Kapitel 9

Hidden Markov Modelle (HMMs)

- Markov-Ketten
- Von der Markov-Kette zum HMM
- HMM Topologien
- Drei klassische Algorithmen für HMMs
- HMMs in der Praxis und Anwendungen

HMM

Der Ursprung der HMMs liegt in der Spracherkennung. Mittlerweile finden sie jedoch in praktisch allen Bereichen der statistischen ME Anwendung.

- HMMs werden typischerweise dort eingesetzt, wo die Länge des Merkmalsvektors von einem Muster zum anderen variiert
- Da HMMs Segmentierung und Klassifikation in integierter Form durchführen, bieten sie auch dort Vorteile, wo es um die Klassifikation komplexer Muster, die aus einfacheren Komponenten zusammengesetzt sind, geht. Z.B. Erkennung handgeschriebener Wörter; jedes Wort besteht aus einer unterschiedlichen Zahl von Buchstaben.

HMMs: Verallgemeinerung homogener Markov-Ketten mit einem Zufallsprozess auf zwei Zufallsprozesse

Markov-Ketten (1)

Ein diskreter stochastischer Prozess $\{q_t|t = 1, 2, \ldots\}$ über einer endlichen Menge von Zuständen $Q = \{S_1, \ldots, S_L\}$ heisst homogene Markov-Kette g.d.w. für alle Zufallsvariablen $q_t \in Q$ der Folge $q_1 \ldots q_{t-1}q_t$ gilt:

$$P(q_t|q_1 \ldots q_{t-2}q_{t-1}) \;=\; P(q_t|q_{t-1})$$

d.h. nur der unmittelbar vorangehende Zustand übt einen Einfluss auf den nachfolgenden Zustand aus (Markov-Eigenschaft).

Mathematical way of saying:

Today is the first day of the rest of your life

Eigenschaft: Homogene Markov-Kette ist stationär, d.h. die absolute Zeit t spielt keine Rolle.

Markov-Ketten (2)

(Zustands-)Übergangswahrscheinlichkeiten:

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

Darstellung durch $L \times L$ Matrix

$$\boldsymbol{A} \;=\; [a_{ij}]_{L \times L}$$

Es gilt: $(\forall i, j)(a_{ij} \geq 0)$ und $(\forall i)(\sum_{j=1}^{L} a_{ij} = 1)$

Wird für die Markov-Kette nicht verlangt, dass sie in einem fest definierten Zustand S_i startet, werden zusätzliche initiale Wahrscheinlichkeiten benötigt:

$$\boldsymbol{\Pi} = (\pi_1, \ldots, \pi_L), \quad \pi_i = P(q_1 = S_i), \quad \sum_{i=1}^{L} \pi_i = 1$$

Das stochastische Verhalten einer homogenen Markov-Kette ist durch die Parameter $(\boldsymbol{\Pi}, \boldsymbol{A})$ vollständig definiert

Markov-Ketten (3)

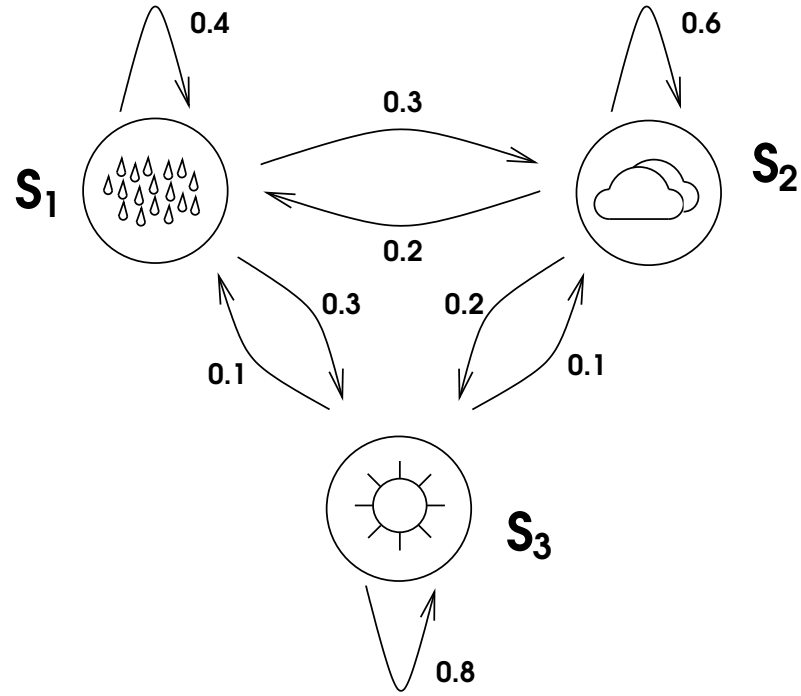
Beispiel: Wetterprognose

$Q = \{S_1, S_2, S_3\}$, d.h. $L = 3$ (Zeiteinheit: Tag)

- S_1 = Regen
- S_2 = Bewölkung
- S_3 = Sonnenschein

Übergangswahrscheinlichkeiten aus Beobachtungen:

$$\boldsymbol{A} = [a_{ij}] = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$



Markov-Ketten (4)

Mit dem Modell verschiedene Fragen beantworten, z.B.

