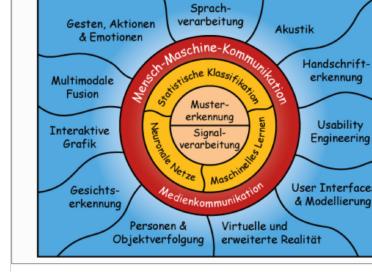


1. Allgemeine Einführung

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



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.

2.3. Menschliche Hörsinn

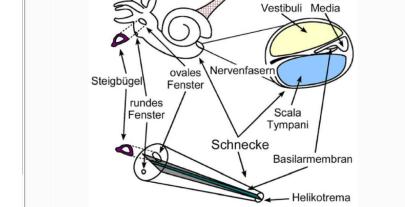
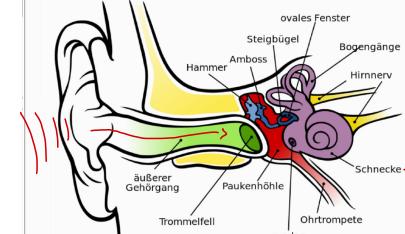
2.3.1. Das Ohr

Außenohr: Ohrmuschel & Gehörgang.

Mittelohr: Trommelfell, Gehörknöchelchen (Hammer, Amboss, Steigbügel) & Eustachische Röhre: Wandlung von Luftschwingung in mech. Schwingung.

Innenohr: Steigbügel über ovale Fenster in mit Flüssigkeit gefüllte Schnecke; Impedanzwandlung von Luft zu Flüssigkeit.

Basislarmembran Haarzellen (25k - 30k Rezeptoren) wandeln Schwingung in elektronische Nervenimpulse Frequenz-Ort-Wandlung, Zerlegung in Frequenzanteile \Rightarrow Hörnerv (30k Nervenfasern) \Rightarrow Hirn



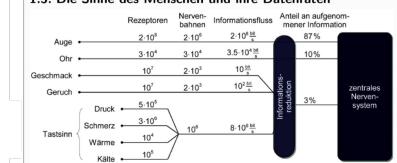
1.3. Trends in der MMK

- 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	taktile	
Druck Kraft	haptisch	mechanische Modal.
Berührung		
Vibration	taktile	oberflächen-sensitiv
Temperatur	thermorezeptorisch	
Bewegung und Orientierung	kinästhetisch	
Gleichgewicht	vestibular	

1.5. Die Sinne des Menschen und ihre Datenraten



Handschriftherkennung:

Offline-Erkennung

gegeben: Schriftzug als Bitmap

Anwendung: - Auswertung Briefadressen

- Überweisungsformulare

- Mit Online-Erkennung: bessere Erkennungsraten

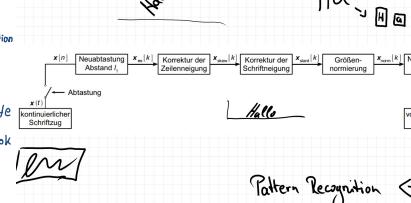
Online-Erkennung

zeitliche u. räumliche Information

zu jedem Ablaufpunkt der Schriftbewegung

z.B. Ort, Druck, Stiftbewegung

durch Schrift direkt bediente Geräte: PDA, Smartphone, Smart Paper Notebook



• Ohr vorarbeitet nicht alle Schallreize linear

• geringe Empfindlichkeit bei sehr tiefen und hohen Frequenzen

⇒ Psychoakustik berücksichtigt diese Eigenschaften d. Wahrnehmung und bietet ein subjektives Maß für das Ohr

Sone \Leftrightarrow dB(A)

• dB(A) berücksichtigt physiologische Beiträge nur ungenügend

• Lautheit N - wie wird Schall subjektiv wahrgenommen

1 sone: empf. Lautstärke bei $f = 1 \text{ kHz}$ und $L = 40 \text{ dB}$

Doppelter Wert \Rightarrow doppelt so laut empfundener Schall (+10 dB)

Umrechnung nur bei frequenzreinen Signalen möglich ($f = \text{const.}$)

