

Tutoriums-Paper zu Hidden Markov Models

Mario Mohr

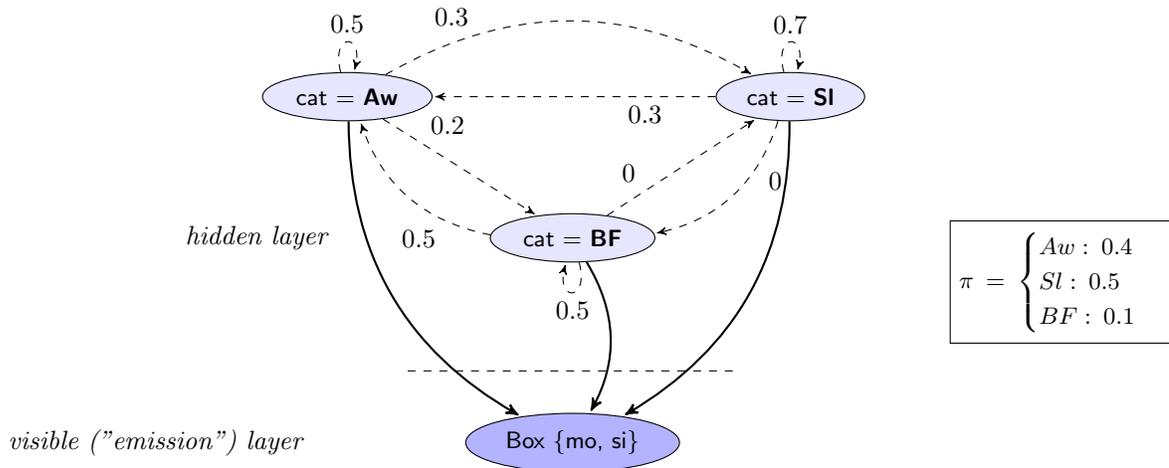
February 1, 2015

Contents

| | | |
|----------|---|----------|
| 1 | Das Modell | 1 |
| 2 | Der Forward-Algorithmus | 2 |
| 2.1 | Wahrscheinlichkeiten von Beobachtungsketten | 4 |
| 2.2 | Filtering | 5 |
| 3 | Der Backward-Algorithmus | 5 |
| 4 | Der Forward-Backward Algorithmus | 7 |
| 4.1 | Smoothing | 8 |
| 5 | Der Viterbi-Algorithmus (Decoding) | 8 |

1 Das Modell

Wir werden in diesem Paper das Beispielmodell aus dem Tutorium verwenden: wir stecken, als Abwandlung von Schrödingers berühmtem Gedankenexperiment, eine Katze in eine schalldichte Schachtel. Dabei betrachten wir drei¹ mögliche Zustände der Katze: den des gemäßigten Wachseins ("Awake" oder *Aw*), den des Schlafes ("Sleeping" oder *Sl*) und den der rasenden Wut ("Bloody Furious" oder *BF*). Die Übergangswahrscheinlichkeiten sowie die Startverteilung π sind dabei die folgenden:



¹In fact, the mere act of opening the box will determine the state of the cat, although in this case there were three determinate states the cat could be in: these being Alive, Dead, and Bloody Furious." - Terry Pratchett, *Lords and Ladies*

Die von uns beobachtete Schachtel hat zu einem beliebigen Zeitpunkt zwei mögliche Zustände: rappelnd ("moving" oder *mo*) und still ("silent" oder *si*). Die Wahrscheinlichkeiten dieser Zustände hängen dabei folgendermaßen vom Zustand der Katze ab:

$$P(\text{Box}|\text{cat}) = \begin{matrix} & \text{cat} & \text{Box} = \mathbf{moving} & \text{Box} = \mathbf{silent} \\ \begin{pmatrix} \mathbf{Awake} \\ \mathbf{Sleeping} \\ \mathbf{BF} \end{pmatrix} & & & \end{pmatrix} \begin{matrix} 0.5 & 0.5 \\ 0.05 & 0.95 \\ 0.9 & 0.1 \end{matrix}$$

Die Grafik des Modells und die Wahrscheinlichkeitstabelle der Schachtel befinden sich noch einmal groß auf der Letzen Seite dieses Papers.

2 Der Forward-Algorithmus

Der erste HMM-Algorithmus, mit dem wir uns beschäftigen, ist der **Forward-Algorithmus**. Der Forward-Algorithmus geht, wie die anderen in diesem Paper behandelten Algorithmen, davon aus, dass wir eine Reihe von Beobachtungen über die Emissions-Variablen unseres Modells (in unserem Fall: über die Zustände der Schachtel) gemacht haben und bestimmte Informationen darüber oder über das dahinterliegende Systemverhalten herausfinden wollen.

Angenommen, wir befinden uns in der folgenden Situation: bei einer dreiminütigen Testreihe haben wir beobachtet, dass die Schachtel in der ersten Minute still dalag ($O_1 = si$), in der zweiten Minute los- und in der dritten Minute weiterrappelte ($O_2 = mo, O_3 = mo$, insgesamt also: $O = \mathbf{si, mo, mo}$).

Zwei mögliche Fragen, die wir uns stellen können, sind:

- **in welchem Zustand befindet sich die Katze jetzt gerade?** - Ist es sicher, die Schachtel zu öffnen, also: wie wahrscheinlich ist es, dass ich mich nach genau dieser Beobachtungsfolge einer wütenden Katze gegenübersehe? ($P(\text{cat}_{t_3} = BF | O = si, mo, mo)$)
- **wie häufig kommt diese Beobachtungsfolge überhaupt vor?** - Lohnt es sich, beim monatlichen Katzenroulette der Felidologen auf sie zu wetten? ($P(O = si, mo, mo)$)

Beide Fragen, also die nach dem Systemzustand am Ende eines Zeitraumes und die nach der allgemeinen Wahrscheinlichkeit einer Beobachtungsfolge, lassen sich durch den Forward-Algorithmus beantworten.