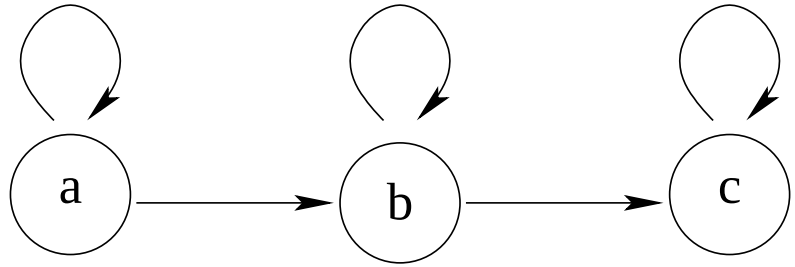
> *Wie gross ist die Wahrscheinlichkeit, dass nach einem sonnigen Tag das Wetter an den folgenden sieben Tagen „sonnig, sonnig, regnerisch, regnerisch, sonnig, bewölkt, sonnig“ ist?*

Wahrscheinlichkeit der Zustandsfolge:

$$\begin{aligned} P(S_3S_3S_1S_1S_3S_2S_3|S_3) &= P(S_3|S_3)P(S_3|S_3)P(S_1|S_3)P(S_1|S_1) \\ &\quad P(S_3|S_1)P(S_2|S_3)P(S_3|S_2) \\ &= a_{33}a_{33}a_{31}a_{11}a_{13}a_{32}a_{23} \\ &= 1.536 \cdot 10^{-4} \end{aligned}$$

Markov-Ketten (5)

Beispiel: Sprachsignal der Buchstaben abc



Man kennt die Zustände a/b/c allerdings nicht. Bekannt sind lediglich die Signale bzw. die abgeleteten Merkmale.

Von der Markov-Kette zum HMM (1)

Annahme: zu jedem Zeitpunkt t nicht nur ein (unbekannter) Zustand eingenommen, sondern auch ein Symbol aus einem endlichen Alphabet

$$V = \{v_1, \ldots, v_K\}$$

ausgegeben

Der Beobachter sieht nur die generierte Symbolfolge

$$\boldsymbol{O} = O_1 \ldots O_T, \quad O_i \in V$$

Die Zustände, welche zur Ausgabe der Symbolfolge $\boldsymbol{O}$ geführt haben, sind unbekannt bzw. verborgen (deshalb der Name *Hidden* Markov Model)

Verhalten: Die Ausgabe der Symbole O_t zufällig und nur vom Zustand q_t abhängig (insb. unabhängig von früher angenommenen Zuständen und früher ausgegebenen Symbolen):

$$P(O_t|O_1 \ldots O_{t-1}, q_1 \ldots q_{t-1} q_t) = P(O_t|q_t)$$

In jedem Zustand S_j kann jedes Symbol v_k ausgegeben werden. Wahrscheinlichkeiten für die Ausgabe von v_k in Zustand S_j:

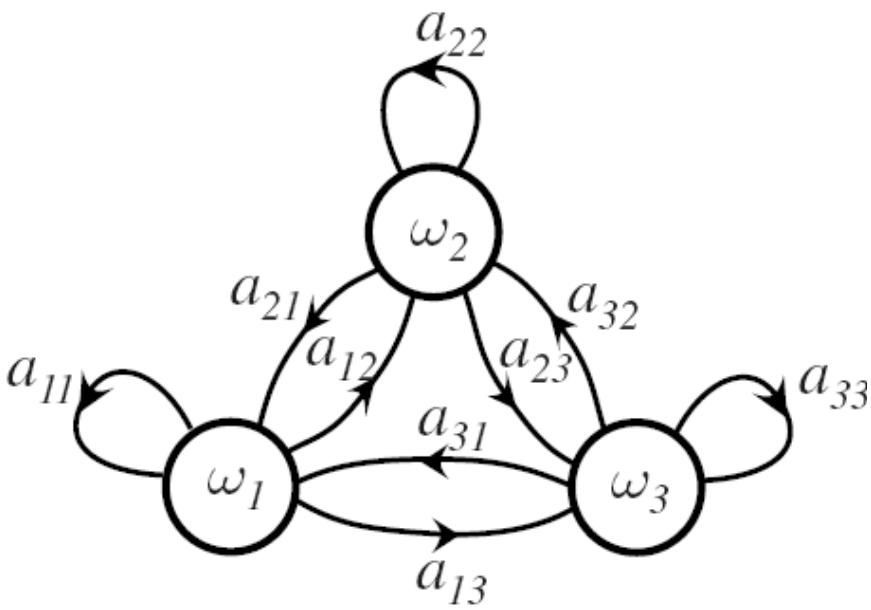
$$\boldsymbol{B} = [b_{jk}]_{L \times K}, \quad b_{jk} = P(v_k|S_j)$$

