KAPITEL 1

Hidden Markov Modell, Viterbi-, Forward- und Backward-Algorithmus

Ausarbeitung von Manuel Allhoff

1.1 Einleitung

Dieses Kapitel befasst sich mit dem Hidden Markov Modell und dem Viterbi-, Forward- und Backward-Algorithmus.

Dazu wird das Beispiel der CpG-Inseln vorgestellt und in einem Exkurs klar gemacht, wieso die CpG-Inseln wichtig für die Forschung sind. In diesem Zusammenhang wird gleichzeitig die Markov Kette verdeutlicht und mit ihrer Hilfe das HMM erläutert. Mit diesem Wissen werden dann der Viterbi-, der Forward-, und der Backward-Algorithmus erklärt.

1.2 CpG-Inseln

In der Bioinformatik ist es von Interesse, wichtige DNA-Teilstücke, die Promotorbereiche, zu erkennen. Diese Promotorbereiche sind häufig auch sogenannte CpG-Inseln. Sie dienen auf dem DNA-Strang als Markierung, um die Gen-Expression zu ermöglichen.

Bei dem Dinukleotid CG eines DNA Stranges ist das Cytosin häufig methyliert und kann somit relativ häufig in ein T mutieren. Somit tritt ein CG-Paar selten auf und widerspricht der intuitiven Annahme einer Gleichverteilung der Paare. Es gibt allerdings Regionen auf dem DNA-Strang, wo das CG Paar relativ häufig vorkommt. Dies ist wie oben beschrieben z.B. bei Promotorbereichen der Fall. Diese Teilstücke nennt man CpG-Inseln. Man schreibt „CpG“ statt „CG“ um die Bindung von C und G mit einer Wasserstoffbrücke zwischen den Strängen zu unterscheiden. Das Kürzel „CpG“ steht für „Cytosin-phosphatidyl-Guanosin“.

Die CpG-Inseln weisen also eine erhöhte CpG Dichte auf als andere Regionen auf dem DNA-Strang und haben eine Länge von ca. 1-2kb.

Exkurs: Bedeutung der CpG-Inseln Die Bedeutung der CpG-Inseln wird schnell klar, wenn man sie in Verbindung mit Krebserkrankungen bringt. Krebs ist in der Medizin als bösartiger Tumor definiert. Unter einem Tumor versteht man die unkontrollierte Neubildung von Gewebe im Körper. Ein bösärtiger Tumor dringt in das ihn umgebende Gewebe ein und zerstört es. Außerdem kann er sich über das Blut- und Lymphsystem im Körper verteilen und Tochtergeschwülste bilden.

Die unkontrollierte Neubildung von Gewebe ist auf defekte DNA-Teilstücke zurückzuführen, sodass Erbinformationen bei jeder DNA-Kopie verfälscht weitergegeben werden. Jeder Kopiervorgang beginnt bei einem Promotor, der wie oben beschrieben auch häufig eine CpG-Insel darstellt. Somit ist leicht ersichtlich, dass es für die Forschung unerlässlich ist, die CpG-Inseln in einem DNA-Strang zu finden (Lokalisierungsproblem) oder zu entscheiden, ob ein Teilstrang der DNA eine CpG-Insel ist oder nicht (Diskriminationsproblem).

1.3 Markov Kette

Da die CpG-Inseln nur eine größere Wahrscheinlichkeit für das Auftreten eines CG-Dinukleotids im Vergleich zum restlichen DNA-Strang beschreiben, kann man bei einem gegebenen DNA-Teilstück auch nicht sagen, ob es sich entweder um eine CpG-Insel handelt oder nicht. Stattdessen muss man eine Wahrscheinlichkeit festlegen, die beschreibt, wie groß die Chance ist, dass dieses Teilstück eine CpG-Insel ist oder nicht. Erst dann kann man eine Entscheidung treffen.

Somit kann man das Diskriminations-Problem wie folgt definieren:

Problem 1: Diskriminations-Problem

Gegeben:
Eine Zeichenkette x mit der Länge L über dem Alphabet $Z = \{A, C, G, T\}$

Gesucht:
Entscheidung, ob x eine CpG-Insel ist oder nicht.

