^0, F^{\rightarrow})$
  - $F^0 = \{f_2^{(0)}, f_3^{(0)}\} \cup \{f_0^{(0)}\}, F^{\rightarrow} = \{f_2^{(\tau)}, f_3^{(\tau)}\} \cup \{f_E\}$
  - Interface $\boldsymbol{I}^{(\tau)} = \{Epid^{(\tau)}, Sick^{(\tau)}\}$



- J^0 für $F^0 \cup \{f_I^{(0)}\}$
  - $f_I^{(0)} = \phi(E^{(0)}, S^{(0)})$
  - Beide Cluster enthalten Interface $\boldsymbol{I}^{(0)}$
    - In der Regel wird das gewählt, welches $f_I^{(0)}$ enthält: $\boldsymbol{C_2^{(0)} = C_{out}^{(0)}}$

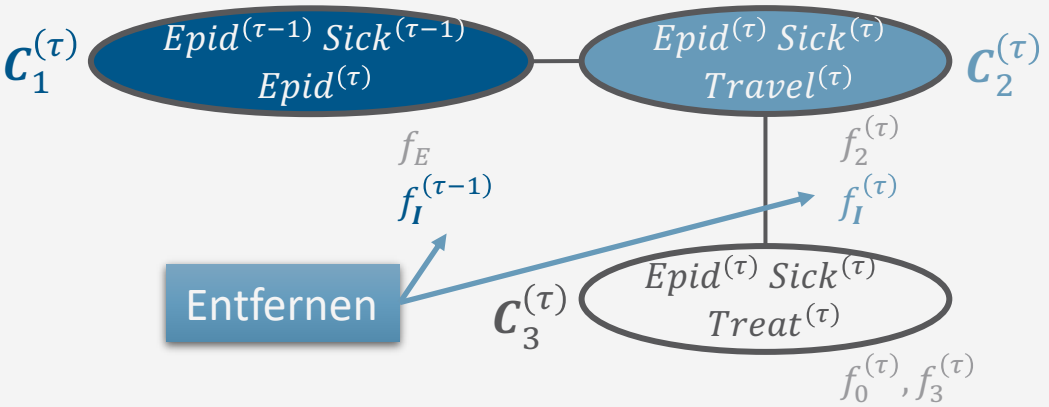




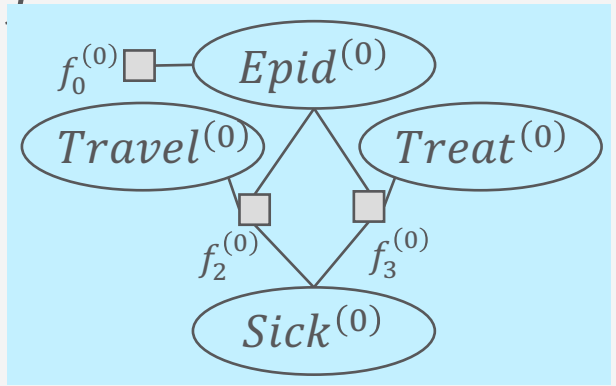
$f_I^{(0)}$ täte hier nicht Not, da $\boldsymbol{I}^{(0)}$ schon zusammen in Faktoren vorkommt.

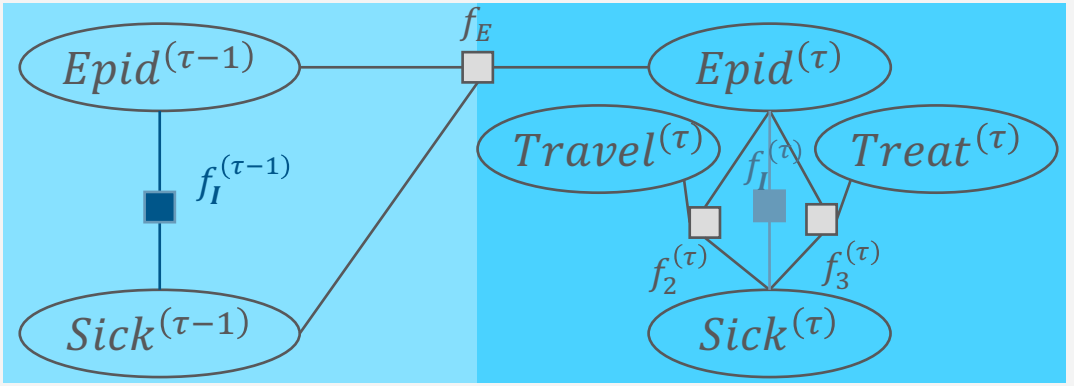
Dynamische Jtrees: Beispiel

- Dynamisches Faktormodell $F = (F^0, F^{\rightarrow})$
  - $F^0 = \{f_2^{(0)}, f_3^{(0)}\} \cup \{f_0^{(0)}\}, F^{\rightarrow} = \{f_2^{(\tau)}, f_3^{(\tau)}\} \cup \{f_E\}$
  - Interface $\boldsymbol{I}^{(\tau)} = \{Epid^{(\tau)}, Sick^{(\tau)}\}$



- $J^{\rightarrow}$ für $F^{\rightarrow} \cup \{f_I^{(\tau-1)}, f_I^{(\tau)}\}$
  - $f_I^{(\tau-1)} = \phi(E^{(\tau-1)}, S^{(\tau-1)})$
  - $f_I^{(\tau)} = \phi(E^{(\tau)}, S^{(\tau)})$
  - Incluster: $\boldsymbol{C}_1^{(\tau)} = \boldsymbol{C}_{in}^{(\tau)}$
  - Outcluster: $\boldsymbol{C}_2^{(\tau)} = \boldsymbol{C}_{out}^{(\tau)}$





Gleiches gilt auch hier für $f_I^{(\tau-1)}, f_I^{(\tau)}$.

Dynamische Jtrees: Ausrollen

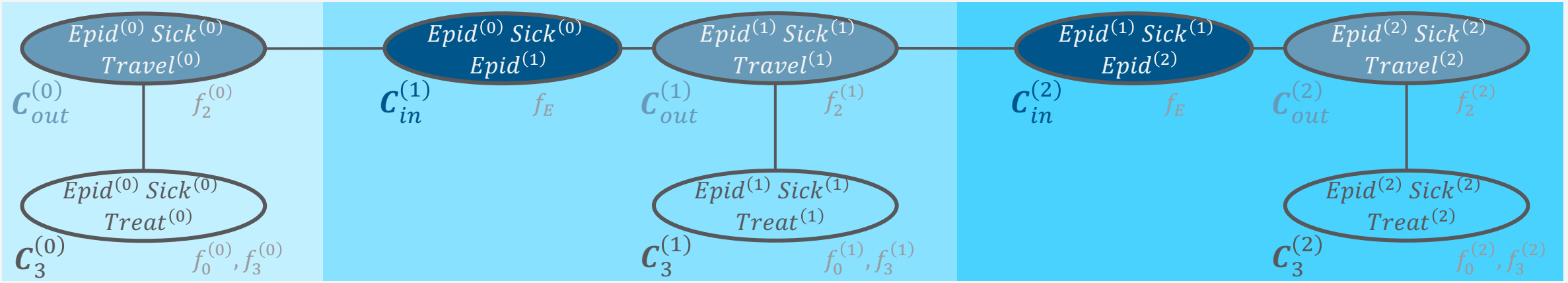
- Gegeben zwei Jtrees $(J^0, J^{\rightarrow})$ für ein dynamische Modell $M = (M^0, M^{\rightarrow})$ und Zahl T, rolle $(J^0, J^{\rightarrow})$ aus durch Bilden eines Jtrees $J^{(0:T)}$ aus Jtrees $J^{(0)}, J^{(1)}, \dots, J^{(T)}$, wobei
  - $J^{(0)} = \left(V^{(0)}, E^{(0)}\right) = J^0$
  - $J^{(t)} = \left(V^{(t)}, E^{(t)}\right) = J^{\rightarrow|\tau=t}$ für alle $J^{(t)}$, $t > 0$
  - Für alle Zeitscheiben $t-1, t$ für $t > 0$, eine Kante zwischen dem Outcluster von $t-1$ und dem Incluster von t eingefügt wird
  - Formal: $J^{(0:T)} = (V, E)$ mit

$$V = V^{(0)} \cup \bigcup_{t=1}^{T} V^{(t)}$$

$$E = E^{(0)} \cup \bigcup_{t=1}^{T} E^{(t)} \cup \bigcup_{t=1}^{T} \left\{ \left\{ \boldsymbol{C}_{out}^{(t-1)}, \boldsymbol{C}_{in}^{(t)} \right\} \middle| \boldsymbol{C}_{out}^{(t-1)} \wedge \boldsymbol{C}_{in}^{(t)} \right\}$$

Dynamische Jtrees: Ausrollen – Beispiel

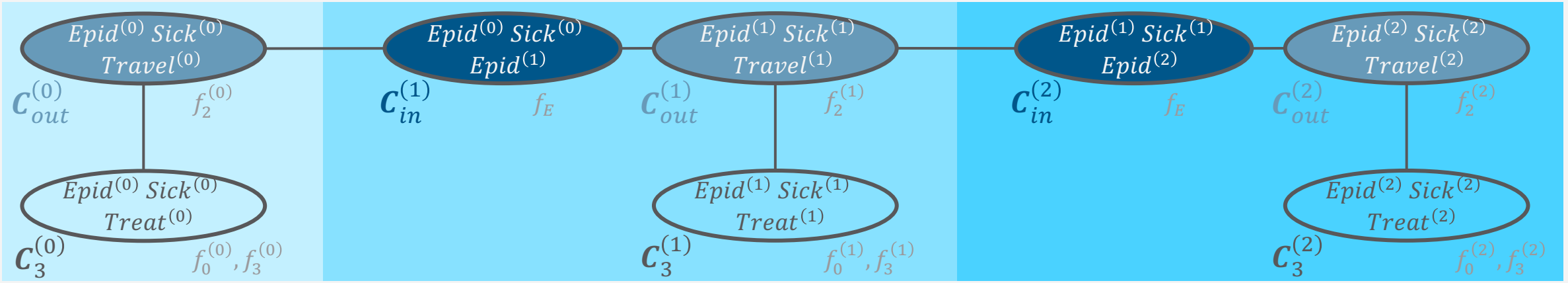
- $T = 2$



- JT für Anfragenbeantwortung nutzen
  - Evidenz behandeln, Nachrichten verschicken, Anfragen beantworten
- Probleme (wie beim Modell ausrollen)
  - Ausgerollter Jtree sehr groß
  - Neu ausrollen (oder anpassen), wenn sich $T = \max\{t, \pi\}$ oder $\boldsymbol{e}^{(1:t)}$ ändern

Dynamische Jtrees: Ausrollen – Beispiel

- $T = 2$



- Ausrollen zur Anfragenbeantwortung innerhalb einer Zeitscheibe unnötig
  - Durch die Interfaces in den In- und Outclustern sind die Zeitscheiben unabhängig von einander
  - Anfragenbeantwortung innerhalb einer Zeitscheibe mittels JT
  - Übergang zur nächsten Zeitscheibe: Nachricht in Outcluster $C_{out}^{(t)}$ über $I^{(t)}$ berechnen und zum Incluster $C_{in}^{(t+1)}$ im neu instanziierten Jtree für $t + 1$ hinzufügen

Mit der Zeit gehen

- Anfragebeantwortung innerhalb einer Zeitscheibe t folgt JT
  - Menge von Anfragen für t bei Evidenz bis einschließlich t effizient beantworten
  - Stellt auch sicher, dass alle notwendigen Informationen bei $\boldsymbol{C}_{out}^{(t)}$ verfügbar sind, um die Vorwärtsnachricht zu berechnen, die $t+1$ unabhängig von t macht
- Übergang zur nächsten Zeitscheibe:
  - Berechne Vorwärtsnachricht $\alpha^{(t)}$ über den Separator zwischen $\boldsymbol{C}_{out}^{(t)}$ und $\boldsymbol{C}_{in}^{(t+1)}$ basierend auf dem lokalen Modell $F_{out}^{(t)}$ und den eingegangenen Nachrichten $m_{j,out}^{(t)}$
  - Instanziiere $J^{\rightarrow}$ für $t+1$
  - Füge $\alpha^{(t)}$ zum lokalen Modell von $\boldsymbol{C}_{in}^{(t+1)}$ hinzu
  - Lasse $J^{(t)}$ fallen
  - Mache Anfragebeantwortung in $J^{(t+1)}$ (Schritte 2-4 von JT)

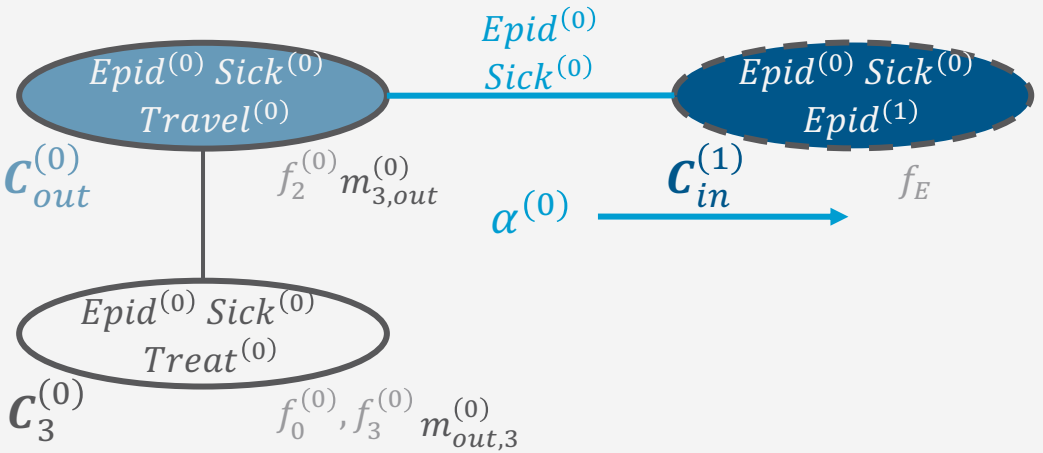
Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

Mit der Zeit gehen: Beispiel

- $t = 0$
  - aktueller Jtree
  - Evidenz $\boldsymbol{e}^{(0)} = \{treat^{(0)}\}$ behandeln
  - Intra-Zeitscheiben-Nachrichten: $m_{3,out}^{(0)}$, $m_{out,3}^{(0)}$
  - Beantworte Anfragen $P\left(R_i^{(0)} \middle| \boldsymbol{e}^{(0)}\right)$
- Inter-Zeitscheiben-Nachricht: $\alpha^{(0)}$
  - Eliminiere alle Nicht-Separator-Variablen, $Travel^{(0)}$, aus dem lokalen Modell $F_{out}^{(0)} = \{f_2^{(0)}\}$ und $m_{3,out}^{(0)}$
  - Ergebnis als Nachricht $\alpha^{(0)}$ zu $\boldsymbol{C}_{in}^{(1)}$

$\alpha^{(0)}$ beinhaltet alle Informationen aus $F^{(0)}$ inklusive $\boldsymbol{e}^{(0)}$, was Schritt 0 und 1 unabhängig macht.

Als nächstes: Instanziiere einen Jtree für $\tau = 1$ und füge $\alpha^{(0)}$ zum lokalen Modell von $\boldsymbol{C}_{in}^{(1)}$ hinzu.

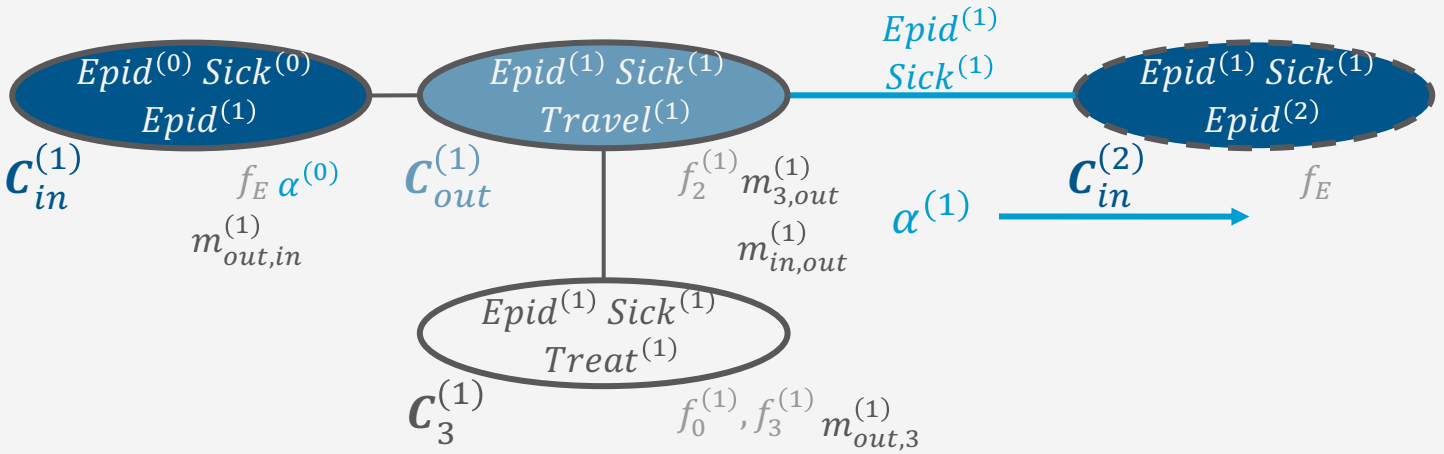


Mit der Zeit gehen: Beispiel

- $t = 1$
  - aktueller Jtree
  - Evidenz $\boldsymbol{e}^{(1)} = \{treat^{(1)}\}$ behandeln
  - Intra-Z.-N.: $m_{3,out}^{(1)}, m_{out,3}^{(1)}, m_{in,out}^{(1)}, m_{out,in}^{(1)}$
  - Beantworte Anfragen $P\left(R_i^{(1)} \middle| \boldsymbol{e}^{(0:1)}\right)$
- Inter-Zeitscheiben-Nachricht: $\alpha^{(1)}$
  - Eliminiere $Travel^{(1)}$ aus dem lokalen Modell $F_{out}^{(1)} = \{f_2^{(1)}\}$, $m_{in,out}^{(1)}$ und $m_{3,out}^{(1)}$
  - Ergebnis als Nachricht $\alpha^{(1)}$ zu $\boldsymbol{C}_{in}^{(2)}$

Während des Nachrichtenversands werden die Informationen aus $\alpha^{(0)}$ verteilt und dadurch in $\alpha^{(1)}$ mit verrechnet. $\alpha^{(1)}$ beinhaltet nun alle Informationen aus $F^{(0:1)}$ inklusive $\boldsymbol{e}^{(0:1)}$, was Schritt 1 und 2 unabhängig macht.

Als nächstes: Instanziiere einen Jtree für $\tau = 2$ und füge $\alpha^{(1)}$ zum lokalen Modell von $\boldsymbol{C}_{in}^{(2)}$ hinzu.

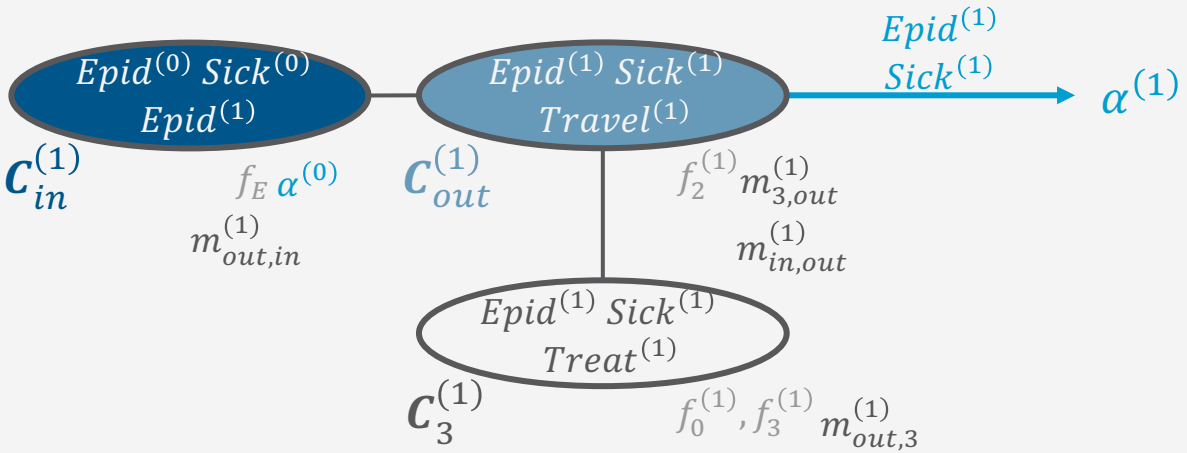


Filtern / Überwachen (*filtering*)

- Anfragen, die auf diese Art beantwortet werden: Filterungsanfragen (*filtering*)
  - $P\left(R^{(t)} \middle| \boldsymbol{e}^{(0:t)}\right)$
  - Anfragen an Zufallsvariablen der aktuellen Zeitscheibe t gegeben die bisherige Evidenz $\boldsymbol{e}^{(0:t)}$
  - Vorteile
    - Nur ein aktueller Jtree nötig
    - Eine zusätzliche Nachricht ($\alpha^{(t)}$), um in der Zeit voranzuschreiten

- Was ist mit Vorhersagen (*prediction*) und Rückschauen (*hindsight*)?
  - $P\left(R^{(\pi)} \middle| \boldsymbol{e}^{(0:t)}\right)$, $\pi \neq t$

Hat jemand eine Idee?



