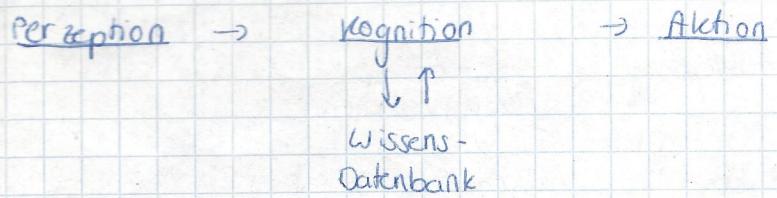


Kognitive Systeme



Perception = Aufnahme der Umwelt durch Sensoren

Wissens-DB = Repräsentation gelernter Zusammenhänge und Reaktionen darauf

Kognition = auf Wissens-DB arbeitende Algorithmen
Deduktion, Induktion, Lernen

Aktion = Veränderung der Umwelt

Maschinensehen: modelliert visuellen Sinn des Menschen
Beobachtung und Prädiktion

Bildverarbeitung

Bildrepräsentation

HSI-Modell: hue, Saturation, Intensity
unempfindlich gegen Beleuchtungsänderung

RGB → HSI:

- $R=G=B$: H undef.
- $R=G=B=0$: S undef.

$$c = \arccos \frac{2R-G-B}{\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$H = \begin{cases} c, & B \leq G \\ 360^\circ - c, & \text{sonst} \end{cases}$$

$$S = 1 - \frac{3}{R+G+B} \min(R, G, B)$$

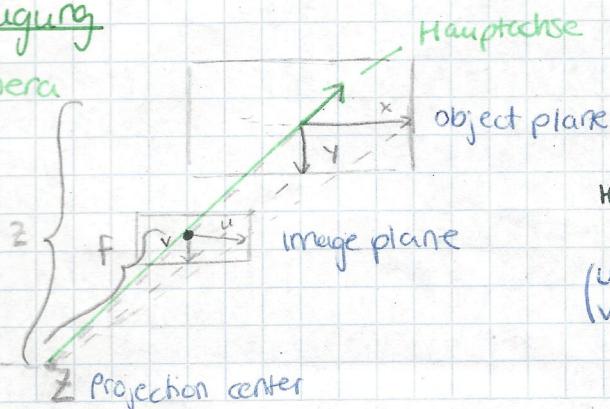
$$I = \frac{1}{3}(R+G+B)$$

RGB2L → 8bit Graustufen

$$g = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

Bilderzeugung

Lochkamera



Hauptpunkt = Ursprung Bildkoord. sys.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z} \begin{pmatrix} x \\ y \end{pmatrix}$$

Bildverarbeitung

Punktooperatoren unabh. von Position/Nachbarn eines Pixels
 $I'_{\text{mg}}(u,v) = f(I_{\text{mg}}(u,v))$

Anwendung: Kontrasterhöhung/-vermindlung
 Helligkeits - " -

Invertierung
 Kombinationen

$$f: [0..q] \rightarrow [0..q], x \mapsto ax + b \quad (\text{affine Punktoperation})$$

Algo: $\forall (u,v) \in I:$

$$g = \text{round}(a \cdot I(u,v) + b)$$

$$I'(u,v) = \begin{cases} 0 & , g < 0 \\ q & , g > q \\ g & , \text{sonst} \end{cases}$$

homogen = unabh. von Pixelpos. (u,v)
 affin = abh. von pos. (u,v)

Kontrastanpassung

Spreizung affine Punktop.

$$I'(u,v) = q \cdot \frac{I(u,v) - \min}{\max - \min}$$

Grauwertistogramm Häufigkeit eines Grauwerts im Bild $H(x) = \#(u,v) : I(u,v) = x$
 $x \in [0,..,q-1]$

Histogrammdehnung Verbesserung der Spreizung, affine Punktop.

Quantilsfunktion $H_q(p)$ ($p \in [0,1]$)

$$H_q(p) = \inf \{x \in \{0,..,q\} : H_a(x) \geq p \cdot H_a(q)\}$$

$$\text{mit } H_a(x) = \sum_{k=0}^x H(k) \quad (\text{akkumuliertes Histogramm})$$

\Rightarrow Histogrammdehnung:

$$I'(u,v) = a \cdot I(u,v) + b$$

$$a = \frac{q}{\max - \min} ; \quad b = - \frac{q \cdot \min}{\max - \min} \quad (= \text{Spreizung})$$

$$\min = H_q(p_{\min}), \max = H_q(p_{\max})$$

Histogrammausgleich erhöht Kontrast in Bereichen des Histogramms mit vielen Treffern

Algo: for $x=0$ to q :
 $H_n(x) = \text{round} \frac{q \cdot H_a(x)}{H_a(q)}$

for $\forall (u,v) \in I$:

$$I'(u,v) = H_n(I(u,v))$$

Bildanalyse durch Frequenzanalyse

Niedrige Frequenz = schwacher Grauwertübergang

Fourier-Transformation $f(x)$ diskret $\Rightarrow F(u)$ periodisch

1-dim. kontinuierlich

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{2i\pi u x} du \quad F(u) = \int_{-\infty}^{\infty} f(x) e^{-2i\pi u x} dx$$

2-dim. diskret (Bildverarb.)

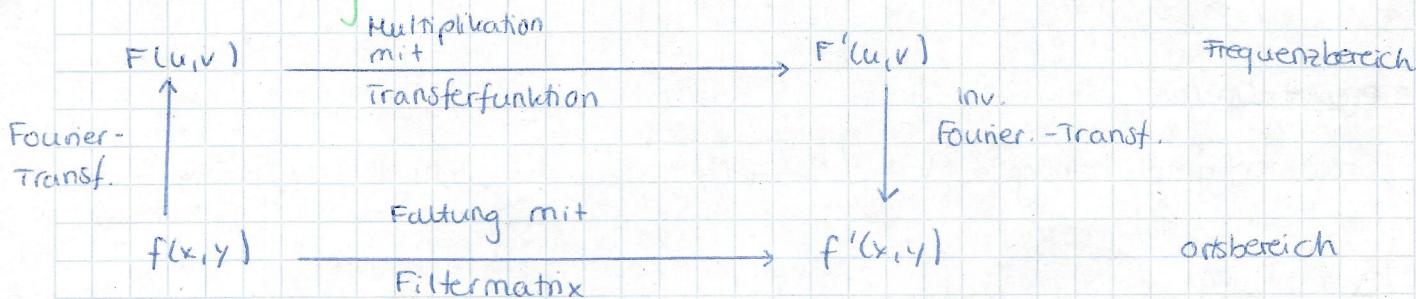
$$F(u,v) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f[x,y] \exp(-2i\pi(ux+vy)\tau)$$

u, v = Frequenzvariablen

$$F(u,v) = R(u,v) + I(u,v) \quad \text{Real- und Imaginärteil}$$

$$F(u,v) = |F(u,v)| + \exp(i\varphi(u,v)) \quad \text{Betrag (Spektrum) und Phase}$$