Um dieses Problem zu lösen, muss man ein Modell finden, dass DNA-Sequenzen generieren kann, die entweder CpG-Inseln sind oder nicht. Die passenden Wahrscheinlichkeiten, z.B., dass in einer CpG-Insel ein G auf ein C folgt, sind empirisch aus DNA-Sequenzen gewonnen. Dazu hat man über eine lange Sequenz die Nukleotide gezählt und anschließend verschiedene Teilstücke miteinander verglichen. Erst mit diesem Verfahren hat man entdeckt, dass es so etwas wie CpG-Inseln überhaupt gibt.

Ein passendes Modell, dass diese Anforderungen erfüllt, ist die Markov Kette, die formal wie folgt definiert ist:

Definition 1 *Eine Markovsche Kette erster Ordnung (kurz: Markovsche Kette) ist ein stochastischer Prozess $X_1, X_2 \ldots$ mit Werten in einer höchstens abzählbaren Zustandsmenge Z, der die Markovsche Eigenschaft besitzt:*

Für jeden Zeitpunkt $t > 0$ und jeder Zustandsfolge $z_1, z_2, \ldots, z_t$ mit

$P(X_1 = z_1, X_2 = z_2, \ldots, X_{t-1} = z_{t-1}) > 0$ *ist*

$P(X_t = z_t | X_1 = z_1, X_2 = z_2, \ldots, X_{t-1} = z_{t-1}) = P(X_t = z_t | X_{t-1} = z_{t-1})$.

Die Wahrscheinlichkeiten $P(X_1 = s) = a_{Bs}$ heißen Startwahrscheinlichkeiten, die Wahrscheinlichkeiten $a_{st} = P(X_i = t | X_{i-1} = s)$ Übergangswahrscheinlichkeiten.

Eine Markov Kette kann man als Graph darstellen, wie man in Abbildung 1.1 sehen kann.

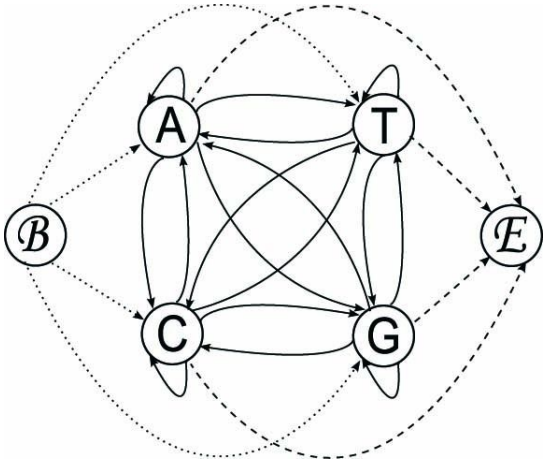

Abbildung 1.1: DNA-Markov Kette als Graph, angelehnt an: (Durbin et al., 1998, S. 50)

Grundlagen Mit dieser Markov Kette kann man eine DNA-Sequenz generieren, indem man sich einen Pfad von B nach E aussucht. Für die Zustandsmenge gilt $Z = \{A, C, G, T\} \cup \{B, E\}$, wenn die Markov-Kette beim Erreichen von E endet. Wenn die Markov-Kette nach einer festen Anzahl an Schritten endet, gilt: $Z = \{A, C, G, T\} \cup \{B\}$. Die Kanten entsprechen den Übergangswahrscheinlichkeiten, die durch

$$a_{st} = P(x_i = t | x_{i-1} = s)$$

bestimmt sind. Dabei entsprechen s und t Zuständen aus Z und x_i bzw. x_{i-1} einem Symbol aus der DNA-Sequenz. Die Übergangswahrscheinlichkeit a_{st} gibt also die Wahrscheinlichkeit an, dass dem Zustand s der Zustand t folgt.

