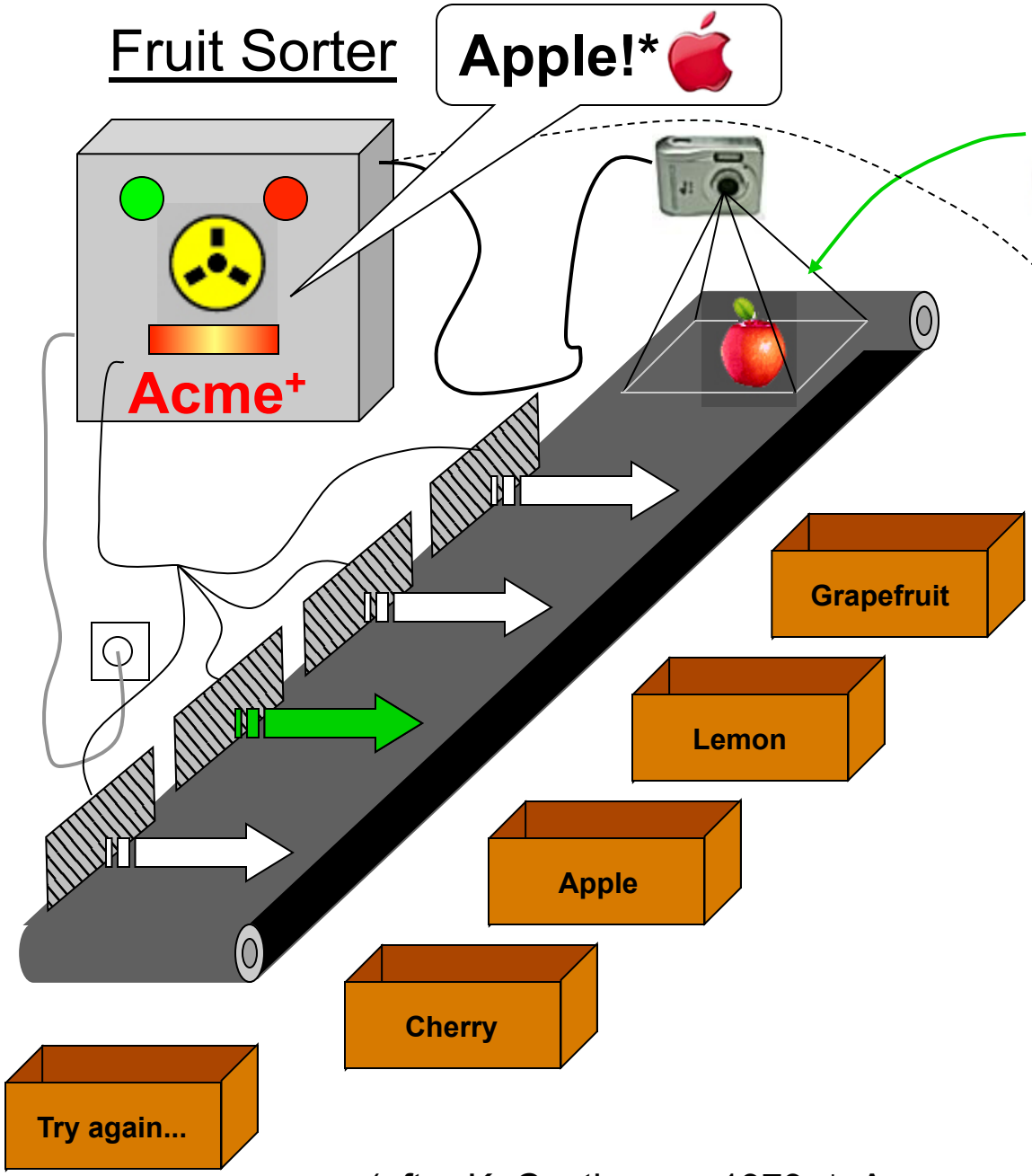
C: Historische Wurzeln der KNN

- **Mustererkennung und Klassifikation**
- **Lineare Diskriminantenfunktion**
- **Merkmal- und Gewichtsraum**

- **Perzeptron-Ansatz des Lernens**
- **Perzeptron-Lernen und Gradientenabstiegsverfahren**
- **Methode der Potentialfunktionen**

Fruit Sorter



- 4 classes of fruit -

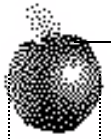
(from orchards)

- **image**

segmentation

- **binary image**

feature extraction



80% „redness“

6 cm diameter

feature vector

F=(80, 6)

DECISION*

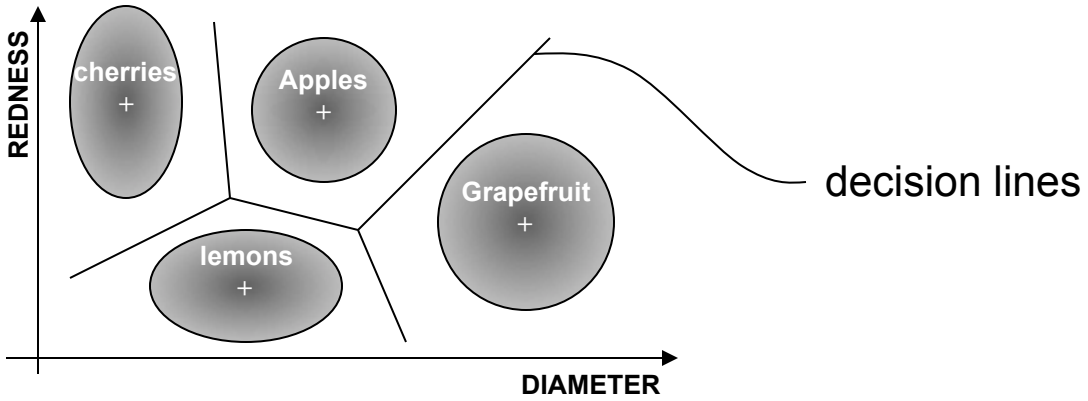
(after K. Castleman, 1979; +: Acme Corp., see en.wikipedia.org)

Fruit Sorter: Prinzipien der Klassifikation

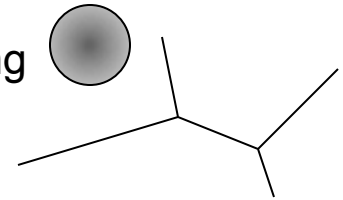
- Klassifikation

  Prinzip: „Ähnlichkeit“ zwischen - Merkmalen eines Objekts
  - Klassen–Charakteristika (aus der Stichprobe)

- Klassifikations-“Regeln“:



- 2-dimensionaler Merkmalsraum
  - Merkmale: • „Röte“
    • Durchmesser
- Stichprobe
  - Klassenvarianz, z.B. Normalverteilung
  - Klassentrennung („Diskriminanten“)
  - Klassen–“Prototypen“ +



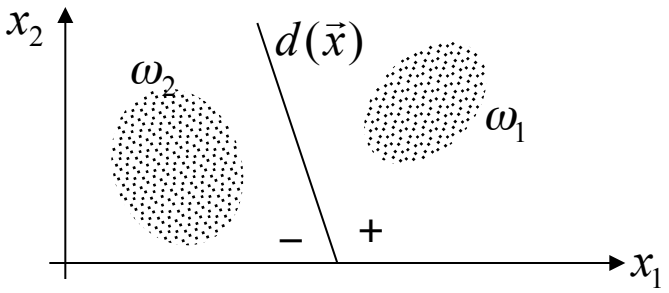
- Anwendung: Mustererkennung („pattern recognition“)

Diskriminantenfunktion und Entscheidungs-Hyperflächen

(Quelle: Tou & Gonzalez, 1974, Kap. 2)

- **Lineare Diskriminantenfunktionen**

$$d(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 = 0$$



  - Entscheidungsregel $\vec{x} \propto \omega_i$

$$\vec{x} \propto \omega_1, \text{ falls } d(\vec{x}) > 0$$
$$\vec{x} \propto \omega_2, \text{ falls } d(\vec{x}) < 0$$

  unbestimmt für $d(\vec{x}) = 0$!

