Kapitel 2

Theorie eindimensionaler Hidden-Markov-Modelle

2.1 Markov-Quellen

Hidden-Markov-Modelle können als eine Erweiterung der Markov-Quellen angesehen werden. Markov-Quellen emittieren Markov-Ketten, die eine Realisierung von Markov-Prozessen darstellen ([Rab89]). Die Markov-Kette besteht aus diskreten Parameterwerten (z.B. diskreten Zeitwerten) und besitzt die sie kennzeichnende Eigenschaft eines (zeitlich-) eingeschränkten *Gedächtnisses*. Dies bedeutet, daß bei einer Markov-Kette *m*-ter Ordnung eine statistische Abhängigkeit zwischen einem emittierten Symbol und *m* diesem unmittelbar vorausgehenden Symbolen besteht. Aus obiger Annahme folgt unmittelbar die Eigenschaft der *Kausalität*. Der im Zusammenhang mit der vorliegenden Arbeit wichtigste Fall $m = 1$ kann formal folgendermaßen formuliert werden:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots) = P(q_t = S_j | q_{t-1} = S_i) \tag{2.1}$$

In der obigen Gleichung ist mit S_j der j-te sog. *Zustand* der Markov-Quelle bezeichnet, während q_t die Zufallsvariable für die Einnahme eines Zustands aus der Menge der möglichen Zustände $(S_1, \ldots, S_N)$ zum Zeitpunkt t darstellt. Es sei an dieser Stelle angemerkt, daß in diesem Kapitel, wie in der Literatur allgemein üblich, die Markov-Quellen als über die *Zeit* emittierend angesehen wird. Diese Annahme verdeutlicht zunächst die kausalen Eigenschaften der Markov-Quellen, jedoch werden in späteren Kapiteln überwiegend *örtliche* diskrete Werte (z.B. x_k) betrachtet. Die im folgenden gemachten Annahmen bzw. vorgestellten Algorithmen gelten auch für diesen Fall. Jeder der N Zustände der Markov-Quelle emittiert genau ein Symbol $Z = Z(q_t)$ aus dem Symbolalphabet. Der Ausdruck auf der rechten Seite in Gleichung 2.1 definiert eine weitere Größe der Markov-Quelle, nämlich die Übergangswahrscheinlichkeit zwischen zwei Zuständen

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{2.2}$$

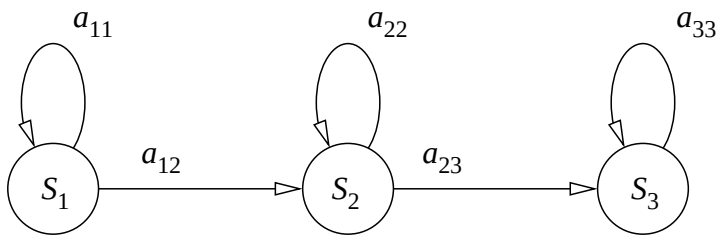

Abbildung 2.1: Graphische Darstellung einer Markov-Quelle mit drei Zuständen

Diese Übergangswahrscheinlichkeiten können zur Übergangsmatrix A zusammengefaßt werden

$$A = \begin{pmatrix} a_{11}, & \dots & ,a_{1N} \\ a_{21}, & \dots & ,a_{2N} \\ \dots, & \dots & ,\dots \\ a_{N1}, & \dots & ,a_{NN} \end{pmatrix} \tag{2.3}$$

Die Addition der Elemente der Zeilen dieser Matrix ergibt stets 1. Matrizen mit den in Gleichung 2.4 angegebenen Eigenschaften werden Markov-Matrizen oder stochastische Matrizen genannt (siehe auch [Dic98]). Es gilt:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{und} \quad a_{ij} \geq 0 \tag{2.4}$$

Die Abbildung 2.1 zeigt eine graphische Darstellung einer Markov-Quelle mit $N = 3$ Zuständen und zugeordneten Übergangswahrscheinlichkeiten. Die Elemente der 3×3 Matrix A in dem hier illustrierten Fall sind *Null* mit Ausnahme der in der Abbildung explizit angegebenen Übergangswahrscheinlichkeiten a_{ij}. Eine solche graphische Darstellung stellt eine Interpretation der Markov-Quelle als einen endlichen statistischen Automaten dar. In der Abbildung ist impliziert, daß der Zustand S_1 der Startzustand des Automaten ist. Bei der vollständigen Definition einer Markov-Quelle muß dies jedoch durch das Festlegen der Wahrscheinlichkeiten für die Startzustände explizit angegeben werden. Diese ist üblicherweise für den Zustand S_j folgendermaßen definiert:

$$\pi_j = P(q_1 = S_j) \quad \text{für} \quad 1 \leq j \leq N \tag{2.5}$$

Ähnlich wie die Übergangswahrscheinlichkeiten a_{ij} bei N Zuständen zu einer $N \times N$ Matrix A zusammengefaßt werden können, können die Wahrscheinlichkeiten für den Startzustand zu einem N-dimensionalen Vektor zusammengefaßt werden.

$$\vec{\pi} = (\pi_1, \dots, \pi_N)^T \tag{2.6}$$

Für den Fall, daß eine Markov-Quelle vollständig definiert ist, d.h. die Anzahl N der Zustände, die jeweilige Symbolausgabe $Z_j(S_j)$, der Vektor $\vec{\pi}$, sowie die Übergangsmatrix A

festgelegt sind, kann die Wahrscheinlichkeit dafür angegeben werden, daß eine vorgegebene Symbolsequenz von dieser Markov-Quelle erzeugt wurde. Sei die Symbolsequenz $O = \{o_1, \ldots, o_T\}$, und sei ferner dieser eine eindeutige Zustandssequenz $Q = \{q_1, \ldots, q_T\}$ zugeordnet, so ist

$$P(O|Modell) = P(Q|Modell) = P(q_1) \cdot \prod_{t=1}^{T} P(q_t|q_{t-1}) \tag{2.7}$$

Dabei ist vorausgesetzt worden, das die Markov-Quelle *stationär* ist, also die Parameter der Quelle nicht von der Zeit abhängig sind. Es ist bei der Markov-Quelle möglich, durch die Beobachtung einer ausgegebenen Sequenz auf deren Zustandsabfolge zu schließen. Anders formuliert, sind die *inneren* Zustände der Quelle von außen sichtbar.

