Alf Gerisch
gerisch@informatik.uni-halle.de

Revision: 1.10, Date: 2003/02/04 10:32:34
Source: /home/gerisch/work/BioInfo2/MyNotes/RCS/HMM.tex,v

1 Definition

Ein HMM ist ein Tupel $(\mathcal{S}, \mathcal{E}, \Sigma, T, E)$, wobei

- $\mathcal{S} = \{S_1, S_2, \ldots S_N\}$ eine Menge N von Zuständen (states) mit Startzustand S_1 und Endzustand S_N,
- $\mathcal{E} = \{E_1, E_2, \ldots E_M\} \subset \mathcal{S}$ die Teilmenge der M, $M \leq N$, emittierenden Zustände ($S_1, S_N \notin \mathcal{E}$),
- Σ das Alphabet mit $|\Sigma| = A$ Buchstaben,
- $T \in \mathbb{R}^{N,N}$ die Übergangsmatrix ($t_{S_i,S_j} \geq 0$ ist die Wahrscheinlichkeit, dass das HMM vom Zustand S_j in den Zustand S_i wechselt; $\sum_{i=1}^{N} t_{S_i,S_j} = 1$ für alle $j = 1, 2, \ldots, N$; ferner sind $t_{S_1,S_j} = t_{S_j,S_N} = 0$ für alle $j = 1, 2, \ldots, N$, d.h. in den Startzustand wird nicht zurückgekehrt und der Endzustand wird nicht wieder verlassen),
- $E \in \mathbb{R}^{M,A}$ die Emissionsmatrix ($e_{E_i,X} \geq 0$ ist die Wahrscheinlichkeit, dass im emittierenden Zustand E_i, dass Zeichen $X \in \Sigma$ ausgegeben wird; $\sum_{X \in \Sigma} e_{E_i,X} = 1$ für alle $i = 1, 2, \ldots, M$)

sind. Sei W die Menge der freien Parameter des HMM-Modells M, kurz $M = M(W)$.

Ein Pfad π der Länge $L(\pi)$ des HMM ist eine Folge von $L(\pi) + 1$ Zuständen, also $\pi = s_0 s_1 s_2 \cdots s_{L(\pi)}$, $s_i \in \mathcal{S}$. Der Zustand s_0 ist immer der Startzustand S_1 und der Zustand $s_{L(\pi)}$ ist immer der Endzustand S_N des HMM. Die Wahrscheinlichkeit für einen Pfad π in $M(W)$ ist demnach

$$P(\pi|W) = \prod_{j=1}^{L(\pi)} t_{s_j,s_{j-1}} .$$

Ein Pfad π enthält eine Folge von $l(\pi) < L(\pi)$ emittierenden Zuständen, $e_1 e_2 e_3 \cdots e_{l(\pi)}$, $e_i \in \mathcal{E}$. Für jeden Pfad gibt es eine eindeutig bestimmte, streng monoton wachsende Funktion $a(i)$, so dass $e_i = s_{a(i)}$ für alle $i = 1, 2, \ldots, l(\pi)$. Der Pfad π kann nur Ausgaben O der Länge $l(\pi)$ erzeugen, $O = o_1 o_2 \cdots o_{l(\pi)}$, $o_i \in \Sigma$. Mit $l(O)$ bezeichnen wir auch die Länge einer Ausgabe.

Sei O eine beliebige Ausgabe. Die Wahrscheinlichkeit, dass ein Pfad π die Ausgabe O erzeugt ist gegeben durch:

$$P(O|\pi, W) = \begin{cases} 0 & : \quad l(\pi) \neq l(O) \\ \prod_{i=1}^{l(O)} e_{e_i,o_i} & : \quad l(\pi) = l(O) \end{cases} .$$

Damit gilt für die Wahrscheinlichkeit, dass M(W) einen Pfad π erzeugt, der eine Ausgabe O emittiert:

$$\begin{aligned} P(\pi, O|W) &= P(\pi|W) P(O|\pi, W) \\ &= \begin{cases} 0 & : \quad l(\pi) \neq l(O) \\ \prod_{j=1}^{L(\pi)} t_{s_j,s_{j-1}} \cdot \prod_{i=1}^{l(O)} e_{e_i,o_i} & : \quad l(\pi) = l(O) \end{cases} . \end{aligned}$$

Jeder mögliche Pfad in $M(W)$ erzeugt entweder eine gewünschte Ausgabe O auf genau eine Weise oder eben gar nicht. Damit gilt für die Wahrscheinlichkeit, dass $M(W)$ die Ausgabe O emittiert

$$P(O|W) = \sum_{\pi} P(\pi, O|W)\,.$$

2 Die Standardarchitektur

3 Algorithmen für HMMs

Wir führen eine Zeit t ein, die nach jeder Emission eines Zeichens um genau eins erhöht wird, d.h. zur Zeit t hat das HMM bereits eine Zeichenfolge $o_1 o_2 \cdots o_t$ emittiert, aber noch nicht o_{t+1}. Das Wechseln des HMMs in einen emittierenden Zustand, die Ausgabe eines Zeichens und das erhöhen der Zeit t um eins sind ein einziger atomarer Vorgang.

3.1 Forward-Algorithmus

Seien ein HMM $M(W)$ und eine Ausgabe O gegeben. Sei $\alpha_i(t) = P(S_i, o_1 o_2 \cdots o_t|W)$ die Wahrscheinlichkeit, dass sich $M(W)$ im Zustand $S_i \in \mathcal{S}$ befindet und $M(W)$ bereits genau $o_1 o_2 \cdots o_t$ emittiert hat. Für die gegebene Ausgabe O des HMMs möchten wir dann $P(O|W) = \alpha_N(l(O))$ berechnen. Die Werte der $\alpha_i(t), i = 1, 2, \ldots, N, t = 0, 1, \ldots, l(O)$ können rekursiv berechnet werden.

