Hidden Markov Models und DNA-Sequenzen

Joana Grah

Seminar: Mathematische Biologie

Sommersemester 2012

Betreuung: Prof. Dr. Matthias Löwe, Dr. Felipe Torres

Institut für Mathematische Statistik

28. Juni 2012

Einführung

Wir haben im siebten Vortrag schon die Theorie der diskreten Markovketten mit endlichem Zustandsraum kennen gelernt, einige elementare Definitionen wie Homogenität, Aperiodizität und Irreduzibilität als für unsere Zwecke hilfreiche Eigenschaften erläutert und daraus dann die Existenz und Eindeutigkeit einer stationären Verteilung hergeleitet. Eine Markovkette ist immer schon vollständig durch den Vektor ihrer Anfangsverteilung $\mu^{(0)}$ und die Übergangsmatrix $\mathbf{P} = (p_{ij})_{i,j\in\mathcal{A}}$ charakterisiert, wobei $\mathcal{A}$ den Zustandsraum bezeichnet. Unabhängig von $\mu^{(0)}$ konvergiert $\mu^{(t)}$, der Vektor der Verteilung zum Zeitpunkt $t \in \mathbb{N}$, dann in Totalvariation gegen die stationäre Verteilung π.

Zur Modellierung von DNA-Sequenzen haben wir Markovketten betrachtet, die zu jedem Zeitpunkt einen Buchstaben des Alphabets $\mathcal{A} = \{A, C, G, T\}$ erzeugen. Dadurch entstehen Sequenzen unendlicher Länge, von denen wir immer nur die ersten L Folgenglieder zur Analyse verwandt haben. Dann summieren sich die Wahrscheinlichkeiten für alle denkbaren Sequenzen mit fester endlicher Länge, die von der Markovkette erzeugt werden können, zu 1 auf. Wir haben auch gesehen, dass diese Eigenschaft sogar ohne Fixierung einer festen Länge gilt, falls die Markovkette nichttrivial verbunden ist.

In der Praxis sind oft so genannte *Trainingsdaten* gegeben, also DNA-Sequenzen, die bereits analysiert und als Gene erkannt wurden und aus denen wir durch Berechnung relativer Häufigkeiten die *A-priori-Konnektivität* festlegen, d.h. wir schätzen die Übergangswahrscheinlichkeiten. Anschließend kann dann die daraus resultierende Markovkette verwandt werden, um weitere DNA-Sequenzen mit den Trainingsdaten zu vergleichen und bestimmte Aussagen zu machen, z.B. ob die gegebene Sequenz ebenfalls ein Gen ist.

Prokaryotische Gene, die Proteine kodieren, haben die folgende Gestalt:

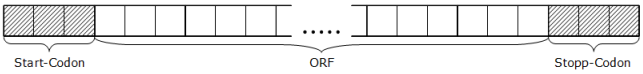

Abbildung 1: schematischer Aufbau eines prokaryotischen Gens

Jedes Quadrat steht für einen Buchstaben (*Nukleotid*) und jedes Tripel ergibt ein *Codon*, welches wiederum eine Aminosäure kodiert. Da die Start- und Stopp-Codons immer aus denselben Buchstabenkombinationen bestehen, ist es leicht, die *offenen Leserahmen* (*ORF = open reading frames*), die zur Bestimmung der Übergangswahrscheinlichkeiten betrachtet werden, zu lokalisieren.

Die Struktur von eukaryotischen Genen ist allerdings komplexer. Es gibt kodierende und nichtkodierende Abschnitte (*Exons* bzw. *Introns*). Ein Ziel der Verwendung von *Hidden Markov Models* ist, dies zu berücksichtigen und dadurch eine realistischere Modellierung zu erzielen. Des Weiteren ist das Auffinden der ORF mit hohem Aufwand verbunden, zumal die Anzahl an Trainingsdaten sehr hoch sein kann und auch sollte, wenn gute Näherungen der Wahrscheinlichkeiten erzielt werden sollen. Daher möchte man Algorithmen finden, die eine gegebene Sequenz direkt analysieren und zusätzlich weitere Eigenschaften untersuchen können.

Hidden Markov Models

Definition
Ein ***Hidden Markov Model (HMM)*** ist eine diskrete, nichttrivial verbundene Markovkette mit endlichem Zustandsraum $\mathcal{Z} = \{Z_1, \ldots, Z_N\}$, Übergangsmatrix $\mathbf{P} = (p_{ij})$, $i, j = Z_1, \ldots, Z_N$, und Anfangsverteilung $\mu^{(0)} = (\mu^{(0)}_{Z_1}, \ldots, \mu^{(0)}_{Z_N})$, die in jedem Zustand Buchstaben aus einem Alphabet $\mathcal{A}$ ausgibt.
Für jeden Zustand Z_k und jeden Buchstaben $a \in \mathcal{A}$ wird eine *Emissionswahrscheinlichkeit* definiert durch $q_{Z_k}(a)$.

Es gilt:

$$\sum_{a \in \mathcal{A}} q_{Z_k}(a) = 1 \ \forall k = 1, \ldots, N. \tag{1}$$

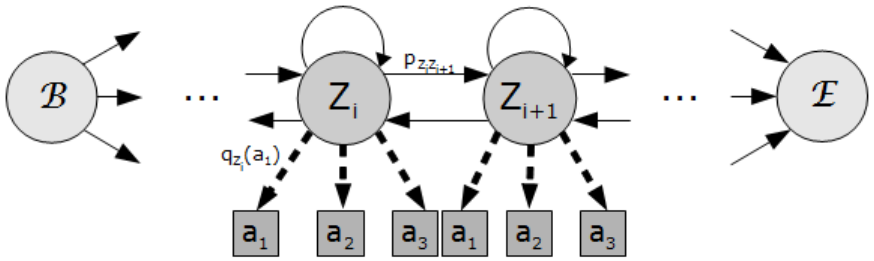

Abbildung 2: Hidden Markov Model

In Abbildung 2 ist ein Hidden Markov Model mit dreielementigem Alphabet dargestellt. $\mathcal{B}$ und $\mathcal{E}$ sind die bereits bekannten Anfangs- und Endzustände. Wir sehen nur die Sequenz aus den Buchstaben des Alphabets, aber nicht, welcher Zustand sie emittiert hat.

Die Zustandssequenz bleibt versteckt. Daher kommt auch der Name *Hidden* Markov Model. Es besitzt also zusätzliche Eigenschaften, die es uns ermöglichen werden, biologische Phänomene besser beschreiben zu können als gewöhnliche Markovketten.

Betrachten wir zunächst ein einfaches, aber wichtiges Beispiel, denn durch folgendes Modell können Exons und Introns modelliert werden.

