Universität Karlsruhe
Institut für Rechnerentwurf und Fehlertoleranz
Institutsleiter: Prof. Dr.-Ing. D. S

P B

WS 2001/2002

M

M

Autor: W G

Zusammenfassung

Seit Anfang der 70er Jahre werden verdeckte Markovmodelle in den Verfahren der Spracherkennung eingesetzt, für die sie ursprünglich entwickelt wurden. Inzwischen hat man auch in der Bioinformatik einige Anwendungsbereiche für sie gefunden, wie z.B. in der Strukturanalyse, der Suche in Proteindatenbanken oder beim multiplen Alignment von DNS-Sequenzen.

Wir betrachten zuerst Markovketten, die es ermöglichen, Abfolgen von bestimmten Symbolen zu modellieren, bei denen die Wahrscheinlichkeit für das Auftreten eines Symbols von dem vorhergehenden Symbol abhängt. Bei verdeckten Markovmodellen existiert zusätzlich zu der beobachtbaren Sequenz von *Symbolen* eine Sequenz von (in der Regel nicht beobachtbaren) *Zuständen*, die die Wahrscheinlichkeit für das Auftreten der einzelnen Symbole beinflussen. Mit Hilfe einiger Algorithmen ist es möglich, von den beobachteten Symbolen auf die Zustände rückzuschließen. Dies sind der *Viterbi*-Algorithmus, um diejenige Sequenz der Zustände mit der größten Wahrscheinlichkeit zu finden, der *Forward*-Algorithmus, um die totale Wahrscheinlichkeit für das Auftreten einer bestimmten Sequenz von Symbolen zu ermittlen, und der *Backward*-Algorithmus, um an einer bestimmten Stelle in der Symbolsequenz auf den momentanen Zustand schließen zu können. Die Exaktheit solcher Algorithmen wird von der Genauigkeit der Fließkommaprozessoren eingschränkt, weswegen wir zwei Möglichkeiten betrachten, Rechenfehler zu minimieren. Weiterhin werden wir uns überlegen, wie die Parameter und die Struktur eines Modells überhaupt erst sinnvoll festzulegen sind.

Ergänzend betrachten wir Markovketten höherer Ordnung, bei denen die Wahrscheinlichkeit für das Auftreten eines Symbols von mehreren vorhergehenden Symbolen abhängt, und inhomogene Markovketten, mit denen man eine Sequenz durch mehrere unterschiedliche Markovketten modellieren kann.

Im Anhang werden einige für die Sequenzanalyse wichtige stochastische Begriffe erklärt.

Inhaltsverzeichnis

1 Markovketten

Beispiel: CpG-Inseln Dieses Beispiel aus [DEKM99] soll die Einführung einiger neuer Begriffe und Verfahren der Wahrscheinlichkeitstheorie motivieren. Tritt im menschlichen Genom das Basenpaar CG (überlicherweise als CpG geschrieben, um es von den C-G Paaren entlang eines DNS-Strangs zu unterscheiden) auf, so ist die Wahrscheinlichkeit sehr groß, dass das C (Cytosin) in ein T (Thymin) mutiert. Folglich sind diese CpG Paare seltener im Genom zu finden, als man aufgrund der jeweiligen relativen Häufigkeiten von C und G erwarten würde. Aus biologischen Gründen wird dieser Umwandlungsprozess jedoch in manchen Abschnitten des Genoms unterdrückt, z.B. in den Anfangsbereichen (Promotern) vieler Gene. In diesen Bereichen finden sich mehr CpG Paare als üblich, und oft ist auch die jeweilige Häufigkeit von C und G in diesen Bereichen höher. Diese Bereiche werden als *CpG-Inseln* bezeichnet, sie sind üblicherweise einige hundert bis einige tausend Basen lang. Es ergeben sich daraus zwei Probleme, deren Lösung man mit Hilfe stochastischer Verfahren näher kommt. Erstens: Wie kann man entscheiden, ob eine bestimmtes Stück einer Genom-Sequenz aus einer CpG-Insel stammt? Zweitens: Wie kann man in einer gegebenen Sequenz die CpG-Inseln herausfinden?

1.1 Markovketten als Modelle für DNS-Sequenzen

Wir benötigen ein Modell, das Sequenzen generiert, in denen die Wahrscheinlichkeit für ein Symbol von dem vorhergehenden Symbol abhängt. Ein einfaches solches Modell sind Markovketten.

Definition 1.1 Eine Familie von Zufallsvariablen X_t, die von einem Parameter $t \in T$ (z.B. der Zeit) abhängen und Werte aus einer endlichen Zustandsmenge Z annehmen, wird als *stochastischer Prozeß* bezeichnet. Eine *Markovkette*[1)] ist ein stochastischer Prozeß $\{X_t : t \in \mathbb{N}_0\}$, der folgende *markovsche Eigenschaft* besitzt:

$$P(X_{t+1} = z_{t+1} | X_0 = z_0, \ldots, X_t = z_t) = P(X_{t+1} = z_{t+1} | X_t = z_t). \tag{1.1}$$

Interpretiert man X_{t+1} als Zustand eines Systems zum Zeitpunkt $t + 1$, so hängt dieser Zustand nur davon ab, in welchem Zustand X_t sich das System zum Zeitpunkt t befand, nicht aber von den früheren Zuständen des Systems. Für alle $z, z' \in Z$ bezeichnen wir mit

$$p_{z \to z'} = P(X_{t+1} = z' | X_t = z) \tag{1.2}$$

die *Übergangswahrscheinlichkeit* vom Zustand z in den Zustand z'. Bei den *homogenen* Markovketten, die wir hier betrachten, ist die Übergangswahrscheinlichkeit unabhängig von t.

Man kann sich eine Markovkette als einen gerichteten Graphen vorstellen, bei dem die Kanten den Übergangswahrscheinlichkeiten und die Knoten den Zuständen entsprechen (Abb. 1). In unserem Beispiel entsprechen die Zustände den vier Buchstaben A, C, G und T des DNS-Alphabets. Die Übergangswahrscheinlichkeiten sind folgendermaßen definiert:

$$p_{\mu \to \lambda} = P(x_i = \lambda | x_{i-1} = \mu),$$

wobei die x_i den aufeinanderfolgenden Nukleotiden in einer DNS-Sequenz entsprechen und $\mu, \lambda \in \{\text{A, C, G, T}\}$ sind.

[1)] A A M (1856-1922)

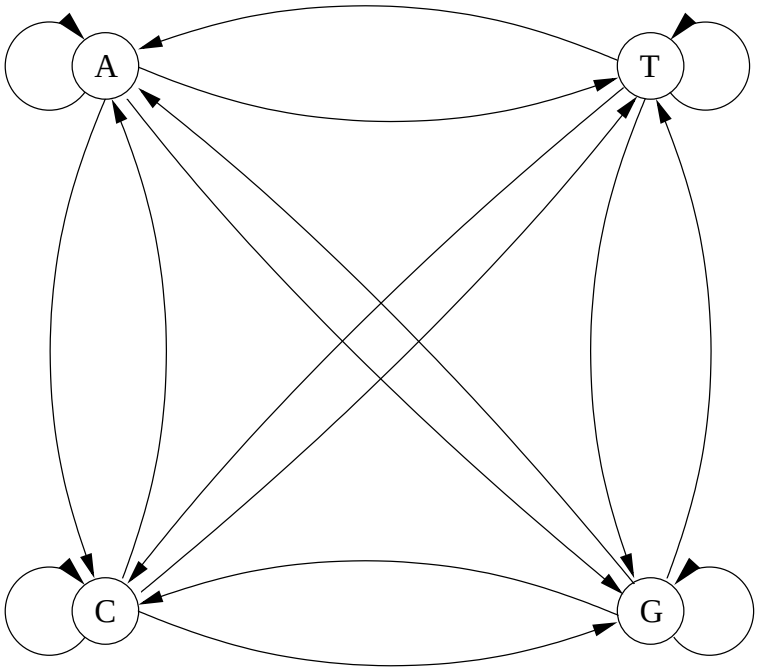


Abbildung 1: Markovkette für DNS

Ist $x = (x_1, \ldots, x_n)$ eine beliebige Sequenz, so ist aus der Stochastik bekannt, dass

$$P(x) = P(x_n, x_{n-1}, \ldots, x_1) = P(x_n|x_{n-1}, \ldots, x_1)P(x_{n-1}|x_{n-2}, \ldots, x_1) \cdots P(x_1)$$

gilt. Mit der markovschen Eigenschaft ergibt dies

$$P(x) = P(x_n|x_{n-1})P(x_{n-1}|x_{n-2}) \cdots P(x_2|x_1)P(x_1) = P(x_1)\prod_{i=2}^{n} p_{x_{i-1} \to x_i}. \tag{1.3}$$