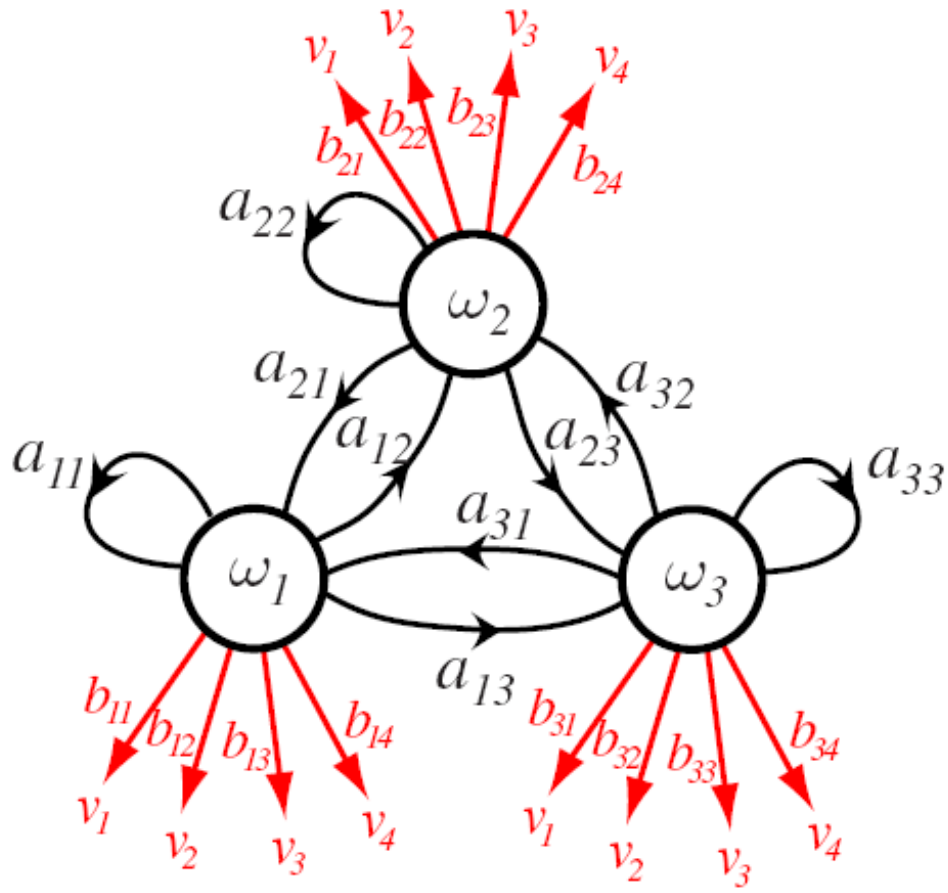
Es gilt: $(\forall j, k)(b_{jk} \geq 0)$ und $(\forall j)(\sum_{k=1}^{K} b_{jk} = 1)$.

HMM: durch Grössen $\lambda = (\boldsymbol{\Pi}, \boldsymbol{A}, \boldsymbol{B})$ vollständig definiert

Von der Markov-Kette zum HMM (2)

Vergleich Markov-Kette und HMM (Zustände mit ω_i dargestellt):





Von der Markov-Kette zum HMM (3)

HMM-Arbeitsweise bei Generierung einer Symbolfolge $\boldsymbol{O}=O_1 \ldots O_T$:

1. Setze $t = 1$ und wähle einen Initialzustand $q_1 = S_i$ unter Berücksichtigung von $\boldsymbol{\Pi}$
2. Wähle ein Beobachtungssymbol $O_t = v_k$ unter Berücksichtigung von $P(v_k|q_t)$ aus der Matrix $\boldsymbol{B}$
3. Falls $t < T$ gehe in Zustand $q_{t+1} = S_j$ über unter Berücksichtigung der Matrix $\boldsymbol{A}$. Sonst beende den Prozess.
4. Setze $t = t + 1$. Gehe zu 2.

Von der Markov-Kette zum HMM (3)

HMM-Arbeitsweise bei Generierung einer Symbolfolge $\boldsymbol{O}=O_1 \ldots O_T$:

1. Setze $t = 1$ und wähle einen Initialzustand $q_1 = S_i$ unter Berücksichtigung von $\boldsymbol{\Pi}$
2. Wähle ein Beobachtungssymbol $O_t = v_k$ unter Berücksichtigung von $P(v_k|q_t)$ aus der Matrix $\boldsymbol{B}$
3. Falls $t < T$ gehe in Zustand $q_{t+1} = S_j$ über unter Berücksichtigung der Matrix $\boldsymbol{A}$. Sonst beende den Prozess.
4. Setze $t = t + 1$. Gehe zu 2.