2.2 Hidden-Markov-Modelle

Der Übergang von der Markov-Quelle zum Hidden-Markov-Modell, kurz HMM, geschieht durch die Einführung eines zweiten statistischen Prozesses. In dieser Terminologie ist mit dem *ersten statistischen Prozeß* der Wechsel der Zustände mittels vorgegebener Übergangswahrscheinlichkeiten bezeichnet. Der zusätzlich eingeführte zweite statistische Prozeß ist die Ausgabe von Symbolen über Ausgabeverteilungsfunktionen bzw. Ausgabeverteilungsdichten. Diese Ausgabeverteilungsfunktionen bzw. Ausgabeverteilungsdichten sind den einzelnen Zuständen des Markov-Modells[1] fest zugeordnet. Die Einführung dieses zweiten statistischen Prozesses hat zur Folge, daß im allgemeinen Fall von einer beobachteten Symbolfolge nicht mehr auf die dieser zugeordneten Zustandssequenz zurückgeschlossen werden kann. Der Wegfall der Möglichkeit, die Zustandssequenz eindeutig *aufdecken* zu können, führte zu der Bezeichnung *Hidden*-Markov-Modell.

2.2.1 Modelldefinition

Es ist in der Literatur üblich (siehe z.B. [Rab89, ST95]), je nach Art der Ausgabeverteilung die Markov-Modelle in diskrete und kontinuierliche Modelle einzuteilen. Die Abb. 2.2 zeigt ein Modell mit drei Zuständen und dem jeweiligen Modelltyp zugeordneten Verteilungs- bzw. Dichtefunktionen. In der Abbildung ist ebenfalls der Fall einer Verteilungsfunktion mit der Wahrscheinlichkeit 1.0 für jeweils genau ein Zeichen des Zeichenvorrates angedeutet, der das Markov-Modell in eine Markov-Quelle überführt. Zunächst wird im folgenden das Hidden-Markov-Modell und dessen Algorithmen für diskrete Ausgabeverteilungen vorgestellt. Den kontinuierlichen Markov-Modellen ist ein eigenes Unterkapitel gewidmet (Kap. 2.2.4).

[1] Aus Gründen der besseren Lesbarkeit wird im folgenden oft der Ausdruck *Markov-Modell* anstelle von *Hidden-Markov-Modell* verwendet. Dies ist stets eindeutig, da der den HMMs zugrundeliegende, einfache statistische Prozeß durchgehend mit *Markov-Quelle* bezeichnet wird.

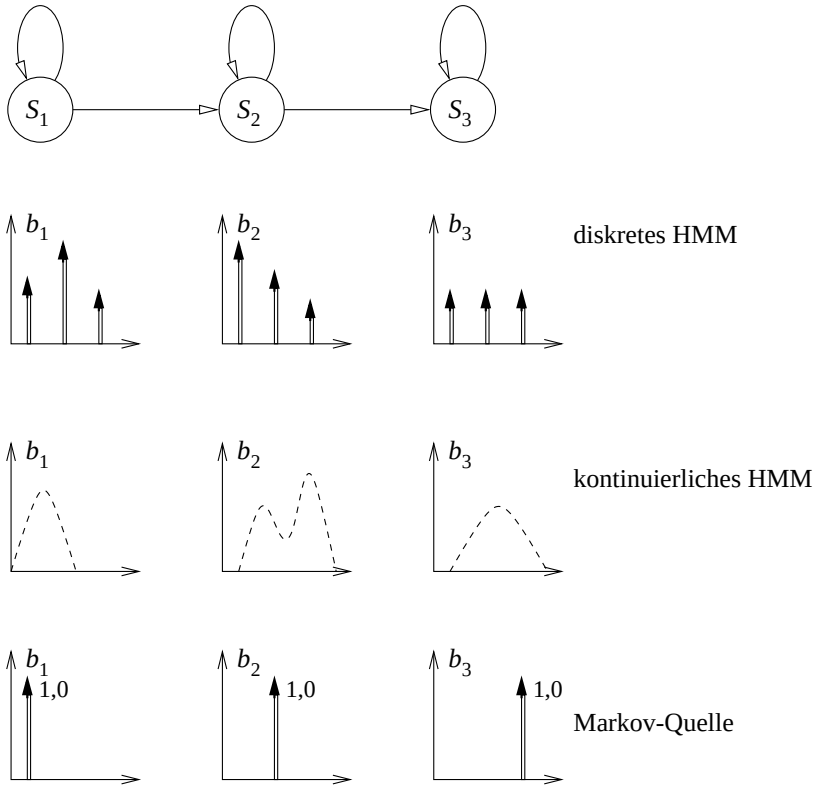

Abbildung 2.2: Ausgabeverteilungsfunktionen und -verteilungsdichten

Bei einem diskreten Markov-Modell gehört zu jedem Modellzustand S_j eine diskrete Ausgabeverteilungsfunktion $b_j(k)$ über einem festgelegten Alphabet der Größe M. Das Alphabet V ist formal folgendermaßen gegeben:

$$V = \{v_1, v_2, \ldots, v_M\} \tag{2.8}$$

Damit ist die Wahrscheinlichkeit für die Ausgabe des Symbols v_k im Zustand S_j

$$b_j(k) = P(v_k | q_t = S_j) \quad \text{für} \quad \begin{matrix} 1 \leq j \leq N \\ 1 \leq k \leq M \end{matrix} \tag{2.9}$$

