Grundlagen der Künstlichen Intelligenz

Wintersemester 25/26

Vorlesung 12

Variablen-basierte Modelle, Bayes‘sche Netze, Hidden-Markov-Modelle

26. Januar 2026

T.T.-Prof. Dr. Peer Nowack
Prof. Dr. Pascal Friederich

Wir werden immer mal wieder Umfragen machen…

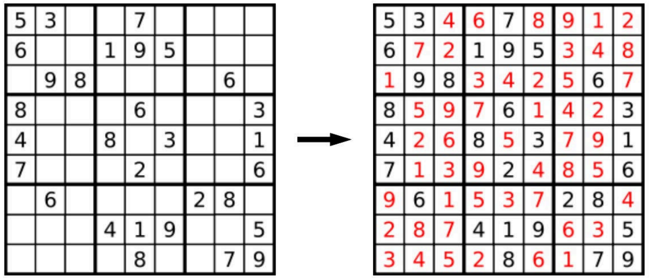
Sie können sich bereits einloggen. Nutzen Sie das KIT Wi-Fi bei schlechtem Empfang.

1. Go to wooclap.com
2. Enter the event code in the top banner

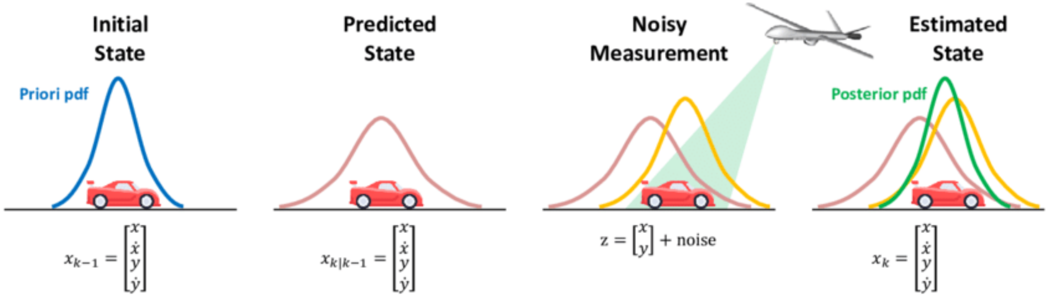
Event code
HIPFAX

Zustandsbasierte vs. variablenbasierte Modelle

- In zustandsbasierten Modellen sind Lösungen Aktionen (actions): schrittweise Anweisungen, die in einer Reihenfolge nacheinander ausgeführt werden.
- In vielen Anwendungen gibt es jedoch **keine Aktionen** und **die Reihenfolge ist nicht wichtig.**



Sudoku



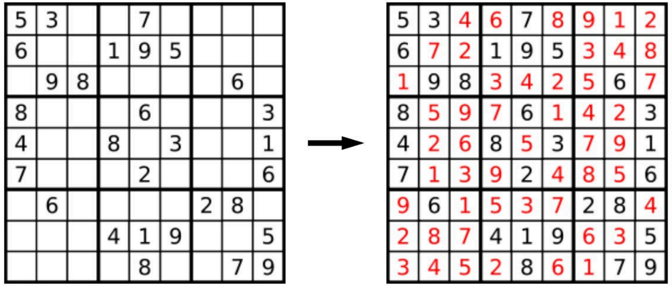
Zustandsschätzung

- Könnte man als Suchproblem formulieren (in dem die Reihenfolge wichtig ist), aber das wäre ineffizient, weil die Reihenfolge eben keine Rolle spielt…wollen für diesen Fall alternative Algorithmen kennenlernen!

Variablenbasierte Modelle

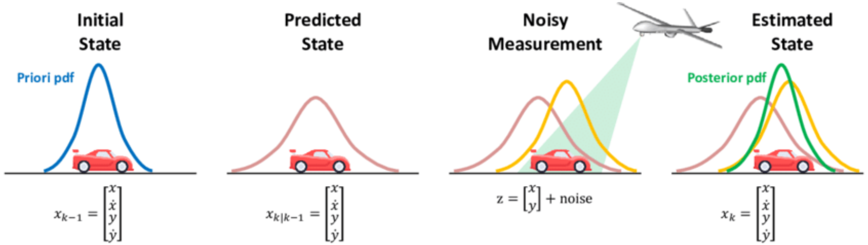
Probleme mit (harten) Randbedingungen:
(nicht in dieser Vorlesung)

- z. B. Sudoku, Stundenplan
- Modell: Constraint Satisfaction Problems



Probleme mit probabilistischen (weichen) Abhängigkeiten:

- Z.B. Sensordaten → Position von Autos
- Modell: Bayes'sche Netze

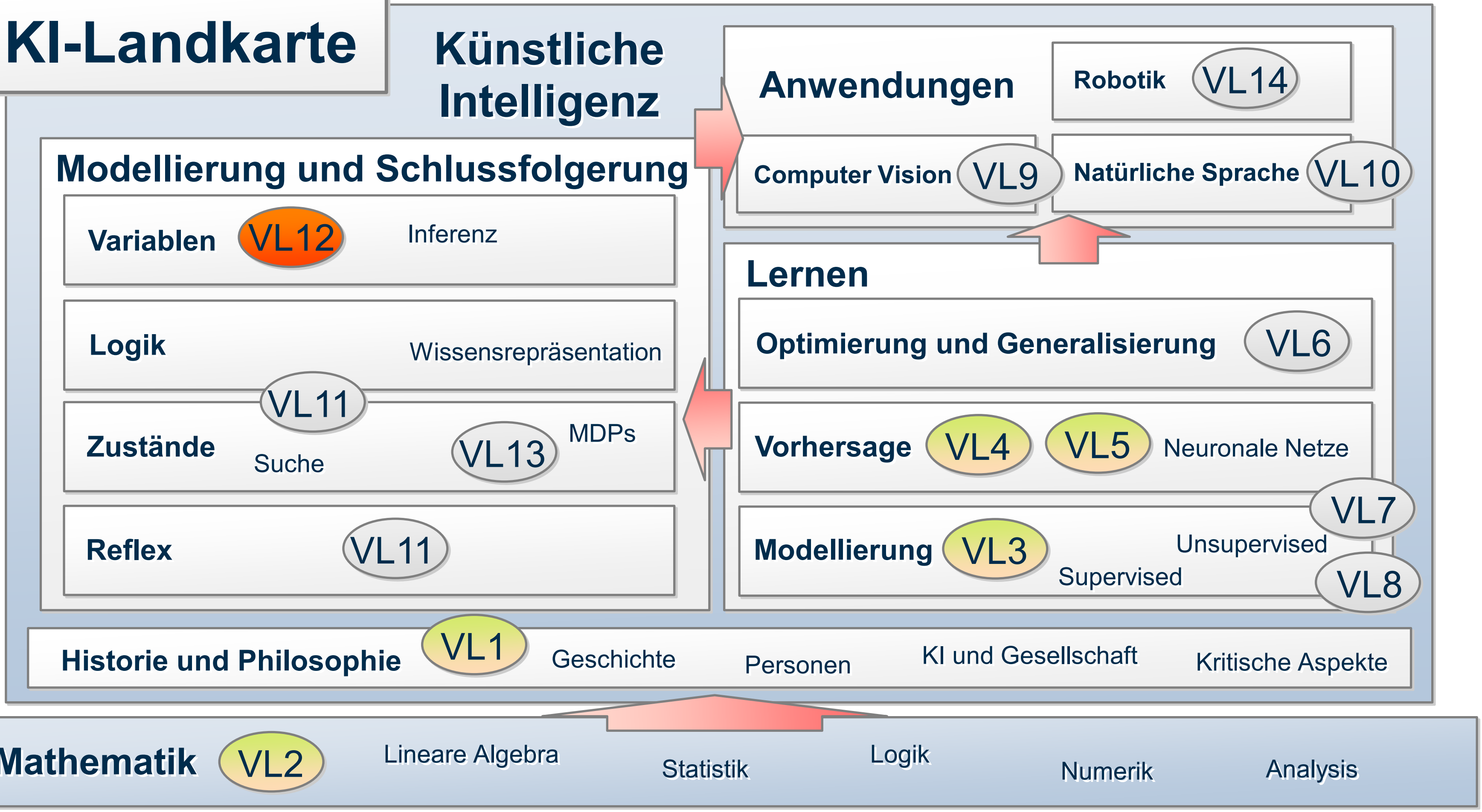


Modellierung
Schlussfolgerung Lernen

Reflex Zustände Variablen Logik

„low-level Intelligenz" Künstliche Intelligenz „high-level Intelligenz"

KI-Landkarte



Lernziele der heutigen Vorlesung

Nach dieser Vorlesung werden Sie in der Lage sein:

- Grundideen der **probabilistischen Inferenz** zu erklären.
- **Bayes'sche Netze** in ihrer mathematischen Idee und als grafische Modelle zu formulieren und einfache Berechnungen anzustellen, sowie den **Explaining-Away Effekt** zu bewerten.
- **Probabilistische Programme** zu entwerfen und einige Anwendungsbeispiele zusammenzufassen.
- Das Konzept von **Markov Modellen** zu erklären, sowie darauf aufbauend zentrale Ideen der **Hidden Markov Modelle** in Beziehung zu setzen, um den **Forward-Backward Algorithmus** zu formulieren, inklusive **Filterung** und **Glättung**.

Kapitel 12, 13, 14 in Norvig & Russell, 4. Edition

Agenda

Bayessche Netze (Bayesian networks)

- Definition
- Explaining Away
- Inferenz

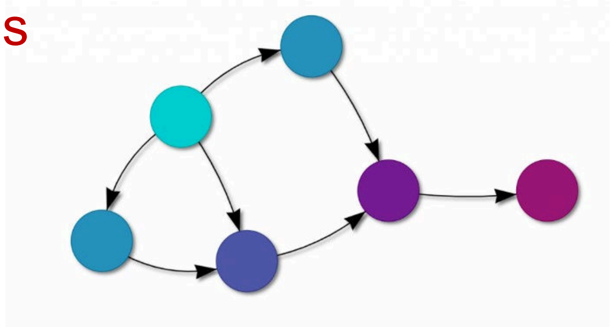
Probalistische Programme

Hidden Markov Modelle

- Zustandsabschätzung und Filtern (state estimation and filtering)
- Vorwärts Algorithmus (forward algorithm): Filtern
- Vorwärts Rückwärts Algorithmus (forward backward algorithm): Glättung (smoothing)

Warum brauchen wir Bayes‘sche Netze?

- Grundlegendes Framework für probabilistisches Schlussfolgern/probabilistic reasoning.



- Umgang mit heterogenen Informationslücken.
- Einbeziehung von Vorwissen (z. B. physikalische Gesetze)
- Kann die Variablen aller Zwischenschritte interpretieren

Recap: probabilistische Inferenz

Inferenz: Schlussfolgerungen aus vorhandenen Informationen ziehen; Evidenz dazu nutzen etwas zu erklären das wir nicht direkt sehen oder sicher wissen können.

Beispiele:

- Nasser Bürgersteig → es hat wahrscheinlich geregnet
- Handybatterie fällt schnell ab → eine App könnte verantwortlich sein

Normalerweise: von Unsicherheiten behaftet → probabilistisch

- Es hat wahrscheinlich geregnet.
- Es könnte die eine oder die andere App sein.

Recap: probabilistische Inferenz

Zufallsvariablen: S (Sonnenschein), R (Regen), T (Verkehr), A (Herbst)

Gemeinsame Verteilung (probabilistische Datenbasis): $p(S, R, T, A)$

Probabilistische Inferenz ist die Beantwortung von Fragen anhand dieser Daten:

- Kondition/Bedingung für die Evidenz (Verkehr, Herbst): T = 1, A = 1
- Interessiert an Query (Regen?): R
- Einige Variablen (S) sind nicht von Interesse – marginalisieren

$$p(R|T=1, A=1) = \sum_S p(\underbrace{R}_{\text{query}}, \underbrace{S}_{\text{marginalize out}} \mid \underbrace{T=1, A=1}_{\text{condition}})$$

- Die Menge der Konditionierungs-Variablen, der Query-Variablen und der marginalisierten Variablen, bilden eine **Partitionierung** aller Variablen.