Es gibt noch eine andere Variante von HMMs, bei welcher die Ausgabe nicht in einem Zustand, sondern beim Übergang zwischen zwei Zuständen erfolgt. Man kann zeigen, dass beide Varianten äquivalent sind.

Von der Markov-Kette zum HMM (4)

Unterscheidung von HMMs anhand der Art von Ausgabesymbolen:

- Diskrete HMMs:
  Anzahl möglicher Ausgabesymbole endlich
  Ausgabewahrscheinlichkeiten durch Matrix $\boldsymbol{B}$ gegeben
- Kontinuierliche HMMs:
  z.B. Ausgabe mit Merkmalsvektor $x \in \Re^n$
  Ausgabewahrscheinlichkeiten in kontinuierlicher Form; Verteilungsdichte $p(x)$

HMM Topologien (1)

Allgemein: Übergang von jedem Zustand in jeden anderen erlaubt

Spezialfälle: einzelne $a_{ij} = 0$ (Übergang von S_i nach S_j nicht möglich)

a) *Ergodisches Modell*
   Allgemeinste Topologie (jeder Zustand von jedem anderen prinzipiell erreichbar; keine Restriktionen bezüglich der Startzustände)

b) *Links-Rechts Modell*
   Nur Übergange in Zustände mit Index grösser oder gleich dem Index des aktuellen Zustandes erlaubt (Ein einmal verlassener Zustand nie mehr eingenommen)

c) *Bakis-Modell*
   Spezialfall des links-Rechts Modells (Von Zustand S_i aus nur S_i, S_{i+1} und S_{i+2} erreichbar; Ausnahme: letzte beide Zustände)

d) *Lineares Modell*
   Nur Verbleib im aktuellen Zustand oder Übergang zum Zustand mit nächst höherem Index möglich

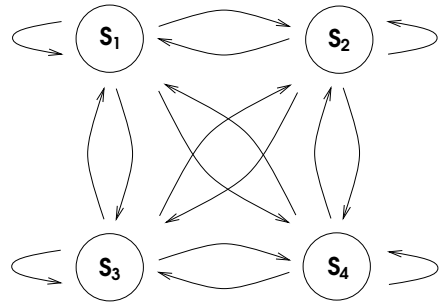
Modelle b-d insb. für Modellierung zeitabhängiger Prozesse geeignet
Sprach- und Handschriftenerkennung: insb. Modelle c und d

Von Modell a zu b, c und d: zunehmend weniger Parameter $\Longrightarrow$ wichtige Rolle beim Trainieren eines HMMs

HMM Topologien (2)

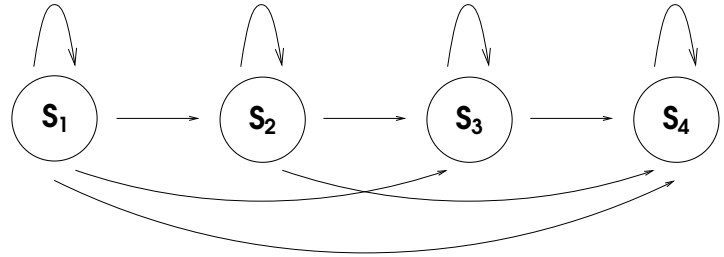
Ergodisches Modell: $\boldsymbol{A} =$

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*



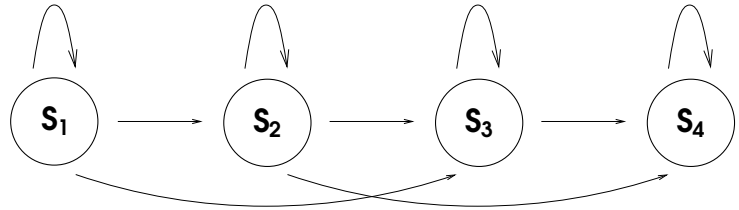
Links-Rechts Modell: $\boldsymbol{A} =$

*	*	*	*
	*	*	*
		*	*
			*



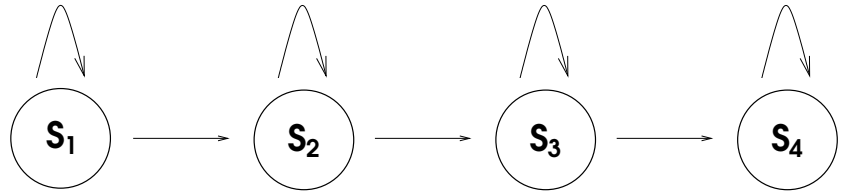
Bakis-Modell: $\boldsymbol{A} =$

*	*	*	
	*	*	*
		*	*
			*



Lineares Modell: $\boldsymbol{A} =$

*	*		
	*	*	
		*	*
			*



Drei klassische Algorithmen für HMMs

- Evaluation Problem:
  Gegeben: Beobachtungsfolge $\boldsymbol{O} = O_1 \dots O_T$, Modell $\lambda = (\boldsymbol{\Pi}, \boldsymbol{A}, \boldsymbol{B})$
  Wie kann die Wahrscheinlichkeit $P(\boldsymbol{O}|\lambda)$, mit der $\boldsymbol{O}$ von λ erzeugt wird, berechnet werden?

