Lernalgorithmen

SoSe 2008 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Lernalgorithmen Gesamtübersicht

0. Einführung
1. Identifikation (aus positiven Beispielen)
2. Zur Identifikation regulärer Sprachen, mit XML-Anwendung
3. HMM — Hidden Markov Models
4. Lernen mittels Anfragen & zur Roboterorientierung
5. Lernen mit negativen Beispielen
6. PAC-Lernen

Hidden Markov Modelle HMM

Quellen

L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77 (1989), pp. 257–286, siehe auch `http://www.cs.ubc.ca/~murphyk/Software/HMM/rabiner.pdf`

R. Durbin, S. Eddy, A. Krogh, G. Mitchinson. Biological sequence analysis. Cambridge University Press, 1998.

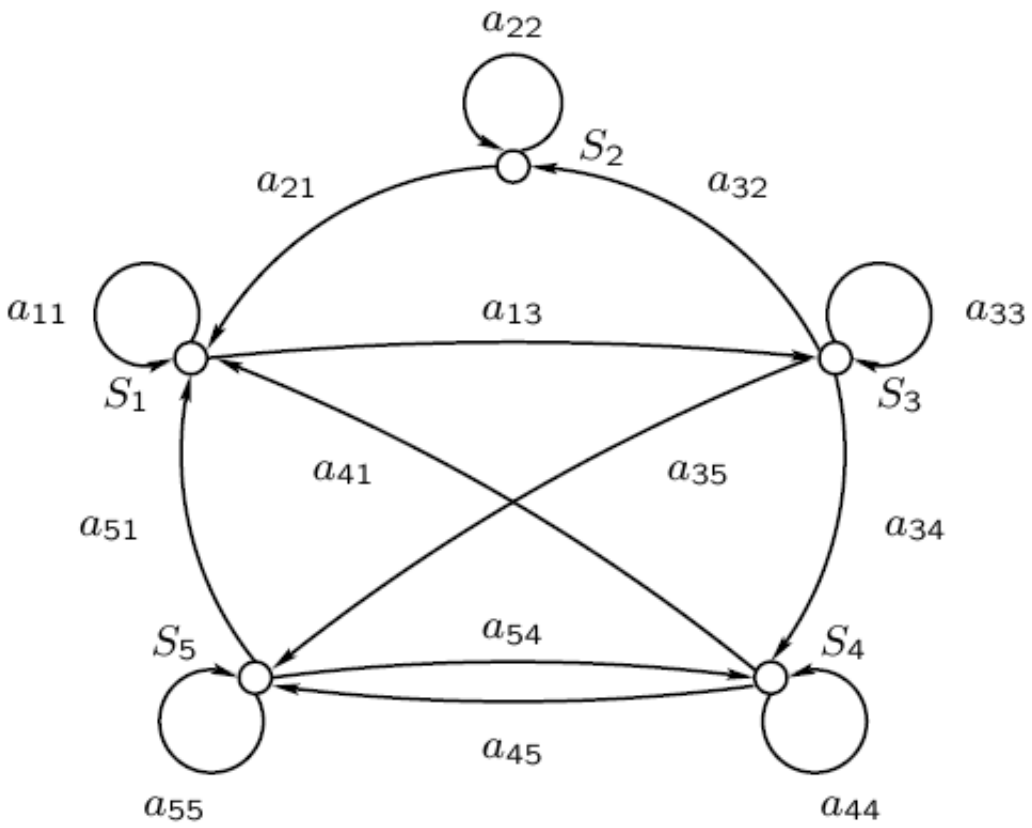
Etwas Wahrscheinlichkeitsrechnung

Ein *Wahrscheinlichkeitsexperiment* (z.B. Würfeln) wird bestimmt durch eine Menge von *Elementarereignis*sen $X = \{x_1, \ldots, x_m\}$ (z.B.: Es wurde eine 6 gewürfelt), die mit einer gewissen *Wahrscheinlichkeit* $P(x_i)$ eintreten können.
Dabei wird $P(x_i) \in [0, 1]$ und $\sum_{i=1}^{m} P(x_i) = 1$ vorausgesetzt.
Die Wahrscheinlichkeit eines *Ereignis*ses $Y \subseteq X$ ist: $\sum_{y \in Y} P(y)$.
Sind $A, B \subseteq X$ zwei Ereignisse mit $A \cap B \neq \emptyset$, so bestimmt sich die Wahrscheinlichkeit dafür, dass A eintritt, vorausgesetzt, B trifft zu, nach Bayes durch