Recap: Satz von Bayes

Der Satz von Bayes ist eine der wichtigsten Gleichungen in der Wahrscheinlichkeitsrechnung und im maschinellen Lernen

$$\underbrace{p(z|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|z)}^{\text{Likelihood}}\overbrace{p(z)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}} = \frac{p(y|z)p(z)}{\sum_{z'} p(y|z')p(z')}$$

- "Umkehrung" der bedingten Wahrscheinlichkeiten
- Oft ist eine Bedingung kompliziert, die andere aber einfach zu erhalten

Recap: Satz von Bayes

Der Satz von Bayes mit „Hintergrundinformationen *x*"

Likelihood **Prior**

$$p(z|y,x) = \frac{p(y|z,x)p(z|x)}{p(y|x)} = \frac{p(y|z,x)p(z|x)}{\sum_{z'} p(y|z',x)p(z'|x)}$$

Posterior **Evidence**

- Alle Verteilungen sind jetzt auch von *x* abhängig

Bayes'sche Netze (BN): Definition

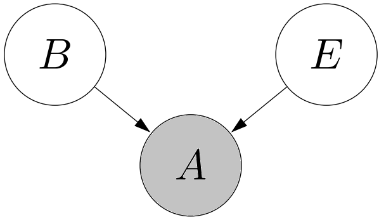
- Gegeben Zufallsvariablen $\boldsymbol{X} = (X_1, \ldots, X_n)$
- Ein BN ist ein Directed Acyclic Graph (DAG), der eine gemeinsame Verteilung über $\boldsymbol{X}$ als Produkt lokaler bedingter Verteilungen spezifiziert (eine pro Knoten):

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{\text{parents(i)}})$$

$$p(A,B,E) = p(B)\,p(E)p(A|B,E)$$

Häufig als grafisches Modell dargestellt:

- Beispiel: 3 Zufallsvariablen (B, A, E)
- Eltern werden durch gerichtete Kanten angezeigt
- Beobachtete Variablen werden oft in grau dargestellt



Grafische Modelle können zum Ablesen von Unabhängigkeitsannahmen und Faktorisierungen verwendet werden, die die Berechnungen vereinfachen.

Ein kleines Rätsel...

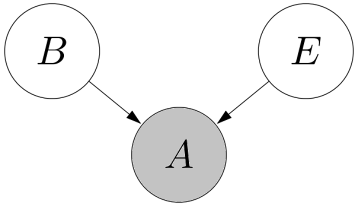
Problemstellung: Erdbeben (E), Einbrüche (B) und Alarme (A)

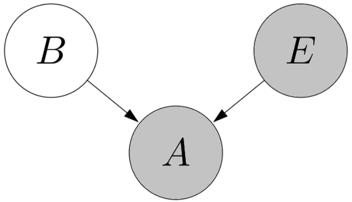
- **Erdbeben** und **Einbrüche** sind unabhängige Ereignisse (Wahrscheinlichkeit ϵ).
- In beiden Fällen wird ein **Alarm** ausgelöst.
- Angenommen, wir erhalten einen **Alarm.**
- Erhöht oder verringert sich die Wahrscheinlichkeit eines Einbruchs, wenn man hört, dass es ein Erdbeben gibt, oder bleibt sie gleich?

Gemeinsame Verteilung: $p(E, B, A) = p(E)p(B)p(A|B, E)$

Fragen:

$p(B = 1|A = 1)$ **?** $p(B = 1|A = 1, E = 1)$





Bayes'sches Netz

- Alarm wird ausgelöst, wenn entweder ein Erdbeben oder Einbruch, oder beides eintritt

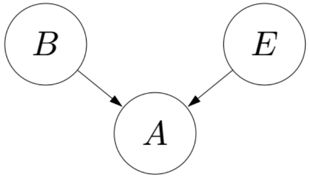
$$p(E=e) = \epsilon^e(1-\epsilon)^{1-e}$$

$$p(B=b) = \epsilon^b(1-\epsilon)^{1-b}$$

$$p(A=a|e,b) = [a=(b \vee e)]$$

- Gemeinsame Verteilung

$$p(E,B,A) = p(E)p(B)p(A|B,E)$$



Wahrscheinlichkeit Einbruch

b	$p(b)$
1	ϵ
0	$1-\epsilon$

Wahrscheinlichkeit Erdbeben

e	$p(e)$
1	ϵ
0	$1-\epsilon$

b	e	a	$p(a \mid b,e)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Bedingte p für Alarm

Joint distribution

b	e	a	$\mathbb{P}(B=b,E=e,A=a)$
0	0	0	$(1-\epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1-\epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1-\epsilon)$
1	1	0	0
1	1	1	ϵ^2

Probabilistische Inferenz

Joint distribution

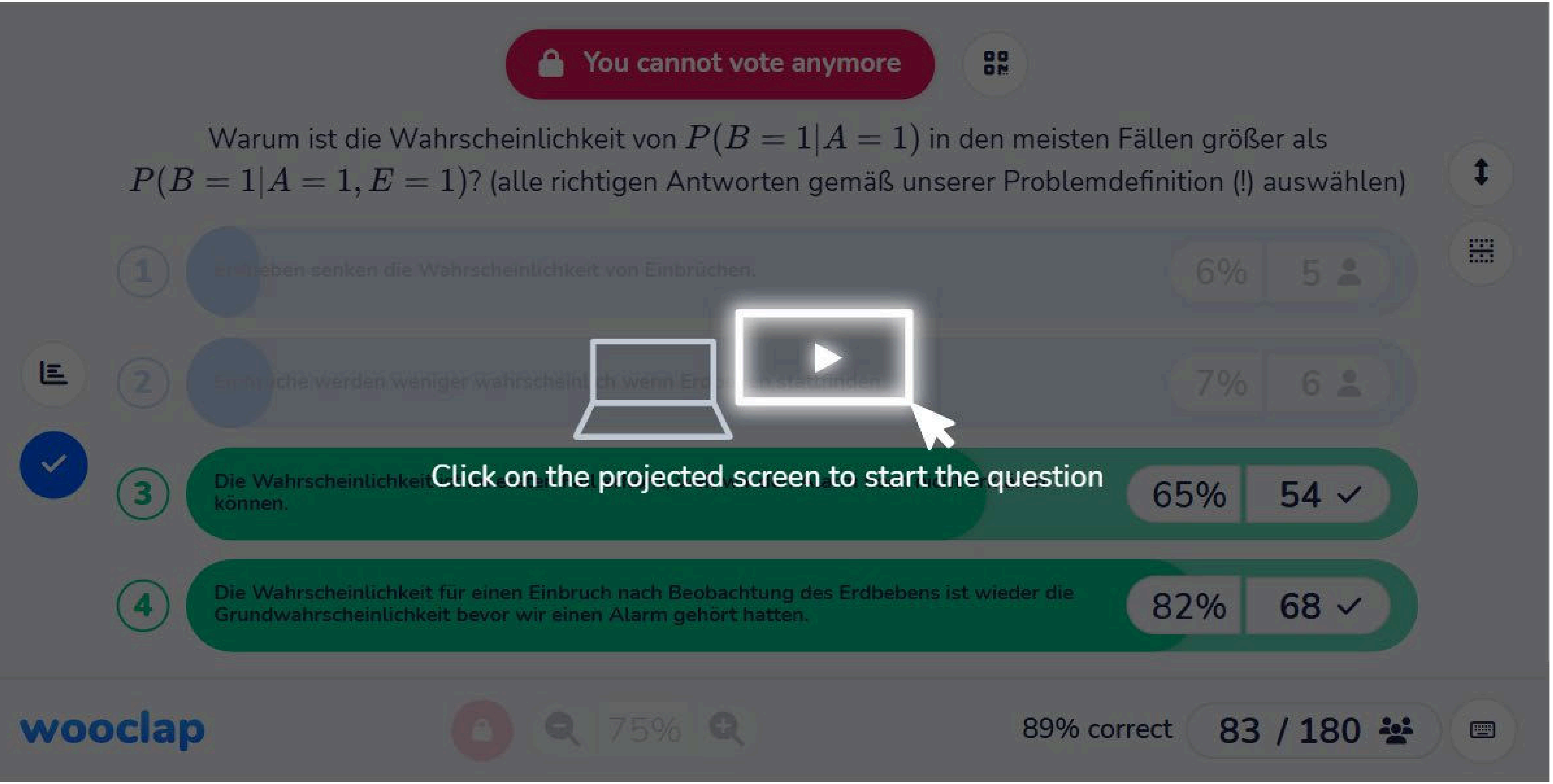
b	e	a	$\mathbb{P}(B=b,E=e,A=a)$
0	0	0	$(1-\epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1-\epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1-\epsilon)$
1	1	0	0
1	1	1	ϵ^2

Query:

$$P(B=1) = \epsilon(1-\epsilon) + \epsilon^2 = \epsilon$$

$$P(B=1|A=1) = \frac{P(B=1,A=1)}{p(A=1)} = \frac{\epsilon(1-\epsilon)+\epsilon^2}{\epsilon(1-\epsilon)+\epsilon^2+(1-\epsilon)\epsilon} = \frac{1}{2-\epsilon}$$

$$P(B=1|A=1,E=1) = \frac{P(B=1,A=1,E=1)}{p(A=1,E=1)} = \frac{\epsilon^2}{\epsilon^2+(1-\epsilon)\epsilon} = \epsilon$$



$$P(B=1|A=1) = \frac{1}{2-\epsilon} \qquad P(B=1|A=1,E=1) = \epsilon$$

Erdbeben (E), Einbrüche (B) und Alarme (A)

Probabilistische Inferenz

Joint distribution

b	e	a	$\mathbb{P}(B=b, E=e, A=a)$
0	0	0	$(1-\epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1-\epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1-\epsilon)$
1	1	0	0
1	1	1	ϵ^2

Query:

$$P(B=1) = \epsilon(1-\epsilon) + \epsilon^2 = \epsilon$$

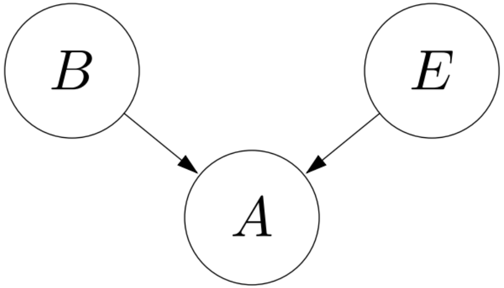
$$P(B=1|A=1) = \frac{P(B=1, A=1)}{p(A=1)} = \frac{\epsilon(1-\epsilon)+\epsilon^2}{\epsilon(1-\epsilon)+\epsilon^2+(1-\epsilon)\epsilon} = \frac{1}{2-\epsilon}$$

$$P(B=1|A=1, E=1) = \frac{P(B=1, A=1, E=1)}{p(A=1, E=1)} = \frac{\epsilon^2}{\epsilon^2+(1-\epsilon)\epsilon} = \epsilon$$

Eilmeldung: Erdbeben verringern Einbrüche!*

*Diese Aussage geht von einer Kausalität aus, die **nicht** gegeben ist.

Der Explaining-Away-Effekt



Hauptprinzip:

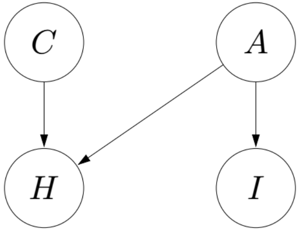
Angenommen zwei Ursachen beeinflussen eine Wirkung A positiv.
Konditioniert man zusätzlich zur Wirkung A auf eine der Ursachen, verringert dies die Wahrscheinlichkeit der anderen Ursache; hier:

$$p(B=1|A=1,E=1) \quad < \quad p(B=1|A=1)$$

Dies geschieht auch, wenn beide Ursachen unabhängig voneinander sind!

Ein anderes Beispiel: Erkältung oder Allergie?

Sie husten und haben juckende Augen. Haben Sie eine Erkältung?



C	p(c)(C)
0	0.9
1	0.1

A	p(a)(A)
0	0.8
1	0.2

C	A	H	p(h|c,a)(C,A,H)
0	0	0	0.9
0	1	1	0.9
1	0	1	0.9
1	1	1	0.9
0	0	1	0.1
0	1	0	0.1
1	0	0	0.1
1	1	0	0.1

A	I	p(i|a)(A,I)
0	0	0.9
1	1	0.9
0	1	0.1
1	0	0.1

Zufallsvariablen

Erkältung C, Allergien A, Husten H, juckende Augen I

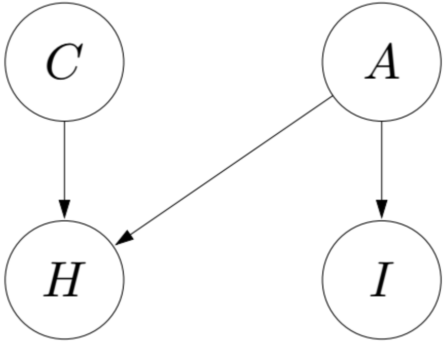
Gemeinsame Verteilung

$$p(c, a, h, i) = p(c)p(a)p(h|c, a)p(i|a)$$

Abfragen: $p(C = 1|H = 1) = 0.28 \quad > \quad P(C = 1|H = 1, I = 1) = 0.13$

Die Beobachtung "juckende Augen" unterstützt die Allergiehypothese, welche die Anzeichen für eine Erkältung ausräumt.