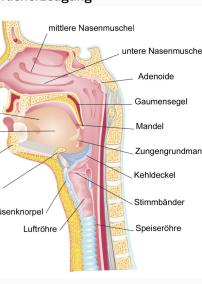
1.6. Datenraten gängiger Systeme der MMK

System	Verhalten	Rate (KByte/sec)
Tastatur (ungeübtl.)	Eingabe	0.01
Tastatur (geübtl.)	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

2.2. Schallquellen und ihre typischen Pegel

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

2.4. Menschliche Spracherzeugung

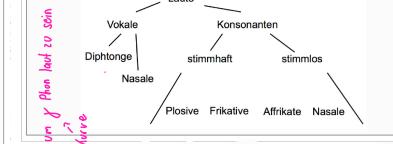


2.4.1. Phoneme

Das Phonem ist die kleinste bedeutungsunterscheidende Einheit des gesprochenen Wortes.

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

Systematische Einteilung der Phoneme:



2.3.2. Psychoakustik

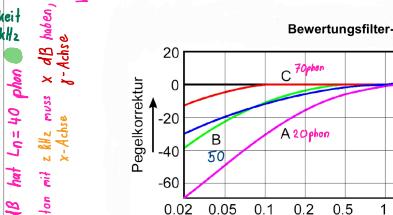
• Empfindlich vor etwa 20 Hz - 20 kHz (\approx 10 Oktaven)

• Starke Dämpfung für sehr niedrige und sehr hohe Frequenzen

• Resonanzfrequenz des Gehörgangs bei ca. 3...3.4 kHz

• Lautheit in [sone] 1 sone \triangleq Lautheit eines 1kHz Sinus mit 40 dB

• Verhältniszahl [mel] 1000 mel \triangleq 1000Hz



• dB(A): Bewerteter Schalldruckpegel nach Filterkurve A

• dB(A) hilfreicher, da die Frequenz bewertende Eigenschaft des Ohrs berücksichtigt und Filterkurve A zu „lesen“ Geräuschen passt

LPA

Verstärkender Effekt

• dB(A) \Leftrightarrow Sone

$N = 16 \text{ Sone} \Leftrightarrow 80 \text{ dB}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ kHz}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ ms}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Bit}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Winkelgrad}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Farben}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Geschmacksarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Gerüche}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Haptikaufnahmen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Temperaturwahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Schmerzwahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Wärmezufuhrwahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Geschwindigkeitswahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Orientierungswahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinneswahrnehmungen}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesmodi}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{ Sone} \Leftrightarrow 16 \text{ Sinnesarten}$

$N = 16 \text{$

3. Grammatiken

Natürlichsprachige Systeme; Modellierung von Dialogen.

3.1. Kontextfreie Grammatiken (CFG)

- $G = \{V, T, P, S\}$ 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:

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

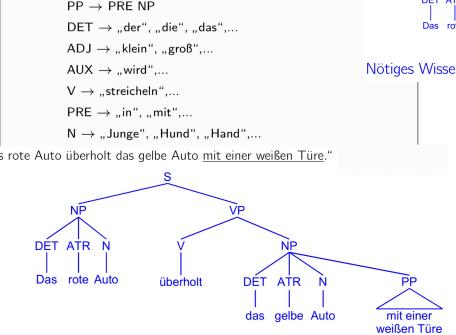
Grammatik-Grammatik:

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

„Das rote Auto überholt das gelbe Auto mit Vollgas.“

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

„Das rote Auto überholt das gelbe Auto mit einer weißen Tür.“



Nötiges Wissen: Das gelbe Auto hat eine weiße Tür. ✓

Grammatik P. Palindrome

$$\begin{aligned} V &= \{S\}, T = \{a, b\}, S = S \\ \Rightarrow & S \rightarrow a | b | aa | bb | aSa | bSb \end{aligned}$$

$$L = \{w | w = a^n b^m \wedge n, m \in \mathbb{N}\}$$

$$V = \{A, B, C\}, T = \{a, b, c\}, S = A$$

$$P = \begin{cases} A \rightarrow aA \mid C \\ B \rightarrow bB \mid E \\ C \rightarrow cB \end{cases}$$

alternativ:
B → b
C → C
leeres Symbol

Mehrdeutig
Grammatik ist ambig wenn für ein durch durch die Grammatik produzierbares Wort mehrere Parse trees existieren.