1.2 Start- und Endzustände

Wie aus (1.3) ersichtlich ist, muss man zusätzlich zu den Überganswahrscheinlichkeiten auch noch die Wahrscheinlichkeit $P(x_1)$ dafür angeben, in einem bestimmten Zustand zu beginnen. Hierfür können wir die Menge der Zustände um einen besonderen Startzustand $\mathcal{S}$ erweitern. Wenn wir $x_0 = \mathcal{S}$ definieren, gilt:

$$P(x_1 = \mu) = p_{\mathcal{S} \to \mu}.$$

Ebenso können wir einen Endzustand $\mathcal{E}$ definieren, um das Ende der Sequenz zu modellieren. Die Wahrscheinlichkeit, dass die Sequenz im Zustand λ abbricht, ist

$$P(\mathcal{E}|x_n = \lambda) = p_{\lambda \to \mathcal{E}}.$$

Üblicherweise wird das Ende einer Sequenz bei Markovketten nicht gesondert betrachtet; die Sequenz kann jederzeit abbrechen. Indem man einen Endzustand hinzufügt erhält man aus (1.3) eine Wahrscheinlichkeitsverteilung über alle möglichen Sequenzen jeder beliebigen Länge.
Bei einfachen Modellen wie den CpG-Inseln werden wir Start- und Endzustand im folgenden einheitlich mit 0 bezeichnen.

1.3 Markovketten zur Unterscheidung benutzen

In der Regel werden die Übergangswahrscheinlichkeiten nicht von vornherein bekannt sein. Falls bereits eine gewisse Datenmenge vorliegt (wie bei unserem Beispiel in Form umfangreicher Gen-Datenbanken), ist es möglich, die Übergangswahrscheinlichkeiten als ML-Schätzer zu bestimmen (vgl. A.7 und [DEKM99]). Man bestimmt dabei das Verhältnis der Übergänge eines Zustands μ in einen Zustand λ und aller Übergänge von μ in irgendeinen Zustand:

$$p_{\mu\to\lambda} = \frac{c_{\mu\to\lambda}}{\sum_{\lambda'} c_{\mu\to\lambda'}}, \tag{1.4}$$

wobei $c_{\mu\to\lambda}$ die Häufigkeit der Übergänge von μ nach λ ist.
Falls es zwei Modelle mit gleichen Zuständen, aber verschiedenen Übergangswahrscheinlichkeiten gibt, die das einer bestimmten Sequenz x zugrunde liegende Modell sein könnten, so kann man über einen *Likelihood-Quotienten-Test* feststellen, für welches Modell die Wahrscheinlichkeit dafür größer ist. Im folgenden nennen wir diese Modelle + und -. Mit (1.4) kann man für beide Modelle die $p^+_{\mu\to\lambda}$ und $p^-_{\mu\to\lambda}$ berechnen und daraus das logarithmische Verhältnis (engl. *log odds score*) der Wahrscheinlichkeiten:

$$S(x) = \log \frac{P(x|\text{Modell +})}{P(x|\text{Modell -})} = \sum_{i=1}^{n} \log \frac{p^+_{x_{i-1}\to x_i}}{p^-_{x_{i-1}\to x_i}}. \tag{1.5}$$

Für positive $S(x)$ ist Modell + das wahrscheinlichere, für negative $S(x)$ Modell -. Ist $S(x) \approx 0$, so sind beide Modelle ungefähr gleich wahrscheinlich.

In unserem Beispiel mit den CpG-Inseln kann man dieses Verfahren verwenden, um zu bestimmen, ob eine DNS-Sequenz eine CpG-Insel ist oder nicht. Man bestimmt für bereits bekannte CpG-Inseln (Modell +) die ML-Schätzer für die Übergangswahrscheinlichkeiten, ebenso für diejenigen Sequenzen, die keine CpG-Inseln sind (Modell -). Dann berechnet man (1.5) für die vorliegende DNS-Sequenz und kann anhand des Ergebnisses eine Aussage darüber machen, ob es sich um eine CpG-Insel handelt oder nicht. Um dies anhand echter Daten zu veranschaulichen, betrachten wir Abschnitte von 122 ausgewählten DNS-Sequenzen des menschlichen Genoms (von denen 48 als CpG-Inseln eingestuft werden), aus denen wir die ML-Schätzer für die Übergangswahrscheinlichkeiten $p_{\mu\to\lambda}$ unserer Modelle gewinnen:

$p^+_{\mu\to\lambda}$	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

$p^-_{\mu\to\lambda}$	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Hierbei steht in Zeile i und Spalte j die Wahrscheinlichkeit, dass die j-te Base der i-ten Base folgt. Die Wahrscheinlichkeiten in jeder Zeile addieren sich zu 1. Wie nicht anders zu erwarten, ist im Modell + die Wahrscheinlichkeit dafür, dass ein G einem C folgt deutlich höher ist als im Modell -.
Mit diesen Werten können wir nun das logarithmische[2)] Verhältnis der Übergangswahrscheinlichkeiten der beiden Modelle berechnen:

[2)] Wir verwenden hier 2 als Basis des Logarithmus.

$\log \frac{p^+_{\mu\to\lambda}}{p^-_{\mu\to\lambda}}$	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

Aus diesen Werten wiederum lässt sich eine Verteilung von (1.5) für die 48 Sequenzen berechnen. Sie ist in Abb. 2 dargestellt, wobei $S(x)$ durch die Länge der jeweiligen Sequenz geteilt wurde. Ohne diese Normierung wäre die Streuung der Verteilung wesentlich größer.

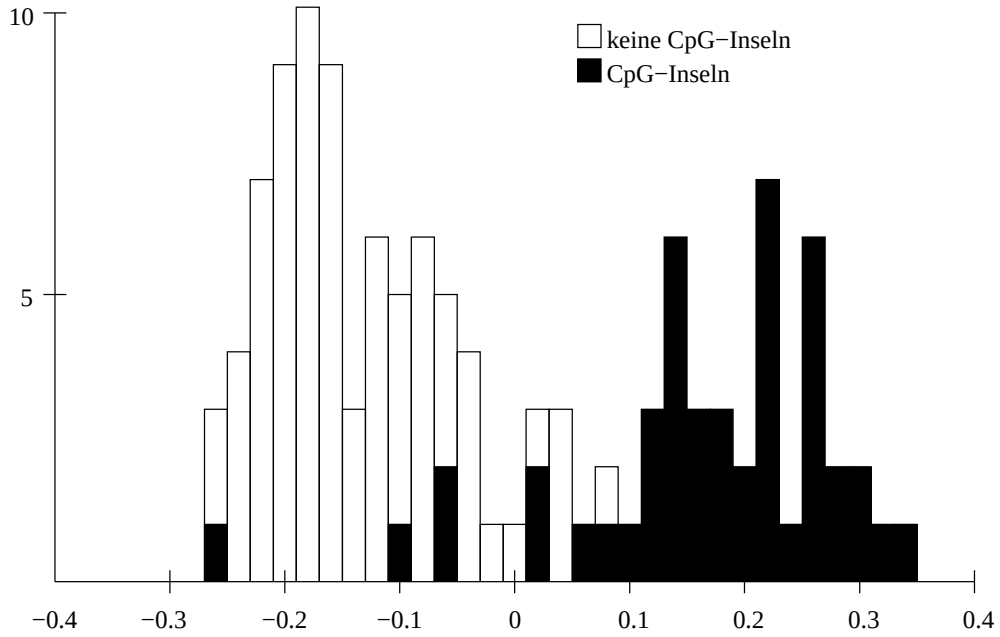


Abbildung 2: Verteilung der $S(x)$.