  - offene Probleme: - u. U. nichtlineare Form von $d(\vec{x})$

$$\Updownarrow$$

  - geometrische/topologische Eigenschaften von ω_i
  - Bestimmung der Koeffizienten w_i (aus der Klassenstichprobe)

- **Generalisierung**: n-dimensionaler Fall

$$d(\vec{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1}$$
$$= \vec{w}_0{}^T \vec{x} + w_{n+1}$$

mit $\vec{w}_0 = [w_1, \ldots, w_n]^T$ als Gewichts-/Parametervektor

oder $d(\vec{x}) = \vec{w}^T \vec{x}$

mit $\vec{w} = [w_1, \ldots, w_{n+1}]^T$

$\vec{x} = [x_1, \ldots, x_n, 1]^T$

„augmented“ = erweiterte Vektor-Notation

Für den 2-Klassen-Fall gilt:

$$d(\vec{x}) = \vec{w}^T \vec{x} = \begin{cases} > 0, \text{ falls } \vec{x} \propto \omega_1 \\ < 0, \text{ falls } \vec{x} \propto \omega_2 \end{cases}$$

- **Multi-Klassen-Fälle** (M Klassen)
  - Fall 1: Jede Klasse ist von den anderen Klassen durch eine Trenn-Hyperfläche separierbar. Es existieren daher M Funktionen

$$\boxed{d_i(\vec{x}) = \vec{w}_i^{\,T} \vec{x} = \begin{cases} > 0, \text{ falls } \vec{x} \propto \omega_i \\ < 0, \text{ sonst} \end{cases}} \quad \text{mit } \vec{w}_i = [w_{i1}, \ldots, w_{i,n}, w_{i,n+1}]^T$$

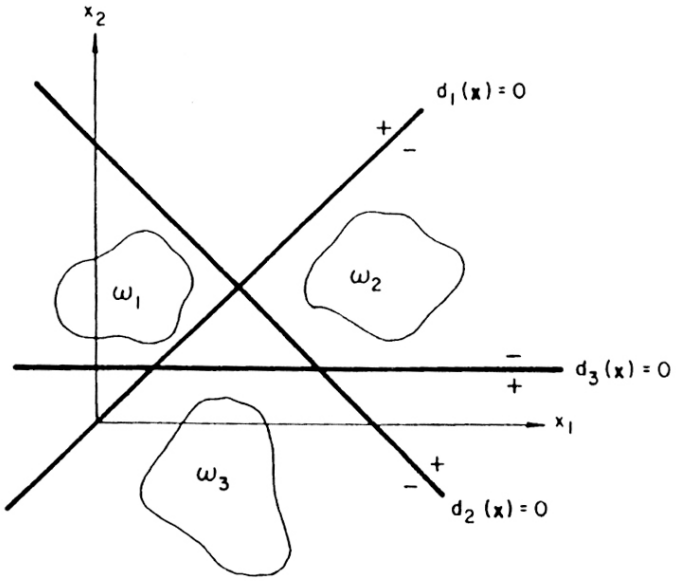
Beispiel: Fig. 2.2 ($M = 3$)

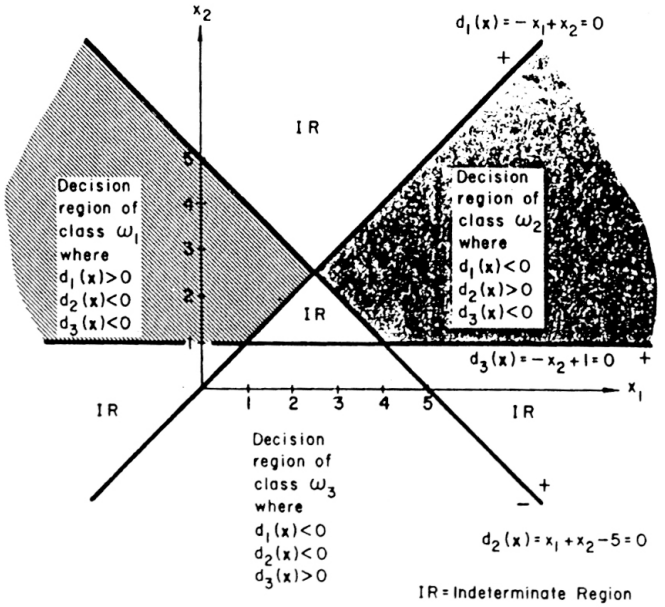
$d_1(\vec{x}) = -x_1 + x_2 = 0$

$d_2(\vec{x}) = x_1 + x_2 - 5 = 0$

$d_3(\vec{x}) = -x_2 + 1 = 0$

z.B. $\vec{x} \propto \omega_1$, falls $d_1(\vec{x}) > 0 \wedge d_2(\vec{x}) < 0 \wedge d_3(\vec{x}) < 0$





(**Figure 2.2** aus Tou & Gonzalez, 1974)

z.B. $\vec{x} = (6,5)^T$,

$$\left.\begin{aligned} d_1(\vec{x}) &= -1 \\ d_2(\vec{x}) &= 6 \\ d_3(\vec{x}) &= -4 \end{aligned}\right\} \Rightarrow \vec{x} \propto \omega_2, \text{ da } d_2(\vec{x}) > 0$$

- Fall 2: Die Klassen sind nur <u>paarweise</u> separierbar, d.h. es existieren $\frac{M(M-1)}{2}$ Trennhyperflächen

$$d_{ij}(\vec{x}) = \vec{w}_{ij}^{\,T}\vec{x},$$

wobei $d_{ij}(\vec{x}) = -d_{ji}(\vec{x})$

$$\boxed{\vec{x} \propto \omega_i, \text{wenn } d_{ij} > 0, \forall\ j \neq i}$$

Beispiel: Fig. 2.3 (mit $d_{ij}(\vec{x}) = -d_{ji}(\vec{x})$)

$d_{12}(\vec{x}) = -x_1 - x_2 + 5 \Leftrightarrow d_{21}(\vec{x}) = x_1 + x_2 - 5$

$d_{13}(\vec{x}) = -x_1 + 3$ etc.

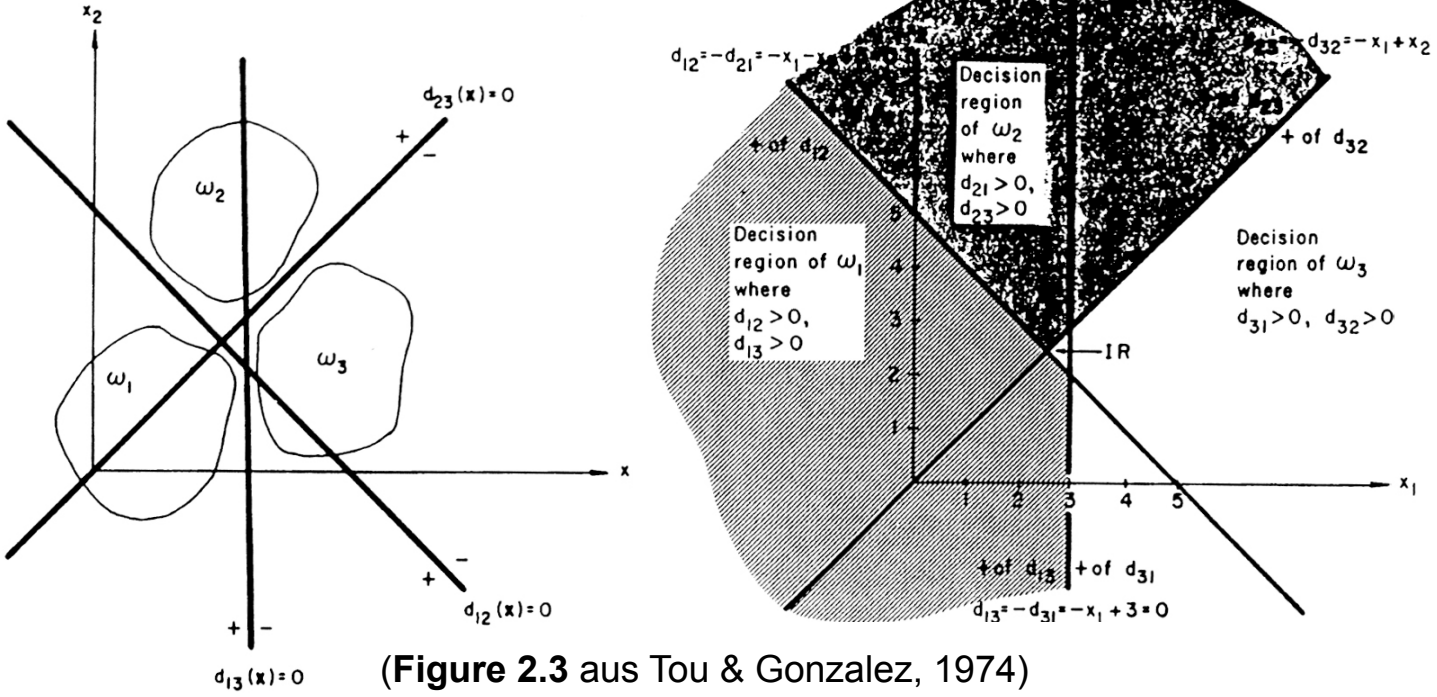
$d_{23}(\vec{x}) = -x_1 + x_2$ etc.