Das „Two-Block"-Modell

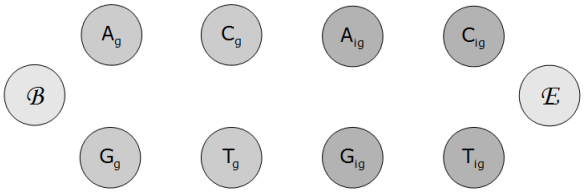

Abbildung 3: Zustände des Two-Block-Modells

Gegeben ist eine Markovkette mit Zustandsraum $\mathcal{Z} = \{A_g, C_g, G_g, T_g, A_{ig}, C_{ig}, G_{ig}, T_{ig}\}$. „g" steht für „gene", also für ein Nukleotid aus einem Exon, „ig" steht für „intergenic", also für Buchstaben aus einem Intron. Wir nehmen an, dass die Markovkette vollständig und nichttrivial verbunden ist, d.h. sämtliche Übergangswahrscheinlichkeiten zwischen den Zuständen existieren und jeder Zustand, der von $\mathcal{B}$ aus erreicht werden kann, hat eine positive Übergangswahrscheinlichkeit zu $\mathcal{E}$. Die Buchstaben des Alphabets, die in jedem Zustand emittiert werden, sind die vier Basen: $\mathcal{A} = \{A, C, G, T\}$. Ihre Emissionswahrscheinlichkeiten sind dann gegeben durch

$$\begin{aligned} q_{A_g}(A) &= q_{A_{ig}}(A) = 1 \\ q_{A_g}(C) &= q_{A_{ig}}(C) = 0 \\ q_{A_g}(G) &= q_{A_{ig}}(G) = 0 \\ q_{A_g}(T) &= q_{A_{ig}}(T) = 0 \end{aligned}$$

$$\begin{aligned} q_{C_g}(A) &= q_{C_{ig}}(A) = 0 \\ q_{C_g}(C) &= q_{C_{ig}}(C) = 1 \\ q_{C_g}(G) &= q_{C_{ig}}(G) = 0 \\ q_{C_g}(T) &= q_{C_{ig}}(T) = 0 \end{aligned}$$

$$\begin{aligned} q_{G_g}(A) &= q_{G_{ig}}(A) = 0 \\ q_{G_g}(C) &= q_{G_{ig}}(C) = 0 \\ q_{G_g}(G) &= q_{G_{ig}}(G) = 1 \\ q_{G_g}(T) &= q_{G_{ig}}(T) = 0 \end{aligned}$$

$$\begin{aligned} q_{T_g}(A) &= q_{T_{ig}}(A) &= 0 \\ q_{T_g}(C) &= q_{T_{ig}}(C) &= 0 \\ q_{T_g}(G) &= q_{T_{ig}}(G) &= 0 \\ q_{T_g}(T) &= q_{T_{ig}}(T) &= 1. \end{aligned}$$

Hierbei sind Übergänge zwischen sämtlichen Zuständen erlaubt und die Übergangsmatrix ist eine 10×10-Matrix (wenn wir $\mathcal{B}$ und $\mathcal{E}$ als Zustände ansehen). Dadurch gibt es sehr viele Parameter in dem Modell und um diese schätzen zu können wird eine große Anzahl an Trainingsdaten benötigt. Deswegen braucht man für die Praxis ein einfacheres Modell, das wir im Beispiel des Abschnitts „Parameterschätzung“ kennen lernen werden.

Es gibt mehrere mögliche Problemstellungen, wenn man mit Hidden Markov Models arbeitet:

1. Wenn sowohl die Übergangs- als auch die Emissionswahrscheinlichkeiten gegeben sind, also z.B. durch Trainingsdaten geschätzt wurden, kann man für eine Sequenz $x = x_1 \dots x_L$ von Buchstaben aus $\mathcal{A}$ die Wahrscheinlichkeit des Sequenzpaares (x, π) definieren. Hierbei ist $\pi = \pi_1 \dots \pi_L$ eine Sequenz von Zuständen aus $\mathcal{Z}$, die wir im Folgenden *Pfad* nennen. Es liegt also die Situation vor, dass wir x kennen und gerne Aussagen über π treffen möchten. Man kann dann den wahrscheinlichsten Pfad π, durch den x erzeugt wurde, finden. Diesen Vorgang nennt man *Dekodierung* (*Entschlüsselung*).

2. Umgekehrt können gegebene Pfade betrachtet und dann Aussagen über Sequenzen getroffen werden. Gesucht ist dann die Wahrscheinlichkeit für eine Sequenz x, aus dem gegebenen Hidden Markov Model generiert worden zu sein (*Gesamtwahrscheinlichkeit*).

3. Ein weiteres Problem ist die Parameterschätzung für eine gegebene Menge von Trainingsdaten. Auch hier unterscheidet man die zwei Fälle, dass die Pfade entweder bekannt oder unbekannt sind. Wir werden uns nur auf den einfachen Fall beschränken, dass Pfade bekannt sind.

4. Oft möchte man auch für eine gegebene Sequenz x ermitteln, ob sie mit bestimmten Trainingsdaten „verwandt“ ist. Es wird die Wahrscheinlichkeit für (x, π) bzw. x berechnet und wenn diese einen vorher festgelegten Schwellenwert überschreitet, steht die gegebene Sequenz tatsächlich in Beziehung zu den Trainingsdaten. Es wird also ein Test durchgeführt. Hier kann es allerdings zu Fehlern 1. und 2. Art kommen.

Im Folgenden werden verschiedene Algorithmen vorgestellt, mit denen diese Problemstellungen gelöst werden können.

Der Viterbi-Algorithmus

Wir widmen uns zunächst dem ersten vorgestellten Problem, der Dekodierung. Zunächst müssen die Parameter des Modells, also hier sowohl die Übergangs- als auch die Emissionswahrscheinlichkeiten, aus Trainingsdaten geschätzt werden. Ein Algorithmus dafür wird später vorgestellt. Sind diese Parameter gegeben, können wir eine A-priori-Konnektivität bestimmen, wobei die zugrundeliegende Markovkette des Modells dann nichttrivial verbunden ist. Für eine Sequenz $x = x_1 \dots x_L$ von Buchstaben aus dem Alphabet $\mathcal{A}$ und einen Pfad $\pi = \pi_1 \dots \pi_L$ der selben Länge ist die *Wahrscheinlichkeit, dass x von π generiert wurde*, definiert als

$$\mathbb{P}(x, \pi) = p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0}. \tag{2}$$

Hier und im Folgenden schreiben wir $p_{\mathcal{B}\pi_1} := p_{01}$ und $p_{\pi_L\mathcal{E}} := p_{\pi_L 0}$. Tatsächlich kann man zeigen, dass die Häufigkeit, mit der x von π generiert wird, unter allen Sequenzen, die auf die Länge L gekürzt worden sind, gegen diesen Wert konvergiert.

Satz 1
Es gilt

$$\sum_{(x,\pi)\in S_e} \mathbb{P}(x, \pi) = 1$$

mit S_e = Menge aller Sequenzpaare aller endlichen Längen.

Beweis

$$\begin{aligned}
&\sum_{(x,\pi)\in S_e} \mathbb{P}(x, \pi) = \sum_{L=1}^{\infty} \sum_{(x,\pi)\in S_L} \mathbb{P}(x, \pi) \\
\overset{(2)}{=} &\sum_{L=1}^{\infty} \sum_{(x,\pi)\in S_L} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
= &\sum_{L=1}^{\infty} \sum_{\pi_1,\ldots,\pi_L\in\mathcal{Z}} \sum_{x_1,\ldots,x_L\in\mathcal{A}} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
\overset{(1)}{=} &\sum_{L=1}^{\infty} \sum_{\pi_1,\ldots,\pi_L\in\mathcal{Z}} \sum_{x_1,\ldots,x_{L-1}\in\mathcal{A}} p_{0\pi_1} q_{\pi_1}(x_1) \cdot \ldots \cdot p_{\pi_{L-2}\pi_{L-1}} q_{\pi_{L-1}}(x_{L-1}) p_{\pi_{L-1}\pi_L} p_{\pi_L 0}
\end{aligned}$$

$$= \sum_{L=1}^{\infty} \sum_{\pi_1,\ldots,\pi_L \in \mathcal{Z}} \sum_{x_1,\ldots,x_{L-2} \in \mathcal{A}} p_{0\pi_1} q_{\pi_1}(x_1) \cdot \ldots \cdot p_{\pi_{L-3}\pi_{L-2}} q_{\pi_{L-2}}(x_{L-2}) p_{\pi_{L-2}\pi_{L-1}} p_{\pi_{L-1}\pi_L} p_{\pi_L 0}$$

$$= \ldots = \sum_{L=1}^{\infty} \sum_{\pi_1,\ldots,\pi_L \in \mathcal{Z}} p_{0\pi_1} p_{\pi_1\pi_2} \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} p_{\pi_L 0} = 1$$

Hierbei bezeichnet S_L die Menge aller Sequenzpaare mit Länge L. Die letzte Gleichheit gilt nach Satz 7.7 des Vortrags über Markovketten.