Als Spezialfall ist hier zum einen die *Startwahrscheinlichkeit* a_{Bs} zu nennen. Sie gibt an bei welchem Symbol die Sequenz beginnt und es gilt:

$$a_{Bs} = P(x_1 = s).$$

Zum anderen die *Endwahrscheinlichkeit*, für die gilt: $a_{tE} = P(x_{L+1} = E|x_L = t),\ \forall\ L \geq 0$. Sie modelliert das Ende der Sequenz. Beide werden oft aus Gründen der Übersichtlichkeit einfach weggelassen.

Für jede Sequenz $x = (x_L, x_{L-1}, \ldots, x_1)$ fester Länge L kann man ihre Wahrscheinlichkeit wie folgt angeben:

$$P(x) = P(x_L, x_{L-1}, \ldots, x_1).$$

Diese Formel kann man durch mehrmalige Anwendung der Regel $P(X, Y) = P(X|Y)P(Y)$ umformen zu

$$P(x) = P(x_L|x_{L-1}, \ldots, x_1)P(x_{L-1}|x_{L-2}, \ldots, x_1) \ldots P(x_1).$$

Laut Definition der Markov Kette hängt der aktuelle Zustand nur von seinem Vorgängerzustand ab, und somit gilt:

$$P(x) = P(x_L|x_{L-1})P(x_{L-1}|x_{L-2}) \ldots P(x_2|x_1)P(x_1).$$

Wenn man dabei den Start- und Endzustand außer Acht lässt, was oft passiert, ist die Formel äquivalent zu:

$$P(x) = P(x_1) \prod_{i=2}^{L} a_{x_{i-1}x_i}.$$

Lösung des Diskriminations-Problems Mit diesem Wissen kann man zwei Markov Ketten erstellen, die entweder eine CpG-Insel (im folgenden „+ -Modell“ genannt), oder keine CpG-Insel („- -Modell“) generieren. Die wie oben beschrieben empirisch gewonnenen Übergangswahrscheinlichkeiten sind durch die Formel

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}}$$

bestimmt. Beim „- -Modell“ erfolgt die Berechnung analog. Die Variable c_{st}^{+} gibt an, wie oft von dem Zustand s in den Zustand t gewechselt wird. Dieser Wert wird anschließend durch die Summe aller Übergänge dividiert. So erhält man folgende Tabelle:

+	A	C	G	T	-	A	C	G	T
A	0,180	0,274	0,426	0,120	A	0,300	0,205	0,285	0,210
C	0,171	0,368	0,274	0,188	C	0,322	0,298	0,078	0,302
G	0,161	0,339	0,375	0,125	G	0,248	0,246	0,298	0,208
T	0,079	0,355	0,384	0,182	T	0,177	0,239	0,292	0,292

Tabelle 1.1: Übergangswahrscheinlichkeiten innerhalb und außerhalb von CpG-Inseln, Quelle: (Durbin et al., 1998, S. 51)

Man definiere nun eine Score-Funktion S:

$$S(x) = \log \frac{P(x|+)}{P(x|-)}$$

Sie berechnet den Quotienten von den Wahrscheinlichkeiten, dass die Zeichenkette x eine CpG-Insel ist bzw. nicht ist.

Es gilt:

$$S(x) = \sum_{i=1}^{L} \log \frac{a^+_{x_{i-1}x_i}}{a^-_{x_{i-1}x_i}},$$

da man für jeden String $x = (x_1, \ldots, x_L)$ mit fester Länge L den Quotienten abschnittsweise berechnet und die Werte addiert.

Man rechnet mit der log-Funktion, da man durch mehrmaliges Multiplizieren von Wahrscheinlichkeiten sehr schnell sehr kleine Zahlen als Ergebnis erhält, die bei der digitalen Berechnung zu einer Underflow-Exception führen. Die Funktion S ist intuitiv vernünftig gewählt, da eine CpG-Insel im „+ -Modell“ eine größere Wahrscheinlichkeit hat als im „- -Modell“. Der Score wird also positiv. Der umgekehrte Fall gilt analog. Somit gilt:

$$S(x) = \begin{cases} > 0, & \text{wenn } x \text{ CpG-Insel,} \\ < 0, & \text{wenn } x \text{ keine CpG-Insel,} \\ = 0, & \text{keine Aussage.} \end{cases}$$