Sind diese Wahrscheinlichkeiten für alle Zustände und Symbole bekannt, so kann der Vektor der Ausgabeverteilungsfunktionen definiert werden als

$$\vec{b} = (b_1(k), \ldots, b_N(k))^T \quad \text{für} \quad 1 \leq k \leq M \tag{2.10}$$

Die vollständige Definition eines Markov-Modells erfordert also, neben der Festlegung der Quantitäten N und M und des Ausgabealphabets, im wesentlichen die Angabe der Elemente in $\vec{\pi}$,$\vec{b}$ und A. Eine übliche Kurzschreibweise für ein solches Modell ist

$$\lambda = (\vec{\pi}, A, \vec{b}) \tag{2.11}$$

Solch eine Modellfestlegung erfolgt jedoch, ähnlich wie bei künstlichen neuronalen Netzen (KNN) nicht *direkt*, sondern über eine Parameterbestimmung anhand von Beispielen. Dies wird bei KNNs mit Lernphase oder Training bezeichnet, während die Verwendung von trainierten Netzen mit Test oder Recall bezeichnet wird. Rabiner definiert in seinem Tutorial [Rab89] drei Aufgabenstellungen, deren Lösung effiziente Algorithmen für die Trainings- und Testphase liefern. Diese drei Aufgaben sind:

1) Finden einer effizienten Methode zur Berechnung von $P(O|\lambda)$. Dies ist die sog. *Produktionswahrscheinlichkeit*, also die Wahrscheinlichkeit dafür, daß eine Symbolsequenz $O = \{o_1, \ldots, o_T\}$ bei gegebenem Modell $\lambda = (\vec{\pi}, A, \vec{b})$ von diesem Modell ausgegeben wird.

2) Aufdecken der wahrscheinlichsten Zustandssequenz Q, unter der Annahme, daß die Symbolsequenz O von dem Markov-Modell erzeugt wurde.

3) Finden von Parameteradaptionsalgorithmen für $\lambda = (\vec{\pi}, A, \vec{b})$, die $P(O|\lambda)$ maximieren.

Zu allen drei Aufgaben wurden effiziente Lösungen gefunden, die in den folgenden beiden Kapiteln vorgestellt werden. Zunächst werden die Lösungen zu den Aufgabenstellungen 1) und 2) präsentiert, die die wesentlichen Algorithmen für eine Verwendung der HMMs in der Test- bzw. Klassifikationsphase liefern. Die Lösung der Aufgabe 2) wird zudem die Grundlagen für den integrierten Klassifikations- und Segmentierungsansatz bereitstellen, der in den Kapiteln 3 und 5 vorgestellt wird.

2.2.2 Klassifikation

Die Lösung der ersten zuvor definierten Aufgabe, nämlich dem Finden eines effizienten Berechnungsverfahrens für $P(O|\lambda)$, wird es ermöglichen, die Markov-Modelle als Klassifikatoren einzusetzten. Die Produktionswahrscheinlichkeit kann als ein Maß dafür angesehen werden, wie gut eine Symbolsequenz bzw. Observationssequenz[2] zu einem gegebenen Modell paßt. Liegen mehrere Modelle vor, so kann durch die Berechnung der Wahrscheinlichkeiten $P(O|\lambda)$ für die konkurrierenden Modelle das am besten zu der Observation passende Modell ausgewählt werden. Dies erfolgt durch die Entscheidung

$$\mathrm{p}^\star = \operatorname*{argmax}_{\mathrm{p}} \left(P\left(O \,|\, \lambda_{\mathrm{p}} \right) \right). \tag{2.12}$$

Durch Gleichung 2.12 wird die unbekannte Observationssequenz O der Klasse $\mathrm{p}^\star$ zugeordnet.

Der direkte Weg um die gesuchte Wahrscheinlichkeit bei gegebener Observationssequenz $O = \{o_1, \ldots, o_T\}$ zu berechnen ist, die Wahrscheinlichkeiten über alle möglichen Zustands-

[2] Die Bezeichnungen Symbol- und Observationssequenz werden im folgenden gleichwertig verwendet.

sequenzen der Länge T aufzusummieren. Dies ergibt (vgl. auch Gleichung 2.7):

$$P(O|\lambda) = \sum_{\text{alle } Q} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \tag{2.13}$$

Die so gefundene Lösung ist sehr rechenaufwendig, sie erfordert $2T \cdot N^T$ Berechnungen [Rab89]. Der Aufwand der Berechnung steigt also mit der Sequenzlänge T exponentiell an. Der sog. *Forward-Backward-Algorithmus* ermöglicht eine Berechnung der Produktionswahrscheinlichkeit, deren Aufwand linear mit der Sequenzlänge ansteigt. Es handelt sich um einen rekursiven Algorithmus, der die wie folgt definierten *Vorwärtswahrscheinlichkeiten* $\alpha_t(j)$ verwendet:

$$\alpha_t(j) = P(o_1 o_2 \ldots o_t, q_t = S_j|\lambda) \tag{2.14}$$

$P(O|\lambda)$ kann unter Verwendung der Wahrscheinlichkeiten $\alpha_t(j)$ folgendermaßen berechnet werden. Zunächst erfolgt eine Initialisierung durch

$$\alpha_1(j) = \pi_j \cdot b_j(o_1) \quad \text{für} \quad 1 \leq j \leq N \tag{2.15}$$

Anschließend werden iterative Berechnungen mit der folgenden Gleichung durchgeführt:

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right) b_j(o_{t+1}) \quad \text{für} \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq t \leq T-1 \end{array} \tag{2.16}$$

Die gesuchte Wahrscheinlichkeit kann schließlich durch das Summieren der berechneten Vorwärtswahrscheinlichkeiten zum Zeitpunkt T ermittelt werden:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j) \tag{2.17}$$

Der Forward-Backward-Algorithmus erfordert lediglich N^2T Rechenoperationen und stellt somit ein effizientes Berechnungsverfahren zur Bestimmung der Produktionswahrscheinlichkeit dar. Alternativ kann die Produktionswahrscheinlichkeit jedoch auch mit Hilfe der *Rückwärtswahrscheinlichkeiten* $\beta_t(j)$ berechnet werden. Diese sind definiert als:

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, \ldots, o_T|q_t = S_j, \lambda) \tag{2.18}$$

Da bereits ein Algorithmus existiert, der $P(O|\lambda)$ berechnet, müßte dieses Alternativverfahren eigentlich nicht vorgestellt werden. Diese Rückwärtswahrscheinlichkeiten erklären jedoch zum einen den Namen dieses Algorithmus und werden andererseits für das in Kapitel 2.2.3 beschriebene Trainingsverfahren benötigt. Die Rückwärtswahrscheinlichkeiten $\beta_T(j)$ werden initialisiert durch:

$$\beta_T(j) = 1 \quad \text{für} \quad 1 \leq j \leq N \tag{2.19}$$

Die iterative Berechnung ist gegeben durch:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j) \quad \text{für} \quad \begin{array}{l} 1 \leq i \leq N \\ t = T-1, T-2, \ldots, 1 \end{array} \tag{2.20}$$

Und schließlich kann $P(O|\lambda)$ durch folgende Gleichung berechnet werden:

$$P(O|\lambda) = \sum_{j=1}^{N} \beta_1(j) \tag{2.21}$$

Es stehen somit zwei effiziente Verfahren zur Verfügung, die, basierend auf den Vorwärts- und Rückwärtswahrscheinlichkeiten, die Produktionswahrscheinlichkeit mit einem linear mit der Sequenzlänge ansteigenden Aufwand berechnen. Dennoch werden diese Verfahren zur Bestimmung von $P(O|\lambda)$ im Klassifikationsschritt nur selten verwendet. Stattdessen wird der sog. *Viterbi-Algorithmus* ([For73]) verwendet, der eine effiziente Lösung der zweiten formulierten Aufgabenstellung darstellt. Der Viterbi-Algorithmus ermittelt die wahrscheinlichste Zustandssequenz $Q^\star$, unter der Annahme, daß die Observationssequenz von dem Modell λ erzeugt wurde und stellt eine Variante des Verfahrens zur Berechnung der Vorwärtswahrscheinlichkeiten dar. Diese *optimale Zustandssequenz* ergibt sich aus

$$P(O, Q^\star|\lambda) = \max_Q P(O, Q|\lambda) \tag{2.22}$$

Die Wahrscheinlichkeit $P(O, Q^\star|\lambda) = P^\star(O|\lambda)$ kann als ein Näherungswert für die Produktionswahrscheinlichkeit verwendet werden. Bei dem Viterbi-Algorithmus werden statt der Vorwärtswahrscheinlichkeiten $\alpha_t(j)$ die maximal erzielbaren Wahrscheinlichkeiten

$$\vartheta_t(j) = \max_Q(P(o_1 \ldots o_t, q_1 \ldots q_t = S_j|\lambda)) \tag{2.23}$$

definiert (vgl. auch Gleichung 2.14). $\vartheta_t(j)$ ist die höchste Wahrscheinlichkeit der Zustandssequenzen, die im Zustand S_j enden, für die ersten t Observationen. Um die optimale Zustandssequenz zurückverfolgen zu können, wird zudem die Matrix $\psi_t(j)$ verwendet. Der Viterbi-Algorithmus beginnt mit der Initialisierung von ϑ und ψ:

$$\begin{array}{l} \vartheta_1(i) = \pi_i \cdot b_i(o_1) \\ \psi_1(i) = 0 \end{array} \quad \text{für} \quad 1 \leq i \leq N \tag{2.24}$$

Der Rekursionsschritt ist gegeben durch

$$\begin{array}{l} \vartheta_t(j) = \max\limits_{1 \leq i \leq N} (\vartheta_{t-1}(i) \cdot a_{ij})\, b_j(o_t) \\ \psi_t(j) = \operatorname{argmax}\limits_{1 \leq i \leq N} (\vartheta_{t-1}(i) \cdot a_{ij}) \end{array} \quad \text{für} \quad \begin{array}{l} 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \tag{2.25}$$

Schließlich ergibt sich der Näherungswert für die Produktionswahrscheinlichkeit aus

$$\begin{array}{l} P^\star = \max\limits_{1 \leq j \leq N} (\vartheta_T(j)) \\ q_T^\star = \operatorname{argmax}\limits_{1 \leq i \leq N} (\vartheta_T(i)) \end{array} \tag{2.26}$$

