

Mensch-Maschine-Kommunikation 1

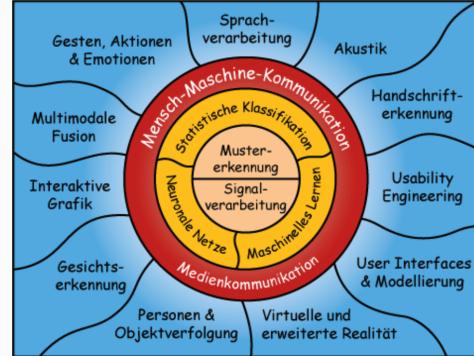
(ab WS 2016/17)

1. Allgemeine Einführung

1.1. Grundbegriffe der MMK

Interaktion	Kommunikation zwischen Mensch und Maschine.
Interaktives System	System, das auf Eingaben reagiert und gegebenenfalls auch Ausgaben generiert.
HCI	Human-Computer Interaction.
MMI	Mensch-Maschine-Interface.
Usability	Gebrauchstauglichkeit bzw. Eignung eines Produkts.
Usability Engineering	Gestaltung und Testen eines Produktes mit dem Ziel optimaler Bedienbarkeit durch die Mensch-Maschine-Schnittstelle.
Software-Ergonomie	Wissenschaft über die Gestaltung von Programmen mit benutzerfreundlicher Mensch-Maschine-Schnittstelle.
Medium	Datenträger für Information, z.B. Papier oder CD.
Multimedia	Datenverarbeitung und -darstellung unter Nutzung verschiedener Medien, z.B. Text, Grafik und Audio und Video.
Modalität	Ein-/Ausgabekanal der menschlichen Kommunikation und Sinneswahrnehmung, z.B. Sprache, Zeigen, Gestik, Tastatur.

1.2. Wichtigste Disziplinen der MMK



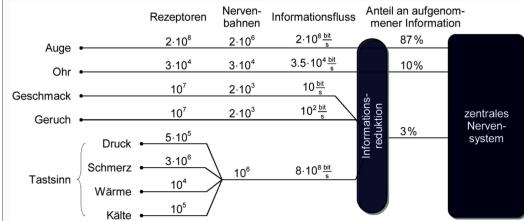
1.3. Trends in der MMK

- Steigerung der Leistungsfähigkeit
- Reduzierung der Kosten
- Erweiterung der Funktionalität
- Verbesserung der Bedienbarkeit

1.4. Übersicht über Sinnesmodalitäten

Sinnesbezeichnung	Modalität	Bemerkung
Sehen	visuell	
Hören	auditiv	
Riechen	olfaktorisch	„5 Sinne“
Schmecken	gustatorisch	
Tasten	taktil	
Druck	haptisch	mechanische Modal.
Kraft		
Berührung	taktil	oberflächen-sensitiv
Vibration		
Temperatur	thermorezeptorisch	
Bewegung und Orientierung	kinästhetisch	
Gleichgewicht	vestibular	

1.5. Die Sinne des Menschen und ihre Datenraten



1.6. Datenraten gängiger Systeme der MMK

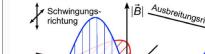
System	Verhalten	Rate (KByte/sec)
Tastatur (ungeübt)	Eingabe	0.01
Tastatur (geübt)	Eingabe	0.025
Handschrift	Eingabe	0.0025
Spracheingabe	Eingabe	0.01-0.02
Maus	Eingabe	0.02
Sprachausgabe	Ausgabe	0.6
Text lesen	Ausgabe	0.03-0.3
Hören (CD)	Ausgabe	40
Sehen (Video)	Ausgabe	20000

2. Sprachkommunikation

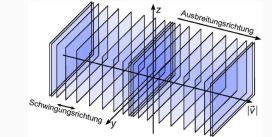
Ermittlung der geäußerten Wortfolge aus einem vorliegenden Sprachsignal und Verarbeitung dieser Information. Die Sprachkommunikation hat größtes Potential aller Eingabemethoden, da sie auch beim Menschen die häufigste und natürlichste Kommunikationsform ist.

2.1. Physikalische Wellen

Transversalwelle:



Longitudinalwelle (z.B. Schall):



2.2. Schallquellen und ihre typischen Pegel

Schall	ggf. Entfernungsangabe	Pegel
Düsengejäger, 30 m		$L = 140 \text{ dB} \equiv p = 2 \cdot 10^{-2} \text{ Pa}$
lautes Händeklatschen, 1 m		$L = 130 \text{ dB} \equiv p = 63 \text{ Pa}$
Trillerpfeife, 1 m		$L = 120 \text{ dB} \equiv p = 20 \text{ Pa}$
Discman		$L = 110 \text{ dB} \equiv p = 6.3 \text{ Pa}$
Presslufthammer, 10 m		$L = 100 \text{ dB} \equiv p = 2 \text{ Pa}$
lauter Fabrikhalle		$L = 90 \text{ dB} \equiv p = 6.3 \cdot 10^{-1} \text{ Pa}$
starker Straßenverkehr		$L = 80 \text{ dB} \equiv p = 2.0 \cdot 10^{-1} \text{ Pa}$
Staubsauger, 1 m		$L = 70 \text{ dB} \equiv p = 6.3 \cdot 10^{-2} \text{ Pa}$
normale Sprache, 1 m		$L = 60 \text{ dB} \equiv p = 2.0 \cdot 10^{-2} \text{ Pa}$
Kühlschrank, 1 m		$L = 50 \text{ dB} \equiv p = 6.3 \cdot 10^{-3} \text{ Pa}$
normale Wohngeräusche, 1 m		$L = 40 \text{ dB} \equiv p = 2.0 \cdot 10^{-3} \text{ Pa}$
Flüstersprache		$L = 30 \text{ dB} \equiv p = 6.3 \cdot 10^{-4} \text{ Pa}$
mechanischer Wecker, 1 m		$L = 20 \text{ dB} \equiv p = 2.0 \cdot 10^{-4} \text{ Pa}$
Blätterraschen in der Ferne		$L = 10 \text{ dB} \equiv p = 6.3 \cdot 10^{-5} \text{ Pa}$