Für alle i mit $S_i \in \mathcal{E}$, also emittierende Zustände, gilt offensichtlich $\alpha_i(0) = 0$. Ferner gilt für nicht emittierende Zustände

$$\sum_{\{i:S_i \notin \mathcal{E}\}} \alpha_i(0) = 1\,.$$

Übung: Wie müssen die Startwerte für die Standardarchitektur gewählt werden?
Lösung: Für alle i mit $S_i \in \mathcal{E}$, also emittierende Zustände, gilt offensichtlich $\alpha_i(0) = 0$. Ferner gilt für nicht emittierende Zustände $\sum_{\{i:S_i \in \mathcal{S}\setminus\mathcal{E}\}} \alpha_i(0) = 1$. Für die Standardarchitektur (S_1 sei der Startzustand) gilt demnach für $S_i \notin \mathcal{E}$ und $S_i \neq S_1$ die Beziehung $\alpha_i(0) = \alpha_j(0) t_{S_i,S_j}$, wobei S_j der einzige Vorgänger von S_i ist, der nicht in $\mathcal{E}$ ist. Wir setzen $\alpha_1(0) = \xi$. Der Wert von ξ kann nun aus der Normierungsbedingung bestimmt werden und damit auch die $\alpha_i(0)$-Werte für alle anderen nicht emittierenden Zustände. Damit sind Anfangswerte $\alpha_i(0)$ für alle Zustände S_i eines HMM mit Standardarchitektur aus den Parametern des HMM eindeutig definiert. □

Die Werte $\alpha_i(t)$ für $t > 0$ ergeben sich aus folgenden Rekursionen:

$$S_i \in \mathcal{E} \quad \Rightarrow \quad \alpha_i(t+1) = \sum_{j=1}^{N} \alpha_j(t) t_{S_i,S_j} e_{S_i,o_{t+1}} = \sum_{\{j:S_j \in N^-(S_i)\}} \alpha_j(t) t_{S_i,S_j} e_{S_i,o_{t+1}}\,.$$

$$S_i \notin \mathcal{E} \quad \Rightarrow \quad \alpha_i(t+1) = \sum_{\{j:S_j \in N^-(S_i)\}} \alpha_j(t+1) t_{S_i,S_j}\,.$$

Hierbei bezeichnet $N^-(S_i)$ die Menge der möglichen, direkten Vorgängerzustände von S_i, also alle die S_j für die $t_{S_i,S_j} > 0$.

Übung: Zeigen Sie, dass diese Rekursion für die Standardarchitektur wohldefiniert ist.
Lösung: Die Startwerte $\alpha_i(0)$ können für die Standardarchitektur berechnet werden (siehe oben). Wir können die Werte $\alpha_i(t+1)$ für alle emittierenden Zustände $S_i \in \mathcal{E}$ aus den Werten $\alpha_i(t)$ berechnen. Für nicht emittierende Zustände S_i nehmen wir an (Induktionsannahme), dass für alle nicht emittierenden Zustände S_j, die zwischen dem Startzustand und S_i liegen, die Werte $\alpha_i(t+1)$ bereits berechnet sind. Diese Zustände sind für die Standardarchitektur eindeutig festgelegt. Für den Startzustand S_1 gilt

offensichtlich (Induktionsanfang) $\alpha_1(t) = 0$ für alle $t > 0$. Weiterhin sind nach Annahme alle $\alpha_j(t+1)$-Werte, die in der Summe der Formel für $\alpha_i(t+1)$ benötigt werden, bereits bekannt und somit ist auch $\alpha_(t+1)$ für $S_i \notin \mathcal{E}$ wohldefiniert (Induktionsschritt). □

Ein gerichteter Pfad, der in einem HMM vom Zustand S_j zum Zustand S_i führt, heißt *stummer Pfad*, wenn er nur durch nicht emittierende Zustände $S_k \notin \mathcal{E}$ verläuft (S_j kann emittierend sein). Mit $t^D_{S_i,S_j}$, $S_i \notin \mathcal{E}$ bezeichnen wir die Wahrscheinlichkeit, dass das HMM sich auf einem stummen Pfad vom Zustand S_j in den Zustand S_i begibt. Damit ist $t^D_{S_i,S_j}$ also die Summe der Wahrscheinlichkeiten aller stummen Pfade, die von S_j nach S_i verlaufen.

Übung: Maximal wieviele stumme Pfade von S_j nach $S_i \notin \mathcal{E}$ gibt es in der Standardarchitektur für jedes mögliche Paar (S_j, S_i)? Wie berechnen sich die zugehörigen $t^D_{S_i,S_j}$ in der Standardarchitektur?
Lösung: Es gibt maximal einen stummen Pfad von einem $S_j \in \mathcal{S}$ zu einem gegebenen $S_i \notin \mathcal{E}$. Damit berechnet sich $t^D_{S_i,S_j}$ einfach als das Produkt der Übergangswahrscheinlichkeiten, die diesen Pfad ausmachen (bzw. als $t^D_{S_i,S_j} = 0$ falls der stumme Pfad nicht existiert). □

Mit Hilfe der Notation $t^D_{S_i,S_j}$ können wir die Berechnung von $\alpha_i(t+1)$ für nichtemittierende Zustände $S_i \notin \mathcal{E}$ schreiben als:

$$S_i \notin \mathcal{E} \quad \Rightarrow \quad \alpha_i(t+1) = \sum_{\{j:S_j \in \mathcal{E}\}} \alpha_j(t+1) t^D_{S_i,S_j} .$$

Übung: Zeigen Sie, dass die Zeitkomplexität des Forward-Algorithmus für die Standardarchitektur mit N Hauptzuständen $\mathcal{O}\left(N^2\right)$ ist, wenn die Berechnung der Werte $\alpha_i(t+1)$ für die Löschzustände in einer günstigen (effizienten) Reihenfolge erfolgt.
Lösung: Für eine Ausgabe der Länge $l(O)$ müssen die α-Werte zu $l(O)$ Zeitpunkten t berechnet werden. Zu jedem Zeitpunkt sind somit $\approx 3N$ (Anzahl der Zustände in der Standardarchitektur) α-Werte zu bestimmen. Die Bestimmung von $\alpha_i(t+1)$ für einen emittierenden Zustand S_i ist in konstanter Zeit möglich, da S_i maximal drei direkte Vorgängerzustände hat. Die Bestimmung von $\alpha_i(t+1)$ für nicht emittierenden Zustände S_i erfolgt in der Standardarchitektur von links nach rechts. Dann sind die zur Bestimmung von $\alpha_i(t+1)$ benötigten Werte $\alpha_j(t+1)$ bereits bestimmt (für den Startzustand S_1 gilt offensichtlich $\alpha_1(t) = 0$ für $t > 0$). Somit kann auch $\alpha_i(t+1)$ für nicht emittierende Zustände S_i in konstanter Zeit bestimmt werden. Es ergibt sich eine Zeitkomplexität von $\mathcal{O}\left(l(O)N\right) = \mathcal{O}\left(N^2\right)$ (letzteres weil in der Standardarchitektur $l(O) = \mathcal{O}\left(N\right)$). □

3.2 Backward-Algorithmus

Der Backward-Algorithmus ist die Umkehrung des Forward-Algorithmus. Für eine Ausgabe O des HMM definieren wir Backward-Variablen $\beta_i(t) = P(S_i, o_{t+1}o_{t+2}\cdots o_{l(O)}|W)$, die die Wahrscheinlichkeit angeben, dass das HMM sich im Zustand $S_i \in \mathcal{S}$ befindet und bis zum Erreichen des Endzustands S_N noch die Zeichenfolge $o_{t+1}o_{t+2}\cdots o_{l(O)}$ ausgeben wird. Für eine gegebene Ausgabe O des HMM sind wir also am Wert $\beta_1(0) = P(O|W)$ interessiert.