Probabilistische Inferenz (formale Definition)



Eingabe:

Bayes'sches Netzwerk: $p(X_1, \ldots, X_n)$

Evidenz: $E = e$ wobei $E \subseteq X$ ist Teilmenge von X

Query: $Q \subseteq X$

Ausgabe:

Berechnung der Wahrscheinlichkeiten für Query Q bei Evidenz $E = e$

$$p(Q = q | E = e) \text{ for all values q}$$

Beispiel:

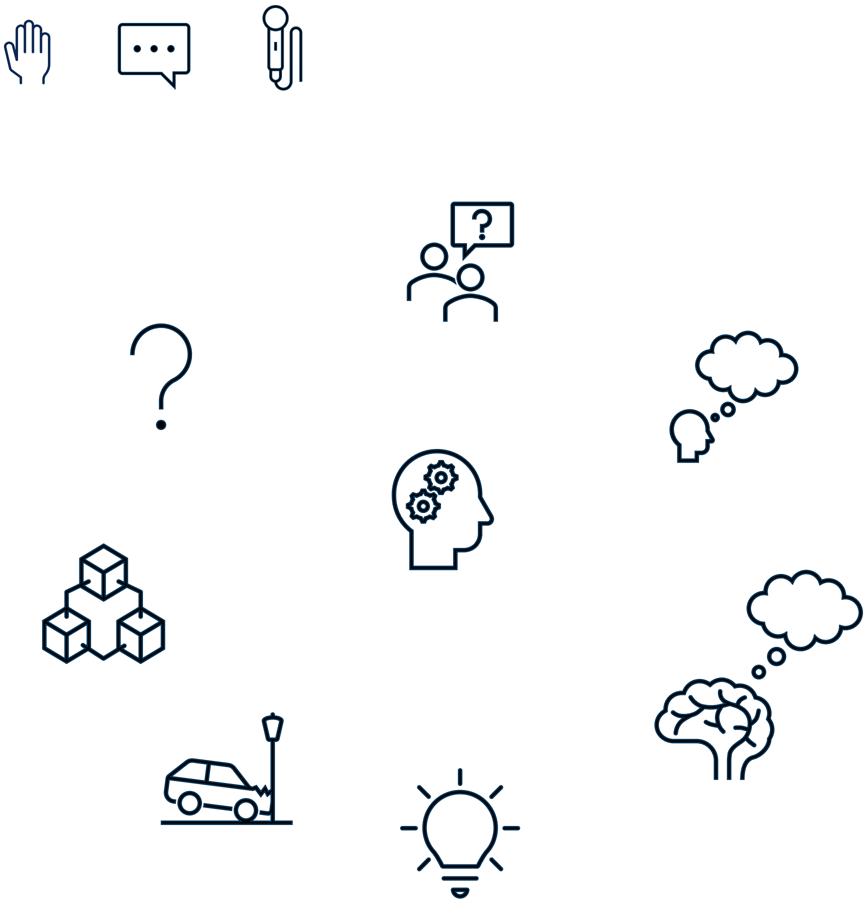
wenn Husten und juckende Augen, hat man dann eine Erkältung?

$$p(C | H = 1, I = 1)$$

Inferenz ist ein Kernproblem der KI:

- Es gibt viele Algorithmen, die die Inferenz effizienter machen
- Faktorgraphen, Message Passing, Expectation Propagation usw...

Fragen?



Agenda

Bayessche Netze (Bayesian networks)

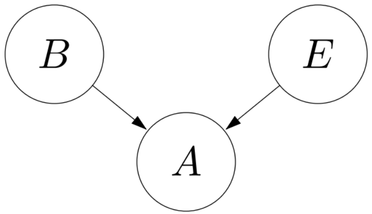
- Definition
- Explaining Away
- Inferenz

Probalistische Programme

Hidden Markov Modelle

- Zustandsabschätzung und Filtern (state estimation and filtering)
- Vorwärts Algorithmus (forward algorithm): Filtern
- Vorwärts Rückwärts Algorithmus (forward backward algorithm): Glättung (smoothing)

Probabilistische Programme



Probabilistic program: alarm

$$B \sim \text{Bernoulli}(\epsilon)$$
$$E \sim \text{Bernoulli}(\epsilon)$$
$$A = B \vee E$$

Gemeinsame Verteilung:

$$p(E, B, A) = p(E)p(B)p(A|B, E)$$

```
def Bernoulli(epsilon):
    return random.random() < epsilon
```

Der Schlüsselgedanke:

- Ein randomisiertes Programm, das die Zufallsvariablen setzt
- Auch "generatives Modell" oder "generativer Prozess" genannt

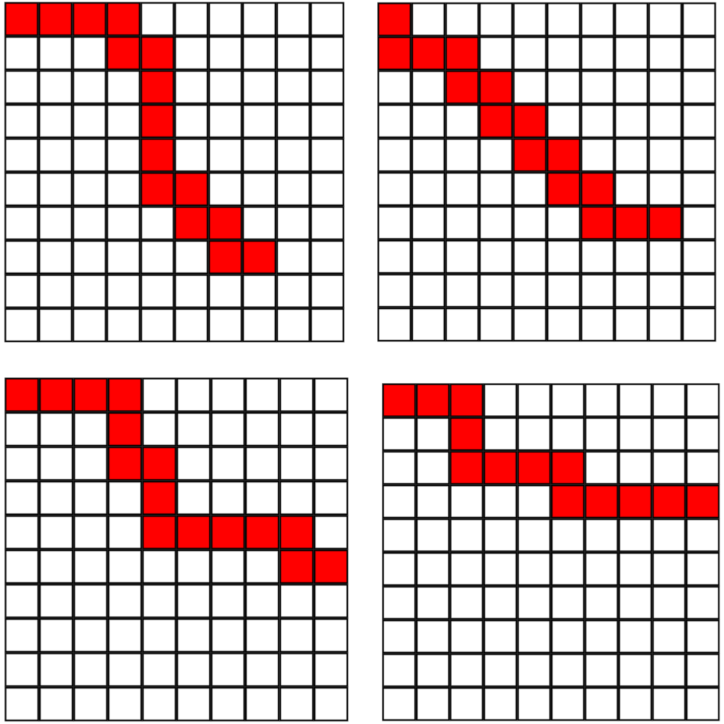
Beispiel: Objektverfolgung

Probabilistic program: object tracking

$X_0 = (0, 0)$
For each time step $i = 1, \ldots, n$:
  if Bernoulli(α):
    $X_i = X_{i-1} + (1, 0)$ [go right]
  else:
    $X_i = X_{i-1} + (0, 1)$ [go down]

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5$$

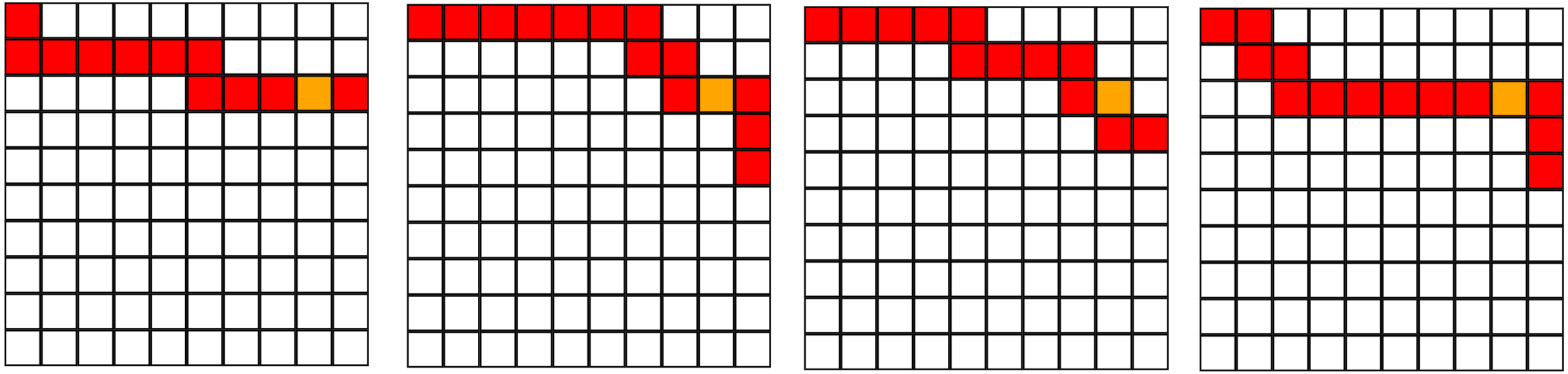
Trajectories for $\alpha = 0.5$



- Wir können beliebig viele Zufallsvariablen (d.h. Zeitschritte) mit Hilfe von for-Schleifen erzeugen
- **Markov-Modell:** Jedes X_i hängt nur von X_{i-1} ab

Beispiel für eine Schlussfolgerung

Frage: Was sind mögliche Wege angesichts des Evidenz? $X_{10} = (8, 2)$



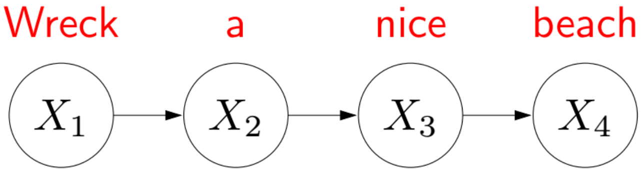
Anwendung: Sprachmodelle (z.B. Spracherkennung, maschinelle Übersetzung)

- Einfaches Modell: Wahrscheinlichkeit des neuen Worts hängt nur vom vorherigen Wort ab
- Besser: hängt von den letzten k Wörtern ab (**Markov-Modelle höherer Ordnung**)

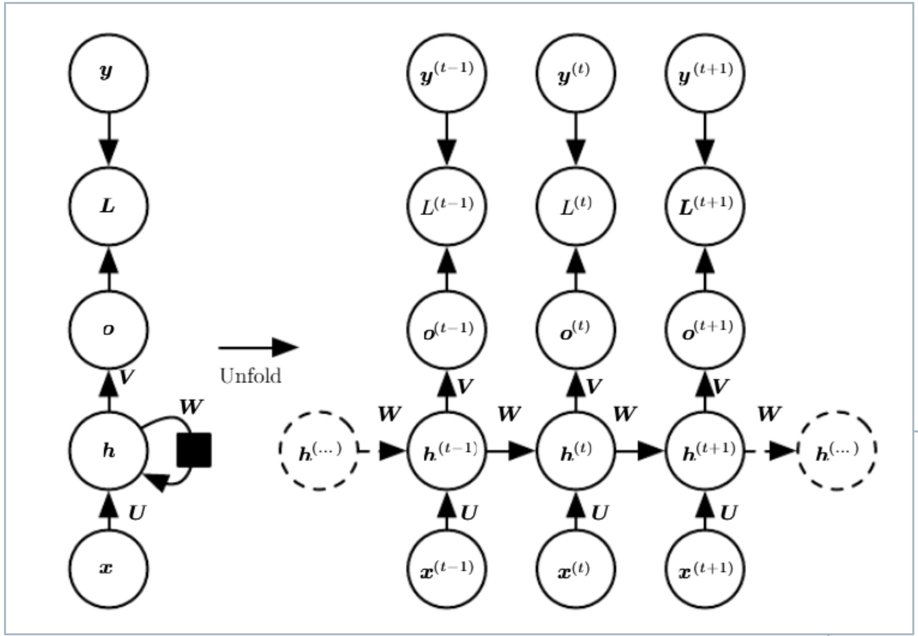
Probabilistic program: Markov model

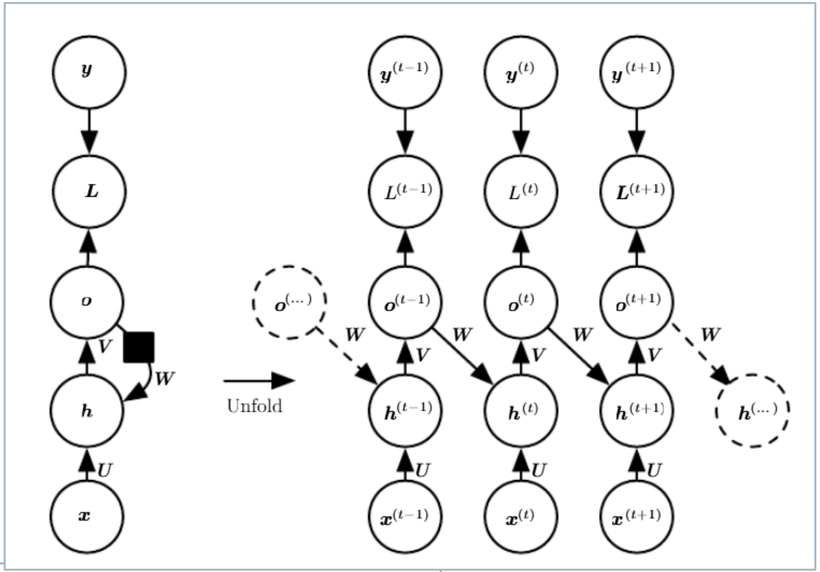
For each position $i = 1, 2, \ldots, n$:

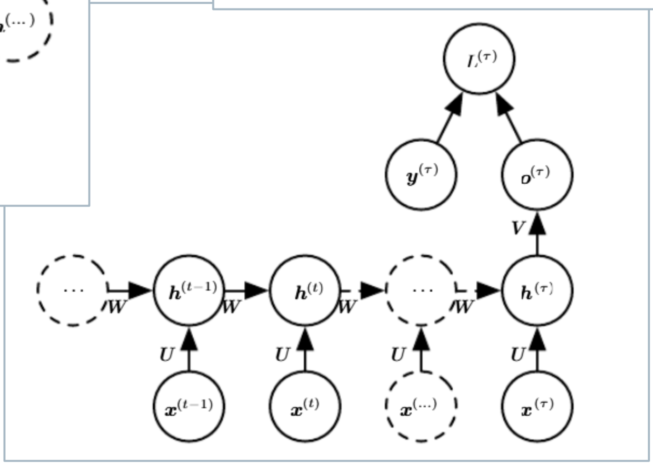
Generate word $X_i \sim p(X_i \mid X_{i-1})$



RNN-Architekturbeispiele







Markov models?