Wir sehen, dass diese Verteilung eine brauchbare Unterscheidung zwischen CpG-Inseln und den übrigens Bereichen einer Sequenz liefert. Die wenigen Fehler können von falsch geschätzten Modellparametern oder von von Anfang an falsch eingestuften Testdaten herkommen.

2 Verdeckte Markovmodelle

2.1 Einführung

Wir wenden uns nun der Frage zu, wie man in einer gegebenen DNS-Sequenz die CpG-Inseln herausfinden kann. Mit dem Verfahren aus 1.3 wäre es möglich, die Sequenz in Abschnitte fester Länge zu unterteilen und (1.5) für jeden Abschnitt zu berechnen. Anhand des Ergebnisses könnte man dann entscheiden, ob es sich um eine CpG-Insel handelt oder nicht.
Dieses Verfahren ist jedoch problematisch, wenn wir davon ausgehen, dass CpG-Inseln genaue Abgrenzungen besitzen und in ihrer Länge stark variieren könnnen. Es ist vorteilhafter, die beiden Modelle aus 1.3 zu einem einzigen Modell für die gesamte Sequenz zusammenzufassen. Das neue Modell besteht also aus den Zuständen $\{A_+, C_+, G_+, T_+, A_-, C_-, G_-, T_-\}$ und den Überganswahrscheinlichkeiten zwischen diesen Zuständen. Bei den Markovketten konnte jedem Symbol x_i aus der Sequenz x eindeutig ein Zustand zugeordnet werden, bei dem neuen Modell gibt es aber für jedes beobachtete Nukleotid *zwei* mögliche Zustände, in denen sich das Modell an Stelle i befinden kann. Allein aus der Kenntnis von x_i ist es also nicht mehr möglich, auf den Zustand an Stelle i zu schließen. Aufgrund dieser Eigenschaft bezeichnet man diese Modelle als *verdeckte Markovmodelle* (engl. *Hidden Markov Models* - HMMs).

Man unterscheidet bei verdeckten Markovmodellen also zwischen einer Sequenz x von Symbolen und einer Sequenz $\pi = (\pi_1, \ldots, \pi_n)$ von Zuständen, dem *Pfad*. Der Pfad selbst ist eine Markovkette mit Übergangswahrscheinlichkeiten

$$p_{\mu\to\lambda} = P(\pi_i = \lambda | \pi_{i-1} = \mu). \tag{2.1}$$

Da die x_i nicht mehr eindeutig einem π_i zugeordnet werden können, muss man für jedes Symbol die *Emissionswahrscheinlichkeit*[3)] in Abhängigkeit von einem bestimmten Zustand angeben:

$$e_\mu(b) = P(x_i = b | \pi_i = \mu). \tag{2.2}$$

Daraus kann man leicht die Wahrscheinlichkeit für eine bestimmte Sequenz und einen bestimmten Pfad bestimmen:

$$P(x, \pi) = p_{0\to\pi_1} \prod_{i=1}^{n} e_{\pi_i}(x_i) p_{\pi_i\to\pi_{i+1}}, \tag{2.3}$$

wobei 0 wieder den Start- und Endzustand beschreibt und $\pi_{n+1} = 0$ gefordert werden muss.

In unserem Beispiel sind die Emissionswahrscheinlichkeiten alle 0 oder 1, da jeder Zustand nur ein Symbol emittieren kann. Dies ist natürlich im Allgemeinen nicht der Fall.

Beispiel: Das gelegentlich unehrliche Kasino In diesem Beispiel (ebenfalls aus [DEKM99]) betrachten wir ein Kasino, das beim Würfelspiel gelegentlich den fairen Würfel A gegen einen verfälschten Würfel B austauscht. Dies geschehe bei jedem Wurf mit der Wahrscheinlichkeit 0.05, mit einer Wahrscheinlichkeit von 0.1 werde von B wieder zurück zu A gewchselt. Der Würfel B habe die Wahrscheinlichkeit 0.5 für das Würfeln einer 6 und 0.1 für jede andere

[3)] Man verwendet diese Bezeichnung, da es oft hilfreich ist, sich verdeckte Markovmodelle als erzeugende Modelle vorzustellen, die eine Sequenz generieren oder „emittieren“.

Zahl. Die Würfel A und B entsprechen den Zuständen, die unser Modell annehmen kann, die Augenzahlen 1 bis 6 entsprechen den beobachteten Symbolen in einer Sequenz. Es sind

$$p_{A\to B} = 0.05,\ p_{B\to A} = 0.1,$$
$$e_A(1) = e_A(2) = e_A(3) = e_A(4) = e_A(5) = e_A(6) = \frac{1}{6},$$
$$e_B(1) = e_B(2) = e_B(3) = e_B(4) = e_B(5) = 0.1 \text{ und } e_B(6) = 0.5.$$

Als Spieler können wir aber nur mitverfolgen, welche Ergebnisse beim Würfeln herauskommen, wissen aber nichts darüber, ob und wann das Kasino einen fairen oder verfälschten Würfel einsetzt - die Sequenz der Zustände ist verdeckt! Wir können bestenfalls vermuten, dass in einer bestimmten Phase des Spiels ein verfälschter Würfel eingesetzt wird, weil die Augenzahl 6 auffällig oft gewürfelt wird. In diesem Fall sind wir natürlich an Verfahren interessiert, die es uns ermöglichen festzustellen, wann mit Sicherheit ein verfälschter bzw. fairer Würfel eingesetzt wurde.

2.2 Algorithmen für verdeckte Markovmodelle

Viterbi-Algorithmus Wie wir bereits wissen, ist es bei verdeckten Markovmodellen nicht möglich, direkt aus den beobachteten Symbolen die Zustände abzulesen. Herauszufinden, welche Zustandssequenz einer Beobachtungssequenz zugrunde liegt, wird als *Dekodierung*[4)] bezeichnet. Dafür gibt es verschiedene Ansätze, wobei wir hier mit dem Viterbi-Algorithmus[5)] nur den gebräuchlichsten betrachten.
Im Allgemeinen können viele verschieden Pfade einer Symbolsequenz x der Länge n zugrunde liegen. Die Pfade (C_+, G_+, C_+, G_+), (C_-, G_-, C_-, G_-) und (C_+, G_-, C_+, G_-) würden beispielsweise alle die Sequenz CGCG erzeugen, allerdings mit unterschiedlichen Wahrscheinlichkeiten. Die Wahrscheinlichkeit für das Auftreteten des ersten Pfades ist sehr viel größer als die Wahrscheinlichkeiten der anderen Pfade, da die Basen C und G ausserhalb von CpG-Inseln nur selten aufeinander folgen. Wenn wir für weitergehende Betrachtungen einen Pfad auswählen müssen, liegt es nahe, denjenigen mit der größten Wahrscheinlichkeit zu wählen:

$$\pi^* = \operatorname*{argmax}_{\pi} P(x, \pi). \tag{2.4}$$

Der wahrscheinlichste Pfad lässt sich rekursiv finden. Sei die Wahrscheinlichkeit $v_\mu(i)$ von π^*, an der i-ten Stelle der Sequenz im Zustand μ zu enden, für alle μ bekannt. Dann lässt sich daraus $v_\lambda(i+1)$ berechnen:

$$v_\lambda(i+1) = e_\lambda(x_{i+1}) \max_{\mu}\{v_\mu(i) p_{\mu\to\lambda}\}. \tag{2.5}$$