Das Prinzip beim Forward-Algorithmus ist das folgende: wir gehen die beobachteten Zeiträume des Systems nacheinander durch und notieren uns dabei für jeden Zustand, wie wahrscheinlich es ist, dass

- erstens unser System die bis dahin beobachteten Emissionen in genau dieser Reihenfolge erzeugt (z.B.: "wie wahrscheinlich ist es, dass Schachtel am Anfang des Experimentes still ist und dann rappelt?", also $P(O_1 = si, O_2 = mo)$) und sich
- zweitens unser System im aktuellen Zeitraum im jeweiligen Zustand befindet (z.B.: "wie wahrscheinlich ist es, dass die Katze während der zweiten Hälfte der oben beschriebenen Beobachtungen schläft?", also $P(\text{cat}_{t_2} = Sl | O_1 = si, O_2 = mo)$).

Dieser Wert wird für den Zustand X im Zeitraum t_n als $\alpha_{t_n}(X)$ notiert (q_t ist der Systemzustand zum Zeitpunkt t):

$$\alpha_t(X) = P(O_1, O_2, \dots, O_t, q_t = X | \lambda)$$

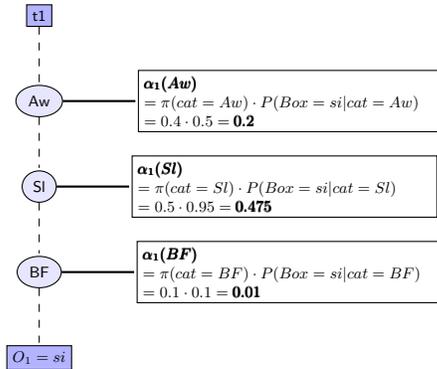
(also: $\alpha_2(Sl) = P(O_1 = si, O_2 = mo, \text{cat}_{t_2} = Sl)$)

Die Berechnung der α -Werte für den ersten Zeitraum ist einfach: wir multiplizieren die Startwahrscheinlichkeit des jeweiligen Zustandes X mit der Wahrscheinlichkeit, dass er die "korrekte" Emission erzeugt, also die, die wir als erstes beobachtet haben:

$$\alpha_1(X) = \pi(X) \cdot P(O_1|q_1 = X)$$

(z.B.: $\alpha_1(BF) = \pi(cat = BF) \cdot P(Box = si|cat = BF)$)

Die α -Werte für unsere erste Teilbeobachtung $O_1 = si$ sehen also folgendermaßen aus:



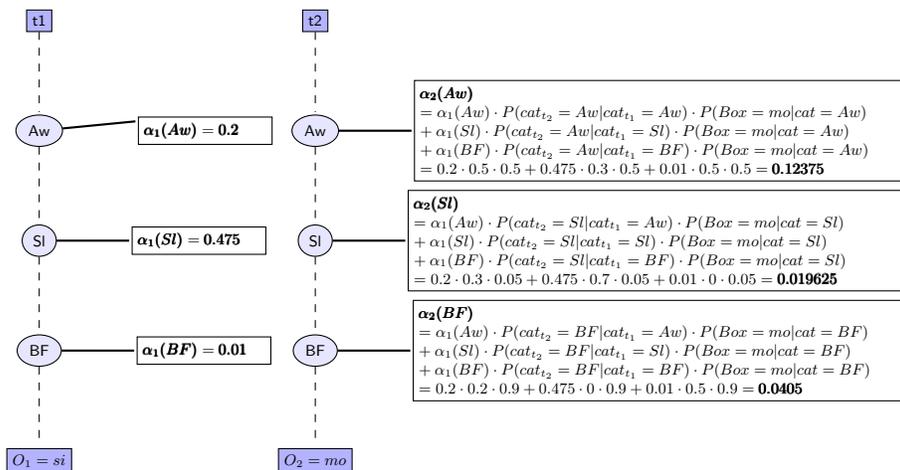
Die Berechnung der darauffolgenden α -Werte ist etwas aufwändiger - immerhin müssen wir abgesehen von der Wahrscheinlichkeit des jeweiligen Zustands nicht nur die Wahrscheinlichkeit der aktuellen, sondern auch die aller vorherigen Beobachtungen berücksichtigen (zur Erinnerung: der α_t -Wert eines Zustandes bezeichnet die Wahrscheinlichkeit, dass dieser Zustand zum Zeitpunkt t herrscht *und alle bisherigen Emissionen mit unseren Beobachtungen übereinstimmen*).

Angenehmerweise bietet uns gerade diese Eigenschaft des α -Wertes eine einfache Lösung des Problems: bei der Berechnung jedes Wertes berücksichtigen wir nur die α -Werte der vorherigen Iteration, die ja gerade die Informationen aller vorangegangenen Zeitpunkte enthalten. Praktisch sieht das folgendermaßen aus: wir multiplizieren den α -Wert jedes Vorgängers mit dessen Übergangswahrscheinlichkeit zum aktuellen Zustand und der Wahrscheinlichkeit, dass der aktuelle Zustand die "richtige" Emission erzeugt (also die momentan beobachtete). Diese Werte summieren wir, und voilà - der neue Alphawert ist berechnet.