Vorhersagen (*prediction*)

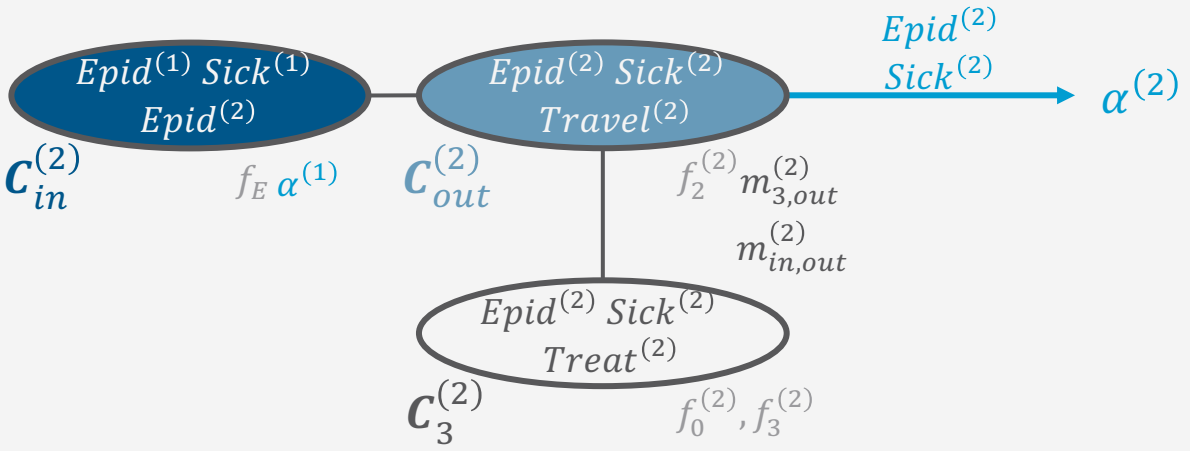
- Vorhersagen: Anfragen an zukünftige Instanzen von Zufallsvariablen $P\left(R^{(\pi)}\middle|\boldsymbol{e}^{(0:t)}\right), \pi > t$
  - Vorausschau in die Zukunft: sich in der Zeit bis π vorwärts bewegen ohne Evidenz zu sehen, i.e.,
    - Filterung: $P\left(R^{(\pi)}\middle|\boldsymbol{e}^{(0:\pi)}\right)$ mit leerer Evidenz zwischen $t+1$ und π in $\boldsymbol{e}^{(0:\pi)}$: $\boldsymbol{e}^{((t+1):\pi)} = \left\{\emptyset^{(t')}\right\}_{t'=t+1}^{\pi}$
- Anfragebeantwortung für Vorhersagen
  - Gehe vorwärts in der Zeit bis $t = \pi$
    - Eine Nachrichten-Phase von der Peripherie zum Zentrum mit *Outcluster* als Zentrum ausreichend
  - Beantworte Anfrage mit Anfragevariable $R^{(\pi)}$
    - Wenn nur eine Anfrage
      - Finde Cluster $\boldsymbol{C}_i^{\pi}$, welches $R^{(\pi)}$ enthält; führe eine Nachrichten-Phase von der Peripherie zum Zentrum mit $\boldsymbol{C}_i^{\pi}$ als Zentrum durch; beantworte Anfrage in $\boldsymbol{C}_i^{\pi}$
    - Sonst:
      - Vollständigen Nachrichtenversand (zwei Phasen) durchführen; Anfragen beantworten

Spart die Hälfte der Nachrichten

Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

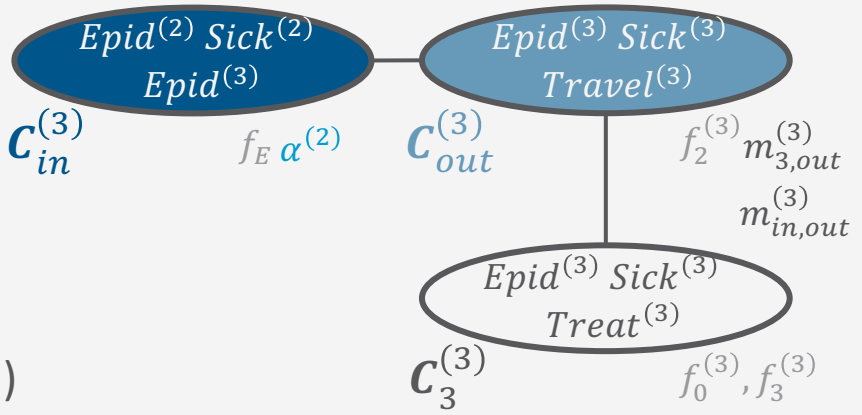
Vorhersagen (*prediction*): Beispiel

- $P\left(Epid^{(3)} \middle| \boldsymbol{e}^{(0:1)}\right)$, $3 > 1$
  - Keine Evidenz für Schritte 2 und 3: $\boldsymbol{e}^{(2:3)} = \left\{\emptyset^{(t')}\right\}_{t'=2}^{3}$
  - Gehe vorwärts in der Zeit bis $t = 3$, dann beantworte die Anfrage mit Anfragevariable $Epid^{(3)}$
  - $t = 2$, aktueller Jtree mit $\alpha^{(1)}$ im lokalen Modell von $\boldsymbol{C}_{in}^{(2)}$
    - Keine Evidenz zu behandeln: $\emptyset^{(2)}$
    - Versende Intra-Zeitscheiben-Nachrichten in Richtung $\boldsymbol{C}_{out}^{(2)}$ (alle Informationen am Outcluster sammeln)
    - Berechne Inter-Zeitscheiben-Nachricht $\alpha^{(2)}$



Vorhersagen (*prediction*): Beispiel

- $P\left(Epid^{(3)} \middle| \boldsymbol{e}^{(0:1)}\right)$, $3 > 1$
  - Keine Evidenz für Schritte 2 und 3: $\boldsymbol{e}^{(2:3)} = \left\{\emptyset^{(t')}\right\}_{t'=2}^{3}$
  - Gehe vorwärts in der Zeit bis $t = 3$, dann beantworte die Anfrage mit Anfragevariable $Epid^{(3)}$
  - $t = 3$, aktueller Jtree mit $\alpha^{(2)}$ im lokalen Modell von $\boldsymbol{C}_{in}^{(3)}$
    - Keine Evidenz zu behandeln: $\emptyset^{(3)}$
    - Wenn nur Anfrage zu $Epid^{(3)}$
      - Versende Intra-Zeitscheiben-Nachrichten in Richtung $\boldsymbol{C}_{out}^{(3)}$ (oder $\boldsymbol{C}_{3}^{(3)}$)
      - Beantworte Anfrage in $\boldsymbol{C}_{out}^{(3)}$
    - Sonst:
      - Alle Nachrichten versenden (auch $m_{out,3}^{(3)}$, $m_{out,in}^{(3)}$)



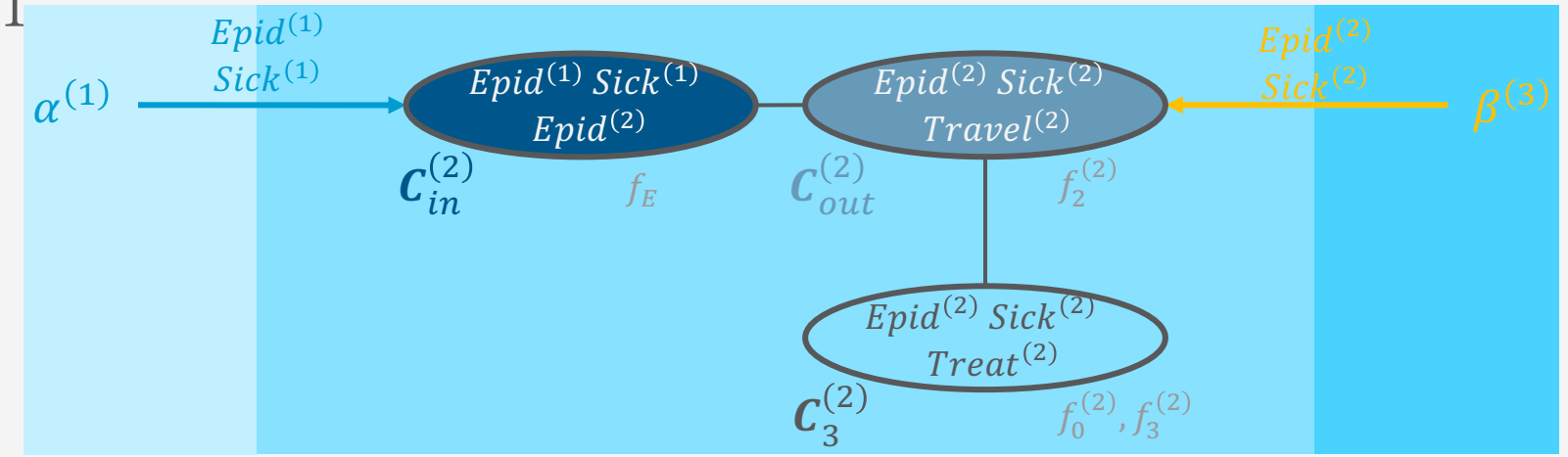
Rückschau (*hindsight*): Wieder zurück in der Zeit

- Rückschau: Anfrage an vergangene Instanzen von Zufallsvariablen $P\left(R^{(\pi)}\middle|\boldsymbol{e}^{(0:t)}\right), \pi < t$
  - Zurückblicken mit dem Wissen von jetzt (t) bezogen auf Evidenz, i.e.,
    - Von der jetzigen Zeitscheibe t rückwärts gehen und die Informationen, die sich zwischen π and t angesammelt haben, nach π bringen; dann eine Filterungsanfrage in π beantworten
- Anfragebeantwortung für Rückschauen
  - Bewege dich rückwärts in der Zeit bis $t = \pi$
    - Berechne dafür *Rückwärtsnachrichten* $\beta^{(t)}$ von $\boldsymbol{C}_{in}^{(t)}$ zu $\boldsymbol{C}_{out}^{(t-1)}$ beginnend bei t bis π
      - Eine Nachrichten-Phase von der Peripherie zum Zentrum mit *Incluster* als Zentrum ausreichend
    - Beantworte Anfrage mit Anfragevariable $R^{(\pi)}$ in J^{π} mit $\alpha^{(\pi-1)}$ im Incluster, $\beta^{(\pi+1)}$ im Outcluster
      - Wenn nur eine Anfrage
        - Eine Nachrichten-Phase von der Peripherie zu Cluster $\boldsymbol{C}_i^{\pi}$, $R^{(\pi)} \in \boldsymbol{C}_i^{\pi}$, als Zentrum; beantworte Anfrage in $\boldsymbol{C}_i^{\pi}$
      - Sonst: Vollständigen Nachrichtenversand (zwei Phasen) durchführen; Anfragen beantworten

Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

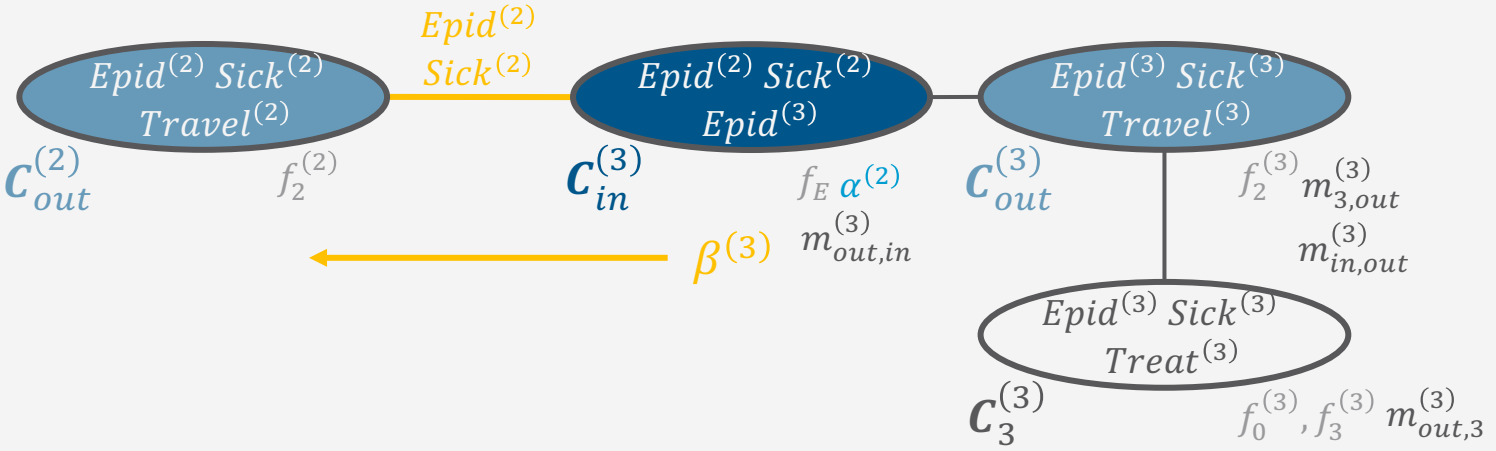
Rückschau (*hindsight*): Rückwärtsnachricht $\beta^{(t)}$

- Idee: Zeitscheibe unabhängig von der Zukunft machen (Vorwärtsnachricht umgekehrt)
- Aus der Perspektive von Zeitscheibe π irgendwo in der Sequenz von 0 bis t
  - Vorwärtsnachricht $\alpha^{(\pi-1)}$ beinhaltet alle Informationen zu $M^{(0:(\pi-1))}$ inklusive $\boldsymbol{e}^{(0:(\pi-1))}$
    - Macht $\pi - 1$ unabhängig von π
  - Rückwärtsnachricht $\beta^{(\pi+1)}$ beinhaltet alle Informationen zu $M^{((\pi+1):t)}$ inklusive $\boldsymbol{e}^{((\pi+1):t)}$
    - Macht π unabhängig von $\pi + 1$
- Immer von $\boldsymbol{C}_{in}^{(t)}$ zu $\boldsymbol{C}_{out}^{(t-1)}$ verschickt, bis $t = \pi$



Rückschau (*hindsight*): Rückwärtsnachricht $\beta^{(t)}$

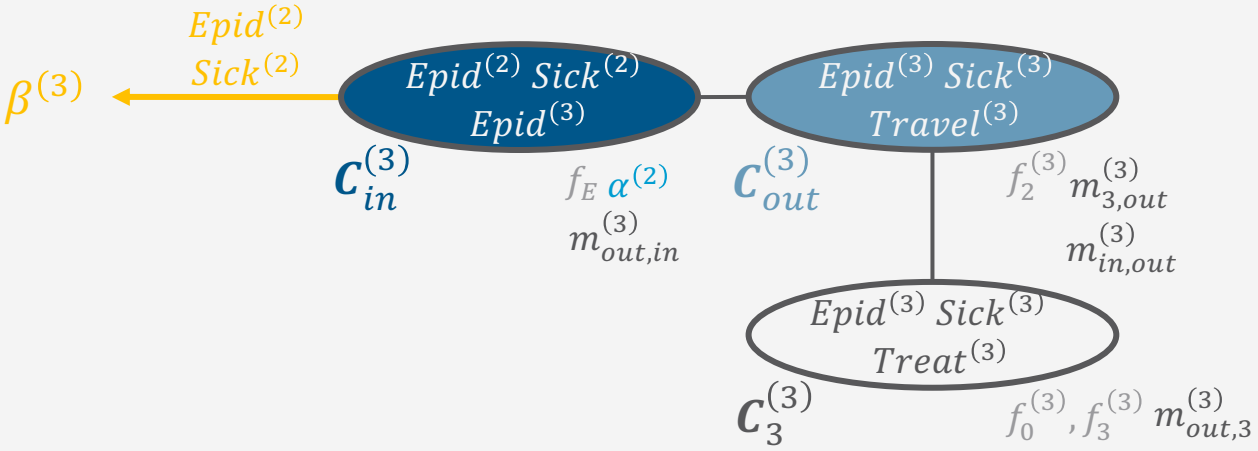
- Berechnung
  - Eliminiere alle Nichtseparator-Variablen zwischen $\boldsymbol{C}_{in}^{(t)}$ und $\boldsymbol{C}_{out}^{(t-1)}$ aus dem lokalen Modell $F_{in}^{(t)}$ und den eingehenden Nachrichten $m_{j,in}^{(t)}$
    - Für Rückwärtsnachricht nicht Vorwärtsnachricht $\alpha^{(t-1)}$ berücksichtigen, da $\alpha^{(t-1)}$ von $\boldsymbol{C}_{out}^{(t-1)}$ kam
      - Diese Information schon in Zeitscheibe $t-1$ vorhanden
- Beispiel:
  - Aktuelle Zeitscheibe $t = 3$
  - Berechne $\beta^{(3)}$ durch Eliminierung von $Epid^{(3)}$ aus $f_E, m_{out,in}^{(3)}$
    - Ohne $\alpha^{(2)}$



Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

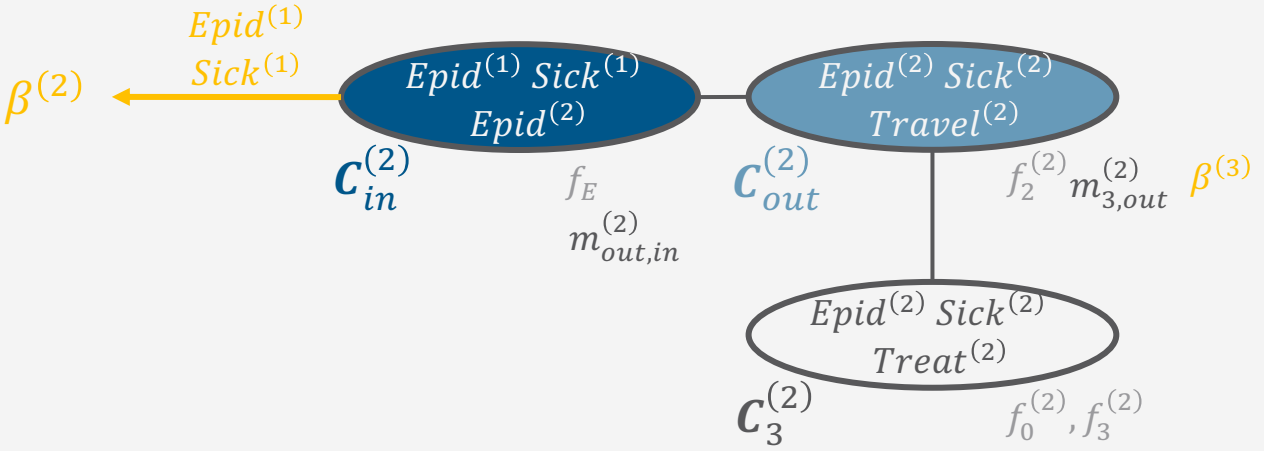
Rückschau (*hindsight*): Beispiel

- $P\left(Epid^{(1)} \middle| \boldsymbol{e}^{(0:3)}\right)$, $1 < 3$
  - Gehe rückwärts in der Zeit bis $t = 1$, dann beantworte die Anfrage mit Anfragevariable $Epid^{(1)}$
  - $t = 3$, aktueller Jtree mit $\alpha^{(2)}$ im lokalen Modell von $\boldsymbol{C}_{in}^{(3)}$
    - Berechne Rückwärtsnachricht $\beta^{(3)}$
    - Instanziiere Jtree für $t = 2$, füge $\beta^{(3)}$ zu lokalem Modell von $\boldsymbol{C}_{out}^{(2)}$ hinzu



Rückschau (*hindsight*): Beispiel

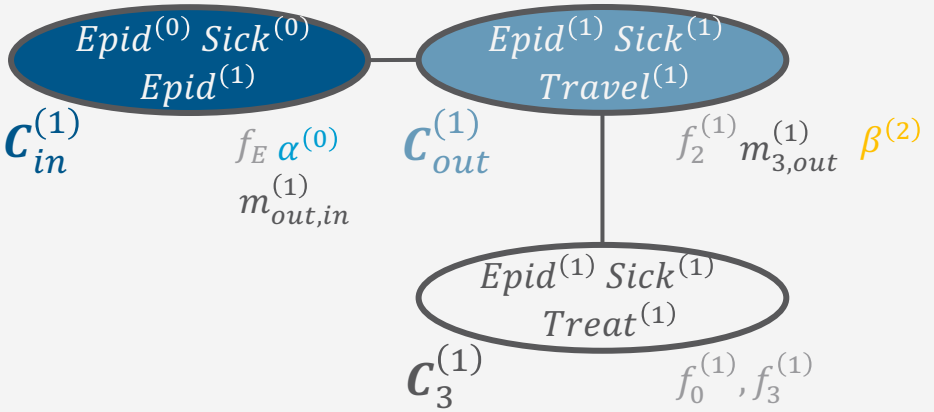
- $P\left(Epid^{(1)} \middle| \boldsymbol{e}^{(0:3)}\right), 1 < 3$
  - Gehe rückwärts in der Zeit bis $t = 1$, dann beantworte die Anfrage mit Anfragevariable $Epid^{(1)}$
  - $t = 2$, aktueller Jtree mit $\beta^{(3)}$ im lokalen Modell von $\boldsymbol{C}_{out}^{(2)}$
    - Behandle Evidenz $\boldsymbol{e}^{(2)}$
    - Sende Nachrichten in Richtung $\boldsymbol{C}_{in}^{(2)}$
    - Berechne Rückwärtsnachricht $\beta^{(2)}$
    - Instanziiere Jtree für $t = 1$, füge $\beta^{(2)}$ zu lokalem Modell von $\boldsymbol{C}_{out}^{(2)}$ hinzu



Rückschau (*hindsight*): Beispiel

- $P\left(Epid^{(1)} \middle| \boldsymbol{e}^{(0:3)}\right)$, $1 < 3$
  - Gehe rückwärts in der Zeit bis $t = 1$, dann beantworte die Anfrage mit Anfragevariable $Epid^{(1)}$
  - $t = 1$, aktueller Jtree mit $\beta^{(2)}$ im lokalen Modell von $\boldsymbol{C}_{out}^{(1)}$, $\alpha^{(0)}$ im lokalen Modell von $\boldsymbol{C}_{in}^{(1)}$
    - Behandle Evidenz $\boldsymbol{e}^{(1)}$
    - Wenn nur Anfrage zu $Epid^{(1)}$
      - Versende Intra-Zeitscheiben-Nachrichten in Richtung $\boldsymbol{C}_{in}^{(1)}$ (oder die anderen beiden)
      - Beantworte Anfrage in $\boldsymbol{C}_{in}^{(1)}$
    - Sonst: Alle Nachrichten versenden

Um Rückschau-Anfragen zu beliebigen Zeitscheiben π zu ermöglichen, müssen wir uns die Vorwärtsnachrichten merken, da $\alpha^{(\pi-1)}$ für die Anfragebeantwortung in π benötigt wird.