2.3.2. Psychoakustik

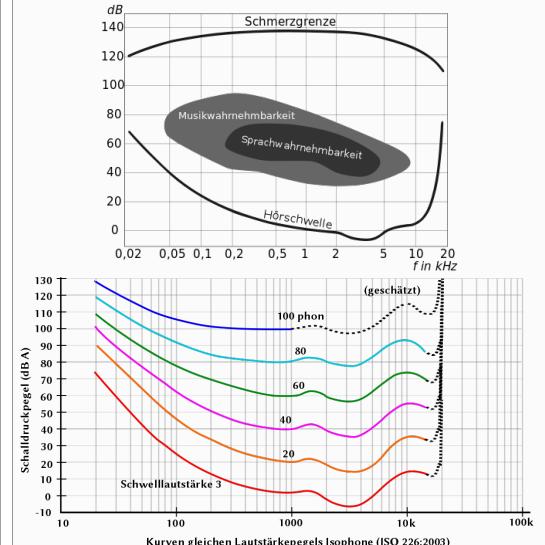
- Empfindlich von etwa 20 Hz - 20 kHz (≈ 10 Oktaven)
- Starke Dämpfung für sehr niedrige und sehr hohe Frequenzen
- Resonanzfrequenz des Gehörgangs bei ca. 3...3.4 kHz;
- Lauteinheit in [sone] 1 sone \triangleq Lautheit eines 1kHz Sinus mit 40 dB
- Verhältnistonalhöhe [mel] 1000 mel \triangleq 1000Hz

Psychoakustik		Physik	
Bezeichnung	Einheit	Bezeichnung	Einheit
Tonheit Z	Bark	Frequenz f	Hz
Verhältnistonalh. V	Mel	Schalldruck p	$\frac{N}{m^2} = Pa$
		Schallschnelle v	$\frac{m}{s}$
		Schallintensität I	$\frac{W}{m^2} = \frac{N}{s m}$
Lautstrk.pegel L_n	Phon	Schalldruckp. L	dB
Lautheit N	sone	Schalleist. P_{ak}	$W = \frac{Nm}{s}$

$$\text{Bezugsschalldruck } p_0 = 2 \cdot 10^{-5} \frac{m^2}{s} = 20 \mu Pa$$

$$\text{Bezugsintensität } I_0 = 1 \cdot 10^{-12} \frac{W}{m^2}$$

Hörfläche Jener Frequenz- und Pegelbereich von Schall, der vom menschlichen Gehör wahrgenommen werden kann:



Frequenzbewertung Verfahren zur frequenzabhängigen Anpassung von Schalldruckpegeln an das menschliche Hörempfinden (nichtlinear zur Lautstärke). Hierfür werden verschiedene Filterkurven verwendet: A(20-40 phon), B(50-70 phon), C(80-90 phon), D(sehr hohe Schalldrücke) mit gleichem Lautstärkeindruck. Lautheit N in Sone ist angepasstes Schema.

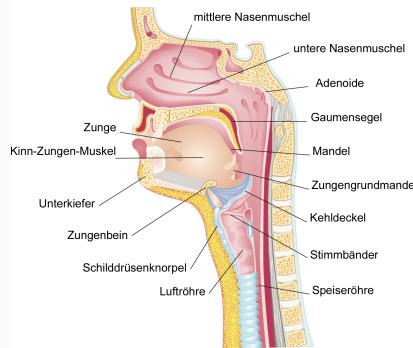
Frequenzgruppen (24) begrenzte Auflösung des Gehörs; jede Frequenzgruppe nimmt gleiche Länge auf Basilarmembran ein (1,3mm - unter 500 Hz = 100Hz, darüber kleine Terz 1,19 der Mittelfrequenz); Bark-Skala; 1.31 Bark = 131 mel = 131 Hz.; Blätterraschen in Ferne L = 10dB; Düsenjäger in 30 m L = 140dB.

Verdeckungen Hörschwelle bei Störschall (Maskieren):

- Spektrale: verbreitet sich mit steigendem Pegel überproportional.
- Zeitliche: Vorverdeckung; Simultanverdeckung; Nachverdeckung (etwa hundert ms).

Kompression: Mithörschwelle über Verdeckungen ermitteln; MP3 ab 160 kBit/s.