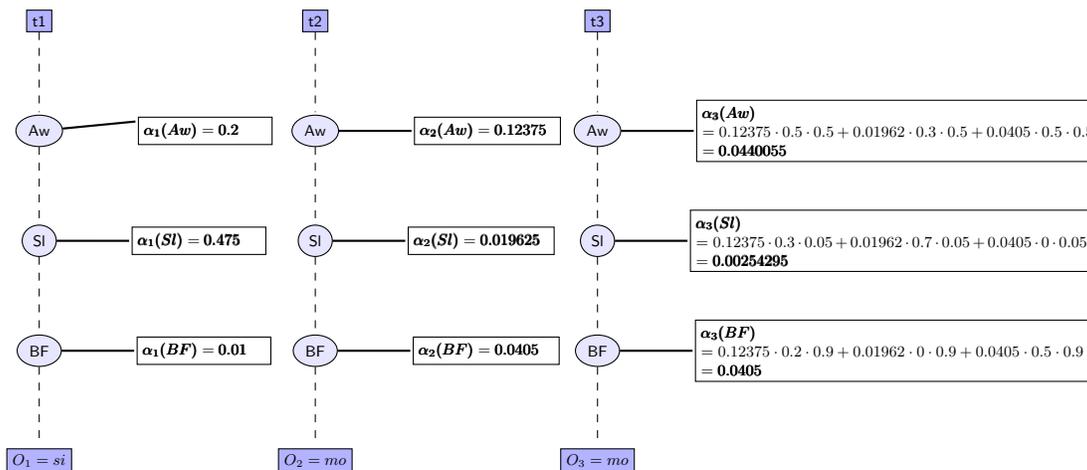
Formal sieht das für N mögliche Systemzustände folgendermaßen aus (keine Sorge, bei der nächsten Grafik wird es etwas anschaulicher):

$$\alpha_{t+1}(Y) = \sum_{X=1}^N [\alpha_t(X) \cdot P(q_{t+1} = Y|q_t = X) \cdot P(O_{t+1}|q_t = Y)]$$

Dadurch kommen wir auf folgende α -Werte für das zweite ...



... und das dritte Intervall:



Damit haben wir für unsere Beobachtung $O = si, mo, mo$ alle α -Werte bestimmt und sind der Beantwortung unserer beiden Fragen ("Sollte ich auf si, mo, mo wetten?" und "Mit welcher Wahrscheinlichkeit ist die Katze jetzt sauer?") sehr nah.

2.1 Wahrscheinlichkeiten von Beobachtungsketten

Wir haben bei der Berechnung der α -Werte in jeder Iteration die Vorgängerzustände aufsummiert; dabei haben wir nichts anderes getan als die Unterscheidung zwischen den möglichen Vorgängern aus unserer Rechnung zu entfernen. $\alpha_3(SI) \approx 0.0025$ bedeutet: die Wahrscheinlichkeit, dass unsere Katze die Beobachtungskette si, mo, mo erzeugt und dabei im letzten Zeitraum geschlafen hat, ist 0.0025 - egal, in welchen beiden Zuständen sie sich in t_1 und t_2 befand.

Wenn wir die α -Werte des letzten Zeitraumes (für uns: t_3) aufsummieren, entfernen wir auch noch die letzte Zustandsunterscheidung aus der Rechnung - $\alpha_3(Aw) + \alpha_3(SI) + \alpha_3(BF) \approx 0.087$ bedeutet also: die Wahrscheinlichkeit, dass unsere Katze die Beobachtungskette si, mo, mo erzeugt, ist 0.087 - egal, in welchen drei Zuständen sie sich in t_1 , t_2 und t_3 befand. Diese Summe ist also genau die Wahrscheinlichkeit, nach der wir in unserer ersten Frage suchen:

"Wie wahrscheinlich ist es, dass die Katze genau die Beobachtung $O = si, mo, mo$ erzeugt?"

$$P(O = si, mo, mo) = \sum_{X \in \{Aw, Sl, BF\}} \alpha_3(X) \approx 0.087$$

Und allgemein ausgedrückt (wobei T der letzte beobachtete Zeitraum ist):

$$P(O) = \sum_{X=1}^N \alpha_T(X)$$

2.2 Filtering

Die andere Frage, die wir uns am Anfang dieses Kapitels gestellt haben, ist: kann ich gefahrlos den Deckel der Schachtel öffnen (also: wie hoch ist die Wahrscheinlichkeit dafür, dass sich die Katze am Ende meiner Messreihe in einem wütenden Zustand befindet - $P(cat_{t_3} = BF | O = si, mo, mo)$)?

Die Antwort darauf steckt erneut in den α -Werten, die wir gerade berechnet haben. Der Wert $\alpha_3(BF) = 0.0405$ bedeutet: die Wahrscheinlichkeit, dass die Katze am Ende des Experimentes wütend ist *und unsere Observationskette si, mo, mo erzeugt* ist 0.0405. Wenn wir also den zweiten Teil aus diesem Wert herausrechnen können, haben wir unsere Antwort gefunden.

Angenehmerweise ist das sehr einfach zu erreichen: wir müssen nur die Wahrscheinlichkeit unserer Observationskette herausdividieren. Genau diese Wahrscheinlichkeit haben wir gerade eben ausgerechnet ($P(O = si, mo, mo) \approx 0.087$). Formal ausgedrückt:

$$P(cat_{t_3} = BF | O = si, mo, mo) = \frac{\alpha_3(BF)}{P(O = si, mo, mo)} \approx \frac{0.0405}{0.087} \approx 0.4655$$

Die Wahrscheinlichkeit, eine tobende Katze anzutreffen, ist also mit 46,5% ziemlich hoch.

Allgemein ausgedrückt (mit der Berechnung für $P(O)$, die wir in 2.1 kennen gelernt haben und wieder T als dem letzten beobachteten Zeitraum):

$$P(q_T = X) = \frac{\alpha_T(X)}{\sum_{Y=1}^N \alpha_T(Y)}$$

Anstatt uns explizit für die Wahrscheinlichkeit eines möglichen Endzustandes zu interessieren (in diesem Fall aus Angst um unsere Hand), können wir die Frage nach dem **wahrscheinlichsten Endzustand** stellen. Mit unseren Mitteln ist der sehr einfach zu finden: wir berechnen die Wahrscheinlichkeit für jedes Endzustandes und bestimmen denjenigen mit dem höchsten Wert. Dieser Vorgang wird als **Filtering** bezeichnet.