Da alle Pfade mit dem Zustand 0 beginnen (und enden), gilt als Anfangsbedingung $v_0(0) = 1$ und $v_\mu(0) = 0$ für $\mu \neq 0$. Die Wahrscheinlichkeit des wahrscheinlichsten Pfades ist damit:

$$P(x, \pi^*) = \max_{\mu}\{v_\mu(n) p_{\mu\to 0}\} \tag{2.6}$$

Man kann den Pfad ermitteln, indem man in jedem Rekursionschritt Zeiger auf den jeweils vorherigen Zustand setzt. Dieser Algorithmus ist (wie alle anderen) in Anhang B zusammengefasst. Beispiele für seine Anwendung findet man in [DEKM99].

[4)] Dieser Begriff stammt aus dem Bereich der Spracherkennung.
[5)] A J. V (1936)

Forward-Algorithmus Für einfache Markovketten konnte man die Wahrscheinlichkeit für das Auftreten einer Sequenz x von Symbolen mit (1.3) ermitteln. Bei verdeckten Markovmodellen muss man berücksichtigen, dass viele verschiedene Pfade der Sequenz x zugrunde liegen können. Um die totale Wahrscheinlichkeit von x zu ermitteln, müssen folglich die Wahrscheinlichkeiten aller möglichen Pfade, die x zugrunde liegen können, summiert werden:

$$P(x) = \sum_{\pi} P(x, \pi). \tag{2.7}$$

Da die Anzahl der möglichen Pfade exponentiell mit der Länge n der Sequenz wächst, ist ein Brute-force-Ansatz äußerst unpraktisch. Stattdessen kann man ein dem Viterbi-Algorithmus ähnliches Verfahren anwenden, wobei im Rekursionsschritt anstelle der maximalen Wahrscheinlichkeit die Summe der Wahrscheinlichkeiten aller relevanten Pfade benutzt wird. Wir bezeichnen mit $f_\mu(i) = P(x_1, \ldots, x_i, \pi_i = \mu)$ die Wahrscheinlichkeit, dass sich eine beobachtete Sequenz $(x_1, \ldots, x_i)$ an der Stelle i im Zustand μ befindet. Sei nun $f_\mu(i)$ für alle μ bekannt. Dann lässt sich $f_\lambda(i+1)$ ähnlich (2.5) berechnen:

$$f_\lambda(i+1) = e_\lambda(x_{i+1}) \sum_{\mu} f_\mu(i) p_{\mu\to\lambda}. \tag{2.8}$$

Hier muss als Anfangsbedingung $f_0(0) = 1$ und $f_\mu(0) = 0$ für $\mu \neq 0$ gelten. Damit ist die totale Wahrscheinlichkeit für das Auftreten einer Sequenz x der Länge n:

$$P(x) = \sum_{\mu} f_\mu(n) p_{\mu\to 0} \tag{2.9}$$

Backward-Algorithmus In bestimmten Situationen kann es interessieren, welches der wahrscheinlichste Zustand für eine Beobachtung x_i bei einer gegebenen Sequenz x ist. Dies muss nicht notwendigerweise das vom Viterbi-Algorithmus ermittelte π_i^* sein. Um dies zu ermitteln, benötigen wir die Wahrscheinlichkeit, dass die Beobachtung x_i bei gegebenem x von einem Zustand μ emittiert wurde: $P(\pi_i = \mu | x)$. Dies bezeichnen wir als die *a-posteriori-Wahrscheinlichkeit*[6] von μ an der Stelle i. Für sie gilt nach (A.1):

$$P(\pi_i = \mu | x) = \frac{P(\pi_i = \mu, x)}{P(x)}. \tag{2.10}$$

Wir benötigen für die Berechnung also die Wahrscheinlichkeit, die gesamte Sequenz x zu erzeugen, wobei das i-te Symbol vom Zustand μ emittiert wird:

$$\begin{aligned} P(\pi_i = \mu, x) &= P(\pi_i = \mu, x_1, \ldots, x_i) P(x_{i+1}, \ldots, x_n | x_1, \ldots, x_i, \pi_i = \mu) \\ &= P(\pi_i = \mu, x_1, \ldots, x_i) P(x_{i+1}, \ldots, x_n | \pi_i = \mu), \end{aligned} \tag{2.11}$$

wobei die zweite Zeile aus der markovschen Eigenschaft folgt: Alles, was nach i kommt, hängt nur vom Zustand π_i ab, nicht von den vorhergehenden Symbolen oder Zuständen. Im ersten Term erkennen wir die vom Forward-Algorithmus berechnete Variable $f_\mu(i)$ wieder. Analog dazu bezeichnen wir den zweiten Term als $b_\mu(i) = P(x_{i+1}, \ldots, x_n | \pi_i = \mu)$. $b_\mu(i)$ kann auf ähnliche Weise bestimmt werden wie $f_\mu(i)$:

$$b_\mu(i) = \sum_{\lambda} p_{\mu\to\lambda} e_\lambda(x_{i+1}) b_\lambda(i+1), \tag{2.12}$$

[6] Einige wichtige Anmerkungen zu a-posteriori-Wahrscheinlichkeiten sind in [HeKa00, S.103] und [Kren99, S.39] zu finden.

wobei $b_\mu(n) = p_{\mu\to 0}$ für alle μ gilt. Es gilt also: $P(\pi_i = \mu, x) = f_\mu(i)b_\mu(i)$. Folglich kann (2.10) geschrieben werden als

$$P(\pi_i = \mu|x) = \frac{f_\mu(i)b_\mu(i)}{P(x)}, \tag{2.13}$$

wobei $P(x)$ vom Forward-Algorithmus ermittelt wird.
Eine wichtige Anwendung von $P(\pi_i = \mu|x)$ ergibt sich, wenn es für eine Sequenz x viele Pfade mit einer Wahrscheinlichkeit ähnlich der von π^* gibt. In diesem Fall ist nicht gerechtfertigt, ausschließlich π^* zu betrachten. Will man etwa an einer ganz bestimmten Stelle i den Zustand π_i mit der größten Wahrscheinlichkeit berücksichtigen, so ist es angebracht

$$\hat{\pi}_i = \operatorname*{argmax}_\mu P(\pi_i = \mu|x)$$

zu betrachten. Je nach Struktur des Modells besteht dabei jedoch die Gefahr, dass $\hat{\pi}$ gar keine gültige Sequenz ist (siehe 2.4).
Eine andere Anwendung für $P(\pi_i = \mu|x)$ ergibt sich, wenn man eine Funktion $g(\mu)$ in Abhängigkeit von den Zuständen betrachtet. In diesem Fall ist häufig die Funktion

$$G(i|x) = \sum_\mu P(\pi_i = \mu|x)g(\mu)$$

von Interesse. Ein wichtiger Spezialfall hiervon ist, dass $g(\mu)$ den Wert 1 für bestimmte Zustände und den Wert 0 für die übrigen Zustände annimmt. Bei unserem Modell für die CpG-Inseln können wir beispielsweise $g(\mu) = 1$ definieren für $\mu \in \{A_+, C_+, G_+, T_+\}$ und $g(\mu) = 0$ für $\mu \in \{A_-, C_-, G_-, T_-\}$. Dann ist $G(i|x)$ die a-posteriori-Wahrscheinlichkeit dafür, dass die i-te Base einer gegebenen Sequenz in einer CpG-Insel liegt.