  $\Longrightarrow$ Muster (durch Folge $\boldsymbol{O}$ von Merkmalsvektoren gegeben) demjenigen Modell λ_i aus $\{\lambda_1, \dots, \lambda_N\}$ von Modellen zuordnen, für welches $P(\boldsymbol{O}|\lambda_i)P(\lambda_i)$ maximal ist, d.h.

$$\lambda_i \;=\; \arg \max_{\lambda_j} \{P(\boldsymbol{O}|\lambda_j)P(\lambda_j) \mid j = 1, \dots, N\}$$

- Decoding Problem:
  Gegeben: $\boldsymbol{O}$ und λ wie bei Problem 1
  Welche Folge $Q^* = q_1 \dots q_T$ von Zuständen kann $\boldsymbol{O}$ optimal erklären?

$$Q^* \;=\; \max_{Q} \; P(Q|\boldsymbol{O}, \lambda)$$

  $\Longrightarrow$ Zusammen mit Erkennung die Segmentierung eines komplexen Musters in einfachere Komponenten vornehmen

- Training Problem:
  Lernen der Parameter: Wie können die statistischen Parameter $\boldsymbol{\Pi}$, $\boldsymbol{A}$ und $\boldsymbol{B}$ anhand von einer Stichprobe bestimmt werden?

Drei klassische Algorithmen für HMMs: Evaluation Problem (1)

Berechnung der Wahrscheinlichkeit $P(O = O_1 \dots O_T|\lambda)$; einfachheitshalber nur $P(O)$

Es gilt:

$$P(O) = \sum_{\text{alle Zustandsfolgen } w=w_1 w_2 \dots w_T} P(O|w)P(w)$$

wobei

$$P(w) = \prod_{t=1}^{T} P(w_t|w_{t-1});\ (P(w_1|w_0) = \Pi(w_1),\ P(w_t|w_{t-1}), t \geq 2,\ \text{aus Matrix } \boldsymbol{A})$$

und

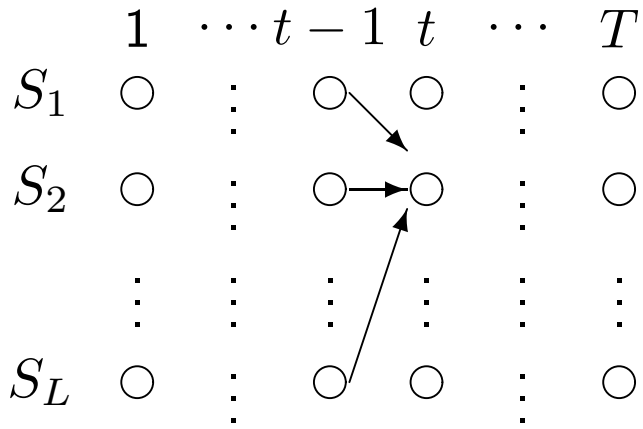
$$P(O|w) = \prod_{t=1}^{T} P(O_t|w_t); \quad (P(O_t|w_t) \text{ aus Matrix } \boldsymbol{B})$$

Gesuchte Wahrscheinlichkeit $P(O)$:

$$P(O) = \sum_{\text{alle Zustandsfolgen } w=w_1 w_2 \dots w_T} \prod_{t=1}^{T} P(O_t|w_t)P(w_t|w_{t-1})$$

Eine direkte Umsetzung dieser Formel führt zu einer Komplexität $O(L^T T)$

Drei klassische Algorithmen für HMMs: Evaluation Problem (2)



$\alpha[t][q]$: Wahrscheinlichkeit für Situation
"zum Zeitpunkt t im Zustand q; $O_1 O_2 \ldots O_t$ generiert"

Recursive Forward Evaluation Algorithm (Komplexität: $O(L^2 T)$)

---

for $(q \in Q = \{S_1, S_2, \ldots, S_L\})$ $\alpha[0][q]$=1;
for $(t = 1;\ t \leq T;\ t{+}{+})$
  for (alle $w \in Q$)

$$\alpha[t][w] = \sum_{\text{alle } q \in Q} \left(\alpha[t-1][q] \cdot a_{qw} \right) \cdot b_{w O_t};$$