3 Der Backward-Algorithmus

Wir haben gerade eine Möglichkeit kennen gelernt, die Wahrscheinlichkeit zu ermitteln, mit der ein Zustand, bei einer bestimmten Observationsreihe als Endzustand auftaucht. Möglicherweise gelangen wir aber in eine Situation, in der uns der wahrscheinlichste Zustand für einen Zeitpunkt irgendwo in der Mitte der Messreihe interessiert. Wie ging es unserer Katze z.B. in der Mitte des Experimentes (also bei t_2)?

Der Forward-Algorithmus alleine bietet uns dafür keine Lösung. Wir könnten ihn natürlich einfach nach der zweiten Iteration abbrechen; allerdings würden wir damit Informationen (in diesem Fall: das Verhalten der Schachtel im dritten Intervall) wegwerfen. Hier kommt der **Backwards-Algorithmus** ins Spiel.

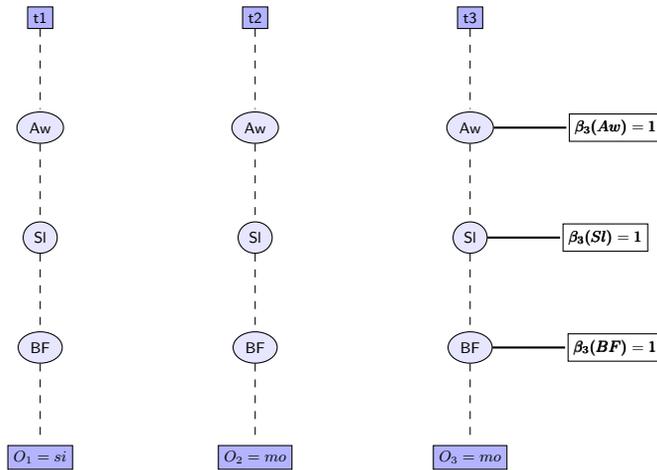
Der Backwards-Algorithmus führt einen β -Wert für jeden Knoten ein, der das Komplement zum α -Wert darstellt: der α -Wert enthält die Wahrscheinlichkeit der bisherigen Beobachtungen (einschließlich der Momentanen), während der β -Wert eines Zustandes X zum Zeitpunkt t ausdrückt, wie wahrscheinlich es ist, dass das System von X aus bis zum Experimentabschluss die "richtigen" Emissionen erzeugt. Formal ausgedrückt:

$$\beta_t(X) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = X)$$

Wie der Name suggeriert, werden beim Backwards-Algorithmus die β -Werte vom Schluss des Experimentes aus zum Start berechnet. Die Initialisierung ist dabei ausgesprochen einfach (erneut T als der letzte beobachtete Zeitraum):

$$\beta_T(i) = 1$$

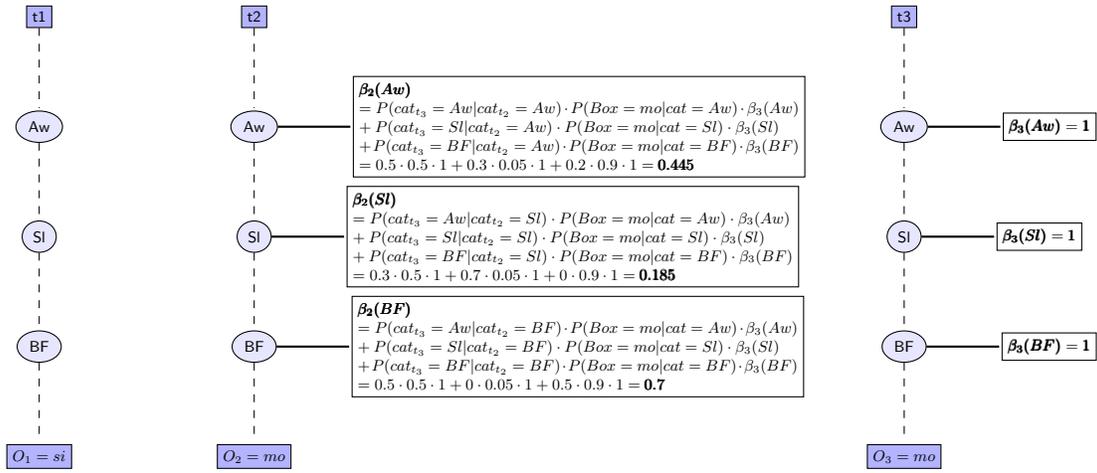
Die Erklärung für den Wert 1 ist recht intuitiv: der β -Wert drückt die Wahrscheinlichkeit der "Richtigkeit" der folgenden Emissionen aus. Wenn es keine nachfolgenden Emissionen mehr gibt - und das ist beim Zeitpunkt T der Fall - sind sie trivialerweise erfüllt; die Wahrscheinlichkeit für "falsche" Emissionen ist also 0.