Numerische Stabilität Beim Multiplizieren vieler Wahrscheinlichkeiten wird man selbst auf modernen Rechnern schnell numerische Probleme bekommen. Wenn man davon ausgeht, dass die meisten Emissions- und Übergangswahrscheinlichkeiten in der Größenordnung 10^{-1} liegen, so kommt man bei Anwendung der eben vorgestellten Algorithmen auf Sequenzen großer Länge schnell in Größenordnungen von $10^{-100000}$(!). Kaum ein Computer wäre in der Lage, Zahlen in dieser Größenordnung sinnvoll zu verarbeiten. Wir betrachten hier zwei Wege, mit diesem Problem umzugehen:

Die Log-Transformation: Für den Viterbi-Algorithmus sollte man stets den Logarithmus aller verwendeter Größen für die Berechnungen benutzen. Wählt man z.B. als Basis 10, so ist der Logarithmus einer Wahrscheinlichkeit von $10^{-100000}$ lediglich -100000. Diese Größenordnung ist wieder in einem Bereich, der von Rechnern weitestgehend exakt verarbeitet werden kann. Dazu kommt, dass sämtliche Produkte in Summen verwandelt werden, was auf vielen Rechnern die Ablaufgeschwindigkeit des Algorithmus zusätzlich erhöht. Natürlich sollte man die logarithmischen Größen bereits vor Ausführung des Algorithmus berechnen.
Bei Forward- und Backwardalgorithmus kann man dieses Verfahren nicht so problemlos anwenden, da man den Logarithmus von Summen von (bereits logarithmierten) Wahrscheinlichkeiten berechnen muss. In der Praxis lassen sich solche Probleme oft mit einer modellspezifischen Anpassung lösen.

Skalieren der Wahrscheinlichkeiten: Eine weitere Möglichkeit besteht darin, die Variablen f_μ und b_μ so zu skalieren, dass sie in einem numerisch handhabbaren Intervall bleiben. Dafür definiert man für alle i Variablen s_i und

$$\tilde{f}_\mu(i) = \frac{f_\mu(i)}{\prod_{j=1}^{i} s_j},$$

woraus eine neue, aber nur leicht modifizierte Rekursionsformel folgt:

$$\tilde{f}_\lambda(i+1) = \frac{1}{s_{i+1}} e_\lambda(x_{i+1}) \sum_\mu \tilde{f}_\mu(i) p_{\mu\to\lambda}.$$

Die s_i können im Prinzip beliebig gewählt werden, wobei es aber praktisch ist sie so zu wählen, dass $\sum_\mu \tilde{f}_\mu(i) = 1$ gilt, woraus

$$s_{i+1} = \sum_\lambda e_\lambda(x_{i+1}) \sum_\mu \tilde{f}_\mu(i) p_{\mu\to\lambda}$$

folgt. Für die b_μ verwendet man die selben s_i.

2.3 Parameterschätzung für verdeckte Markovmodelle

Eine Schwierigkeit beim Benutzen verdeckter Markovmodelle ist es, sinnvolle Werte für die Modellparameter θ wie Übergangs- und Emissionswahrscheinlichkeiten festzulegen. In der Regel wird man über eine gewisse Menge an Testdaten, also bereits bekannten und analysierten Sequenzen von der Art, für die unser Modell passen soll, verfügen. Wir bezeichnen sie mit $x^{(1)}, \ldots, x^{(m)}$ und nehmen an, dass sie unabhängig von einander sind und deshalb nach Definition A.5 $P_\theta(x^{(1)}, \ldots, x^{(j)}) = P_\theta(x^{(1)}) \cdots P_\theta(x^{(j)})$ gilt. Dies ist gerade die Likelihood-Funktion (A.8).

Parameterschätzung bei bekannten Pfaden Oft sind bei bereits untersuchten Sequenzen die Pfade, die diesen Sequenzen zugrunde liegen, bekannt. In diesem Fall kann man die Häufigkeiten zählen, wie oft ein bestimmter Zustandsübergang benutzt wurde und wie oft ein bestimmtes Symbol in einem bestimmten Zustand emittiert wurde. Wir bezeichnen sie mit $c_{\mu\to\lambda}$ und $d_\mu(b)$. Aus ihnen lassen sich ML-Schätzer für $p_{\mu\to\lambda}$ und $e_\mu(b)$ berechnen:

$$p_{\mu\to\lambda} = \frac{c_{\mu\to\lambda}}{\sum_{\lambda'} c_{\mu\to\lambda'}} \tag{2.14}$$

$$e_\mu(b) = \frac{d_\mu(b)}{\sum_{b'} d_\mu(b')}. \tag{2.15}$$

In der Regel sind ML-Schätzer umso besser, je mehr Testdaten zur Verfügung stehen. Falls nur unzureichende Datenmengen zur Verfügung stehen, besteht die Gefahr, dass die ML-Schätzer zu sehr an die Testdaten angepasst sind und keine Allgemeingültigkeit haben. Sollte ein Zustand sogar überhaupt nicht in den Testsequenzen benutzt werden, so ist der ML-Schätzer für diesen Zustand undefiniert. Will man dies vermeiden, addiert man zu den $c_{\mu\to\lambda}$ und $d_\mu(b)$ jeweils noch bestimmte, vordefinierte Konstanten, sogenannte *Pseudozähler*. Die Wahl dieser

Konstanten sollte Vorwissen bezüglich der jeweiligen Wahrscheinlichkeiten widerspiegeln, das sich aus den Rahmenbedingungen des betrachteten Modells ergibt. Kleine Pseudozähler deuten auf ein geringes Vorwissen hin, große Pseudozähler auf ein genaueres Wissen, das mehr Daten erfordert, um geändert zu werden.

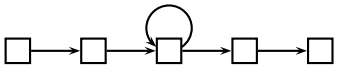
Parameterschätzung bei unbekannten Pfaden Wenn die Pfade der Testdaten nicht bekannt sind, gibt es keine geschlossene Formel, mit der man die Parameter bestimmen könnte und man muss daher ein iteratives Verfahren benutzen. Das für verdeckte Markovmodelle gebräuchlichste ist der *Baum-Welch*-Algorithmus, der aber zu kompliziert ist, um ihn hier näher auszuführen. Eine ausführlichere Beschreibung dazu findet man in [DEKM99].

2.4 Strukturen verdeckter Markovmodelle

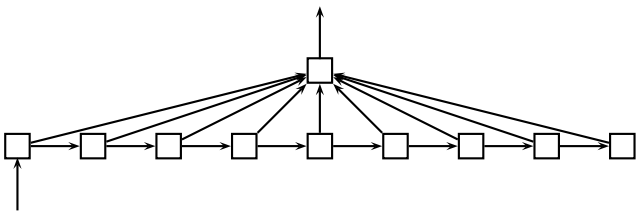
Beim Entwurf eines Modells muss man sich Gedanken darüber machen, welche Zustände das Modell enthalten soll, zwischen welchen Zuständen es überhaupt Übergänge geben soll und wie oft ein Zustand vorkommen kann. Es gibt kein bestimmtes Verfahren hierfür, man sollte die Struktur des Modells so wählen, dass sie unser bisheriges Wissen über die Zusammenhänge des Problems möglichst gut wiedergibt.