□

Wie finden wir jetzt aber einen *wahrscheinlichen* Pfad für x und was bedeutet das eigentlich genau? Eine Idee ist, einen Pfad π^* zu finden, der $\mathbb{P}(x, \pi)$ unter allen π (beachte: x und π haben beide Länge L) maximiert. π^* muss nicht eindeutig sein; es können mehrere solcher Pfade existieren. Wir werden sie auch *Viterbi-Pfade* nennen. Man könnte $\mathbb{P}(x, \pi)$ für alle möglichen Pfade ermitteln, was aber viel zu aufwändig und praktisch nicht möglich ist, da die Anzahl mit L exponentiell wächst. Der Viterbi-Algorithmus dagegen ist effizienter.

Gegeben ist ein Hidden Markov Model mit Zustandsraum $\mathcal{Z} = \{Z_1, \ldots, Z_N\}$, Übergangswahrscheinlichkeiten $p_{\mathcal{B}Z_j} := p_{0j},\ p_{Z_iZ_j} := p_{ij},\ p_{Z_j\mathcal{E}} := p_{j0},\ i, j = 1, \ldots, N$, und Emissionswahrscheinlichkeiten $q_{Z_j}(x_k) := q_j(x_k),\ j = 1, \ldots, N,\ k = 1, \ldots, L$.

Viterbi-Algorithmus

$$v_k(1) = p_{0k} q_k(x_1) \text{ für } k = 1, \ldots, N$$

$$v_k(i+1) = q_k(x_{i+1}) \max_{l=1,\ldots,N} v_l(i) p_{lk} \ \forall\ i = 1, \ldots, L-1,\ k = 1, \ldots, N \text{ mit}$$

$$v_k(i) = \max_{\pi_1,\ldots,\pi_{i-1} \in \mathcal{Z}} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{i-2}\pi_{i-1}} q_{\pi_{i-1}}(x_{i-1}) p_{\pi_{i-1}k} q_k(x_i)$$

$$\text{für } i = 2, \ldots, L,\ k = 1, \ldots, N$$

Man berechnet also Anfangswerte und ermittelt dann rekursiv $v_k(i+1)$. Dabei wird in jedem Schritt eine Menge $\mathcal{V}_k(i)$ erstellt, die alle ganzen Zahlen m mit

$$v_m(i) p_{mk} = \max_{l=1,\ldots,N} v_l(i) p_{lk}$$

für $i = 1, \ldots, L-1,\ k = 1, \ldots, N$ enthält. Außerdem gilt:

$$\max_{\pi \in P_L} \mathbb{P}(x, \pi) = \max_{l=1,\ldots,N} v_l(L) p_{l0}$$

mit P_L = Menge aller Pfade der Länge L.

Dann definieren wir $\mathcal{V}(L)$ als Menge, die alle ganzen Zahlen m enthält, für die

$$v_m(L)p_{m0} = \max_{l=1,\dots,N} v_l(L)p_{l0}$$

gilt. Jeden Viterbi-Pfad kann man nun folgendermaßen „rückverfolgen“: Wähle $m_L \in \mathcal{V}(L)$, dann $m_{L-1} \in \mathcal{V}_{m_L}(L-1)$, $m_{L-2} \in \mathcal{V}_{m_{L-1}}(L-2), \dots, m_1 \in \mathcal{V}_{m_2}(1)$. Für den daraus resultierenden Pfad $\pi^* = Z_{m_1} Z_{m_2} \dots Z_{m_L}$ gilt dann

$$\mathbb{P}(x, \pi^*) = \max_{\pi \in P_L} \mathbb{P}(x, \pi).$$

In der Praxis wird der Viterbi-Algorithmus in dem Programm „GENSCAN“ angewandt, um menschliche Gene zu dekodieren. Für nähere Informationen kann man die Website http://genes.mit.edu/GENSCAN.html besuchen.

Beispiel
Wir betrachten ein einfaches Beispiel mit Zustandsraum $\mathcal{Z} = \{Z_1, Z_2\}$ und Alphabet $\mathcal{A} = \{A, C, G, T\}$. Die zugehörige Markovkette hat folgende Gestalt:

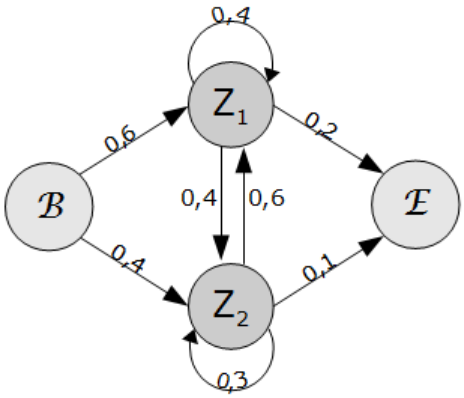

Abbildung 4: Graph der zum HMM gehörigen Markovkette

Die Pfeile entsprechen den Übergangswahrscheinlichkeiten. Die Emissionswahrscheinlichkeiten sind gegeben durch

$$q_1(A) = 0,3;\ q_1(C) = 0,25;\ q_1(G) = 0,25;\ q_1(T) = 0,2;$$
$$q_2(A) = 0,1;\ q_2(C) = 0,4;\ q_2(G) = 0,2;\ q_2(T) = 0,3.$$

Die beobachtete Sequenz lautet $x = GATTACA$.

Wir beginnen damit, die Anfangswerte zu berechnen:

$$v_1(1) = p_{01}q_1(G) = 0,6 \cdot 0,25 = 0,15$$
$$v_2(1) = p_{02}q_2(G) = 0,4 \cdot 0,2 = 0,08$$

Aus diesen ermitteln wir nun die Werte für $i = 1$:

$$v_1(2) = q_1(A) \cdot \max_{l=1,2}(v_l(1)p_{l1}) = 0,3 \cdot \max(0,06;\ 0,048) = 0,3 \cdot 0,06 = 0,018$$

$\Rightarrow \mathcal{V}_1(1) = \{1\}$, denn das Maximum wird nur für $l = 1$ erreicht.

$$v_2(2) = q_2(A) \cdot \max_{l=1,2}(v_l(1)p_{l2}) = 0,1 \cdot \max(0,06;\ 0,024) = 0,1 \cdot 0,06 = 0,006$$

$\Rightarrow \mathcal{V}_2(1) = \{1\}$.