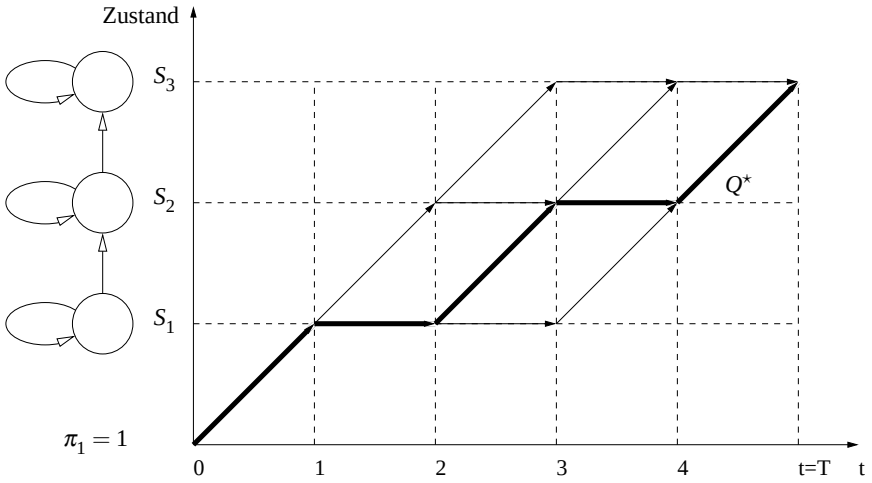

Abbildung 2.3: Mögliche Zustandssequenzen als Pfade in einem Trellis-Diagramm. Der hervorgehobene Pfad repräsentiert die wahrscheinlichste Zustandssequenz $Q^\star$.

Die gesuchte Zustandssequenz $Q^\star$ kann folgendermaßen zurückverfolgt werden:

$$q_t^\star = \psi_{t+1}(q_{t+1}^\star) \quad \text{für} \quad t = T-1,\ldots,1 \tag{2.27}$$

Bei Betrachtung der Gleichungen für $\vartheta_t(j)$ (2.24 bis 2.26) fällt die große Ähnlichkeit zur Berechnung der Produktionswahrscheinlichkeit mit Hilfe der Vorwärtswahrscheinlichkeit $\alpha_t(j)$ auf (siehe Gleichungen 2.15 bis 2.17). Der Hauptunterschied besteht darin, daß in den Gleichungen für den Rekursionsschritt beim Viterbi-Algorithmus der Maximum-Operator anstelle der Summation verwendet wird (vgl. Gleichung 2.25 und 2.16).

Der Viterbi-Algorithmus bzw. die ermittelte optimale Zustandssequenz $Q^\star$ kann in einem sog. *Trellis*-Diagramm veranschaulicht werden ([For73]). Die Abb. 2.3 zeigt beispielhaft für ein sog. Links-Rechts-Modell ein solches Diagramm, das alle möglichen Zustandssequenzen als Pfad darstellt. Nach [Lev83] ist ein Links-Rechts-Modell durch folgende Eigenschaften gekennzeichnet: Die erste Ausgabe erfolgt im ersten Zustand des Modells oder anders formuliert gilt $\pi_1 = P(q_1 = S_1) = 1$. Die Observation zum Zeitpunkt $t = T$, also die letzte Observation der Sequenz wird vom Zustand $q_T = S_N$ des Markov-Modells emittiert. Dieser letzte Zustand S_N wird auch als *absorbierender* Zustand bezeichnet, da er nach dem erstmaligen Erreichen nicht mehr verlassen werden kann ($a_{NN} = 1$). Die Topologie des Links-Rechts-Modells ist, wie in Abb. 2.3 dargestellt, dadurch gekennzeichnet, daß ein einmal verlassener Zustand nicht wieder erreicht werden kann. Diese speziellen Markov-Modelle werden in dieser Arbeit häufig verwendet. Der hervorgehobene Pfad in Abb. 2.3 repräsentiert die, durch den Viterbi-Algorithmus bestimmte, wahrscheinlichste Zustandssequenz.

2.2.3 Training

Die Klassifikation mit den Algorithmen aus Unterkapitel 2.2.2 ist nur dann sinnvoll, wenn die Parameter der Markov-Modelle zuvor auf die Trainingsdaten der jeweiligen Klassenelemente angepaßt wurden. Dies wurde schon in Unterkapitel 2.2.1 als eines der zu lösenden Aufgabenstellungen formuliert. Da es bisher keine analytische Methode gibt, um die Modellparameter direkt zu bestimmen, wird üblicherweise ein iteratives Verfahren verwendet, das als *Baum-Welch-Algorithmus* bekannt ist. Der Baum-Welch-Algorithmus basiert auf der Maximum-Likelihood(ML)-Schätzung der Modellparameter. Es werden, beginnend mit einer Initialisierung, die Modellparameter mit Hilfe des Forward-Backward-Algorithmus neu geschätzt, so daß die Produktionswahrscheinlichkeit $P(O|\lambda)$ in jeder Iteration vergrößert wird. Der Algorithmus verwendet die Rechengröße $\xi_t(i,j)$, die die Wahrscheinlichkeit darstellt, daß sich das Modell zum Zeitpunkt t im Zustand S_i befindet und zum darauf folgenden Zeitschritt in den Zustand S_j wechselt. Dies kann formal folgendermaßen angegeben werden:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{2.28}$$

Die Größe ξ kann unter Verwendung der Vorwärts- und Rückwärtswahrscheinlichkeiten auch dargestellt werden als:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{2.29}$$

Dabei wurden die folgenden zwei Beziehungen verwendet:

$$\alpha_t(j) \cdot \beta_t(j) = P(O, q_t = S_j | \lambda) \tag{2.30}$$

und

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j) \tag{2.31}$$

Ferner wird die Wahrscheinlichkeit $\gamma_t(i) = P(q_t = S_i | O, \lambda)$ definiert als die Wahrscheinlichkeit, daß sich bei gegebener Observationssequenz das Modell zum Zeitpunkt t im Zustand S_i befindet. Diese Wahrscheinlichkeit kann unter Verwendung der Größe ξ angegeben werden als:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.32}$$