z.B. $\vec{x} \propto \omega_1$, wenn $d_{12}(\vec{x}) > 0$ etc. $d_{13}(\vec{x}) > 0$

z.B. $\vec{x} = (4,3)^T, d_{12}(\vec{x}) = -2 \quad d_{21}(\vec{x}) = 2$

$d_{13}(\vec{x}) = -1 \Leftrightarrow \boxed{d_{31}(\vec{x}) = 1}$ $\Rightarrow \vec{x} \propto \omega_3$, da $d_{3j}(\vec{x}) > 0$ für $\forall\ j \neq 3$

$d_{23}(\vec{x}) = -1 \quad \boxed{d_{32}(\vec{x}) = 1}$



(**Figure 2.3** aus Tou & Gonzalez, 1974)

• Fall 3: Es existieren M Trennhyperflächen

$$d_k(\vec{x}) = \vec{w}_k{}^T \vec{x}, \quad k = 1, 2, \ldots, M$$

so daß $\boxed{\vec{x} \propto \omega_i, \text{ wenn } d_i(\vec{x}) > d_j(\vec{x}),}$ für $\forall\ j \neq i$

Hinweis: Spezialfall aus Fall 2, da

$$\begin{aligned} d_{ij}(\vec{x}) &= d_i(\vec{x}) - d_j(\vec{x}) \\ &= (\vec{w}_i - \vec{w}_j)^T \vec{x} \\ &= \vec{w}_{ij}{}^T \vec{x} \end{aligned}$$

somit : $d_i(\vec{x}) > d_j(\vec{x}) \Rightarrow d_{ij}(\vec{x}) > 0$, für $\forall\ j \neq i$

Beispiel: Fig. 2.4 (für $M = 3$)

$$\vec{x} \propto \omega_1 \Leftrightarrow d_1(\vec{x}) > d_2(\vec{x}) \wedge d_1(\vec{x}) > d_3(\vec{x})$$

$$\Updownarrow$$

$$d_1(\vec{x}) - d_2(\vec{x}) > 0 \wedge d_1(\vec{x}) - d_3(\vec{x}) > 0$$

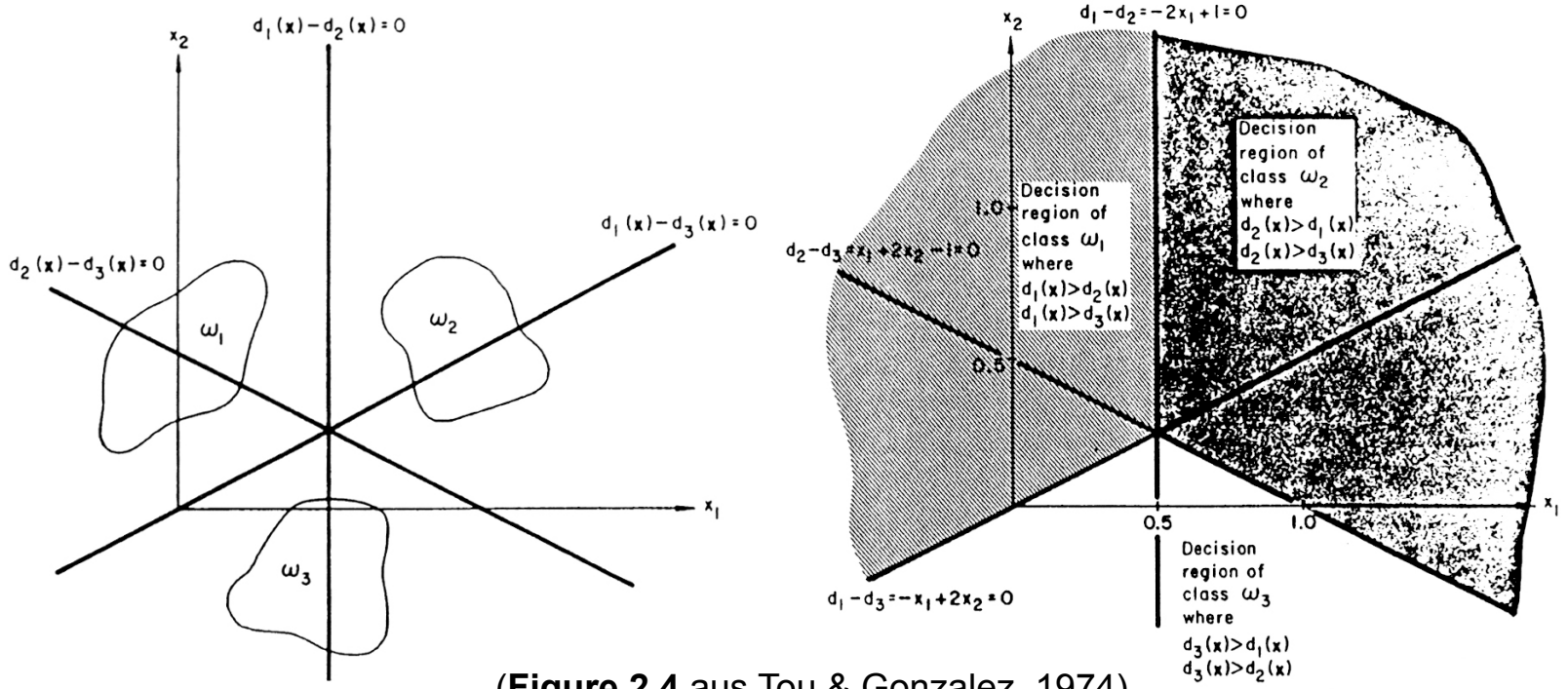
z.B. $\vec{x} = (1, 1)^T$

$$\begin{aligned} d_1(\vec{x}) &= -x_1 + x_2 && = 0 \\ d_2(\vec{x}) &= x_1 + x_2 - 1 && = 1 \\ d_3(\vec{x}) &= -x_2 && = -1 \end{aligned}$$

$$\left.\begin{array}{ll} d_{12}(\vec{x}) = -1 & \boxed{d_{21}(\vec{x}) = 1} \\ \Rightarrow d_{13}(\vec{x}) = 1 & \Leftrightarrow d_{31}(\vec{x}) = -1 \\ \boxed{d_{23}(\vec{x}) = 2} & d_{32}(\vec{x}) = -2 \end{array}\right\} \Rightarrow \vec{x} \propto \omega_2, \text{ da } d_{2j}(\vec{x}) > 0 \text{ für } \forall\ j \neq 2$$

! d.h. wenn die Klassifikation durch einen der o.g. Fälle von linearen Funktionen möglich ist, dann sind die Klassen *linear separierbar*!

jedoch: grundlegendes Problem ist die Bestimmung der Koeffizienten w_i



(**Figure 2.4** aus Tou & Gonzalez, 1974)

- **Generalisierung**

$$d(\vec{x}) = w_1 f_1(\vec{x}) + w_2 f_2(\vec{x}) + \ldots + w_k f_k(\vec{x}) + w_{k+1} = 0$$

$$= \sum_{i=1}^{k+1} w_i f_i(\vec{x})$$

mit $f_i(\vec{x})$: reelle Fkt., $i = 1, 2, \ldots, k$

$f_{k+1}(\vec{x}) = 1$

$k+1$: Anzahl der Terme in der Entwicklung

d.h. $d(\vec{x})$ abhängig von der Wahl von $f_i(\vec{x})$ und von $k+1$.