Bildbearbeitung



Ortsbereich

Faltung 1D kontinuierlich

f = Filterfunktion

g = Bildfunktion

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a) g(x-a) da$$

Faltung 1D diskret

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a) g(x-a)$$

Faltung Filtermatrizen

neuer Bildwert ist gewichtete Summe der Pixel unter gespiegelter Matrix
→ Gewichte = Matrizenwerte

Filteroperationen

Tiefpass: Glättung, Rauschelim.

- Mittelwert
- Gauß

Hochpass: Kanteneklektion

- Prewitt
- Sobel
- Laplace
- Roberts

Mittelwertfilter

Maske: $\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ \rightarrow Mittelwert der 8er-Nachbarschaft

Gauß-Filter

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \leftrightarrow \quad F(u,v) = e^{-\frac{u^2+v^2}{2\sigma^2}}$$

zB $\sigma = 0,85$: $F_G = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$

σ bestimmt Stärke der Glättung

$n \times n$: beeinflusst Approximationsgüte $\rightarrow n = \lfloor 2\sigma \rfloor + 2 + 1$

Prewitt-Filter

Prewitt-X: $p_x = \frac{\partial g(x,y)}{\partial x}$

$$p_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Prewitt-Y: $p_y = \frac{\partial g(x,y)}{\partial y}$

$$p_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Prewitt-Operator

$$\text{Grauwertgrad. betrag } M \approx \sqrt{p_x^2 + p_y^2} \quad \rightarrow \text{Schwellwertfilterung}$$

Sobel-Filter

Sobel-X: $s_x = \frac{\partial g(x,y)}{\partial x}$

$$s_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Sobel-Y: $s_y = \frac{\partial g(x,y)}{\partial y}$

$$s_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

$$M \approx \sqrt{s_x^2 + s_y^2} \quad \rightarrow \text{Schwellwertfilterung}$$

Roberts-Filter Kanten detektion

$$R(g(x,y)) = |R_x(g(x,y))| + |R_y(g(x,y))|$$

$$R_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad R_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Laplace-Filter

$$\nabla^2 \approx \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Kombination Hoch-Tiefpass: LoG

$$\text{LoG}(g(x,y)) = \nabla^2 (f(x,y) * g(x,y))$$

$$\nabla^2 F(x,y) = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Gauß-Filter

Garry-Kantendetektor

- (1) Gauß-Filter
 (2) Prewitt $x+y$ oder Sobel $x+y$

→ Berechne $\varphi = \arctan(g_y/g_x)$ Richtung
 Einteilung Richtung in 4 Quadranten
 $[-67,5^\circ, -22,5^\circ]$ $[22,5^\circ, 67,5^\circ]$
 $[-22,5^\circ, 22,5^\circ]$ $[67,5^\circ, 90^\circ]$

- (3) Non-Maximum Supression
 (4) Hysterese-Schwellwertverfahren

Non-Max-Sup.

Gradient muss lokales Max. sein bei Betrachtung der 2 direkten Nachbarn in Gradientenrichtung

Hyster.-schw. 2 Schwellwerte low, high
 Betrag des Gradienten für Pixel > high → akzeptieren
 ↳ verfolge anhand dieser Pixel Kanten
 → 8 Nachbarn prüfen

Segmentierung

Schwellwertfilterung

Schwellwert T

$$(img'(u,v) = \begin{cases} g & , (img(u,v)) \geq T \\ 0 & , \text{sonst} \end{cases}$$

Multilevel Otsu Verfahren Findet optimale Schwellwerte
 Interklassen Varianz σ_b^2 möglichst groß
 Intraklassen Varianz σ_w^2 möglichst klein

Farbsegmentierung

Probleme: wechselnde Lichtbed., Reflexionen, Schattenwürfe

Verfahren: Histogrammbasiert

Intervallshranke im HSV-Farbraum

Mahalanobis-Distanz

Morphologische Operatoren

Dilatation: vergrößert Pixel zu Bereichen → Vereinigung Maske + Bild
 Hohlräume schließen
 Objekte verbinden/vergrößern

Erosion: entfernen einzelner Pixel, schwach zsm. hängender Pixelgruppen
 dünne Linien verschwinden

→ verunden Maske + Bild

Opening: Erosion → Dilatation

Closing: Dilatation → Erosion

Bewegung

Differenzbild $\text{Img}'_t(u, v) = (\text{Img}_t(u, v) - \text{Img}_{t-1}(u, v))$ Schwellwertfiltern

Region Growing

- (1) Saatpunkt $p_0(u_0, v_0)$ wählen
- (2) Region $R = \{p_0\}$, Schwelle ϵ
- (3) solange $\exists p \in R, q \notin R$ mit $\|p - q\| \leq 1$ (Nachbarn), $|\text{Img}(p_0) - \text{Img}(q)| \leq \epsilon$
 $R = R \cup \{q\}$

3D-Transformationen

Rotationen

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

regular, invertierbar, $\det = 1$

$$H = \begin{pmatrix} u & v & x & 1 \\ j & j & x & 0 \\ R^{3 \times 3} & y & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H^{-1} = \begin{pmatrix} R^{3 \times 3} & -u & -v & -x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quaternionen

$$q = (q_w, q_x, q_y, q_z) = q_w + iq_x + jq_y + kq_z \quad \text{nicht Kommutativ!}$$

$$\begin{aligned} i^2 = j^2 = k^2 = ijk &= -1 & \Rightarrow \bar{q} = (q_w, -q) \\ ij &= k & ji &= -k \\ jk &= i & kj &= -i \\ ki &= j & ik &= -j \end{aligned}$$

$$\Rightarrow \|q\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2}$$

$$\Rightarrow q^{-1} = \frac{\bar{q}}{\|q\|^2}$$

$$\Rightarrow q = (\cos \frac{\theta}{2}, u \cdot \sin \frac{\theta}{2}) \quad \text{Winkel } \theta, \text{ Rot.-achse } u, \|u\| = 1$$

