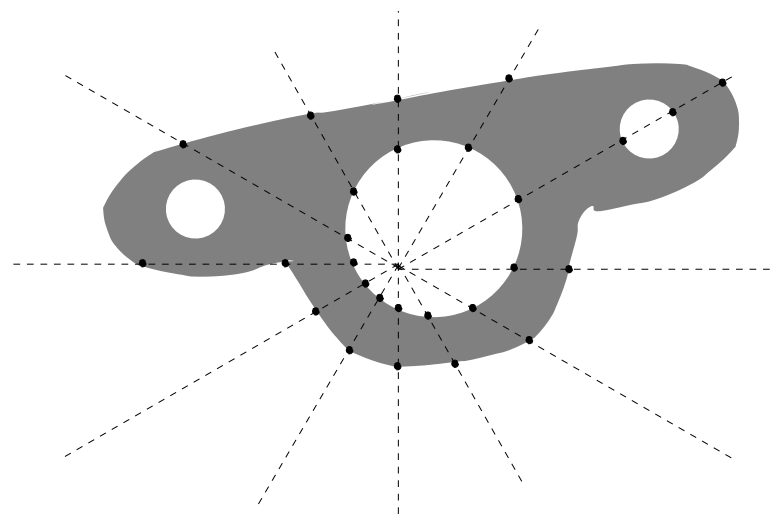
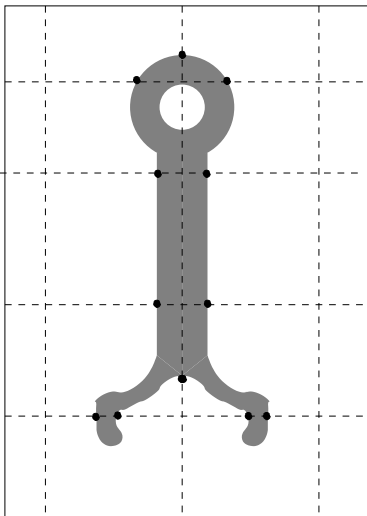


2 Intuitive Klassifikation

“intuitiv” bedeutet nicht schlecht oder primitiv, sondern bezeichnet einen leicht zu verstehenden Zugang zum Problem der Merkmalsgewinnung und Klassifikation