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

$P(A|B)$ heißt auch *bedingte Wahrscheinlichkeit*.
Für die *gemeinsame Wahrscheinlichkeit* von A und B schreibt man oft $P(A, B)$ statt $P(A \cap B)$.
Die Ereignisse A und B heißen *unabhängig*, falls $P(A) = P(A|B)$ gilt (mit $P(B) \in (0, 1)$).
Für unabhängige Ereignisse gilt ferner $P(A, B) = P(A)P(B)$.

Diskrete Markov-Prozesse



Markov-Ketten erster Ordnung

N Systemzustände $S_1, \ldots, S_N$
q_t sei der Zustand des Systems zum Zeitpunkt t.
Markov-Ketten erster Ordnung erfüllen:

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

Daher setzen wir:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), \quad 1 \leq i, j \leq N \quad (2)$$

Dafür gilt:

$$a_{ij} \geq 0 \quad (3)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad (4)$$

Ein Beispiel für ein Markov-Modell mit drei Zuständen

Wetterbeobachtungen gemäß folgender Klassifikation:

Zustand 1: Niederschlag
Zustand 2: bewölkt
Zustand 3: sonnig.

Die (angenommenen) Zustandsübergangswahrscheinlichkeiten seien:

$$A = (a_{ij}) = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}.$$

Frage: Was ist die Wahrscheinlichkeit, dass die "Wetterfolge" der nächsten 7 Tage Sonne-Sonne-Niederschlag-Niederschlag-Sonne-Wolken-Sonne ist?
Formal betrachten wir die *Observationsfolge* $O = S_3S_3S_3S_1S_1S_3S_2S_3$ an den Zeitpunkten $t = 1, \ldots, 8$.
Unter den gemachten Modellannahmen berechnet sich die Wahrscheinlichkeit für O durch:

$$\begin{aligned}
P(O|\text{Modell}) &= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|\text{Modell}) \\
&= P(S_3) \cdot P(S_3|S_3) \cdot P(S_3|S_3) \cdot P(S_1|S_3) \cdot P(S_1|S_1) \cdot P(S_3|S_1) \\
&\quad \cdot P(S_2|S_3) \cdot P(S_3|S_2) \\
&= \pi_3 \cdot a_{33} \cdot \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\
&= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}$$

Dabei bezeichnen wir die Anfangszustandswahrscheinlichkeiten durch:

$$\pi_j = P(q_1 = S_i), \quad 1 \leq i \leq N.$$

Frage: Wie groß ist die Wahrscheinlichkeit dafür, dass das Wetter genau d Tage lang dasselbe bleibt?
Wir diskutieren dazu die Wahrscheinlichkeit für Observationsfolgen der Art $O = S_i^d S$, $S \neq S_i$.
Diese beträgt:

$$P(O|\text{Modell}, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d)$$

Der Erwartungswert für die Verweildauer in Zustand S_i berechnet sich deshalb durch:

$$\begin{aligned} \bar{d}_i &= \sum_{d=1}^{\infty} d p_i(d) \\ &= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}} \end{aligned}$$