Im Falle des Backward-Algorithmus werden die Variablen $\beta_i(t)$ für den Zeitpunkt $t = l(O)$ initialisiert. $\beta_i(l(O))$ bezeichnet die Wahrscheinlichkeit, dass das HMM sich im Zustand S_i befindet und in Zukunft kein Zeichen mehr emittieren wird, also der folgende Pfad nur noch nichtemittierende Zustände enthält bis schließlich der Endzustand S_N erreicht wird. Der auf S_i folgende Pfad ist also ein stummer Pfad von S_i nach S_N. Damit initialisieren wir $\beta_i(l(O)) = t^D_{S_N,S_i}$ für alle $i = 1, 2, \ldots, N$.

Für $t < l(O)$ ergibt sich die Wahrscheinlichkeit $\beta_i(t)$, dass das HMM sich in einem Zustand S_i befindet und in Zukunft noch $o_{t+1}o_{t+2}\cdots o_{l(O)}$ ausgeben wird aus der Summe der Wahrscheinlichkeiten über alle emittierenden Zustände S_j, dass man auf einem stillen Pfad von S_i nach S_j kommt, dort o_{t+1} ausgibt und von S_j in Zukunft noch genau $o_{t+2}\cdots o_{l(O)}$ ausgeben wird. In Formeln also:

$$\beta_i(t) = \sum_{\{j:S_j \in \mathcal{E}\}} t^D_{S_j,S_i} e_{S_j,o_{t+1}} \beta_j(t+1) .$$

3.3 Ableitung weiterer Wahrscheinlichkeiten aus $\alpha_i(t)$ und $\beta_i(t)$

Zu einem gegebenen HMM $M(W)$ und einer gegebenen Ausgabe O nehmen wir im folgenden an, dass die Werte $\alpha_i(t)$ und $\beta_i(t)$ für $t = 0, 1, 2, \ldots, l(O)$ und alle Zustände $S_i \in \mathcal{S}$ gegeben sind.

Die Wahrscheinlichkeit, dass das HMM sich zur Zeit t im Zustand S_i befindet, wenn die Ausgabe O und die Parameter W festliegen, ergibt sich zu

$$\gamma_i(t) = P(t, S_i|O, W) = \frac{P(t, S_i, o_1 o_2 \cdots o_t o_{t+1} \cdots o_{l(O)}|W)}{P(O|W)} = \frac{\alpha_i(t)\beta_i(t)}{P(O|W)} = \frac{\alpha_i(t)\beta_i(t)}{\sum\limits_{\{j:S_j \in \mathcal{S}\}} \alpha_j(t)\beta_j(t)} .$$

Die Wahrscheinlichkeit, dass das HMM zur Zeit t von einem Zustand S_i in einen Zustand S_j übergeht, wenn die Ausgabe O und die Parameter W festliegen, ergibt sich zu

$$\begin{aligned}\gamma_{j,i}(t) &= P(t, S_i \to S_j|O, W) = \frac{P(t, S_i \to S_j, O|W)}{P(O|W)} \\ &= \frac{1}{P(O|W)} \begin{cases} \alpha_i(t) t_{S_j,S_i} e_{S_j,o_{t+1}} \beta_j(t+1) & \text{if} \quad S_j \in \mathcal{E} \\ \alpha_i(t) t_{S_j,S_i} \beta_j(t) & \text{if} \quad S_j \notin \mathcal{E} \end{cases} .\end{aligned}$$

Es gilt offensichtlich auch

$$\gamma_i(t) = \sum_{\{j:S_j \in \mathcal{S}\}} \gamma_{j,i}(t)$$

und S_i mit

$$i = \arg \max_{i=1,2,\ldots,N} \gamma_i(t)$$

ist der wahrscheinlichste Zustand des HMMs zur Zeit t. Der wahrscheinlichste Pfad zur Erzeugung einer Ausgabe O kann mit dem Viterbi-Algorithmus berechnet werden.

3.4 Der Viterbi-Algorithmus

Sei O eine beliebige Ausgabe des HMMs $M(W)$ mit Parametern W. Der wahrscheinlichste Pfad im HMM zur Erzeugung von O kann mit dem Viterbi-Algorithmus berechnet werden.

Ein Pfad $\pi_i(t)$ heißt *Präfixpfad* von O, wenn der Pfad im Zustand $S_i \in \mathcal{S}$ endet und die Zeichen $o_1 o_2 \cdots o_t$ emittiert werden. Wir definieren $\delta_i(t)$ als die Wahrscheinlichkeit für den wahrscheinlichsten Pfad im HMM, der die ersten t Zeichen von O ausgibt und im Zustand S_i endet, also

$$\delta_i(t) = \max_{\pi_i(t)} P(\pi_i(t), o_1 o_2 \cdots o_t|W) .$$

Initialisiert werden die Werte mit $\delta_i(0) = \max_{i=1,2,\ldots,N} t^{D_{max}}_{S_i,S_1}$. Hierbei bezeichnet $t^{D_{max}}_{S_i,S_j}$ die *maximale* Wahrscheinlichkeit aller stummen Pfade von S_j nach S_i im Gegensatz zu $t^{D}_{S_i,S_j}$ aus dem Forward-Algorithmus, welches die *Summe* der Wahrscheinlichkeiten aller stummen Pfade von S_j nach S_i bezeichnet. Die Aktualisierung dieser Wahrscheinlichkeiten erfolgt analog zum Forward-Algorithmus, wobei allerdings Summen durch Maximumbildung zu ersetzen sind:

$$\begin{aligned}\delta_i(t+1) &= e_{S_i,o_{t+1}} \cdot \max_{\{j:S_j \in N^-(S_i)\}} \delta_j(t) t_{S_i,S_j} && \text{wenn } S_i \in \mathcal{E} , \\ \delta_i(t+1) &= \max_{\{j:S_j \in \mathcal{E}\}} \delta_j(t+1) t^{D_{max}}_{S_i,S_j} && \text{wenn } S_i \notin \mathcal{E} .\end{aligned}$$