4. Automatentheorie

Verarbeitung von Symbolfolgen; Modellierung von Dialogen;

4.1. Zustandsautomat

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

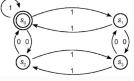
Beispiel für einen deterministischen Zustandsautomaten

$$\begin{aligned} \mathcal{S} &= \{s_0, s_1, s_2, s_3\} \\ \mathcal{X} &= \{0, 1\} \\ \mathcal{F} &= \{s_0\} \end{aligned}$$

Transitionsregeln in Tabellenform:

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

Beispiel für einen nicht-deterministischen Zustandsautomaten



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{S} Set mit endlicher Anzahl Zustände
- \mathcal{X} zulässiges Alphabet für die zu verarbeitende Symbolfolge X
- \mathcal{Y} zulässiges Alphabet für den Stack
- \mathcal{T} Transitionenfunktion für die Zustände in \mathcal{S}
- s_0 Anfangszustand
- y_0 Startsymbol für den Stack
- \mathcal{F} ein Set von festgelegten Endzuständen (leer wenn Endzustand über leeren Stack definiert ist) # Erstes Symbol auf Stack

Beispiel für einen Kellerautomaten:

$$\begin{aligned} \mathcal{S} &= \{S_0, S_1\} \\ \mathcal{X} &= \{a, b\} \xrightarrow{\text{z.B. Folge aab}} \\ \mathcal{Y} &= \{\#, \text{A}\} \\ y_0 &= \# \\ \mathcal{F} &= \{\} \text{ (Ende durch leeren Stack)} \end{aligned}$$

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

Angaben in Klammern:

(Voraussetzung auf Stack in \mathcal{Y} , Eingabe in \mathcal{X} , Aktion push(...)/pop)

$$P = \{S \rightarrow aB | bA, A \rightarrow a | aS | bAA, B \rightarrow b | bS | aBB\} \quad S = aaabbabbba$$

1. Anzahl Symbole = Anzahl an Ausgängen

2. Nicht \emptyset sondern \emptyset

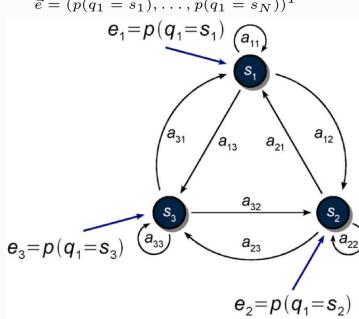
6. Hidden-Markov-Modelle und Algorithmen

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

6.1. Markov-Modelle (MM) Von Zeile zu Spalte

Abbildung stochastischer Prozesse, deren aktueller Zustand nur vom vorangegangenen 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 Zustandsautomaten; Zustandsübergänge und Symbolemissionen nicht deterministisch.

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

$$B = \begin{bmatrix} p(v_1 | s_1) & \dots & p(v_1 | s_N) \\ \vdots & \ddots & \vdots \\ p(v_M | s_1) & \dots & p(v_M | s_N) \end{bmatrix}^{V_1 \dots V_N}$$

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

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

Dabei durchlaufende (vorliegende/hidden) Zustandsfolge:

Zustände nicht direkt sichtbar

HMM - Eigenschaften Beobachtungen über Emissionswsk.

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

- Wo sind „0“-Werte in B , e_1, q_T ?
- Einsprungmöglichkeiten
- An Ende alles außer q_T streichen

Trellis Diagramm zeichnen:

1. „Unmögliche“ Zustände streichen
2. Alle möglichen Übergänge einzeichnen
3. Streiche alle eingehenden Kanten in einen Zustand ohne ausgehende Kanten (= Sackgassen)

6.3. HMM in der Spracherkennung

Cepstrum; Merkmalsextraktion; 12D Merkmalsvektor;

6.3.1. Modelle

Einzelworterkennung vs. fließende Sprache; Phoneme, kleinste bedeutsungsunterscheidende 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 g 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 Produktionswahrscheinl.