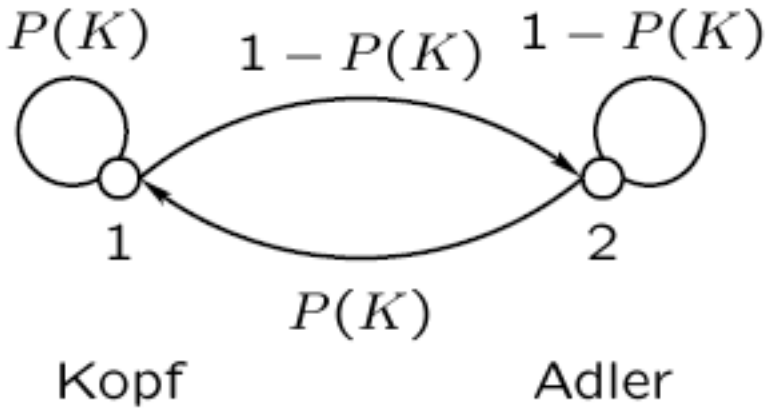
Versteckte Markov-Modelle – Einleitung

JETZT: Zustände nicht mehr direkt beobachtbar

Beispiel: Münzwurfexperiment

Hinter einem Vorhang steht eine Person, die uns Ergebnisse eines (angeblichen) Münzwurfexperimentes mitteilt. Wir kennen nur die Ergebnisse, wissen aber nicht, wie sie zustande gekommen sind. Für diese Szenerie könnten wir verschiedene HMMs ersinnen, z.B.

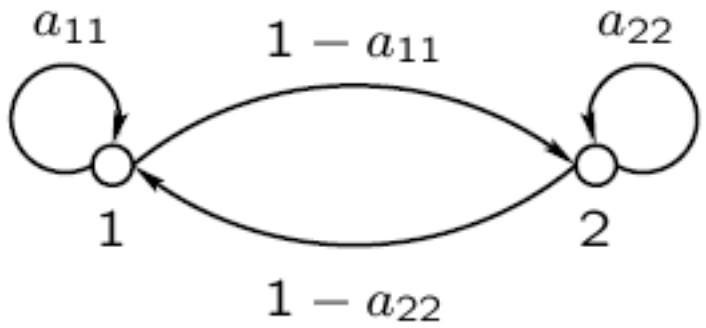
Modell A: Münzwurfexperiment mit einer möglicherweise gefälschten Münze. Dies führt auf ein (observables) MM mit 2 Zuständen:



O = KKAAKAKKAAK ...

S = 11221211221 ...

Modell B: zwei unterscheidbare Münzen werden verwendet.



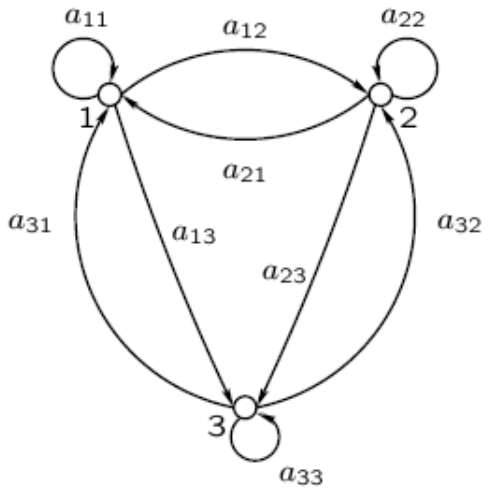
O = KKAAKAKKAAK ...

S = 21122212212 ...

$P(K) = P_1$ $\quad$ $P(K) = P_2$

$P(A) = 1 - P_1$ $\quad$ $P(A) = 1 - P_2$

Modell C: drei unterscheidbare Münzen werden verwendet.




```
O = KKAAKAKKAAK ...
S = 31233112313 ...
```

Zustand	1	2	3
$P(K)$	P_1	P_2	P_3
$P(A)$	$1-P_1$	$1-P_2$	$1-P_3$

Elemente eines HMM

1. Die Anzahl N der Zustände des Systems (häufig physikalisch deutbar).

2. Die Anzahl M verschiedener beobachtbarer Ausgabezeichen je Zustand. Ausgabezeichenalphabet: $V = \{v_1, \ldots, v_M\}$.

3. Die Zustandsübergangswahrscheinlichkeitsverteilungen sind durch eine Matrix

$$A = (a_{ij} \mid 1 \leq i, j \leq N)$$

gegeben, wobei

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$$

ist.