Übung: (Wohldefiniertheit der $t^{D_{max}}_{S_i,S_j}$) Zeigen Sie, dass für beliebige Zustände $S_i \notin \mathcal{E}, S_j \in \mathcal{E}$ die Wahrscheinlichkeit $t^{D_{max}}_{S_i,S_j}$ immer definiert ist. Damit sind auch die Viterbi-Variablen $\delta_i(t)$ wohldefiniert.
Lösung: Sei ein beliebiger stummer Pfad π von S_j nach S_i gegeben. Angenommen dieser Pfad enthalte einen Zyklus $S_k \rightarrow S_{k+1} \rightarrow S_{k+2} \rightarrow \cdots \rightarrow S_{k+l} = S_k$. Die Wahrscheinlichkeit für diesen Zyklus ist gegeben durch $\prod_{j=0}^{l-1} t_{S_{k+j+1},S_{k+j}}$, wobei alle Übergangswahrscheinlichkeiten ≤ 1 sind und damit ist auch die Wahrscheinlichkeit für den Zyklus ≤ 1. Damit kann der Zyklus aus π entfernt werden und wir erhalten einen stummen Pfad $\tilde{\pi}$ von S_j nach S_i dessen Wahrscheinlichkeit nicht kleiner als die von π ist. Das bedeutet, dass man zur Bestimmung von $t^{D_{max}}_{S_i,S_j}$ nur Pfade betrachten muss, die keine Zyklen enthalten. Davon gibt es aufgrund der endlichen Zustandsanzahl im HMM aber nur endlich viele, somit nur endlich viele zugehörige Wahrscheinlichkeitswerte und deren Maximum ist $t^{D_{max}}_{S_i,S_j}$. □

Übung: (Konstruktion optimaler Pfade) Für jedes Paar (S_i, S_j), $S_i \notin \mathcal{E}, S_j \in \mathcal{E}$, mit $t^{D_{max}}_{S_i,S_j} > 0$ sei ein stummer Pfad π_{S_i,S_j} maximaler Wahrscheinlichkeit von S_j nach S_i gegeben. Mit Hilfe diese Pfade und den Größen $\delta_i(t)$ gebe man einen Algorithmus an, der einer Pfad vom Startzustand S_1 zum Endzustand S_N ermittelt, der die größte Wahrscheinlichkeit zur Ausgabe von O hat. (Hinweis: Zu jedem $\delta_i(t)$ merke man sich den vorhergehenden, optimalen Zustand, also jenes j, für das das Maximum angenommen wird.)
Lösung: Zu jedem $\delta_i(t)$ merken wir uns den vorhergehenden, optimalen Zustand in der Größe $\vartheta_i(t)$. Die ϑ-Werte sind also definiert durch $\vartheta_i(t+1) = \arg\max_{\{j:S_j \in N^-(S_i)\}} \delta_j(t) t_{S_i,S_j}$ wenn $S_i \in \mathcal{E}$ und $\vartheta_i(t+1) = \arg\max_{\{j:S_j \in \mathcal{E}\}} \delta_j(t+1) t^{D_{max}}_{S_i,S_j}$ wenn $S_i \notin \mathcal{E}$.

Weiterhin bezeichne π_{S_i,S_j} für jedes Paar (S_i, S_j), $S_i \notin \mathcal{E}, S_j \in \mathcal{E}$ mit $t^{D_{max}}_{S_i,S_j} > 0$ einen stummen Pfad maximaler Wahrscheinlichkeit von S_j nach S_i.

Wir nehmen an, dass die Größen $\delta_i(t)$ und $\vartheta_i(t)$ für eine gewünschte Ausgabe O (der Länge $l(O)$) des HMM mit dem Viterbi-Algorithmus berechnet worden sind. Wenn die Wahrscheinlichkeit $\delta_N(l(O)) > 0$ dann gibt es einen Pfad durch das HMM mit der Ausgabe O. Einen Pfad π mit der maximalen Wahrscheinlichkeit können wir dann durch folgenden Algorithmus berechnen.

```
π = []                                  % der zu konstruierende optimale Pfad (anfangs leer)
i = N                                   % der Zustand in dem wir uns befinden (anfangs Endzustand S_N)
t = l(O)                                % die aktuelle Zeit (anfangs Länge der Ausgabe)
while i ≠ 1                             % solange nicht im Anfangszustand S_1
  if S_i ∈ E
    π = [S_ϑi(t) → S_i, π]              % erweitere Pfad um vorhergehende Kante S_ϑi(t) → S_i
    i = ϑ_i(t)                          % sind nun im Zustand S_ϑi(t)
    t = t − 1                           % reduziere Zeit um 1
  else                                  % S_i ∉ E
    π = [π_{S_i,S_ϑi(t)}, π]            % erweitere Pfad um optimalen stummen Pfad von S_ϑi(t) nach S_i
    i = ϑ_i(t)                          % sind nun im Zustand S_ϑi(t)
  end
end
return π
```

□

Übung: (Zeitkomplexität des Viterbi-Algorithmus für Standardarchitektur) Wir betrachten ein HMM mit Standardarchitektur und N Hauptzuständen. Damit ist die Länge der Ausgaben dieses HMM von der Ordnung N. Leiten Sie einen Ausdruck für die Zeitkomplexität des Viterbi-Algorithmus, also der Berechnung der $\delta_i(t)$ für dieses HMM, her.
Lösung: Die Lösung ist analog zur Zeitkomplexität beim Forward-Algorithmus, wird hier aber nochmal ausführlicher gegeben. Die Anzahl der Zeiten t, für die die δ-Werte berechnet werden, ist gleich der Länge $l(O)$ der gewünschten Ausgabe O. (Die Werte für $t = 0$ sind bekannt.) Die Länge $l(O)$ ist von der Ordnung $\mathcal{O}(N)$, da die Länge der Standardarchitektur der Länge der erwarteten Ausgaben entsprechen soll. Für jeden dieser $l(O)$ Zeitpunkte muß für jeden Zustand ein δ-Wert berechnet werden. In der Standardarchitektur sind das also $3N + 3 = \mathcal{O}(N)$ zu berechnende Werte je Zeitpunkt.
Es bleibt also, die Komplexität der Berechnung eines Wertes $\delta_i(t)$ zu bestimmen. Wenn $S_i \in \mathcal{E}$ dann ist dazu das Maximum einer Menge zu bestimmen, deren Elementezahl von N und $l(O)$ unabhängig ist. Damit ist die Berechnung von $\delta_i(t)$ für $S_i \in \mathcal{E}$ in konstanter Zeit möglich, d.h. $\mathcal{O}(1)$. Das impliziert, dass der Viterbi-Algorithmus für ein HMM *ohne* nicht emittierende Zustände die Komplexität *durchschnittliche Wortlänge* × *Anzahl der Zustände* hat. Zur Bestimmung von $\delta_i(t)$ mit $S_i \notin \mathcal{E}$ nach der angege-