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)

Info über Wahrscheinlichkeit geht vorwärts

1. Initialisierung:

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

Info über Wahrscheinlichkeit, dass man v_i in Zustand s_i beobachtet

2. Induktion:

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

3. Terminierung

$$P(o | \lambda_k) = \sum_{i=1}^N \alpha_T(i) = d_x(1) + d_x(2) + d_x(3) + \dots$$

Benötigte OPS $\sim T \cdot N^2$ Letzte Iteration

Brute force: Für Produktionswsk. \Rightarrow Alle Pfade summieren

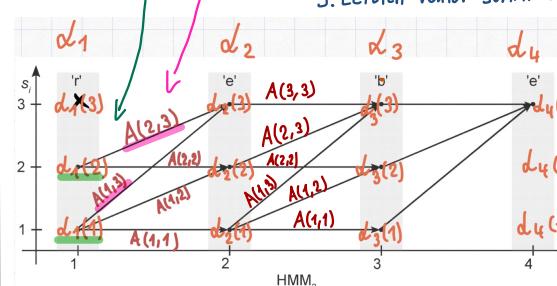
Vorwärtsalgorithmus:

1. Anfangswerte d_1 bestimmen ($d_1(i) = e_i \cdot b_i(o_1)$)

2. Ab d_2 : Pfade zu diesem Knoten $d_1(y)$ anschauen

[doppelt: Übergang + ...] $\cdot b_i(o)$

3. Letzten Vektor summieren



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 “ aller Wörter, die diesen Zustand aufweisen.

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

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 “

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

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

6.4.4. Viterbi-Algorithmus

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:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(o_t) \quad 2 \leq t \leq T; \quad 1 \leq j \leq N$$

3. Terminierung:

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

4. Ermittlung der wahrsch. Zustandsfolge: $q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \dots, 1 \quad \dots \rightarrow 3 \rightarrow 1 \rightarrow 2$

Backtracking über höchste Werte in δ oder Ψ

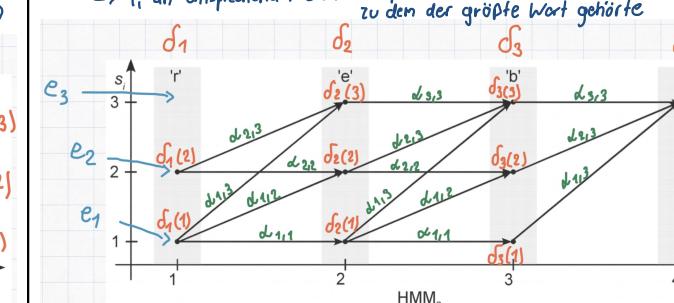
Viterbi Algorithmus

1. Anfangswerte ($\delta_1(i) = e_i \cdot b_i(o_1)$), $\Psi_1 = \begin{cases} s & \text{wenn } e_i \neq 0, \text{ sonst } x \\ s & \text{wenn } e_i = 0, \text{ sonst } x \\ s & \text{wenn } e_i = 0, \text{ sonst } x \end{cases}$

2. Ab d_2 : Alle Pfade zu aktuellem Knoten betrachten

\rightarrow Pfad mit größtem Wert nehmen ($\delta_{t-1}(i) \cdot A(i, j)$)

$\rightarrow \Psi_i$ an entsprechender Stelle entspricht dem Zustand zu dem der größte Wert gehörte



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:

a) Queue leer? \Rightarrow SZiel nicht gefunden"

b) Entnehme nächsten Knoten

c) Knoten == Ziel? \Rightarrow SZiel erreicht"

d) Schreibe alle Kinder des Knotens in die Queue

Update Queue

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

Suchalgorithmus

Art der Queue

Breitensuche FIFO-Queue

Tiefensuche LIFO-Queue (Stack)