Allgemeine Anfragenbeantwortung

- Vorwärts-Rückwärts-Algorithmus für die Anfragentypen Filterung, Vorhersage, Rückschau
- Gegeben: Menge von Anfragen $\boldsymbol{Q}^{(0:T_q)} = \left\{\left\{\boldsymbol{Q}_i^{(\pi_i)}\right\}_{i=1}^{m_t}\right\}_{t=0}^{T_q}$
  - Bei anwachsender Evidenz $\boldsymbol{e}^{(0:t)}, t \in \{0, \dots, T_e\}$
  - Quasi zwei Ströme an Anfragen und Evidenz über die Zeit t
  - Aus Effizienzgründen, gehe durch $\boldsymbol{Q}^{(0:T_q)}$ mit ansteigendem t und durch die Anfragen $\left\{\boldsymbol{Q}_i^{(\pi_i)}\right\}_{i=1}^{m_t}$ bei gegebener Evidenz $\boldsymbol{e}^{(0:t)}$ basierend auf dem Typ und π_i
    1. Filterungsanfragen $\boldsymbol{Q}_i^{(\pi_i)}, \pi_i = \mathrm{t}$
    2. Vorhersage-Anfragen $\boldsymbol{Q}_i^{(\pi_i)}$, geordnet nach ansteigendem π_i ($\pi_i > t$)
    3. Rückschau-Anfragen $\boldsymbol{Q}_i^{(\pi_i)}$, geordnet nach absteigendem π_i ($\pi_i < t$)
    - Reihenfolge von 2. und 3. könnte man tauschen

Kevin Murphy: Dynamic Bayesian Networks: Representation, Inference and Learning. *Dissertation*, University of California, Berkeley, 2022.

Allgemeine Anfragenbeantwortung: Genaues Vorgehen bei Anfragen

- Gegeben t als die aktuelle Zeitscheibe, Anfragevariablen $\left\{\boldsymbol{Q}_i^{(\pi_i)}\right\}_{i=1}^{m_t}$ und einem abgeschlossenen Nachrichtenversand in t
  - Für alle $\boldsymbol{Q}_i^{(\pi_i)}$ mit $t = \pi_i$, beantworte $\boldsymbol{Q}_i^{(\pi_i)}$ in $J^{(t)}$
  - Behalte $J^{(t)}$ und bewege dich mit $t' = t$ in der Zeit
  - Für alle $\boldsymbol{Q}_i^{(\pi_i)}$ mit $t < \pi_i$, bewege dich ohne Evidenz vorwärts in der Zeit bis $t' = \max_i \pi_i$:
    - Instanziiere Jtree $J^{(t')}$ und berechne entsprechende Nachrichten
    - Wann immer $t' = \pi_i$, beantworte $\boldsymbol{Q}_i^{(\pi_i)}$
  - Für alle $\boldsymbol{Q}_i^{(\pi_i)}$ mit $t > \pi_i$, bewege dich rückwärts in der Zeit bis $t' = \min_i \pi_i$
    - Instanziiere Jtree $J^{(t')}$, behandle Evidenz $\boldsymbol{e}^{(t')}$ und berechne entsprechende Nachrichten
    - Wann immer $t' = \pi_i$, beantworte $\boldsymbol{Q}_i^{(\pi_i)}$

Welche Nachrichten müssen wir berechnen?

Dynamischer Jtree Algorithmus (DJT) – AKA *Murphy's Interface Algorithm*

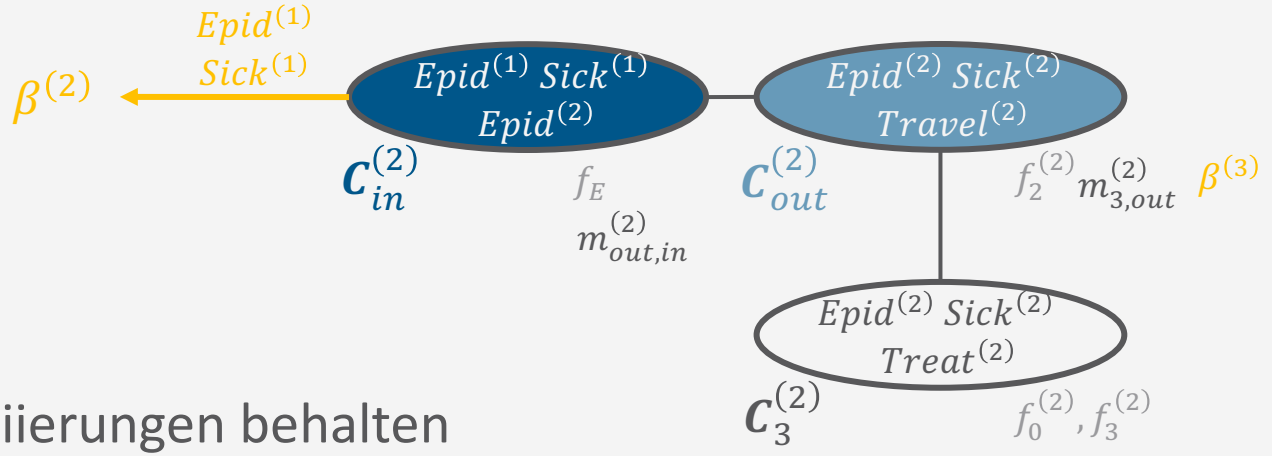
procedure DJT$((M^0, M^{\rightarrow}), \boldsymbol{Q}^{(0:T_q)}, \boldsymbol{e}^{(0:T_e)})$
- Baue Jtrees $(J^0, J^{\rightarrow})$ für $M^0, M^{\rightarrow}$
- **for** t **in** $0 \dots T_q$ **do**
  - Instanziiere $J^{(t)}$ und lösche $J^{(t-1)}$ ▹ J^0 bei $t = 0$; $J^{\rightarrow}$ sonst
  - Füge $\alpha^{(t-1)}$ zum Incluster von $J^{(t)}$ hinzu ▹ falls $t > 0$
  - Behandle Evidenz $\boldsymbol{e}^{(t)}$ in $J^{(t)}$
  - Verschicke Nachrichten in $J^{(t)}$
  - Beantworte Anfragen $\boldsymbol{Q}^{(t)}$ ▹ Vorgehen auf vorheriger Folie
  - Berechne $\alpha^{(t)}$

DJT

- Bei Online-Inferenz, zwei Ströme für eingehende Anfragen $\boldsymbol{Q}^{(t)}$ und Evidenz $\boldsymbol{e}^{(t)}$
  - Anstatt Eingaben $\boldsymbol{Q}^{(0:T_q)}, \boldsymbol{e}^{(0:T_e)}$

Vorgehensweisen beim Rückwärtsgehen

- Für beliebige Rückschau-Anfragen müssen Vorwärtsnachrichten gespeichert werden
  - Vorwärtsnachricht im Incluster gebraucht für Rückschau-Anfragen an diesen Zeitschritt
- Rückwärts gehen bedeutet zwischendrin Nachrichten erneut zu berechnen
  - Jeder Schritt t rückwärts ohne Anfrage
    - Berechne eine Nachrichtenphase zum Incluster mit $n-1$ Nachrichten, n Anzahl an Cluster in J_t
  - Jeder Schritt t rückwärts mit Anfragen
    - Berechne vollständigen Nachrichtenversand mit $2(n-1)$ Nachrichten
  - Vorteil
    - Nur ein Jtree im Speicher zur Zeit
- Was, wenn DJT die Jtrees behalten hätte?
  - Bedarfsweise instanziieren (bisher) vs. Instanziierungen behalten

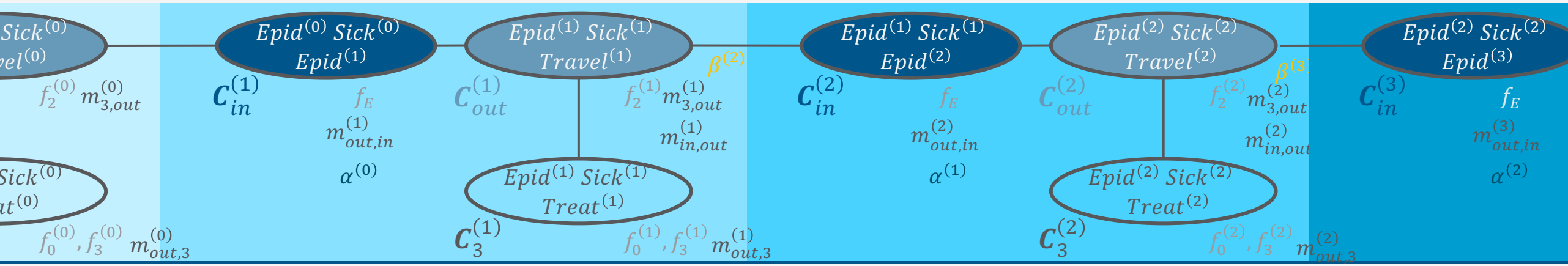


Instanziierungen behalten

- Was zahlt DJT an Speicher?
  - Lokale Modelle, Intra-Zeitscheiben-Nachrichten für jede Zeitscheibe speichern
- Wie viel Aufwand kann DJT einsparen?
  - Evidenz behandeln nicht nötig, da schon in lokalen Modellen absorbiert beim Vorwärtsgehen
  - Adaptiver Nachrichtenversand: Nur Nachrichten aktualisieren, die durch β_{t+1} geändert
    - Jeder Schritt t rückwärts ohne Anfrage
      - β_{t+1} neue Information, die beim Outcluster ankommt und zum Incluster geschafft werden muss
      - Berechne eine Nachrichten auf dem Pfad zwischen Outcluster und Incluster → bis zu $n-1$ Nachrichten, n Anzahl an Cluster in J_t (Worst Case: alle Cluster in Reihe mit In- und Outcluster an den jeweiligen Enden)
    - Jeder Schritt t rückwärts mit Anfragen
      - β_{t+1} neue Information, die beim Outcluster ankommt und zu allen anderen Clustern geschafft werden muss
      - Berechne eine Nachrichtenphase vom Outcluster zur Peripherie mit $n-1$ Nachrichten, n Anzahl an Cluster in J_τ

Beispiel: $P\left(Treat^{(1)} \middle| \boldsymbol{e}^{(0:3)}\right)$ – Zu berechnende Nachrichten

- In $t = 2$ (keine Anfrage)
  - Instanziierungen behalten: $m^{(2)}_{out,in}$ (auf Pfad)
  - Bedarfsweise instanziieren: $m^{(2)}_{3,out}, m^{(2)}_{out,in}$ (einwärts nach C_2^{in})
- In $t = 1$ (mit Anfrage)
  - Instanziierungen behalten: $m^{(1)}_{out,3}$ (auswärts nach $\boldsymbol{C}_3^{(1)}$) + $m^{(1)}_{out,in}$ (für beliebige Anfragen)
  - Bedarfsweise instanziieren: $m^{(1)}_{in,out}, m^{(1)}_{out,3}$ (nach $\boldsymbol{C}_3^{(1)}$) + $m^{(1)}_{out,in}, m^{(1)}_{3,out}$ (für beliebige Anfragen)



Vorgehensweisen beim Rückwärtsgehen

	Instanziierungen behalten	Bedarfsweise instanziieren
Schritte ohne Anfragen	$\leq n-1$	$n-1$
Schritte mit beliebigen Anfragen	$n-1$	$2(n-1)$
Zusätzlicher Speicher	Lokale Modelle, Intra-Zeitscheiben-Nachrichten, $\alpha^{(t)}$ Nachrichten	$\alpha^{(t)}$ Nachrichten

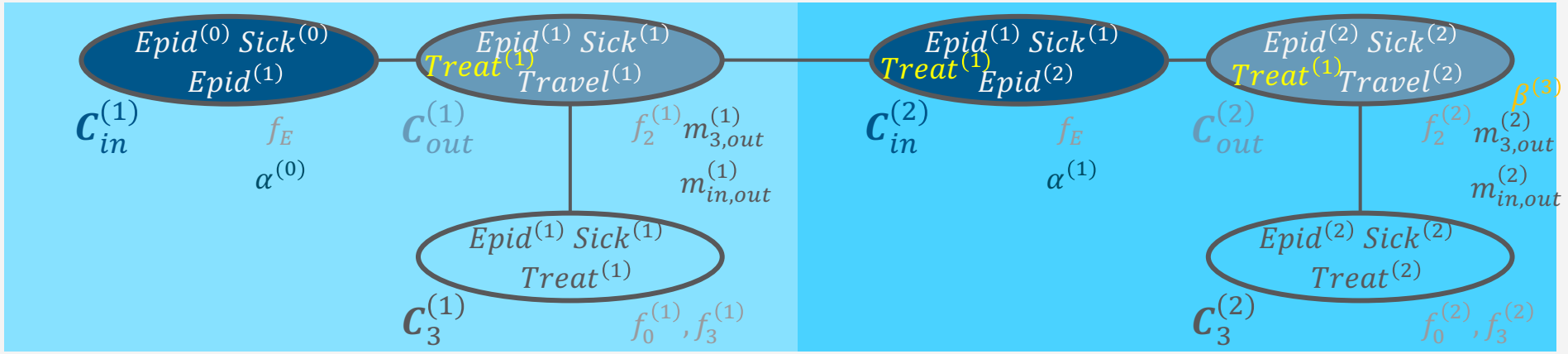
- Alle Instanziierungen im Speicher vorhalten nicht durchführbar
  - Ziel war ein speicher-effizienter Algorithmus, welcher nicht das gesamte Modell vom ersten Zeitschritt an im Speicher benötigt
- Laufzeit vs. Speicher gegeneinander abwägen
  - Letzte k Instanziierungen behalten
    - Daten / Anfragen für typischen Versatz in Anfragen analysieren
    - Bedarfsweise für Anfragen mit größerem Versatz instanziieren

Wahrscheinlichkeitsanfragen über Zeitschritte hinweg

- Anfragevariablen aus unterschiedlichen Zeitscheiben
  - Beispiel: $P\left(Treat^{(1)}, Travel^{(2)} \middle| \boldsymbol{e}^{(0:3)}\right)$ bei $t = 3 \rightarrow min = 1, max = 2$
- Anfragebeantwortung (eine von vielen Möglichkeiten):
  - Rolle Jtrees für Zeitschritte aus, die in der Anfrage vorkommen: $J^{(min:max)}$
    - $min =$ minimale Zeitscheibe in Anfrage
    - $max =$ maximale Zeitscheibe in Anfrage
    - $\alpha^{(min-1)}$ in Incluster von $J^{(min)}$; $\beta^{(max+1)}$ in Outcluster von $J^{(max)}$, wenn $max < t$
  - Füge Evidenz $\boldsymbol{e}^{(min:max)}$ ein
  - Füge alle Anfragevariablen zu Cluster $\boldsymbol{C}$ mit größter Überlappung mit Anfragevariablen hinzu
  - Stelle *running intersection property* durch Erweitern von Clustern wieder her
  - Führe eine einwärts Nachrichtenphase mit $\boldsymbol{C}$ als Zentrum durch und beantworte Anfrage in $\boldsymbol{C}$
    - Berechnet auch eigentliche α Nachrichten neu, da wir weniger Variablen eliminieren als sonst

Wahrscheinlichkeitsanfragen über Zeitschritte hinweg

- Anfragevariablen aus unterschiedlichen Zeitscheiben
  - Beispiel: $P\left(Treat^{(1)}, Travel^{(2)} \middle| \boldsymbol{e}^{(0:3)}\right)$ bei $t = 3 \rightarrow min = 1, max = 2$
- Anfragebeantwortung:
  - Jtrees ausrollen $(1 : 2)$ mit $\alpha^{(0)}$ in $\boldsymbol{C}_{in}^{(1)}$ und $\beta^{(3)}$ in $\boldsymbol{C}_{out}^{(2)}$, Evidenz eingeben
  - Anfragevariablen zu $\boldsymbol{C}_{out}^{(2)}$ hinzufügen
  - Running intersection property herstellen
  - Nachrichten zu $\boldsymbol{C}_{out}^{(2)}$ senden
  - Anfrage beantw.