$\Rightarrow qa\bar{q}$ rotiert v mit $a = (0, v)$ um u mit Winkel θ

$$\Rightarrow \text{SLERP: } q \frac{\sin((1-t)\theta)}{\sin \theta} + r \frac{\sin(t\theta)}{\sin \theta} \quad \text{berechnet kürzeste Verb. zw. } q, r \text{ mit Winkel } \theta \text{ zw. } r, q$$

Kameramodelle

Erweitertes Kameramodell

→ Intrinsiche Parameter: Brennweite, Bildhauptpunkt
 \rightsquigarrow def. Abb. von Kamerakoords. in Bild-KS

→ Extrinsiche Parameter: def. Beziehung zw. Kamera-KS, WCS

→ nicht quadr. Pixel $\rightarrow f_x, f_y$ (Einheit: Pixel)

$$\begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = K \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (c_x, c_y) = \text{Hauptpunkt}$$

K = Kalibriermatrix

Extrinsische Kamera Kalibrierung

WKS \rightarrow Kamera K'S

$$x_c = Rx_w + t \quad R = \text{Rot. matr.} \quad t = \text{Transl.}$$

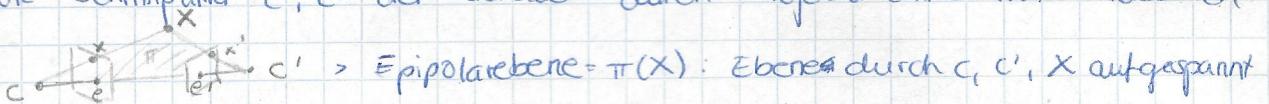
Projektionsmatrix

$$P = (KR|Kt) \Rightarrow \begin{pmatrix} uw \\ vw \\ w \end{pmatrix} = P \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Berechnung: Direkte Lineare Transformation

Epipolar Geometrie Zusammenhang zw. zwei Kameras

> Epipole = Schnittpunkt e, e' der Geraden durch Projektionszentren mit Bildebenen



> Epipolarlinie = Schnittgerade von $\pi(X)$ mit Bildebene

Fundamentalmatrix 3×3 , Rang = 2

Epipolarlinien

$$e = F^T x' \quad , \quad e' = F^T x$$

Epipole

$$Fe = 0 = F^T e'$$

Rektifizierung

bewirkt, dass alle Epipolarlinien horizontal mit derselben v -Koordinate verlaufen, nach Korrespondenzen muss nur noch in eine Richtung gesucht werden

$$F = K'^{-T} E K^{-1} \quad \text{mit} \quad E = [t]_x \times R = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} R$$

Kamera 1 Transformation: $(1|1|0)$
2 : $(R|I|t)$

Hough-Transformation Ziel: Geraden-Erkennung

Geradengleichung $x \cos(\theta) + y \sin(\theta) = r$

r = Abst. vom Nullpunkt, Normalenvektor \vec{n} $\rightarrow \theta$ = Winkel von \vec{n} zur x-Achse

zB Punkt $P_1 = (-1|1|2)$ $\Rightarrow r = -\cos \theta + 2 \sin \theta$

Signalverarbeitung

Faltung = Filtern eines Signals

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau \quad \text{kontinuierlich}$$

$$(f * g)[i] = \sum_{j=-\infty}^{\infty} f[i-j] g[j] \quad \text{diskret}$$

$$\geq F(f_1(t) * f_2(t)) = F_1(\omega) \cdot F_2(\omega)$$

$$\geq F(f_1(t) \cdot f_2(t)) = F_1(\omega) * F_2(\omega)$$

Graphisch: g spiegeln, von links gegen f schieben
 → bis hab wann hat Faltung Wert 0?
 ansonsten: Wert = Fläche der Überlappung

Dirac Distribution = beliebig schmale, beliebig hohe Funktion mit Fläche 1

$$S(f) = \int_{-\infty}^{\infty} S(x) f(x) dx = f(0)$$

$$\Rightarrow \int_{-\infty}^{\infty} S(x) g(x) dx = g(0) \quad \Rightarrow \int_{-\infty}^{\infty} S(x-t) g(x) dx = g(t)$$

Abtastung (Sampling)

Häufigkeit = Abtastrate (in Hz)

Multiplication des Signals mit Impulsequig

$$s_{\text{alt}}(t) = s(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Abtasttheorem

Abtasten mit Frequenz > 2 · F (F = Grenzfreq.)

Fourier-Transformation

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(\omega nt) + b_n \sin(\omega nt) \quad \omega = \frac{2\pi}{T}$$

$$\text{Amplitude-Phase: } f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \sin(\omega nt + \varphi_n) \quad A_n = \sqrt{a_n^2 + b_n^2} \quad \tan \varphi_n = \frac{a_n}{b_n}$$

$$\text{Komplexe Darstellung: } f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega nt} \quad c_n = \begin{cases} \frac{1}{2} a_0, & n=0 \\ \frac{1}{2} (a_n - i b_n), & n>0 \\ \frac{1}{2} (a_{-n} + i b_{-n}), & n<0 \end{cases}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad \Leftrightarrow f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

$$F(\omega) = |F(\omega)| e^{i\vartheta(\omega)}$$

↑ Phasenspektrum
 Spektrum Amplitudenspektrum

$$\begin{aligned} F(f * g) &= F(f) \cdot F(g) \\ F(f \cdot g) &= F(f) * F(g) \end{aligned}$$

Komplexes Spektrum

$$\cos(\varphi) = \frac{1}{2}(e^{i\varphi} + e^{-i\varphi})$$

$$\sin(\varphi) = \frac{1}{2i}(e^{i\varphi} - e^{-i\varphi})$$

Fouriertransformationen

$$f(x) = a \sin(\omega_0 x)$$

$$\Leftrightarrow F(\omega) = ia\pi S(\omega + \omega_0) - ia\pi S(\omega - \omega_0)$$

$$f(x) = a \cos(\omega_0 x)$$

$$\Leftrightarrow F(\omega) = a\pi S(\omega - \omega_0) + a\pi S(\omega + \omega_0)$$