2.1 Einfache heuristische Merkmale

2.1.1 Schnittpunkte mit Testlinien

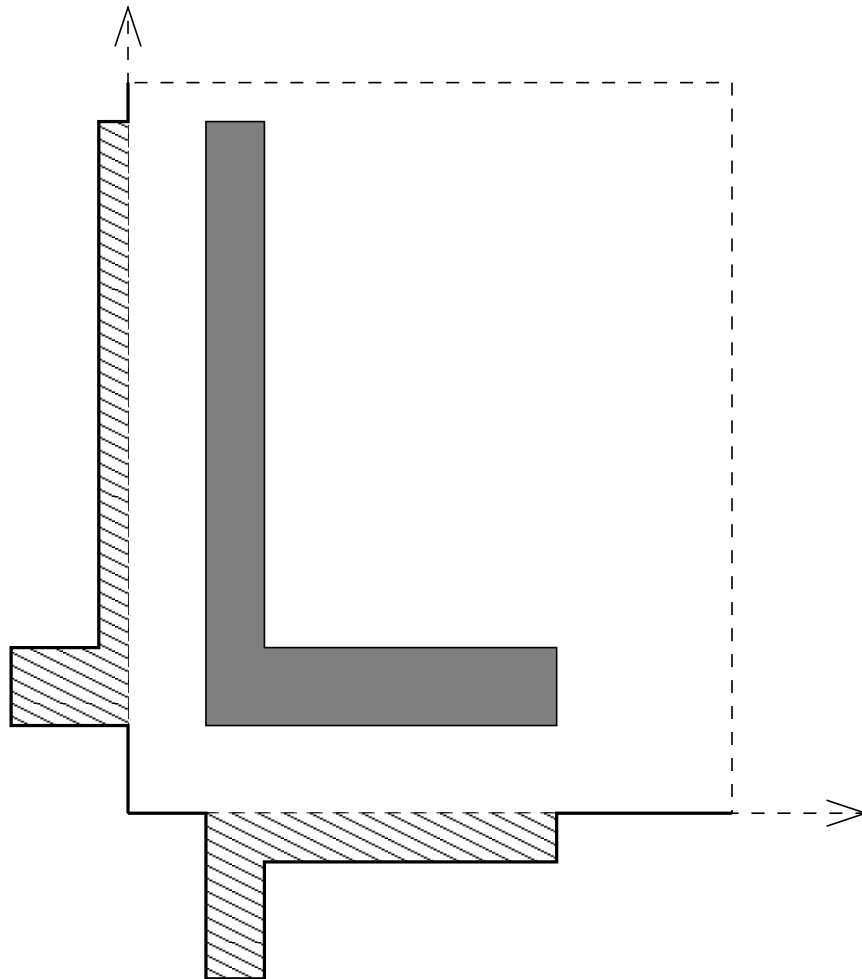


damit z.B.:

- Anzahl Schnittpunkte
- Länge der Testlinie(n) innerhalb des Objektes
- Koordinaten der Schnittpunkte

z.B. Testlinien sternförmig um den Schwerpunkt

2.1.2 Projektion auf Geraden

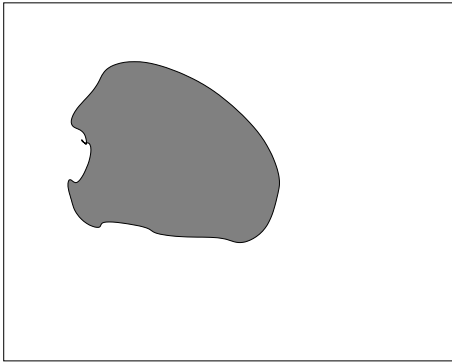


daraus z.B.:

- Minima und Maxima
- Projektionskurve
- Projektion:

$$p_j^{(x)} = \sum_{k=0}^{M_x-1} f_{jk}$$

2.1.3 Globale Parameter, Kennzahlen



z.B. Formfaktor

$$C = \frac{L^2}{A} (\text{Kompaktheit})$$

L = Länge der Umrißlinie (äußerer Umriß, keine Löcher)

A = Fläche

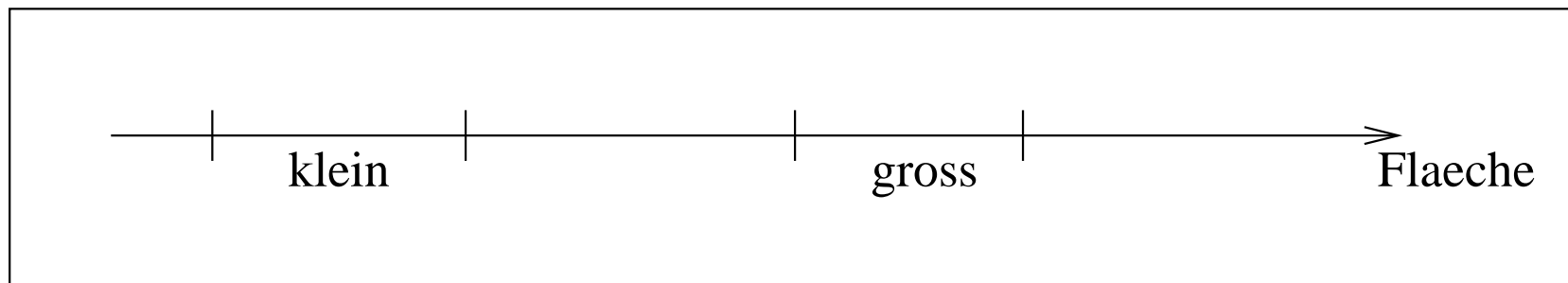
2.2 Klassifikation mit Hyperquadern

Für jede Komponente des Merkmalvektors wird je Klasse ein zulässiges Intervall vorgegeben.

Klassifikation (oder Rückweisung) kann dann einfach erfolgen.