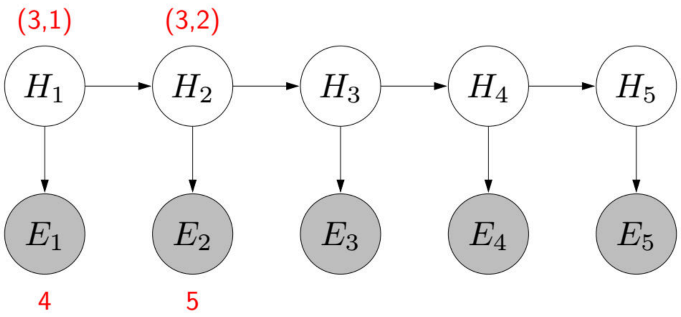
Anwendung: Objektverfolgung aus unsicheren Beobachtungen

Probabilistic program: hidden Markov model (HMM)

For each time step $t = 1, \ldots, T$:

Generate object location $H_t \sim p(H_t \mid H_{t-1})$

Generate sensor reading $E_t \sim p(E_t \mid H_t)$



- Wo befindet sich das Objekt aufgrund der Reihe von Sensormesswerten?
- Das allgemeine Problem wird als Zustandsschätzung bezeichnet

Agenda

Bayessche Netze (Bayesian networks)

- Definition
- Explaining Away
- Inferenz

Probalistische Programme

Hidden Markov Modelle

- Zustandsabschätzung und Filtern (state estimation and filtering)
- Vorwärts Algorithmus (forward algorithm): Filtern
- Vorwärts Rückwärts Algorithmus (forward backward algorithm): Glättung (smoothing)

Feedback Lehrevaluation

Positive Kommentare durch die LLM-Brille

Zusammenfassung (max. 5 Stichpunkte):

- Sehr hohe **Interaktivität** durch Umfragen, Quizze (z. B. Wooclap) und aktive Einbindung der Studierenden, die zum Mitdenken anregen.
- **Didaktisch starke Aufbereitung**: komplexe Inhalte werden verständlich erklärt, mit vielen anschaulichen Beispielen, Diagrammen und klaren Einführungen.
- **Angenehmer, motivierter Vortragsstil** des Dozenten: sympathisch, kompetent, offen, strukturiert und gut nachvollziehbar.
- **Praxis- und Anwendungsbezug** sowie hohe Aktualität der Themen, mit qualitativ hochwertiger Einordnung jenseits von bloßem „Hype".
- **Gute Lernunterstützung** durch sinnvolle Übungsblätter, Notebooks, deutsche Folien sowie kleine Anreize (z. B. Schokolade, humorvolle Elemente).

Einige Verbesserungsvorschläge (spezifisch)

- Folien: oft zu viele Formeln und zu voll; Mathematik recht anspruchsvoll (gerade live); nicht immer alleine ausreichend für Nachbearbeitung und Übungen.
- Auch aktuelle Vorlesungsaufzeichnungen bitte (mehrfach genannt…).
- (Noch) klarere Definitionen für einige Begriffe.
- Einige abstrakte Konzepte hätten noch weiter aufgedröselt werden können.
- Tempo am Ende oft erhöht (in den letzten 15-30 Minuten)…Schwerer zu folgen.
- Mathematische Notation mit LA/Numerik abstimmen.
- Wechsel zwischen Deutsch und Englisch, Notationen nicht immer konsistent.
- Mehr Quellenverweise für Hintergrund, gerade bei der Mathematik.
- Vorlesungsfolien früher hochladen.
- Zu viele englische Fachbegriffe die aus dem Nichts kommen.

Go to **wooclap.com** and use the code **HIPFAX**

Inhalte der Vorlesungen: "What could go? (-)" and "What should stay?" (+)
Vor Ihrer Antwort mit Plus oder Minus andeuten in welche Kategorie Ihre Antwort gehört. Vielen Dank!

wooclap

75%

18

Agenda

Bayessche Netze (Bayesian networks)

- Definition
- Explaining Away
- Inferenz

Probalistische Programme

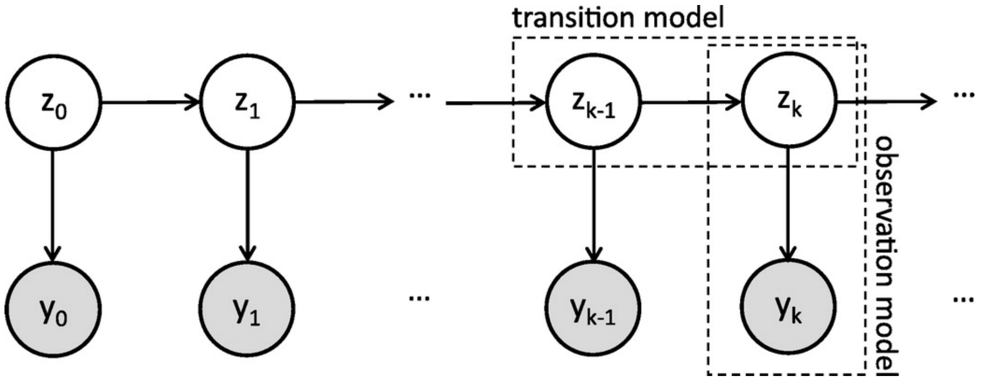
Hidden Markov Modelle

- Zustandsabschätzung und Filtern (state estimation and filtering)
- Vorwärts Algorithmus (forward algorithm): Filtern
- Vorwärts Rückwärts Algorithmus (forward backward algorithm): Glättung (smoothing)

Hidden Markov Modelle (HMM)

Betrachten **Markov-Kette** über die nicht beobachtbaren Zustände z_t und beobachten die Ausgaben y_t zu jedem Zeitschritt t

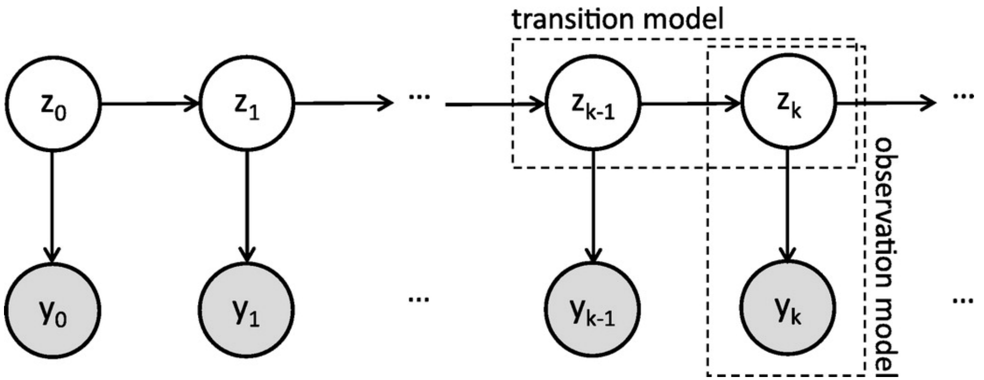
- Anfangsverteilung: $p(z_0)$
- Übergangsmodell: $p(z_{t+1}|z_t)$
- Beobachtungsmodell: $p(y_t|z_t)$



In Anwendungen mit kontinuierlichen Zuständen bezeichnen wir dieses Modell oft als Zustandsraummodell

- Gleiches Konstrukt wie ein HMM
- Wir haben wieder die Markov-Annahme für das Übergangsmodell
  Beispiel: Kalman-Filter.
- Hier: bleiben für Anschaulichkeit bei diskreten Zuständen.

Hidden Markov Modelle (HMM)



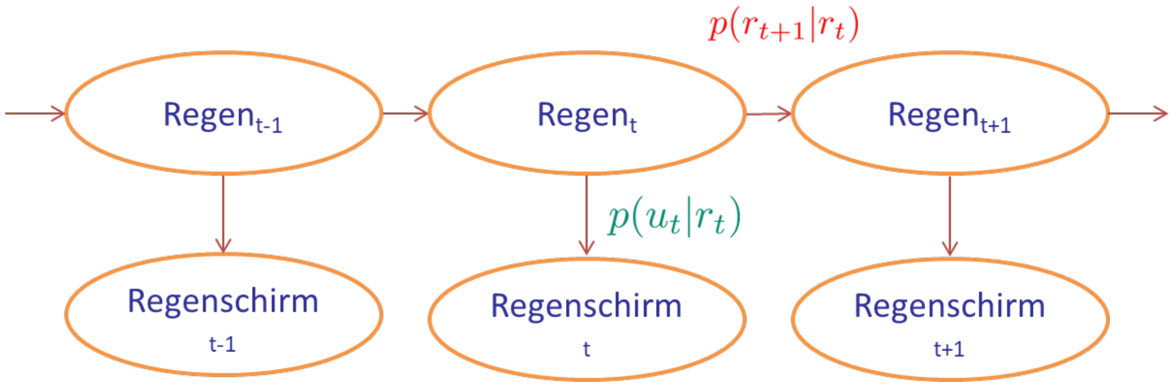
Gemeinsame Verteilung

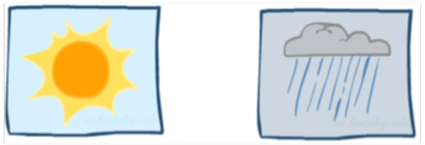
$$p(z_{0:t}, y_{0:t}) = p(z_0)p(y_0|z_0)\prod_{i=1}^{t} p(z_t|z_{t-1})p(y_t|z_t)$$

HMMs haben zwei wichtige Unabhängigkeitseigenschaften (ausnutzen!):

- Angesichts der Gegenwart ist die Zukunft unabhängig von der Vergangenheit
- Aktuelle Beobachtung unabhängig von allem anderen gegebenen aktuellen Zustand

Beispiel: Wetter-HMM





- **HMM besteht aus:**
  - Anfangsverteilung: $p(r_0)$
  - Transitionsmodell: $p(r_{t+1}|r_t)$
  - Beobachtungsmodell: $p(u_t|r_t)$

r_0	$P(r_0)$
+	0.6
-	0.4

r_t	r_{t+1}	$P(r_{t+1} \mid r_t)$
+	+	0.7
+	-	0.3
-	+	0.3
-	-	0.7

r_t	u_t	$P(u_t \mid r_t)$
+	+	0.9
+	-	0.1
-	+	0.2
-	-	0.8

Beispiel: Wetter-HMM

- Für diskrete Zufallsvariablen lassen sich die bedingten Verteilungen als Matrizen schreiben

r_t	r_{t+1}	$P(r_{t+1} \mid r_t)$
+	+	0.7
+	-	0.3
-	+	0.3
-	-	0.7

➡

r_{t+1} \ r_t	+	-
+	0.7	0.3
-	0.3	0.7

  – Spaltenindex: bedingende Variable (r_t), Zeilenindex: bedingte Variable (r_{t+1})

- **HMM besteht aus:**
  – Anfangsverteilung: **Vektor** $p(r_0)$
  – Übergangsmodell : **Matrix** $p(r_{t+1}|r_t)$
  – Beobachtungsmodell : **Matrix** $p(u_t|r_t)$

Hidden Markov Modelle Anwendungen

- HMMs für die Spracherkennung:
  - Beobachtungen sind akustische Signale (mit kontinuierlichem Wert)
  - Zustände sind bestimmte Positionen in bestimmten Wörtern (also Zehntausende)

- Maschinelle Übersetzung HMMs:
  - Beobachtungen sind Wörter (Zehntausende)
  - Staaten sind Übersetzungsoptionen

- Roboter-Verfolgung:
  - Beobachtungen sind Positionsmessungen (kontinuierlich)
  - Zustände sind Positionen auf einer Landkarte (kontinuierlich)