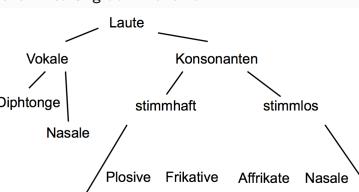
2.4. Menschliche Spracherzeugung



2.4.1. Phoneme
Das Phonem ist die kleinste bedeutungsunterscheidende Einheit des gesprochenen Wortes.

Phonem	Aussprache	Phonem	Aussprache	Phonem	Aussprache
/a/	Kampf	/a:/	Kahn	/ai/	weit
/au/	Haus	/ax/	mache	/b/	Ball
/d/	deutsch	/eh/	wenn	/eh:/	Affäre
/ey/	wen	/f/	fern	/g/	gern
/h/	Hand	/i/	Himmel	/i:/	Hier
/j/	Junge	/jh/	Joystick	/k/	Kind
/l/	links	/m/	matt	/n/	Nest
/ng/	lang	/o/	offen	/o:/	Öfen
/oe/	Hölle	/oe:/	Höhle	/oy/	freut
/p/	Paar	/r:/	rennen	/s/	fassen
/sh/	schön	/t/	Tafel	/u/	Mutter
/u/	Mut	/v/	wer	/x/	lachen
/y/	Typ	/y:/	Kübel	/z/	singen
/zh/	Ingenieur	/sp/	„short pause“	/sil/	„silence“

Systematische Einteilung der Phoneme:



3. Grammatiken

Natürlichsprachige Systeme; Modellierung von Dialogen.

3.1. Kontextfreie Grammatiken (CFG)

- $\mathcal{G} = \{V, T, P, S\}$ mit
- $V \equiv$ Variable (Großbuchstaben)
 - $T \equiv$ Terminale (Kleinbuchstaben)
 - $P \equiv$ Produktionsregel ($A \rightarrow \alpha$ mit $A \in \{V\}$ und $\alpha \in \{V \cup T\}$)
 - $S \equiv$ Startsymbol

3.1.1. Chomsky-Normal Form (CNF)

Enthält nur Produktionsregeln, bei denen auf der rechten Seite nur zwei Variablen oder nur ein terminaler Ausdruck steht:

$$A \rightarrow BC \text{ oder } A \rightarrow a$$

3.1.2. Backus-Naur-Form (BNF)

Formal exakte Definition von Programmiersprachen. Nichtterminalsymbole werden syntaktische Variablen genannt und durch < > gekennzeichnet. Darstellung von Wiederholungen durch Rekursion.

- | Alternative
- (...) Gruppierung
- [...] oder (...)? Option
- (...) * optionale Wiederholung (keinmal, ein- oder mehrfach)
- (...) + Wiederholung (ein- oder mehrfach)

3.1.3. Erweiterte Backus-Naur-Form (EBNF)

- [...] Option
- ... optionale Wiederholung (keinmal, ein- oder mehrfach)
- n* abgezählte Wiederholung

3.1.4. Parsing

Satzgenerierung: Produktionsregeln solange anwenden, bis alle Variablen V durch terminale Symbole T ersetzt sind; Parse-Tree; Ambiguitäten;

3.1.5. Anwendung von Grammatiken in KI Sprache; Mustererkennung;

3.2. Beispiele Grammatiken

Palindrom-String:

$$S \rightarrow aSa|bSb|a * |b *$$

Doppelte Anzahl a wie b:

$$\begin{aligned} S &\rightarrow A|SA|AS|aSC|CSa|aSD|DSa|bSB|BSb \\ A &\rightarrow Bb|Ca|Da \\ B &\rightarrow aa \quad C \rightarrow ab \quad D \rightarrow ba \end{aligned}$$

Grammatik-Grammatik:

S (Satz), NP (Nominalphrase), VP (Verbalphrase), PP (Päpositionalphrase), DET (Determinator, Artikel), ADJ (Adjektiv), AUX (Hilfswort), V (Verb), PRE (Präposition) und N (Nomen)

$$\begin{aligned} S &\rightarrow NP\ VP|VP\ NP \\ NP &\rightarrow DET\ N|ADJ\ N|DET\ NP|NP\ PP \\ VP &\rightarrow V\ NP|AUX\ V|V\ PP|V\ NP|VP\ PP|AUX\ VP \\ PP &\rightarrow PRE\ NP \\ DET &\rightarrow „der“, „die“, „das“, ... \\ ADJ &\rightarrow „klein“, „groß“, ... \\ AUX &\rightarrow „wird“, ... \\ V &\rightarrow „streichen“, ... \\ PRE &\rightarrow „in“, „mit“, ... \\ N &\rightarrow „Junge“, „Hund“, „Hand“, ... \end{aligned}$$

4. Automatentheorie

Verarbeitung von Symbolfolgen; Modellierung von Dialogen;

4.1. Zustandsautomat

Graphform; bestimmte Anzahl von Knoten (Zustände) und Verbindungen (Transitionen).

$$Z = (\mathcal{S}, \mathcal{X}, \mathcal{T}, s_0, \mathcal{F})$$

- \mathcal{S} Set mit endlicher Anzahl Zustände
 - \mathcal{X} zulässiges Alphabet für die zu verarbeitende Symbolfolge X
 - \mathcal{T} Transitionsfunktionen für die Zustände in \mathcal{S}
 - s_0 Anfangszustand
 - \mathcal{F} ein Set von festgelegten Endzuständen
- Transitionsfunktion als Regel: $t(s_i^-, x_i) = s_i^+$

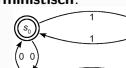
Umwandlung: Zustandsautomat in Grammatik

1. Zustände werden Variable: $S \Rightarrow V$
2. Eingabealph. wird zu Terminal: $X \Rightarrow T$
3. Transitionen werden Produktionsregeln: $\mathcal{T} \Rightarrow P$, z.B. $P = \{S \rightarrow aA, AabA\}$
4. Für jeden Endzustand s_E erstelle Produktionsregel, z.B. für B als Endzustand $\Rightarrow P = \{\dots, B \rightarrow \epsilon\}$