Beispiel: Unterscheidung von großen und kleinen Kreisen

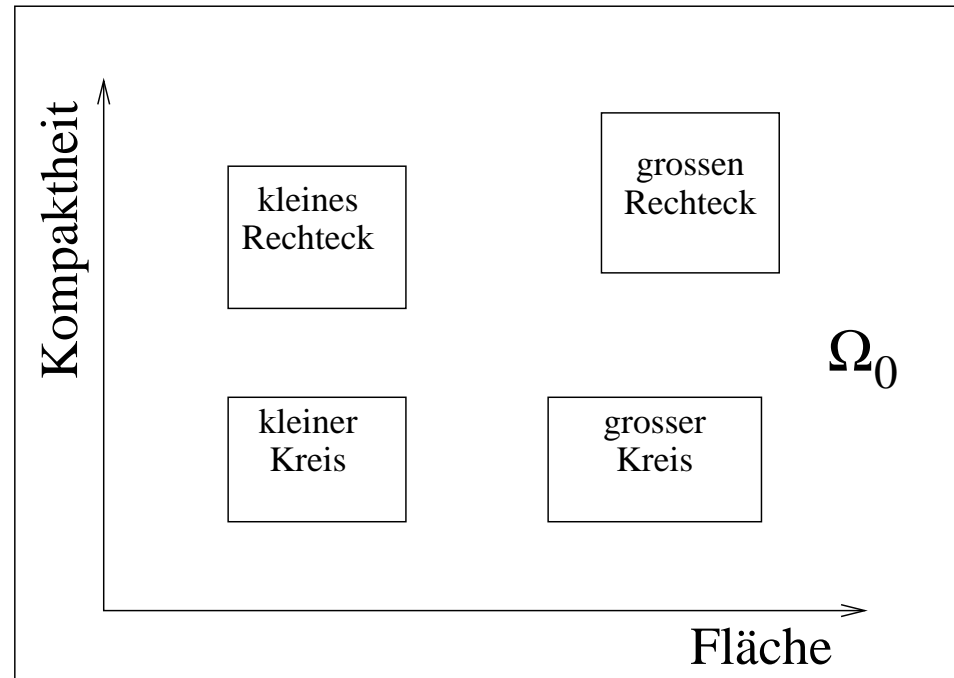
Als Merkmal: Fläche



2.3 Abstandsmessung zur Klassifikation

Beispiel: Unterscheidung von großen und kleinen Kreisen und Rechtecken

Merkmale: Fläche und Kompaktheit



Hyperquader-Klassifikatoren lassen sich sehr einfach realisieren, aber sind in ihren Möglichkeiten zur Trennung des Merkmalsraums sehr eingeschränkt.

2.3 Abstandsmessung zur Klassifikation

quantisiere die Ähnlichkeit von Mustern mit einer Metrik. z.B.:

$$d^{(r)}(\vec{c}_i, \vec{c}_j) = \sqrt[r]{\sum_{\nu} |c_{i\nu} - c_{j\nu}|^r}$$

$$c_i, c_j \in \omega$$

$c_{i\nu}, c_{j\nu} \in \omega$ sind die Komponenten der Merkmalsvektoren

- $r = 1$ City-Block Metrik
- $r = 2$ euklidischer Abstand
- $r = \infty$ Maximum Norm

Eigenschaften einer Metrik

1. $d(\vec{x}, \vec{y}) = d(\vec{y}, \vec{x})$
2. $d(\vec{x}, \vec{y}) \geq 0$
3. $d(\vec{x}, \vec{y}) = 0 \Rightarrow \vec{x} = \vec{y}$
4. $d(\vec{x}, \vec{y}) + d(\vec{y}, \vec{z}) \leq d(\vec{x}, \vec{z})$

2.3.1 Nächster-Nachbar-Klassifikator

Bei gegebener klassifizierter Stichprobe ($\omega = \{\vec{c}_j | j = 1, \dots, N\}$)

klassifizieren wir eine (neues) Muster in die Klasse des nächsten Musters innerhalb dieser Stichprobe:

\vec{c} ordnen wir Ω_κ zu, falls

$$i = \operatorname{argmin}_{j=1, \dots, N} \{d(\vec{c}, \vec{c}_j)\} \text{ mit } \vec{c}_i \in \Omega_\kappa$$