Aliasing

Überlappung beim Abtasten

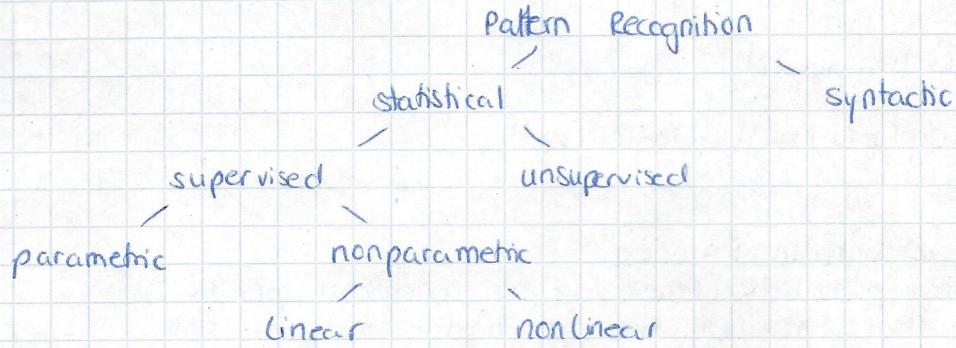
Maßnahmen:

- Erhöhen der Abtastrate \ominus große Datenmenge
- Tiefpassfilter \ominus Informationsverlust

Klassifikation

Pattern Recognition

Static Patterns: no dependence on Time or Sequential Order



Supervised

Class to be recognized is known for each sample in training data
 → requires a priori knowledge

unsupervised

Class not known, structures and features to be discovered automatically
 → e.g. clustering

Parametric (Gauß)

assume underlying probability distribution, estimate parameters of this distribution

non-parametric k-nearest-neighbor

don't assume distribution
 estimate probability of error or error criterion directly from training data

Bayes Decision Theory

$$\text{Bayes-Rule: } P(w_j | x) = \frac{p(x | w_j) P(w_j)}{P(x)}$$

$$P(x) = \sum_j p(x | w_j) P(w_j)$$

$p(w_j)$ a priori probability $\xrightarrow{\text{observation of } x}$ a posteriori probability $p(w_j | x)$

$p(x | w_j)$ = class-conditional Probability density

minimale Fehlerrate beim Schnitt
 $P(w_1) p(x | w_1) = P(w_2) p(x | w_2)$

$$\text{min. Fehlerwkt: } p(\text{Fehler}) = P(w_1) \int_{\text{S}}^{\max} p(x | w_1) dx + P(w_2) \int_{\text{S}}^{\max} p(x | w_2) dx$$

$$P(\text{error} | x) = \begin{cases} p(w_1 | x) & \text{if we decide for } w_2 \\ p(w_2 | x) & \text{else} \end{cases}$$

~ Decide w_1 if $p(w_1 | x) > p(w_2 | x)$
 w_2 otherwise

$$\text{or } \begin{cases} w_1 & \text{if } p(x | w_1) P(w_1) > p(x | w_2) P(w_2) \\ w_2 & \text{otherwise} \end{cases}$$

~ mehr als 2:
 Decide w_i if $p(w_i | x) > p(w_j | x) \quad \forall j \neq i$

$S = \text{Grenzwert}$
 $\max = \text{max. Det. Wert}$

Classifier discriminant functions

$$g_i(x), i = 1, \dots, c$$

Assign x to class w_i if $g_i(x) > g_j(x) \forall j \neq i$

$$g_i(x) = p(w_i|x) = \frac{p(x|w_i) P(w_i)}{\sum_{j=1}^c p(x|w_j) P(w_j)}$$

$$g_i(x) = p(x|w_i) P(w_i)$$

$$g_i(x) = \log(p(x|w_i)) + \log(P(w_i))$$

Classifier Design in Practice

need a prior probability $P(w_i)$, class conditional $p(x|w_i)$
 \rightarrow assume that $p(x|w_i)$ has particular parametric form

Gaussian Classifier

Univariate Normal Density $p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \sim N(\mu, \sigma^2)$

Multivariate Density $p(\vec{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^1/2} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right) \sim N(\mu, \Sigma)$

$$\Rightarrow g_i(\vec{x}) = -\frac{1}{2} (\vec{x} - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) - \frac{1}{2} \log(2\pi) - \frac{1}{2} \log|\Sigma_i| + \log P(w_i)$$

\rightarrow estimate covariance matrix Σ_i , mean vector $\vec{\mu}_i$ from training data
 for each class:

Estimation of Parameters

Maximum Likelihood Estimation

$$\vec{\mu}_i = \frac{1}{N} \sum_{k=1}^N \vec{x}_k$$

$$\text{Erwartungswert } \mu = \frac{1}{n} \sum_{i=1}^n x_i, \text{ Varianz } \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

$$\Sigma_i = \frac{1}{N} \sum_{k=1}^N (\vec{x}_k - \vec{\mu}_i)(\vec{x}_k - \vec{\mu}_i)^T$$

Curse of Dimensionality

adding more features indiscriminately leads to worse performance

\hookrightarrow select features carefully
 reduce dimensionality
 PCA

mit steigender Dim. der Daten wächst die Anzahl benötigter Trainingsdaten exponentiell an

Principle component Analysis (PCA)

Assumption: single dimensions are correlated

\rightarrow reduce number of dimensions with minimum loss of information

(1) Find axis along highest variance

(2) Rotate space along the axis

\rightarrow dimensions now uncorrelated

(3) Remove dimensions with low variance

Risk

$\Omega = \{\omega_1, \omega_2, \dots, \omega_S\}$
states of nature

$A = \{a_1, \dots, a_A\}$
possible actions

Loss Function

$\lambda(a_i | \omega_j)$ loss of action a_i given state ω_j

$$P(\omega_j | \vec{x}) = \frac{P(\vec{x} | \omega_j) P(\omega_j)}{P(\vec{x})}$$

Expected loss of taking action a_i : $R(a_i | \vec{x}) = \sum_{j=1}^S \lambda(a_i | \omega_j) P(\omega_j | \vec{x})$
 → minimize expected loss by selecting action a_i that minimizes conditional risk

Decision

$$\lambda(a_i | \omega_j) = \lambda_{ij}$$

decide ω_1 if $R(a_1 | \vec{x}) < R(a_2 | \vec{x})$

$$\rightarrow (\lambda_{21} - \lambda_{11}) P(\omega_1 | \vec{x}) > (\lambda_{12} - \lambda_{22}) P(\omega_2 | \vec{x})$$

$$\rightarrow (\lambda_{21} - \lambda_{11}) P(\vec{x} | \omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) P(\vec{x} | \omega_2) P(\omega_2)$$

$$\rightarrow \frac{P(\vec{x} | \omega_1)}{P(\vec{x} | \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}$$