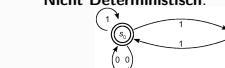
Endliche Zustandsautomaten Automat mit endlicher Menge von Zuständen S

Deterministische Zustandsautomaten Automat, in dem die Folgezustände immer eindeutig definiert sind durch den aktuellen Zustand und Eingabesymbol

Deterministisch:



Nicht Deterministisch:



Transitionsregeln in Tabellenform:

alter Zustand	Symbol-Input	0	1
s_0	s_2	1	
s_1	s_3		0
s_2	s_3	0	
s_3	s_1		1

$$\mathcal{S} = \{s_0, s_1, s_2, s_3\}$$

$$\mathcal{X} = \{0, 1\}$$

$$\mathcal{F} = \{s_0\}$$

4.2. Kellerautomaten

Komplexere Grammatiken; Erweiterung mit Stack (LIFO-Queue); Transition abhängig von Stack und Eingang; Stack leer \Rightarrow Folge akzeptiert;

$$Z = (\mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{T}, s_0, y_0, \mathcal{F})$$

- \mathcal{Y} zulässiges Alphabet für den Stack
- y_0 Startsymbol für den Stack
- \mathcal{F} ein Set von festgelegten Endzuständen (leer wenn Endzustand über leeren Stack definiert ist)
- (Für \mathcal{S} , \mathcal{X} , \mathcal{T} , s_0 siehe Zustandsautomat)

Aktionen:

- push(x) lege x auf den Stack
- pop() lese und entferne oberste Stack-Element
- ε keine Aktion
- # leeres Stack (erster und letzter Schritt)

Beispiel für einen Kellerautomaten:

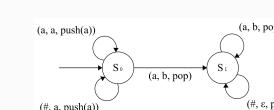
$$\mathcal{S} = \{S_0, S_1\}$$

$$\mathcal{X} = \{a, b\}$$

$$\mathcal{Y} = \{\#, a\}$$

$$y_0 = \#$$

$$\mathcal{F} = \{\} \text{ (Ende durch leeren Stack)}$$



Generiert Sprache: $L(a^n b^n)$ mit $n > 0$

Angaben in Klammern:
(Voraussetzung auf Stack in \mathcal{Y} , Eingabe in \mathcal{X} , Aktion)

5. Spracherkennung

Spracherkennung beschäftigt sich mit der Untersuchung und Entwicklung von Verfahren, die Automaten, insbesondere Computer, die gesprochene Sprache der automatischen Datenerfassung zugänglich macht.

5.1. Klassifizierung

Zuordnung zu Bedeutungseinheiten; Merkmalsextraktion; Merkmalsvektor; Merkmalsraum; Klassen; Training;

5.2. Abstandsklassifikatoren

Distanz eines Mustervektors zu Klasse;

- \vec{x} unbekannt, zu klassifizierende Mustervektor
- $\vec{r}_{k,i}$ i-ter Referenzvektor für die k-te Klasse
- \vec{m}_k Klassenzentrum der Klasse k
- $d_k(\vec{x}, \vec{m}_k)$ Abstandsformel
- k_x Klasse mit minimalen Abstand zu \vec{x}

Formeln

$$\vec{m}_k = \frac{1}{M_k} \sum_{i=1}^{M_k} \vec{r}_{k,i}$$

$$d_k(\vec{x}, \vec{m}_k) = (\vec{x} - \vec{m}_k)^T \cdot \underline{W}_k \cdot (\vec{x} - \vec{m}_k)$$

$$k_x = \operatorname{argmin}_{\vec{x}} d_k(\vec{x}, \vec{m}_k)$$

Trennfunktion:

$$d_1(x, m_1) - d_2(x, m_2) = 0$$

Gewichtsmatrix W_k entscheidet über Ergebnis; m_k wird im Training ermittelt; x gehört zur Klasse k mit minimalen Abstand;

Quadratischer Abstand W_k ist Einheitsmatrix; Trennfunktion ist eine Gerade;

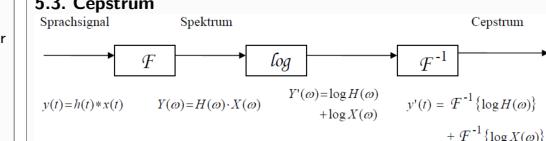
Mahalanobis Abstand Inverse der Kovarianzmatrix; Abhängig von Klasse; Bestandteil des Trainings; Trennfunktion ist Kegelschnitt (Gerade, Ellipse, Parabel, Hyperbel).

$$\underline{C}_k = \left(\frac{1}{M_k} \sum_{i=1}^{M_k} \vec{r}_{k,i} \cdot \vec{r}_{k,i}^T - \vec{m}_k \cdot \vec{m}_k^T \right)$$

$$\underline{W}_k = \underline{C}_k^{-1}$$

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

5.3. Cepstrum



Praktische Berechnung:

- Selektion eines Zeitfensters für das betrachtete Sprachsignal
- Fourier-Transformation dieses Signals in den Frequenzbereich
- Bilden des Betrags des resultierenden (komplexen) Spektrums
- Logarithmierung des Amplitudenspektrums
- Rücktransformation mit inverser FT