Sequentielles MPE

- Wahrscheinlichste Zuweisungen an alle Zufallsvariablen $\boldsymbol{V}^{(0:T)}$ ohne Evidenz gegeben Evidenz $\boldsymbol{e}^{(0:T)}$

$$MPE_M\left(\boldsymbol{e}^{(0:T)}\right) = \underset{\boldsymbol{v}^{(0:t)} \in \mathrm{Val}\left(\boldsymbol{V}^{(0:T)}\right)}{\arg\max} P\left(\boldsymbol{v}^{(0:T)} \middle| \boldsymbol{e}^{(0:T)}\right)$$

  - $\boldsymbol{V}^{(0:T)} = \boldsymbol{R}^{(0:T)} \setminus \mathrm{rv}\left(\boldsymbol{e}^{(0:T)}\right)$
  - Auch als laufende Anfrage $MPE_M\left(\boldsymbol{e}^{(0:t)}\right)$ möglich
- Vorgehen ähnlich zu Filterungsanfragen
  - Für jeden Zeitschritt $t = 0, \dots, T$
    - Jtree $J^{(t)}$ instanziieren, Vorwärtsnachricht $\alpha^{(t-1)}$ zum Incluster von $J^{(t)}$ hinzufügen
    - Evidenz $\boldsymbol{e}^{(t)}$ behandeln
    - Nachrichten mit MPE-VE Operatoren berechnen und zum Outcluster von $J^{(t)}$ schicken
    - Vorwärtsnachricht $\alpha^{(t)}$ mit MPE-VE Operatoren berechnen

Nur eine Nachrichtenphase zum Outcluster nötig (wie bei MPE-JT)

Sequentielles MAP (und MPE)

- Wahrscheinlichste Zuweisungen an Sequenzen von Zufallsvariablen $\boldsymbol{U}^{(t_1:t_2)}$ gegeben Evidenz $\boldsymbol{e}^{(0:t)}$

$$MAP_M\left(\boldsymbol{U}^{(t_1:t_2)}\middle|\boldsymbol{e}^{(0:t)}\right) = \underset{\boldsymbol{u}^{(t_1:t_2)}\in \mathrm{Val}\left(\boldsymbol{U}^{(t_1:t_2)}\right)}{\arg\max} \sum_{\boldsymbol{v}^{(0:T)}\in \mathrm{Val}\left(\boldsymbol{V}^{(0:T)}\right)} P\left(\boldsymbol{u}^{(t_1:t_2)}, \boldsymbol{v}^{(0:T)}\middle|\boldsymbol{e}^{(0:t)}\right)$$

  - $\boldsymbol{V}^{(0:T)} = \boldsymbol{R}^{(0:T)} \setminus \boldsymbol{U}^{(t_1:t_2)} \setminus \mathrm{rv}\left(\boldsymbol{e}^{(0:t)}\right), T = \max\{t, t_1, t_2\}$
- Vorgehen wie bei MAP-JT
  - Jtrees über $t_1 : t_2$ ausrollen: $J^{(t_1:t_2)}$; $\alpha^{(t_1-1)}$ in Incluster von $J^{(t_1)}$; $\beta^{(t_2+1)}$ in Outcluster von $J^{(t_2)}$
  - Finde Subgraph über $\boldsymbol{U}^{(t_1:t_2)}$ in $J^{(t_1:t_2)}$
  - Schicke Nachrichten an den Rand des Subgraphen (mittels VE Operatoren berechnet)
  - Eliminiere $\boldsymbol{V}^{(0:T)}$ mit VE Operatoren im Modell des Subgraphen
  - Eliminiere $\boldsymbol{U}^{(t_1:t_2)}$ mit MPE-VE Operatoren im verbleibenden Modell

Sequentielles MAP (und MPE)

- Wahrscheinlichste Zuweisungen an Sequenzen von Zufallsvariablen $\boldsymbol{U}^{(t_1:t_2)}$ gegeben Evidenz $\boldsymbol{e}^{(0:t)}$

$$MAP_M\left(\boldsymbol{U}^{(t_1:t_2)}\middle|\boldsymbol{e}^{(0:t)}\right) = \underset{\boldsymbol{u}^{(t_1:t_2)}\in \mathrm{Val}\left(\boldsymbol{U}^{(t_1:t_2)}\right)}{\arg\max} \sum_{\boldsymbol{v}^{(0:T)}\in \mathrm{Val}\left(\boldsymbol{V}^{(0:T)}\right)} P\left(\boldsymbol{u}^{(t_1:t_2)}, \boldsymbol{v}^{(0:T)}\middle|\boldsymbol{e}^{(0:t)}\right)$$

  - $\boldsymbol{V}^{(0:T)} = \boldsymbol{R}^{(0:T)} \setminus \boldsymbol{U}^{(t_1:t_2)} \setminus \mathrm{rv}\left(\boldsymbol{e}^{(0:t)}\right), T = \max\{t, t_1, t_2\}$
- MAP Anfragen über die letzten k Zeitschritte $(t-k):t$ = MPE Anfrage über Zeitschritte$(t-k):t$ (wie bei MAP Anfragen an vollständige Subgraphen bei MAP-JT)
  - Bis $t-k$ Nachrichten mittels VE Operatoren berechnen
  - Ab $t-k$ Vorgehen wie bei MPE Anfragen
    - Nachrichten mit MPE-VE Operatoren berechnen und zum Outcluster von $J^{(t)}$ schicken
    - Vorwärtsnachricht $\alpha^{(t)}$ mit MPE-VE Operatoren berechnen

Komplexität

- JT Komplexität für Nachrichtenversand und Anfragebeantwortung (einfache Anfragen)

$$O_{MP} = O\left(n_J \cdot r^w\right)$$
$$O_{QA} = O\left(r^w\right)$$

Größt mögliches Zwischenergebnis

  - n_T, n_J Anzahl der Knoten im Dtree / Jtree
  - r größte Domäne
  - w Baumweite (größtes Cluster)
- DJT: JT + mit der Zeit gehen
  - Nachrichtenversand innerhalb einer Zeitscheibe → O_{MP}
  - Vorwärtsnachricht eine Anfrage an Outcluster → O_{QA}
  - Rückwärtsnachricht eine Anfrage an Incluster → O_{QA}

Was sind die Best-Case- und Worst-Case-Fälle bzgl. Anfragen?

Komplexität

- Gegeben maximaler Zeitschritt T, der in allen Anfragen vorkommt
  - m_t Anzahl an Anfragen pro Zeitschritt t, alle Anfragen: $M = \sum_{t=0}^{T} m_t$, durchschnittliche Anfragen pro Zeitschritt: $m = \frac{1}{T}\sum_{t=0}^{T} m_t$
  - Vorwärts gehen: Informationen verteilen + Vorwärtsnachricht, i.e., $T \cdot \left(O_{MP} + O_{QA}\right)$
  - *Best*-Worst-Case bzgl. Anfragen im Zeitschritt $t \in \{0, \dots, T\}$
    - Filterungsanfrage zu t, kostet je O_{QA}
  - Zusammen,

$$T \cdot \left(O_{MP} + O_{QA}\right) + M \cdot O_{QA} = \left(T \cdot n_J + T + M\right) \cdot O(r^w)$$
$$= O\left(\left(T \cdot n_J + M\right) \cdot r^w\right) = O\left(\left(T \cdot n_J + T \cdot m\right) \cdot r^w\right)$$

Vergleiche JT Komplexität: $O\left(\left(n_J + m\right) \cdot r^w\right)$

Komplexität

- *Worst*-Worst-Case bzgl. Anfragen im Zeitschritt $t \in \{0, \dots, T\}$
  - Rückschau-Anfrage für $\pi = 0$
    - Rückwärts-Nachrichtenphase zu $\pi = 0$: $t \cdot (O_{MP} + O_{QA})$
  - Vorhersagen-Anfrage für $\pi = T$
    - Vorwärts-Nachrichtenphase zu $\pi = T$: $(T - \pi) \cdot (O_{MP} + O_{QA})$
  - Zusammen führen eine Rückschau- und eine Vorhersagen-Anfragen zu einer Komplexität von
$$\pi \cdot (O_{MP} + O_{QA}) + (T - \pi) \cdot (O_{MP} + O_{QA})$$
$$= T \cdot (O_{MP} + O_{QA})$$
  - Für m Anfragen pro Zeitscheibe haben wir
$$T \cdot (O_{MP} + O_{QA}) + m \cdot O_{QA}$$
- Für T Zeitschritte haben wir
$$T \cdot T \cdot (O_{MP} + O_{QA}) + T \cdot m \cdot O_{QA} = T^2 \cdot (O_{MP} + O_{QA}) + M \cdot O_{QA}$$

Komplexität

- *Worst*-Worst-Case
  - Vorwärts gehen: Informationen verteilen + Vorwärtsnachricht, i.e., $T \cdot \left(O_{MP} + O_{QA}\right)$
  - Anfragen beantworten

$$T^2 \cdot \left(O_{MP} + O_{QA}\right) + M \cdot O_{QA}$$

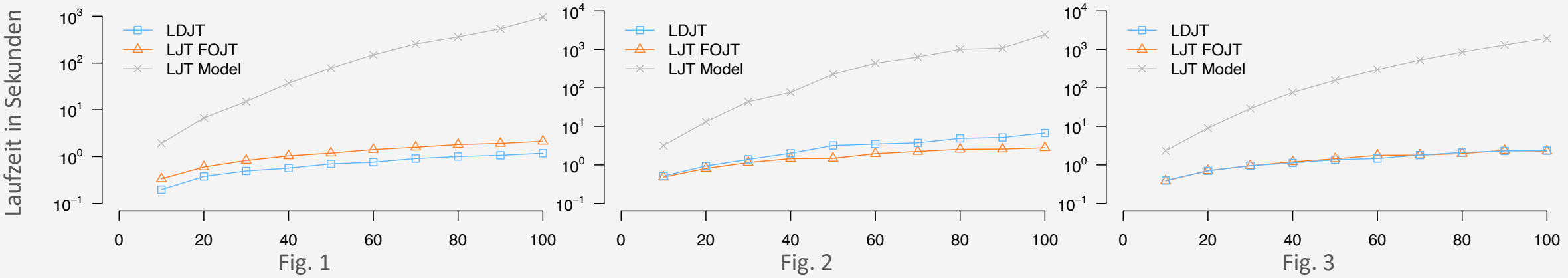
  - Zusammen,

$$\begin{aligned} & T \cdot \left(O_{MP} + O_{QA}\right) + T^2 \cdot \left(O_{MP} + O_{QA}\right) + M \cdot O_{QA} \\ = & \left(T \cdot n_J + T + T^2 \cdot n_J + T^2 + M\right) \cdot O_{QA} \\ = & O\left(\left((T^2 + T) \cdot n_J + M\right) \cdot r^w\right) \\ = & O\left(\left((T^2 + T) \cdot n_J + T \cdot m\right) \cdot r^w\right) \end{aligned}$$

Vergleiche Komplexität Filterungsanfragen: $O\left(\left(T \cdot n_J + T \cdot m\right) \cdot r^w\right)$

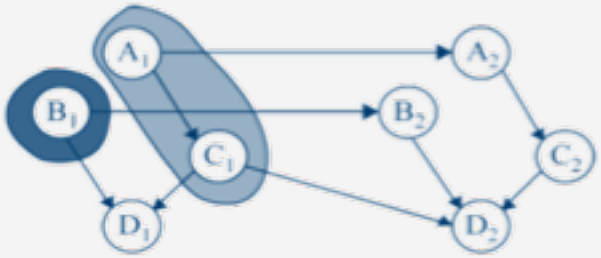
Laufzeiten

- Algorithmen im Vergleich: LDJT, LJT FOJT: Inferenz auf ausgerolltem Jtree mittels JT, LJT Modell: Inferenz auf ausgerolltem Modell mittels JT
  - Das ‚*L*' (von Lifted) bitte ignorieren; gezeigtes Verhalten gilt so für die propositionalen Algorithmen hier
  - LJT FOJT ähnlich zu LDJT, da die gleichen Berechnungen durchgeführt werden, nur LJT FOJT hält das gesamte Modell im Speicher (klein genug, dass das geht)
- Fig. 1: Filterungsanfragen, Fig. 2: Rückschau/ Vorhersage-Anfragen zu $0, T$, Fig. 3: Rückschau-Anfragen (Vorhersage-Anfragen mit ähnlichen Laufzeiten)



Approximationen

- Boyen-Koller Algorithmus
  - Annahme: Unabhängigkeiten im Interface
    - Partitionierung des Interfaces als Eingabe gefordert
      - Beispiel: $\{\{A, C\}, \{B\}\}$ in einem Interface $\{A, B, C\}$



  - Anfrage nicht mehr über das gesamte Interface, sondern als Produkt der unabhängigen Teile
    - Inter-Zeitscheiben-Nachrichten: Vereinigung der Resultate der Anfragen über die einzelnen Partitionen
    - Beispiel: VE aufrufen mit dem lokalen Modell und Nachrichten von $\boldsymbol{C}_{out}^{(t)}$ und der Anfrage mit Anfragevariablen $\{A, C\}$ und der Anfrage mit Anfragevariable $\{B\}$:

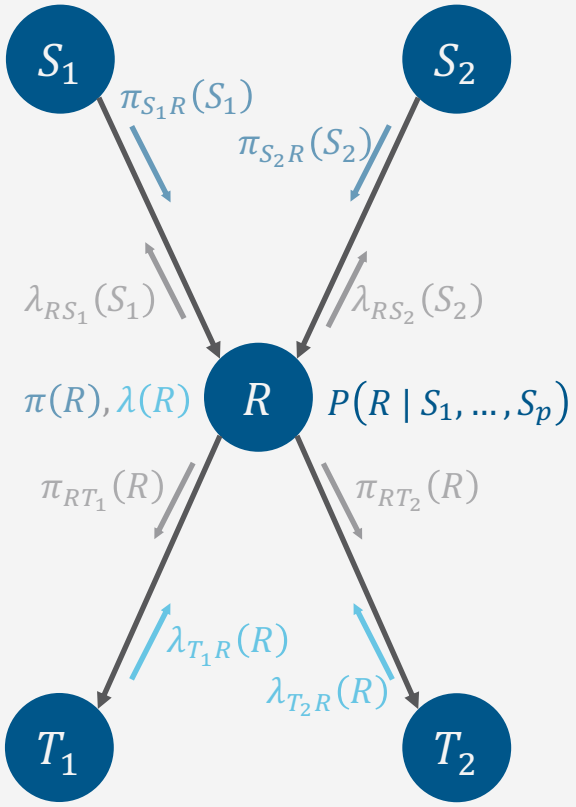
$$\alpha^{(t)} = \left\{ \mathrm{VE}\left(F_{out}^{(t)} \cup \bigcup_{j \in \mathrm{Nb}\left(\boldsymbol{C}_{out}^{(t)}\right)} m_{j,out}^{(t)}, \{A, C\}, \emptyset \right), \mathrm{VE}\left(F_{out}^{(t)} \cup \bigcup_{j \in \mathrm{Nb}\left(\boldsymbol{C}_{out}^{(t)}\right)} m_{j,out}^{(t)}, \{B\}, \emptyset \right) \right\}$$

  - Exakter Algorithmus, wenn Unabhängigkeiten tatsächlich gelten, sonst approximativ

Xavier Boyen and Daphne Koller: Tractable Inference for Complex Stochastic Processes. In: *UAI-98 Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

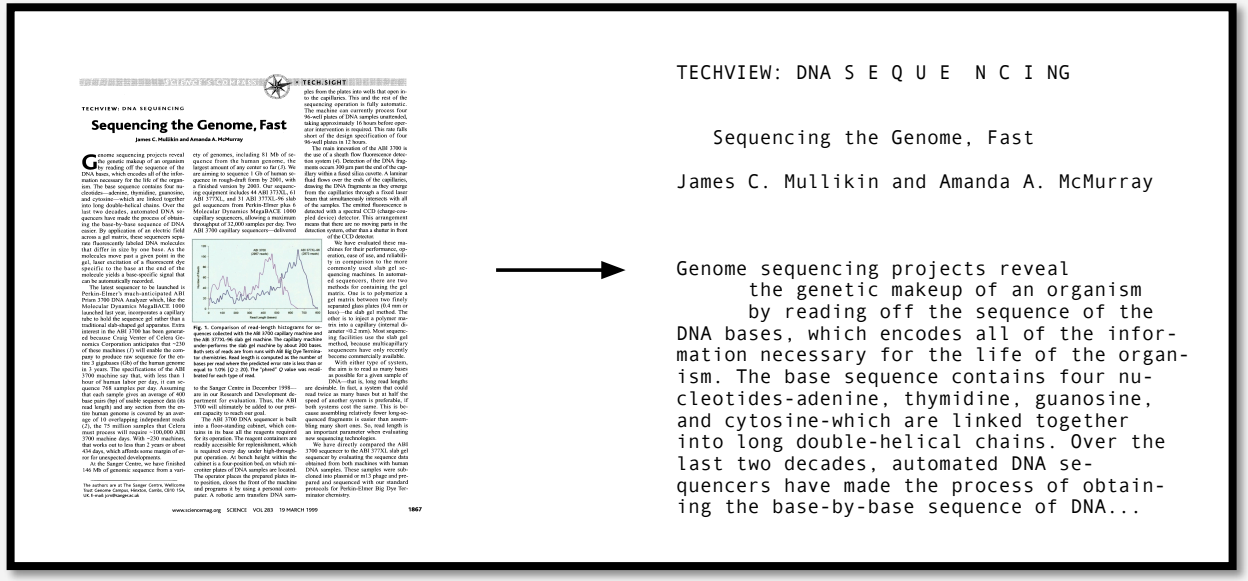
Approximationen

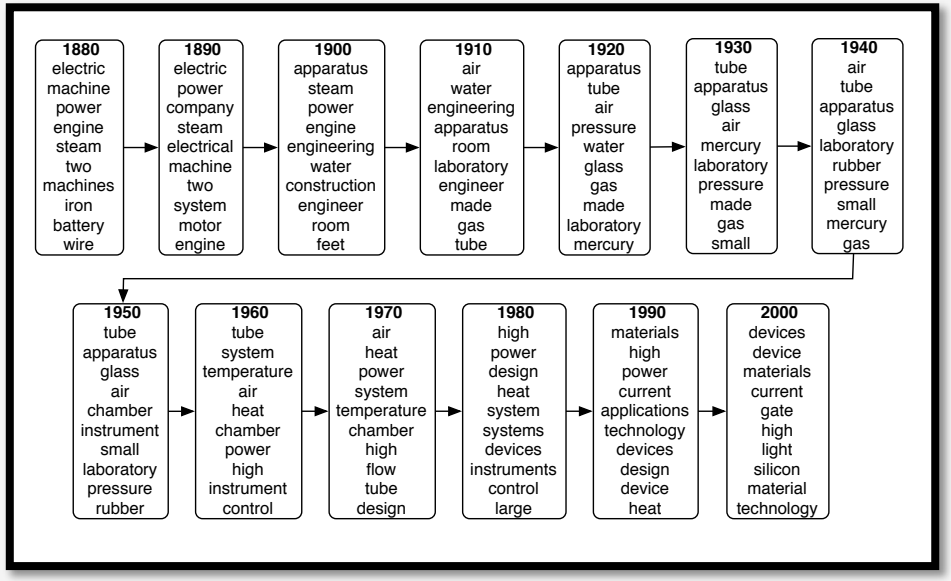
- *Factored-frontier* Algorithmus [Kevin Murphys Dissertation, 2002]
  - Zur Erinnerung:
    - Probability Propagation auf Polytree BNs: Nachrichten direkt zwischen den Knoten versandt
    - Probability Propagation in allgemeinen Graphen → (loopy) belief propagation: Nachrichten direkt zwischen den Knoten versandt
      - Exakt bei Polytrees, sonst approximativ
  - Kerngedanke: Nutze loopy belief propagation über die Zeit



Topic Modellierung über die Zeit: Dynamic Topic Modell

- Wie auch Standard LDA, *Sampling-basiert*
- Beispiel: Journal *Science* (1880-2002) JSTOR Corpus
  - Vokabular beschränkt auf die 30.000 Terme, die mehr als 10 mal vorkommen
  - Ca. 76 Mio. Worte in rund 130.000 Dokumenten





Abbildungen: David Blei, Modelling Science

Zwischenzusammenfassung

- Interfaces separieren die Vergangenheit von der Gegenwart und die Gegenwart von der Zukunft
  - Inter-Zeitscheiben-Nachrichten
    - Vorwärtsnachricht transferiert alle Informationen aus der Vergangenheit bis zur aktuellen Zeitscheibe einschließlich zur nächsten Zeitscheibe
    - Rückwärtsnachricht transferiert alle Informationen aus der Zukunft bis zur aktuellen Zeitscheibe einschließlich zur vorherigen Zeitscheibe
- DJT Algorithmus
  - JT für Intra-Zeitscheiben-Anfragebeantwortung
  - Inter-Zeitscheiben-Nachrichten um sich in der Zeit zu bewegen
  - Komplexität: Abhängig von der maximalen Zeitscheibe T
  - Approximationen über Nachrichten direkt auf dem Graph oder Unabhängigkeiten im Interface

Überblick: 6. Sequentielle PGMs und Inferenz

A. Sequentielle PGMs

- Templates, dynamische BNs, dynamische Faktormodelle, Hidden-Markov-Modelle; Semantik
- Inferenzaufgaben: Wahrscheinlichkeitsanfragen (Filtering, Prediction, Hindsight), Zustandsanfragen (MPE, MAP)

B. Sequentielle Inferenz

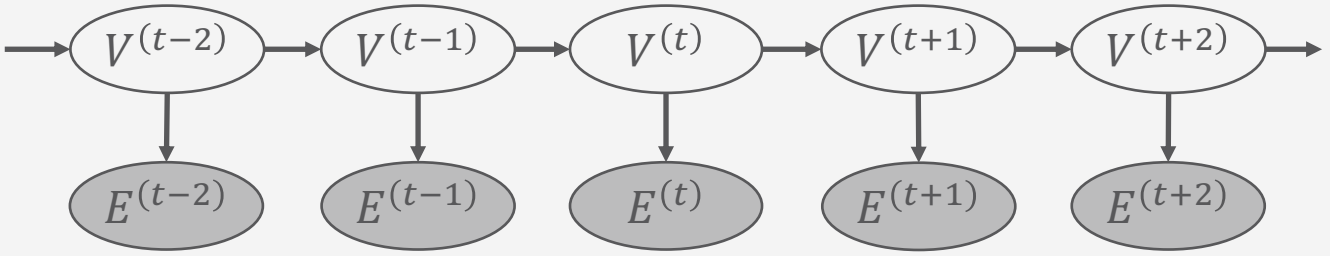
- Naïve Inferenz mittels Ausrollen, Interface Algorithmus, Komplexität, Approximationen

C. Spezialfall Hidden-Markov-Modelle

- Viterbi-Algorithmus für MPEs
- Anfragebeantwortung durch Matrixoperationen
- Baum-Welch-Algorithmus zum Lernen

Wiederholt: Hidden Markov Modell (HMM)