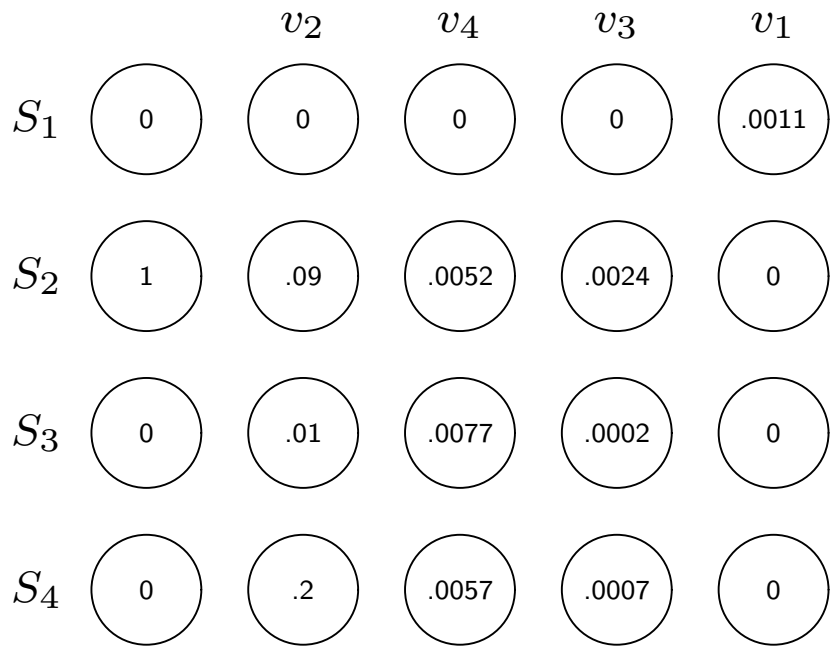
$$P(O) = \sum_{\text{alle } q \in Q} \alpha[T][q];$$

Drei klassische Algorithmen für HMMs: Evaluation Problem (3)

Beispiel: Angenommen, der initiale verborgene Zustand zu Zeit $t = 1$ sei S_2. Was ist die Wahrscheinlichkeit für die Beobachtungsfolge $\boldsymbol{O} = *v_2v_4v_3v_1$? Das Symbol $*$ bedeutet, dass zu Zeit $t = 1$ ein beliebiges Symbol ausgegeben werden kann.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0 & 0.1 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{bmatrix}$$



$\alpha[2][2] = 0.3 \cdot 0.3 = 0.09$ (Generierung von $*v_2$, Zustand S_2 zu Zeit $t = 2$)

$\alpha[3][2] = (0.09 \cdot 0.3 + 0.01 \cdot 0.5 + 0.2 \cdot 0.1) \cdot 0.1$

$= 0.0052$ (Generierung von $*v_2v_4$, Zustand S_2 zu Zeit $t = 3$)

$\alpha[5][1] = (0.0024 \cdot 0.2 + 0.0002 \cdot 0.2 + 0.0007 \cdot 0.8) \cdot 1$

$= 0.00108\ (\approx 0.0011)$ (Generierung von $*v_2v_4v_3v_1$, Zustand S_1 zu Zeit $t = 5$)

Antwort: $P(*v_2v_4v_3v_1) = 0.0011$

HMMs in der Praxis (1)

Es sind effiziente Algorithmen für alle drei Probleme (Evaluation, Decoding, Training) bekannt. Softwaretools (public-domain, kommerziell) sind vorhanden, welche diese grundsätzlichen Algorithmen bereitstellen.

Einsatz eines HMMs umfasst folgende Schritte:

- Bereitstellung/Entwicklung von Verfahren zur Vorverarbeitung und Merkmalsextraktion
- Definition eines HMMs pro Musterklasse; insb. Wahl der
  - Topologie
  - Anzahl Zustände
  - Form der Ausgabewahrscheinlichkeiten

  sowie ggf. Konkatenation einzelner Modelle (um z.B. aus Buchstabenmodellen Wortmodellen zu bilden)
- Bereitstellung von Trainings- und Validierungsdaten
- Training der HMMs unter Verwendung der Trainingsdaten
- Validierung und Re-Design

HMMs in der Praxis (2)

Bei längeren Rechengängen treten numerische Probleme auf, da extrem kleine (nahe Null liegende) Werte manipuliert werden müssen.

Beispiel: Brechnung der Wahrscheinlichkeit $P(w_1 \dots w_T)$:

$$P(w_1 \dots w_T) \;=\; \prod_{t=1}^{T} P(w_t|w_{t-1})$$

Selbst wenn in jedem Zustand nur jeweils zwei Nachfolger mit gleicher Wahrscheinlichkeit vorkommen, also alle $P(w_t|w_{t-1}) = 0.5$ sind, erhält man ab einer Länge $T > 100$ bereits Zahlwerte kleiner als $5 \cdot 10^{-100}$.