Weiter berechnet man:

$$\begin{aligned}
v_1(3) &= 0,2 \cdot \max(0,0072;\ 0,0036) = 0,00144 && \Rightarrow \mathcal{V}_1(2) = \{1\}\\
v_2(3) &= 0,3 \cdot \max(0,0072;\ 0,0018) = 0,00216 && \Rightarrow \mathcal{V}_2(2) = \{1\}\\
v_1(4) &= 0,2 \cdot \max(0,000576;\ 0,001296) = 0,0002592 && \Rightarrow \mathcal{V}_1(3) = \{2\}\\
v_2(4) &= 0,3 \cdot \max(0,000576;\ 0,000648) = 0,0001944 && \Rightarrow \mathcal{V}_2(3) = \{2\}\\
v_1(5) &= 0,3 \cdot \max(0,00010368;\ 0,00011664) = 0,000034992 && \Rightarrow \mathcal{V}_1(4) = \{2\}\\
v_2(5) &= 0,1 \cdot \max(0,00010368;\ 0,00005832) = 0,000010368 && \Rightarrow \mathcal{V}_2(4) = \{1\}\\
v_1(6) &= 0,25 \cdot \max(0,000013997;\ 0,000006221) = 0,000003499 && \Rightarrow \mathcal{V}_1(5) = \{1\}\\
v_2(6) &= 0,4 \cdot \max(0,000013997;\ 0,00000311) = 0,000005599 && \Rightarrow \mathcal{V}_2(5) = \{1\}\\
v_1(7) &= 0,3 \cdot \max(0,0000014;\ 0,000003359) = 0,000001008 && \Rightarrow \mathcal{V}_1(6) = \{2\}\\
v_2(7) &= 0,1 \cdot \max(0,0000014;\ 0,00000168) = 0,000000168 && \Rightarrow \mathcal{V}_2(6) = \{2\}
\end{aligned}$$

$$\Rightarrow \max_{\pi \in P_7} \mathbb{P}(x, \pi) = \max_{l=1,2}(v_l(7)p_{l0}) = \max(0,000000202;\ 0,000000017) = 0,000000202$$

$\Rightarrow \mathcal{V}(7) = \{1\}$

$\Rightarrow m_7 = 1,\ m_6 = 2,\ m_5 = 1,\ m_4 = 2,\ m_3 = 2,\ m_2 = 1,\ m_1 = 1$

$\Rightarrow \pi^* = Z_1 Z_1 Z_2 Z_2 Z_1 Z_2 Z_1$.

Die Ergebnisse sind auf neun Stellen nach dem Komma gerundet. Hier liegt der Spezialfall vor, dass die Mengen $\mathcal{V}_k(i)$ und $\mathcal{V}(L)$ jeweils nur ein Element enthalten. Daher gibt es auch einen eindeutigen Viterbi-Pfad.

Der Forward-Algorithmus

Dieser Algorithmus ist dem Viterbi-Algorithmus sehr ähnlich. Gesucht ist hier jedoch die Gesamtwahrscheinlichkeit

$$\mathbb{P}(x) = \sum_{\pi \in P_L} \mathbb{P}(x, \pi) \tag{3}$$

für eine gegebene Sequenz $x = x_1 \dots x_L$. In diesem Fall sind Pfade also nicht bekannt und wir beschäftigen uns mit Problemstellung 2.

Satz 2
Es gilt

$$\sum_{x \in M_e} \mathbb{P}(x) = 1,$$

wobei M_e die Menge aller Sequenzen aller endlichen Längen ist.

Beweis
Es gilt

$$\sum_{x \in M_e} \mathbb{P}(x) = \sum_{L=1}^{\infty} \sum_{x \in M_L} \mathbb{P}(x) \overset{(3)}{=} \sum_{L=1}^{\infty} \sum_{x \in M_L} \sum_{\pi \in P_L} \mathbb{P}(x, \pi) = 1,$$

wobei die letzte Gleichheit direkt aus Satz 1 folgt und M_L die Menge aller Sequenzen der Länge L bezeichnet.

□

Theoretisch könnte man alle möglichen Pfade der Länge L ermitteln, jeweils $\mathbb{P}(x, \pi)$ berechnen und dann aufsummieren, aber praktisch ist der Aufwand natürlich viel zu hoch. Der Forward-Algorithmus berechnet $\mathbb{P}(x)$ effizienter.

Forward-Algorithmus

$$f_k(1) = p_{0k} q_k(x_1) \text{ für } k = 1, \dots, N$$

$$f_k(i+1) = q_k(x_{i+1}) \sum_{l=1}^{N} f_l(i) p_{lk} \ \forall\ i = 1, \dots, L-1,\ k = 1, \dots, N \text{ mit}$$

$$f_k(i) = \sum_{\pi_1, \dots, \pi_{i-1} \in \mathcal{Z}} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1 \pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{i-2} \pi_{i-1}} q_{\pi_{i-1}}(x_{i-1}) p_{\pi_{i-1} k} q_k(x_i)$$

$$\text{für } i = 2, \dots, L,\ k = 1, \dots, N$$

Auch hier berechnet man zunächst die Anfangswerte und dann rekursiv $f_k(i+1)$. Ist schließlich der Wert $f_k(L)$ bestimmt, kann $\mathbb{P}(x)$ direkt durch

$$\mathbb{P}(x) = \sum_{k=1}^{N} f_k(L) p_{k0}$$

ermittelt werden. Der numerische Aufwand durch das Speichern der Werte für die $\mathcal{V}_k(i)$ entfällt hier. Seinen Namen hat der vorgestellte Algorithmus daher, dass er die Sequenz x *vorwärts* einliest.

Beispiel
Wir betrachten wieder das Hidden Markov Model aus dem Beispiel für den Viterbi-Algorithmus mit der beobachteten Sequenz $x = GATTACA$. Wir erhalten folgende Anfangswerte:

$$f_1(1) = p_{01} q_1(G) = 0,6 \cdot 0,25 = 0,15$$
$$f_2(1) = p_{02} q_2(G) = 0,4 \cdot 0,2 = 0,08$$

Nun folgen die Rekursionsschritte:

$$\begin{aligned} f_1(2) &= q_1(A) \sum_{l=1}^{2} f_l(1) p_{l1} = q_1(A) \cdot (f_1(1)p_{11} + f_2(1)p_{21}) \\ &= 0,3 \cdot (0,15 \cdot 0,4 + 0,08 \cdot 0,6) = 0,0324 \\ f_2(2) &= q_2(A) \sum_{l=1}^{2} f_l(1) p_{l2} = q_1(A) \cdot (f_1(1)p_{12} + f_2(1)p_{22}) \\ &= 0,1 \cdot (0,15 \cdot 0,4 + 0,08 \cdot 0,3) = 0,0084 \end{aligned}$$