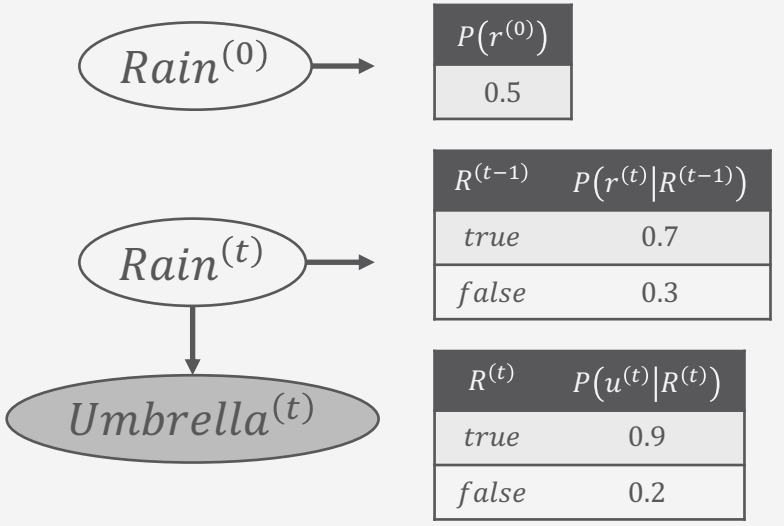
- Spezialfall des DBN mit $\boldsymbol{R} = \{V, E\}$ bzw. $\boldsymbol{V} = \{V\}, \boldsymbol{E} = \{E\}$
  - Latente Zufallsvariable V
  - Evidenzvariable E
  - Modell:
    - $B^0 = P\left(V^{(0)}\right)$
    - $B^{\rightarrow} = \left\{P\left(V^{(\tau)} \middle| V^{(\tau-1)}\right), P\left(E^{(\tau)} \middle| V^{(\tau)}\right)\right\}$
  - Repräsentierte Verteilung: $P_{V,E}^{T} = P\left(V^{(0)}\right) \prod_{t=1}^{T} P\left(V^{(t)} \middle| V^{(t-1)}\right) P\left(E^{(t)} \middle| V^{(t)}\right)$



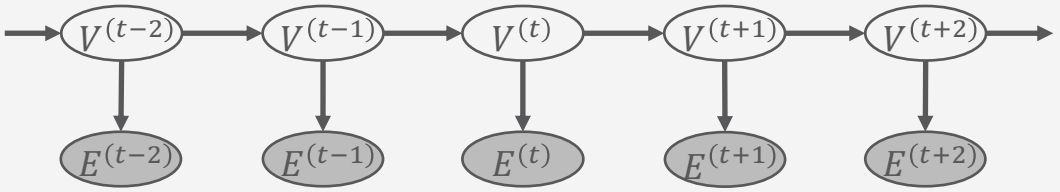
Ein DBN mit $\boldsymbol{R} = \boldsymbol{V} \cup \boldsymbol{E}$ kann jederzeit in ein HMM umgewandelt werden:

- Mit jedem $\boldsymbol{v} \in \mathrm{Val}(\boldsymbol{V})$ als abstrakten Wert v, $\mathrm{Val}(V) = \mathrm{Val}(\boldsymbol{V})$
- Mit jedem $\boldsymbol{e} \in \mathrm{Val}(\boldsymbol{E})$ als abstrakten Wert e, $\mathrm{Val}(E) = \mathrm{Val}(\boldsymbol{E})$

$\mathrm{Val}(V)$ und $\mathrm{Val}(E)$ entsprechend groß!



Filterungsanfrage



- $P\left(\boldsymbol{V}^{(t)} \middle| \boldsymbol{e}^{(1:t)}\right)$
  - Verteilung über den Zustand der Umgebung → Grundlage für Agent um rationale Entscheidung zu treffen
  - Bei HMMs: $P\left(V^{(t)} \middle| e^{(1:t)}\right)$
- Was passiert bei den Berechnungen?
  - Gegeben den Ergebnissen der Filterung bis zum Zeitpunkt t, Ergebnis für Zeitpunkt $t+1$ aus der neuen Evidenz $\boldsymbol{e}^{(t+1)}$ berechnen

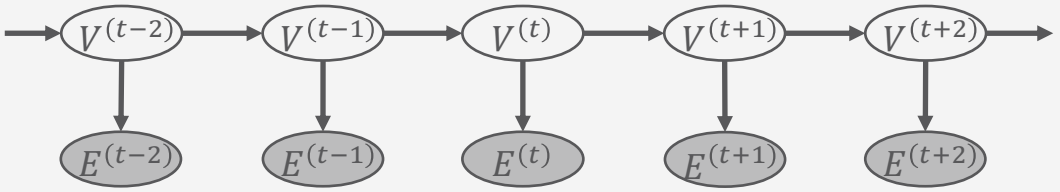
$$P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{e}^{(0:t+1)}\right) = f\left(\boldsymbol{e}^{(t+1)}, P\left(\boldsymbol{V}^{(t)} \middle| \boldsymbol{e}^{(0:t+1)}\right)\right) \quad \text{(für eine Funktion } f\text{)}$$

$$= P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{e}^{(0:t)}, \boldsymbol{e}^{(t+1)}\right) \quad \text{(Evidenz aufteilen)}$$

$$\propto P\left(\boldsymbol{e}^{(t+1)} \middle| \boldsymbol{V}^{(t+1)}, \boldsymbol{e}^{(0:t)}\right) P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{e}^{(0:t)}\right) \quad \text{(durch Bayes Theorem)}$$

$$\propto \underbrace{P\left(\boldsymbol{e}^{(t+1)} \middle| \boldsymbol{V}^{(t+1)}\right)}_{\text{Aktualisierung der Ein-Schritt-Vorhersage mit Evidenz}} \underbrace{P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{e}^{(0:t)}\right)}_{\text{Ein-Schritt-Vorhersage}} \quad \text{(durch Markov Annahme)}$$

Filterungsanfrage



- Was passiert bei den Berechnungen?
  - Gegeben den Ergebnissen der Filterung bis zum Zeitpunkt t, Ergebnis für Zeitpunkt $t+1$ aus der neuen Evidenz $\boldsymbol{e}^{(t+1)}$ berechnen

$$P\left(\boldsymbol{V}^{(t+1)}\middle|\boldsymbol{e}^{(0:t+1)}\right) \propto P\left(\boldsymbol{e}^{(t+1)}\middle|\boldsymbol{V}^{(t+1)}\right)P\left(\boldsymbol{V}^{(t+1)}\middle|\boldsymbol{e}^{(0:t)}\right)$$

$$\propto P\left(\boldsymbol{e}^{(t+1)}\middle|\boldsymbol{V}^{(t+1)}\right)\sum_{\boldsymbol{v}^{(t)}\in \mathrm{Val}\left(\boldsymbol{V}^{(t)}\right)} P\left(\boldsymbol{V}^{(t+1)}\middle|\boldsymbol{v}^{(t)},\boldsymbol{e}^{(0:t)}\right)P\left(\boldsymbol{v}^{(t)}\middle|\boldsymbol{e}^{(0:t)}\right) \quad \text{(Konditionieren auf } \boldsymbol{V}^{(t)}\text{)}$$

$$\propto P\left(\boldsymbol{e}^{(t+1)}\middle|\boldsymbol{V}^{(t+1)}\right)\sum_{\boldsymbol{v}^{(t)}\in \mathrm{Val}\left(\boldsymbol{V}^{(t)}\right)} P\left(\boldsymbol{V}^{(t+1)}\middle|\boldsymbol{v}^{(t)}\right)P\left(\boldsymbol{v}^{(t)}\middle|\boldsymbol{e}^{(0:t)}\right) \quad \text{(durch Markov Annahme)}$$

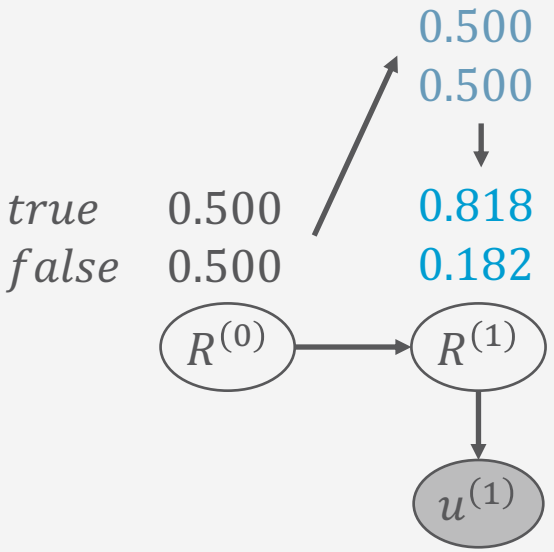
Vorwärtsnachricht $\alpha^{(t+1)}$ $= \textsc{Forward}\left(\alpha^{(t)}, \boldsymbol{e}^{(t+1)}\right)$

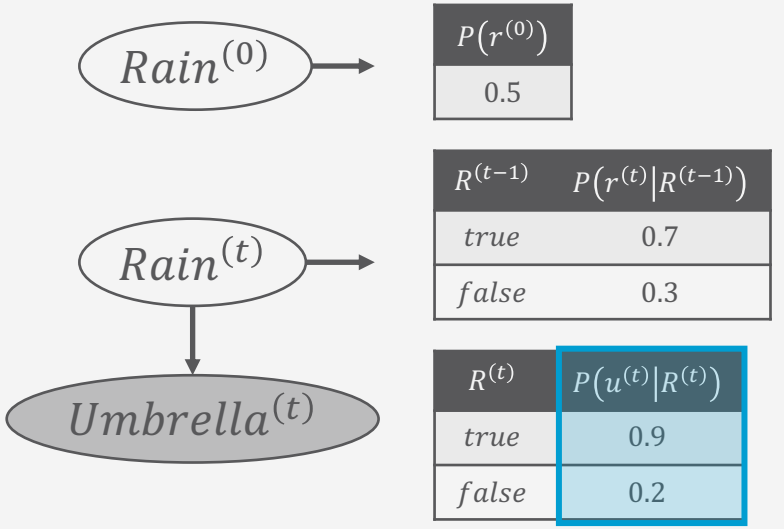
Übergangsmodell

Filterungsergebnis bis zum Zeitpunkt t = Vorwärtsnachricht $\alpha^{(t)}$

Beispiel

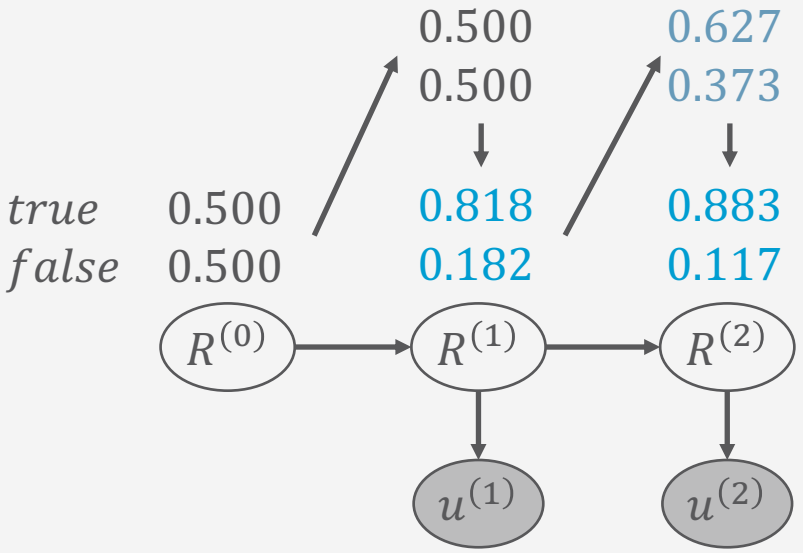
- Tag $t = 1$: Regenschirm beobachtet $\rightarrow u^{(1)}$
  - Vorhersage von $t = 0$ nach $t = 1$ ist: $P\left(R^{(1)}\right) = \sum_{r^{(0)} \in \mathrm{Val}\left(R^{(0)}\right)} P\left(R^{(1)} \middle| r^{(0)}\right) P\left(r^{(0)}\right)$
  - Aktualisieren mit der Evidenz von $t = 1$ ergibt: $P\left(R^{(1)} \middle| u^{(1)}\right) \propto P\left(u^{(1)} \middle| R^{(1)}\right) P\left(R^{(1)}\right)$

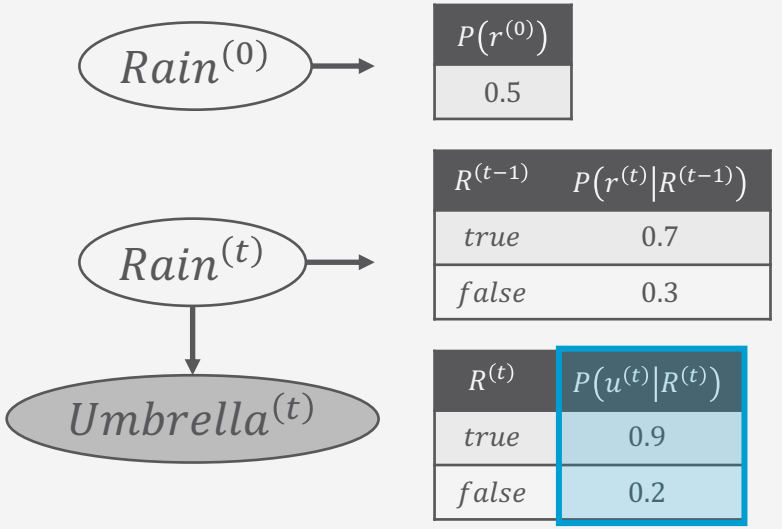




Beispiel

- Tag $t = 2$: Regenschirm beobachtet $\rightarrow u^{(2)}$
  - Vorhersage von $t = 1$ nach $t = 2$ ist: $P\left(R^{(2)}\middle|u^{(1)}\right) = \sum_{r^{(1)} \in \mathrm{Val}(R^{(1)})} P\left(R^{(2)}\middle|r^{(1)}\right) P\left(r^{(1)}\middle|u^{(1)}\right)$
  - Aktualisieren mit der Evidenz von $t = 2$ ergibt: $P\left(R^{(2)}\middle|u^{(1:2)}\right) \propto P\left(u^{(2)}\middle|R^{(2)}\right) P\left(R^{(2)}\middle|u^{(1)}\right)$





Vorhersagen und Rückschauen

- Vorhersage $P\left(\boldsymbol{V}^{(t+k)}\middle|\boldsymbol{e}^{(0:t)}\right)$
  - Filterungsanfrage ohne Evidenz von t bis $t+k$
  - Berechnungstechnisch passiert das gleiche wie auf den Folien zuvor
- Rückschau $P\left(\boldsymbol{V}^{(t-k)}\middle|\boldsymbol{e}^{(0:t)}\right)$
  - Bessere Schätzung des Zustandes zum Zeitpunkt $t-k$, da jetzt mehr Evidenz vorhanden
  - Berechnungen: Evidenz aufteilen in $\boldsymbol{e}^{(0:t-k)}, \boldsymbol{e}^{(t-k+1:t)}$

$$
\begin{aligned}
P\left(\boldsymbol{V}^{(t-k)}\middle|\boldsymbol{e}^{(0:t)}\right) &= P\left(\boldsymbol{V}^{(t-k)}\middle|\boldsymbol{e}^{(0:t-k)}, \boldsymbol{e}^{(t-k+1:t)}\right) && \text{(Evidenz aufteilen)}\\
&\propto P\left(\boldsymbol{V}^{(t-k)}\middle|\boldsymbol{e}^{(0:t-k)}\right) P\left(\boldsymbol{e}^{(t-k+1:t)}\middle|\boldsymbol{V}^{(t-k)}, \boldsymbol{e}^{(0:t-k)}\right) && \text{(durch Bayes Theorem)}\\
&\propto P\left(\boldsymbol{V}^{(t-k)}\middle|\boldsymbol{e}^{(0:t-k)}\right) P\left(\boldsymbol{e}^{(t-k+1:t)}\middle|\boldsymbol{V}^{(t-k)}\right) && \text{(durch Markov Annahme)}\\
&\propto \alpha^{(t-k)}\beta^{(t-k+1)}
\end{aligned}
$$

  - Vorwärtsnachricht abspeichern
  - Rückwärtsnachricht mittels Rückwärtsrekursion

Vorhersagen und Rückschauen → *Vorwärts-Rückwärts-Algorithmus*

- Berechnungen: Evidenz aufteilen in $\boldsymbol{e}^{(0:t-k)}, \boldsymbol{e}^{(t-k+1:t)}$

$$P\left(\boldsymbol{V}^{(t-k)} \middle| \boldsymbol{e}^{(0:t)}\right) = P\left(\boldsymbol{V}^{(t-k)} \middle| \boldsymbol{e}^{(0:t-k)}, \boldsymbol{e}^{(t-k+1:t)}\right) \quad \text{(Evidenz aufteilen)}$$

$$\propto P\left(\boldsymbol{V}^{(t-k)} \middle| \boldsymbol{e}^{(0:t-k)}\right) P\left(\boldsymbol{e}^{(t-k+1:t)} \middle| \boldsymbol{V}^{(t-k)}, \boldsymbol{e}^{(0:t-k)}\right) \quad \text{(durch Bayes Theorem)}$$

$$\propto P\left(\boldsymbol{V}^{(t-k)} \middle| \boldsymbol{e}^{(0:t-k)}\right) P\left(\boldsymbol{e}^{(t-k+1:t)} \middle| \boldsymbol{V}^{(t-k)}\right) \quad \text{(durch Markov Annahme)}$$

$$\propto \alpha^{(t-k)} \beta^{(t-k+1)}$$

  - Vorwärtsnachricht abspeichern
  - Rückwärtsnachricht mittels Rückwärtsrekursion

$$P\left(\boldsymbol{e}^{(t-k+1:t)} \middle| \boldsymbol{V}^{(t-k)}\right) = \sum_{\boldsymbol{v}^{(t-k+1)}} P\left(\boldsymbol{e}^{(t-k+1:t)} \middle| \boldsymbol{V}^{(t-k)}, \boldsymbol{v}^{(t-k+1)}\right) P\left(\boldsymbol{v}^{(t-k+1)} \middle| \boldsymbol{V}^{(t-k)}\right)$$

(Konditionieren auf $\boldsymbol{V}^{(t-k+1)}$)

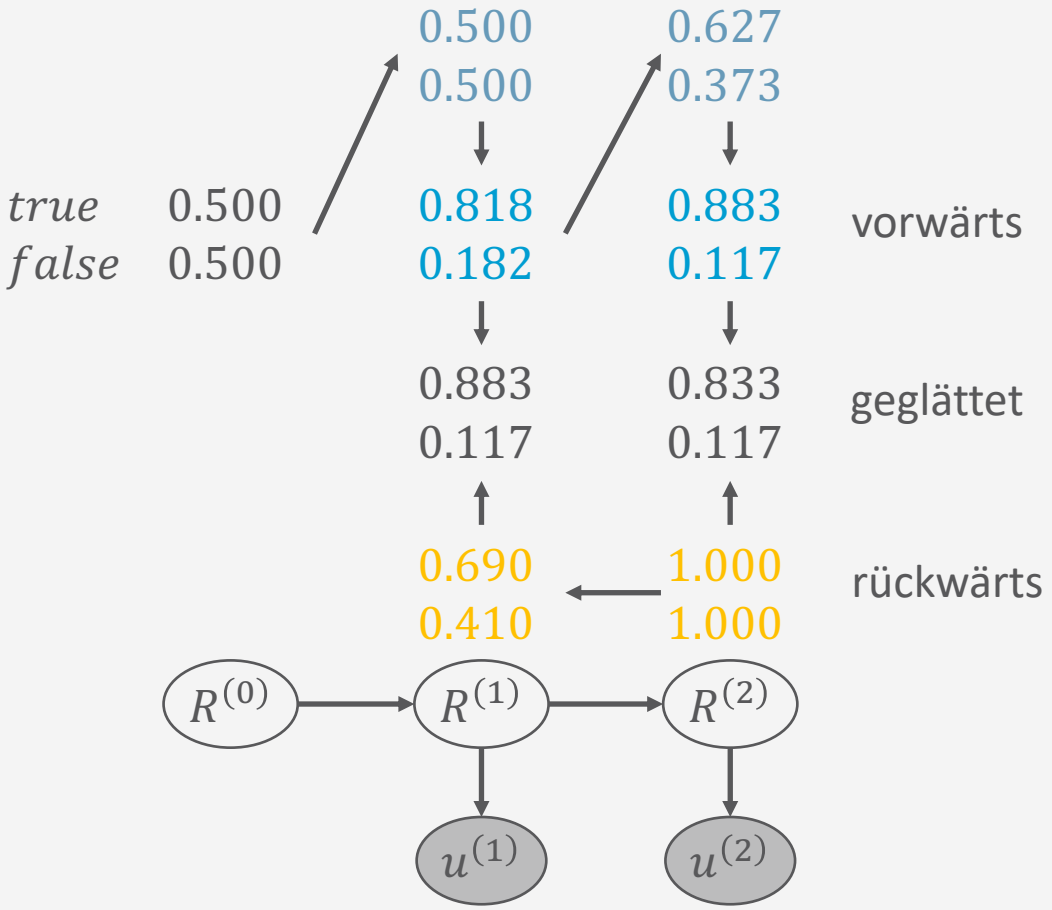
$$\propto \sum_{\boldsymbol{v}^{(t-k+1)}} P\left(\boldsymbol{e}^{(t-k+1:t)} \middle| \boldsymbol{v}^{(t-k+1)}\right) P\left(\boldsymbol{v}^{(t-k+1)} \middle| \boldsymbol{V}^{(t-k)}\right)$$