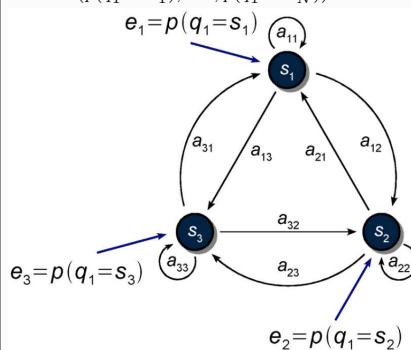
6. Hidden-Markov-Modelle und Algorithmen

Wahrscheinlichkeit Statistischer Klassifikator. Liefert Wahrscheinlichkeit p , dass eine Beobachtung einer bestimmten Klasse zuordnet werden kann. Klassifizieren ganze Sequenzen (dynamische Folgen). „Finde diejenige Klasse, die die Beobachtung $o = (o_1, o_2, \dots, o_T)$ am besten nachbilden kann.“.

6.1. Markov-Modelle (MM)

Abbildung stochastischer Prozesse, deren aktueller Zustand nur vom vorausgegangenen Zustand abhängt.

- Matrix der Übergangswkt.: $A = p\{q_{t+1} = s_j | q_t = s_i\}$
- Vektor der Einsprungwkt.: $\vec{e} = (p(q_1 = s_1), \dots, p(q_1 = s_N))^T$



6.2. Hidden-Markov-Modelle (HMM)

Stochastische Version eines endlichen Zustandautomaten; Zustandsübergänge und Symbolemissionen nicht deterministisch.

- Matrix A und Vektor \vec{e} siehe MM
- Beobachtungsfolge: $\vec{o} = (o_1, \dots, o_T)^T$
- Alphabet: $\vec{v} = (v_1, \dots, v_M)^T$
- Beobachtungswahrscheinlichkeiten: $b_{mi} = p(v_m | s_i)$
- Matrix der Beobachtungswahrscheinlichkeiten:

$$B = \begin{bmatrix} p(v_1 | s_1) & \dots & p(v_1 | s_N) \\ \vdots & \ddots & \vdots \\ p(v_M | s_1) & \dots & p(v_M | s_N) \end{bmatrix}$$

Zusammengefasste Parameter des HMMs: $\lambda = (\vec{e}, A, B)$

Beobachtungs- bzw. Produktionswkt.: $p(\vec{o} | \lambda)$

Dabei durchlauene (verborgene/hidden) Zustandsfolge:

$$\vec{q} = (q_1, \dots, q_T)$$

HMM - Eigenschaften

Ergodisches HMM Es kann aus jedem Zustand in jeder andere Zustand erreicht werden; A ist voll besetzt

Links-Rechts-HMM keine Rücksprünge; kausal; A hat rechte obere Dreiecksform; Graphisch nach rechts aufsteigend

6.2.1. Klassifizierung mit HMM

Pro Klasse ein HMM; das HMM welches die größte Produktionswahrscheinlichkeit $p(o|\lambda_k)$ liefert, repräsentiert die gesuchte Klasse k_x .

6.2.2. Training von HMM

Kompensation von Störungen; Bed.: geeignete Parameter λ_k ; Training mit iterativen Verfahren; \Rightarrow Baum-Welch-Algorithmus

6.3. HMM in der Spracherkennung

Cepstrum; Merkmalsextraktion; 12D Merkmalsvektor;

6.3.1. Modelle

Einzelworterkenner vs. fließende Sprache; Phoneme, kleinste bedeutungsunterscheidenden Lauteinheiten; HMM pro Phonem; Pausen;

6.3.2. Training

Zusammenfassung der Phonem HMM zu einem HMM;

6.3.3. Erkennung

Wörterbücher, Grammatiken, Wahrscheinlichkeiten bestimmter Phonemkombinationen, Sprachmodelle für Wortkombinationen;

6.4. HMM-Algorithmen

6.4.1. Trellis

Mathematische Formel zur Berechnung der Beobachtungswkt.

Für verschiedene Wege q gilt:

$$p(\vec{o}, \vec{q} | \lambda_k) = e_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1} q_t} b_{q_t}(o_t) \text{ Beobachtungswahrscheinlichkeit:}$$

$$\begin{aligned} p(\vec{o} | \lambda_k) &= \sum_{q \in Q} p(\vec{o}, \vec{q} | \lambda_k) \\ &= \sum_{q \in Q} e_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1} q_t} b_{q_t}(o_t) \end{aligned}$$

Benötigte OPS $\sim 2T \cdot N^T$ (sehr rechenintensiv)

6.4.2. Vorwärts-Algorithmus

Vorwärts-Wahrscheinlichkeit:

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i | \lambda_k)$$

d.h. die Wahrscheinlichkeit, dass die Teilbeobachtung o_i emittiert werden und das sich das HMM zu t im Zustand s_i befindet;

Vorwärts-Algorithmus (Rekursiv)

1. Initialisierung:

$$\alpha_1(i) = e_i b_i(o_1), \quad 1 \leq i \leq N$$

2. Induktion:

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

3. Terminierung

$$\mathbb{P}(o | \lambda_k) = \sum_{i=1}^N \alpha_T(i)$$

Benötigte OPS $\sim T \cdot N^2$

6.4.3. Baum-Welch-Algorithmus

Rückwärtswahrscheinlichkeit:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = s_i, \lambda_k);$$

d.h. Wahrscheinlichkeit, die restlichen Teilbeob. zu emittieren;

Baum-Welch-Algorithmus (Rekursiv)

1. Initialisierung

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

2. Induktion

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \\ t &= T-1, T-2, \dots, 1 \quad 1 \leq i \leq N \end{aligned}$$

Wahrscheinlichkeit, dass sich das HMM zu t im Zustand s_i befindet und o emittiert wird; Summe drüber \Rightarrow „alle Aufenthalte im Zustand s_i “

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

Wahrscheinlichkeit, dass sich das HMM zu t in s_i und zu $t+1$ in s_j befindet; Summe drüber \Rightarrow „aller Übergänge von s_i zu s_j “

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}$$

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

6.4.4. Viterbi-Algo

Berechnet die Beobachtungswahrscheinlichkeit des wahrscheinlichsten Pfades.