1.4 Hidden Markov Modell

Definition 2 *Ein Hidden Markov Modell ist ein Quintupel $\lambda = (S, A, \pi_{Start}, B, V)$ mit:*

- *endlicher Menge S von Zuständen $\{s_1, \ldots, s_n\}$*
- *Matrix von Übergangswahrscheinlichkeiten $A = \Big(a_{ij} | a_{ij} = P(s_t = j | s_{t-1} = i)\Big)$*
- *Vektor π_{Start} von Zustandsstartwahrscheinlichkeiten*
- *Ausgabealphabet V*
- *Emissionswahrscheinlichkeiten $e_k(b) = P(x_i = b | \pi_i = k)$ in Form einer Matrix E mit $k \in S$ und $b \in V$*

Grundlagen In der bis jetzt beschriebenen Thematik kann man ein HMM als eine erweiterte Markov Kette betrachten. Neben dem Zufallsexperiment der einzelnen Zustandsübergänge wird in jedem Zustand ein weiteres Experiment durchgeführt: Die Auswahl des Emissionssymbols b, das durch die Matrix der Emissionswahrscheinlichkeiten E gegeben ist. Bei einem HMM wird also das Symbol b von dem Zustand k entkoppelt und es gilt:

$$e_k(b) = P(x_i = b | \pi_i = k)$$

Dabei beschreibt π die Sequenz der Zustände. Der Übergang von einem Zustand zum anderen ist durch folgende Formel bestimmt:

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k)$$

Die gemeinsame Wahrscheinlichkeit einer Symbolsequenz x und einer Zustandsfolge π mit fester Länge L lautet

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

Zuerst wird also der Startzustand durch $a_{0\pi_1}$ bestimmt. Dann wird für jedes einzele Symbol das Produkt aus der Emissionswahrscheinlichkeit und der Übergangswahrscheinlichkeit bestimmt.

Hier wird der Unterschied zu einer Markov Kette deutlich, der auch das Wort „hidden“ in HMM rechtfertigt. Das einzelne Symbol b ist nicht eindeutig durch einen Zustand bestimmt, da es in mehreren Zuständen möglich ist b zu generien. Der Pfad π ist also nicht eindeutig.

Beispiel Um die Definition zu verdeutlichen, wird diese anhand des Beispiels des „unehrlichen Casinos“ erklärt.

Gegeben sei ein Casino, das zeitweise einen gezinkten Würfel benutzt, sodass die Gleichverteilung der Zahlen 1 bis 6 nicht mehr gegeben ist. Bei dem gezinkten Würfel fällt die 6 mit einer Wahrscheinlichkeit von 1/2, alle anderen Zahlen mit einer Wahrscheinlichkeit von 1/10. Der Croupier wechselt nach jedem Wurf mit einer gewissen Wahrscheinlichkeit den Würfel oder fährt mit dem aktuellen fort. Diese Situation kann man als HMM modellieren, das sich wie die Markov Kette als Graph darstellen lässt (vgl. Abbildung 1.2).

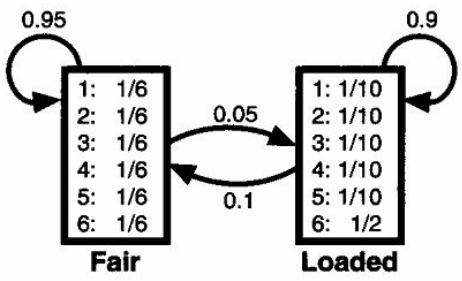

Abbildung 1.2: HMM als Graph: Das unehrliche Casino, Quelle: (Durbin et al., 1998, S. 55)

Hierbei gilt:

- $S = \{1, 2\}$ mit 1=Fair und 2=Loaded
- $V = \{1, 2, 3, 4, 5, 6\}$
- $\pi_{Start} = (\frac{1}{2}, \frac{1}{2})$
- $A = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$

- $E = \begin{pmatrix} 1/6 & 1/10 \\ 1/6 & 1/10 \\ 1/6 & 1/10 \\ 1/6 & 1/10 \\ 1/6 & 1/10 \\ 1/6 & 1/2 \end{pmatrix}$

Lokalisierungsproblem HMMs haben gegenüber den Markov Ketten den Vorteil, dass sie DNA-Sequenzen generieren, in denen CpG-Inseln auftreten können, aber nicht müssen. Das passende HMM ist in Abbildung 1.3 dargestellt. Hierbei kommen noch die Übergangswahrscheinlichkeiten innerhalb der „+“ - und „-“ -Modelle hinzu, wie es bei Abbildung 1.1 der Fall ist.