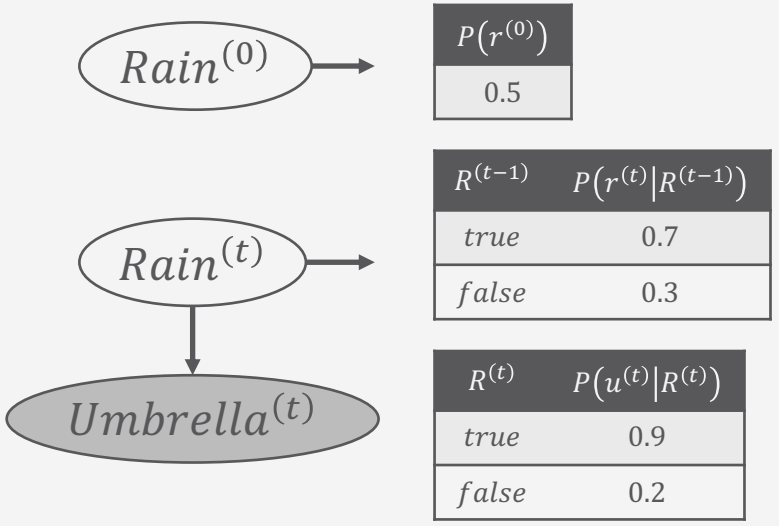
(durch Markov Annahme)

$$\propto \sum_{\boldsymbol{v}^{(t-k+1)}} P\left(\boldsymbol{e}^{(t-k+1)} \middle| \boldsymbol{v}^{(t-k+1)}\right) P\left(\boldsymbol{e}^{(t-k+2:t)} \middle| \boldsymbol{v}^{(t-k+1)}\right) P\left(\boldsymbol{v}^{(t-k+1)} \middle| \boldsymbol{V}^{(t-k)}\right)$$

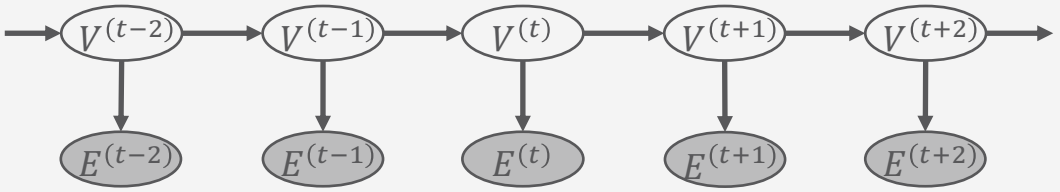
(Kettenregel)

Beispiel


0.500
0.627
0.500
0.373
true 0.500
false 0.500
0.818
0.883
0.182
0.117
vorwärts
0.883
0.833
0.117
0.117
geglättet
0.690
1.000
0.410
1.000
rückwärts
$R^{(0)}$
$R^{(1)}$
$R^{(2)}$
$u^{(1)}$
$u^{(2)}$


$Rain^{(0)}$
$P(r^{(0)})$
0.5
$Rain^{(t)}$
$R^{(t-1)}$
$P(r^{(t)}|R^{(t-1)})$
true 0.7
false 0.3
$Umbrella^{(t)}$
$R^{(t)}$
$P(u^{(t)}|R^{(t)})$
true 0.9
false 0.2

MPE



- MPE Definition

$$MPE\left(\boldsymbol{e}^{(1:t)}\right) = \arg\max_{\boldsymbol{v}^{(1:t)}} P\left(\boldsymbol{v}^{(1:t)} \middle| \boldsymbol{e}^{(1:t)}\right)$$

  - Bei HMMs: $\arg\max_{v^{(1:t)}} P\left(v^{(1:t)} \middle| e^{(1:t)}\right)$
- Was passiert bei den Berechnungen?
  - Wahrscheinlichster Pfad zu jedem $\boldsymbol{v}^{(t+1)}$ = wahrscheinlichster Pfad zu einem $\boldsymbol{v}^{(t)}$ plus ein weiterer Schritt

Übergangsmodell

MPE-Vorwärtsnachricht $\alpha^{(t)}$

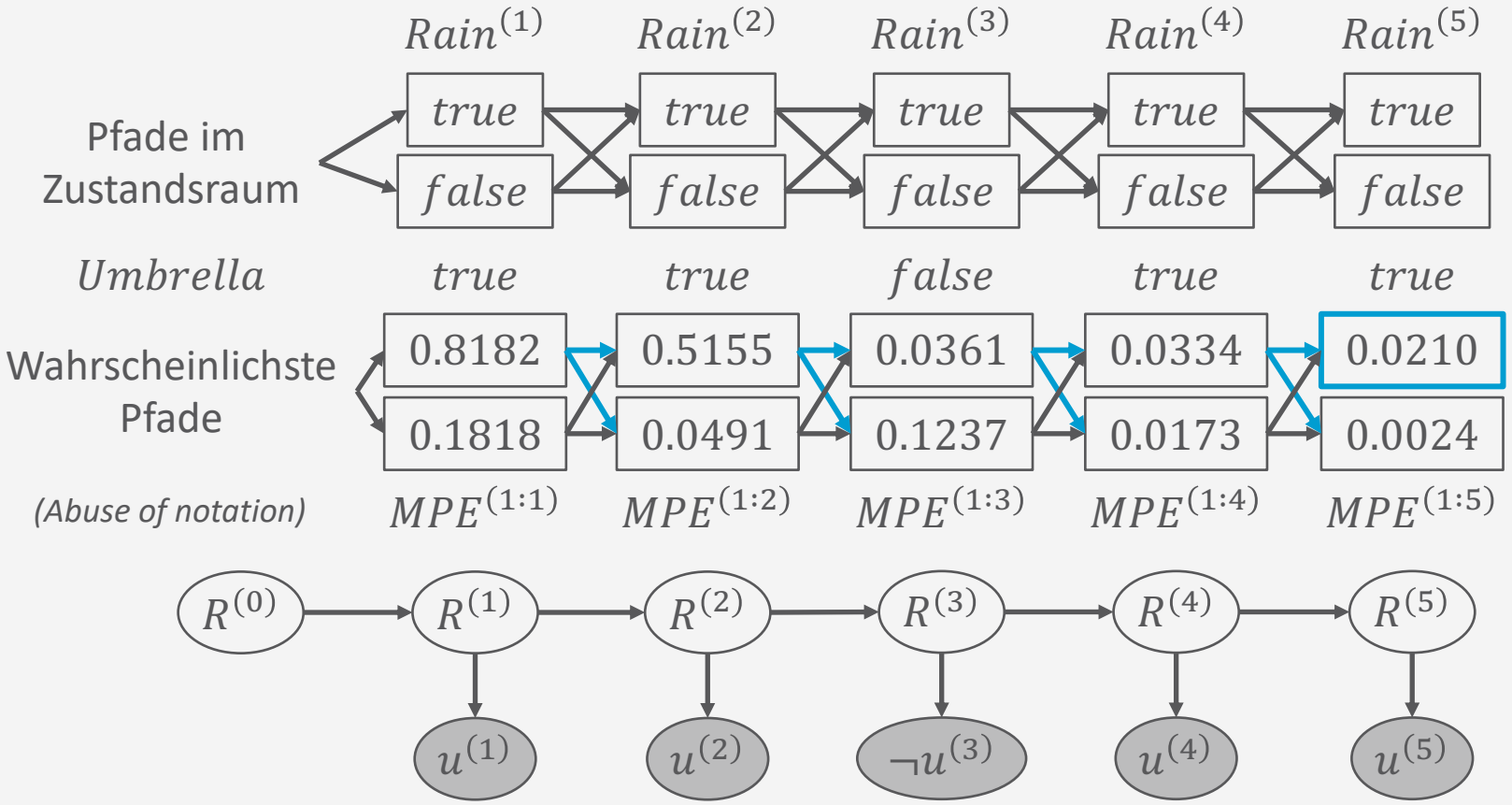
$$\max_{\boldsymbol{v}^{(1:t)}} P\left(\boldsymbol{v}^{(1:t)}, \boldsymbol{V}^{(t+1)} \middle| \boldsymbol{e}^{(1:t)}\right) = P\left(\boldsymbol{e}^{(t+1)} \middle| \boldsymbol{V}^{(t+1)}\right) \max_{\boldsymbol{v}^{(t)}} \left(P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{v}^{(t)}\right) \max_{\boldsymbol{v}^{(1:t-1)}} P\left(\boldsymbol{v}^{(1:t-1)}, \boldsymbol{v}^{(t)} \middle| \boldsymbol{e}^{(1:t)}\right) \right)$$

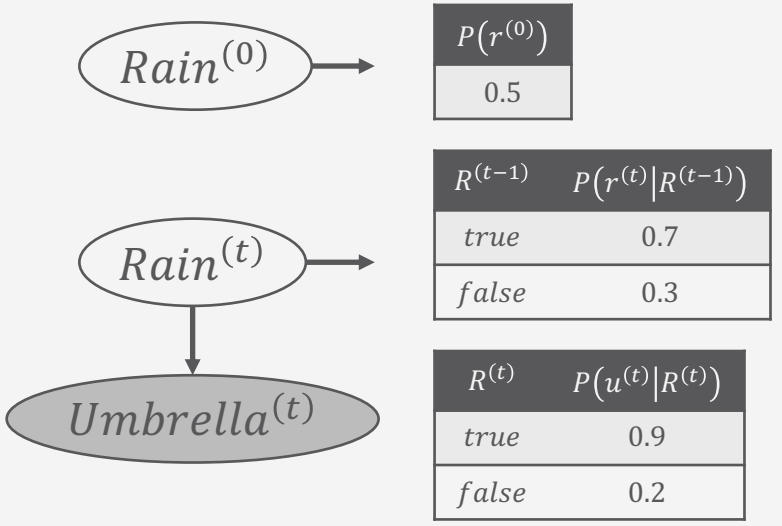
$$= P\left(\boldsymbol{e}^{(t+1)} \middle| \boldsymbol{V}^{(t+1)}\right) \max_{\boldsymbol{v}^{(t)}} \left(P\left(\boldsymbol{V}^{(t+1)} \middle| \boldsymbol{v}^{(t)}\right) \cdot \alpha^{(t)} \right)$$

MPE-Vorwärtsnachricht $\alpha^{(t+1)}$
$= \text{MPE}-\text{FORWARD}\left(\alpha^{(t)}, \boldsymbol{e}^{(t+1)}\right)$

Viterbi-Algorithmus für HMMs

Beispiel





$P(r^{(0)})$
0.5

$R^{(t-1)}$	$P(r^{(t)}\|R^{(t-1)})$
true	0.7
false	0.3

$R^{(t)}$	$P(u^{(t)}\|R^{(t)})$
true	0.9
false	0.2

HMMs als Matrizen

Rechnungen als Matrixoperationen
→ Effiziente Implementierung möglich

- Latente Zufallsvariable V, $|\mathrm{Val}(V)| = n$
- Evidenzvariable E
- Übergangsmatrix $\mathfrak{T}$ eine $n \times n$ Matrix mit
  - $\mathfrak{T}_{ij} = P\left(V^{(\tau)} = v_i \middle| V^{(\tau-1)} = v_j\right)$
- Sensormatrix $\mathfrak{O}^{(t)}$ für jeden Zeitschritt t gemäß Beobachtung $e^{(t)}$ eine $n \times n$ Matrix
  - Diagonale Einträgen $P\left(e^{(t)} \middle| V^{(t)} = v_i\right)$
- Beispiel:
  - Übergangsmatrix:

$$\text{neuer Zustand } v_i: \quad \begin{matrix} \\ t \\ f \end{matrix} \overset{\text{alter Zustand } v_j}{\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}}$$

Sensormatrix:

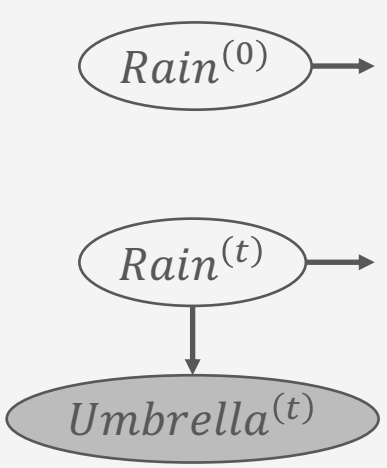
$$U^{(1)} = true, \mathfrak{O}^{(1)} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

$$U^{(3)} = false, \mathfrak{O}^{(3)} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.8 \end{pmatrix}$$

- Vorwärts- und Rückwärtsnachrichten als Spaltenvektoren

$$\alpha^{(t+1)} \propto \mathfrak{O}^{(t+1)} \mathfrak{T}^{\top} \alpha^{(t)}$$

$$\beta^{(t-k+1)} \propto \mathfrak{T} \mathfrak{O}^{(t-k+1)} \beta^{(t-k+2)}$$



$P(r^{(0)})$
0.5

$R^{(t-1)}$	$P(r^{(t)}\|R^{(t-1)})$
$true$	0.7
$false$	0.3

$R^{(t)}$	$P(u^{(t)}\|R^{(t)})$
$true$	0.9
$false$	0.2

Country Dance Algorithmus für HMMs

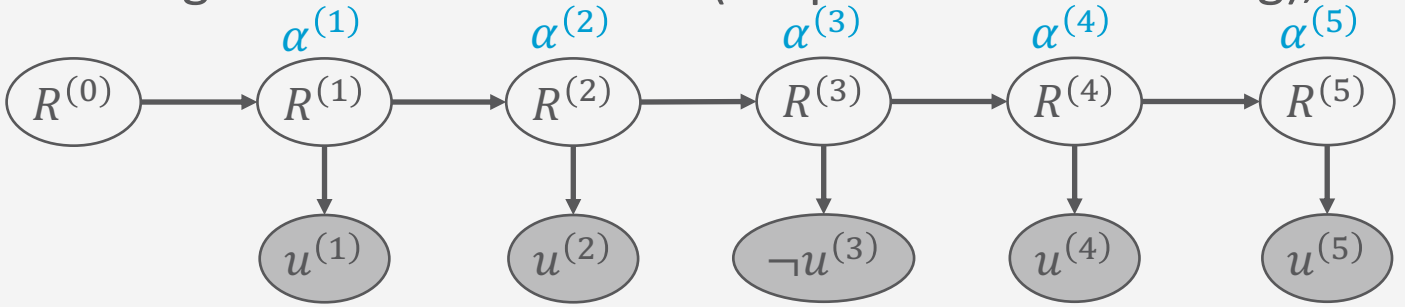
- Vorwärtsnachrichten nicht abspeichern zum Glätten, sondern Nachrichten bedarfsweise neu berechnen, indem man den Vorwärtspass rückwärts laufen lässt → Matrixinvertierung

$$\alpha^{(t+1)} = \frac{1}{Z}\mathfrak{O}^{(t+1)}\mathfrak{T}^{\top}\alpha^{(t)}$$

$$\mathfrak{O}^{(t+1)^{-1}}\alpha^{(t+1)} = \frac{1}{Z}\mathfrak{T}^{\top}\alpha^{(t)}$$

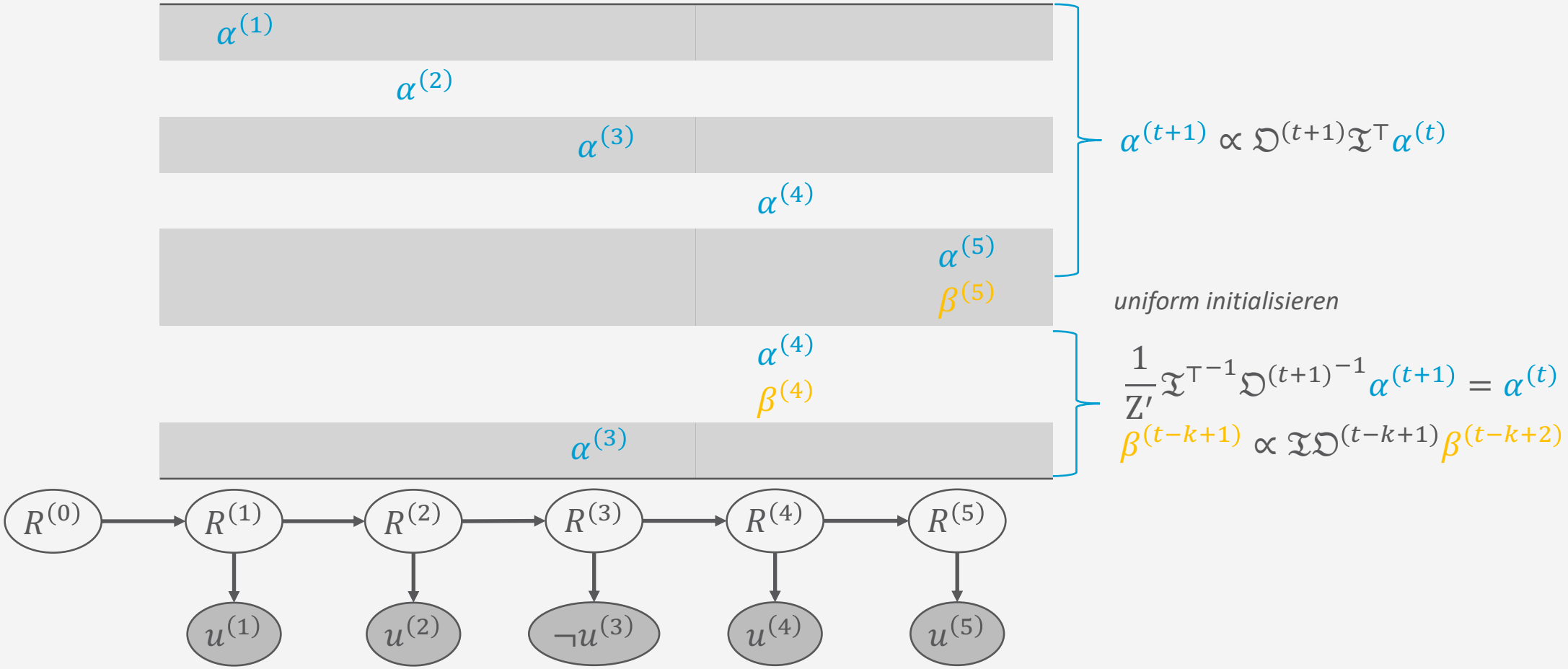
$$\frac{1}{Z'}\mathfrak{T}^{\top -1}\mathfrak{O}^{(t+1)^{-1}}\alpha^{(t+1)} = \alpha^{(t)}$$

- Vorwärts gehen berechnet $\alpha^{(t)}$ (abspeichern nicht nötig), rückwärts gehen berechnet $\alpha^{(t')}, \beta^{(t')}$