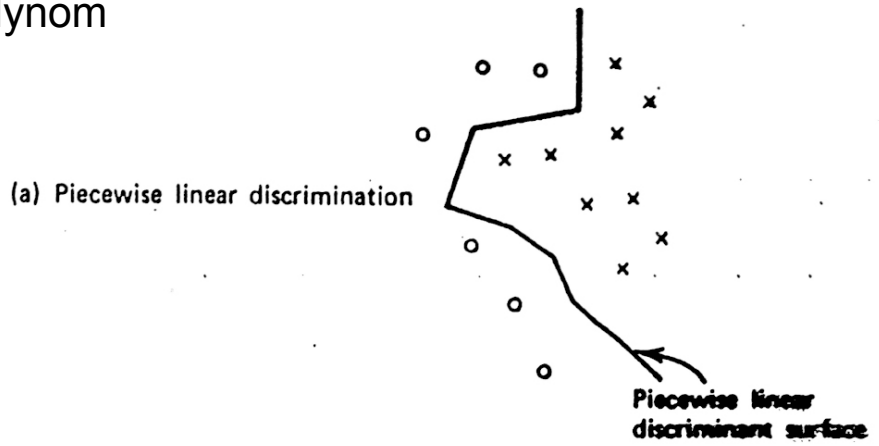
Beispiel: $f_i(\vec{x})$ ist ein lineares Polynom

d.h. für $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ ist

$$f_i(\vec{x}) = x_i, \qquad k = n$$

$$\Rightarrow d(\vec{x}) = \vec{w}^T \vec{x} + w_{n+1}$$

(lineares Polynom)



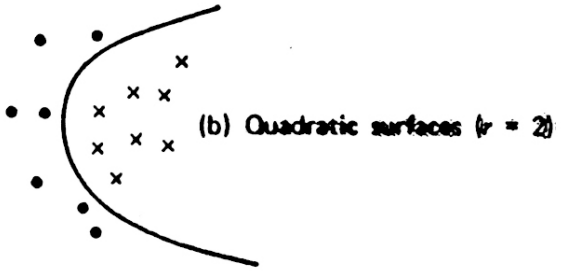
Beispiel: $f_i(\vec{x})$ ist eine quadratische Funktion

z.B. für $\vec{x} = [x_1, x_2]^T$ ist

$$d(\vec{x}) = w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 + w_1x_1 + w_2x_2 + w_3$$

z.B. für $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ ist

$$d(\vec{x}) = \sum_{j=1}^{n} w_{jj}x_j^2 + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} w_{ij}x_jx_k + \sum_{j=1}^{n} w_jx_j + w_{n+1}$$



$$\Rightarrow d(\vec{x}) = \sum_{i=1}^{k+1} w_i f_i(\vec{x}), \text{ mit } f_i(\vec{x}) = x_p^s x_q^t \quad p, q : 1, 2, \ldots, n$$
$$s, t : 0, 1$$

Beispiel: $f_i(\vec{x})$ ist ein Polynom r-ter Ordnung

$$f_i(\vec{x}) = x_{p_1}^{s_1} \cdot x_{p_2}^{s_2} \cdot \ldots \cdot x_{p_r}^{s_r}, \text{ mit } p_1, \ldots, p_r = 1, 2, \ldots, n$$
$$s_1, \ldots, s_r = 0, 1$$

<u>rekursive Form</u>:

$$d^r(\vec{x}) = \left(\sum_{p_1=1}^{n} \sum_{p_2=p_1}^{n} \ldots \sum_{p_r=p_{r-1}}^{n} w_{p_1p_2\ldots p_r} x_{p_1} x_{p_2} \ldots x_{p_r} \right) + d^{r-1}(\vec{x}) \quad \text{mit } d^0(\vec{x}) = w_{n+1}$$

Anzahl der Koeffizienten für Grad r im n-dim Fall:

$$N_w = \frac{(n+r)!}{n! \cdot r!} = \binom{n+r}{r}$$

FIGURE 3.5. Nonlinear separating surfaces.

z.B. für $r = 2, n = 2$

$$d^2(\vec{x}) = \left(\sum_{p_1=1}^{2} \sum_{p_2=p_1}^{2} w_{p_1 p_2} x_{p_1} x_{p_2} \right) + d^1(\vec{x})$$

wobei $d^1(\vec{x}) = \left(\sum_{p_1=1}^{2} w_{p_1} x_{p_1} \right) + d^0(\vec{x})$

$$= w_1 x_1 + w_2 x_2 + w_3$$

damit wird $d^2(\vec{x}) = w_{11} x_1^2 + w_{12} x_1 x_2 + w_{22} x_2^2 + w_1 x_1 + w_2 x_2 + w_3$

(quadratische Funktion, s.o.)

- **Merkmal- und Gewichtsraum**

Annahme: Klassen ω_1 und ω_2 sind linear separierbar

$\{\vec{x}_1^1, \vec{x}_2^1\}$ und $\{\vec{x}_1^2, \vec{x}_2^2\}$ $\quad x_i^{1,2}$: für ω_1, ω_2

↓ ↓ ↓ ↓

(1) (2) (3) (4) (in **Fig. 2.5**)

Regel: $\vec{x} \propto \omega_i$, wenn $d_{ij}(\vec{x}) > 0, \forall\ j \neq i$

gesucht: Vektor $\vec{w} = (w_1, w_2, w_3)^T$, so dass $w_1 x_{11}^1 + w_2 x_{12}^1 + w_3 > 0$

$$w_1 x_{21}^1 + w_2 x_{22}^1 + w_3 > 0$$

$$\mp w_1 x_{11}^2 \mp w_2 x_{12}^2 \mp w_3 \gtrless 0$$

$$\mp w_1 x_{21}^2 \mp w_2 x_{22}^2 \mp w_3 \gtrless 0$$

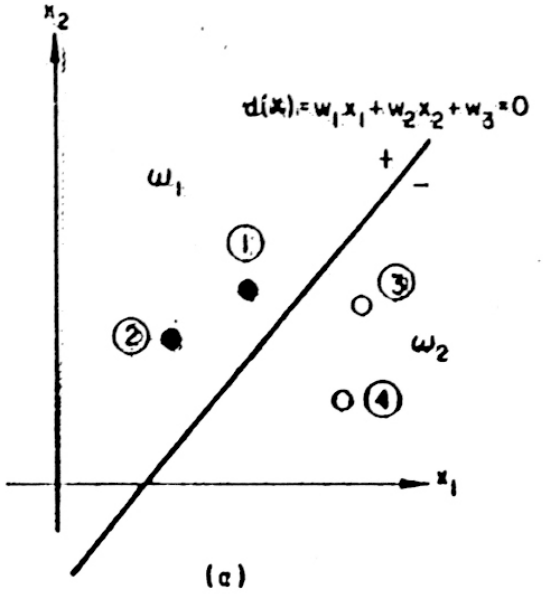
(I) und (II)

$\cdot(-1)$ für ω_2

d.h. $\vec{w}$ ist Lösung des linearen Ungleichungssystems (mit 4 Ungleichungen), so dass lineare Trennfunktion zwischen ω_1 und ω_2 resultiert.

Beispiel: Fig 2.5 aus Tou/Gonzalez

- n-dim. Merkmalraum mit Koord. x_1 und x_2
- $(n+1)$-dim. Gewichtsraum mit Koord. w_1, w_2, und w_3 z.B. mit Trenn-Hyperebene $w_1 x_{11}^1 + w_2 x_{12}^1 + w_3 = 0$, für (1)
- Ungleichung = positive/negative „Seite“ einer Hyperebene durch den Ursprung. Mit „Seite“ ist im 2-dim. Fall eine Halbebene gemeint (allg.: Halbhyperebene)



- Lösung für Ungleichungssystem (I): beliebiger Vektor $\vec{w}$, der auf positiver „Seite“ aller Ebenen für ω_1 und auf negativer „Seite“ aller Ebenen für ω_2 (d.h. jeweils aller Ebenen, die durch die Repräsentanten der jeweiligen Klasse bestimmt sind!) bzw. Lösung für Ungleichungssystem (II): $\vec{w}$ nur auf positiver „Seite“ aller Hyperebenen (wg. Multiplikation mit (-1))