benen Formel muß ein Maximum über eine Menge gebildet werden, die soviele Elemente enthält wie es emittierende Zustände gibt. Damit ergibt sich im allgemeinen ein Komplexität (des Forward-Algorithmus) für ein HMM mit nicht emittierenden Zuständen von *durchschnittliche Wortlänge* $\times$ $(|\mathcal{E}| + |\mathcal{S} \setminus \mathcal{E}||\mathcal{E}|)$. Im Fall der Standardarchitektur ergibt sich daraus also also wegen $|\mathcal{E}| = \mathcal{O}(N)$ und ebenso $|\mathcal{S} \setminus \mathcal{E}| = \mathcal{O}(N)$ eine Komplexität von $\mathcal{O}\left(N^3\right)$. Bei genauerer Betrachtung des Algorithmus für die Standardarchitektur kann man aber zeigen, dass die Komplexität nur $\mathcal{O}\left(N^2\right)$ ist. Das wird im folgenden getan. Wir brauchen nur noch die Aktualisierung von $\delta_i(t)$ auf $\delta_i(t+1)$ betrachten, wenn S_i ein nicht emittierender Zustand, also entweder Start-, End- oder einer der Löschzustände D_1, D_2, …, D_N, ist. Falls S_i der Startzustand ist, dann wissen wir $\delta_i(t) = 0$ für alle $t > 0$, da kein Pfad wieder in den Startzustand zurückführt. Den Endzustand betrachten wir als Löschzustand D_{N+1} im folgenden. Wir führen die Berechnung der Werte $\delta_i(t+1)$ in der Reihenfolge für D_1, D_2, …, D_{N+1} aus. Sei $S_i = D_i$ ein solcher Zustand. Dann gilt (ohne die Verwendung stummer Pfade)

$$\delta_i(t+1) = \max_{\{j:S_j \in \mathcal{S}\}} \delta_j(t+1) t_{D_i,S_j} .$$

Aufgrund der Verbindungsstruktur der Standardarchitektur ist $t_{D_i,S_j} \neq 0$ nur für maximal drei Zustände S_j (zwei im Falle $S_i = D_1$). Für diese Zustände ist $\delta_j(t+1)$ bereits berechnet. Damit kann jedes $\delta_i(t+1)$ für nicht emittierende Zustände S_i in konstanter Zeit berechnet werden (wie es auch der Fall für emittierende Zustände ist). Die Komplexität des Viterbi-Algorithmus für die Standardarchitektur ergibt sich somit (bei entsprechender Implementierung) zu $\mathcal{O}\left(N^2\right)$. □

4 Lernalgorithmen

Ziel: Erste Ebene von Bayes'scher Inferenz, nämlich MAP-Schätzung der Parameter W.
Hier zuerst: ML-Schätzung für die Parameter, wobei der Datensatz nur eine einzige Sequenz O enthält (Online-Lernen).
Später dann: Batch-Learning, geteilte Parameter.

4.1 Erwartungswerte zu gegebenem HMM $M(W)$ und Ausgabe O

Die Wahrscheinlichkeit $P(\pi|O,W)$ definiert eine Posteriori-Verteilung für die Pfade π (versteckte Variablen). Wir definieren folgende Anzahlen:

- Sei $n(i|\pi,O)$ die Anzahl der Vorkommen von S_i im gegebenen Pfad π mit Ausgabe O (und Parametern W).
- Sei $n(i,X|\pi,O)$ die Anzahl der Ausgaben $X \in \Sigma$ vom Zustand S_i gegeben Pfad π mit Ausgabe O (und Parameter W).
- Sei $n(j,i|\pi,O)$ die Anzahl der Übergänge von S_i nach S_j im gegebenen Pfad π mit Ausgabe O (und Parametern W).

Damit ergeben sich die zugehörigen Erwartungen als

$$\begin{aligned}
n_i &= \sum_{\pi \text{ mit Ausgabe } O} n(i|\pi,O) P(\pi|O,W) = \sum_{t=0}^{l(O)} P(t,S_i|O,W) = \sum_{t=0}^{l(O)} \gamma_i(t) , \\
n_{i,X} &= \sum_{\pi \text{ mit Ausgabe } O} n(i,X|\pi,O) P(\pi|O,W) = \sum_{\{t:o_t=X\}} P(t,S_i|O,W) = \sum_{\{t:o_t=X\}} \gamma_i(t) , \\
n_{j,i} &= \sum_{\pi \text{ mit Ausgabe } O} n(j,i|\pi,O) P(\pi|O,W) = \sum_{t=0}^{l(O)} \gamma_{j,i}(t) .
\end{aligned}$$

4.2 Die Likelihoodfunktion und stationäre Punkte

Wir betrachten die Likelihood $P(O|W) = \sum_\pi P(\pi, O|W)$. Für die ML-Schätzung muss diese maximiert werden unter Beachtung der Nebenbedingungen $1 - \sum_{i=1}^{N} t_{S_i,S_j} = 0$ für alle $j = 1, 2, \ldots, N$ und $1 - \sum_{X\in\Sigma} e_{E_i,X} = 0$ für alle $i = 1, 2, \ldots, M$, sowie $t_{S_i,S_j}, e_{E_i,X} \geq 0$. Die Gleichheitsnebenbedingungen werden über Lagrange-Multiplikatoren an die Zielfunktion $P(O|W)$ angekoppelt. Wir erhalten die Lagrange-Funktion

$$\mathcal{L}(W, \lambda, \mu) = P(O|W) + \sum_{i=1}^{M} \lambda_i \left(1 - \sum_{X\in\Sigma} e_{E_i,X}\right) + \sum_{j=1}^{N} \mu_j \left(1 - \sum_{i=1}^{N} t_{S_i,S_j}\right) .$$

Wir suchen stationäre Punkte der Lagrange-Funktion.