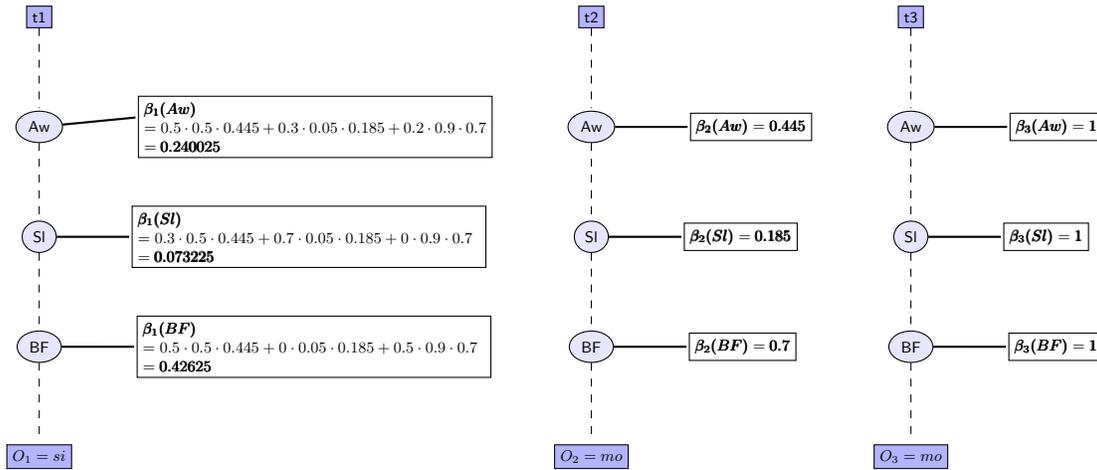
Hinter der Formel für die anschließenden Iterationen steckt dasselbe Prinzip wie beim Forward-Algorithmus. Deswegen nur eine kurze Erläuterung: die Übergangswahrscheinlichkeit zu jedem möglichen Nachfolgerzustand wird mit dessen β -Wert und seiner Wahrscheinlichkeit, die "richtige" Emission zu erzeugen, multipliziert. Formal:

$$\beta_t(X) = \sum_{Y=1}^N P(cat_{t+1} = Y | cat_t = X) \cdot P(O_{t+1} | Y) \cdot \beta_{t+1}(Y)$$

Also für t_2 :



... und für t_1 :



Die so berechneten β -Werte alleine bringen uns unserem Ziel - der Berechnung der Wahrscheinlichkeit von Zuständen zu irgendeinem Zeitpunkt - noch nicht näher.

4 Der Forward-Backward Algorithmus

Wenn wir allerdings die α - und β -Werte der beiden Algorithmen kombinieren (also miteinander multiplizieren), erhalten wir die Wahrscheinlichkeit $\alpha_t(X) \cdot \beta_t(X)$, dass das System die "richtige" Emissionsfolge erzeugt und dabei zum Zeitpunkt t durch den Zustand X geht. Formal:

$$P(O, q_t = X) = \alpha_t(X) \cdot \beta_t(X)$$

Auch hier, wie beim Filtering, sind wir nur an der Wahrscheinlichkeit des Zustandes interessiert. Also müssen wir erneut durch die Wahrscheinlichkeit der Beobachtungsfolge teilen. Wir können dazu hier eine ähnliche Methode wie beim Forward-Algorithmus verwenden. Dort haben wir über den letzten Zeitraum T aufsummiert; hier können wir z.B. über die Werte des Zeitpunktes summieren, für den wir gerade die Zustandswahrscheinlichkeiten ermitteln wollen (wir verwandeln "die Wahrscheinlichkeit, dass die richtige Emissionsfolge erzeugt wird, zum Zeitpunkt t Zustand X und zu allen anderen

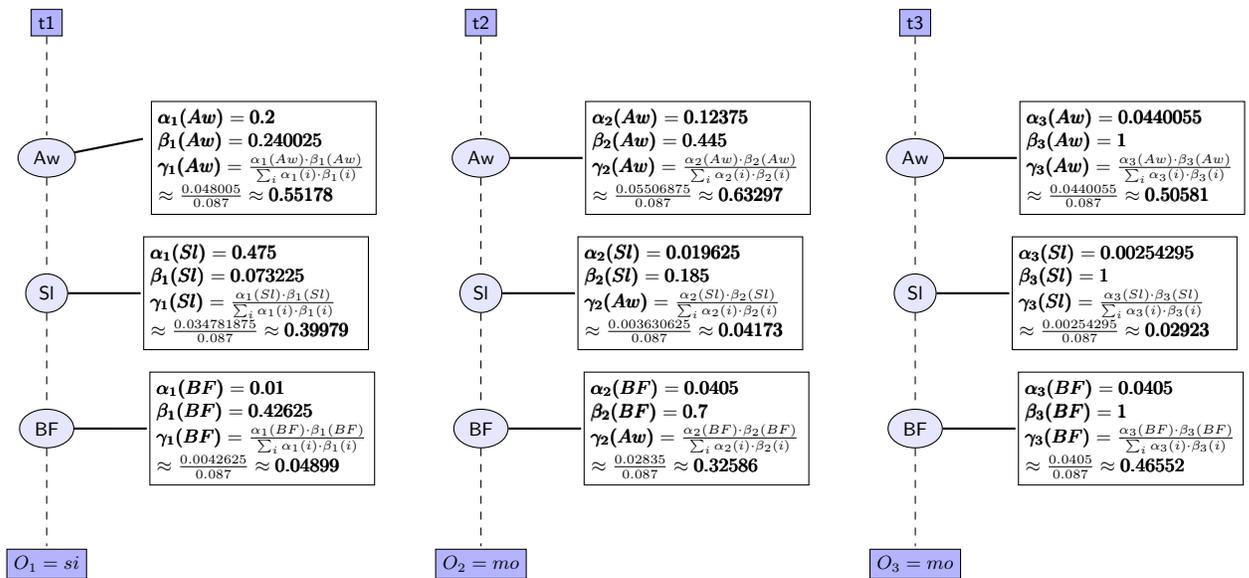
Zeitpunkten beliebige Zustände durchlaufen werden" in "die Wahrscheinlichkeit, dass die richtige Emissionsfolge erzeugt wird und zu allen Zeitpunkten beliebige Zustände durchlaufen werden", also gerade unser gesuchtes $P(O)$). Formal (für einen beliebigen Zeitpunkt t):

$$P(O) = \sum_{X=1}^N \alpha_t(X) \cdot \beta_t(X)$$

Die Wahrscheinlichkeit eines Zustandes X zu einem beliebigen Zeitpunkt t (üblicherweise mit γ bezeichnet) lässt sich also folgendermaßen berechnen:

$$\gamma_t(X) = \frac{\alpha_t(X) \cdot \beta_t(X)}{\sum_{Y=1}^N \alpha_t(Y) \cdot \beta_t(Y)}$$