4. Wahrscheinlichkeitsverteilungen der beobachtbaren Ausgabezeichen in Zustand j wird gegeben durch

$$B = \{b_j(k) \mid 1 \leq j \leq N, 1 \leq k \leq M\}.$$

5. Die Anfangszustandsverteilung $\pi = (\pi_i \mid 1 \leq i \leq N)$ ist definiert durch $\pi_i = P(q_1 = S_i)$.

Da N und M durch B implizit spezifiziert werden, kann ein HMM durch das Tripel

$$\lambda = (A, B, \pi)$$

vollständig angegeben werden.

Urnenmodell eines HMM

Urne 1		Urne 2		...	Urne N	
$P(\text{rot})$	$= b_1(1)$	$P(\text{rot})$	$= b_2(1)$		$P(\text{rot})$	$= b_N(1)$
$P(\text{blau})$	$= b_1(2)$	$P(\text{blau})$	$= b_2(2)$		$P(\text{blau})$	$= b_N(2)$
$P(\text{grün})$	$= b_1(3)$	$P(\text{grün})$	$= b_2(3)$		$P(\text{grün})$	$= b_N(3)$
$P(\text{gelb})$	$= b_1(4)$	$P(\text{gelb})$	$= b_2(4)$		$P(\text{gelb})$	$= b_N(4)$
$\vdots$		$\vdots$			$\vdots$	
$P(\text{orange})$	$= b_1(M)$	$P(\text{orange})$	$= b_2(M)$		$P(\text{orange})$	$= b_N(M)$

Problemstellungen bei HMM

Problem 1: Gegeben sei eine Observationsfolge $O = O_1 \dots O_T$ und ein HMM-Modell $\lambda = (A, B, \pi)$. Wie berechnet man $P(O \mid \lambda)$?

Problem 2: Gegeben sei eine Observationsfolge $O = O_1 \dots O_T$ und ein HMM-Modell $\lambda = (A, B, \pi)$. Wie sieht eine/die Zustandsfolge $Q = q_1 \dots q_T$ aus, die O "am besten" erklärt?

Problem 3: Wie kann man die Modellparameter von λ bei gegebener Observationsfolge O optimieren, um $P(O \mid \lambda)$ zu maximieren?
Dies ist das eigentliche *Lernproblem* beim HMM.

Eine Lösung zu Problem 1

Wir betrachten eine fixierte Zustandsfolge

$$Q = q_1 q_2 \dots q_T.$$

Hierbei ist q_1 der Anfangszustand. Die Wahrscheinlichkeit für die Beobachtungsfolge $O = O_1 \dots O_T$ unter Annahme der Zustandsfolge Q ist:

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda),$$

wobei wir die statistische Unabhängigkeit der Beobachtungen voraussetzen. $\rightsquigarrow$

$$P(O|Q,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T).$$

Die Wahrscheinlichkeit für solch eine Zustandsfolge Q lässt sich ermitteln durch:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}.$$

Die Wahrscheinlichkeit dafür, dass im gewählten Modell λ der Beobachtungsfolge O die Zustandsfolge Q entspricht, ist nach dem Bayesschen Gesetz:

$$P(O, Q|\lambda) = P(O|Q, \lambda)P(Q, \lambda).$$

Daher errechnet sich die Wahrscheinlichkeit für O durch:

$$\begin{aligned} P(O|\lambda) &= \sum_{\text{alle } Q} P(O|Q, \lambda) \\ &= \sum_{q_1, q_2, \cdots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(Q_T). \end{aligned}$$

Eine naive Implementierung dieser Formel liefert ein Verfahren, bei dem $O(TN^T)$ viele Additionen und Multiplikationen auszuführen sind.