Wir wollen zuerst $\frac{\partial P(\pi,O|W)}{\partial e_{E_i,X}}$ berechnen. Ist $l(\pi) \neq l(O)$, dann ist $P(\pi, O|W) = 0$ und das ändert sich auch nicht, wenn sich $e_{E_i,X}$ ändert. Also ist in diesem Fall die Ableitung gleich Null. Ist andererseits $l(\pi) = l(O)$, dann folgt mit $\frac{\mathrm{d}\, a x^n}{\mathrm{d}\, x} = n\frac{a x^n}{x}$

$$\frac{\partial P(\pi, O|W)}{\partial e_{E_i,X}} = n(i, X|\pi, O)\frac{P(\pi, O|W)}{e_{E_i,X}} .$$

Diese Formel ist offensichtlich auch für den ersten Fall gültig.

Damit erhalten wir für die Ableitung der Lagrange-Funktion nach dem Emissionsparameter $e_{E_i,X}$

$$\frac{\partial \mathcal{L}(W, \lambda, \mu)}{\partial e_{E_i,X}} = \sum_\pi \frac{\partial P(\pi, O|W)}{\partial e_{E_i,X}} - \lambda_i = \sum_\pi n(i, X|\pi, O)\frac{P(\pi, O|W)}{e_{E_i,X}} - \lambda_i .$$

Nullsetzen ergibt:

$$\lambda_i e_{E_i,X} = \sum_\pi n(i, X|\pi, O)P(\pi, O|W) = P(O|W)\sum_\pi n(i, X|\pi, O)P(\pi|O, W) = P(O|W)n_{i,X} .$$

Die letzte Gleichung über alle $X \in \Sigma$ summiert führt zu $\lambda_i = P(O|W)n_i$ wegen

$$\lambda_i \underbrace{\sum_{X\in\Sigma} e_{E_i,X}}_{=1} = P(O|W)\sum_\pi \sum_{X\in\Sigma} n(i, X|\pi, O)P(\pi|O, W) = P(O|W)\sum_\pi n(i|\pi, O)P(\pi|O, W) = P(O|W)n_i .$$

Damit erhalten wir als optimale Emissionswahrscheinlichkeiten

$$e_{E_i,X} = \frac{n_{i,X}}{n_i} = \frac{\sum_{\{t:o_t=X\}} \gamma_i(t)}{\sum_{t=0}^{l(O)} \gamma_i(t)}$$

und auf analogem Wege für die Übergangswahrscheinlichkeiten

$$t_{S_j,S_i} = \frac{n_{j,i}}{n_i} = \frac{\sum_{t=0}^{l(O)} \gamma_{j,i}(t)}{\sum_{t=0}^{l(O)} \gamma_i(t)} .$$

Das Ergebnis ist ein typisches Henne-Ei-Problem:
Die optimalen Parameter $e_{E_i,X}$ und t_{S_j,S_i} hängen von der Posterior-Verteilung $Q(\pi) := P(\pi|O, W)$ der Pfade ab. Andererseits hängt $Q(\pi)$ von den Parametern $e_{E_i,X}$ und t_{S_j,S_i} ab.
Deshalb verwende iterativen Algorithmus:

(1) Initialisiere alle $e_{E_i,X}$ und t_{S_j,S_i}.
repeat
(2) Schätze Posterior-Verteilung $Q(\pi)$ mit aktuellen Parametern $e_{E_i,X}$ und t_{S_j,S_i}.
(3) Aktualisiere Parameter $e_{E_i,X}$ und t_{S_j,S_i} mit (aktueller Schätzung von $Q(\pi)$.
until Konvergenz.

Ein Beispiel für einen solchen Algorithmus ist der Baum-Welsh-Algorithmus.

4.3 Baum-Welsh-Algorithmus

Hier wird die Schätzung der Posterior-Verteilung $Q(\pi)$ mit den aktuellen Parametern implizit ausgeführt, d.h. $Q(\pi)$ wird nicht explizit geschätzt:

(1) Initialisiere alle $e_{E_i,X}$ und t_{S_j,S_i}.
repeat
(2a) Berechne $\gamma_i(t), \gamma_{j,i}(t)$ mit Forward- und Backward-Algorithmus.
(2b) Bestimme aus den $\gamma_i(t), \gamma_{j,i}(t)$ die Erwartungen $n_i, n_{i,X}, n_{j,i}$.
(3) Aktualisiere Parameter $e_{E_i,X}$ und t_{S_j,S_i} durch $e_{i,X} = n_{i,X}/n_i$ und $t_{S_j,S_i} = n_{j,i}/n_i$.
until $\|$Parameteränderung$\| < \varepsilon$ oder maximale Iterationszahl erreicht.

Jetzt: K Beobachtungen O_k. Es gibt zwei Möglichkeiten;

1. Online-Lernen, d.h. alle Beobachtungen einzeln und in zufälliger Reihenfolge zum Lernen präsentieren. Dies funktioniert oft nicht sehr gut, da wir keine Lernschrittweite einstellen können und deshalb große Parameter-„sprünge“ auftreten können.
2. Batch-Lernen: Ähnliche Rechnung führt zu optimalen Parametern

$$e_{i,X} = \frac{\sum_{k=1}^{K} \sum_{\pi} n(i, X|\pi, O_k) P(\pi|O_k, W)}{\sum_{k=1}^{K} \sum_{\pi} n(i|\pi, O_k) P(\pi|O_k, W)} \quad \text{bzw.} \quad t_{S_j,S_i} = \frac{\sum_{k=1}^{K} \sum_{\pi} n(j, i|\pi, O_k) P(\pi|O_k, W)}{\sum_{k=1}^{K} \sum_{\pi} n(i|\pi, O_k) P(\pi|O_k, W)} .$$

Der Aufwand für das Lernen von K Beobachtungen mit der Standardarchitektur mit N Hauptzuständen ist in beiden Fällen $\mathcal{O}(KN^2)$ (je einmal Forward- und Backward-Algorithmus für jeder der K Beobachtungen).

4.4 Viterbi-Training

Idee: Modifiziere Baum-Welsh-Algorithmus, so dass die Erwartungen $n_i, n_{i,X}, n_{j,i}$ nicht über alle möglichen Pfade berechnet werden, sondern nur über eine kleine Anzahl wahrscheinlicher Pfade (Viterbi-Baum-Welsh-Alg.) bzw. nur über den wahrscheinlichsten Pfad (einfacher Viterbi-Baum-Welsh-Alg.).