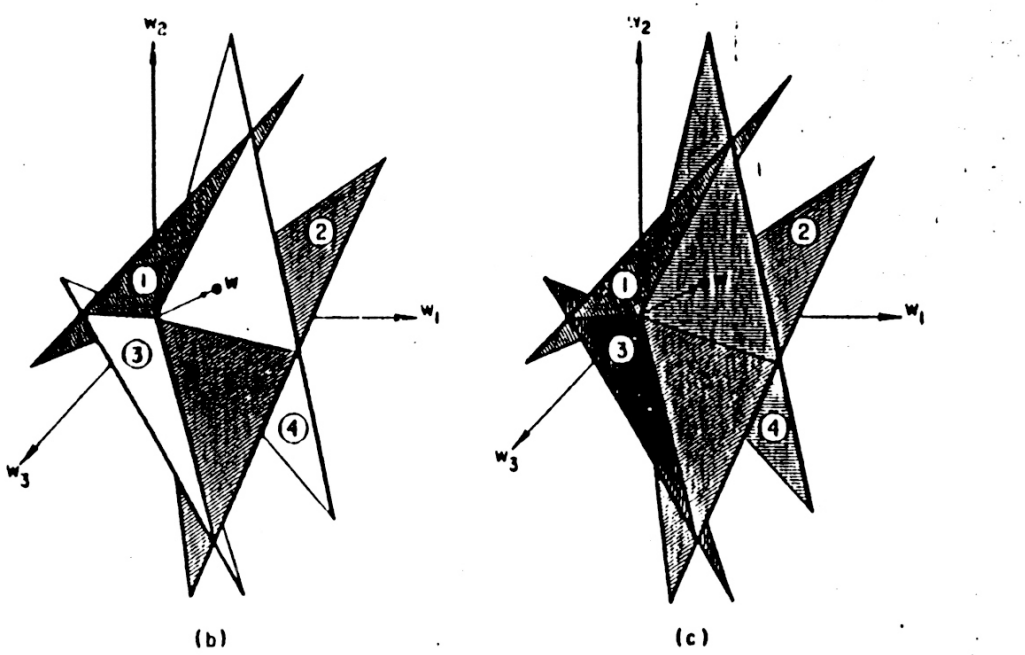
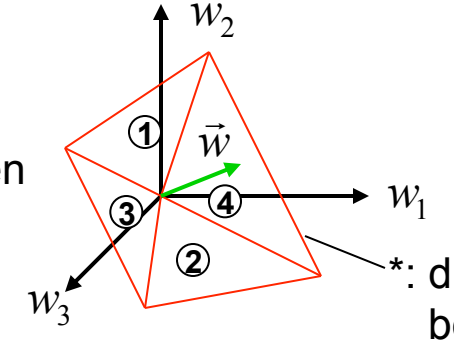


Figure 2.5. Geometrical illustration of the pattern space and the weight space. (a) Pattern space. (b) Weight space corresponding to inequalities (2.4–1). (c) Weight space corresponding to inequalities (2.4–2). Shaded areas indicate positive side of planes

⇒ identischer Lösungsvektor,
i.a. Fall ist „Lösungsraum“ durch konvexen polyhedralen Konus* begrenzt!



*: durch ebene Flächen begrenzter „Kegel“

- geometrische Eigenschaften linearer Entscheidungsfunktionen:
  - 2-Klassen-Fall-Hyperebene:

$$d(\vec{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1} = 0$$

  - Fall 1: $d_i(\vec{x}) = w_{i1} x_1 + w_{i2} x_2 + \ldots + w_{in} x_n + w_{i,n+1} = 0$

2: $d_{ij}(\vec{x}) = w_{ij_1} x_1 + w_{ij_2} x_2 + \ldots + w_{ij_n} x_n + w_{i,n+1} = 0$

3: $d_{ij}(\vec{x}) = d_i(\vec{x}) - d_j(\vec{x}) = 0$

$$= (w_{i1} - w_{j1}) x_1 + (w_{i2} - w_{j2}) x_2 + \ldots + (w_{in} - w_{jn}) x_n + \left(w_{i,n+1} - w_{j,n+1}\right) = 0$$

⇒ verallgemeinerte Form:

$$d(\vec{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1} = 0$$

$$= \vec{w}_0{}^T \vec{x} + w_{n+1} = 0$$

mit $\vec{w}_0 = (w_1, w_2, \ldots, w_n)^T$

$n = 2$: Gerade, $n = 3$: Ebene, $n > 3$: Hyperebene ($(n\text{-}1)$–dimensional)

Fig 2.6 (aus Tou & Gonzalez): Hyperebene am Punkt $\vec{p}$:

$$\vec{u}^T(\vec{x} - \vec{p}) = 0$$

$$\vec{u}^T\vec{x} = \vec{u}^T\vec{p}$$

Normale: $\vec{u} = \frac{\vec{w}_0}{\|\vec{w}_0\|}$ („Orientierung“ der Hyperebene)

mit der Norm $\|\vec{w}_0\| = \left(w_1^2 + w_2^2 + \ldots + w_n^2\right)^{\frac{1}{2}}$

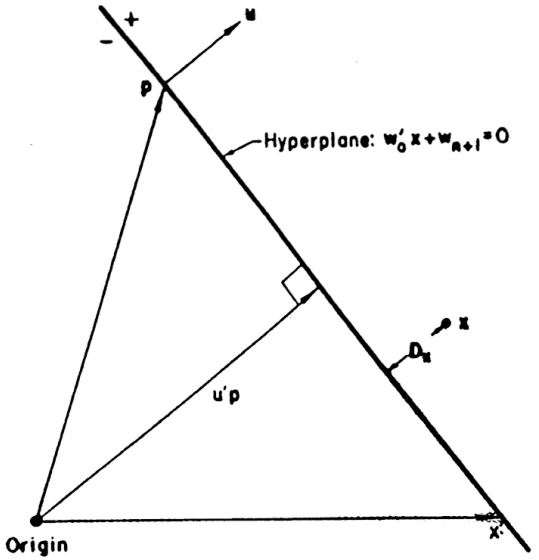


Figure 2.6. Some geometrical properties of hyperplanes

da $\vec{w}_0{}^T\vec{x} + w_{n+1} = 0 \qquad \left|\div\|\vec{w}_0\|\right.$

$$\frac{\vec{w}_0{}^T\vec{x}}{\|\vec{w}_0\|} = -\frac{w_{n+1}}{\|\vec{w}_0\|} \quad \Leftrightarrow \vec{u}^T\vec{x} = \vec{u}^T\vec{p}$$

$$\Rightarrow \vec{u}^T = \frac{\vec{w}_0{}^T}{\|\vec{w}_0\|} \Leftrightarrow \vec{u} = \frac{\vec{w}_0}{\|\vec{w}_0\|} \quad \text{und } \vec{u}^T\vec{p} = -\frac{w_{n+1}}{\|\vec{w}_0\|}$$

somit Abstand vom Ursprung zur Hyperebene: $D_u = \frac{|w_{n+1}|}{\|\vec{w}_0\|} = |\vec{u}^T \vec{p}|$

D_x für beliebigen Punkt $\vec{x}$: $D_x = |\vec{u}^T \vec{x} - \vec{u}^T \vec{p}|$

$$= \left| \frac{\vec{w}_0{}^T \vec{x} + w_{n+1}}{\|\vec{w}_0\|} \right|$$ = Abstand eines beliebigen Punktes zur (Trenn-)Hyperebene

ergo: - eine der $\vec{u}$-Komponeten ist gleich Null, dann ist Trennfläche parallel zu der jeweils korrespondierenden Achse ($\vec{u} = \vec{w}_0 / \|\vec{w}_0\|$)!

- $w_{n+1} = 0$: Hyperebene schneidet Ursprung!

- **Dichotomien**
  - Beurteilung der „Unterscheidungskraft“ (‘discriminatory power‘) von Diskriminantenfunktionen

- Dichotomien: Anzahl der Möglichkeiten, eine Mustermenge zu klassifizieren

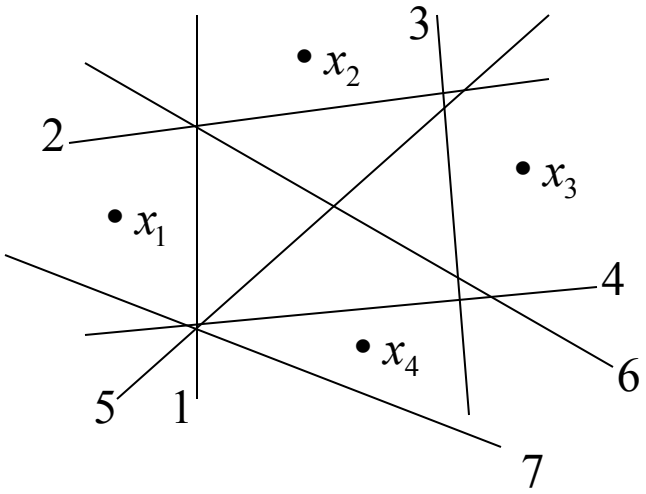


Fig 2.7 „lineare Dichotomien von 4 wohlverteilten Mustern in 2 Dimensionen“

z.B. $x_1 \propto \omega_1$ oder $x_1 \propto \omega_2$:
2 mögl. Klassifikationen durch Linie 1
($\omega_1 = \{x_1\}$ und $\omega_2 = \{x_2, x_3, x_4\}$
oder umgekehrt)!