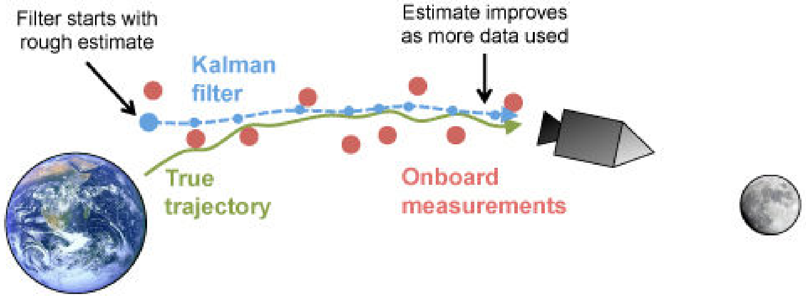
Sonar Pacman

Tracking mit HMMs

Normalerweise können wir nur verrauschte Messungen erhalten

Schätzung des "Zustands" anhand von Sensordaten:

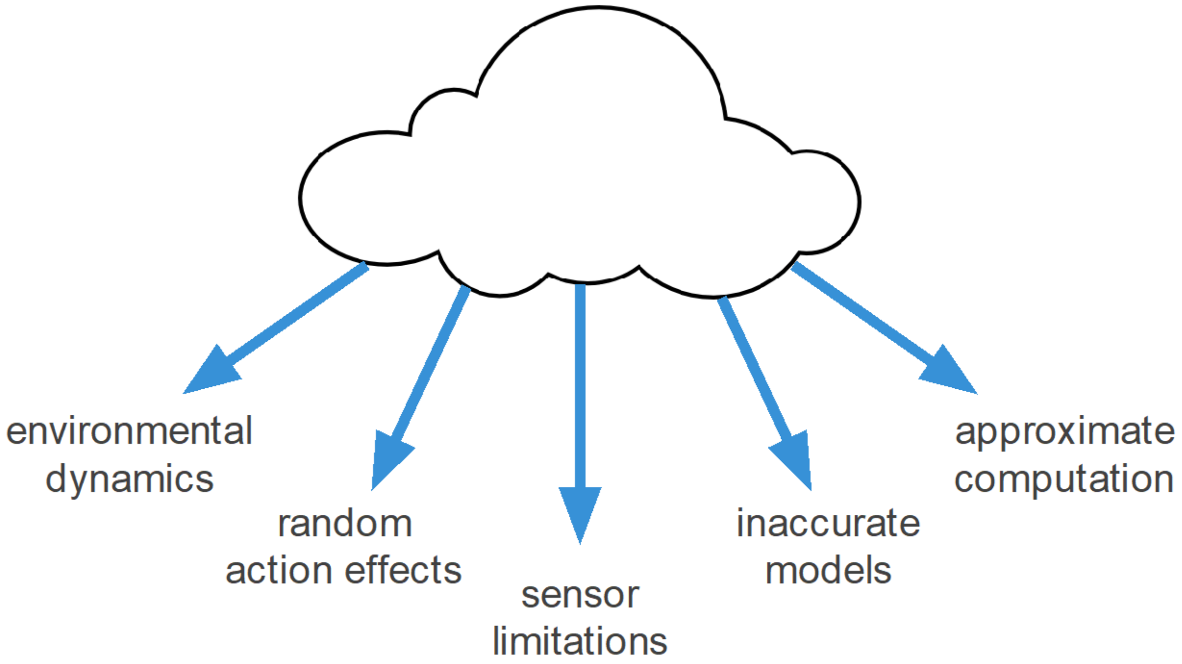
- Position und Ausrichtung des Roboters
- Karte der Umwelt
- Ortung von Personen, anderen Robotern, Objekten usw.



Indoor-Lokalisierung für mobile Roboter (gegebene Karte)

- Zusammenführung von Informationen aus verschiedenen Sensoren (LIDAR, Raddrehgeber)
- Probabilistische Filterung

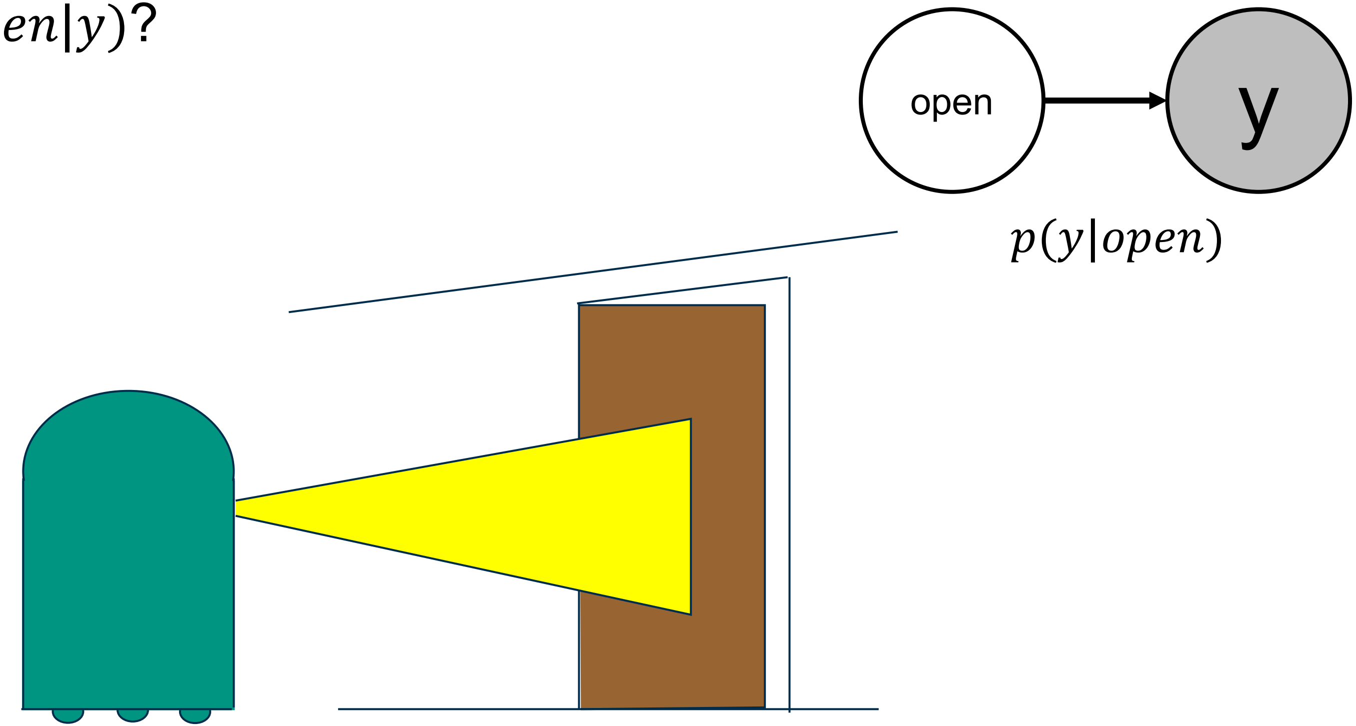
Warum ein probabilistischer Ansatz?



Explizite Darstellung der **Unsicherheit** mit Hilfe der Wahrscheinlichkeitsrechnung

Einfaches Beispiel für eine Zustandsschätzung

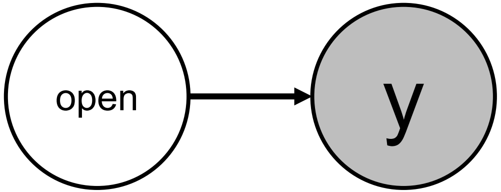
- Angenommen, ein Roboter erhält die Messung y einer Tür
- Was ist $p(open|y)$?



Einfaches Beispiel für eine Zustandsschätzung

Bayes'sches Netzwerk: Likelihood + Prior

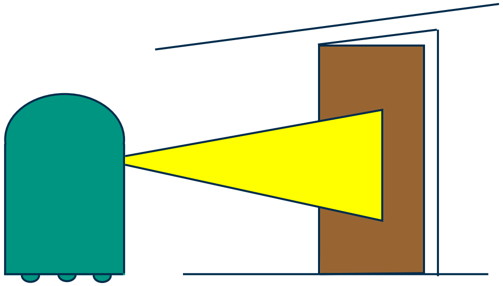
$$p(y|\text{open}) = 0.6, \quad p(y|\neg\text{open}) = 0.3$$
$$p(\text{open}) = 0.5, \quad p(\neg\text{open}) = 0.5$$



Satz von Bayes anwenden:

$$p(\text{open}|y) = \frac{p(y|\text{open})p(\text{open})}{p(y|\text{open})p(\text{open}) + p(y|\neg\text{open})p(\neg\text{open})}$$

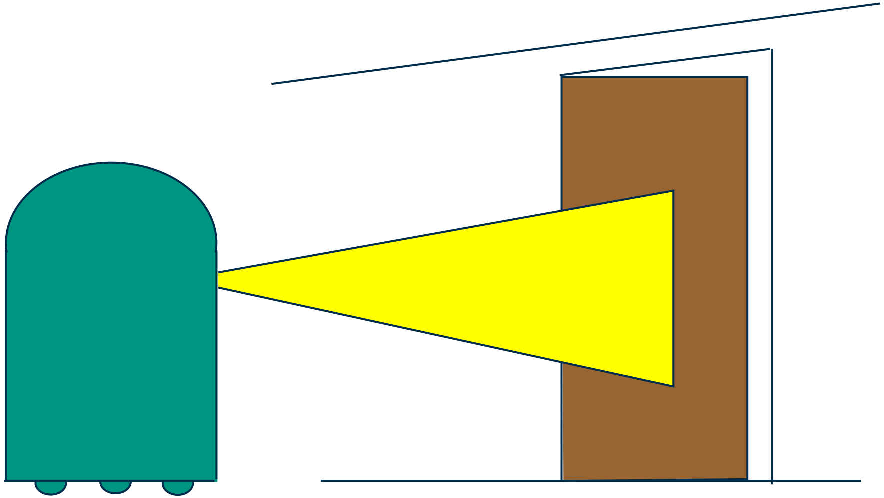
$$p(\text{open}|y) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$



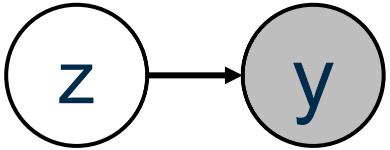
- y erhöht die Wahrscheinlichkeit, dass die Tür offen ist.
- Auch “Konditionierung auf Beobachtung” genannt

Einfaches Beispiel für eine Zustandsschätzung

- Angenommen, unser Roboter erhält weitere Beobachtung y_2
- Wie lautet $p(open|y_1, y_2)$?



Sequentielle Bayes‘sche Aktualisierung



Angenommen, wir haben mehrere Beobachtungen y_1 bis y_n, dann:

$$p(z|y_{1:n}) = p(z|y_n, y_{1:n-1}) = \frac{p(y_n|z, y_{1:n-1})p(z|y_{1:n-1})}{p(y_n|y_{1:n-1})}$$

- Da wir wissen, dass die Beobachtungen y unabhängig gegebenen z sind

$$p(y_n|z, y_{1:n-1}) = p(y_n|z)$$

- können wir den Satz von Bayes rekursiv anwenden (für jede neue Beobachtung!)

Likelihood **Alter Posterior = Prior**

$$p(z|y_{1:n}) = \frac{p(y_n|z)p(z|y_{1:n-1})}{p(y_n|y_{1:n-1})}$$

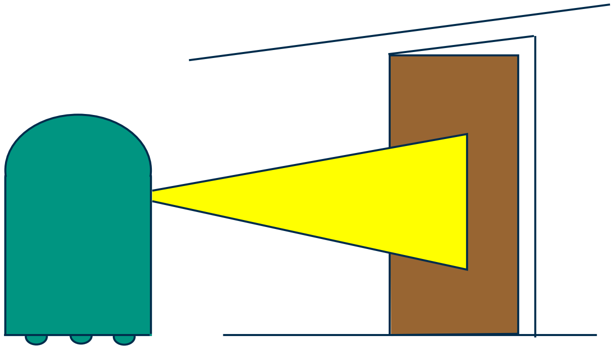
Neuer Posterior **Evidence**

Zurück zur zweiten Beobachtung

Likelihood + Prior (andere Beobachtung als anderer Likelihood)

$$p(y_2|\text{open}) = 0.5, \quad p(y_2|\neg\text{open}) = 0.6$$

$$p(\text{open}|y_1) = \frac{2}{3}, \quad p(\neg\text{open}|y_1) = \frac{1}{3}$$



Satz von Bayes anwenden:

$$p(\text{open}|y_1, y_2) = \frac{p(y_2|\text{open})p(\text{open}|y_1)}{p(y_2|\text{open})p(\text{open}|y_1) + p(y_2|\neg\text{open})p(\neg\text{open}|y_1)}$$

$$p(\text{open}|y_1, y_2) = \frac{0.5 \cdot 2/3}{0.5 \cdot 2/3 + 0.6 \cdot 1/3} = \frac{5}{8} = 0.625$$

- y_2 verringert die Wahrscheinlichkeit, dass die Tür offen ist.