Werden diese Wahrscheinlichkeiten $\gamma_t(i)$ über der Zeit t aufsummiert, so ergibt sich eine Größe, die als die Anzahl der Einnahmen des Zustandes S_i interpretiert werden kann. Eine andere Interpretation ist, diese Größe bei einer Summation über der Zeit von $t = 1$ bis $t = T-1$ als die Anzahl der Übergänge, die vom Zustand S_i ausgingen, anzusehen. In analoger Weise kann die Summation über $(0 \leq t \leq T-1)$ von $\xi_t(j,i)$ als die Anzahl der Übergänge vom Modellzustand S_i zum Zustand S_j interpretiert werden. Zusammenfassend ergeben sich daraus die folgenden Schätzformeln für die Modellparameter:

$$\begin{aligned}\hat{\pi}_i &= \text{Häufigkeit der Einnahme des Zustands } S_i \text{ zum Zeitpunkt } t = 1 \\ &= \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.33}$$

$$\begin{aligned}\hat{a}_{ij} &= \frac{\text{Übergänge vom Zustand } S_i \text{ in den Zustand } S_j}{\text{Alle Übergänge aus Zustand } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.34}$$

$$\begin{aligned}\hat{b}_j(y) &= \frac{\text{Aufenthaltswahrscheinlichkeit im Zustand } S_j \text{ wenn } y \text{ emittiert wird}}{\text{Aufenthaltswahrscheinlichkeit im Zustand } S_j} \\ &= \frac{\sum_{\substack{t=1 \\ o_t=y}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} = \frac{\sum_{\substack{t-1 \\ o_t=y}}^{T} \alpha_t(j)\beta_t(j)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{2.35}$$

Eine Alternative zur Berechnung mit dem Baum-Welch-Algorithmus ist der sog. Segmental-KMeans-Algorithmus, der auch Viterbi-Trainings-Algorithmus genannt wird. Beim Viterbi-Training wird anders als beim Baum-Welch-Algorithmus nicht die Größe $P(O|\lambda)$ bei jedem Iterationsschritt vergrößert, sondern die Zielgröße $P^\star(O|\lambda)$ (siehe auch Gleichung 2.22). Diese Größe ist der bereits eingeführte Näherungswert für die Produktionswahrscheinlichkeit, der auf der optimalen Zustandsfolge $Q^\star$ basiert. Der Algorithmus beginnt mit der Auswahl der Modellparameter für ein Startmodell λ^0. Anschließend werden Iterationen mit folgenden Einzelschritten durchgeführt:

1) Bestimmen der Zustandsfolge $Q^\star$ mit dem Viterbi-Algorithmus

$$P(O, Q^\star|\lambda^{(n-1)}) = \max_Q P(O, Q|\lambda^{(n-1)}) \tag{2.36}$$

2) Es werden die folgenden Häufigkeiten bestimmt:

$$\overline{\pi}_i = \chi_{[q_1^\star = S_i]} \tag{2.37}$$

$$\overline{a}_{ij} = \sum_{t=1}^{T-1} \chi_{[q_t^\star = S_i, q_{t+1}^\star = S_j]} \tag{2.38}$$

$$\overline{b}_{jk} = \sum_{t=1}^{T} \chi_{[q_t^\star = S_j, o_t = v_k]} \tag{2.39}$$

3) Normierung durch

$$\hat{\pi}_i = \frac{\overline{\pi}_i}{\sum_{i=1}^{N} \overline{\pi}_i} \tag{2.40}$$

$$\hat{a}_{ij} = \frac{\overline{a}_{ij}}{\sum_{i=1}^{N} \overline{a}_{ij}} \tag{2.41}$$

$$\hat{b}_j(v_k) = \frac{\overline{b}_j(v_k)}{\sum_{k=1}^{K} \overline{b}_j(v_k)} \tag{2.42}$$

4) Übernehmen der neuen Modellparameter

$$\lambda^{(n)} = (\hat{\pi}_i, \hat{a}_{ij}, \hat{b}_j(v_k)) \tag{2.43}$$

5) Gehe zu Schritt 1)

Die in den Gleichungen zur Bestimmung der Häufigkeiten (2.37 bis 2.39) verwendete Funktion χ ist die sog. Kronecker-Delta-Funktion. Eine detailliertere Darstellung des Segmental-KMeans-Algorithmus, einschließlich der Untersuchung des Konvergenzverhaltens, findet sich in den Arbeiten [Jua90] und [Rab76].

2.2.4 Kontinuierliche Ausgabefunktionen

Bisher wurden diskrete Markov-Modelle betrachtet, die Observationssequenzen der Form $O = \{o_1, \ldots, o_T\}$ emittieren, mit Elementen o_t, die aus einem festgelegten Alphabet V stammen. Sollen reellwertige Vektorsequenzen mit Markov-Modellen trainiert bzw. klassifiziert werden, so können die Vektoren der Sequenz $\{\vec{o}_1, \ldots, \vec{o}_T\}$ durch eine Vektorquantisierung in ein diskretes Symbolalphabet z.B. durch den K-Means Algorithmus überführt werden und somit die Algorithmen der Kapitel 2.2.2 und 2.2.3 weiter verwendet werden. Alternativ können die Vektorsequenzen durch Markov-Modelle mit kontinuierlichen Ausgabeverteilungen auch direkt modelliert werden. Der Vorteil hierbei liegt in dem Wegfall des Quantisierungsschrittes, der stets zu einer Verzerrung der Eingabedaten und einem damit verbundenen

Informationsverlust führt ([ST95]). Aus diesem Grund wurden in der vorliegenden Arbeit kontinuierliche Modelle verwendet. Die Dichtefunktionen eines Zustands S_j der kontinuierlichen Modelle sind üblicherweise als Gaußsche Mischverteilungsdichten der Form

$$b_j(\vec{o}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\vec{o}, \vec{\mu}_{jm}, \Sigma_{jm}) \quad \text{mit} \quad \sum_{m=1}^{M} c_{jm} = 1 \tag{2.44}$$

gegeben. Dabei ist c_{jm} die Gewichtung der m-ten Mischungskomponente, M die Anzahl der Mischungskomponenten und $\mathcal{N}$ eine multivariate Gaußdichte. Mit einer solchen Überlagerung von hinreichend vielen Gaußdichten können beliebige Dichtefunktionen approximiert werden. Die Gaußdichte ist für D-dimensionale Größen folgendermaßen angebbar:

$$\mathcal{N}(\vec{o}, \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \cdot e^{-\frac{1}{2}(\vec{o}-\vec{\mu})^T \Sigma^{-1} (\vec{o}-\vec{\mu})} \tag{2.45}$$

Der Vektor der Mittelwerte ist in obiger Gleichung mit $\vec{\mu}$ bezeichnet, die Kovarianzmatrix bzw. deren Inverse mit Σ und Σ^{-1}.