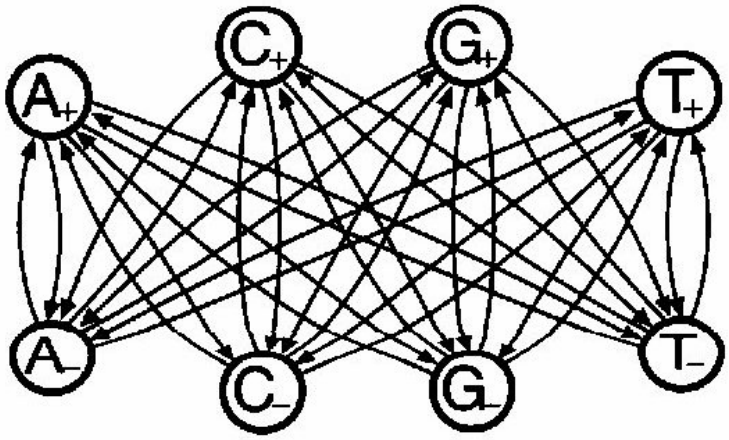

Abbildung 1.3: DNA HMM, Quelle: (Durbin et al., 1998, S. 53)

Das HMM wird sich aufgrund der Übergangswahrscheinlichkeiten wahrscheinlich öfter im „- -Modell“ befinden. Das ist intuitiv klar, da die CpG-Inseln in einer DNA-Sequenz auch selten vorkommen. Es stellt sich die Frage, wo diese Inseln in der gegebenen DNA-Sequenz sind. Dies wird als Lokalisierungs-Problem bezeichnet.

Problem 2: Lokalisierungs-Problem

Gegeben:
Eine Zeichenkette x mit der Länge L über dem Alphabet $Z = \{A, C, G, T\}$

Gesucht:
Position der CpG-Inseln

Anmerkung zum Lokalisierungs-Problem: Die Lösung, die vorgestellt wird, erklärt nicht, wie genau die Position angegeben wird. Dies hängt von der konkreten Implementierung ab, auf die hier nicht eingegangen wird.

Eine Lösung des Problems ist, es auf das Diskriminations-Problem zu reduzieren. Man legt ein Fenster der Größe l Nukleotide fest und betrachtet die DNA-Sequenz als Eingabe-Stream. Dieser Stream wird nun Nukleotid für Nukleotid in das Fenster eingegeben und bei jedem Schritt als neues Diskriminations-Problem betrachtet. Bei dieser Lösung ist aber nicht klar, wie l gewählt werden muss, um sinnvolle Ergebnisse zu erhalten.

Stattdessen kann man bei einer gegebenen Zeichenkette x den wahrscheinlichsten Pfad durch das HMM in Abbildung 1.3 suchen. Dieser kann durch das „+ -Modell“ führen, und somit hat man die CpG-Insel(n) lokalisiert. Dieses Problem löst der Viterbi-Algorithmus.

1.5 Viterbi-Algorithmus

Um den Viterbi-Algorithmus zu erklären, bietet sich eine andere Darstellungsart von HMMs an: Die „Irrfahrt“. Mit einer „Irrfahrt“ kann man eine zeitliche Abfolge der Zustandsübergänge simulieren. Jede Spalte entspricht dabei einem Zeitpunkt. Die Kanten geben weiterhin die Übergangswahrscheinlichkeiten an. Abbildung 1.4 zeigt die „Irrfahrt“ für das zeitweise „unehrliche Casino“.