Die γ -Werte für unsere Observationskette sehen also folgendermaßen aus:



4.1 Smoothing

Wie beim Forward-Algorithmus können wir auch beim Forward-Backward-Algorithmus nach dem wahrscheinlichsten Zustand zu einem Zeitpunkt suchen. Diese Auswahl des Zustands mit dem höchsten γ -Wert für einen Zeitpunkt nennt man **Smoothing**.

Beim Smoothing kann ein großes Problem auftauchen: dadurch, dass die Zeitintervalle isoliert betrachtet werden, kann es bei der Verwendung des Smoothings zum Finden der wahrscheinlichsten Zustandsfolge zu einer Beobachtungskette (durch das einfache Aneinanderhängen der Zustände mit den jeweils besten γ -Werten) zu unwahrscheinlichen oder sogar unmöglichen Zustandsfolgen als Ergebnis kommen (in unserem Beispiel könnte es zum Aufeinanderfolgen von BF und Sl kommen, was nach den Übergangswahrscheinlichkeiten unseres Modells unmöglich ist).

Hier kommt der **Viterbi-Algorithmus** ins Spiel.

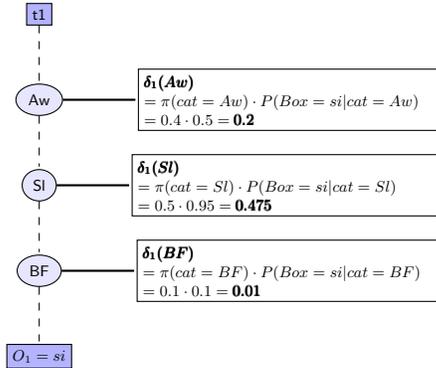
5 Der Viterbi-Algorithmus (Decoding)

Wenn wir uns für die **wahrscheinlichste Zustandsfolge zu einer Beobachtungskette** interessieren ("Wie ging es der Katze über die Dauer des Experimentes, wenn wir $O = si, mo, mo$ beobachtet

haben?“), verwenden wir üblicherweise den **Viterbi-Algorithmus**. Der Grundgedanke dabei ist ähnlich wie beim Forward-Algorithmus: wir berechnen iterativ Wahrscheinlichkeitswerte (in diesem Fall: δ) für alle Zustände zu allen Zeitpunkten. Die Berechnung der Anfangswerte funktioniert dabei genau wie beim Forward-Algorithmus - wir multiplizieren die Startwahrscheinlichkeit des jeweiligen Zustands mit der Wahrscheinlichkeit, dass er die "richtige" erste Emission erzeugt:

$$\delta_1(X) = \pi(X) \cdot P(O_1|q_1 = X)$$

Die δ_1 Werte für unsere Beobachtungskette sehen also folgendermaßen aus:



Die Berechnung der höheren Iterationen ist etwas interessanter. Wir nutzen wieder die indirekte Weitergabe von Informationen durch unsere δ -Werte in die nächste Iteration aus; im Gegensatz zum Forward-Algorithmus, bei dem wir über die Vorgänger summiert haben (weil uns der genaue "Herkunftspfad" des aktuellen Knotens nicht interessiert hat), berechnen wir hier die Werte für alle möglichen Vorgängerknoten separat und wählen den besten aus.

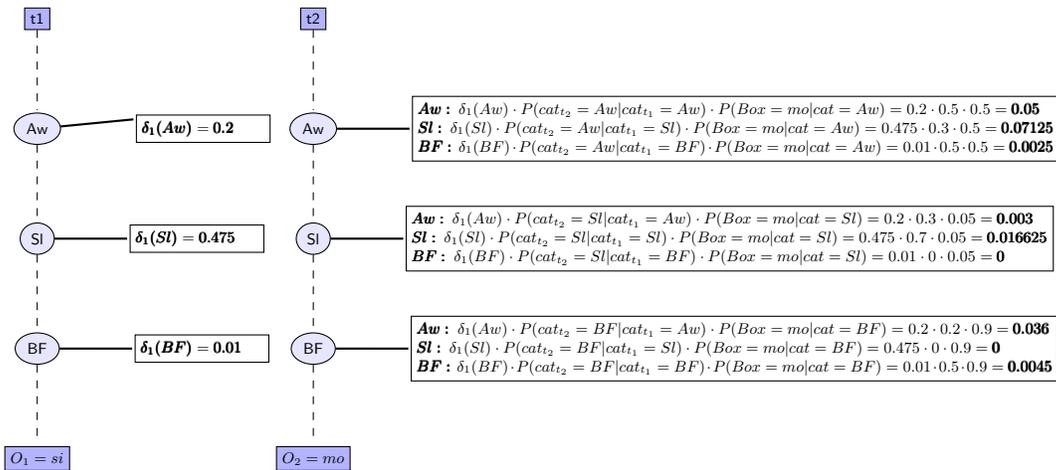
Aus der Formel für den α -Wert

$$(\alpha_{t+1}(Y) = \sum_{X=1}^N [\alpha_t(X) \cdot P(q_{t+1} = Y|q_t = X) \cdot P(O_{t+1}|q_t = Y)])$$

leiten wir durch das Ersetzen des Summenzeichens mit dem Maximum-Operator (und α durch δ , natürlich) die Formel für den δ -Wert ab:

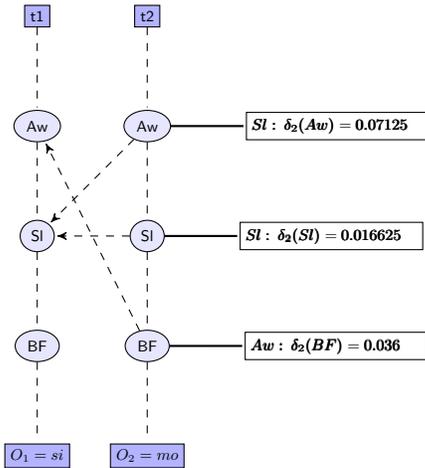
$$\delta_{t+1}(Y) = \max_{1 \leq X \leq N} [\delta_t(X) \cdot P(q_{t+1} = Y|q_t = X) \cdot P(O_{t+1}|q_t = Y)]$$

Der δ_{t+1} -Wert eines Knotens drückt also die Wahrscheinlichkeit seines *wahrscheinlichsten* Herkunftspfades (mal der Wahrscheinlichkeit der korrekten bisherigen Emissionen) aus. Die Kandidaten für die δ -Werte der zweiten Iteration sehen folgendermaßen aus:

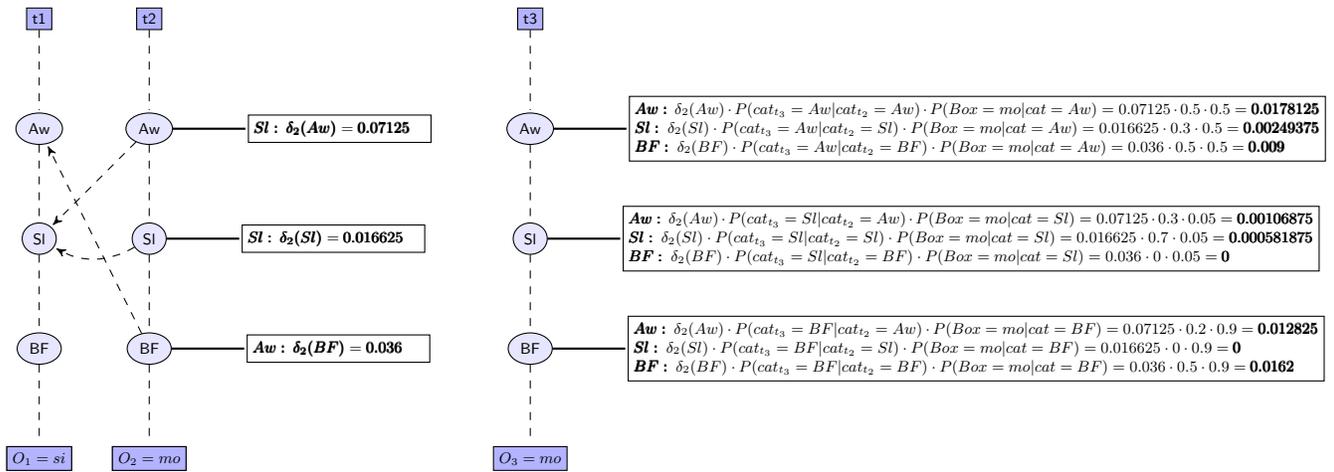


Dabei können wir schon gut den Vorteil des Viterbi-Algorithmus gegenüber dem Filtering sehen: dadurch, dass die Übergangswahrscheinlichkeit von BF zu Sl null ist, ist der $\delta_2(Sl)$ -Kandidat, der BF als Vorgänger betrachtet, ebenfalls null. In anderen Worten: die Wahrscheinlichkeit, dass die Katze in unserem Experiment nach einem wütenden Zustand in t_1 in einen schlafenden Zustand in t_2 gewechselt ist, ist null.

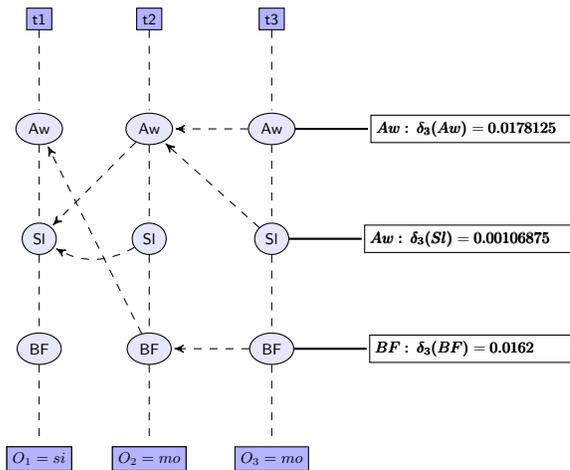
Jetzt wählen wir für jeden Zustand den besten δ_2 -Kandidaten aus (und merken uns den dazugehörigen Vorgänger - hier mit gestrichelten Pfeilen dargestellt):



... und für t_3 :

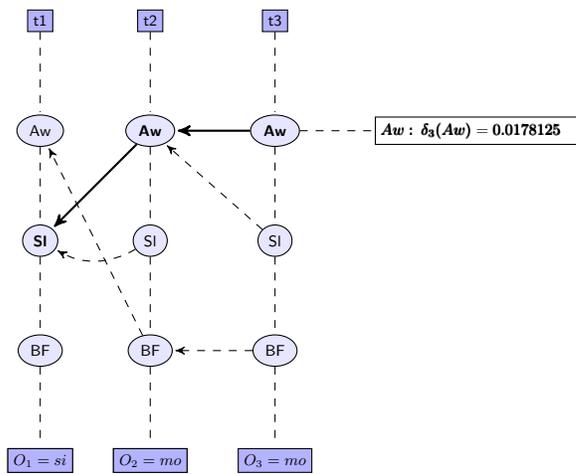


Also:



Beim Viterbi-Algorithmus behandeln wir das Problem der Suche nach der wahrscheinlichsten Zustandsfolge zu einer Beobachtungsreihe als eine Form der Suche: wir suchen den besten (also den wahrscheinlichsten) Pfad durch die möglichen Zustände vom Start zum Ende des Experiments. Dabei wählen wir bei jedem Schritt nur den besten, d.h. den wahrscheinlichsten, Pfad zu jedem möglichen Zwischenzustand aus.