Bemerkung:

- gibt gute Klassifikationsergebnis (was immer das heißen mag an dieser Stelle), wenn die Stichprobe (sehr) groß ist.
- Voronoi-Tesselierung des Merkmalsraums (keine Rückweisung)
- einfach zu Lernen, großer Speicher- und Rechenaufwand für die Klassifikation (es gibt Methoden zum Ausdünnen der Stichprobe)

mNN-Klassifikator

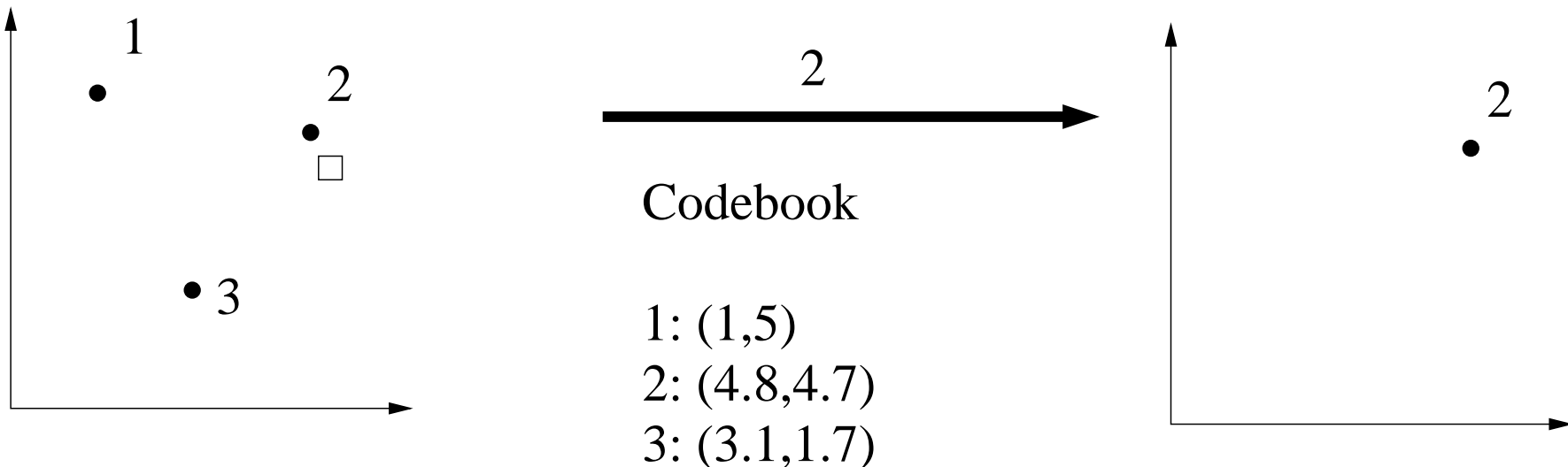
Modifikation, um Ausreißer oder Störungen an der Klassengrenzen besser zu Berücksichtigen:

- Entscheidung gemäß der Majorität der m nächsten Nachbarn
- Rückweisung, falls das Maximum von Vektoren aus einer Klasse kleiner als Schwelle $l < m$

Auch hier wird i.d.R. in Gebieten des Merkmalsraumes klassifiziert, die nicht in der Stichprobe/Problemkreis enthalten sind.

2.3.2 Klassifikation durch Vektorquantisierung

- Idee ist hier, Merkmalsvektoren im Zentrum der Klassengebiete zu finden.
- Einfachste Variante: bestimme den Schwerpunkt jeder Klasse.
(Das gibt nur sehr einfache, insbesondere zusammenhängende Klassengebiete.)
- Vektorquantisierung dient (eigentlich) der kompakten Repräsentation oder Codierung von (hochdimensionalen) Vektoren durch Repräsentanten, den Codebook-Vektoren.



2.3.3 k-Means

Teile jede Klasse Ω_{κ} in L_{κ} (Teil-)Gebiete, die jeweils durch ihren Schwerpunkt oder Mittelpunktvektor $\vec{\mu}_{\kappa}^l$ repräsentiert werden.