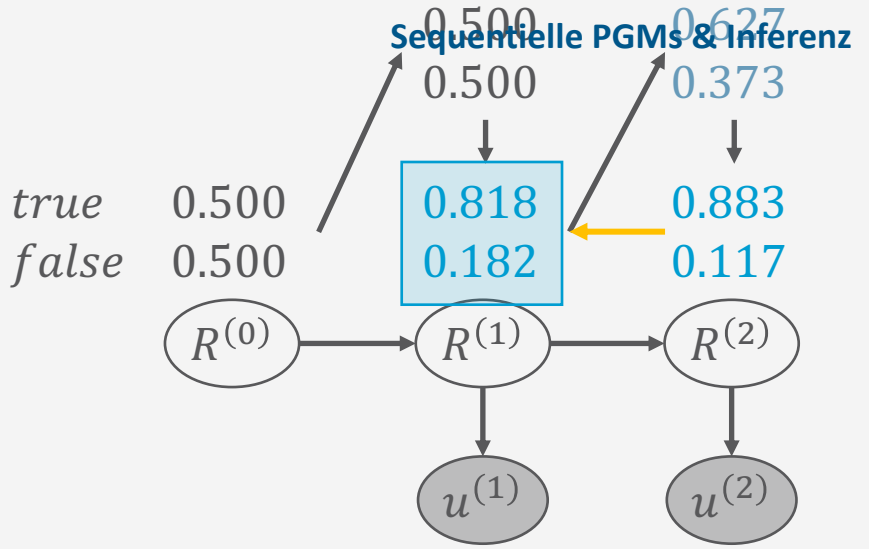


Speichereffizient

- Anfragen: Filterungsanfragen $P\left(R^{(t)}\middle|u^{(1:t)}\right)$, Rückschau-Anfrage $P\left(R^{(3)}\middle|u^{(1:5)}\right)$
- Vorwärts gehen bis einschließlich $t = 5$, jeweils $P\left(R^{(t)}\middle|u^{(1:t)}\right) = \alpha^{(t)}$ berechnen
- Bei $t = 5$, rückwärts gehen bis $t = 3$, jeweils $\alpha^{(t)}$, $\beta^{(t-2+1)}$ berechnen, dann $P\left(R^{(3)}\middle|u^{(1:5)}\right) = \alpha^{(3)}\beta^{(4)}$



$$\frac{1}{Z'}\mathfrak{T}^{\mathsf{T}^{-1}}\mathfrak{O}^{(t+1)^{-1}}\alpha^{(t+1)} = \alpha^{(t)} \rightarrow \frac{1}{Z'}\mathfrak{T}^{\mathsf{T}^{-1}}\mathfrak{O}^{(2)^{-1}}\alpha^{(2)} = \alpha^{(1)}$$



$$U^{(2)} = true, \mathfrak{O}^{(2)} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

$$\mathfrak{T} = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \rightarrow \mathfrak{T}^{\mathsf{T}} = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$$

$$\mathfrak{O}^{(2)^{-1}} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}^{-1} = \frac{1}{0.9 \cdot 0.2 - 0 \cdot 0}\begin{pmatrix} 0.2 & 0 \\ 0 & 0.9 \end{pmatrix} = 5.56\begin{pmatrix} 0.2 & 0 \\ 0 & 0.9 \end{pmatrix} = \begin{pmatrix} 1.1 & 0 \\ 0 & 5.0 \end{pmatrix}$$

$$\mathfrak{T}^{\mathsf{T}^{-1}} = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}^{-1} = \frac{1}{0.49 - 0.09}\begin{pmatrix} 0.7 & -0.3 \\ -0.3 & 0.7 \end{pmatrix} = 2.5\begin{pmatrix} 0.7 & -0.3 \\ -0.3 & 0.7 \end{pmatrix} = \begin{pmatrix} 1.75 & -0.75 \\ -0.75 & 1.75 \end{pmatrix}$$

$$\frac{1}{Z'}\begin{pmatrix} 1.75 & -0.75 \\ -0.75 & 1.75 \end{pmatrix}\begin{pmatrix} 1.1 & 0 \\ 0 & 5.0 \end{pmatrix}\begin{pmatrix} 0.883 \\ 0.117 \end{pmatrix} = \frac{1}{Z'}\begin{pmatrix} 1.944 & -3.75 \\ -0.825 & 8.75 \end{pmatrix}\begin{pmatrix} 0.883 \\ 0.117 \end{pmatrix} = \frac{1}{Z'}\begin{pmatrix} 1.278 \\ 0.295 \end{pmatrix} = \underbrace{\begin{pmatrix} 0.812 \\ 0.188 \end{pmatrix}}_{= \alpha^{(1)}}$$

$P(r^{(0)})$
0.5

$R^{(t-1)}$	$P(r^{(t)}\|R^{(t-1)})$
true	0.7
false	0.3

$R^{(t)}$	$P(u^{(t)}\|R^{(t)})$
true	0.9
false	0.2

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{\det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Lernen von temporalen Modellen

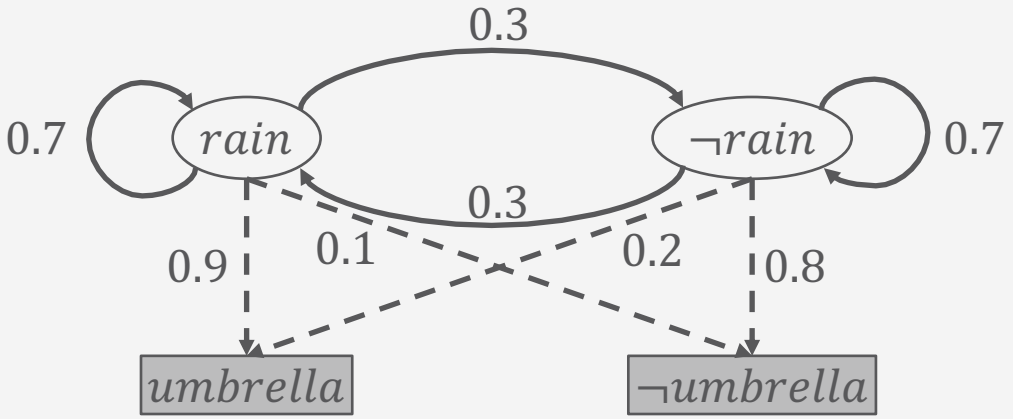
- Lernen benötigt Glättung, um bessere Schätzungen der Zustände zu bekommen
- Lernen von allgemeinen temporalen Modellen noch in den Kinderschuhen
  - Benötigt einen iterativen Ansatz im Sinne von Expectation-Maximisation (EM), um latente Zustandsvariablen zu schätzen
- Bekannter Lernalgorithmus für HMMs: Baum-Welch-Algorithmus
  - Interpretation eines HMMs als Zustandsübergangssystem

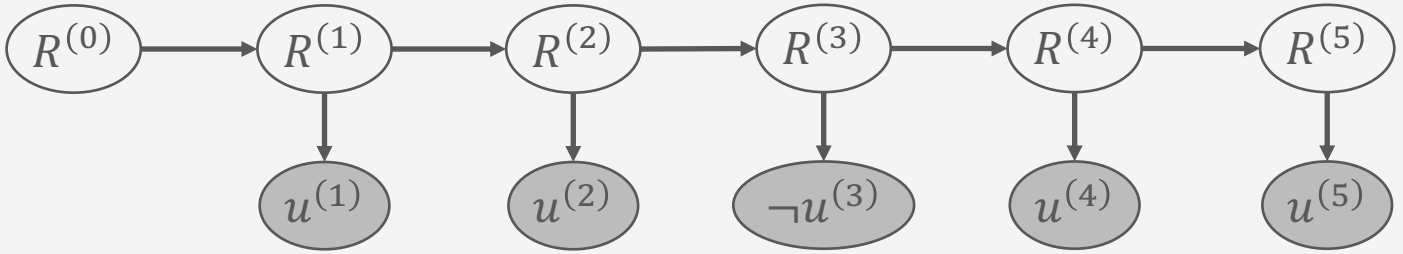
HMMs als Zustandsübergangssystem

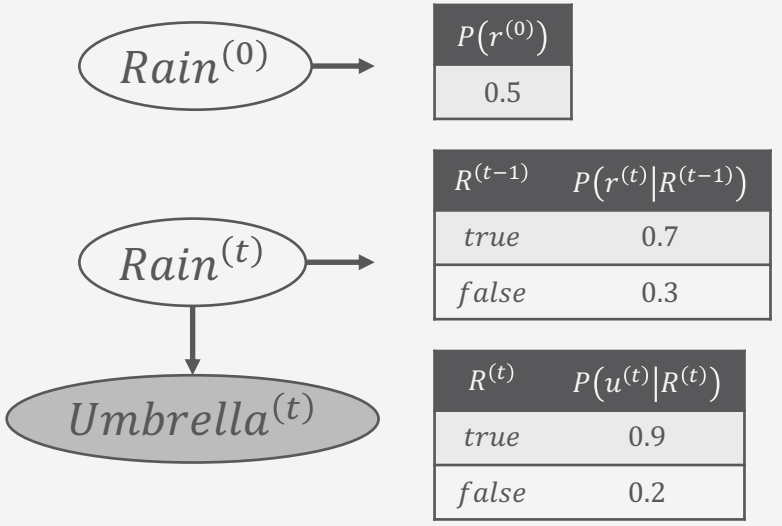
- Menge von Zuständen $\{s_1, \dots, s_n\}$
  - Prozess bewegt sich von einem Zustand in den nächsten, was eine Zustandssequenz generiert: $s_{i1}, \dots, s_{ik}$
  - Markov Annahme: Nächster Zustand hängt nur vom jetzigen Zustand ab: $P(s_{ik}|s_{i1}, \dots, s_{ik-1}) = P(s_{ik}|s_{ik-1})$
  - Matrix A: Übergangswahrscheinlichkeiten
- Zustände nicht beobachtbar, aber jeder Zustand generiert eine von m Beobachtungen (bzw. sichtbaren Zuständen) $\{v_1, \dots, v_m\}$
  - Matrix B: Emissionswahrscheinlichkeiten
- Für HMM:
  - $\mathrm{Val}(V) = \{s_1, \dots, s_n\}$
  - $\mathrm{Val}(E) = \{v_1, \dots, v_m\}$
  - Übergänge:
    $$A = (a_{ij}), a_{ij} = P(s_j|s_i)$$
    - Wie in der HMM Matrixdarstellung
  - Beobachtungen:
    $$B = (b_i(v)), b_i(v) = P(v|s_i)$$
    - Wie in der HMM Matrixdarstellung
  - Vektor mit initialen Wahrscheinlichkeiten:
    $$\pi = (\pi_i), \pi_i = P(s_i)$$
  - HMM dargestellt durch $M = (A, B, \pi)$

Zustandsübergangssystem: Beispiel

- Latente Zufallsvariable R
- Evidenzvariable U







Lernproblem

- Gegeben eine Menge von Beobachtungssequenzen $O = o_1, \dots, o_k$ als Trainingsdaten und die generelle Struktur des HMMs (Anzahl an versteckten und sichtbaren Zuständen)
- Bestimme HMM Parameter $M = (A, B, \pi)$, welche am besten zu den Trainingsdaten passen, i.e., welche $P(\boldsymbol{O}|M)$ maximieren
- Wenn wir Daten zu S hätten, könnten wir direkt ML-basiert die Einträge in M bestimmen

$$a_{ij} = P(s_j|s_i) = \frac{\text{Anzahl an Übergängen von } s_i \text{ nach } s_j}{\text{Anzahl an Übergängen aus } s_i \text{ raus}}$$

$$b_i(v) = P(v|s_i) = \frac{\text{Anzahl an Beobachtungen von } v \text{ in } s_i}{\text{Anzahl an Besuchen von } s_i}$$

- Andernfalls: Iterativ mittels EM Einträge als lokales Optimum bestimmen
  → Baum-Welch-Algorithmus

Baum-Welch-Algorithmus

- Generelle Idee: Erwartete Zähler nutzen (EM)

$$a_{ij} = P(s_j|s_i) = \frac{\text{Erwartete Anzahl an Übergängen von } s_i \text{ nach } s_j}{\text{Erwartete Anzahl an Übergängen aus } s_i \text{ raus}}$$

$$b_i(v) = P(v|s_i) = \frac{\text{Erwartete Anzahl an Beobachtungen von } v \text{ in } s_i}{\text{Erwartete Anzahl an Besuchen von } s_i}$$

$$\pi_i = P(s_i) = \text{erwartete relative Häufigkeit von } s_i \text{ zum Zeitpunkt } k = 1$$

E-Schritt

- Hilfsvariable $\xi_k(i,j)$: Wahrscheinlichkeit in Zustand s_i zum Zeitpunkt k und in Zustand s_j zum Zeitpunkt $k+1$ zu sein, gegeben die Beobachtungssequenzen $o_1, \dots, o_T, k < T$

$$\xi_k(i,j) = P\left(Q_k = s_i, Q_{k+1} = s_j \middle| o_1, \dots, o_T\right)$$

$$\xi_k(i,j) = \frac{P\left(Q_k = s_i, Q_{k+1} = s_j, o_1, \dots, o_T\right)}{P(o_1, \dots, o_T)}$$

$$= \frac{P(Q_k = s_i, o_1, \dots, o_k) a_{ij} b_i(o_{k+1}) P\left(o_{k+2}, \dots, o_T \middle| Q_{k+1} = s_j\right)}{P(o_1, \dots, o_T)}$$

$$= \frac{1}{Z} \textsc{Forward}_k(i) a_{ij} b_i(o_{k+1}) \textsc{Backward}_{k+1}(j)$$

E-Schritt

- Hilfsvariable $\gamma_k(i)$: Wahrscheinlichkeit in Zustand s_i zum Zeitpunkt k zu sein, gegeben die Beobachtungssequenzen $o_1, \dots, o_T, k < T$

$$\gamma_k(i) = P(Q_k = s_i | o_1, \dots, o_T)$$
$$\gamma_k(i) = \frac{P(Q_k = s_i, o_1, \dots, o_T)}{P(o_1, \dots, o_T)}$$
$$= \frac{1}{Z} \textsc{Forward}_k(i) \textsc{Backward}_{k+1}(j)$$

E-Schritt

- Hilfsvariablen berechnen mit anfangs geratenen Parametern
  - $\xi_k(i,j) = P(Q_k = s_i, Q_{k+1} = s_j | o_1, \dots, o_T)$
  - $\gamma_k(i) = P(Q_k = s_i | o_1, \dots, o_T)$
- Parameter schätzen
  - Erwartete Anzahl an Übergängen von Zustand s_i nach Zustand s_j = $\sum_k \xi_k(i,j)$
  - Erwartete Anzahl an Übergängen aus Zustand s_i = $\sum_k \gamma_k(i)$
  - Erwartete Anzahl an Beobachtungen v in Zustand s_i = $\sum_{k,o_k=v} \gamma_k(i)$
  - Erwartete relative Häufigkeit von Zustand s_i zum Zeitpunkt $(k = 1)$ = $\gamma_1(i)$

M-Schritt

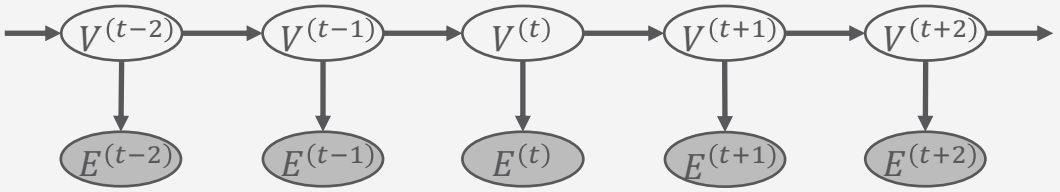
- Parameter maximieren

$$a_{ij} = P(s_j|s_i) = \frac{\text{Erwartete Anzahl an Übergängen von } s_i \text{ nach } s_j}{\text{Erwartete Anzahl an Übergängen aus } s_i \text{ raus}} = \frac{\sum_k \xi_k(i,j)}{\sum_k \gamma_k(i)}$$

$$b_i(v) = P(v|s_i) = \frac{\text{Erwartete Anzahl an Beobachtungen von } v \text{ in } s_i}{\text{Erwartete Anzahl an Besuchen von } s_i} = \frac{\sum_{k,o_k=v} \gamma_k(i)}{\sum_k \gamma_k(i)}$$

$$\pi_i = P(s_i) = \text{erwartete relative Häufigkeit von } s_i \text{ zum Zeitpunkt } (k=1) = \gamma_1(i)$$

Anwendungen



- Gensequenzanalyse
  - Beispiel: MPE-Anfrage mit RNA-Strang als Evidenz E, DNA-Sequenz als latenter Zustand V
    - $\mathrm{Val}(V) = \{A, C, G, T\}, \mathrm{Val}(E) = \{A, C, G, U\}$
    - Änderungen durch: Baustein fällt weg, kommt hinzu, wird falsch übersetzt
- Signalverarbeitung
  - Rauschen herausfiltern: Was war der ursprünglich gesendete Wert $\{0,1\}$? → MPE-Anfrage

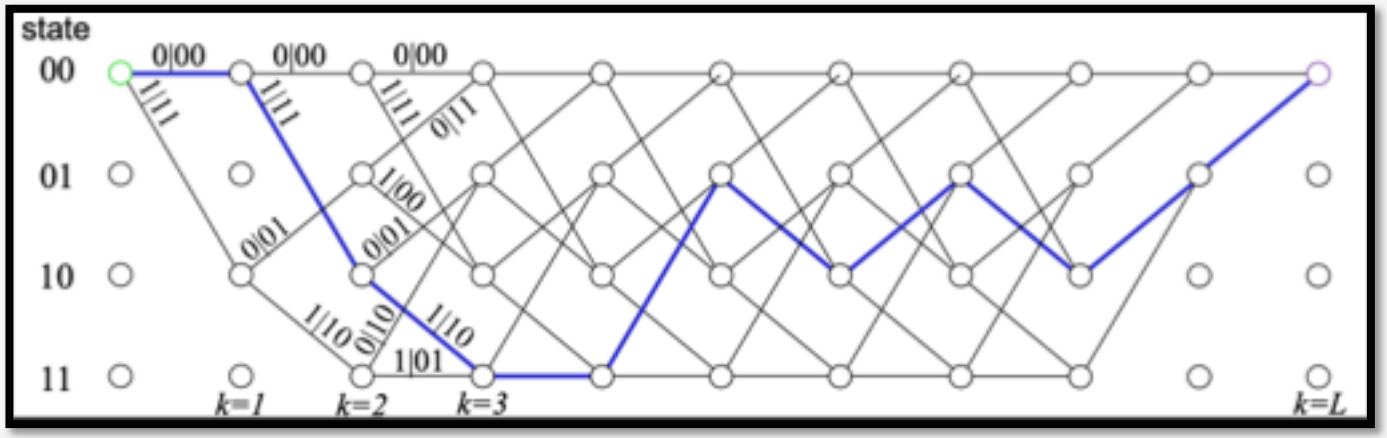
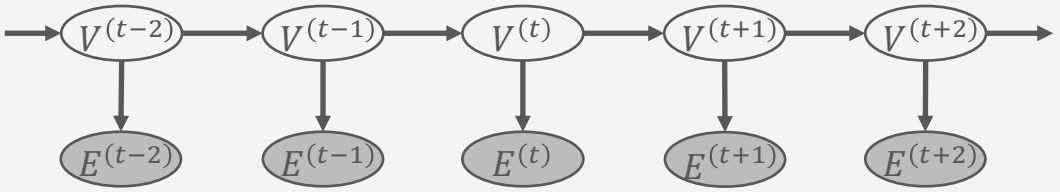
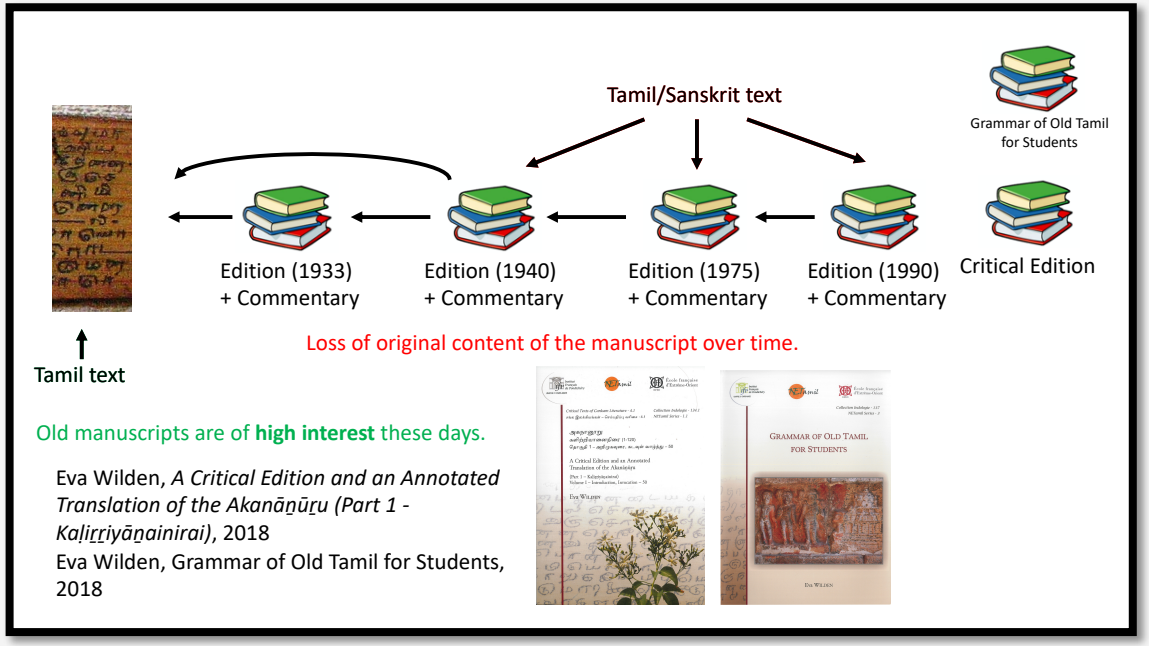


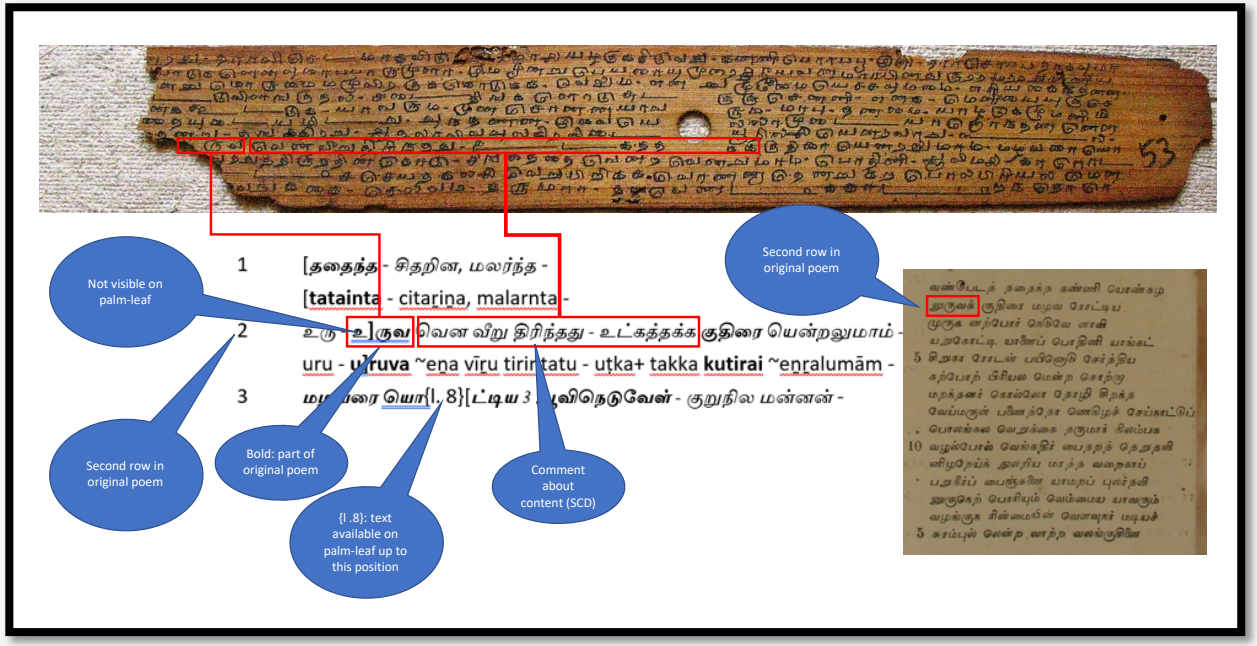
Abbildung: Gerhard Bauch, Skript Informations- und Codierungstheorie – SoSe 2014. TUHH, 2014.