Minimum Error Rate Classification

$R(a_i | \vec{x}) = \dots = 1 - p(\omega_i | \vec{x})$ $\lambda(a_i | \omega_j) = \begin{cases} 0, & i=j \\ 1, & i \neq j \end{cases}$ $i, j = 1, \dots, c$
 → select i that minimizes posterior $p(\omega_i | \vec{x})$

Decide ω_i if $p(\omega_i | \vec{x}) > p(\omega_j | \vec{x}) \quad \forall j \neq i$

Parzen windows non-parametric

choose window of volume V , count # of samples inside window

$$\rightarrow p(x) \approx \frac{k/n}{V} \quad k = \text{count}, n = \# \text{ samples}$$

→ volume too large → lose resolution
 too small → poor estimate

$$\rightarrow \text{set } V_n = \frac{1}{n^\alpha}$$

$$\lim_{n \rightarrow \infty} V_n = 0$$

$$\lim_{n \rightarrow \infty} k_n = \infty$$

$$\lim_{n \rightarrow \infty} k_n/n = 0$$

$$p_{\text{guess}}(x) \rightarrow p(x)$$

K-Nearest-Neighbors

set $k = \lceil n^{1/\alpha} \rceil$

classify x :

- (1) find k nearest-neighbors of x
- (2) determine class most freq. repr. in those k
- (3) assign x to that class

Decision Function

$$g(\vec{x}) = \sum_{i=1}^n w_i x_i + w_0$$

\vec{x} = feature vector

\vec{w} = weight vector

w_0 = threshold weight

Fisher - Linear Discriminant

- Dimensionality Reduction
- Project a set of multidim. points onto a line $y = \vec{w}^T \vec{x}$

$$\text{maximize criterion } g(x) = \frac{|\vec{m}_1 - \vec{m}_2|}{S_1 + S_2}$$

$$\vec{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y$$

$$\tilde{S}_i = \sum_{y \in Y_i} (y - \vec{m}_i)^2$$

$$\rightarrow \vec{w} = S_w^{-1} (\vec{m}_1 - \vec{m}_2), S_w = S_1 + S_2, S_i = \sum_{x \in X_i} (x - \vec{m}_i)^2$$

Unsupervised Learning

- (1) choose initial μ_1, \dots, μ_c
- (2) classify n samples to closest mean
- (3) recompute means from samples in class
- (4) Means changed? \rightarrow (2), else stop

Clustering

- (1) init. $c = n$
- (2) find nearest pair of distinct clusters x_i and x_j
- (3) merge x_i, x_j , decrement c
- (4) $c \leq c_{\text{stop}}$? stop : goto (2)

Generalisierungsfähigkeit: Fähigkeit eines Klassifizierers im Training nicht enthaltene Daten korrekt zu klassifizieren

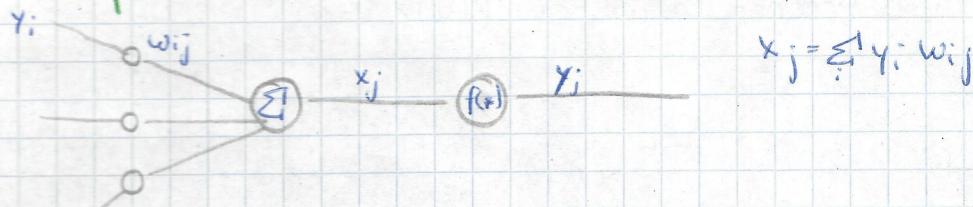
Overfitting: zu häufige Trainingsiterationen \rightarrow Klass. lernt Trainingsdaten auswendig \rightarrow versagt auf ungesuchten Daten (Schwach bei Trainingsdaten bekannt)
Abhilfe: nach jeder Tr. It. neue Daten testen, Training abbrechen wenn Klassifikation auf neuen Daten schlechter wird

Parametrisch:	
- Logistic Regression	5
- LDA	"
- Perception	5)
- Naive Bayes	5
- Simple Neural Networks	

Nicht-parametrisch	
- K-nearest neighbors	u
- Decision Trees	
- Parzen window	

Machine Learning

Perceptrons

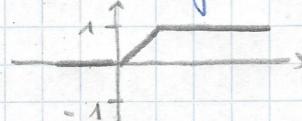


$f(*)$

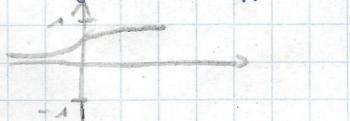
> Hard Limiter



> Threshold Logic



> Sigmoid $y_j = \frac{1}{1+e^{-x_j}}$



Learning

$$J_p(\vec{w}) = \sum_{\vec{x} \in X} (-\vec{w} \cdot \vec{x})$$

X = set of misclassified tokens

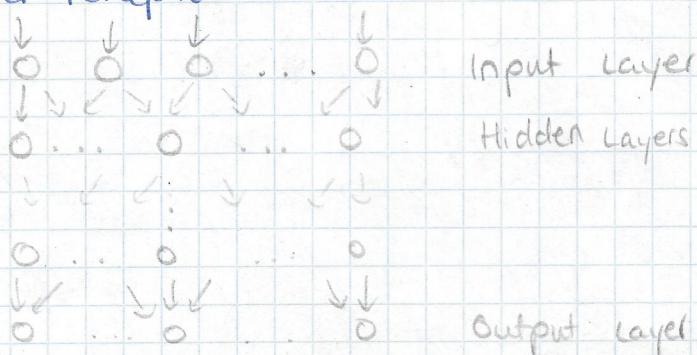
$J_p = 0 \rightarrow$ solution for \vec{w} found

$$\nabla J_p = \sum_{\vec{x} \in X} (-\vec{x})$$

$$\text{Variation: } J_p(\vec{w}) = \sum_{\vec{x} \in X} (\vec{x} \cdot \vec{w})^2$$

Neuronale Netze

Multi-Layer Perceptron



Backpropagation of Error

choose random initial weights \rightarrow apply input \rightarrow compare output to desired output
 \rightarrow compute error \rightarrow backpropagate error through net, compute $\frac{\partial E}{\partial w_{ij}}$
 (contribution of weight to error) \rightarrow adjust weights to reduce error

$$E = \frac{1}{2} \sum_j (y_j - d_j)^2 \quad y_j = \frac{1}{1+e^{-x_j}} \quad x_j = \sum_i x_i w_{ij} \quad (\text{Netz siehe oben})$$

$$(1) \quad \frac{\partial E}{\partial y_j} = y_j - d_j$$

$$(2) \quad \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_j} = \frac{\partial E}{\partial y_j} y_j [1-y_j]$$

$$(3) \quad \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = \frac{\partial E}{\partial x_j} \cdot y_i$$

$$(4) \quad \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} w_{ij}$$

$$\frac{\partial y_j}{\partial x_j} = (-1)(1 + e^{-x_j})^{-2} \cdot (-1)e^{-x_j} = \dots = y_j(1 - y_j)$$