A-Suche Priority-Queue

A*-Suche Priority-Queue mit heuristischen Kosten als Priorität

Dijkstra Priority-Queue mit bisherige Weg als Heuristik

7.2. Tiefensuche und Breitensuche

1. einelementige Liste mit Wurzelknoten

2. bis Liste leer / Ziel erreicht:
-prüfe erstes Element auf Ziellisten 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 / ausgewählte Suchbaum muss gespeichert werden;

7.3. Heuristische Suche / A-Suchalgorithmus

Verarbeitung zusätzlicher Informationen; Bewertungsmöglichkeit für Erfolgsaussichten eines bestimmten Pfades; Entscheidungen ordnen; Vielversprechende Alternative zuerst, „dem atm billigsten folgen“; Heuristik besteht in Definition einer geeigneten Bewertung (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

7.4. A*-Algorithmus optimistisch schätzen !!

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

3. $P^* = \text{größter Wert aus letztem } \delta$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{„alle Aufenthalte im Zustand } s_i \text{“}$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{„alle Übergänge vom Zustand } s_i \text{ zum Zustand } s_j \text{“}$$

$$\delta_{ij} = \frac{\text{„alle Übergänge vom Zustand } s_i \text{ zum Zustand } s_j \text{“}}{\text{„alle Aufenthalte im Zustand } s_i \text{“}}$$

$$\delta_{ij} = \frac{\text{„alle Aufenthalte im Zustand } s_i \text{ mit Beobachtung des Symbols } s_j \text{“}}{\text{„alle Aufenthalte im Zustand } s_i \text{“}}$$

$$\begin{aligned} \delta_{ij} &= \frac{\sum_{t=1}^T (\gamma_t(j) \cdot \delta_{ij})}{\sum_{t=1}^T \gamma_t(j)}, \text{ mit Kronecker-Delta } \delta_{xy} = \begin{cases} 1, & \text{wenn } x = y \\ 0, & \text{sonst} \end{cases} \\ \delta_{ij} &= \frac{\sum_{t=1}^T (\gamma_t(j) \cdot \delta_{ij})}{\sum_{t=1}^T \gamma_t(j)} \end{aligned}$$

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 Schlussfolgern (Resolution, Unifikation - effektiver);

Umformregeln:

1. Doppelte Negation $\neg\neg A \equiv A$
2. Idempotenz $A + A \equiv A$ und $A \cdot A \equiv A$
3. Kommutativität $A + B \equiv B + A$
4. Assoziativität $A + (B + C) \equiv (A + B) + C$
5. Distributivität $A + (B \cdot C) \equiv (A + B) \cdot (A + C)$
6. De Morgan $\neg(A \cdot B) \equiv \neg A + \neg B$
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. $\neg(\forall x) A(x) \equiv (\exists x) \neg A(x)$ $\neg(\exists x) A(x) \equiv (\forall x) \neg A(x)$
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: $(A_1 \cdot A_2 \cdot \dots) + (B_1 \cdot B_2 \cdot \dots) + \dots$

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

a	w	w	f	f	Ausdruck	Name	Umgangssprachlich
b	w	f	w	w	$\neg a$	Negation	Nicht A
w	f	w	w	w	$a \cdot b$	Konjunktion	A und B
w	w	w	f	a + b	Disjunktion	A oder B	
f	w	w	f	$a \leftrightarrow b$	Excl. Disjunktion	Entweder A oder B	
w	f	w	w	$a \Rightarrow b$	Implikation	Wenn A, dann B	
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	

a	b	wichtige	a · b	a + b	„oder“	xor	implikation	äquivalenz
0	0	1	0	0	0	1	1	1
1	0	0	0	1	1	0	0	0
0	1	1	0	1	1	1	0	0
1	1	0	1	1	0	1	1	1

Drück: $p_n = p_2 + k(p_3 - p_2)$



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

Anwendung beim Theorembeweis:

Geg: Satz von n existierenden und bewiesenen Axiomen $S = \{S_1 \dots S_n\}$; Es gilt T zu beweisen
Vorgehen: Erweiterung von S zu $S^* = \{S_1 \dots S_n, \neg T\}$ Und Resolution in leere Klausel erzeugt wird.
Erklärung: Statt Beweis wird Unerfüllbarkeit seines Gegenteils gezeigt.