$$f_1(3) = q_1(T) \cdot (f_1(2)p_{11} + f_2(2)p_{21}) = 0,2 \cdot (0,0324 \cdot 0,4 + 0,0084 \cdot 0,6) = 0,0036$$
$$f_2(3) = q_2(T) \cdot (f_1(2)p_{12} + f_2(2)p_{22}) = 0,3 \cdot (0,0324 \cdot 0,4 + 0,0084 \cdot 0,3) = 0,004644$$
$$f_1(4) = q_1(T) \cdot (f_1(3)p_{11} + f_2(3)p_{21}) = 0,2 \cdot (0,0036 \cdot 0,4 + 0,004644 \cdot 0,6) = 0,00084528$$
$$f_2(4) = q_2(T) \cdot (f_1(3)p_{12} + f_2(3)p_{22}) = 0,3 \cdot (0,0036 \cdot 0,4 + 0,004644 \cdot 0,3) = 0,00084996$$
$$f_1(5) = q_1(A) \cdot (f_1(4)p_{11} + f_2(4)p_{21}) = 0,000254426$$
$$f_2(5) = q_2(A) \cdot (f_1(4)p_{12} + f_2(4)p_{22}) = 0,00005931$$
$$f_1(6) = q_1(C) \cdot (f_1(5)p_{11} + f_2(5)p_{21}) = 0,000034339$$
$$f_2(6) = q_2(C) \cdot (f_1(5)p_{12} + f_2(5)p_{22}) = 0,000047825$$
$$f_1(7) = q_1(A) \cdot (f_1(6)p_{11} + f_2(6)p_{21}) = 0,000012729$$
$$f_2(7) = q_2(A) \cdot (f_1(6)p_{12} + f_2(6)p_{22}) = 0,000002802$$

$$\begin{aligned} \Rightarrow \mathbb{P}(x) &= \sum_{k=1}^{2} f_k(7) p_{k0} = f_1(7)p_{10} + f_2(7)p_{20} \\ &= 0,000012729 \cdot 0,2 + 0,000002808 \cdot 0,1 = 0,000002827 \end{aligned}$$

Der Backward-Algorithmus

Dieser Algorithmus bestimmt ebenso wie der gerade vorgestellte Forward-Algorithmus die Gesamtwahrscheinlichkeit $\mathbb{P}(x)$. Wie der Name schon verrät, wird die Sequenz $x = x_1 \dots x_L$ diesmal rückwärts eingelesen. Aber warum gibt es zwei Algorithmen für die Ermittlung der gleichen Wahrscheinlichkeit? Wir werden später sehen, dass ein Algorithmus, welcher wie der Viterbi-Algorithmus beim Dekodieren eingesetzt wird, sowohl vom Forward- als auch vom Backward-Algorithmus Gebrauch macht.

Backward-Algorithmus

$$b_k(L) = p_{k0} \text{ für } k = 1, \dots, N$$

$$b_k(i) = \sum_{l=1}^{N} p_{kl} q_l(x_{i+1}) b_l(i+1) \text{ für } i = 1, \dots, L-1,\ k = 1, \dots, N \text{ mit}$$

$$b_k(i) = \sum_{\pi_{i+1}, \dots, \pi_L \in \mathcal{Z}} p_{k\pi_{i+1}} q_{\pi_{i+1}}(x_{i+1}) p_{\pi_{i+1}\pi_{i+2}} q_{\pi_{i+2}}(x_{i+2}) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0}$$

$$\text{für } i = 1, \dots, L-1,\ k = 1, \dots, N$$

Zunächst werden wieder die Anfangswerte $b_k(L)$ berechnet und anschließend werden die Werte $b_k(L-1), \dots, b_k(1)$ mit der Rekursionsformel bestimmt. Damit erhält man dann

$$\mathbb{P}(x) = \sum_{k=1}^{N} p_{0k} q_k(x_1) b_k(1).$$

Beispiel
Gegeben sind wieder die selben Daten wie bei den vorherigen Beispielen mit $x = GATTACA$. Die Anfangswerte sind:

$$b_1(7) = p_{10} = 0,2$$
$$b_2(7) = p_{20} = 0,1$$

Die Rekursionsschritte erfolgen ebenfalls „rückwärts“:

$$\begin{aligned} b_1(6) &= \sum_{l=1}^{2} p_{1l} q_l(A) b_l(7) = p_{11} q_1(A) b_1(7) + p_{12} q_2(A) b_2(7) \\ &= 0,4 \cdot 0,3 \cdot 0,2 + 0,4 \cdot 0,1 \cdot 0,1 = 0,024 + 0,004 = 0,028 \\ b_2(6) &= \sum_{l=1}^{2} p_{2l} q_l(A) b_l(7) = p_{21} q_1(A) b_1(7) + p_{22} q_2(A) b_2(7) \\ &= 0,6 \cdot 0,3 \cdot 0,2 + 0,3 \cdot 0,1 \cdot 0,1 = 0,036 + 0,003 = 0,039 \end{aligned}$$

$$\begin{aligned} b_1(5) &= p_{11} q_1(C) b_1(6) + p_{12} q_2(C) b_2(6) = 0,0028 + 0,00624 = 0,00904 \\ b_2(5) &= p_{21} q_1(C) b_1(6) + p_{22} q_2(C) b_2(6) = 0,0042 + 0,00468 = 0,00888 \\ b_1(4) &= p_{11} q_1(A) b_1(5) + p_{12} q_2(A) b_2(5) = 0,0010848 + 0,0003552 = 0,00144 \\ b_2(4) &= p_{21} q_1(A) b_1(5) + p_{22} q_2(A) b_2(5) = 0,0016272 + 0,0002664 = 0,0018936 \\ b_1(3) &= p_{11} q_1(T) b_1(4) + p_{12} q_2(T) b_2(4) = 0,0001152 + 0,000227232 = 0,000342432 \\ b_2(3) &= p_{21} q_1(T) b_1(4) + p_{22} q_2(T) b_2(4) = 0,0001728 + 0,000170424 = 0,000343224 \\ b_1(2) &= p_{11} q_1(T) b_1(3) + p_{12} q_2(T) b_2(3) = 0,000027395 + 0,000041187 = 0,000068582 \\ b_2(2) &= p_{21} q_1(T) b_1(3) + p_{22} q_2(T) b_2(3) = 0,000041092 + 0,00003089 = 0,000071982 \\ b_1(1) &= p_{11} q_1(A) b_1(2) + p_{12} q_2(A) b_2(2) = 0,00000823 + 0,000002879 = 0,000011109 \\ b_2(1) &= p_{21} q_1(A) b_1(2) + p_{22} q_2(A) b_2(2) = 0,000012345 + 0,000002159 = 0,000014504 \end{aligned}$$

$$\begin{aligned} \Rightarrow \mathbb{P}(x) &= \sum_{k=1}^{2} p_{0k} q_k(G) b_k(1) = p_{01} q_1(G) b_1(1) + p_{02} q_2(G) b_2(1) \\ &= 0,000001666 + 0,00000116 = 0,000002826 \end{aligned}$$

Wie erwartet ist dies die gleiche Wahrscheinlichkeit wie beim Forward-Algorithmus, wobei sich die letzte Ziffer auf Grund von Rundungsfehlern um 1 unterscheidet.

A-posteriori-Dekodierung

Wie schon angekündigt wird hier sowohl der Forward- als auch der Backward-Algorithmus angewandt. Zunächst wiederholen wir einige Begriffe aus der Wahrscheinlichkeitstheorie speziell für Hidden Markov Models.