Modellierung der Länge Es ist wichtig, die Verteilung der möglichen Längen von Sequenzen in das Modell einzubeziehen. Als einfachstes Beispiel betrachten wir eine Modell, das aus nur einem Zustand besteht, der mit der Wahrscheinlichkeit p in sich selbst übergeht oder mit der Wahrscheinlichkeit $1 - p$ abbricht. Die Wahrscheinlichkeit, dass eine Sequenz der Länge l emittiert wird, nimmt hier exponentiell ab: $P(\text{Länge } l) = p^l(1 - p)$. Im Allgemeinen wird das nicht immer ein angemessenes Modell für die Verteilung der Längen sein. Wir betrachten hier einige Beispiele dafür, wie man bestimmte Verteilungen der Länge durch geeignete Modelle darstellen kann.

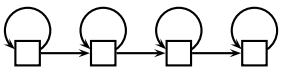
Das folgende Modell hat eine minimale Länge von fünf Zuständen und eine exponentiell abnehmende Verteilung über längere Sequenzen:



Ganz ähnlich kann man mit diesem Modell jede beliebige Länge zwischen zwei und zehn Zuständen modellieren:



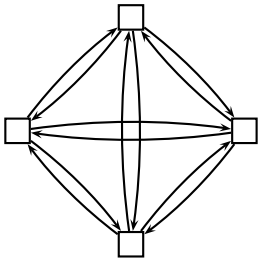
Unter Umständen wird man noch reflexive Übergänge für jeden Zustand haben wollen:



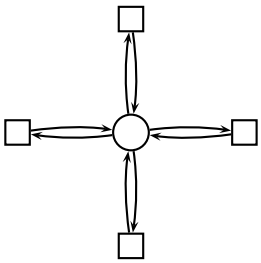
Die Verteilungsfunktion ergibt sich dann aus den Übergangswahrscheinlichkeiten zwischen den Zuständen.

Nullzustände Wir kennen aus 1.2 bereits die Zustände $\mathcal{S}$ und $\mathcal{E}$, die zwar zum Pfad gehören, aber keine Symbole emittieren. Solche Zustände bezeichnet man als *Nullzustände* (engl. *null states* oder *silent states*). Sie können auch an anderen Stellen als am Anfang oder am Ende in einem Modell nützlich sein. Hat man etwa ein Modell mit sehr vielen verschiedenen Zuständen, von denen viele miteinander verbunden sein sollen, so ist ein große Menge an Übergängen erforderlich. In solchen Fällen ist es schwer, von realistischen Testdaten noch sinnvolle Modellparameter zu schätzen. Durch die Einführung von Nullzuständen kann die Anzahl der Übergänge erheblich reduziert werden.

Betrachten wir als einfaches Beispiel vier Zustände, die jeweils paarweise miteinander verbunden sein sollen. Ohne Nullzustände würde dies 12 Übergänge erfordern:



Fügt man einen Nullzustand ein, so reduziert sich die Anzahl der nötigen Übergänge auf 8:



Stellt man sich ein entsprechendes Modell mit 100 Zuständen vor, so reduziert sich die Anzahl der Übergänge von 9900 auf 200.

Es ist wichtig zu beachten, dass man einen Übergang von einem Zustand *a* zu einem Zustand *b* nicht durch einen Umweg über einen Zustand *c* darstellen kann, da jeder dieser Zustände ein Symbol emittiert, und dadurch das ganze Modell verfälscht werden würden. Nullzustände erlauben es jedoch, beliebige Umwege zu gehen, ohne dass ein Symbol emittiert wird. Es ist also möglich, dass ein Pfad mit Nullzuständen länger ist als die von ihm emittierte Symbolsequenz.

3 Komplexere Markovketten

3.1 Markovketten höherer Ordnung

Definition 3.1 Eine Markovkette *n-ter Ordnung* ist ein stochastischer Prozess mit folgender Eigenschaft:

$$P(X_t|X_0, \ldots, X_{t-1}) = P(X_t|X_{t-n}, \ldots, X_{t-1}). \tag{3.1}$$

Der Zustand an der Stelle t hängt also von den n vorhergehenden Zuständen ab. Die Markovketten, die wir bisher betrachtet haben, waren alle von Ordnung 1.

Eine Markovkette n-ter Ordnung über einer Menge M von Symbolen ist äquivalent zu einer Markovkette erster Ordnung über einer Menge M^n von n-Tupeln aus M. Dies folgt direkt aus $P(X_{t-n+1}, \ldots, X_{t-1}, X_t|X_{t-n}, \ldots, X_{t-1}) = P(X_t|X_{t-n}, \ldots, X_{t-1})$.

Betrachten wir als Beispiel eine Markovkette zweiter Ordnung, die nur aus den Zuständen A und B besteht. Eine Sequenz ist hier äquivalent zu einer Sequenz von Paaren aus einer Markovkette erster Ordnung (Abb. 3). Die Sequenz ABBAB beispielsweise enspricht AB-BB-BA-AB.

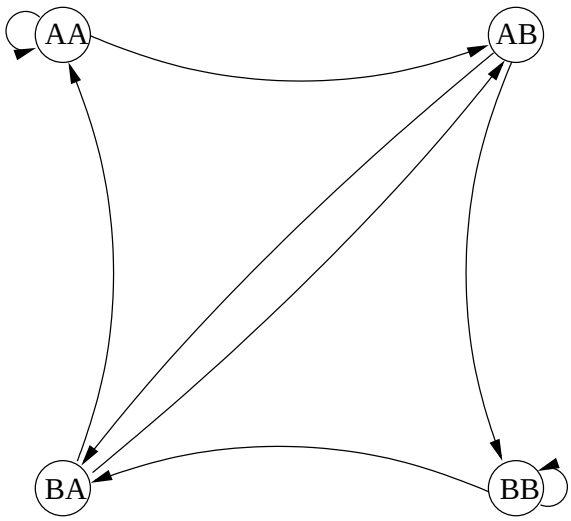


Abbildung 3: Markovkette für 2-Tupel mit vier Zuständen

In dieser Kette sind nicht alle Übergänge erlaubt, da auf ein Symbol jeweils nur zwei andere Folgen dürfen. Auf den Zustand AB können beispielsweise nur die Zustände BB und BA folgen, sonst wäre keine Äquivalenz zu einer Kette zweiter Ordnung möglich.

Obwohl sich alle Markovketten n-ter Ordnung auf Markovketten erster Ordnung zurückführen lassen, ist es in der Anwendung manchmal praktischer, das Modell der höheren Ordnung zu benutzen. Theoretisch können sie aber immer gleich behandelt werden.

3.2 Inhomogene Markovketten

Als *inhomogene* Markovketten bezeichnet man stochastische Prozesse, bei denen sich die Übergangswahrscheinlichkeiten periodisch ändern, also die $p_{\mu\to\lambda}$ von der Position i in der Sequenz und einer Periodenlänge T abhängen (wir schreiben dafür $p^{(i)}_{\mu\to\lambda}$). Dies kann man so interpretieren, dass man eine Reihe von T verschiedenen Markovketten mit gleichen Zuständen, aber unterschiedlichen Übergangswahrscheinlichkeiten betrachtet. Jedesmal, wenn ein Zustand gewechselt wird, wechselt man auch zur nächsten Markovkette in der Reihe. Nach dem T-ten Übergang wechselt man wieder zur ersten Kette[7)]. Die Definition für inhomogene Markovketten höherer Ordnung lässt sich problemlos von Definiton 3.1 übertragen.

Man sieht leicht, dass inhomogene Markovketten erster Ordnung äquivalent sind zu verdeckten Markovmodellen mit $T\cdot$[Anzahl der Zustände] Zuständen. Auch inhomogene Markovketten höherer Ordnung lassen sich auf verdeckte Markovmodelle übertragen, allerdings ist dies komplizierter und erfordert das Einfügen vieler zusätzlicher Zustände.