Ein geschicktere Vorgehensweise liefert die im folgende dargestellte **Vorwärts-prozedur**: Dazu betrachten wir die *Vorwärtsvariable* $\alpha_t(i)$:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda)$$

$\alpha_t(i)$ lässt sich mit folgendem Verfahren induktiv bestimmen:

1. Initialisierung:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N.$$

2. Induktionsschritt:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \quad 1 \leq j \leq N.$$

S_j kann zum Zeitpunkt $t+1$ von jedem der N Zustände S_i zum Zeitpunkt t erreicht werden. $\alpha_t(i)a_{ij}$ ist die Wahrscheinlichkeit dafür, dass $O_1 \dots O_t$ beobachtet wurde, während das System zum Zeitpunkt t im Zustand S_i und zum Zeitpunkt $t+1$ im Zustand S_j ist.
Der Faktor $b_j(O_{t+1})$ berücksichtigt die gemachte Beobachtung zum Zeitpunkt $t+1$.

3. Terminierung:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

Nach Definition ist nämlich

$$\alpha_T(i) = P(O_1 O_2 \cdots O_T, q_T = S_i|\lambda).$$

Die Vorwärtsprozedur benötigt lediglich $O(N^2T)$ viele Rechenschritte. Damit haben wir eine effiziente Methode zur Lösung von Problem 1 gefunden.

In ähnlicher Weise berechnet man die sogenannte *Rückwärtsvariable* $\beta_t(i)$:

$$\beta_t(i) = P(O_{t+1}O_{t+2}\cdots O_T|q_t = S_i, \lambda).$$

Prozedur:

1. Initialisierung:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

2. Induktionsschritt:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1, \quad 1 \leq i \leq N$$

Auch diese Rückwärtsprozedur braucht $O(N^2T)$ Rechenschritte.

Zwei Lösungen zu Problem 2:

Problem: Aufgabenstellung ist ungenau. . .

Frage: Was für ein Optimalitätskriterium wird für eine “passende” Zustandsfolge gewählt?

1. Wähle Zustände aus, die *individuell* gesehen am wahrscheinlichsten erscheinen. Dazu:

$$\gamma_t(i) = P(q_t = S_i|O, \lambda).$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)};$$

Der individuell wahrscheinlichste Zustand ist daher:

$$q_t = \text{argmax}_{1\leq i\leq N}[\gamma_t(i)], \quad 1 \leq t \leq T$$

Problem: $a_{ij} = 0$ für gewisse i, j!

2. Wähle “wahrscheinlichsten Pfad”.

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda),$$

m.a.W., $\delta_t(i)$ besitzt die höchste Wahrscheinlichkeit unter allen Pfaden zur Zeit t, welche die ersten t Beobachtungen berücksichtigen und in Zustand S_i enden. Die “nächsten” Werte dieser Größe lassen sich mit

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1})$$

bestimmen. Um die auf diese Weise definierte Zustandsfolge ebenfalls zu erhalten, müssen wir uns zu jedem t und i einen Pfad merken, der $\delta_t(i)$ maximiert. Dazu speichern wir in einem Array $\psi(t, j)$ einen Zustand ab, der $\delta_t(i) a_{ij}$ maximiert.

Viterbi-Algorithmus

1. Initialisierung: $\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$

2. Induktionsschritt:

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \\ \psi_t(j) &= \text{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \end{aligned}$$

3. Terminierung:

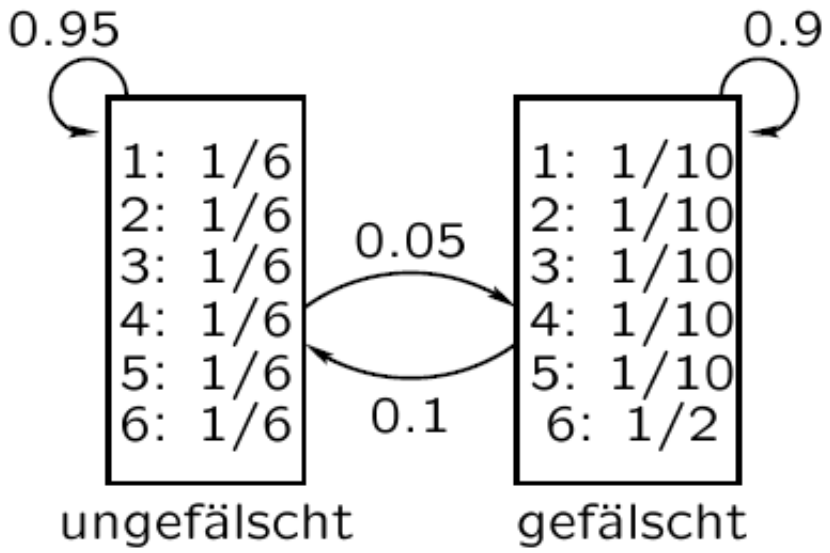
$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} [\delta_T(i)] \\ q_T^* &= \text{argmax}_{1 \leq i \leq N} [\delta_T(i)] \end{aligned}$$