Im restlichen Abschnitt lassen wir die Klassenindizes κ zur Vereinfachung weg:

- sei $\Omega = \{\vec{c}_i | i = 1, \dots, N\}$ die (einzige) Klasse
- sie wird in L disjunkte Teilgebiete R^l zerlegt:
$$\Omega = R^1 \dot{\cup} R^2 \dot{\cup} \dots \dot{\cup} R^L$$
- jedes Teilgebiet R^l wird durch seinen Schwerpunkt repräsentiert:

$$\vec{\mu}^l = \frac{1}{|R^l|} \sum_{\vec{c}_i \in R^l} \vec{c}_i$$

- Ziel ist Minimierung des Quantisierungsfehlers

$$\epsilon = \frac{1}{N} \sum_{l=1}^L \sum_{\vec{c}_i \in R^l} d(\vec{c}_i, \vec{\mu}^l)$$

2.3 Abstandsmessung zur Klassifikation

wähle aufgrund von Vorwissen oder zufällig (z.B. die ersten L Vektoren der Stichprobe mit der Größe N) initiale Mittelpunkte $\vec{\mu}^l, l = 1, \dots, L$

$\epsilon^0 := \infty$; der Quantisierungsfehler in der Iteration 0

$t = 0$; Iterationszähler

$t := t + 1, \quad \epsilon^{(t)} := 0$

FOR alle Gebiete $l = 1, \dots, L$

$N^l := 0; \quad \hat{\vec{\mu}}^l := \vec{0}$

FOR alle Vektoren \vec{c}_i der Stichprobe

bestimme $\vec{\mu}^l$ mit minimalem Abstand zu \vec{c}_i

$\epsilon^{(t)} := \epsilon^{(t)} + d(\vec{c}_i, \vec{\mu}^l)$

berechne neuen Schätzwert für den Mittelpunkt $\hat{\vec{\mu}}^l := \vec{\mu}^l + \vec{c}_i$

$N^l := N^l + 1$

$\epsilon^{(t)} := \epsilon^{(t)} / N$

FOR alle Gebiete $l = 1, \dots, L$

$\vec{\mu}^l := \hat{\vec{\mu}}^l / N^l$

UNTIL $(\epsilon^{(t-1)} - \epsilon^{(t)}) / \epsilon^{(t)} \leq \epsilon$

2.3 Abstandsmessung zur Klassifikation

- k-means konvergiert “fast immer”
- es gibt keine Garantie in eine (lokales) Minimum des Quantisierungsfehlers zu kommen
starte den -means mit unterschiedlichen Initialisierungen und nimm bestes Ergebnis
wir werden später noch eine Variante kennenlernen, die garantiert zu einem lokalen Minimum konvergiert
- es wird (praktisch immer) der eulidische Abstand verwendet
- neben der Vektorquantisierung wird das k-means eine Verfahren oft zum Clustern benutzt:
eine gegebene Menge von Datenpunkten soll in (eine vorgegebene Anzahl von) Cluster/Häufungsgebiete aufgeteilt werden

2.3.4 Learning-Vector-Quantization

betrachte beim Lernen die gesamte Stichprobe (nicht Klassen unabhängig)

LVQ1-Algorithmus

wähle auf der Basis einer klassifizierten Stichprobe L Vektoren $\vec{\mu}^l$ als initiale Mittelpunkte, d.h. für jeden Vektor $\vec{\mu}_l$ ist die zugehörige Klasse bekannt (z.B. berechne je Klasse Ω_κ L/K Mittelpunkte $\vec{\mu}^l$ mit Hilfe des k-means-Algorithmus aus Abschnitt 2.3.3)