Viterbi-Algorithmus

1. Initialisierung:

$$\delta_1(i) = e_i b_i(o_1) \quad 1 \leq i \leq N$$

$$\psi_1(i) = 0$$

2. Induktion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

$$2 \leq t \leq T; \quad 1 \leq j \leq N$$

3. Terminierung:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

4. Ermittlung der wahrsch. Zustandsfolge:

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

7. Suchverfahren

Formulierung und Darstellung eines Problems im Zustandsraum; Graphen-Darstellung; Suchbaum;

Zyklische Wiederholungen unterbinden (gerichtete Kanten im Baum).

7.1. Allgemeiner Algorithmus für Suche

Suchalgorithmus

1. Initialisiere Queue

2. Schreibe Startknoten in Queue

3. Wiederhole:

- Queue leer? \Rightarrow Ziel nicht gefunden
- Entnehme nächsten Knoten
- Knoten == Ziel? \Rightarrow Ziel erreicht
- Schreibe alle Kinder des Knotens in die Queue
- Update Queue

Art des Algorithmus bestimmt die Art der Queue, und damit die Update-Funktion:

Suchalgorithmus

Breitensuche

Tiefensuche

A-Suche

A*-Suche

Dijkstra

Art der Queue

FIFO-Queue

LIFO-Queue (Stack)

Priotiy-Queue

Priotiy-Queue mit heuristischen Kosten als Priorität

7.2. Tiefensuche und Breitensuche

1. einelementige Liste mit Wurzelknoten

2. bis Liste leer / Ziel erreicht:

- prüfe erstes Element auf Zielknoten bzw. max. Suchtiefe
- wenn ja, fertig
- wenn nein, entferne dieses Element und füge all seine Nachfolger an gleicher Stelle / am Ende ein.

Voraussetzung: Elemente der Warteliste werden systematisch erzeugt; Suchtiefe wird geeignet groß festgesetzt / ausgewertete Suchbaum muss gespeichert werden;

7.3. Heuristische Suche / A-Algorithmus

Verarbeitung zusätzlicher Informationen; Bewertungsmöglichkeit für Erfolgsaussichten eines bestimmten Pfades; Entscheidungen ordnen; Vielversprechende Alternative zuerst, „dem atm billigsten folgen“; Heuristik besteht in Definition einer geeigneten Bewertungs (Kostenfunktion) $f(n)$:

$$f(n) = g(n) + h(n)$$

Bewertungsfunktion = Bisherige Kosten + Schätzfunktion (hier: falsche Plättchen)

Falls $h(n) \equiv 0$ gewählt wird identisch zur Dijkstra-Algorithmus (wenn dazu $g(n) \equiv \text{Tiefe des Knotens}$ identisch zur Breitensuche)

7.4. A*-Algorithmus

Schätzfunktion $h(n)$ monoton, d.h. Kosten werden nicht überschätzt; terminiert wenn Zielknoten gefunden und keine geringere Kostenschätzung existiert; A* somit optimaler Pfad; wird die optimale Kostenfkt $h1^*(n)$ verwendet, so wird kürzester Pfad auf Anhieb gefunden (sprich: informierte Suche); Liste mit allen Elementen erstellen + sortieren; dem insg. billigsten folgen; nix verwerfen.

8. Logik und Theorembeweisen

Wissen algorithmisch darstellen; Fakten ableiten; Behauptungen bestätigen / widerlegen;

8.1. Aussagenlogik

atomare Aussagen; wahr oder falsch; UND , ODER, NICHT; Implikation \Rightarrow :

8.2. Prädikatenlogik

Analyse und Bewertung von Beziehungen und logischen Verknüpfungen
1. Ordnung \Rightarrow nur Veränderung von Objekten, nicht Prädikaten
Prädikate und Funktionen, Konstanten, Variablen, Funktionen, Negation, Disjunktion, Konjunktion, Existenz-Quantor, All-Quantor, Implikation, Äquivalenz.

Beispiel: „In jeder Stadt gibt es einen Bürgermeister“
 $(\forall x) \{ \text{Stadt}(x) \Rightarrow (\exists y) [\text{Mensch}(y) \cdot \text{Bgm}(x, y)] \}$

Regeln und Zusammenhänge aufstellen; \Rightarrow Regelwerk (Axiome); Frage (Theorem); Beweis durch Wahrheitstabelle oder Umformen der Regeln und Schlussfolgerungen (Resolution, Unifikation - effektiver);

a	w	w	f	f	Ausdruck	Name	Umgangssprachlich
b	w	w	w	f	$\neg a$	Negation	Nicht A
w	f	f	w	w	$a \cdot b$	Konjunktion	A und B
w	w	w	f	w	$a + b$	Disjunktion	A oder B
f	w	w	f	w	$a \Leftrightarrow b$	Exkl. Disjunktion	Entweder A oder B
w	w	f	w	w	$a \Rightarrow b$	Implikation	Wenn A, dann B
w	w	f	w	w	$a \Leftrightarrow b$	Bijunktion	A genau dann, wenn B
w	w	w	w	w	Tautologie	Stets wahr	
f	f	f	f	f	Kontradiktion	Stets falsch	