[7)] Das ist gleichbedeutend mit: $p^{(i)}_{\mu\to\lambda} = p^{(i \bmod T)}_{\mu\to\lambda}$.

A Stochastische Grundlagen

A.1 Stichprobenräume und Ereignisse

Definition A.1 1. Die Menge Ω der möglichen Ergebnisse eines Zufallsexperiments heißt *Grundraum* oder *Stichprobenraum*. Ihre Elemente ω repräsentieren die möglichen Ergebnisse des Zufallsexperiments.

2. Die Teilmengen $A \subset \Omega$ heißen *Ereignisse*. Wir identifizieren also A mit dem Ereignis, dass ein $\omega \in A$ der beobachtete Ausgang des Zufallsexperiments ist. Für $\omega \in \Omega$ bezeichnen wir $\{\omega\}$ als *Elementarereignis*, Ω als *sicheres Ereignis* und $\emptyset$ als *unmögliches Ereignis*.

Definition A.2 Sei $\mathcal{A}$ die Menge der Ereignisse von Ω. Eine Abbildung $P : \mathcal{A} \to [0 : 1]$ heißt *Wahrscheinlichkeitsverteilung*, wenn sie folgende Eigenschaften hat:

1. $P(\Omega) = 1, P(\emptyset) = 0$
2. $P(A) \geq 0$ für alle $A \in \mathcal{A}$
3. $P(A \cup B) = P(A) + P(B)$ für alle disjunkten $A, B \in \mathcal{A}$

Die Verteilung ordnet also einfach jedem möglichen Ergebnis ω eine bestimmte Wahrscheinlichkeit $P(\omega)$ zu. Daraus lassen sich auch die Wahrscheinlichkeiten für bestimmte Ereignisse ermitteln.

A.2 Zufallsvariablen

In vielen Zufallsexperimenten interessiert nicht so sehr das Ergebnis ω, sondern eine bestimmte Größe $X(\omega)$, die durch ω bestimmt ist.

Definition A.3 Ist Ω ein Grundraum mit Verteilung P und $\mathcal{M}$ eine beliebige Menge, so nennen wir eine Abbildung $X : \Omega \to \mathcal{M}$ eine ($\mathcal{M}$-wertige) *Zufallsvariable*.

Von welcher Art die Werte aus $\mathcal{M}$ sind, hängt dabei von der Art des Zufallsexperiments ab. Dies können zum Beispiel die gewonnenen (oder verlorenen) Geldbeträge bei einem Glücksspiel sein (dann wäre $\mathcal{M} = \mathbb{R}$), aber auch die Symbole eines Alphabets sind als Wertebereich zulässig (z.B. $\mathcal{M} = \{\text{A, G, C, T}\}$).

A.3 Bedingte Wahrscheinlichkeit

Häufig steht, bevor das Ergebnis eines Zufallsexperiments bekannt ist, schon die Information zur Verfügung, dass das Ergebnis zu einer bestimmten Teilmenge des Stichprobenraums gehört.

Definition A.4 Ist $P(A, B)$ (oder in Mengenschreibweise $P(A \cap B)$) die Wahrscheinlichkeit dafür, dass das Ergebnis eines Zufallsexperiments sowohl dem Ereignis A als auch dem Ereignis B (mit $P(B) > 0$) entspricht, so heißt

$$P(A|B) = \frac{P(A, B)}{P(B)} \tag{A.1}$$

die *bedingte Wahrscheinlichkeit von A unter der Bedingung B*. Im Fall $P(B) = 0$ kann man $P(A|B) = 0$ setzen.

Zum Beispiel ist in einem DNS-Strang die Wahrscheinlichkeit, dass nebeneinander liegende Basen C und G in eine Base T mutieren, relativ groß, so dass die bedingte Wahrscheinlichkeit für das Auftreten eines G nach einem C kleiner ist als die generelle Wahscheinlichkeit für das Auftreten eines G.

Definition A.5 Zwei Ereignisse A, B heißen *unabhängig*, wenn $P(A, B) = P(A)P(B)$ gilt. Dies ist äquvalent zu $P(A|B) = P(A)$ für $P(B) > 0$.

So hängt zum Beispiel beim zweimaligen Werfen eines Würfels das Ergebnis des zweiten Wurfes nicht vom Ergebnis des ersten ab; die beiden Werte sind unabhängig.

A.4 Satz von Bayes

1. (*Formel von der totalen Wahrscheinlichkeit*) $\{B_1, B_2, \ldots\}$ heißt *Zerlegung* von Ω, wenn die B_k disjunkt sind mit $\bigcup_k B_k = \Omega$. Für jede Zerlegung und jedes Ereignis A gilt:

$$P(A) = \sum_k P(B_k)P(A|B_k). \tag{A.2}$$

2. (*Satz von Bayes*[8)]) Ist $P(A) > 0$, so gilt für jedes i

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum_k P(B_k)P(A|B_k)}. \tag{A.3}$$

A.5 Maßzahlen von Verteilungen

Maßzahlen charakterisieren bestimmte Eigenschaften einer Verteilung. Will man beispielsweise einen „mittleren Wert“ einer reelwertigen Zufallsvariable X angeben, so ist es sinnvoll, die Werte $X(\omega)$ ihren Wahrscheinlichkeiten $P(\omega)$ entsprechend zu gewichten.

Definition A.6 Sei P eine Verteilung auf Ω, $X : \Omega \to \mathbb{R}$ eine Zufallsvariable. Dann heißt die Zahl

$$EX = \sum_{\omega \in \Omega} X(\omega)P(\omega)$$

der *Erwartungswert* von X.

Bei einem Glücksspiel etwa drückt der Erwartungswert die durschnittliche Gewinnerwartung auf lange Sicht aus.
Oft ist es auch von Interesse, wie stark sich eine Verteilung um ihren Erwartungswert konzentriert. Hier ist es sinnvoll, den Abstand von $X(\omega)$ zum Erwartungswert entsprechend $P(\omega)$ zu gewichten.

Definition A.7 Hat eine Zufallsvariable X den Erwartungswert EX, so heißt

1. $V(X) = E(X - EX)^2$ die *Varianz* von X.
2. $\sigma_X = \sqrt{V(X)}$ die *Standardabweichung* oder *Streuung* von X.

[8)] T B (1702-1761).

A.6 Verteilungen

Binomialverteilung Eine der einfachsten und bekanntesten (diskreten) Verteilungen ist die *Binomialverteilung*. Sie beschreibt die Wahrscheinlichkeiten bei n-facher Wiederholung eines Zufallsexperiments mit nur zwei möglichen Ergebnissen (z.B. $\omega \in \{0, 1\}$). Ist hierbei p die Wahrscheinlichkeit für $\omega = 1$ und folglich $1 - p$ die Wahrscheinlichkeit für $\omega = 0$, so ist die Wahrscheinlichkeit für k 1en aus n Durchführungen des Zufallsexperiments

$$\mathrm{Bin}_{(n,p)}(k) = \binom{n}{k} p^k (1-p)^{n-k}. \tag{A.4}$$

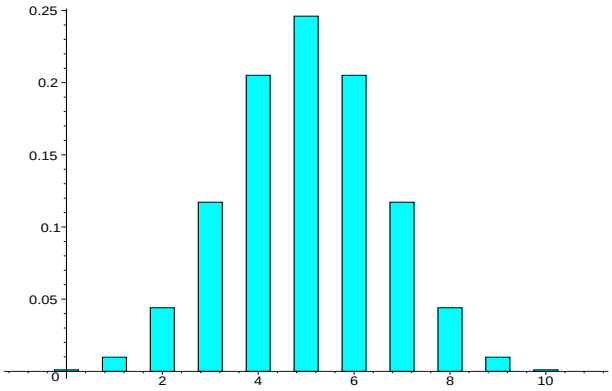


Abbildung 4: Binomialverteilung mit $n = 10$ und $p = 5$

Multinomialverteilung Eine etwas allgemeinere Form der Binomialverteilung mit r unabhänigen Ergebnissen und Wahrscheinlichkeiten $(p_1, \ldots, p_r)$ ist die *Multinomialverteilung*. Die Wahrscheinlichkeit, bei $n = k_1 + k_2 + \ldots + k_r$ Durchführungen des Zufallsexperiments k_1-mal das Ergebnis 1, k_2-mal das Ergebniss 2, ... und k_r-mal das Ergebnis r zu erhalten, ist

$$\mathrm{Mult}_{(n,p_1,p_2,\ldots,p_r)}(k_1, k_2, \ldots, k_r) = \frac{n!}{k_1! k_2! \cdots k_r!} {p_1}^{k_1} {p_2}^{k_2} \cdots {p_r}^{k_r}. \tag{A.5}$$