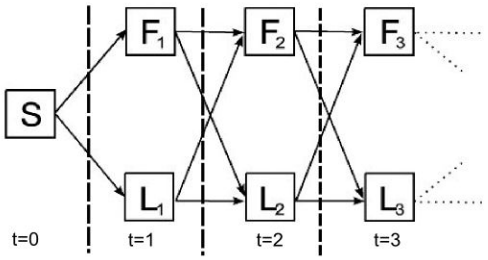

Abbildung 1.4: unehrliches Casino, Darstellung als „Irrfahrt“

Der Viterbi-Algorithmus arbeitet in drei Phasen, der Initialisierungs-, der Pfadberechnungs-, und der Backtrackingsphase. Dabei werden 2 Datenstrukturen verwendet. Für jeden Zustand in der „Irrfahrt“ wird ein Wert abgespeichert, der angibt, wie wahrscheinlich es ist, zum Zeitpunkt t in diesem Zustand i zu sein. Diese lokale Pfadwahrscheinlichkeit ist durch

$$\delta_t(i) = \max_{\pi} P(x_1, \ldots, x_t, \pi_1, \ldots, \pi_t) \text{ mit } \pi_t = i$$

gegeben. Weiter wird für jeden Zustand i ein Zeiger abgespeichert, der auf den Vorgängerzustand $i-1$ zeigt: $\phi_k(i)$.

Initialisierung Bei der Initialisierung wird die lokale Pfadwahrscheinlichkeit für den Zeitpunkt $t = 1$ berechnet. Der Zustand bei $t = 0$ wird immer mit einer Wahrscheinlichkeit von 1 eingenommen. Die Pfadwahrscheinlichkeit ergibt sich bei einer Eingabesequenz $x = (x_1, \ldots, x_L)$ aus dem Produkt der Emissionswahrscheinlichkeit $e_t(x_i)$ und der Zustandsstartwahrscheinlichkeit π_i:

$$\delta_1(i) = e_i(x_1)\pi_i$$

Pfadberechnung Während der Pfadberechnungsphase wird für jeden Zeitpunkt und Zustand eine lokale Pfadwahrscheinlichkeit berechnet. Es gilt:

$$\delta_t(i) = \max_j \left(\delta_{t-1}(j) a_{ji} e_i(x_t)\right)$$

Die neu zu berechnende Pfadwahrscheinlichkeit $\delta_t(i)$ ergibt sich aus dem Produkt der Pfadwahrscheinlichkeit des vorangegangenen Zustandes, die Übergangswahrscheinlichkeit von diesem zu dem aktuellen Zustand und der Emissionswahrscheinlichkeit des passenden Symbols des aktuellen Zustandes. Da es mehrere Pfade zu dem aktuellen Zustand geben kann, wird das Maximum als neuer Wert gewählt. Der Zustand, mit dessen Hilfe man diesen maximalen Wert berechnet hat, wird als Vorgängerzustand in $\phi_t(i)$ eingetragen.

Backtrackingsphase Wenn für jeden Zeitpunkt und Zuständ die Pfadwahrscheinlichkeit berechnet wurde, wird für den letzten Zeitpunk $t = L$ der Zustand i_L mit der maximalen Wahrscheinlichkeit ermittelt. Mit Hilfe von $\phi_L(i_L)$ kann der wahrscheinlichste Pfad von diesem Zustand bestimmt werden. Die umgekehrte Ausgabe dieser Vorgängerzustände entspricht dem gesuchten wahrscheinlichsten Pfad.

Laufzeit Zur Laufzeitbestimmung sind nur die Schlüsselvergleiche bei der Pfadberechnung zu beachten, da die anderen Berechnungen des Algorithmus in konstanter Zeit geschehen können. Man kann sich intuitiv an der Abbildung 1.5 die Laufzeit des Algorithmus verdeutlichen. Gegeben sei das Alphabet $Z = \{A, B, C\}$ und eine Sequenz der Länge L. Zum Zeitpunkt $k = 1$ werden $|Z|$ Schlüsselvergleiche, und zum Zeitpunkt $k = 2, k = 3 \ldots$ $|Z^2|$ benötigt. Insgesamt werden also $(L-1)|Z|^2 + |Z|$ Schlüsselvergleiche vorgenommen. Und diese liegen in $O(L|Z|^2)$.