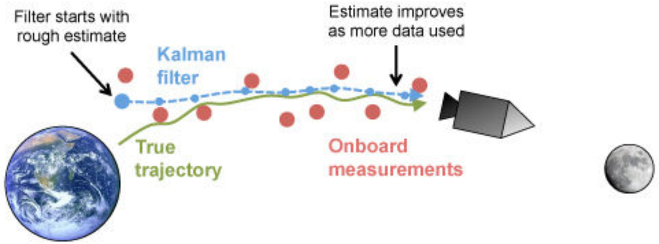
Eine wichtige Anwendung dieser rekursiven Ideen: Filtern

- Berechnung der **Zustands-Verteilung** $b_t(z_t) = p(z_t|y_{0:t})$ (der “Belief State”) gegeben aller vergangenen Beobachtungen

- Wir beginnen mit einer Anfangsverteilung $b_0(z_0)$ (keine Beobachtungen)

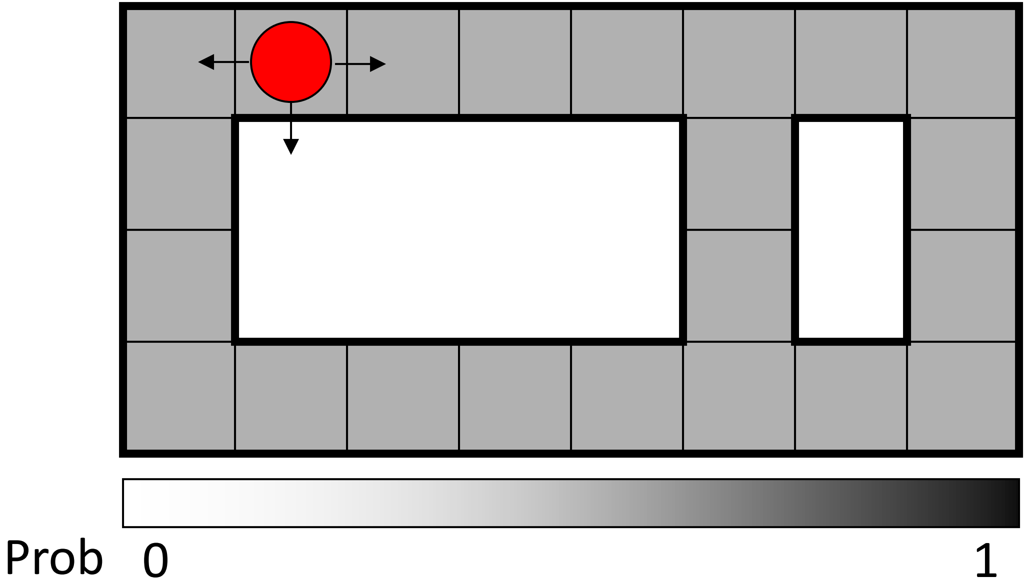
- Wenn ein Zeitschritt vergeht oder wir neue Beobachtungen machen, aktualisieren wir den Belief $b_t(z_t)$

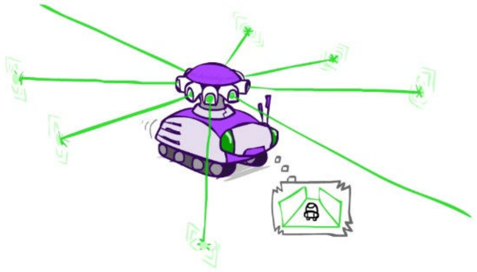
- Für kontinuierliche Systeme: **Kalman-Filter**
  - wurde in den 60er Jahren erfunden und erstmals als Methode zur Flugbahnabschätzung für das Apollo-Programm eingesetzt



Beispiel: Roboter-Lokalisierung

Beispiel von
Michael Pfeiffer





Sensor-Modell:

- Kann messen, in welchen Richtungen eine Wand liegt (4 binäre Werte)
- Max 1 Fehler (Wand falsch detektiert) mit geringer Wahrscheinlichkeit

Bewegungsmodell:

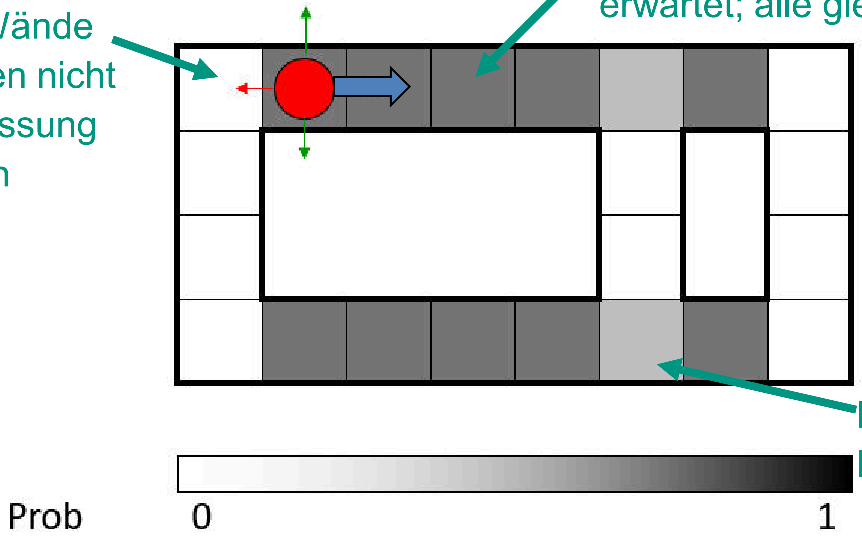
Bewegung + Rauschen (Roboter kann auch stehen bleiben).

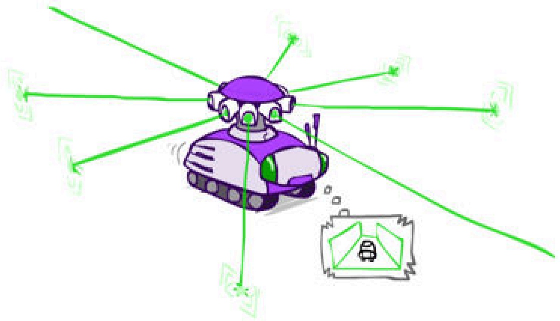
Beispiel: Roboter-Lokalisierung – kennen die Karte - wo ist der Roboter?

Erste Messung: von t=0 nach t=1

An allen dunkelgrauen Positionen hätten wir konsistente Messungen erwartet; alle gleich wahrscheinlich

Zwei Wände stimmen nicht mit Messung überein





Hellgrau: da ein Messfehler möglich ist, sind diese Felder unwahrscheinlicher, aber nicht unmöglich…

Dunkelgrau: Zustand konsistent mit Beobachtung

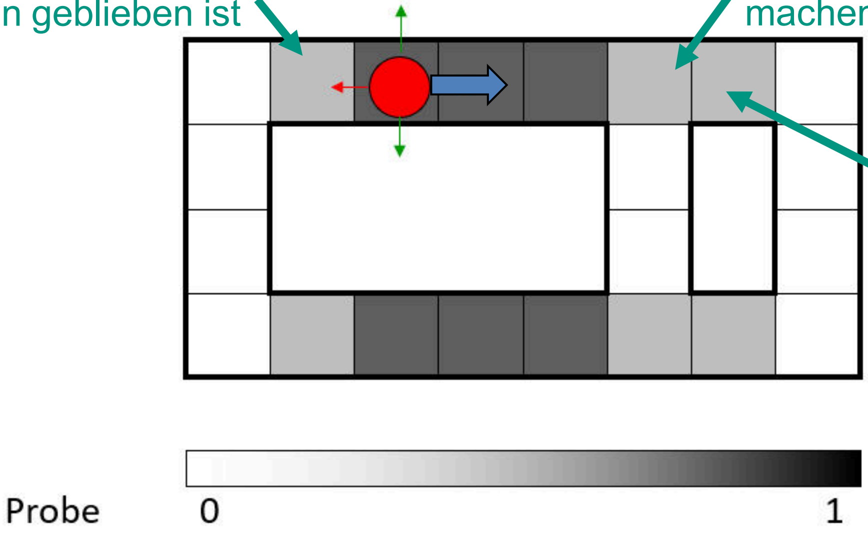
Hellgrau: Zustand weniger wahrscheinlich, da hierfür ein Fehler erforderlich wäre

Weiß: Zustand aufgrund Beobachtung nicht möglich

Beispiel: Roboter-Lokalisierung

Die neue Messung wäre noch konsistent, falls der Roboter fehlerhaft stehen geblieben ist

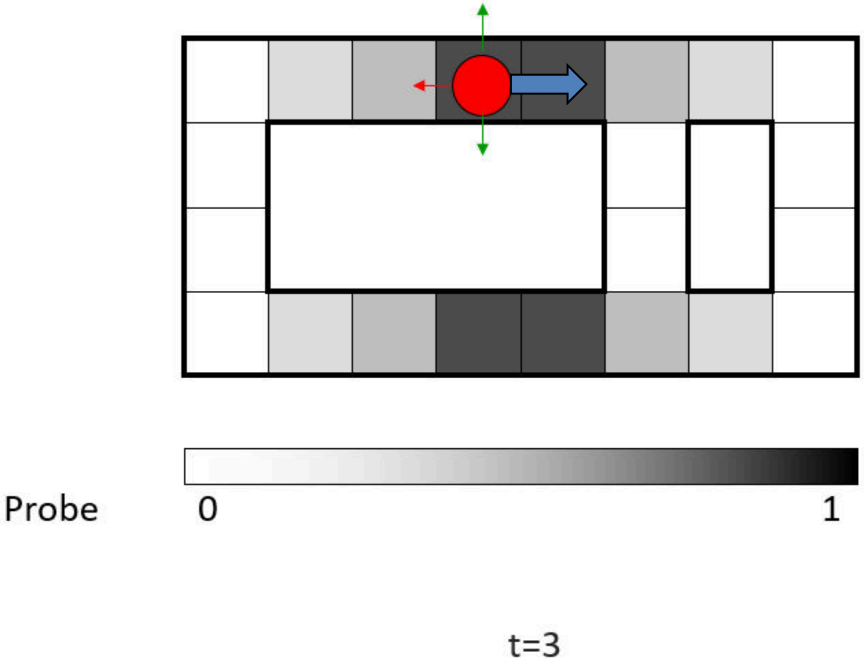
Wäre möglich, weil wir vom Feld links gekommen sein könnten und eventuell jetzt eine fehlerhafte Messung machen

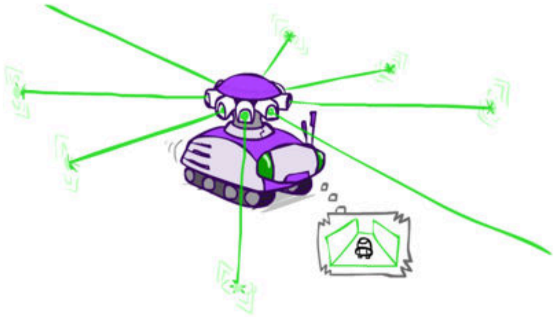


Noch möglich, weil ein Fehler auch in der ersten Messung aufgetreten sein könnte und jetzt korrekt gemessen wurde