Repräsentation auf negativ-logarithmischer Skala: Anstatt der eigentlichen Wahrscheinlichkeitswerte oder Dichtewerte p wird mit Transformationswert:

$$\tilde{p} \;=\; -\log_b p$$

gearbeitet, wobei die Wahl von b keinen nennenswerten Einfluss hat.

Die ursprünglich multiplikativen Wahrscheinlichkeitswerte können dann als additive Kosten interpretiert werden. Auch Berechnungsabläufe, die Summationen von Wahrscheinlichkeitswerten erfordern, können einheitlich im logarithmischen Bereich erfolgen.

HMMs in der Praxis (3)

HTK: The Hidden Markov Model Toolkit is a portable toolkit for building and manipulating hidden Markov models. HTK is primarily used for speech recognition research although it has been used for numerous other applications including research into speech synthesis, character recognition and DNA sequencing. HTK is in use at hundreds of sites worldwide.

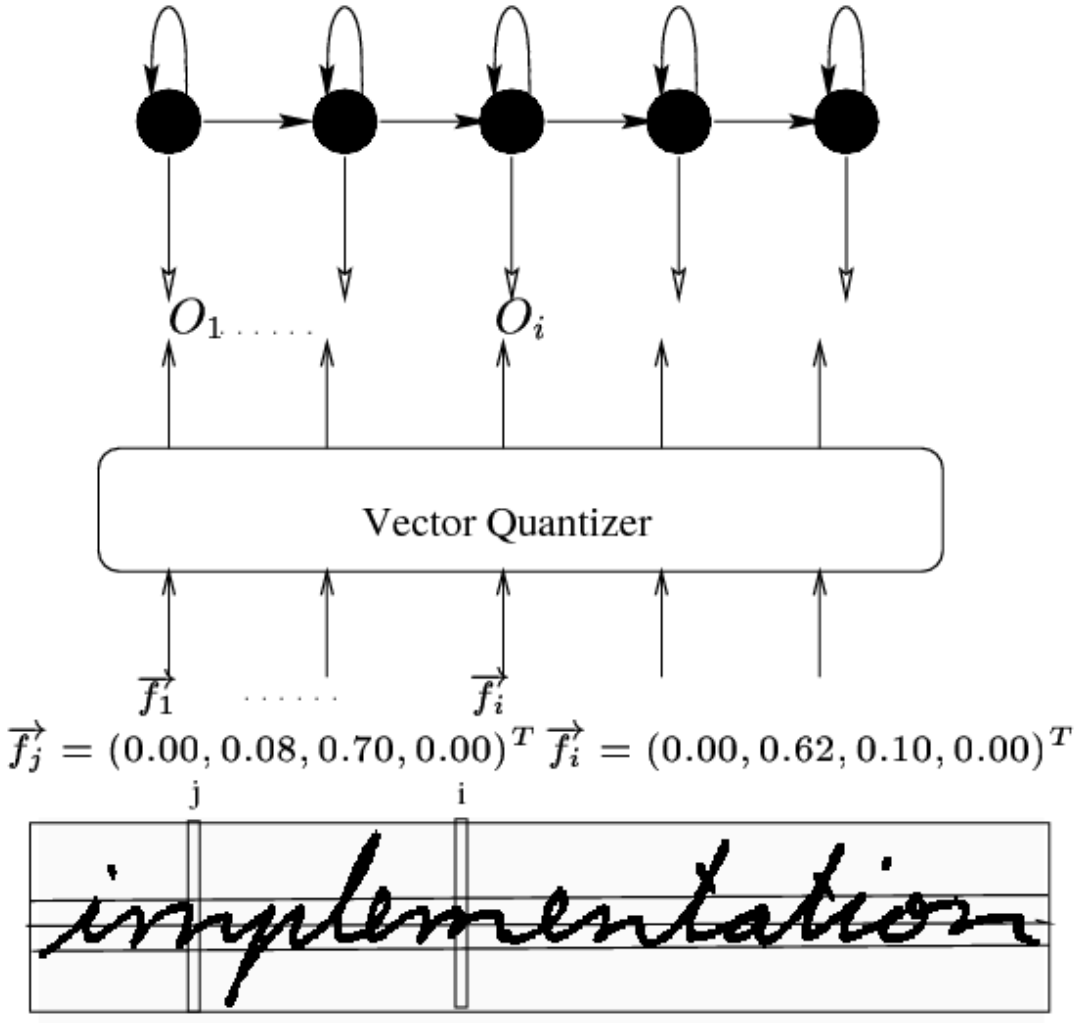
HTK consists of a set of library modules and tools available in C source form. The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be used to build complex HMM systems. The HTK release contains extensive documentation and examples.