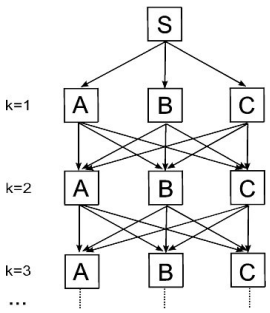

Abbildung 1.5: Laufzeit des Viterbi-Algorithmus

1.6 Forward-Algorithmus

Der Forward-Algorithmus berechnet die Wahrscheinlichkeit, dass ein HMM zum Zeitpunkt t in einem Zustand k ist und die Sequenz $x = (x_1, x_2, \ldots, x_t)$ erzeugt hat (Evaluierungsproblem I).

Problem 3: Evaluierungs-Problem I

Gegeben:
Eine Beobachtung $x = (x_1, x_2, \ldots, x_t, \ldots, x_L)$ über einem Alphabet und ein HMM δ

Gesucht:
Wahrscheinlichkeit, dass das HMM die Sequenz $(x_1, \ldots, x_t)$ erzeugt und im Zustand k ist.

Der Algorithmus hat eine Initialisierungs-, Rekursions-, und Terminationsphase. Zudem wird die Vorwärtsvariable

$$f_k(i) = P(x_1, \ldots, x_i, \pi_i = k)$$

berechnet. Sie gibt die Wahrscheinlichkeit an, dass die Sequenz $(x_1, \ldots, x_i)$ generiert wurde und sich das HMM im Zustand $\pi_i = k$ befindet. Sie ist rekursiv durch

$$f_l(i+1) = e_l(x_{i+1}) \sum_k f_k(i) a_{kl}$$

definiert. Dazu wird in der Initialisierungsphase $f_S(i)$ festgelegt.

Dabei ist e_l die Wahrscheinlichkeit, das passende Symbol auszugeben. Der Term $f_k(i)a_{kl}$ beschreibt die Wahrscheinlichkeit, sich im Zustand k, der zeitlich vor l liegt, zu befinden, und die passende Ausgabe bis zu diesem Punkt generiert zu haben. Dies wird mit der Übergangswahrscheinlichkeit zu dem aktuellen Zustand l multipliziert, um die Gesamtwahrscheinlichkeit zu erhalten. Da mehrere Kanten in einer „Irrfahrt“ nach l führen können, werden die Wahrscheinlichkeiten addiert. Wenn alle $f_l(i)$ festgelegt sind, wird die Gesamtwahrscheinlichkeit für die Sequenz x durch

$$P(x) = \sum_k f_k(L) a_{k0}$$

berechnet, die für den Forward-Backward-Algorithmus wichtig ist.

1.7 Backward-Algorithmus

Im Gegensatz zum Forward-Algorithmus berechnet der Backward-Algorithmus nicht die Wahrscheinlichkeit in einen Zustand k zu gelangen. Er geht davon aus in dem Zustand k zu sein und berechnet die Wahrscheinlichkeit eine (Teil-)Sequenz zu erzeugen (Evaluierungsproblem II).

Problem 4: Evaluierungs-Problem II

Gegeben:
Eine Beobachtung $x = (x_1, x_2, \ldots, x_t, \ldots, x_L)$ über einem Alphabet und ein HMM δ

Gesucht:
Wahrscheinlichkeit, von einem Zustand k aus $(x_{t+1}, \ldots, x_L)$ zu erzeugen.

Dazu wird die Rückwärtsvariable $b_k(x_i)$ definiert:

$$b_k(i) = P(x_{i+1}, \ldots, x_L | \pi_i = k)$$

Sie gibt genau die gesuchte Wahrscheinlichkeit an: Die Sequenz $(x_{i+1}, \ldots, x_L)$ soll erzeugt werden unter der Bedingung, dass sich das HMM zu Anfang im Zustand k befand.