Der einfache Viterbi-Baum-Welsh-Algorithmus macht im Online-Lernmodus auf der Standardarchitektur wenig Sinn (wegen $n(i, X|\pi^*(O))$ nur 0 oder 1 (außer in Insert-Zuständen). Ferner spart der Viterbi-Ansatz im Vergleich zum Standard-Baum-Welsh-Algorithmus nur die Ausführung des Backward-Algorithmus und ist somit nur um einen Faktor zwei schneller. Viterbi-Training führt oft zu schlechteren Parametern für HMMs. Es ist gut geeignet für die Modellierung von Proteinfamilien (weil hier wahrscheinlich optimale Pfade eine signifikante Rolle spielen), aber schlecht für die Modellierung von allgemeinen DNA-Sequenzen (wie zum Beispiel Exon- oder Promotor-Regionen).

4.5 HMM's mit geteilten Parametern—Modifizierter Baum-Welsh-Algorithmus

Wir betrachten ein HMM, in dem Emissionswahrscheinlichkeiten $e_{E_i,X}$ geteilt werden, das heißt, dass eine Teilmenge der emittierenden Zuständen die gleichen Emissionswahrscheinlichkeiten haben soll. Sei eine Teilmenge $\mathcal{I} \subset \{1, 2, \ldots, M\}$ gegeben und gelte für alle $i \in \mathcal{I}$

$$e_{E_i,X} = e_{\mathcal{I},X} \geq 0 \quad \text{für alle} \quad X \in \Sigma \quad \text{und} \quad \sum_{X \in \Sigma} e_{\mathcal{I},X} = 1 .$$

Die $e_{\mathcal{I},X}$, $X \in \Sigma$ bezeichnen also die geteilten Emissionswahrscheinlichkeiten in der durch $\mathcal{I}$ definierten Teilmenge von $\mathcal{E}$.

Ziel ist es, eine Formel für die Maximum-Likelihood Schätzung der geteilten Parameter des HMMs zu ermitteln. Diese können dann in einem leicht modifizierten Baum-Welsh-Algorithmus verwendet werden. Wir betrachten den Fall des Online-Lernens. Es ist also eine Ausgabe O gegeben.

Übung: (Aufstellen der Lagrange-Funktion) Der Lagrange-Multiplikator zur Nebenbedingung $\sum_{X\in\Sigma} e_{\mathcal{I},X} = 1$ sei mit $\lambda_{\mathcal{I}}$ bezeichnet (die anderen Multiplikatoren bezeichnen wir wie bisher als λ_i und μ_j). Geben Sie die Lagrangefunktion $\mathcal{L}(W,\lambda,\mu)$ für die ML-Schätzung an.
Lösung: Für die ML-Schätzung ergibt sich folgende Lagrange-Funktion

$$\mathcal{L}(W,\lambda,\mu) = P(O|W) + \lambda_{\mathcal{I}}\left(1 - \sum_{X\in\Sigma} e_{\mathcal{I},X}\right) + \sum_{i\in\{1,2,\ldots,M\}\setminus\mathcal{I}} \lambda_i \left(1 - \sum_{X\in\Sigma} e_{E_i,X}\right) + \sum_{j=1}^{N} \mu_j \left(1 - \sum_{i=1}^{N} t_{S_i,S_j}\right) .$$

□

Übung: (Berechnung der Ableitung der Lagrangefunktion) Wir suchen stationäre (kritische) Punkte der oben erhaltenen Lagrange-Funktion. Zeigen Sie, dass für die Ableitung von $\mathcal{L}(W,\lambda,\mu)$ nach dem geteilten Parameter $e_{\mathcal{I},X}$ gilt

$$\frac{\partial \mathcal{L}(W,\lambda,\mu)}{\partial e_{\mathcal{I},X}} = \sum_{\pi}\sum_{i\in\mathcal{I}} n(i,X|\pi,O)\frac{P(\pi,O|W)}{e_{\mathcal{I},X}} - \lambda_{\mathcal{I}} .$$

(Hinweis: Sei $f(x,y)$ eine differenzierbare Funktion und seien $x = x(t)$ sowie $y = y(t)$ ebenfalls differenzierbar. $f(x(t),y(t))$ heißt *mittelbare Funktion*. Geben Sie die Ableitung von $f(x(t),y(t))$ nach t an. Was ist die Ableitung von $f(x(t),y(t))$ nach t wenn $x(t) = y(t) = t$ gilt?)
Lösung: Für eine Funktion $f(x,y)$ mit $x = x(t)$ und $y = y(t)$ gilt

$$\frac{\mathrm{d}\, f(x(t),y(t))}{\mathrm{d}\, t} = \frac{\partial f(x(t),y(t))}{\partial x}\frac{\partial x(t)}{\partial t} + \frac{\partial f(x(t),y(t))}{\partial y}\frac{\partial y(t)}{\partial t}$$

und mit $x(t) = t$ und $y(t) = t$ somit

$$\frac{\mathrm{d}\, f(x(t),y(t))}{\mathrm{d}\, t} = \frac{\partial f(x(t),y(t))}{\partial x} + \frac{\partial f(x(t),y(t))}{\partial y} .$$

Damit folgt

$$\frac{\partial P(\pi,O|W)}{\partial e_{\mathcal{I},X}} = \sum_{i\in\mathcal{I}} n(i,X|\pi,O)\frac{P(\pi,O|W)}{e_{\mathcal{I},X}} .$$

Es folgt für die Ableitung der Lagrange-Funktion nach dem geteilten Emissionsparameter $e_{\mathcal{I},X}$

$$\frac{\partial \mathcal{L}(W,\lambda,\mu)}{\partial e_{\mathcal{I},X}} = \sum_{\pi}\frac{\partial P(\pi,O|W)}{\partial e_{\mathcal{I},X}} - \lambda_{\mathcal{I}} = \sum_{\pi}\sum_{i\in\mathcal{I}} n(i,X|\pi,O)\frac{P(\pi,O|W)}{e_{\mathcal{I},X}} - \lambda_{\mathcal{I}} .$$

□

Übung: (Berechnung des stationären Punkts) Setzen Sie die erhaltene Ableitung gleich Null und summieren Sie über alle Zeichen des Alphabets Σ auf. Leiten Sie daraus einen Ausdruck für den Multiplikator $\lambda_{\mathcal{I}}$ ab. Nutzen Sie diesen, um zu zeigen, dass für einen stationären Punkt der Lagrange-Funktion gilt

$$e_{E_{\mathcal{I}},X} = \frac{\sum_{i\in\mathcal{I}} n_{i,X}}{\sum_{i\in\mathcal{I}} n_i} .$$