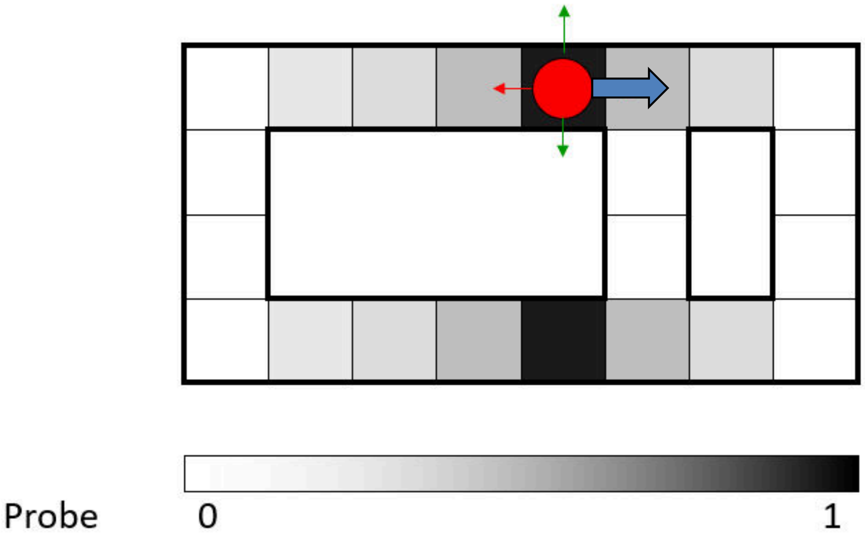
t=2

Beispiel: Roboter-Lokalisierung

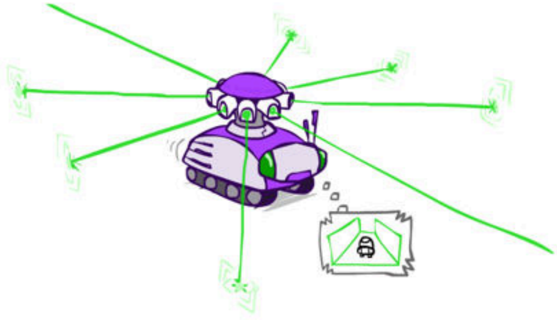




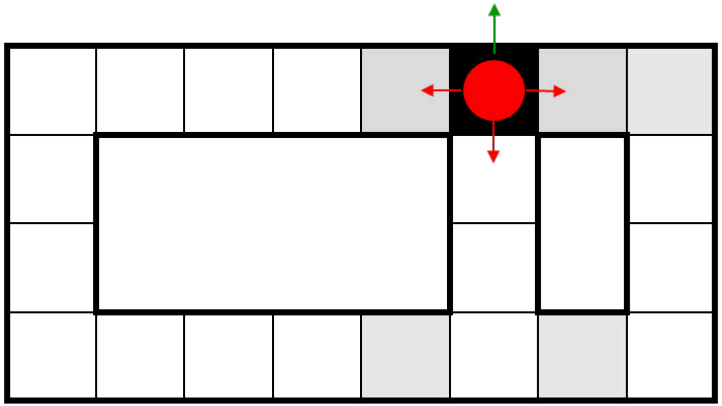
Beispiel: Roboter-Lokalisierung

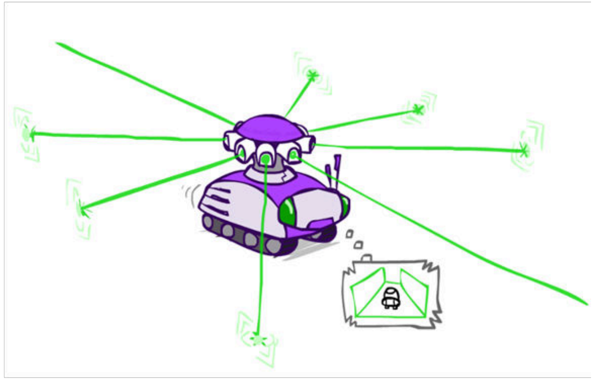


t=4



Beispiel: Roboter-Lokalisierung

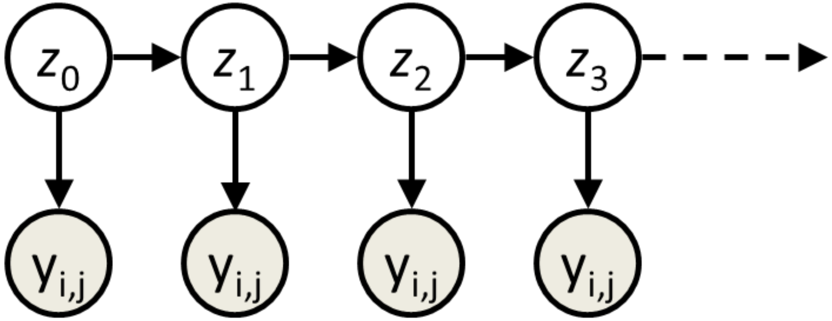


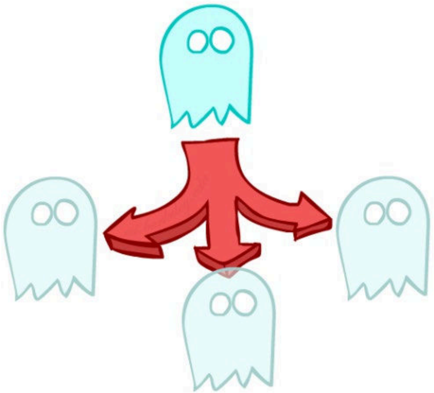


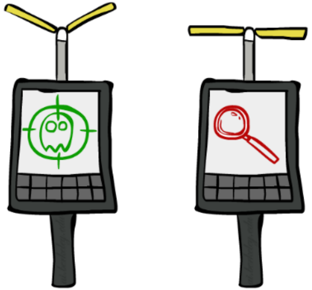
t=5

Beispiel: Geisterjäger HMM

- Anfangszustand: $p(z_0)$ = gleichverteilt
- Bewegungsmodell für jeden Geist: $p(z'|z)$ = bewegt sich normalerweise im Uhrzeigersinn, manchmal aber in eine zufällige Richtung oder bleibt an Ort und Stelle
- Zustandsübergangsmodell: $p(y_{ij}|z)$ = kann ein bestimmtes Feld (ij) daraufhin untersuchen, ob es einen Geist gibt, rot bedeutet nah, grün bedeutet weit weg.







1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

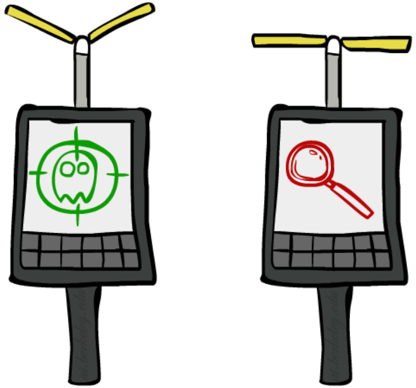
$p(z\ )_0$

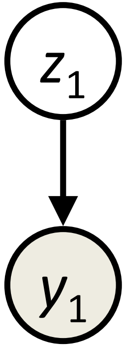
1/6	1/6	1/2
0	1/6	0
0	0	0

$p(z'|z =< 1{,}2 >)$

Schlussfolgerung: Basis-Fälle

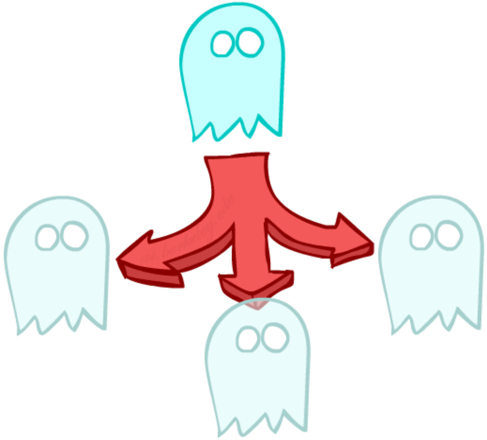
Aktualisierung durch Beobachtung

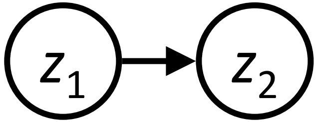




$$p(z_1|y_1) \propto p(y_1|z_1)p(z_1)$$

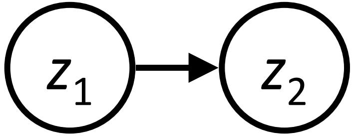
Vorhersage für nächsten Zeitschritt





$$p(z_2) = \sum_{z_1} p(z_2|z_1)p(z_1)$$

Vorhersage: Übergang in den nächsten Zeitschritt



- Angenommen, wir haben eine aktuellen Belief p(z_t | bisherige Erkenntnisse)

$$b_t(z_t) = p(z_t | y_{0:t})$$

- Nachdem Zeitschritt verstrichen, berechne “Prior Belief” für t+1

y_{t+1} noch nicht enthalten

$$b_{t+1}^{-}(z_{t+1}) = p(z_{t+1} | y_{0:t}) = \sum_{z_t} p(z_{t+1} | z_t) p(z_t | y_{0:t}) = \sum_{z_t} p(z_{t+1} | z_t) b(z_t)$$

- Grundidee: Beliefs werden durch die Übergangswahrscheinlichkeit in den nächsten Zeitschritt „geschoben“
  - $b_{t+1}^{-}(z_{t+1})$ ist der prior Belief für den Zeitschritt t+1, d. h. sie enthält keine Beobachtung

Beispiel: Vorhersage in die Zukunft

Bewegungsmodell:
Geister gehen normalerweise im Uhrzeigersinn

- Mit der Zeit "akkumuliert" sich die Unsicherheit

t=1

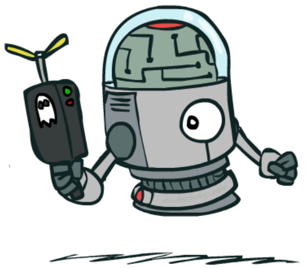
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

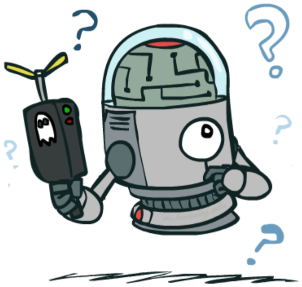
t=2

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

t=5

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01





Wir brauchen also neue Messungen, um die Unsicherheiten einzuschränken!

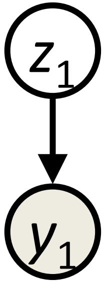
- Angenommen, wir haben einen aktuellen prior Belief $p(z_t \mid y_{<t})$:

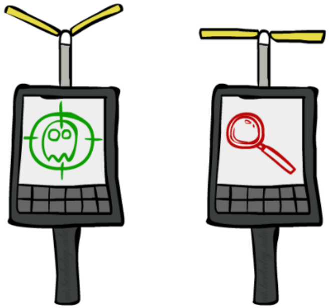
$$b_t^-(z_t) = p(z_t | y_{0:t-1})$$

- Sobald eine neue Beobachtung vorliegt, kann der posterior Belief berechnet werden (Satz von Bayes)

$$b_t(z_t) = p(z_t | y_{0:t}) = \frac{p(z_t, y_t | y_{0:t-1})}{p(y_t | y_{0:t-1})}$$

$$= \frac{p(y_t | z_t) p(z_t | y_{0:t-1})}{p(y_t | y_{0:t-1})} \propto p(y_t | z_t) b_t^-(z_t)$$





- **Grundidee:** prior Beliefs werden nach der Wahrscheinlichkeit der neuen Beobachtung "neu gewichtet".

- Im Gegensatz zur Vorhersage müssen wir renormieren

Belief-Update durch Beobachtung (t=5)

Wenn wir Beobachtungen erhalten, werden Beliefs neu gewichtet, die Unsicherheit "nimmt ab".

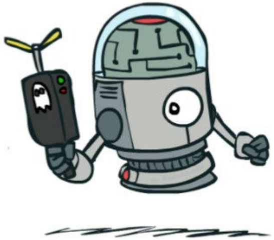
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Vor der Beobachtung

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	$\boxed{<0.01}$	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

Nach der Beobachtung

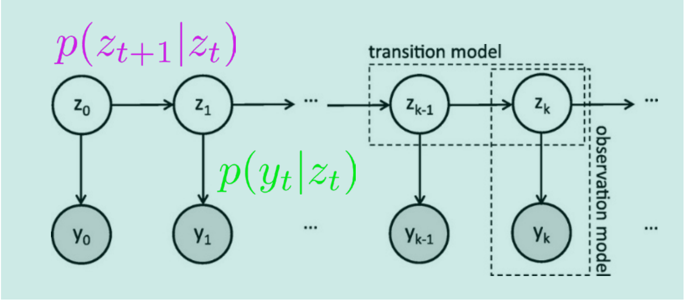
Annahme: beobachten hier – ist der Geist in der Nähe oder nicht?



$$b_t(z_t) \propto p(y_t|z_t)b_t^-(z_t)$$

Der Bayes-Filter (auch bekannt als der Vorwärts Algorithmus)

Berechnen des Belief-States $b_t(z_t) = p(z_t|y_{0:t})$



1. Aktualisierung durch Vorhersage:

$$\underbrace{b^-_{t+1}(z_{t+1})}_{\text{Prior Belief for } t+1} = \sum_{z_t} p(z_{t+1}|z_t) \underbrace{b_t(z_t)}_{\text{Posterior belief for } t}$$

- Übergangsmodell anwenden, vergangenen Zustand ausintegrieren/ marginalisieren

2. Aktualisierung durch Beobachtung:

$$\underbrace{b_{t+1}(z_t)}_{\text{Posterior Belief}} = p(z_{t+1}|y_{0:t+1}) \propto p(y_{t+1}|z_{t+1}) \overbrace{b^-_{t+1}(z_{t+1})}^{\text{Prior Belief for } t+1}$$

- Bayes-Theorem zur Berechnung des Posterior Beliefs
- In der Praxis müssen wir **den Normalisator Z** erst am Ende für den letzten Zeitschritt **berechnen.** Daher arbeiten wir in der Regel mit nicht normalisierten Beliefs (auch **Messages** genannt).