d.h. hier 14 mögl. 2-Klassen-Trennungen
lin. Dichotomien!

von 2^4=16 maximal mögl. Dichotomien
(davon 2 nichtlineare)

Verallgemeinerung: „Die Anzahl ***D*** der (linearen) Dichotomien von N Punkten ≡ Vektoren in einem n-dimensionalen Raum ist gleich dem Zweifachen der Möglichkeiten, die N Punkte mit (n-1)-dimensionalen Hyperebenen zu trennen, d.h. nur wenn N Punkte „wohlverteilt“ sind, d.h. keine $n+1$ Punkte liegen auf n-1-dim. Hyperebene (z.B. keine 3 Punkte auf einer Linie), gilt

$$\boldsymbol{D}(N,n) = \begin{cases} 2\sum_{k=0}^{n} C_k^{N-1}, & N > n+1 \\ 2^N, & N \le n+1 \end{cases} \quad \text{mit } C_k^{N-1} = \frac{(N-1)!}{(N-1-k)!k!} = \binom{N-1}{k}$$

somit: ***D*** als Maß für Unterscheidungskraft

z.B. für $N = 10$ Punkte und $n = 2$ Dimensionen des Merkmalraums,

aus Tab. 2.2: $\boldsymbol{D}(10,2)=92$ lineare Dichotomien

z.B. für Polynom 2. Grades $d^2(\vec{x}) = w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 + w_1x_1 + w_2x_2 + w_3$

gilt (!) $n = N_w - 1 = 6 - 1 = 5$ für den Gewichtsraum*,

aus Tab. 2.2: $\boldsymbol{D}(10,5)=764$ lin. Dichotomien

TABLE 2.2. *Evaluation of* $\mathscr{D}(N, n)$

N \ n	1	2	3	4	5	6
1	2	2	2	2	2	2
2	4	4	4	4	4	4
3	6	8	8	8	8	8
4	8	14	16	16	16	16
5	10	22	30	32	32	32
6	12	32	52	62	64	64
7	14	44	84	114	126	128
8	16	58	128	198	240	254
9	18	74	186	326	438	494
10	20	92	260	512	764	932
25	50	602	4,650	15,662	100,670	379,862
50	100	2,452	39,300	463,052	4,276,820	32,244,452
100	200	9,902	323,600	7,852,352	150,898,640	2,391,957,152
200	400	39,802	2,627,200	129,409,702	5,073,927,280	164,946,662,302

⇒ ***D***(N, n) lineare Dichotomien bei wohlverteilten N Punkten, dann ***D***(N, n) konvexe polyhedrale Koni im Gewichtsraum von N n-dimensionalen Merkmalsvektoren (= ‘patterns ‘)

⇒ ***D*** als Maß für „Klassifikationskraft“, d.h. “... the greater the number of implementable dichotomies for a given N, the better our chances are of finding a solution to the given inequalities ...“

- Dichotomisierungs-Kapazität für verallgemeinerte Diskriminantenfunktionen für $d(\vec{x}) = w_1 f_1(\vec{x}) + w_2 f_2(\vec{x}) + \ldots + w_k f_k(\vec{x}) + w_{k+1}$ mit $k+1$ Gewichten bei N transformierten wohlverteilten Punkten (im k-dimensionalen Raum der transformierten Punkte) existieren daher 2^N Dichotomien, von denen ***D***(N, k) “linearly implementable“ sind!

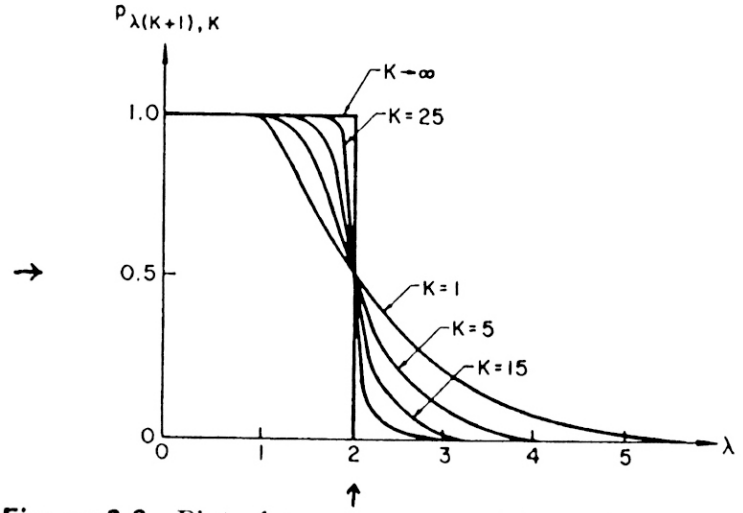


Figure 2.8. Plot of $p_{\lambda(K+1),K}$ versus λ for various values of K

- Wahrscheinlichkeit, dass eine zufällig gewählte Diskriminantenfunktion linear ist

$$p_{N,k} = \frac{\boldsymbol{D}(N,k)}{2^N} = \begin{cases} 2^{1-N} \sum_{j=0}^{k} C_j^{N-1}, & \text{für } N > k+1 \\ 1, & \text{für } N \le k+1 \end{cases}$$

d.h. für $N \le k+1$ immer linear separierbar, auch wenn Punkte nicht wohlverteilt!

Beispiel: Fig. 2.8 mit $N = \lambda(k+1)$

$k \to \infty$: "... almost guaranteed the ability to totally classify $N = 2(k+1)$ well-distributed patterns with a generalized decision fct. of $k+1$ parameters ..."

$N > 2(k+1)$: "... propability of achieving a dichotomy declines sharply for similarly large values of k..."

- "dichotomization capacity" von verallgemeinerten Diskriminantenfunktionen

$$C_k = 2(k+1)$$

für Hyperebene: $2(n+1)$

Hyperkugel: $2(n+2)$

allg. quadr. Fläche: $(n+1)(n+2)$

polynomische Fläche r-ter Ordnung: $2C_r^{n+r}$

für jeweils n-dimensionalen Merkmalraum

Diskriminantenfunktionen – Teil 2

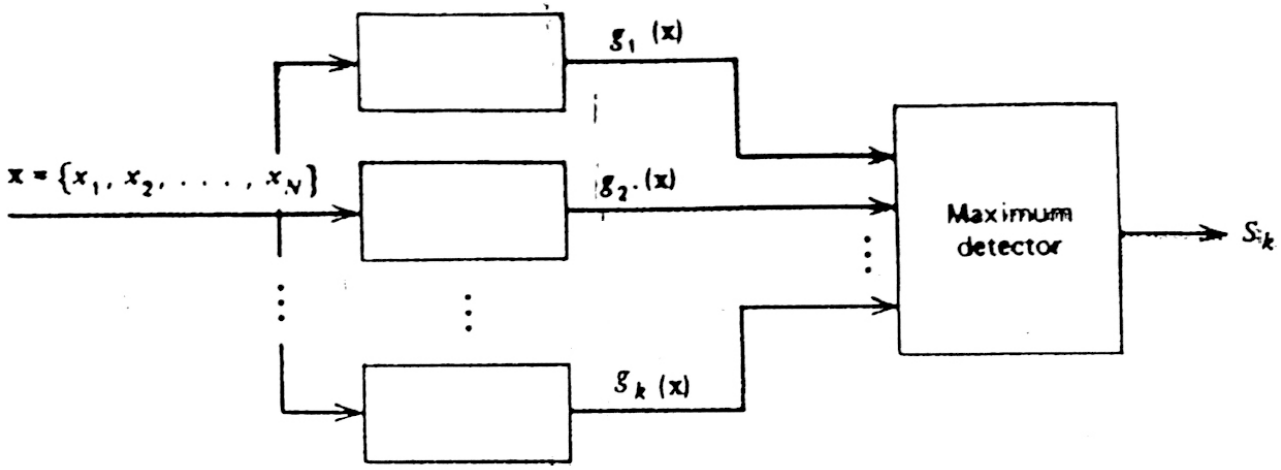
(Quellen: Tou & Gonzalez, 1974, chapt. 5 “Trainable Pattern Classifiers – The Deterministic Approach“;
siehe auch: Duda & Hart, 1973, chapt. 5 “Linear Discriminant Functions“ zur Vertiefung)