→ Output represents a Posterior Probabilities $p(w_j | x)$

Spracherkennung

Human speech production



Formants

Resonance Frequencies of the vocal tract transfer function

Word Error Rate

$$WER = \frac{1}{N} \cdot (\# \text{ insertions} + \# \text{ deletions} + \# \text{ substitutions})$$

$N = \# \text{ Wörter in Ref.}$

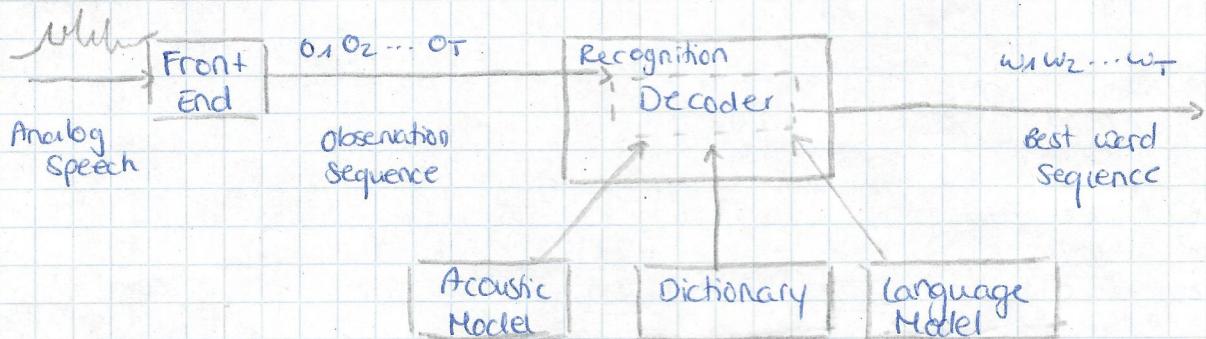
$$\Rightarrow ACC = 1 - WER \quad \text{Genauigkeit}$$

Fundamentalsformel

$$\hat{w} = \arg \max_w P(w|x) = \arg \max_w \frac{P(x|w) P(w)}{P(x)} = \arg \max_w P(x|w) P(w)$$

↓ akustisches Modell
 Hypothese Merkmalsvektoren Sprachmodell

Speech Recognizer



Finde w, w_1, w_2, \dots, w_n so dass gegeben $A = a_1, a_2, \dots, a_k$ $P(w|A)$ maximiert

$$\rightarrow \text{Bayes Regel: } P(w|A) = \frac{P(A|w) P(w)}{\text{acoustic models}} \leftarrow \text{language model}$$

Front End

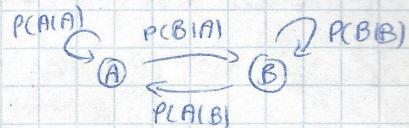
- Sampling → Spectrum contains most important info
- Reduced influence of undesired components → accuracy
- Reduce amount of data → speed
- > Analog-Digital-Wandlung
- > FFT: Spektrum der Signals
- > Anti-Aliasing-Filter

Acoustic Model

Markov Models

States $S = \{S_0, S_1, \dots, S_n\}$

Transition Probabilities $P(q_t = S_i | q_{t-1} = S_j)$



Markov Assumption: Transition probability depends only on current state

Hidden Markov Models (HMM) berechnen wkt., dass eine gegebene Wortsseq.
dass ~~sgf~~ erzeugt hat

States $S = \{S_0, S_1, \dots, S_n\}$

Transition probabilities, output probability distribution $P(y_t = o_k | q_t = S_j) = b_j(o_k)$

Evaluation

Probability of observation sequence $O = o_1 o_2 \dots o_T$ given HMM λ :

$$\begin{aligned} P(O|\lambda) &= \sum_{Q \in Q} P(O, Q|\lambda) \quad Q = q_0 q_1 \dots q_T \text{ sequence of states} \\ &= \sum_{Q \in Q} a_{q_0} q_1 b_{q_1}(o_1) a_{q_1} q_2 b_{q_2}(o_2) \dots a_{q_{T-1}} q_T b_{q_T}(o_T) \end{aligned}$$

\Rightarrow Laufzeit $O(N^T)$; $N = \# \text{ states}, T = \# \text{ observations in sequence}$

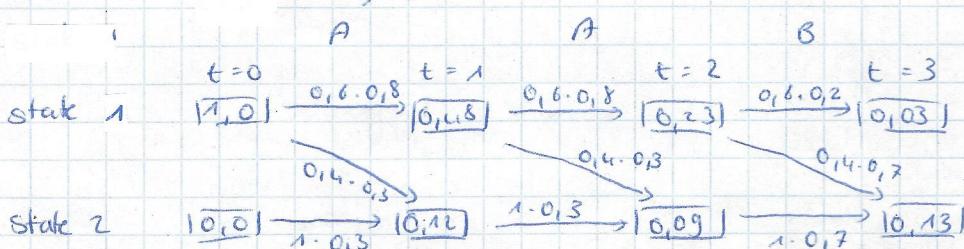
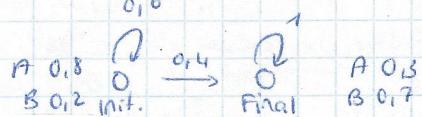
Forward Algorithm

$$P(O|\lambda) = \alpha_T(S_N)$$

$$\alpha_0(j) = \begin{cases} 1, & j \text{ is start state} \\ 0, & \text{else} \end{cases}$$

$$\alpha_t(j) = \left[\sum_{i=0}^N \alpha_{t-1}(i) a_{ij} \right] b_j(o_t) \quad O(N^2 T)$$

Bsp.:



Backward Algorithm

$$P(O|\lambda) = \beta_0(s_0)$$

$$\beta_T(i) = \begin{cases} 1, & i \text{ is end state} \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_t(i) = \sum_{j=0}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad t < T$$