4. Der so konstruierte Pfad per Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1.$$

Hinweis: Vorlesung über Informationstheorie

Beispiel: In einem Casino wird manchmal falsch gespielt



Der Viterbi-Algorithmus liefert – im Vergleich zur tatsächlichen Würfelfolge:

```
Ergebnis   3151162464466442453113216311641521336251445436316566265666666
Würfel     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUGGGGGGGGGGGGGGG
Viterbi    UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUGGGGGGGGGGGGG

Ergebnis   651166453132651245636664631636663162326455236266666625151631
Würfel     GGGGGGUUUUUUUUUUUUGGGGGGGGGGGGGGGGUUUGGGGGGGGGGGGGGGUUUUUUUUUU
Viterbi    GGGGGGUUUUUUUUUUUUGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGUUUUUUUUU

Ergebnis   222555441666566563564324364121513465146353411126414626253356
Würfel     UUUUUUUUGGGGGGGGGGGGGUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUGG
Viterbi    UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUG

Ergebnis   366163666466232534413661661163252562462255265252266435353336
Würfel     GGGGGGGGUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
Viterbi    GGGGGGGGGGGGUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU

Ergebnis   233121625364414432335163243633665562466662632666612355245242
Würfel     UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUGGGGGGGGGGGGGGGGGGGGGGGUUUUUUUUUUU
Viterbi    UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUGGGGGGGGGGGGGGGGGGGGGGUUUUUUUUUUU
```

Beispiel: Bestimmung von CG-Inseln

Das Dinukleotid CG (Cytosin gefolgt von Guanin) im menschlichen Genom recht selten auf, denn Mutation von C zu T wahrscheinlich.

In den sogenannten CG-Inseln findet dieser Prozess nicht statt.

Experimentell ermittelte Nukleotidübergangswahrscheinlichkeiten innerhalb von CG-Inseln (markiert durch $+$) und außerhalb:

$+$	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

$-$	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

Bei der Abfolge CGCG ergibt sich daher:

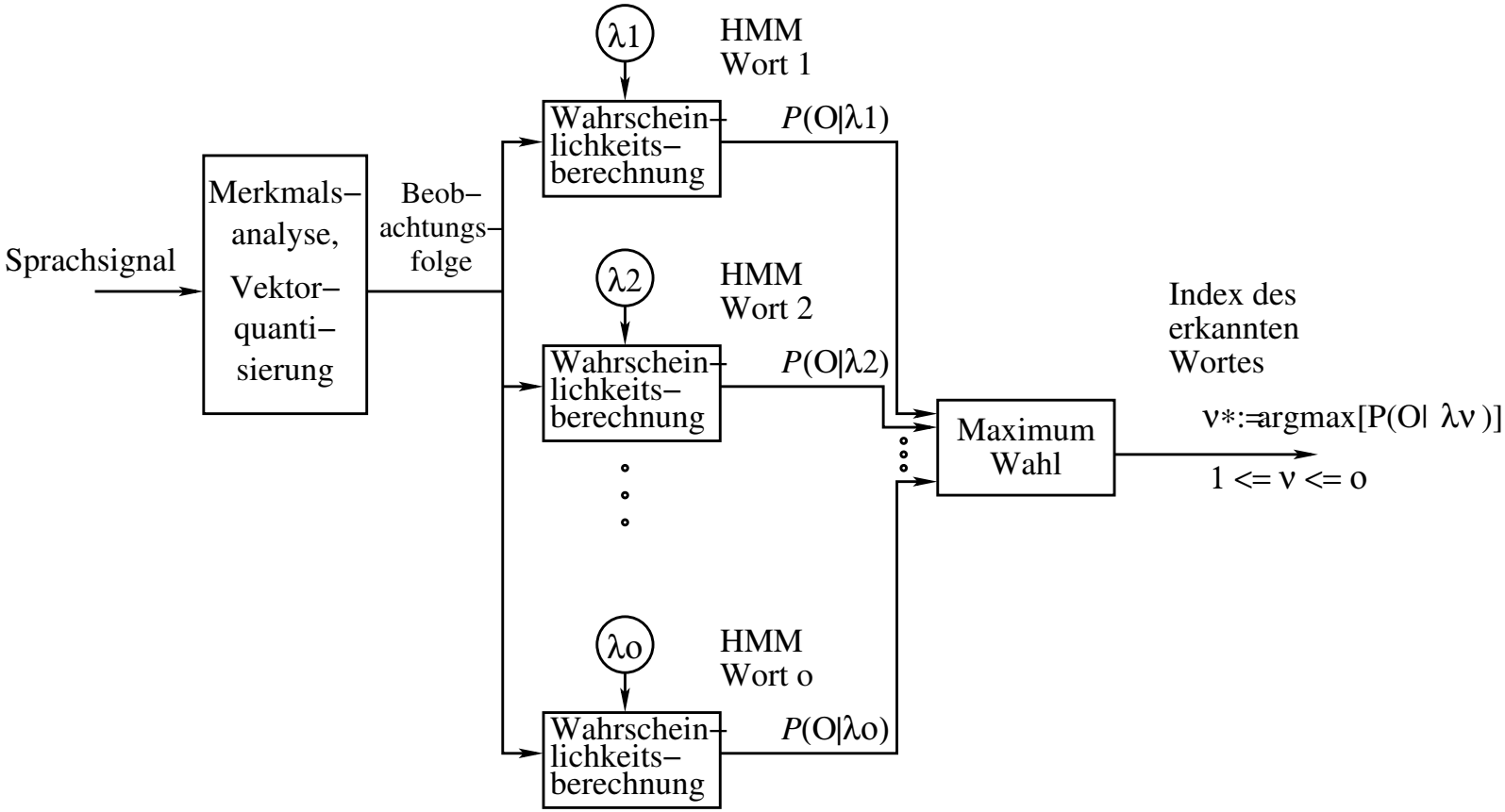
		C	G	C	G
Startzustand	1	0	0	0	0
A_+	0	0	0	0	0
C_+	0	**0.13**	0	**0.012**	0
G_+	0	0	**0.34**	0	**0.0032**
T_+	0	0	0	0	0
A_-	0	0	0	0	0
C_-	0	0.13	0	0.0026	0
G_-	0	0	0.010	0	0.00021
T_-	0	0	0	0	0

Also können wir schließen, dass die Folge CGCG sich ganz in einer CG-Insel befindet.

Einsatz von HMM bei der Spracherkennung

1. Isolierung einzelner Wörter mit HMMs: Zustände “Sprechpause” und “Sprache”

2. Erkennung von Einzelwörtern: für jedes Wort ν aus dem Wörterbuch (bestehend aus V verschiedenen Wörtern) ist ein spezielles HMM λ^{ν} zu erstellen (Problem 3!)