- Motivation: Definition von Klassifikationen, deren Entscheidungsfunktionen durch „lernende Verfahren“ aus der sog. Lernstichprobe bestimmt werden,
  d.h. „Lernen“ der Koeffizienten einer Diskriminantenfunktion ohne Annahmen über die Statistik der Stichprobe → deterministischer Ansatz

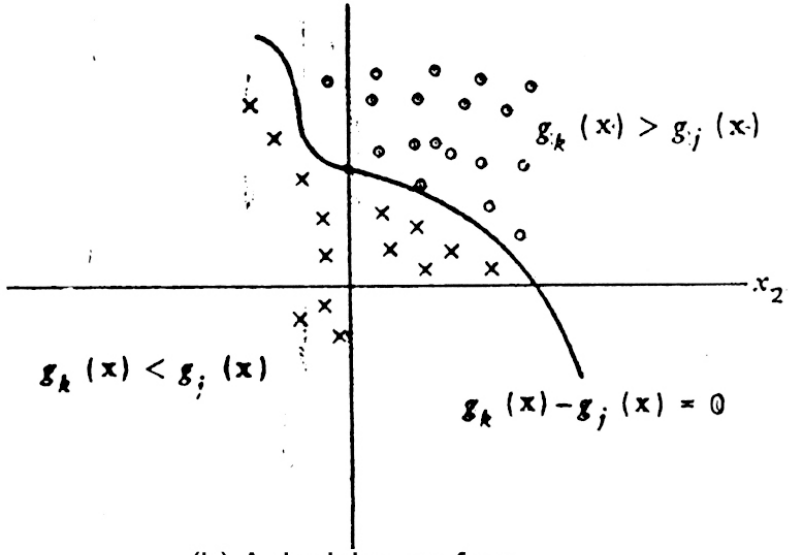
- Hintergrund:
  Lösung des 2-Klassen-Problems = Lösung eines Systems linearer Ungleichungen,
  d.h. für ω_1 und ω_2 wird Lösungsvektor gesucht, so daß gilt:

$$\vec{x} \propto \omega_1 \Leftrightarrow \vec{w}^T \vec{x} > 0$$
und
$$\vec{x} \propto \omega_2 \Leftrightarrow \vec{w}^T \vec{x} < 0$$

! Einschränkung: “... whenever these training pattern sets are separable by the specified decision functions.“



(a) A typical classifier



(b) A decision surface

FIGURE 3.2. Calculation and example of discriminant functions.

- Ungleichungssystem (zur Erinnerung!)

  - durch Multiplikation mit -1 für $\vec{x} \in \omega_2$ ergibt sich:

$$\vec{x} \propto \omega_2, \text{ wenn } \vec{w}^T \vec{x} > 0$$

  und damit (für N Stichproben):

$$\boxed{\mathbf{X}\vec{w} > \vec{0}} \quad \text{mit} \quad \vec{w} = (w_1, w_2, \ldots, w_n, w_{n+1})^T$$

$\vec{0}$: Nullvektor

$$\mathbf{X} = \begin{pmatrix} \vec{x}_1' \\ \vec{x}_2' \\ \vdots \\ \vec{x}_n' \end{pmatrix}$$

! Achtung: jeweils "augmented" Vektoren

  - Ungleichungssystem ist konsistent, wenn Lösungsvektor $\vec{w}$ existiert, d.h. die Klassen sind separierbar (bzw. inkonsistent für den nicht-separierbaren Fall)

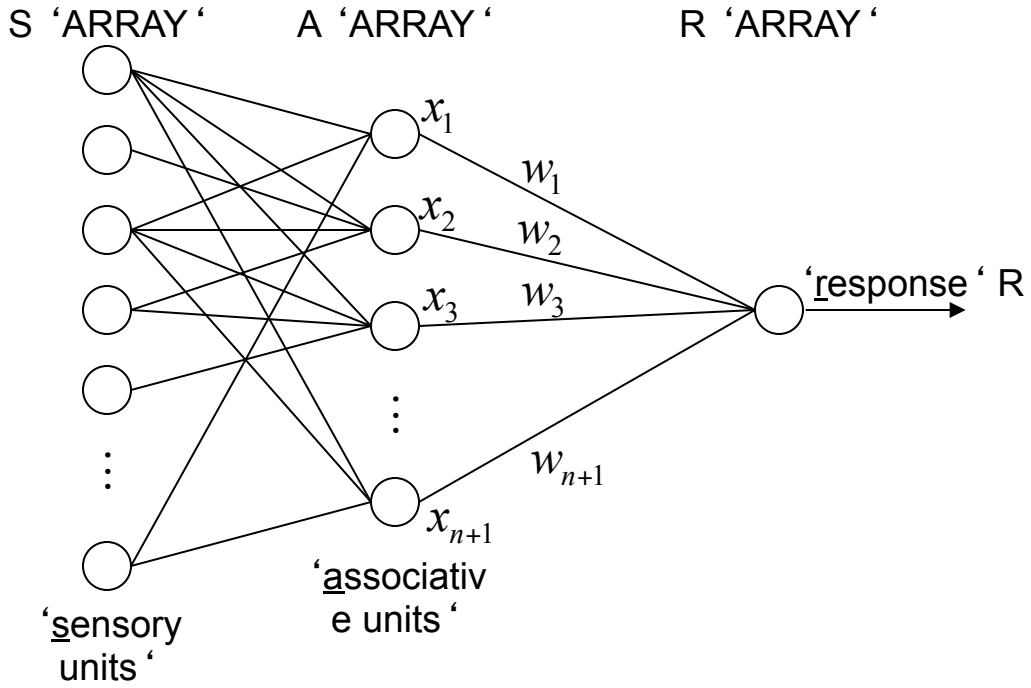
  - Problem: Bestimmung von $\vec{w}$, so dass Klassen „widerspruchsfrei" getrennt werden

  - Hinweis: siehe auch „Merkmal- und Gewichtsraum", „Dichotomien", etc.

- **Perzeptron-Ansatz**

  - Theorie: Rosenblatts frühe bionisch-motivierte Arbeiten zum maschinellen Lernen (1957)

  - Modell (für 2-Klassen-Problem)



$$R = \sum_{i=1}^{n+1} w_i x_i = \vec{w}^T \vec{x}$$

Klassifikation mit $\vec{x} \propto \omega_1$, wenn $R > 0$

$\vec{x} \propto \omega_2$, wenn $R < 0$

! Multi-Klassen-Problem: M ‘units‘ $R_1, R_2, ..., R_M$

$\Rightarrow \vec{x} \propto \omega_i$, wenn $R_i > R_j, \forall\, i \neq j$

- „Lernen“ des Gewichtsvektors $\vec{w}$: trivialer Algorithmus
  - iterative Bestimmung von $\vec{w}$
  - Belohnung-Bestrafung-Konzept (‘reward & punishment ‘)*

1) initialer Gewichtsvektor $\vec{w}(1)$, für $k = 1$

2) im k-ten Trainingsschritt:

c := Korrekturinkrement (konstant und $c > 0$)

begin

if $\vec{x}(k) \propto \omega_1 \ \wedge \ \vec{w}^T(k)\vec{x} \leq 0$

then { $\vec{w}(k+1) := \vec{w}(k) + c \cdot \vec{x}(k);$ goto end; }

else if $\vec{x}(k) \propto \omega_2 \ \wedge \ \vec{w}^T(k)\vec{x}(k) \geq 0$

then { $\vec{w}(k+1) := \vec{w}(k) - c \cdot \vec{x}(k);$ goto end; }

fi

fi

$\vec{w}(k+1) := \vec{w}(k)$

end

„Bestrafung“

„Belohnung“

d.h. $\vec{w}(k)$ ändert sich nur bei Fehl-Klassifikation (!) in k-ter Iteration

z.B. muss gelten: $\vec{x}(k) \in \omega_1$, wenn $\vec{w}^T\vec{x} > 0$; ist jedoch $\vec{w}^T\vec{x} \leq 0$, so wird $\vec{w}(k)$ inkrementiert

* gemäß der Theorie des Lernens im Kontext des Behaviorismus!