Iterationszähler $t = 0$

wähle zufällig \vec{c}_i aus der Stichprobe aus

bestimme $\vec{\mu}^l$ mit kleinstem Abstand zu \vec{c}_i

IF $\vec{\mu}^l$ und \vec{c}_i sind aus derselben Klasse

THEN $\vec{\mu}^l := \vec{\mu}^l + \alpha(t)[\vec{c}_i - \vec{\mu}^l]$

ELSE $\vec{\mu}^l := \vec{\mu}^l - \alpha(t)[\vec{c}_i - \vec{\mu}^l]$

$t := t + 1$

UNTIL $t > \beta * L$

LVQ2-Algorithmus

initialisiere L Vektoren $\vec{\mu}_l$ (analog zu LVQ1)	
Iterationszähler $t = 0$	
wähle zufällig \vec{c}_i aus der Stichprobe aus	
bestimme den nächsten Nachbarn $\vec{\mu}_l$ und “übernächsten” $\vec{\mu}_k$	
IF	$\vec{\mu}_l$ und $\vec{\mu}_k$ gehören zu unterschiedlichen Klassen UND \vec{c}_i gehört in Klasse von $\vec{\mu}_l$ oder $\vec{\mu}_k$ UND $\min \left(\frac{d(\vec{c}_i, \vec{\mu}_l)}{d(\vec{c}_i, \vec{\mu}_k)}, \frac{d(\vec{c}_i, \vec{\mu}_k)}{d(\vec{c}_i, \vec{\mu}_l)} \right) > \frac{1-w}{1+w}$ mit $0.2 < w < 0.3$
THEN	$\vec{\mu}_l := \vec{\mu}_l + \alpha(t)[\vec{c}_i - \vec{\mu}_l]$ $\vec{\mu}_k := \vec{\mu}_k - \alpha(t)[\vec{c}_i - \vec{\mu}_k]$
$t := t + 1$	
UNTIL $t > \beta * L$	

- nur die “Grenzfläche” zwischen den Vektoren $\vec{\mu}_l$ und $\vec{\mu}_k$ wird modifiziert
- w bestimmt die Breite dieser “Grenzfläche”

LVQ3-Algorithmus

initialisiere L Vektoren $\vec{\mu}_l$ (analog zu LVQ1)	
Iterationszähler $t = 0$	
wähle zufällig \vec{c}_i aus der Stichprobe aus	
bestimme die beiden nächsten Nachbarn $\vec{\mu}_l$ und $\vec{\mu}_k$	
IF	$\vec{\mu}_l$ und $\vec{\mu}_k$ gehören zu unterschiedlichen Klassen UND \vec{c}_i gehört in Klasse von $\vec{\mu}_l$ oder $\vec{\mu}_k$ UND $\min \left(\frac{d(\vec{c}_i, \vec{\mu}_l)}{d(\vec{c}_i, \vec{\mu}_k)}, \frac{d(\vec{c}_i, \vec{\mu}_k)}{d(\vec{c}_i, \vec{\mu}_l)} \right) > \frac{1-w}{1+w} \quad \text{mit} \quad 0.2 < w < 0.3$
THEN	$\vec{\mu}_l := \vec{\mu}_l + \alpha(t) [\vec{c}_i - \vec{\mu}_l]$ $\vec{\mu}_k := \vec{\mu}_k - \alpha(t) [\vec{c}_i - \vec{\mu}_k]$
IF	$\vec{\mu}_l$ und $\vec{\mu}_k$ gehören zur richtigen Klasse
THEN	$\vec{\mu}_l := \vec{\mu}_l + \varepsilon \alpha(t) [\vec{c}_i - \vec{\mu}_l]$ $\vec{\mu}_k := \vec{\mu}_k + \varepsilon \alpha(t) [\vec{c}_i - \vec{\mu}_k]$
$t := t + 1$	
UNTIL $t > \beta * L$	

- auch wenn beide Vektoren in der richtigen Klasse wird modifiziert
- mit $0.1 < \varepsilon < 0.3$

2.3.5 Bemerkungen

1. die Wahl guter Merkmale ist schwierig (Postulat 2)
2. gestörte Muster
3. optimale Klassifikatoren
4. bei der (deterministischen) Klassifikation wird jeweils der Merkmalsraum in Klassengebiete (ev. inklusive Rückweisungsklasse) partitioniert.
hier finden wir die Partitionierung des Problemkreises wieder, wobei aber i.A. der Problemkreis nur einen Teil des vollständigen Merkmalsraumes abdeckt.