Decoding

Find state sequence which maximizes probability of observation sequence

Viterbi Algorithm

$$P(O, Q|\lambda) = VP_T(S_N)$$

$$VP_t(j) = \max_{i=0, \dots, N} VP_{t-1}(i) a_{ij} b_j(O_t) \quad t > 0$$

Training HMM Parameters

Forward-Backward Algorithm

Probability of transitioning from s_i to s_j at time t given O

$$g_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O|\lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

$$\overbrace{\alpha_t(i)}^0 \quad a_{ij} b_j(O_{t+1}) \quad \overbrace{\beta_{t+1}(j)}^0$$

Baum-Welch Reestimation

$$\text{calculate } \bar{a}_{ij} = \frac{\sum_{t=1}^T g_t(i,j)}{\sum_{t=1}^T \sum_{j=0}^N g_t(i,j)}, \quad \bar{b}_j(k) =$$

$$\rightarrow \bar{\lambda} = (\bar{a}, \bar{b}), \text{ replace } \lambda \text{ with } \bar{\lambda}$$

Voronoi Diagramme Region = Menge aller Punkte der Abst. zum Zentrum geringer als zu allen anderen Zentren

→ Klassifizierte je nachdem in welche Region eine Probe fällt

Dictionaries

word Dictionaries: words and word models, assign certain number of states to each word model

Phonetic Dictionaries: Convert orthography to phoneme strings, represent alternate pronunciations, Multiwords

Tree-structured Dict.: Faster Search, Overall Run-time

Language Model liefert Infos ob geg. Sequenz von Wörtern = Satz der Sprache

Grammar Based

write grammar of possible sentence patterns
→ work overhead, Expert muss Gram. schreiben

N-Grams

- predict next word based on history
 - history approximated by $n-1$ past words

1 = unigram

$Z = \text{bigram}$

3 =ogram

⑦ welche Bewertung basierend auf Wahrscheinlichkeiten
→ keine ja/nein -Entscheidung

④ für jede neue Domäne große Trainingsdatenmenge notwendig

Minimale Editierdistanz

$$D_{\theta,0} = 0$$

$$D_{i,j} = \begin{cases} D_{i-1, j-1 + 0}, & \text{match} \\ D_{i-1, j-1 + 1}, & \text{sub} \\ D_{i-1, j} + 1, & \text{del} \\ D_{i, j-1 + 1}, & \text{ins} \end{cases}$$

→ Tabelle:	A	U	G	U	S	T
	0	1	2	3	4	5

A	1	0	1	2	3	4	5
R	2	1	2	3	4	5	6
G	3	2	3	2	3	4	5
U	4	3	2	3	2	3	4
E	5	4	3	4	3	4	5

Wissen und Planung

Logik

$L = (\text{Symbolmenge}, \text{Belegungsmaenge}, \text{Syntax}, \text{Semantik}, \text{Folgerungsoperator } \models)$

- > Symbolmenge: enthält alle Symbole über die Aussagen gemacht werden können
- > Belegung: Abbildung die jedem Symbol einen Wert aus Wertebereich D zuordnet
- > Belegungsmaenge: Menge aller möglichen Belegungen
- > Syntax: legt fest welche Sätze wohlgeform (\Rightarrow gültig) sind
- > Semantik: bestimmt Wahrheitswert eines Satzen bzgl. Belegung
- > $\alpha \models \beta : \beta$ folgt aus α
 $\alpha \models \beta \Leftrightarrow$ für alle Belegungen in den α wahr ist, ist β auch wahr

Deduktion

WB $\vdash ; \alpha$ heißt: es ex. ein Algo i der Satz α auf Wissensbasis ableiten kann

- i korrekt: leitet nur Sätze ab die aus der WB folgen
- i vollst.: leitet alle Sätze aus WB ab die aus WB folgen

Aussagenlogik

Syntax:

Satz	\rightarrow	Atom Komplex
Atom	\rightarrow	truell falsel Symbol
Symbol	\rightarrow	P Q R ...
Komplex	\rightarrow	\neg Satz Satz \wedge Satz \vee \Rightarrow \leftrightarrow

Normalformen

KNF: $(P \vee \bar{Q} \vee R) \wedge (\bar{P} \vee \bar{R}) \wedge Q$

DNF: $(P \wedge Q \wedge \bar{R}) \vee \bar{P} \vee P \wedge \bar{Q} \vee R$

Muster

Modus Ponens

$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$ aus $\alpha \Rightarrow \beta$ und α kann β abgeleitet werden

Und-Elimination

$$\frac{\alpha \wedge \beta}{\alpha}, \frac{\alpha \wedge \beta}{\beta}$$

Einheits-Resolutionsregel P_1, \dots, P_k Literale, $\emptyset \models ?P_i$

Dann:

$$\frac{(P_1 \vee \dots \vee P_k), Q}{P_1 \vee \dots \vee P_{i-1} \vee P_{i+1} \vee \dots \vee P_k}$$

Resolutionsalgo zeige $WB \models \alpha$ durch $WB \vdash \alpha$ widerlegen

- (1) WB in KNF als Klauseln schreiben
- (2) α in KNF bringen, als Klausel schreiben, zu WB hinzufügen
- (3) Resolutionsregel anwenden bis leere Klausel erzeugt wird
⇒ Klauselmenge unerfüllbar

$(WB \vdash \alpha)$ unerfüllbar $\Rightarrow WB \models \alpha$

Resolutionsregel: $(\bar{p} \vee \alpha)$ und $(p \vee \beta)$ schließt auf $(\alpha \vee \beta)$

Horn-Klauseln

Disjunktion von Literaten, von denen höchstens eins positiv

Typen:

- > Fakt, Axiom: keine negativen Literale
- > Definition: genau ein positives
- > Integritätsbeschränkung: kein positives

Und-Oder-Graph

$$A \wedge B \Rightarrow C \quad \begin{array}{c} A \\ \diagdown \\ B \end{array} \quad \text{D} \rightarrow C$$

$$\bar{P} \vee \bar{Q} \vee R \Leftrightarrow (\underbrace{P \wedge Q}_{\text{Körper}}) \Rightarrow R_{\text{Kopf}}$$

Vorwärtsverkettung

- (1) bekannte Fakten als wahr markieren
- (2) vorwärts durch Baum arbeiten
→ alle durchführbaren Folgerungentscheidungen

Rückwärtsverkettung

von Anfrage aus rückwärts durch Baum, Stoppen bei bekannten Fakten
→ Wahrheitsgehalt der Anfrage

DPLL Davis-Putnam-Logemann-Loveland

Prüfen der Erfüllbarkeit

→ Früher Abbruch: $(P \vee Q)(P \vee R)$ = wahr sobald P wahr

Einheitsklauseln: Klausel mit einem Literal

reine Symbole: Literal kommt nur negiert oder nichtnegiert in allen Klauseln vor

Algo:

- > leere Klausel → nicht erfüllbar
- > (1) Einheitsklauseln zu wahr belegen, in anderen Klauseln entsprechend ersetzen / streichen
- > (2) Variable wählen, Fälle durchgehen ($x = \text{true}, x = \text{false}$)

Prädikatenlogik Universum D ; Konstanten a, b, c und Variablen $x, y, z = \text{Werte aus } D$

Planungssprachen

STRIPS

- Zustände: Konjunktion positiver aussagenlog. Literale
- keine Variablen / Funktionen

Closed-World-Assumption

- keine Negationen im Zustand und Vorbed. (implizit falsch falls nicht vorkommend)
aber in Nachbed.)
- jeder Zustand vollst. bekannt

Aktionen

- Aktionsname, Parameter, vorbed., Effekte

ADL

- offene Welt: alle nicht angegebenen Literale = unbekannt
- negative Literale, Disjunktionen erlaubt
- Gleichheits- Prädikat

Planungsstrategien

Suche im Zustandsraum

erzeuge Baum mit allen erreichbaren Zuständen

Heuristik gezielter Suchen

- liefert optimistischen Schätzwert der Entfernung zum Ziel
- Greedy-Ansatz: weitersuchen vom Zustand aus, der laut Heur. am nächsten zum Ziel

A*-Algorithmus O=Open set, C=Closed Set

Update für Knoten v_n : pred(v_n), Kosten $g(v_n)$ um v_n zu erreichen
 $h(v_n)$ Heur. für erwartete Kosten zum Ziel

Init: $O = \{v_S\}$, $C = \emptyset$, $g(v_i) = \infty$, $g(v_S) = 0$

while $O \neq \emptyset$:

- bestimme zu erweiternden Knoten $v_i \in O$ mit $f(v_i) = g(v_i) + h(v_i)$ minimal

- $v_i = \text{Ziel} \Rightarrow \text{stop}$

- $O.$.remove(v_i)

- $C.$.add(v_i)

- Update für alle Nachfolger v_j von v_i :

- $v_j \in C$? continue

- $v_j \notin O$? $O.$.add(v_j)

- $g(v_j) = \text{cost}(v_i, v_j) < g(v_j)$?

- $g(v_j) = g(v_i) + \text{cost}(v_i, v_j)$

- $h(v_j) = \text{heuristic}(v_j, \text{Ziel}) \rightarrow$ meist eukl. Distanz

- $\text{pred}(v_j) = v_i$

Partial-Order-Planning

PO-Plan

- Aktionen, Ordnungsbed., Abhängigkeiten, offene Verbed.
- Pseudo-Aktionen Start, Ziel, Start < Ziel \in Ord. bed.

Init: Ord. bed: Start < Ziel

keine Abh. k.

offene Verbed. von Ziel

Loop: solange offene Verbed. existiert wähle Verbed. P einer Aktion B
ergänze Abh. um $A \rightarrow_p B$ \wedge Aktionen A deren Effekte P erfüllen
Bed. $A \leftarrow B$
füge entspr. in Baum ein

\rightarrow es darf kein Zyklus existieren

Planungsgraphen

Folge von Abschnitten die jeweils einem Zeitschritt im Plan entsprechen

Ablöwechselnde Schritte:

- Menge der Literale, die zu diesem Zeitpunkt wahr sein könnten
- Menge der Aktionen, deren Verbed. zu diesem Zeitpunkt erfüllt sein können
- für jedes Literal: Pseudoaktion die es unverändert erhält
- zu jedem Zeitpunkt: Info welche Paare von Literalen/Aktionen sich ausschließen
 \hookrightarrow mutex-link im Graphen

Mutex Links:

- Literale komplementär oder aus sich ausschließenden Aktionen resultiert
- Aktionen schließen sich aus, wenn
 - Effekte enthalten komplementäres Literal
 - durch Effekt der einen wird Verbed. der anderen zerstört
 - die Verbed. der einen ein Literal enthalten, das ein Literal aus den Verbed. der anderen ausschließt

Konstruktion

Erster Abschnitt:

alle Lit. des Startzustandes

Zweiter:

alle mögl. Aktionen, sich ausschließende mit mutex-link verbinden

Dritter:

alle Literale die aus Aktionen resultieren, mutex-links

Vierter

siehe 2.

:

bis keine Veränderung mehr bei den Literalen

Geometrisches Wissen und Planung

Umweltmodell

bildet reale Umwelt auf innere Repräsentation ab

Kantenmodelle

Ermittlung markanter Punkte, verbinden durch Kanten

Oberflächenmodelle

Nachbildung Objektoberfläche mit Polygonen (ebene Flächen) oder gekrümmten Flächenelementen

Volumenmodelle

Begrenzungsflächen

Körper durch umgebende geom. Elemente beschreiben

Constructive Solid Geometry

Zusammensetzung von Körpern aus parametrischen Grundelementen

Repräsentation von Hindernissen

- Freiraum und Hindernisraum
- Konfigurationsraum: Raum aus Dim. der relevanten Parameter (z.B. Gelenkwinkelraum)

Bahnplanung

Polygonzerlegung

Trenmlinie von jeder Hindernisecke nach oben und unten bis zur nächsten Hinderniskante

→ Wegenetz durch die Freiräume anlegen

Sichtgraphen finde kürzesten Weg

- Sichtverbindungen zw. Ecken der Hindernisse
- Ecken sind Knoten des Graphen
- Ränder der Hindernisse = Verbindungen

Quadratrees

Aufteilung des Raumes in Quadrate bis nur noch Quadrate die frei oder nicht frei sind

Voronoi-Diagramm max. Abstand zu den Hindernissen

teilt Raum, sodass jedes Hindernis in genau einer Fläche

Grenzlinien:

- Gerade zw. 2 punktförmigen Hindernissen
- Parabel zw. Punkt und Gerade
- Parabel im Streifen der Strecke zw. Punkt und endl. Strecke, Halbgerade außerhalb
- 2 Strecken: Gerade im Schnitt der Streifen, parabel bzgl. Endpunkt in einzelnen Streifen
Gerade bzgl. Eckpunkten außerhalb der Streifen

Grenzen = Wegenetz

Kreuzungen = Knoten des Suchgraphen

Potentialfeldmethode

- lokale Suche durch Gradientenabstieg im Potentialfeld des Konfigurationsraums
- globales Minimum im Zielpunkt
- Hindernisse: hohes Potential