Anwendungen



- Textverarbeitung
  - Gedichte in Tamil, angereichert mit Kommentaren im Laufe der Jahrhunderte
  - MPE-Anfrage: Was im Text ist Kommentar, was ist Gedicht?





Bender, B, Gehrke, Kuhr, Möller und Schiff: Identifying Subjective Content Descriptions among Texts.
In: *ICSC-21 Proceedings of 15th IEEE International Conference on Semantic Computing*, 2021.
Abbildungen: Vortrag Felix Kuhr, ICSC 2021.

Zwischenzusammenfassung

- Direkte Interpretation der Wahrscheinlichkeiten in HMMs möglich
  - Vorwärts- und Rückwärtsnachrichten
  - Viterbi-Algorithmus für MPEs
- Berechnungen über Matrixoperationen
  - Erlauben effiziente Umsetzung
  - Country-Dance-Algorithmus um Vorwärtsnachrichten beim Rückwärtsgehen zu rekonstruieren
- Baum-Welch-Algorithmus
  - Interpretation eines HMMs als Zustandsübergangssystem
  - Lernen der Parameter eines HMMs ML- und EM-basiert

Überblick: 6. Sequentielle PGMs und Inferenz

A. Sequentielle PGMs

- Templates, dynamische BNs, dynamische Faktormodelle, Hidden-Markov-Modelle; Semantik
- Inferenzaufgaben: Wahrscheinlichkeitsanfragen (Filtering, Prediction, Hindsight), Zustandsanfragen (MPE, MAP)

B. Sequentielle Inferenz

- Naïve Inferenz mittels Ausrollen, Interface Algorithmus, Komplexität, Approximationen

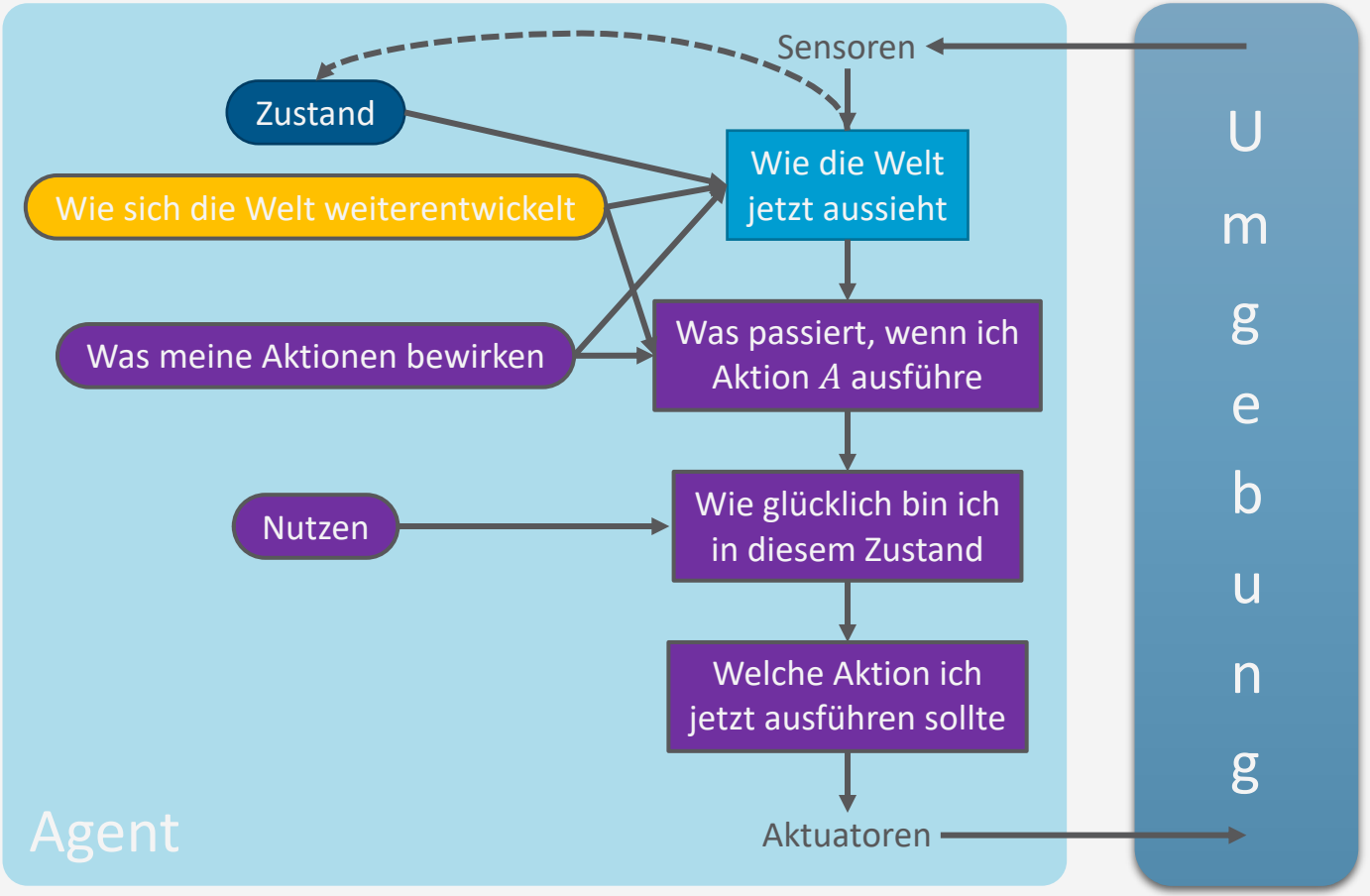
C. Spezialfall Hidden-Markov-Modelle

- Viterbi-Algorithmus für MPEs
- Anfragebeantwortung durch Matrixoperationen
- Baum-Welch-Algorithmus zum Lernen

→ Entscheidungstheoretische PGMs und Inferenz

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



„Anhang“

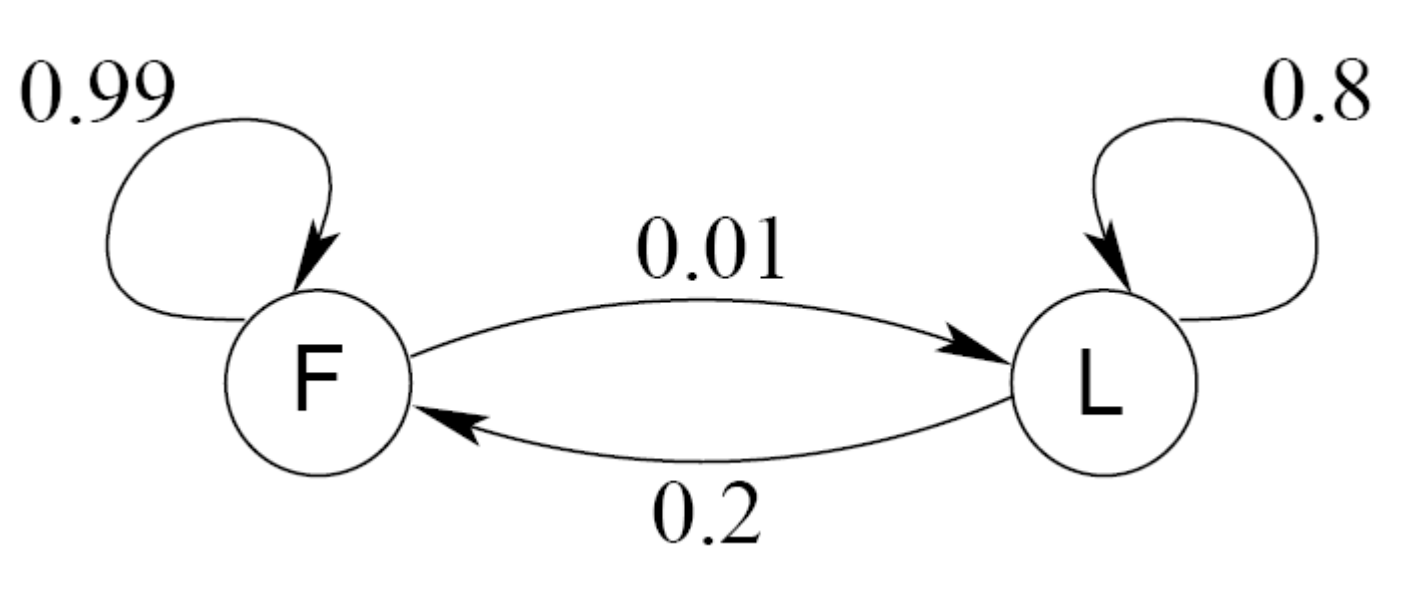
Nachfolgende Folien beinhalten ein nicht genutztes Beispiel zum Viterbi-Algorithmus

ACHTUNG: Folien sind nicht ordentlich gesetzt oder übersetzt!

The occasionally dishonest casino

- A casino uses a fair die most of the time, but occasionally switches to a loaded one
  - Fair die: Prob(1) = Prob(2) = . . . = Prob(6) = 1/6
  - Loaded die: Prob(1) = Prob(2) = . . . = Prob(5) = 1/10, Prob(6) = ½
  - These are the emission probabilities
- Transition probabilities
  - Prob(Fair $\rightarrow$ Loaded) = 0.01
  - Prob(Loaded $\rightarrow$ Fair) = 0.2
  - Transitions between states modeled by a Markov process

Transition model for the casino



The occasionally dishonest casino

- Known:
  - The structure of the model
  - The transition probabilities
- Hidden: What the casino did
  - FFFFFLLLLLLLFFFF...
- Observable: The series of die tosses
  - 3415256664666153...
- What we must infer:
  - When was a fair die used?
  - When was a loaded one used?
    - The answer is a sequence FFFFFFFLLLLLLFFF...

Making the inference

- Model assigns a probability to each explanation of the observation:

  $P(326|FFL)$
  $= P(3|F)\cdot P(F\rightarrow F)\cdot P(2|F)\cdot P(F\rightarrow L)\cdot P(6|L)$
  $= 1/6 \cdot 0.99 \cdot 1/6 \cdot 0.01 \cdot ½$

  Determine which explanation is most likely

  - Find the path *most likely* to have produced the observed sequence

    Determine probability that observed sequence was produced by the model

  - Consider *all* paths that could have produced the observed sequence

Notation

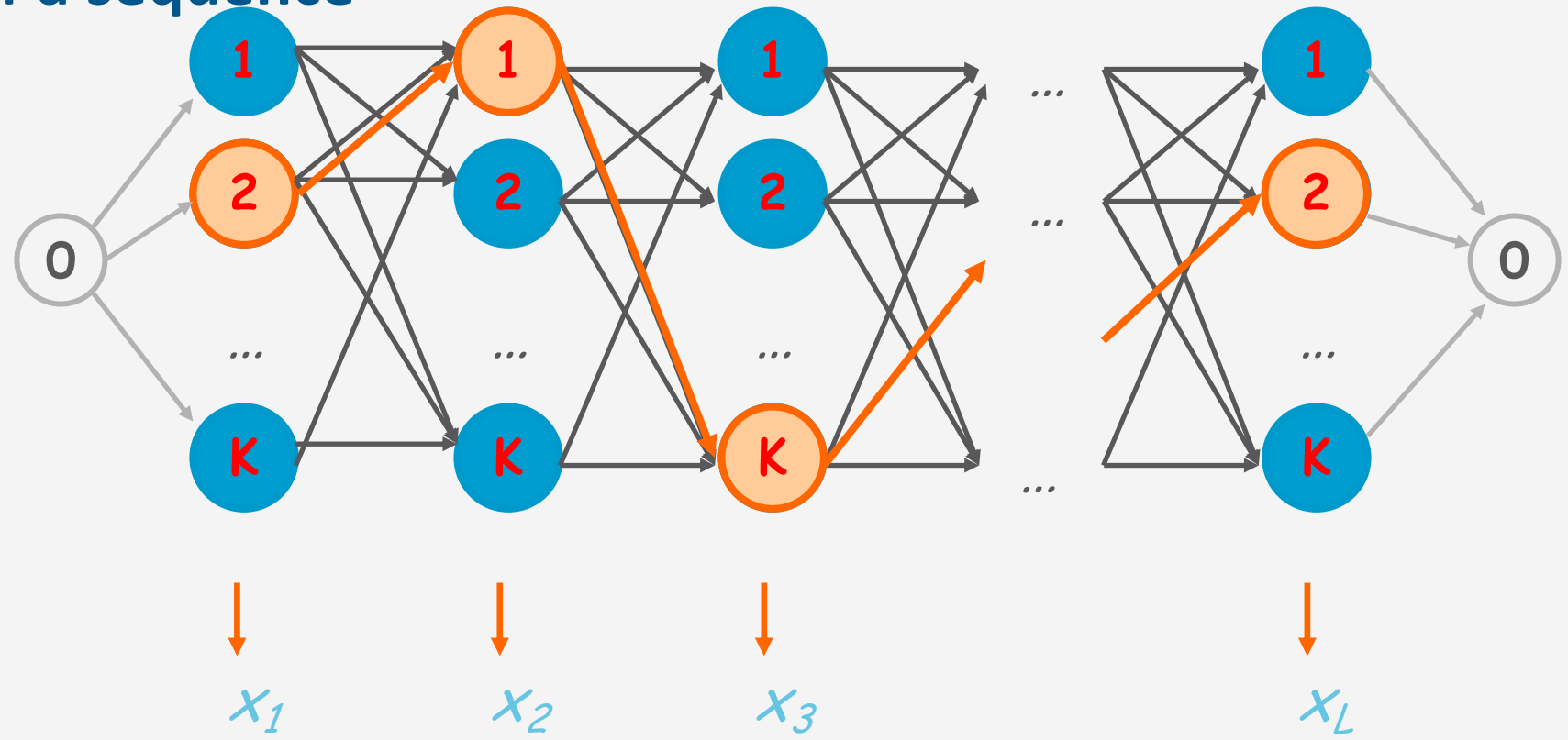
- x is the sequence of symbols emitted by model
  - x_i is the symbol emitted at time i
- A ***path***, π, is a sequence of states
  - The i-th state in π is π_i
- a_{kr} is the probability of making a transition from state k to state r:

$$a_{kr} = \Pr(\pi_i = r \mid \pi_{i-1} = k)$$

- $e_k(b)$ is the probability that symbol b is emitted when in state k

$$e_k(b) = \Pr(x_i = b \mid \pi_i = k)$$

A “parse” of a sequence



$$\Pr(x,\pi) = a_{0\pi_1}\prod_{i=1}^{L} e_{\pi_i}(x_i)\cdot a_{\pi_i\pi_{i+1}}$$

The occasionally dishonest casino

$$x = \langle x_1, x_2, x_3 \rangle = \langle 6, 2, 6 \rangle$$

$\pi^{(1)} = FFF$

$\pi^{(2)} = LLL$

$$\Pr(x, \pi^{(2)}) = a_{0L} e_L(6) a_{LL} e_L(2) a_{LL} e_L(6)$$
$$= 0.5 \times 0.8 \times 0.1 \times 0.8 \times 0.5$$
$$= 0.008$$

$\pi^{(3)} = LFL$

$$\Pr(x, \pi^{(3)}) = a_{0L} e_L(6) a_{LF} e_F(2) a_{FL} e_L(6) a_{L0}$$
$$= 0.5 \times 0.5 \times 0.2 \times \frac{1}{6} \times 0.01 \times 0.5$$
$$\approx 0.0000417$$

The most probable path

The most likely path π^* satisfies

$$\pi^* = \arg\max_{\pi} \Pr(x, \pi)$$

To find π^*, consider all possible ways the last symbol of x could have been emitted

Let

$$v_k(i) = \text{Prob. of path } \langle \pi_1, \cdots, \pi_i \rangle \text{ most likely to emit } \langle x_1, \ldots, x_i \rangle \text{ such that } \pi_i = k$$

Then

$$v_k(i) = e_k(x_i) \max_r \left(v_r(i-1) a_{rk} \right)$$

The Viterbi Algorithm

- Initialization ($i = 0$)

$$v_0(0) = 1, \quad v_k(0) = 0 \text{ for } k > 0$$

- Recursion ($i = 1, \ldots, L$): For each state k

$$v_k(i) = e_k(x_i) \max_r \left(v_r(i-1) a_{rk} \right)$$

- Termination:

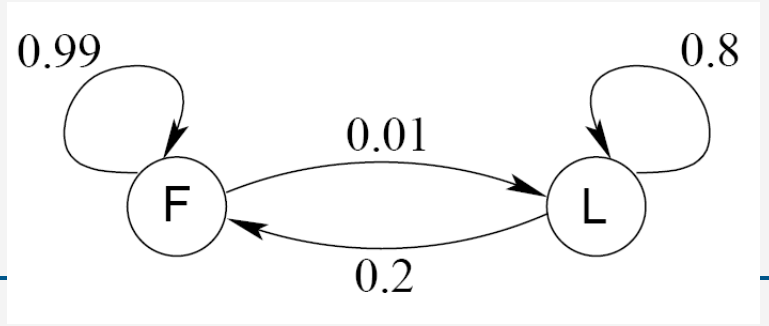
$$\Pr(x, \pi^*) = \max_k \left(v_k(Length) a_{k0} \right)$$

To find π^*, use trace-back, as in dynamic programming

Viterbi: Example

		ε	6	2	6
				x	
	B	1	0	0	0
π	F	0	(1/6)×(1/2) = 1/12	(1/6)×max{(1/12)×0.99, (1/4)×0.2} = 0.01375	(1/6)×max{0.01375×0.99, 0.02×0.2} = 0.00226875
	L	0	(1/2)×(1/2) = 1/4	(1/10)×max{(1/12)×0.01, (1/4)×0.8} = 0.02	(1/2)×max{0.01375×0.01, 0.02×0.8} = 0.08

$$v_k(i) = e_k(x_i)\max_r\left(v_r(i-1)a_{rk}\right)$$



Viterbi gets it right more often than not

```
Rolls    31511624644664424532113163116415213362514454363165662656666
Die      FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLL
Viterbi  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLL
---------------------------------------------------------------------
Rolls    65116645313265124563666463163666316232645523526666625151631
Die      LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLLFFFLLLLLLLLLLLLLLFFFFFFFFF
Viterbi  LLLLLLFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFF
---------------------------------------------------------------------
Rolls    22255544166656656356432436413151346514635341112641462625335 6
Die      FFFFFFFFLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
Viterbi  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFL
---------------------------------------------------------------------
Rolls    36616366646623253441366166116325256246225526525226643535333 6
Die      LLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Viterbi  LLLLLLLLLLLLFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
---------------------------------------------------------------------
Rolls    23312162536441443233516324363366556246666263266661235524524 2
Die      FFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
Viterbi  FFFFFFFFFFFFFFFFFFFFFFFFFFFLLLLLLLLLLLLLLLLLLLFFFFFFFFFF
```