Umformregeln

1. Doppelte Negation $\neg\neg A \equiv A$
2. Idempotenz $A + A \equiv A$ und $A \cdot A \equiv A$
3. Kommutativität $A + B \equiv B + A$ und $A \cdot B \equiv B \cdot A$
4. Assoziativität $A + (B + C) \equiv (A + B) + C$ und $A \cdot (B \cdot C) \equiv (A \cdot B) \cdot C$
5. Distributivität $A + (B \cdot C) \equiv (A + B) \cdot (A + C)$ und $A \cdot (B + C) \equiv (A \cdot B) + (A \cdot C)$
6. De Morgan $\neg(A \cdot B) \equiv \neg A + \neg B$ und $\neg(A + B) \equiv \neg A \cdot \neg B$
7. Kontraposition $A \Rightarrow B \equiv \neg B \Rightarrow \neg A$
8. $A \Rightarrow B \equiv \neg A + B$
9. $A \Rightarrow B \equiv (A \Rightarrow B) \cdot (B \Rightarrow A) \equiv (A \cdot B) + (\neg A \cdot \neg B)$
10. $\neg(\forall x)A(x) \equiv (\exists x)(\neg A(x))$
11. $\neg(\exists x)A(x) \equiv (\forall x)(\neg A(x))$
12. $(\forall x)(A(x) \cdot B(x)) \equiv (\forall x)A(x) \cdot (\forall y)B(y)$
13. $(\exists x)(A(x) + B(x)) \equiv (\exists x)A(x) + (\exists y)B(y)$

8.3. Standardformen

Konjunktive Normalform (KNF):
 $(A_1 + A_2 + \dots) \cdot (B_1 + B_2 + \dots) \cdot \dots$

Disjunktive Normalform (DNF):
 $(A_1 \cdot A_2 \cdot \dots) + (B_1 \cdot B_2 \cdot \dots) + \dots$

Regeln zur Umformung in Normalform:

1. Eliminierung aller Äquivalenzen (# 9)
2. Eliminierung aller Implikationen (# 8)
3. Einziehung der Negation nach innen (#6, #10, #11)
4. Einführung neuer Variablen für jeden Quantifizierer
5. Eliminierung aller Existenz Quantoren
6. Ausklammern der All-Quantoren und Entfallen dieser
7. Anwendung des Distributivgesetzes zur Transformation in Konjunktive Normalform (#5)
8. Eliminierung der UND-Verknüpfungen durch Auflistung der Klauseln
9. Einführung getrennter Variablen für jede Klausel

8.4. Theorembeweis mit Resolutionsverfahren

Allgemeines Resolutionsgesetz:

$$(X + A) \cdot (\neg X + B) \equiv (X + A) \cdot (\neg X + B) \cdot \underbrace{(A + B)}_{\text{Resolvente}}$$

Spezielles Resolutionsgesetz:

$$(X + A) \cdot (\neg X + A) \equiv A$$

Absorptionsgesetz:

$$(A + B) \cdot A \equiv A$$

Weitere Sonderfälle:

1. A
 2. $A + B$
 3. A
 4. $A \Rightarrow B \equiv \neg A + B$
- $$R \equiv B$$
- $$\neg A + B \quad R \equiv B + B = B$$
- $$R \equiv NIL$$
- $$B \Rightarrow C \equiv \neg B + C \quad R \equiv \neg A + C \equiv A \Rightarrow C$$

Anwendung beim Theorembeweis:

Geg.: Set von n existierenden und bewiesenen Axiomen $\mathcal{S} = \{S_1 \dots S_n\}$; Es gilt T zu beweisen

Vorgehen: Erweiterung von \mathcal{S} zu $\mathcal{S}^* = \{S_1 \dots S_n, \neg T\}$ Und Resolutionen bis leere Klausel erzeugt wird.

Erklärung: Statt Beweis wird Unerfüllbarkeit seines Gegenteils gezeigt.

Tautologie beweisen

Beweis durch Widerspruch: $\neg w \equiv f$:

1. Negiere Aussage
2. Bringe Aussage in KNF
3. Zeige Kontradiktion

9. Wissensrepräsentation

effizient speichern; strukturiert darstellen; Menge von Fakten, Regeln, Prozeduren, Modellen, Daten, Heuristiken; interpretierbar mit Hilfe von Repräsentationsmechanismen;

9.1. Prädikatenlogik

Aufteilung in Fakten und Regeln; Standardisiert durch KNF; Resolution als Inferenzmechanismus; Formulierung aufwändig und unnatürlich; zwingend Umformung in KNF;

9.2. Produktionsregeln

keine Umformung in KNF; Wenn-Dann bleibt erhalten; Vorwärts-Rückwärtsverkettung als Inferenzmechanismus; Darstellung im UND/ODER-Graphen; Fakten als Blatt, Regeln als Verzweigung;

9.3. Semantische Netze

Graphische Modelle zur Darstellung von Wissen über Beziehungen zw. Objekten; entsprechen etwa Fakten der Prädikatenlogik; Knoten = Objekte; Kanten = Prädikate; Verwendung bei natürlichsprachlichen Systemen; keine 2 Knoten gleicher Beschriftung; Richtung der Kanten von Bedeutung;