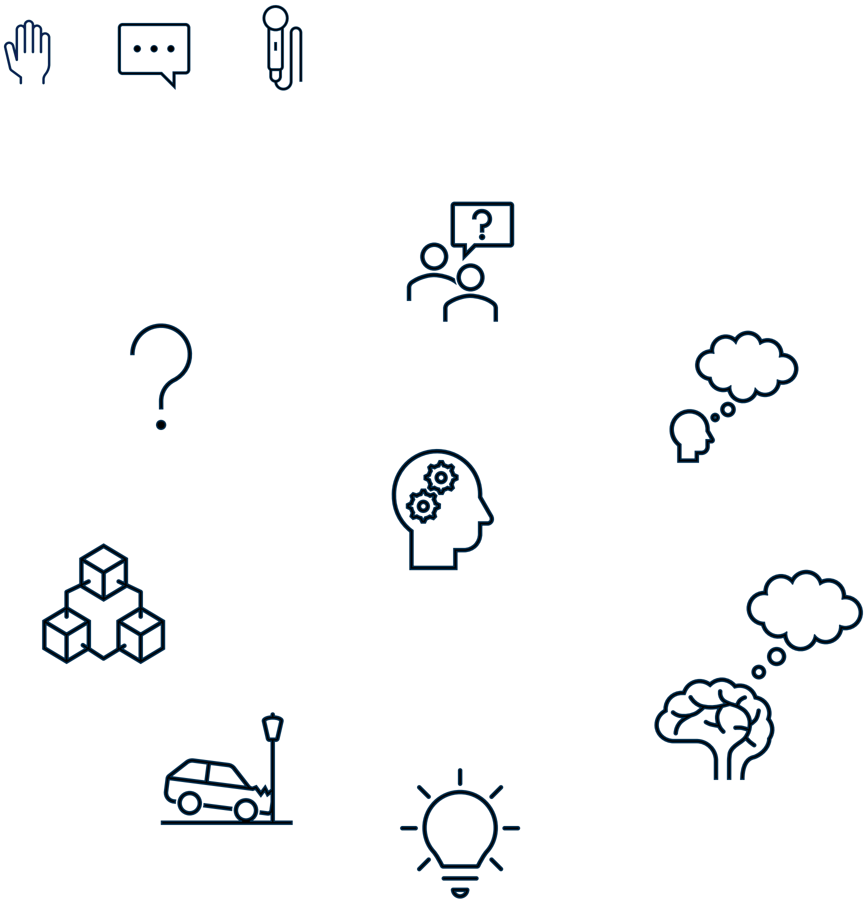
Message weiterleiten

- Es reicht aus, den **unnormalisierten Belief** für Zwischenschritte zu berechnen
- Die Aktualisierung der Vorhersage und die Aktualisierung der Beobachtung können **in einer Gleichung kombiniert werden**
- Unnormierter Belief wird auch als **Message** bezeichnet

Für das Filtern berechnen wir die **Forward-Message**

$$\underbrace{\alpha_{t+1}(z_{t+1})}_{\text{Forward message for } t+1} \propto p(z_{t+1}|y_{0:t+1}) \propto p(y_{t+1}|z_{t+1}) \sum_{z_t} p(z_{t+1}|z_t) \underbrace{\alpha_t(z_t)}_{\text{Forward message for } t}$$

Fragen?



Forward Algorithmus

Init: Initialisierung $\forall z : \alpha_0(z) = p(y_0|z)p(z_0)$

Für k = 1 bis t

Berechnung der Forwardmessage $\alpha_k(z)$ für Zeitschritt k

$$\forall z_k : \alpha_k(z_k) = p(y_k|z_k) \sum_{z_{k-1}} p(z_k|z_{k-1})\alpha_{k-1}(z_{k-1})$$

Normalisiere am Ende, um den Belief zu erhalten:

$$\forall z : b_t(z) = \alpha_t(z) / \sum_{z} \alpha_t(z)$$

Dynamische Programmierung
(siehe MDPs)

- Verwendung von Zwischenergebnissen (Belief) zur Berechnung der Endergebnisse
- Verwendet Messages als unnormalisierten Belief
- Beispiel für Message Passing in Bayes'schen Netzen

Rechenkomplexität:

- Quadratisch in der Anzahl der hidden States
- Für jeden nächsten State muss über alle aktuellen verborgenen Zustände summiert werden

Vermeide numerische Underflows
(zu kleine Zahlen)

- Reskalierung (Normalisierung) nach einigen Zeitschritten

Sonar Pacman mit Beliefs

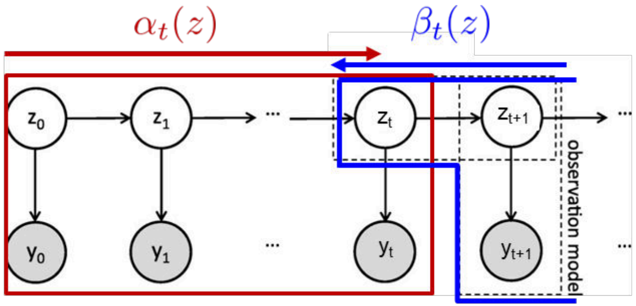
Glättung (Smoothing)

Was ist, wenn wir auch Beobachtungen aus der Zukunft $T > t$ haben?

- Wirkt sich das auf unseren Belief für den Zeitschritt t aus? Ja!
- Das Einbeziehen von Erkenntnissen aus der Zukunft wird als Glättung (Smoothing) bezeichnet.

Wir wollen berechnen: $p(z_t | y_{0:T}) \quad \text{with } t < T$

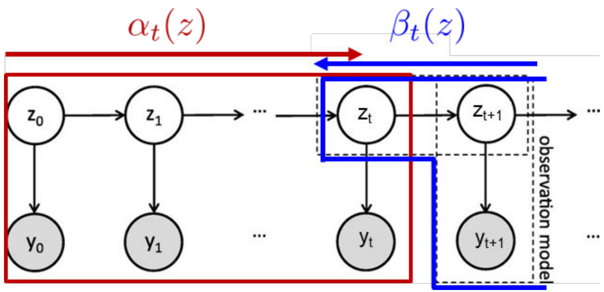
Faktorisiertes Belief-Update:



$$p(z_t | y_{0:T}) \propto \underbrace{p(z_t | y_{0:t})}_{\text{forward message } \alpha_t(z)} \cdot \underbrace{p(y_{t+1:T} | z_t)}_{\text{backward message } \beta_t(z)}$$

- Backward-Message $\beta_t(z)$ "sammelt" Informationen aus der Zukunft

Backward Message



- Anfangs-Message $\beta_T(z_T) = 1$
- Beobachtungs-Update

$$\beta_t^+(z_t) = p(y_{t:T}|z_t) = p(y_t|z_t)p(y_{t+1:T}|z_t) = p(y_t|z_t)\beta_t(z_t)$$

$\beta_t^+(z_t)$: "+" in zeigt an, dass auch Beobachtungen aus dem Zeitschritt t enthalten sind

- Umgedrehtes Vorhersage-Update: (zukünftiger Zeitschritt wird ausintegriert)

$$\beta_t(z_t) = p(y_{t+1:T}|z_t) = \sum_{z_{t+1}} p(y_{t+1:T}, z_{t+1}|z_t) = \sum_{z_{t+1}} p(y_{t+1:T}|z_{t+1})p(z_{t+1}|z_t) = \sum_{z_{t+1}} \beta_{t+1}^+(z)p(z_{t+1}|z_t)$$

- Beide Updates können in einem Schritt berechnet werden:

$$\underbrace{\beta_t(z_t)}_{\text{Backward message for } t} = \sum_{z_{t+1}} p(z_{t+1}|z_t)p(y_t|z_t) \underbrace{\beta_{t+1}(z_{t+1})}_{\text{Backward message for } t+1}$$

Forward-Backward-Algorithmus

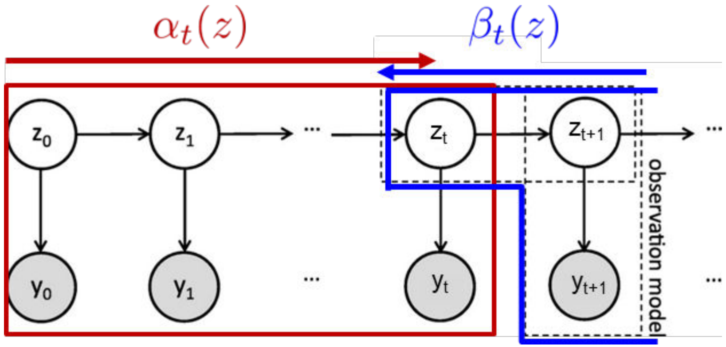
Um $p(z_t|y_{0:T}) \quad \text{mit } t < T$ zu berechnen, müssen wir:

- Berechnung der Forwardmessage

$$\alpha_t(z_t) \propto p(z_t|y_{0:t})$$

- Berechnung der Backwardmessage

$$\beta_t(z_t) = p(y_{t+1:T}|z_t)$$



- Berechnung des geglätteten Beliefs als normalisiertes Produkt

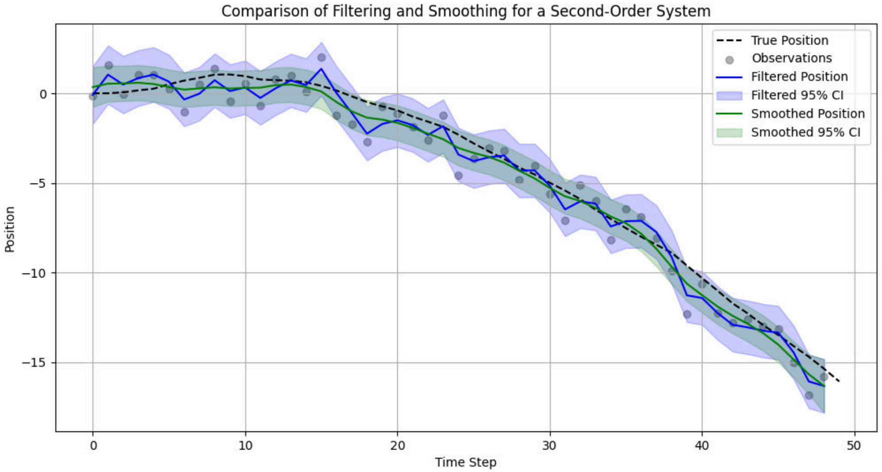
$$b_t(z_t) = p(z_t|y_{0:T}) \propto p(z_t|y_{0:t})p(y_{t+1:T}|z_t) \propto \alpha_t(z_t)\beta_t(z_t)$$

$$\Rightarrow \quad b_t(z_t) = \frac{\alpha_t(z_t)\beta_t(z_t)}{\sum_{z_t} \alpha_t(z_t)\beta_t(z_t)}$$

Smoothing vs. Filtering

Visualisierung für ein kontinuierliches System

Dieselben Gleichungen, nur mit Gaußverteilungen (Kalman-Filter/Smoother)



- „Glätteres" Ergebnis als Filter, näher dran am echten Zustand
- Smoother hat weniger Unsicherheit (da auch zukünftige Informationen verwendet werden)

Was wir uns heute nicht angesehen haben...

- **HMM-Dekodierung**
  - Welches ist die wahrscheinlichste Abfolge $z_{0:t}$ von Zuständen bei gegebenen Beobachtungen $y_{0:t}$?
  - Berechnet mit dem Viterbi-Algorithmus, ebenfalls ein dynamisches Programm

- **Lernen mit HMM**
  - Einfach, wenn z_t auch beobachtet wird: Einfaches Zählen (alle Übergänge und Beobachtungsemissionen)
  - Normalerweise ist z_t unbeobachtet.
  - Kann mit dem Baum-Welch-Algorithmus gelernt werden
    - **Expectation-Maximization**
    - Expectation-Schritt: Berechnung der Anzahl der erwarteten Zustandsbesuche und Übergänge mit Hilfe des Forward-Backward-Algorithmus
    - Maximization-Schritt: Schätzung der HMM-Parameter auf der Grundlage der erwarteten Übergänge

- **Kalman-Filterung:**
  - Bayes-Filter für lineare Systeme mit Gaußschem Rauschen

HMM Zusammenfassung

Hidden-Markov-Modelle und der Bayes-Filter bilden die Grundlage für viele andere Methoden:

- Kalman-Filter
- Partikelfilter
- Partial-Observable Markov-Decision Processes (POMDPs)

Berechnung eines Beliefs (einer Verteilung) über den Zustand zum Zeitschritt t

- **Vorhersage:** Belief um einen Zeitschritt nach vorne verschieben (Marginalisierung)
- **Beobachtung:** Neue Informationen einbeziehen (Bayes-Regel)
- **Glättung:** Auch Informationen aus der Zukunft einbeziehen

Forward-Backward-Algorithmus:

- Effizienter Weg zur Berechnung des Beliefs unter Verwendung der Markov-Faktorisierungsannahmen
- Berechnet Forward und Backward-Messages, um den Belief an dem Zeitschritt t zu berechnen

Fragen zum Selbsttest

Was Sie jetzt wissen sollten:

- Grundregeln der Wahrscheinlichkeitsrechnung
- Wie man die Bayes-Regel verwendet und wie man sie rekursiv verwendet
- Wie der Belief State definiert ist
- Welche Aktualisierungen sind erforderlich, um den Belief State in einem Bayes-Filter zu erhalten?
- Die Aktualisierungsgleichungen für den Bayes-Filter

Fragen?