Normalverteilung Die *Normalverteilung* ist der (stetige) Grenzfall der Binomialverteilung für $n \to \infty$. Für die *Standardnormalverteilung* (Abb. 5) mit Erwartungswert 0 und Varianz 1 gilt:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}. \tag{A.6}$$

Γ-Verteilung Die zu

$$g_{(\alpha,\beta)}(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \quad (x > 0) \tag{A.7}$$

gehörende Verteilung heißt Γ-*Verteilung* mit den Parametern α und β.
Hierbei ist $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx$, $t > 0$.

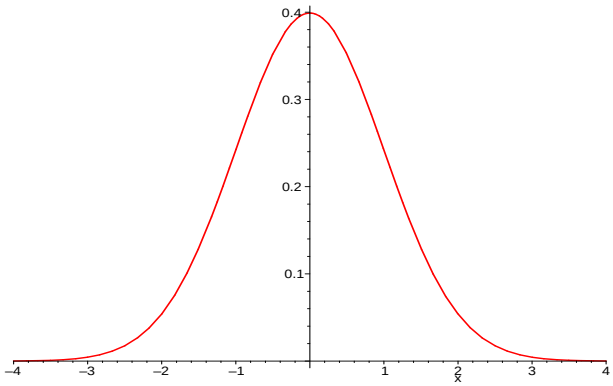


Abbildung 5: Standardnormalverteilung

A.7 Maximum-Likelihood-Schätzer

In der Statistik kann man das bei einer Messung gewonnene Datenmaterial (*Stichprobe*) als Realisierung $x = (x_1, x_2, \ldots, x_n)$ eines Vektors von Zufallsvariablen $X = (X_1, X_2, \ldots, X_n)$ auffassen. Bei einer konkreten Problemstellung werden gewisse Kenntnisse bezüglich der „Rahmenbedingungen“ eines Zufallsexperiments vorhanden sein, aus denen sich Schlüsse über eine mögliche Verteilung P von X ziehen lassen. Ist P bis auf endlich viele Parameter $(\theta_1, \theta_2, \ldots, \theta_m)$ bestimmt, spricht man von einer *parametrischen Verteilungsannahme*. $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ ist dann der unbekannte *Parameter-Vektor* der Verteilung und man schreibt P_θ anstelle von P. Die Menge der möglichen Werte für θ heißt *Parameterbereich* Θ. Meistens gilt $\Theta \subseteq \mathbb{R}^m$.
Welche Parameter unbekannt sind, hängt von der Art der zugrunde gelegten Verteilung ab. So können z.B. der Erwartungswert, die Varianz oder die Wahrscheinlichkeit für ein bestimmtes Ergebnis als unbekannte Paramter in θ auftreten.

Um anhand der vorliegenden Daten ein günstiges θ schätzen zu können, betrachten wir bei einer vorliegenden Stichprobe x die Wahrscheinlichkeit $P(X_1 = x_1, \ldots, X_n = x_n) = P_\theta(x_1) \cdots P_\theta(x_n)$ als Funktion von θ und halten denjenigen Wert θ für den „glaubwürdigsten“, der dem Auftreten der beobachteten Ergebnisse die größte Wahrscheinlichkeit verleiht. Dieses Verfahren nennt man *Maximum-Likelihood-Schätzmethode*.

Definition A.8 Für eine feste Stichprobe $x = (x_1, \ldots, x_n)$ und $\theta \in \Theta$ heißt

$$\theta \mapsto L_x(\theta) = \prod_{i=1}^{n} P_\theta(x_i)$$

die *Likelihood-Funktion* zu x. Wenn L_x einen Maximalwert in $\hat{\theta}(x)$ annimt, also

$$L_x(\hat{\theta}) = \sup\{L_x(\theta) : \theta \in \Theta\} \tag{A.8}$$

gilt, heißt $\hat{\theta}(x)$ *Maximum-Likelihood-Schätzer* von θ.

Häufig gilt $\Theta \subseteq \mathbb{R}^m$, und $\hat{\theta}(x)$ kann durch Differentiation gefunden werden. Dabei ist es zweckmäßig, statt L_x die Funktion $\mathcal{L}_x = \log(L_x)$ zu betrachten, die wegen der Monotonie des Logarithmus das Maximum an der gleichen Stelle hat (*Log-Likelihood-Funktion*).
In der Praxis existiert meistens ein ML-Schätzer $\hat{\theta}$, und er ist in der Regel ein „guter“ Schätzer.

B Algorithmen

Viterbi-Algorithmus

Initialisierung ($i = 0$):	$v_0(0) = 1, v_\mu(0) = 0$ für $\mu > 0$
Rekursion ($i = 1, \ldots, n$):	$v_\lambda(i) = e_\lambda(x_i) \max_\mu \{v_\mu(i-1) p_{\mu\to\lambda}\}$;
	$\text{Zeiger}_i(\lambda) = \operatorname{argmax}_\mu (v_\mu(i-1) e_\mu(\lambda))$
Terminierung:	$P(x, \pi^*) = \max_\mu \{v_\mu(n) p_{\mu\to 0}\}$;
	$\pi^* = \operatorname{argmax}_\pi P(x, \pi)$
Zurückverfolgen ($i = n, \ldots, 1$):	$\pi^*_{i-1} = \text{Zeiger}_i(\pi^*_i)$

Forward-Algorithmus

Initialisierung ($i = 0$):	$f_0(0) = 1$, $f_\mu(0) = 0$ für $\mu > 0$
Rekursion ($i = 1, \ldots, n$):	$f_\lambda(i) = e_\lambda(x_i) \sum_\mu f_\mu(i-1) p_{\mu\to\lambda}$
Terminierung:	$P(x) = \sum_\mu f_\mu(n) p_{\mu\to 0}$

Backward-Algorithmus

Initialisierung ($i = n$):	$b_\mu(n) = p_{\mu\to 0}$ für alle μ
Rekursion ($i = n-1, \ldots, 1$):	$b_\mu(i) = \sum_\lambda e_\lambda(x_{i+1}) b_\lambda(i+1) p_{\mu\to\lambda}$
Terminierung:	$P(x) = \sum_\lambda e_\lambda(x_1) b_\lambda(n) p_{0\to\mu}$

Literatur

[DEKM99] R D , S E , A K , G M . *Biological Sequence Analysis*. Cambridge University Press, Cambridge 1999.

[HeKa00] N H , D K . *Wahrscheinlichkeitstheorie und Statistik für Studierende der Informatik*, 2.Auflage. Vorlesungsskript, Karlsruhe 2000.

[Kren99] U K . *Einführung in die Wahrscheinlichkeitstheorie und Statistik*, 5.Auflage. Vieweg, Göttingen 1999.