Gegeben ist ein Hidden Markov Model mit nichttrivial verbundener Markovkette. Wir definieren den *Ergebnisraum* S als Menge, die alle Paare (y, π), die aus dem Hidden Markov Model hervorgehen können, enthält, wobei y eine endliche Sequenz von Buchstaben aus dem Alphabet $\mathcal{A}$ und π ein Pfad der selben Länge ist. Teilmengen von S nennen wir *Ereignisse*. Wir sagen, ein Ereignis E *tritt* zu einer bestimmten Zeit *ein*, wenn das Element aus dem Ergebnisraum, welches zu diesem Zeitpunkt von dem Hidden Markov Model erzeugt wurde, in E liegt. Die *Wahrscheinlichkeit (des Eintritts) eines Ereignisses* E ist definiert durch

$$\mathbb{P}(E) = \sum_{(y,\pi)\in E} \mathbb{P}(y, \pi).$$

Die *bedingte Wahrscheinlichkeit* des Ereignisses E_1 gegeben E_2 mit $\mathbb{P}(E_2) > 0$ ist

$$\mathbb{P}(E_1|E_2) = \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)}. \tag{4}$$

Betrachte nun für eine feste Sequenz $x = x_1 \dots x_L$ das Ereignis

$$E(x) = \{(y, \pi) \in S : \ y = x\}.$$

Es sagt aus, dass die Sequenz x vom Hidden Markov Model ausgegeben wird. Es gilt

$$\mathbb{P}(E(x)) = \mathbb{P}(x). \tag{5}$$

Betrachte außerdem das Ereignis

$$E_{i,k} = \{(y, \pi) \in S : \text{ Die Länge von } y \text{ und } \pi \text{ ist größer oder gleich } i \text{ und } \pi_i = Z_k\}$$

für feste $1 \leq i \leq L$ und $1 \leq k \leq N$. Anschaulich bedeutet es, dass der Zustand Z_k an der i-ten Stelle der Sequenz den Buchstaben y_i emittiert.

Satz 3
Wir nehmen an, dass $\mathbb{P}(x) > 0$ ist. Dann gilt

$$\mathbb{P}(E_{i,k}|E(x)) = \frac{f_k(i)b_k(i)}{\mathbb{P}(x)}$$

für $i = 1, \dots, L$ und $k = 1, \dots, N$.

Beweis

$$\mathbb{P}(E_{i,k}|E(x)) \overset{(4)}{=} \frac{\mathbb{P}(E_{i,k} \cap E(x))}{\mathbb{P}(E(x))} \overset{(5)}{=} \frac{\mathbb{P}(E_{i,k} \cap E(x))}{\mathbb{P}(x)}.$$

1. Fall: $2 \leq i \leq L-1$

$$\begin{aligned}
&\mathbb{P}(E_{i,k} \cap E(x)) \\
=&\sum_{\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_L \in \mathcal{Z}} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \\
&\cdot p_{\pi_{i-2}\pi_{i-1}} q_{\pi_{i-1}}(x_{i-1}) p_{\pi_{i-1}k} q_k(x_i) p_{k\pi_{i+1}} q_{\pi_{i+1}}(x_{i+1}) p_{\pi_{i+1}\pi_{i+2}} q_{\pi_{i+2}}(x_{i+2}) \cdot \ldots \\
&\cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
=&\sum_{\pi_1,\ldots,\pi_{i-1} \in \mathcal{Z}} p_{0\pi_1} q_{\pi_1}(x_1) p_{\pi_1\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{i-2}\pi_{i-1}} q_{\pi_{i-1}}(x_{i-1}) p_{\pi_{i-1}k} q_k(x_i) \\
&\cdot \sum_{\pi_{i+1},\ldots,\pi_L \in \mathcal{Z}} p_{k\pi_{i+1}} q_{\pi_{i+1}}(x_{i+1}) p_{\pi_{i+1}\pi_{i+2}} q_{\pi_{i+2}}(x_{i+2}) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
=&f_k(i) b_k(i).
\end{aligned}$$

2. Fall: $i = 1$

$$\begin{aligned}
&\mathbb{P}(E_{1,k} \cap E(x)) \\
=&\sum_{\pi_2,\ldots,\pi_L \in \mathcal{Z}} p_{0k} q_k(x_1) p_{k\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
=&p_{0k} q_k(x_1) \cdot \sum_{\pi_2,\ldots,\pi_L \in \mathcal{Z}} p_{k\pi_2} q_{\pi_2}(x_2) \cdot \ldots \cdot p_{\pi_{L-1}\pi_L} q_{\pi_L}(x_L) p_{\pi_L 0} \\
=&f_k(1) b_k(1).
\end{aligned}$$

3. Fall: $i = L$

$$\begin{aligned}
&\mathbb{P}(E_{L,k} \cap E(x)) \\
=&\sum_{\pi_1,\ldots,\pi_{L-1}} (p_{0\pi_1} q_{\pi_1}(x_1) \cdot \ldots \cdot p_{\pi_{L-2}\pi_{L-1}} q_{\pi_{L-1}}(x_{L-1}) p_{\pi_{L-1}k} q_k(x_L)) \cdot p_{k0} \\
=&f_k(L) b_k(L).
\end{aligned}$$

□

Die Wahrscheinlichkeit $\mathbb{P}(E_{i,k} \cap E(x))$ bedeutet anschaulich, dass die Sequenz x ausgegeben wird und $\pi_i = Z_k$ x_i emittiert. $f_k(i)$ wird mit dem Forward-Algorithmus und $b_k(i)$ mit dem Backward-Algorithmus berechnet. Die Gesamtwahrscheinlichkeit $\mathbb{P}(x)$ kann mit einem von den beiden Algorithmen bestimmt werden. $\mathbb{P}(E_{i,k}|E(x))$ ist anschaulich die Wahrscheinlichkeit, mit der die i-te Komponente x_i von x im Zustand Z_k emittiert wird, und heißt *A-posteriori-Wahrscheinlichkeit des Zustands Z_k bei Beobachtung i gegeben x.*

Neben dem Viterbi-Algorithmus gibt es noch weitere Verfahren der Dekodierung. Nun wollen wir die gerade kennen gelernten A-posteriori-Wahrscheinlichkeiten dazu benutzen, einen weiteren Algorithmus vorzustellen. Wir definieren für jedes $i = 1, \ldots, L$ die Menge

$$B(i) = \left\{ Z_m \in \mathcal{Z} : \ \mathbb{P}(E_{i,m}|E(x)) = \max_{k=1,\ldots,N} \mathbb{P}(E_{i,k}|E(x)) \right\}.$$

Die Elemente von $B(i)$ sind die *wahrscheinlichsten Zustände für* x_i mit $i = 1, \ldots, L$. Den Vorgang, die Menge $B(i)$ für alle i zu ermitteln, nennt man *A-posteriori-Dekodierung*. Möchte man nicht nur die wahrscheinlichsten Zustände, sondern auch wie beim Viterbi-Algorithmus die wahrscheinlichsten Pfade für eine gegebene Sequenz x finden, sucht man alle Pfade $\hat{\pi}$ der Länge L mit $\hat{\pi} \in B(i)$ für $i = 1, \ldots, L$, wobei die Wahrscheinlichkeiten $\mathbb{P}(x, \hat{\pi})$ sehr klein oder sogar Null werden können.