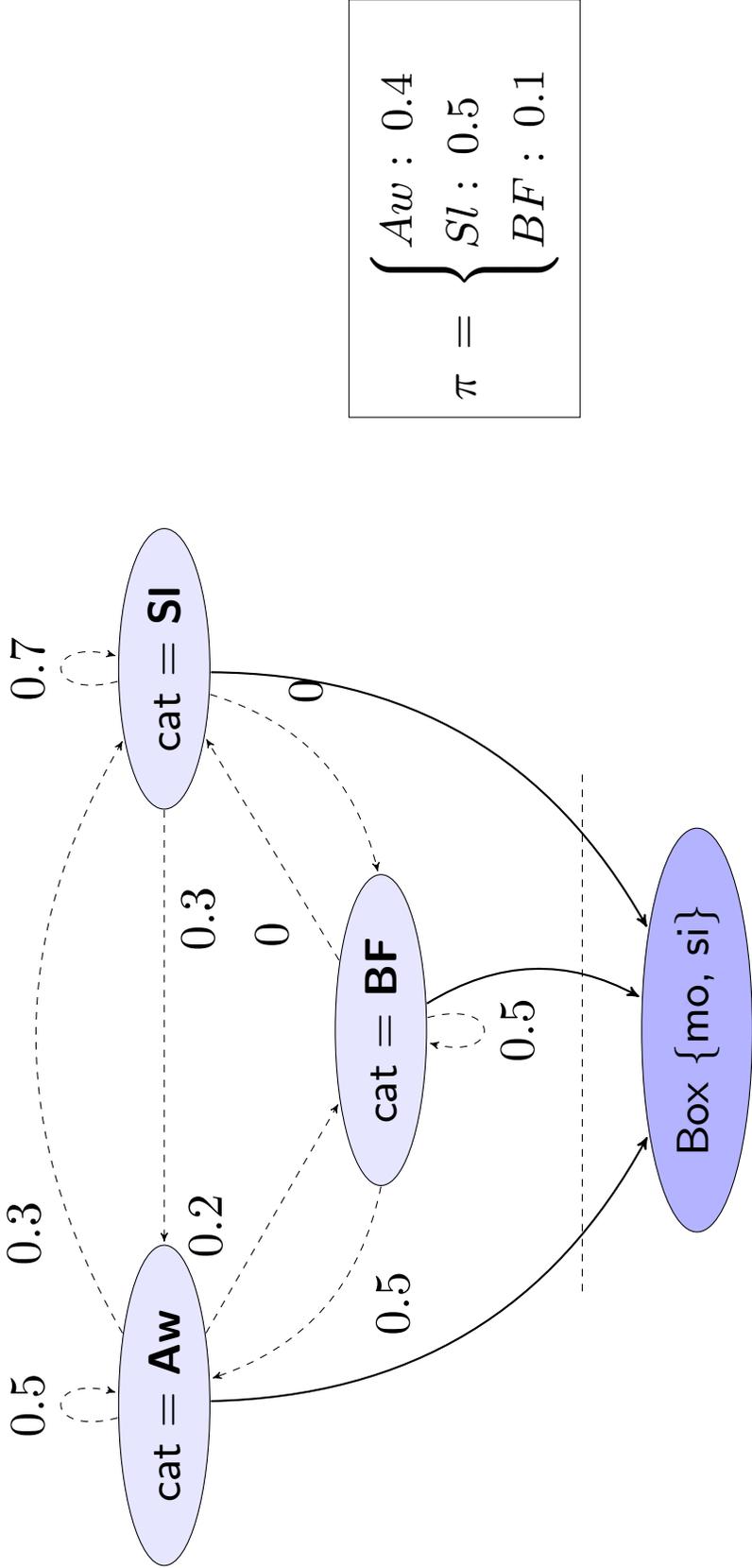
In den δ_3 -Werten steckt also jeweils die Gesamtqualität eines Pfades, d.h. die Wahrscheinlichkeit der jeweiligen Zustandsfolge. Die beste Zustandsfolge finden wir heraus, indem wir - wie bei einem klassischen Suchproblem - vom besten Endknoten aus rückwärts bis zum Start die Vorgänger aneinanderreihen. In unserem Beispiel ist der beste Endzustand Aw ($\delta_3(Aw) > \delta_3(BF) > \delta_3(SI)$). Wir betrachten also von Aw aus nacheinander den jeweils besten Vorgänger und kommen dabei auf folgenden Pfad:



Die wahrscheinlichste Erklärung für die Beobachtung $O = si, mo, mo$ ist also, dass die Katze das Experiment schlafend begonnen hat und in der zweiten und dritten Minute wach war.

Diese Methode, die wahrscheinlichste Zustandsfolge zu einer Beobachtungsreihe zu finden, nennt man **Decoding**.

Viel Erfolg bei den Prüfungen!



$$P(\text{Box}|\text{cat}) = \begin{matrix} & \text{cat} & \text{Box} = \text{moving} & \text{Box} = \text{silent} \\ \begin{pmatrix} \text{Awake} \\ \text{Sleeping} \\ \text{BF} \end{pmatrix} & \begin{pmatrix} 0.5 & 0.5 \\ 0.05 & 0.95 \\ 0.9 & 0.1 \end{pmatrix} \end{matrix}$$