Tautologie beweisen:

1. Wahrheit auf KNF bringen $\begin{array}{l} 1. \text{ Negiere Formel} \\ 2. \text{ Bringe Formel in KNF} \\ 3. \text{ Zeige, dass Gegenteil } \{ \text{ ist.} \end{array}$
2. Gegenteil auf KNF bringen $\begin{array}{l} 2. \text{ Bringe Formel in KNF} \\ 3. \text{ Zeige Kontradiktion} \end{array}$
3. Zeige, dass Gegenteil $\{ \text{ ist.}$

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-Ablegung“: Fakten als Fragezeichen markiert; mit aktuellen Daten auffüllen;

S_n berechnen, liegt auf $[S_2, S_3] = S_{23}$, Abstand ℓ zu s_1

$$|S_{1n}| = \ell = |S_2 + k \cdot S_{23} - s_1| = |S_{12} + k \cdot S_{23}|$$

$$|S_{12} + k \cdot S_{23}|^2 = \ell^2 \Rightarrow S_{12}^2 + k^2 \cdot S_{23}^2 + 2 \cdot S_{12} \cdot S_{23} \cdot k + (S_{12} - \ell)^2 = 0$$

$$k = \frac{-2 \cdot S_{12} \cdot S_{23} + \sqrt{4 \cdot (S_{12} \cdot S_{23})^2 + 4 \cdot S_{23}^2 \cdot (\ell^2 - S_{12}^2)}}{2 \cdot S_{23}^2}$$

$$\Rightarrow S_n = S_2 + S_{23} \cdot \begin{cases} k_1, & \text{falls } 0 \leq k_1 \leq 1 \\ k_2, & \text{sonst} \end{cases}$$

10. Handschrifterkennung

10.1. Vorverarbeitung

Eingabemethoden

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

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

$$\begin{aligned} S_{12} &= \overrightarrow{S_{12}} = s_2 - s_1 & s_1 & \text{x-Koordinate} \\ y(t) & S_{23} = \overrightarrow{S_{23}} = s_3 - s_2 & s_2 & \text{y-Koordinate} \\ p(t) & & & \text{Druck (des Stifts)} \end{aligned}$$

10.1.1. Abtastung

Neublastung: $\ell = \text{konst}$

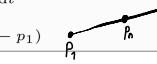
Abtastung / Neublastung

1. Diskretisierung von $\vec{x}(t)$ mit $n \cdot \Delta t \Rightarrow$ zeitäquidistante Abtastung
2. Lineare Interpolation der Stifttrajektorie mit euklidischem Abstand $d(s_i, s_j) = \sqrt{\sum (x_i - x_j)^2}$
3. **Neublastung** \Rightarrow ortäquidistante Abtastpunkte $\vec{x}_{re}[k]$

Länge einer Kurve $\vec{r}(t) = (x(t), y(t))^T$:

$$L(a, b) = \int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt$$

Druckkomponente: $p_n = p_1 + k \cdot (p_2 - p_1)$



lineare Interpolation

polynomiale (quadratische) Interpolation

Steigung

Kleinste Quadrate

Schriftneigung (slant)

Punkte auf Grenze \rightarrow nächsthöherer Bin

Rekurrenz

10.2. Merkmalsextraktion

Extraktion aus dem normalisierten Schriftzug

Sekantensteigungswinkel:

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe erfüllen? Achten Sie auf die korrekte Reihenfolge!

1) Komposition der Zeitreihenfolge,

2) Kompensation der Schriftfreigabe,

3) Größenbildung auf eine Kenngröße von 1.

Welche Voraussetzung müssen die bestimmbaren Eckenlinien nach obigen Schrittangabe er