Beispiele
Betrachte erneut das Hidden Markov Model aus den vorherigen Beispielen und die ausgegebene Sequenz $x = GATTACA$. Wir bestimmen nun die A-posteriori-Wahrscheinlichkeiten $\mathbb{P}(E_{i,k}|E(x))$ für $i = 1, \ldots, 7,\ k = 1, 2$. Die benötigten Werte bekommen wir aus den Berechnungen in den Beispielen zum Forward- und Backward-Algorithmus. Wir verwenden außerdem den Wert $0,000002827$ für $\mathbb{P}(x)$ aus dem Forward-Algorithmus. Dann erhalten wir:

$$\mathbb{P}(E_{1,1}|E(x)) = \frac{f_1(1)b_1(1)}{\mathbb{P}(x)} = \frac{0,15 \cdot 0,000011109}{0,000002827} = 0,589441104,$$
$$\mathbb{P}(E_{1,2}|E(x)) = \frac{f_2(1)b_2(1)}{\mathbb{P}(x)} = \frac{0,08 \cdot 0,000014504}{0,000002827} = 0,410442165,$$
$$\mathbb{P}(E_{2,1}|E(x)) = \frac{f_1(2)b_1(2)}{\mathbb{P}(x)} = 0,78601231,\ \mathbb{P}(E_{2,2}|E(x)) = \frac{f_2(2)b_2(2)}{\mathbb{P}(x)} = 0,213883551,$$
$$\mathbb{P}(E_{3,1}|E(x)) = \frac{f_1(3)b_1(3)}{\mathbb{P}(x)} = 0,436064804,\ \mathbb{P}(E_{3,2}|E(x)) = \frac{f_2(3)b_2(3)}{\mathbb{P}(x)} = 0,56382464,$$
$$\mathbb{P}(E_{4,1}|E(x)) = \frac{f_1(4)b_1(4)}{\mathbb{P}(x)} = 0,430563566,\ \mathbb{P}(E_{4,2}|E(x)) = \frac{f_2(4)b_2(4)}{\mathbb{P}(x)} = 0,569325878,$$
$$\mathbb{P}(E_{5,1}|E(x)) = \frac{f_1(5)b_1(5)}{\mathbb{P}(x)} = 0,813587209,\ \mathbb{P}(E_{5,2}|E(x)) = \frac{f_2(5)b_2(5)}{\mathbb{P}(x)} = 0,186300955,$$
$$\mathbb{P}(E_{6,1}|E(x)) = \frac{f_1(6)b_1(6)}{\mathbb{P}(x)} = 0,340110364,\ \mathbb{P}(E_{6,2}|E(x)) = \frac{f_2(6)b_2(6)}{\mathbb{P}(x)} = 0,659771843,$$
$$\mathbb{P}(E_{7,1}|E(x)) = \frac{f_1(7)b_1(7)}{\mathbb{P}(x)} = 0,900530598,\ \mathbb{P}(E_{7,2}|E(x)) = \frac{f_2(7)b_2(7)}{\mathbb{P}(x)} = 0,09911567.$$

$\Rightarrow B(1) = \{Z_1\}$, denn für $k = 1$ wird $\mathbb{P}(E_{1,k}|E(x))$ maximal.

Weiter ist $B(2) = \{Z_1\}$, $B(3) = \{Z_2\}$, $B(4) = \{Z_2\}$, $B(5) = \{Z_1\}$, $B(6) = \{Z_2\}$ und $B(7) = \{Z_1\}$.

$\Rightarrow \hat{\pi} = Z_1Z_1Z_2Z_2Z_1Z_2Z_1$. Dies ist genau jener Pfad, den wir mit dem Viterbi-Algorithmus bestimmt haben.

Betrachten wir ein weiteres Beispiel, das das Vorgehen in der Praxis mit Hilfe der bisher vorgestellten Algorithmen verdeutlicht. Gegeben ist die Markovkette

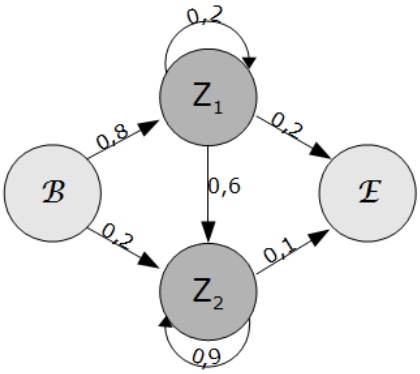

Abbildung 5: Beispiel-Markovkette

sowie das Alphabet $\mathcal{A} = \{A, B\}$ und die Emissionswahrscheinlichkeiten

$$q_1(A) = 0,5;\ q_1(B) = 0,5;\ q_2(A) = 0,9;\ q_2(B) = 0,1.$$

Um zu entscheiden, ob eine beobachtete Sequenz x zu einem eukaryotischen Gen gehört oder nicht, wird die Gesamtwahrscheinlichkeit $\mathbb{P}(x)$ berechnet. Wenn sie größer als ein vorgegebener Schwellenwert ist, wird die Sequenz dem Gen zugeordnet, andernfalls nicht. In diesem Beispiel gilt

$$\mathbb{P}(x) \begin{cases} \geq 0,02 & \Rightarrow x \text{ wird als Teil des eukaryotischen Gens akzeptiert;} \\ < 0,02 & \Rightarrow x \text{ wird nicht als Teil des eukaryotischen Gens akzeptiert.} \end{cases}$$

Die beobachteten Sequenzen sind $y^1 = ABB$ und $y^2 = BAA$. Werden sie als Genbestandteile akzeptiert?

Dafür berechnen wir zunächst $\mathbb{P}(y^1)$ mit dem Forward-Algorithmus. Wie die Werte bestimmt werden, haben wir in einem Beispiel bereits gesehen und daher geben wir nur die Gesamtwahrscheinlichkeit an. Es gilt

$$\mathbb{P}(y^1) = 0,0014018 < 0,02 \Rightarrow y^1 \text{ wird dem Gen nicht zugeordnet.}$$

Als Gesamtwahrscheinlichkeit für y^2 bekommen wir mit dem Forward-Algorithmus

$$\mathbb{P}(y^2) = 0,0217682 > 0,02 \Rightarrow y^2 \text{ wird als Teil des eukaryotischen Gens akzeptiert.}$$

Wie lauten die wahrscheinlichsten Pfade $\pi^*(y^1)$ und $\pi^*(y^2)$? Diese Frage können wir mit Hilfe des Viterbi-Algorithmus beantworten. Auch dazu wurde schon ein Beispiel vorgerechnet. Wir bekommen $\pi^*(y^1) = Z_1Z_1Z_1$ und $\pi^*(y^1) = Z_1Z_2Z_2$.

Parameterschätzung

Eine Familie von Hidden Markov Models wird von einer nichttrivial verbundenen Markovkette mit gegebener A-priori-Konnektivität durch die Übergangs- und Emissionswahrscheinlichkeiten parametrisiert. Wie können diese Parameter am besten geschätzt werden? Zunächst benötigt man eine Menge von Trainingsdaten. Diese besteht entweder aus Sequenzpaaren $(x^1, \pi^1), \ldots, (x^n, \pi^n)$, wobei x^j eine endliche Sequenz von Buchstaben aus dem Alphabet $\mathcal{A}$ und π^j ein Pfad der selben Länge für $j = 1, \ldots, n$ ist, oder nur aus Sequenzen $x^1, \ldots, x^n$. Gesucht sind Parameterwerte, für die $\mathbb{P}(x^1, \pi^1) \cdot \ldots \cdot \mathbb{P}(x^n, \pi^n)$ bzw. $\mathbb{P}(x^1) \cdot \ldots \cdot \mathbb{P}(x^n)$ maximal wird. Das daraus hervorgehende Hidden Markov Model *modelliert* dann die Trainingsdaten.