λ1
HMM
Wort 1
Wahrschein-
lichkeits-
berechnung
P(O|λ1)
Merkmals-
analyse,
Vektor-
quanti-
sierung
Beob-
achtungs-
folge
Sprachsignal
λ2
HMM
Wort 2
Wahrschein-
lichkeits-
berechnung
P(O|λ2)
Index des
erkannten
Wortes
Maximum
Wahl
ν*:=argmax[P(O| λν)]
1 <= ν <= o
λo
HMM
Wort o
Wahrschein-
lichkeits-
berechnung
P(O|λo)

Nach dieser “Trainingsphase” wird das System wie folgt eingesetzt:

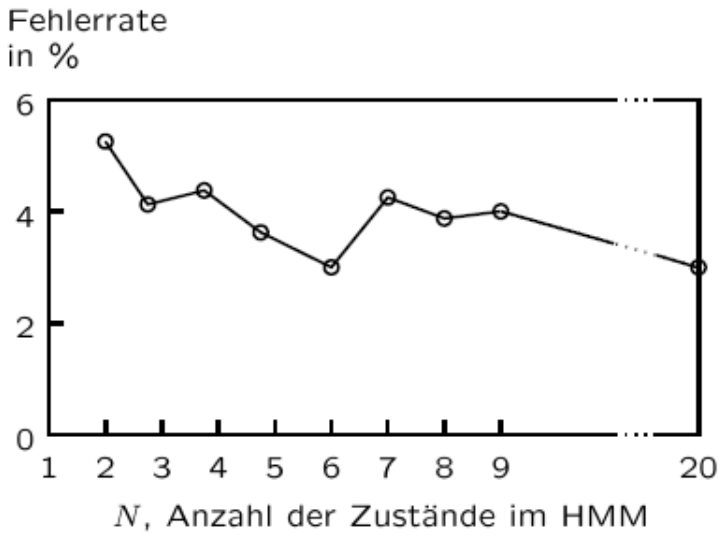
1. Ein einkommendes Sprachsignal (eines unbekannten, zu analysierenden Einzelwortes) wird, z.B. mit der LPC Merkmalsanalyse (LPC=Linear predictive coding) in eine Beobachtungsfolge O übersetzt.

2. Für jedes HMM $\lambda\nu$ wird nun die Wahrscheinlichkeit $P(O \mid \lambda\nu)$ errechnet.

3. Es wird dasjenige Wort ν^* “erkannt”, dessen Wahrscheinlichkeit $P(O \mid \lambda\nu^*)$ maximal ist.

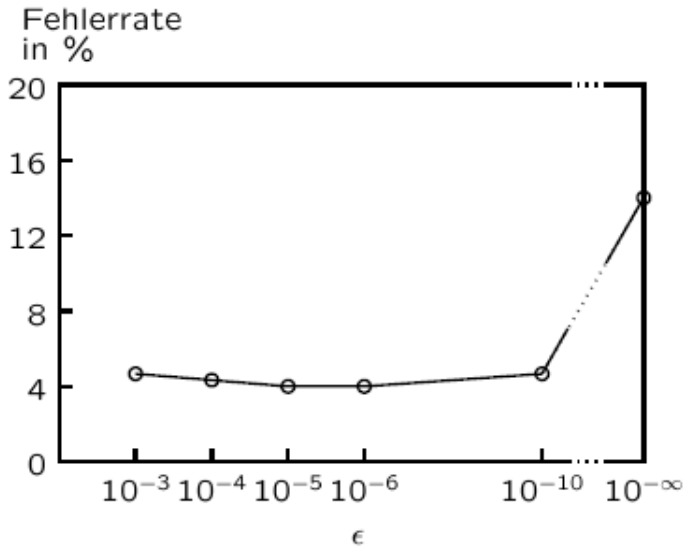
Zur Abschätzung von $P(O \mid \lambda\nu)$ wird meist der Viterbi-Algorithmus eingesetzt $\leadsto$ der wahrscheinlichste Pfad wird betrachtet.

Parameterschätzung am Beispiel: Zustandszahl der HMMs



Allzuviel Zustände bringen also keine wesentliche Verbesserung des Modells mehr.

Parameterschätzung am Beispiel: Mindestwahrscheinlichkeiten bei HMMs



Allzu geringe Mindestwahrscheinlichkeiten sind also kontraproduktiv.