HTK is available for free download: http://htk.eng.cam.ac.uk/

Anwendungen (1)

Spracherkennung: Hierbei gelten HMMs als das Standardverfahren

Schriftlesen:

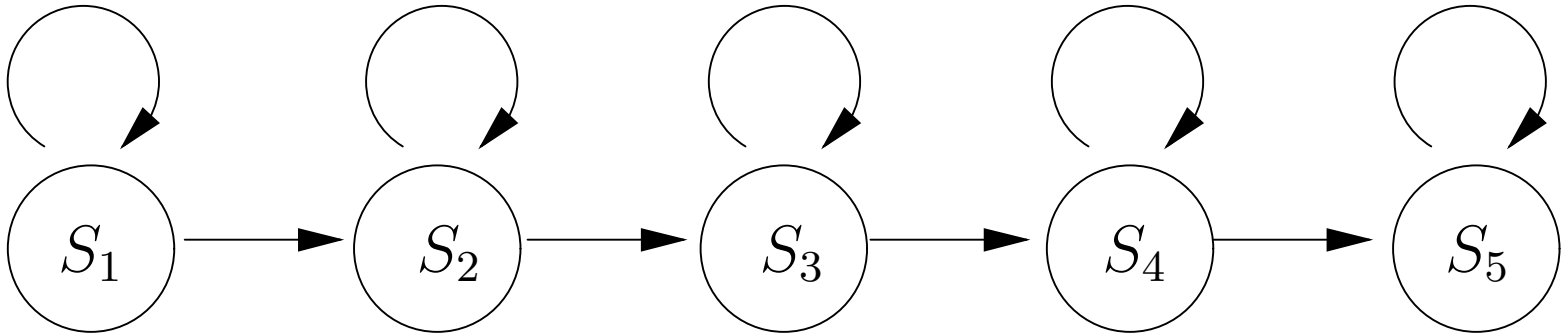


Anwendungen (2)

Gesichtserkennung:

2D-Bild wird in 1D-Repräsentation gebracht

(Ein Fenster fester Größe über das Bild schieben. Die Grauwerte in einem Fenster werden als Merkmale dem HMM übergeben)

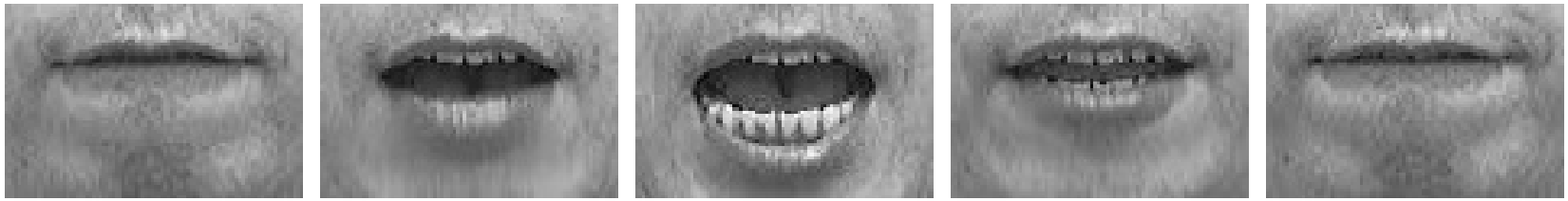


Zustände:

- S_1: Stirnbereich
- S_2: Augenbereich
- S_3: Nasenbereich
- S_4: Mundbereich
- S_5: Kinnbereich

Anwendungen (3)

Lipreading: email commands



Beispiele:
Show All New Messages / Display Message Number One
Save Previous Message In Folder Inbox / Help

Für jedes Wort wird ein HMM konstruiert. Anhand der Grammatik werden diese dann zu komplexen HMMs zusammengefügt

Anwendungen (4)

Befehle durch Grammatik definiert:

N_e = {commands, part1, part2, part3, part4, msg_spec, folders, msg_no, digit};

T_e = {Show, List, All, Any, New, Messages, Display, Reply, To, Forward,
Delete, Jump, Go, Save, Export, In, Folder, Help, Quit, Exit, Return,
Main, Menu, Message, Number, Previous, Next, First, Last, Current,
This, Inbox, Outbox, One, Two, . . . , Nine, Zero, Oh};

S_e = commands;

P_e = { commands → part1 | part2 | par t3 | part4,
part1 → [Show | List] [All | Any] New Messages,
part2 → (Display | Reply [To] | Forward | Delete | Jump | Go To) msg_spec,
part3 → (Save | Export) msg_spec (In | To) [Folder] folders,
part4 → Help | Quit | Exit | ([Return To] Main [Menu]),
msg_spec → Message Number msg_no | (Previous | Next |
First | Last | Current | This) [Message],
folders → Inbox | Outbox,
msg_no → digit [digit [digit]],
digit → One | Two | . . . | Nine | Zero | Oh}.