Die Gleichungen für die Parameteranpassungen an die Trainingssequenzen können unter Verwendung der Wahrscheinlichkeit $\zeta_t(j,k)$, daß die Mischungskomponente $m_t = k$ im Zustand S_j zur Zeit t ausgewählt wurde, abgeleitet werden. Dies entspricht der folgenden formalen Formulierung

$$\zeta_t(j,k) = P(q_t = S_j, m_t = k | \vec{O}, \lambda) \tag{2.46}$$

Die Wahrscheinlichkeiten $\zeta_t(j,k)$ können ähnlich wie die Größen γ und ξ durch die Vorwärts- und Rückwärtswahrscheinlichkeiten α und β bestimmt werden. Schließlich ergeben sich die folgenden Schätzformeln (aus [ST95]):

$$\hat{c}_{jk} = \frac{1}{\sum_{t=1}^{T} \sum_{k=1}^{M} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \tag{2.47}$$

$$\hat{\mu}_{jk} = \frac{1}{\sum_{t=1}^{T} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \cdot \vec{o}_t \tag{2.48}$$

$$\hat{\Sigma}_{jk} = \frac{1}{\sum_{t=1}^{T} \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \cdot (\vec{o}_t - \mu_{jk})(\vec{o}_t - \mu_{jk})^T \tag{2.49}$$

Bei dem Training eines kontinuierlichen Markov-Modells werden neben den Gleichungen 2.47 bis 2.49 auch in unveränderter Weise die Baum-Welch-Formeln für die Größen $\hat{\pi}$ (Gleichung 2.33) und $\hat{a}_{ij}$ (Gleichung 2.34) verwendet.

In späteren Kapiteln dieser Arbeit wird auch die folgende Variante der Modellierung der Dichtefunktion der Modellzustände (vgl. Gleichung 2.44) verwendet:

$$b_j(\vec{o}) = \prod_{s=1}^{S} b_{js}(\vec{o}_s)^{\gamma_s} \tag{2.50}$$

In der Gleichung 2.50 werden S sog. Merkmalströme (engl. Streams) verwendet, die bei großen, inhomogenen Merkmalvektoren Vorteile aufweisen ([Gup97]). Merkmalströme sind ein oder mehrere Komponenten des Merkmalvektors, die als statistisch unabhängig angenommen werden und denen sog. Merkmalstrom-Gewichtungen γ_s zugeordnet werden. Es wird beispielsweise oft in der automatischen Spracherkennung der vieldimensionale Merkmalvektor unterteilt in Komponenten, die auf die gleiche Weise berechnet wurden. Ein Merkmalstrom enthält nach dieser Unterteilung ausschließlich cepstrale Koeffizienten, zwei weitere die Differenzen dieser Koeffizienten bzw. Differenzen höherer Ordnung und schließlich ein Merkmalstrom die Komponenten, die aus der Signalenergie berechnet wurden. Eine detaillierte Beschreibung dieser Aufteilung in Merkmalströme findet sich in [Neu98].

2.2.5 Aspekte der Implementierung

Im Kontext dieser Arbeit wurde überwiegend das sog. Hidden Markov Toolkit (HTK) der Cambridge University verwendet (siehe z.B. [You94]). Diese Software ist auf die Verwendung für die automatische Spracherkennung ausgerichtet und mußte mithin für die Erkennung von Bildern und Bildinhalten angepaßt werden. Aus der Ausrichtung auf die Spracherkennung ergibt sich der Bedarf, aus einzelnen Markov-Modellen komplexere Strukturen aufbauen zu können. So sollen beispielsweise aus *Phonem*-basierten Markov-Modellen mittels einer Phonemisierungstabelle Worte gebildet werden können und aus diesen dann wiederum ganze Sätze. Dieses Aneinanderhängen von Modellen wird durch die Einführung von nichtemittierenden Zuständen jeweils *vor* und *hinter* dem eigentliche Modell ermöglicht. Die Übergangswahrscheinlichkeiten vom ersten nichtemittierenden Zustand zu den emittierenden Zuständen stellen eine alternative Formulierung der schon erwähnten Wahrscheinlichkeiten der Anfangszustände $\pi_j = P(q_1 = S_j)$ dar. Die Wahrscheinlichkeit π_1 für die Einnahme des Zustandes S_1 zum Zeitpunkt $t = 1$ kann beispielsweise bei Verwendung des nichtemittierenden Zustandes S_0 als Übergangswahrscheinlichkeit folgendermaßen dargestellt werden:

$$\pi_1 = P(q_1 = S_1) = a_{01} = P(q_1 = S_1 | q_0 = S_0) \tag{2.51}$$

Trotz der Verwendung zweier zusätzlicher Zustände soll in den folgenden Kapiteln jedoch wie bisher unter einem Modell mit N Zuständen ein Modell mit N *emittierenden* Zuständen verstanden werden. Die nichtemittierenden Zustände und damit die Möglichkeiten zur Verkettung von Modellen sind für die in den folgenden Kapiteln dargestellten Methoden wichtig.