Wir betrachten hier nur den einfacheren, aber leider auch unrealistischen Fall, dass Sequenzpaare $(x^1, \pi^1), \ldots, (x^n, \pi^n)$ gegeben, die Pfade also bekannt sind. Ähnlich wie bei gewöhnlichen Markovketten schätzt man die Übergangs- und Emissionswahrscheinlichkeiten dann durch Bestimmung der Häufigkeiten. Die Trainingsdaten, genauer die Pfade, müssen wieder mit der A-priori-Konnektivität der zugrundeliegenden Markovkette übereinstimmen. Dann bezeichnen wir die Häufigkeit, mit der ein Zustand α in einen Zustand β übergeht, mit $H_{\alpha\beta}$ und die Häufigkeit, mit der ein Zustand den Buchstaben $a \in \mathcal{A}$ emittiert, mit $J_l(a)$, wobei $\alpha = \mathcal{B}, 1, \ldots, N$, $\beta = 1, \ldots, N, \mathcal{E}$ und $l = 1, \ldots, N$. Dann lassen sich die Übergangswahrscheinlichkeiten durch

$$p_{\alpha\beta} = \frac{H_{\alpha\beta}}{\sum\limits_{\gamma=1,\ldots,N,\mathcal{E}} H_{\alpha\gamma}} \tag{6}$$

bestimmen und die Emissionswahrscheinlichkeiten sind gegeben durch

$$q_l(a) = \frac{J_l(a)}{\sum\limits_{b \in \mathcal{A}} J_l(b)} \,. \tag{7}$$

Die Markovkette des resultierenden Hidden Markov Models ist dann nichttrivial verbunden. Wenn ein Zustand in den Trainingsdaten gar nicht vorkommt, ist es klar, dass die Wahrscheinlichkeiten, die diesen Zustand betreffen, auch nicht geschätzt werden können. Sollte dieser ungünstige Fall dennoch auftreten, werden sie beliebig gewählt.

Beispiel
Wir haben bereits erwähnt, dass das Two-Block-Modell zu viele Parameter enthält, um in der Praxis kostengünstig angewandt zu werden. Dennoch möchten wir gerne eukaryotische Gene modellieren und Exons und Introns unterscheiden. Dazu verwenden wir ein Hidden Markov Model mit folgender Markovkette:

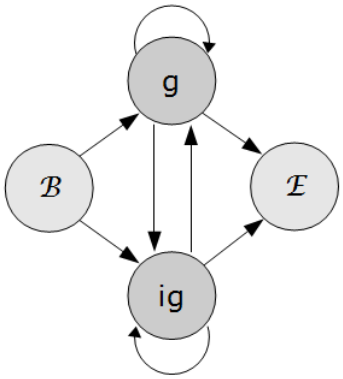

Abbildung 6: A-priori-Konnektivität der Markovkette

Auch hier steht „g“ wieder für „gene“ und „ig“ für „intergenic“. Es werden die Buchstaben des DNA-Alphabets $\mathcal{A} = \{A, C, G, T\}$ emittiert. Gegeben sind folgende Trainingsdaten:

$$
\begin{aligned}
x^1 &: \; ATG|ACT|\mathbf{ATG}|\mathbf{CTA}|\mathbf{TTG}|\mathbf{ATT}|\mathbf{TAA}|CGC \\
x^2 &: \; CCC|ATG|\mathbf{GTG}|\mathbf{AAA}|\mathbf{GAC}|\mathbf{TTC}|\mathbf{TAA}|GAT \\
x^3 &: \; AAA|GTG|ACT|\mathbf{ATG}|\mathbf{CCC}|\mathbf{GAT}|\mathbf{GAA}|\mathbf{CGC}|\mathbf{TAG}|GAA \\
x^4 &: \; ATG|GAT|\mathbf{ATG}|\mathbf{AAG}|\mathbf{CAT}|\mathbf{GAT}|\mathbf{TAA}|CAT
\end{aligned}
$$

Die fett gedruckten Codons sind bereits als Gensequenzen identifiziert worden. Daher kennen wir die Pfade:

$$
\begin{aligned}
\pi^1 &: \; 0|ig\ ig\ ig|ig\ ig\ ig|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|ig\ ig\ ig|0 \\
\pi^2 &: \; 0|ig\ ig\ ig|ig\ ig\ ig|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|ig\ ig\ ig|0 \\
\pi^3 &: \; 0|ig\ ig\ ig|ig\ ig\ ig|ig\ ig\ ig|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|ig\ ig\ ig|0 \\
\pi^4 &: \; 0|ig\ ig\ ig|ig\ ig\ ig|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|g\ g\ g|ig\ ig\ ig|0
\end{aligned}
$$

Jetzt können wir die Übergangs- und Emissionswahrscheinlichkeiten mit Hilfe der Formeln (6) und (7) schätzen:

$$
\begin{aligned}
p_{0g} &= \frac{H_{0g}}{\sum_{\gamma=g,ig,0} H_{0\gamma}} = \frac{0}{4} = 0, & p_{0ig} &= \frac{H_{0ig}}{\sum_{\gamma=g,ig,0} H_{0\gamma}} = \frac{4}{4} = 1 \\
p_{g,g} &= \frac{H_{g,g}}{\sum_{\gamma=g,ig,0} H_{g\gamma}} = \frac{59}{63} \approx 0,94, & p_{g,ig} &= \frac{H_{g,ig}}{\sum_{\gamma=g,ig,0} H_{g\gamma}} = \frac{4}{63} \approx 0,06 \\
p_{ig,g} &= \frac{H_{ig,g}}{\sum_{\gamma=g,ig,0} H_{ig\gamma}} = \frac{4}{39} \approx 0,10, & p_{ig,ig} &= \frac{H_{ig,ig}}{\sum_{\gamma=g,ig,0} H_{ig\gamma}} = \frac{31}{39} \approx 0,79 \\
p_{g0} &= \frac{H_{g0}}{\sum_{\gamma=g,ig,0} H_{g\gamma}} = \frac{0}{63} = 0, & p_{ig0} &= \frac{H_{ig0}}{\sum_{\gamma=g,ig,0} H_{ig\gamma}} = \frac{4}{39} \approx 0,10
\end{aligned}
$$

$$q_g(A) = \frac{J_g(A)}{\sum_{a\in\mathcal{A}} J_g(a)} = \frac{23}{63} \approx 0,37, \qquad q_{ig}(A) = \frac{J_{ig}(A)}{\sum_{a\in\mathcal{A}} J_{ig}(a)} = \frac{13}{39} \approx 0,33$$

$$q_g(C) = \frac{J_g(C)}{\sum_{a\in\mathcal{A}} J_g(a)} = \frac{9}{63} \approx 0,14, \qquad q_{ig}(C) = \frac{J_{ig}(C)}{\sum_{a\in\mathcal{A}} J_{ig}(a)} = \frac{8}{39} \approx 0,21$$

$$q_g(G) = \frac{J_g(G)}{\sum_{a\in\mathcal{A}} J_g(a)} = \frac{13}{63} \approx 0,21, \qquad q_{ig}(G) = \frac{J_{ig}(G)}{\sum_{a\in\mathcal{A}} J_{ig}(a)} = \frac{9}{39} \approx 0,23$$

$$q_g(T) = \frac{J_g(T)}{\sum_{a\in\mathcal{A}} J_g(a)} = \frac{18}{63} \approx 0,29, \qquad q_{ig}(T) = \frac{J_{ig}(T)}{\sum_{a\in\mathcal{A}} J_{ig}(a)} = \frac{9}{39} \approx 0,23$$