Wie die Vorwärtsvariable, so ist auch die Rückwärtsvariable rekursiv definiert:

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$

Die Initialisierung erfolgt durch $b_k(L) = a_{k0}$. Somit kann man $b_k(i)$ für jeden Zustand und Zeitpunkt berechnen. Dazu ermittelt man das Produkt aus der schon bekannten Rückwärtsvariable, der Emissionswahrscheinlichkeit und der Übergangswahrscheinlichkeit zu dem aktuellen Zustand. Da sich in einer „Irrfahrt“ so ebenfalls mehrere Pfade ergeben können, werden die einzelnen Wahrscheinlichkeiten addiert.

1.8 Forward-Backward-Algorithmus

Der Forward-Backward-Algorithmus kombiniert -wie der Name schon sagt- die letzten beiden vorgestellten Algorithmen. Mit seiner Hilfe kann man die Wahrscheinlichkeit berechnen, mit der sich das HMM bei einer gegebenen Sequenz x im Zustand k befindet:

Problem 5: Evaluierungs-Problem III

Gegeben:
Eine Beobachtung $x = (x_1, x_2, \ldots, x_t, \ldots, x_L)$ über einem Alphabet und ein HMM δ

Gesucht:
$P(\pi_i = k|x)$, d.h. die Wahrscheinlichkeit, dass sich das HMM bei einer gegebenen Beobachtung x im Zustand k befindet.

Die Wahrscheinlichkeit, dass x generiert wird und sich das HMM im Zustand k befindet, kann man somit wie folgt in einem Ausdruck zusammenfassen:

$$P(x, \pi_i = k)$$

Man kann die Sequenz x der Länge L auch als $(x_1, \ldots, x_i, x_{i+1}, \ldots, x_L)$ schreiben und somit ergibt sich der äquivalente Ausdruck:

$$P(x_1, \ldots, x_i, x_{i+1}, \ldots, x_L, \pi_i = k)$$

Mit Hilfe der Regel $P(A, B) = P(A|B)P(B)$ formt man mit $A = x_{i+1}, \ldots, x_L$ und $B = x_1, \ldots, x_i, \pi_i = k$ den Ausdruck um:

$$P(x_{i+1}, \ldots, x_L | x_1, \ldots, x_i, \pi_i = k) P(x_1, \ldots, x_i, \pi_i = k)$$

Da nach der Definition der Markov Kette der aktuelle Zustand nur von seinem Vorgängerzustand abhängt, erhält man:

$$P(x_{i+1}, \ldots, x_L | \pi_i = k) P(x_1, \ldots, x_i, \pi_i = k)$$

Dies ist nach Definition äquivalent zu

$$b_k(i) f_k(i)$$

Die im Evaluierungsproblem III gesuchte Wahrscheinlichkeit $P(\pi_i = k|x)$, ergibt sich aus dem Quotienten dieser Formel und der Wahrscheinlichkeit von x, die man mit dem Forward-Algorithmus berechnen kann. Es gilt:

$$P(\pi_i = k|x) = \frac{P(x, \pi_i = k)}{P(x)} = \frac{f_k(x_i) b_k(x_i)}{P(x)}$$

Somit löst der Forward-Backward-Algorithmus das Evaluierungsproblem III.

Literaturverzeichnis

CpG-Insel, 2009. URL `http://www.diss.fu-berlin.de/diss/servlets/MCRFileNodeServlet/FUDISS_derivate_000000000325/08_kap4.pdf`. [Online; Zugriff am 19.1.2009].

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

G. Fink. *Mustererkennung mit Markov-Modellen.Theorie - Praxis - Anwendungsgebiete.* B.G. Teubner Verlag, 2003.

Krebsinformationsdienst, 2009. URL `http://www.krebsinformationsdienst.de/themen/grundlagen/index.php`. [Online; Zugriff am 20.12.2008].

R. Merkl and S. Waack. *Bioinformatik Interaktiv.Algorithmen und Praxis.* WILEY-VCH Verlag, 2003.

Wikipedia: Krebs, 2009. URL `http://de.wikipedia.org/wiki/Krebs_(Medizin)`. [Online; Zugriff am 20.12.2008].