2.2.6 Bayes Netze

Die in der Literatur am häufigsten gewählte graphische Darstellungsweise von Markov-Modellen ist die Darstellung als finiter statistischer Automat (siehe auch Abb. 2.2). Einer

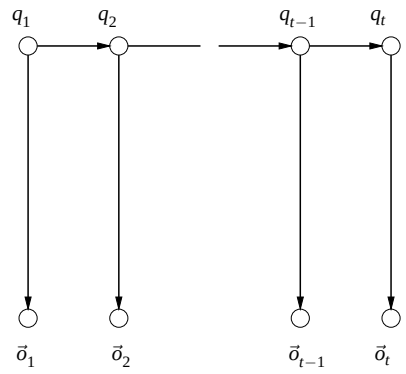

Abbildung 2.4: Darstellung des Hidden-Markov-Modells als dynamisches Bayes-Netz

solchen Darstellung ist vor allem die Topologie des Modells zu entnehmen und mithin die vorhandenen Parameter. So ist zum Beispiel in Abb. 2.2 zu sehen, daß es sich bei dem dargestellten Markov-Modell um ein sog. Links-Rechts-Modell mit drei Zuständen handelt. Impliziert wird zudem, daß es sich um ein Markov-Modell erster Ordnung handelt, da die Verbindungspfeile üblicherweise die Übergangswahrscheinlichkeiten und somit die Wahrscheinlichkeiten $P(q_t = S_j | q_{t-1} = S_i)$ repräsentieren. Eine alternative graphische Darstellung der Markov-Modelle bei der die statistischen Abhängigkeiten verdeutlicht werden, ist in Abb. 2.4 gezeigt und findet sich ebenfalls in [Smy97, Luc95, Mur00]. Die Abb. 2.4 stellt das Hidden-Markov-Modell als gerichtetes graphisches Modell oder dynamisches Bayes-Netz (DBN, engl. Dynamic Bayesian Network) dar. DBNs sind graphische Modelle von statistischen Prozessen und stellen die Abhängigkeiten zwischen der Observationssequenz und den Zufallsvariablen des Modells dar. Die Zufallsvariablen sind in Abb. 2.4 durch die Knoten des Graphen gegeben. Die horizontalen und vertikalen Verbindungslinien in der Abbildung repräsentieren die statistischen Abhängigkeiten, die durch die Annahme eines Markov-Prozesses erster Ordnung impliziert werden. Insbesondere stellt bei DBNs die Abwesenheit von Verbindungen in den Graphendarstellungen statistische Unabhängigkeitsannahmen dar. Dynamische Bayes-Netze stellen eine Obermenge einer Vielzahl statistischer Modelle dar, die aus unterschiedlichen wissenschaftlichen Disziplinen stammen. Beispiele für solche statistischen Modelle sind neben den Markov-Modellen, Kalman-Filter und probabilistische Experten-Systeme ([Smy97]). Die DBNs wiederum gehören der übergeordneten Klasse von sog. *graphischen Modellen* an, bei denen wahrscheinlichkeitstheoretische Ansätze mit der Graphentheorie verbunden werden. Neben den Bayes-Netzen sind die in der Physik und der Computer-Vision sehr populären ungerichteten Graphen Mitglieder dieser übergeordneten Familie. Zur Klasse der ungerichteten Graphen gehören z.B. die *Markov-Random-Fields* und die *Boltzmann-Maschine*. Mit Hilfe der graphischen Modelle ist es möglich, Gemeinsamkeiten zwischen den in unterschiedlichen wissenschaftlichen Disziplinen entwickelten Modellen und Algorithmen zu entdecken und zu nutzen. Dies wird sehr ausführlich in dem Übersichtsartikel von Murphy in [Mur00] behandelt.

In der vorliegenden Arbeit werden die graphischen Modelle genutzt, um die Zusammenhänge zwischen den in Kapitel 4.1 vorgestellten Markovschen Zufallsfeldern (engl. Markov-Random-Fields), dem in Kapitel 5.4.2 verwendeten Kalman-Filter und den bereits vorgestellten Hidden-Markov-Modellen (einschließlich der zweidimensionalen Variante in Kapitel 4.3) zu erläutern.

2.3 Kapitelzusammenfassung

Es wurde in die Theorie der eindimensionalen Hidden-Markov-Modelle eingeführt. Diese Modelle stellen die dominierende Klassifikationsmethode für zeitlich veränderliche Muster, insbesondere Sprachmuster, dar. Es stehen sehr effiziente Algorithmen für die Berechnung der Produktionswahrscheinlichkeiten, die für die Klassifikation benötigt werden, und das Modelltraining zur Verfügung. Diese Berechnungsvorschriften sind der Viterbi- bzw. der Baum-Welch-Algorithmus, die beide in diesem Kapitel vorgestellt wurden. Der Viterbi-Algorithmus berechnet nicht die Produktionswahrscheinlichkeit selbst, sondern einen Approximationswert, der auf der wahrscheinlichsten Zustandssequenz basiert. Der Algorithmus hat für die folgenden Kapitel eine wichtige Bedeutung, da er die Grundlage für integrierte Segmentierungs- und Klassifikationsverfahren darstellt. Dies ergibt sich aus der Tatsache, daß der Viterbi-Algorithmus die Klassifikation durch die Bestimmung eines Schätzwertes für die Produktionswahrscheinlichkeit erlaubt und im selben Schritt eine Segmentierung durch das Aufdecken der wahrscheinlichsten Zustandsabfolge ermöglicht. Diese kombinierte Segmentierung und Klassifikation wird im folgenden Kapitel genutzt, um bei gedrehten Objekten in Bildern die Orientierung herauszufinden und diese zu erkennen.