Lösung: Nullsetzen ergibt:

$$\lambda_{\mathcal{I}} e_{\mathcal{I},X} = \sum_{\pi}\sum_{i\in\mathcal{I}} n(i,X|\pi,O)P(\pi,O|W) = P(O|W)\sum_{i\in\mathcal{I}}\sum_{\pi} n(i,X|\pi,O)P(\pi|O,W) = P(O|W)\sum_{i\in\mathcal{I}} n_{i,X} .$$

Die letzte Gleichung über alle $X \in \Sigma$ summiert führt zu

$$\lambda_i = P(O|W) \sum_{i \in \mathcal{I}} \sum_{\pi} \sum_{X \in \Sigma} n(i, X|\pi, O) P(\pi|O, W) = P(O|W) \sum_{i \in \mathcal{I}} \sum_{\pi} n(i|\pi, O) P(\pi|O, W) = P(O|W) \sum_{i \in \mathcal{I}} n_i \,.$$

Damit erhalten wir, dass für stationäre Punkte der Lagrangefunktion gelten muss

$$e_{E_{\mathcal{I}}, X} = \frac{\sum_{i \in \mathcal{I}} n_{i,X}}{\sum_{i \in \mathcal{I}} n_i} \,.$$

□

Übung: (Anpassung des Baum-Welsh-Algorithmus an den Fall geteilter Emissionswahrscheinlichkeiten) Geben Sie die nötigen Änderungen am Baum-Welsh-Algorithmus an, so dass er auch mit geteilten Emissionswahrscheinlichkeiten verwendet werden kann.
Lösung: Nur die Aktualisierung der HMM-Parameter (Schritt (3) des Baum-Welsh-Algorithmus) muss entsprechend der letzten Formel der vorigen Übung modifiziert werden. □

Analoge Aussagen sind auch für geteilte Übergangswahrscheinlichkeiten möglich.

5 Weitere Aspekte

5.1 Skalierung

Problem: $P(\pi|O, W)$ i.A. sehr klein da Produkt von Wkts. ≤ 1. Für realistische HMMs wird Maschinengenauigkeit unterschritten, besonders im Forward- und Backward-Algorithmus. Es gibt skalierte Varianten (recht technisch), bei denen die $\alpha_i(t)$ und $\beta_i(t)$ während der Berechnung skaliert werden und somit Underflow vermieden wird.

5.2 Lernen der Architektur

Es gibt Algorithmen dazu, die auch im Zusammenhang mit biologischen Sequenzen angewendet wurden. Es gibt 2 Grundideen:

1. Starte mit komplexen Modell (1 Zustand pro Datenbuchstabe) und führe iterativ Zustände zusammen, wobei Kriterien dafür aus der Posteriori-Verteilung abgeleitet sind.
2. Starte mit kleinem, voll vernetztem HMM. Iterativ werden dann Übergänge mit kleinen Wkts. entfernt und am meisten verbundene Zustände werden dupliziert. Iteration wird beendet, wenn die Likelihood oder Posterior ein gewünschtes Level erreicht haben.

Mit diesen Ansätzen wurden gute Ergebnisse bei kleinen HMMs erzielt (bis 50 Zustände). Die Algorithmen sind aber für heutige Computer und Probleme zu langsam und nicht praktikabel.

5.3 Anpassung der Länge der Standardarchitektur

⇒ Abgeschwächte Form des Lernens der Architektur.
Bisher: Feste Länge der Standardarchitektur (ducrhschnittliche Sequenzlänge). In Praxis bewährt und falls Länge ungünstig erscheint, dann kann man diese modifizieren und erneut Lernen.

Jetzt: Algorithmus zum Anpassen der Länge während des Lernens. Die Idee ist, Zustände hinzuzufügen bzw. zu entfernen, aber die Verbindungstruktur der Standardarchitektur zu erhalten. Z.B. wird ein Insert-Zustand von mehr als 50% der Sequenzen der Modellfamilie genutzt, dann füge neuen Hauptzustand mit zugehörigen Insert- und Delete-Zuständen ein. Analog werden Delete-Zustände (zusammen mit den entsprechenden Haupt- und Insert-Zuständen) entfernt, wenn Sie von mehr als 50% der Sequenzen genutzt werden.

5.4 Variation der Architektur

Variationen sind meist von der Standardarchitektur abgeleitet. Beispiele sind multiple HMMs zur Klassifikation und HMMs vom *Wheel*- oder *Loop*-Typ zur Charakterisierung periodischer Muster.

5.5 Mehrdeutige Symbole

Mehrdeutige Symbole kennt man über verschiedenen Alphabeten. Sie können in die HMMs einbezogen werden, in dem z.B. mehrdeutige Symbole in einer Sequenz durch die wahrscheinlichste Alternative ersetzt.

6 Anwendung von HMMs: Allgemeine Aspekte

1. Multiple Alignments
2. Datenbanksuche und Klassifikation von Sequenzen und Fragmenten
3. Strukturaufklärung und Mustererkennung

Die Basis für die Anwendungen ist, dass für eine gegebene Sequenz die Likelihood und der wahrscheinlichste Pfad berechnet werden. HMMs wurden auf allen drei Problemfeldern erfolgreich eingesetzt.

Multiple Alignments

- Die Berechnung des Viterbi-Pfads einer Sequenz wird auch als Ausrichten der Sequenz am Modell bezeichnet.
- Ein multiples Alignment erhält man durch Berechnung eines Alignments der Viterbi-Pfade.
- HMM-Training dauert lang, aber dann ist Alignment offline möglich. Multiples Alignment von K Sequenzen: $\mathcal{O}(KN^2)$, also linear in K, im Vergleich zu $\mathcal{O}(N^K)$, also exponentiell in K, für mehrdimensionales dynamisches programmieren.

Datenbanksuche und Klassifikation von Sequenzen und Fragmenten
Trennen von Sequenzen, die mit der Trainingsfamilie assoziiert sind, von solchen, die das nicht sind, kann durch Auswertung der Likelihood der Sequenzen und zugehöriger Viterbi-Pfade erfolgen. Zur Klassifikation kann man dann zum Beispiel ein HMM je Klasse Trainieren.

Strukturaufklärung und Mustererkennung

- Analysiere Struktur und Parameter eines trainierten HMMs. Hohe Wahrscheinlichkeiten deuten zum Beispiel auf konservierte Regionen bzw. Consensus-Muster hin.

- Das entdecken schwacher Muster in der Struktur oder den Parameter kann hilfreich sein bei der Entwicklung angepaßterer Architekturen.
- Fähigkeit von HMMs zur Erkennung von schwachen Mustern in rohen, nicht ausgerichteten Daten ist eine sehr nützliche Eigenschaft von HMMs.