9.4. Rahmen

Darstellung der Zerlegung von Objekten oder Situationen in ihre Bestandteile; Ähnlichkeit zu semantischen Netzen, wesentlich mächtiger und flexibler; FrameName - zentraler Knoten, Slots - Kanten, Filler - Knoten;
1. Suchverfahren zur Ermittlung von Beziehungen;
2. „Rahmen-Abgleich“; Fakten als Fragezeichen markiert; mit aktuellen Daten auffüllen;

10. Handschrifterkennung

10.1. Vorverarbeitung

Eingabemethoden

1. freie Eingabe (hohe Vorverarbeitung)
2. liniengeführte Eingabe
3. feldgeführte Eingabe

Eingangssignal: $\vec{x}(t) = (x(t), y(t), p(t))^T$

x(t)	x-Koordinate
y(t)	y-Koordinate
p(t)	Druck (des Stifts)

Schriftgröße

1. Schätzen der Referenzlinien

2. Berechnung der Kernhöhe

3. Normierung des Schriftzuges

W: Höhe der Bins, P: Projektionsprofil

Oberlängenlinie: $y_{ober} = y_{max}$, Unterlängenlinie: $y_{unter} = y_{min}$

Kernlinie: $y_{kern} = \text{argmin}(\frac{d}{dy} P_y(j)) - 0.5)W + y_{min}$

Basislinie: $y_{grund} = \text{argmax}(\frac{d}{dy} P_y(j)) - 0.5)W + y_{min}$

Kernhöhe: $h_{kern} = |y_{kern} - y_{grund}|$

Normierung:

$$\vec{x}_{norm}[k] = \frac{1}{h_{kern}} \left[\begin{array}{c} x[k] - x_{min} \\ y[k] - (y_{grund} + \frac{h_{kern}}{2}) \end{array} \right]$$

10.2. Merkmalsextraktion

Extraktion aus dem normalisierten Schriftzug

Sekantensteigungswinkel:

$$\theta[k] = \frac{\pi}{2} + \begin{cases} \arctan(\frac{\Delta y}{\Delta x}) - \frac{\pi}{2} \text{sgn}(\Delta x) & \text{für } \Delta x \neq 0 \\ \frac{\pi}{2} (1 - \text{sgn}(\Delta x)) & \text{für } \Delta x = 0 \end{cases}$$

$$\Delta x = x_{norm}[k+1] - x_{norm}[k], \Delta y = y_{norm}[k+1] - y_{norm}[k]$$

Richtungsänderung:

$$\Delta \theta[k] = \theta[k+1] - \theta[k]$$

$$5\text{-dim. Merkmalsvektor: } \vec{m}[k] = \begin{bmatrix} \sin(\theta[k]) \\ \cos(\theta[k]) \\ \sin(\Delta \theta[k]) \\ \cos(\Delta \theta[k]) \\ p[k] \end{bmatrix}$$

10.3. Erkennung

Training und Erkennung läuft über Hidden-Markov-Modelle (HMM) mit Graphemen (z.B. Buchstabe, Sonderzeichen od. Ziffern) als kleinste Einheit

Training: Baum-Welch-Algorithmus

Erkennung: Viterbi-Algorithmus

11. Dialogsystem (Anhang)

- fortgeschrittene intuitive Ein-/Ausgabetechniken
- Hohes Maß an Interaktivität durch Benutzerfreundlichkeit und ausgeprägte Dialogfähigkeit
- Intelligentes Systemverhalten, selbstständig logische Schlüsse ziehen;

Teilgebiete der KI: Maschinelles Lernen, Bildverstehende Systeme, Expertensysteme, Robotik, Logik und automatisches Beweisen, natürlichsprachliche Systeme;

Zeilenneigung (skew)

Entropie: (B: Anzahl d. Bins, N(B_i): Anzahl d. Punkte in Bin i)

$$H_y(\alpha) = \sum_{i=1}^B I(i)$$

$$I(B_i) = -\frac{N(B_i)}{\sum_{j=1}^B N(B_j)} (\text{ld}) \frac{N(B_i)}{\sum_{j=1}^B N(B_j)}$$

Regressionsgerade $y = mx + b$:

$$m = \frac{\sum_{i=1}^N [(x_i - \bar{x})(y_i - \bar{y})]}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad \text{und } b = \bar{y} - m\bar{x}$$

$$\text{Rotation: } \vec{x}_{skew}[k] = \begin{bmatrix} \cos \alpha_0 & -\sin \alpha_0 & 0 \\ \sin \alpha_0 & \cos \alpha_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\vec{x}_{re}[k] - \vec{m}) + \vec{m}$$

Schriftneigung (slant)

1. Scherung der Schrift an der Grundlinie y_S
2. Scherung um den Winkel ϕ_0
3. Bestimmen von ϕ_0 mit Projektionsprofilen oder Richtungshistogrammen in x-Richtung, $H_x(\phi)$ muss möglichst klein sein

Scherung:

$$\vec{x}_{slant}[k] = \begin{bmatrix} 1 & -\tan \phi_0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\vec{x}_{skew}[k] - \begin{bmatrix} 0 \\ y_S \\ 0 \end{bmatrix}) + \begin{bmatrix} 0 \\ y_S \\ 0 \end{bmatrix}$$