„A synapse

is strengthened or stabilized

when pre- and post-synaptic activations

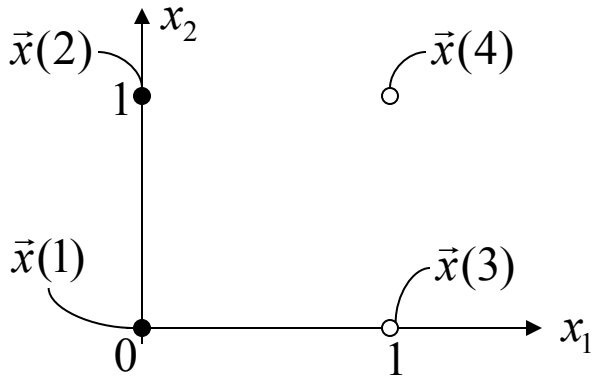
are correlated.“

(D. Hebb, *Organization of Behaviour*, New York: Wiley)

- Beispiel (aus Tou & Gonzalez)
  - ‘augmented‘ Klassenvektoren einer vorklassifizierten Stichprobe
    (d.h. überwachtes Lernen!)

$$\omega_1 : \left\{(0,0,1)^T, (0,1,1)^T\right\} (\bullet \in \omega_1)$$

$$\omega_2 : \left\{(1,0,1)^T, (1,1,1)^T\right\} (\circ \in \omega_2)$$

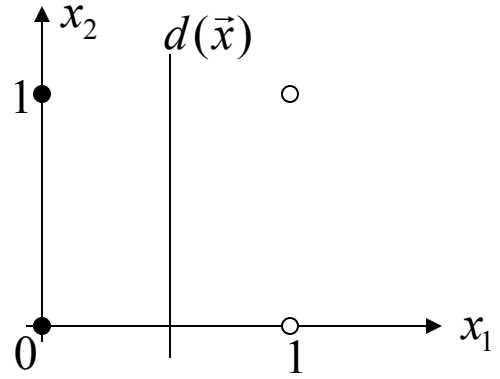


- Algorithmus mit $c = 1$ und $\vec{w}(0) = \vec{0}$
  (siehe Buchauszug aus Tou & Gonzalez, 1974, pp. 162-164 auf den nächsten Folien)
- Ergebnis: lineare Diskriminantenfunktion

$$d(\vec{x}) = -2x_1 + 1 = 0$$

$$\Downarrow$$

$$x_1 = \frac{1}{2}$$



- äquivalente Form des Algorithmus (für den Fall der Multiplikation der Ungleichungen – mit ‘augmented vectors‘ – für z.B. ω_2 mit -1:

$$\vec{w}(k+1) := \begin{cases} \vec{w}(k), & \text{falls } \vec{w}^T(k)\vec{x}(k) > 0 \\ \vec{w}(k) + c\vec{x}(k), & \text{falls } \vec{w}^T(k)\vec{x}(k) \le 0 \end{cases}$$

Iterative Bestimmung des Gewichtsvektors mit dem Perzeptron-Algorithmus

(aus: Tou & Gonzalez, 1974, chapt. 5)

$$\mathbf{w}'(1)\mathbf{x}(1) = (0, 0, 0)\begin{pmatrix}0\\0\\1\end{pmatrix} = 0, \qquad \mathbf{w}(2) = \mathbf{w}(1) + \mathbf{x}(1) = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(2)\mathbf{x}(2) = (0, 0, 1)\begin{pmatrix}0\\1\\1\end{pmatrix} = 1, \qquad \mathbf{w}(3) = \mathbf{w}(2) = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(3)\mathbf{x}(3) = (0, 0, 1)\begin{pmatrix}1\\0\\1\end{pmatrix} = 1, \qquad \mathbf{w}(4) = \mathbf{w}(3) - \mathbf{x}(3) = \begin{pmatrix}-1\\0\\0\end{pmatrix}$$

$$\mathbf{w}'(4)\mathbf{x}(4) = (-1, 0, 0)\begin{pmatrix}1\\1\\1\end{pmatrix} = -1, \quad \mathbf{w}(5) = \mathbf{w}(4) = \begin{pmatrix}-1\\0\\0\end{pmatrix}$$

where corrections on the weight vector were made in the first and third steps because of misclassification, as indicated in Eqs. (5.2-2) and (5.2-3). Since a solution has been obtained only when the algorithm yields a complete, error-free iteration through all patterns, the training set must be presented again. The machine learning process is continued by letting $\mathbf{x}(5) = \mathbf{x}(1)$, $\mathbf{x}(6) = \mathbf{x}(2)$, $\mathbf{x}(7) = \mathbf{x}(3)$, and $\mathbf{x}(8) = \mathbf{x}(4)$. The second iteration through the patterns yields:

$$\mathbf{w}'(5)\mathbf{x}(5) = 0, \qquad \mathbf{w}(6) = \mathbf{w}(5) + \mathbf{x}(5) = \begin{pmatrix}-1\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(6)\mathbf{x}(6) = 1, \qquad \mathbf{w}(7) = \mathbf{w}(6) = \begin{pmatrix}-1\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(7)\mathbf{x}(7) = 0, \qquad \mathbf{w}(8) = \mathbf{w}(7) - \mathbf{x}(7) = \begin{pmatrix}-2\\0\\0\end{pmatrix}$$

$$\mathbf{w}'(8)\mathbf{x}(8) = -2, \quad \mathbf{w}(9) = \mathbf{w}(8) = \begin{pmatrix}-2\\0\\0\end{pmatrix}$$

Since two errors occurred in this iteration, the patterns are presented again

$$\mathbf{w}'(9)\mathbf{x}(9) = 0, \qquad \mathbf{w}(10) = \mathbf{w}(9) + \mathbf{x}(9) = \begin{pmatrix}-2\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(10)\mathbf{x}(10) = 1, \qquad \mathbf{w}(11) = \mathbf{w}(10) = \begin{pmatrix}-2\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(11)\mathbf{x}(11) = -1, \qquad \mathbf{w}(12) = \mathbf{w}(11) = \begin{pmatrix}-2\\0\\1\end{pmatrix}$$

$$\mathbf{w}'(12)\mathbf{x}(12) = -1, \qquad \mathbf{w}(13) = \mathbf{w}(12) = \begin{pmatrix}-2\\0\\1\end{pmatrix}$$

It is easily verified that in the next iteration all patterns are classified correctly. The solution vector is, therefore, $\mathbf{w} = (-2, 0, 1)'$. The corresponding decision function is $d(\mathbf{x}) = -2x_1 + 1$, which, when set equal to zero, becomes the equation of the decision boundary shown in Fig. 5.2(b). ●

- letzte Iteration nur für

$$x(13) = x(9) = x(1) \in \omega_1$$

ergo: $$w'(13) \cdot x(13) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}^T \cdot x(13)$$

$$= \begin{pmatrix} -2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0 + 0 + 1 = 1 > 0!$$

$\Rightarrow$ Lösungsvektor $\vec{w}^T = \begin{pmatrix} -2 & 0 & 1 \end{pmatrix}$

damit ergibt sich durch „algorithmisches Lernen“ die lineare Diskriminantenfunktion zu

$$d(\vec{x}) = \vec{w}^T \cdot \vec{x}$$

$$= \begin{pmatrix} -2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

‘augmented vector‘!

$$= -2x_1 + 1$$

d.h. ‘decision boundary‘ ist $-2x_1 + 1 = 0$ und es folgt $x_1 = \frac{1}{2}$.

- Konvergenz: in endlicher Zahl von Iterationen für den Fall linear separierbarer Klassen
- Variation des Perzeptron-Algorithmus
  - 'fixed increment'-Algorithmus: Konstante $c > 0$
  - 'absolute correction'-Algorithmus:

    für $\vec{w}^T(k)\vec{x}(k) \leq 0$: Wahl von c so, daß

    $\vec{w}^T(k+1)\vec{x}(k) = \left(\vec{w}(k) + c \cdot \vec{x}(k)\right)^T \vec{x}(k) > 0$ wird,

    mit z.B. c als kleinste ganze Zahl größer als $\dfrac{\left|\vec{w}^T(k)\vec{x}(k)\right|}{\vec{x}^T(k)\vec{x}(k)}$
  - 'fractional correction'-Algorithmus:

    $c = \lambda \dfrac{\left|\vec{w}^T(k)\vec{x}(k)\right|}{\vec{x}^T(k)\vec{x}(k)}$ mit $\vec{w}(1) \neq 0$

    z.B. für $\lambda > 1$ korrekte Klassifikation eines Musters nach jeder Gewichtsänderung

    (Konvergenz für $0 < \lambda < 2$)

- **Gradienten-Verfahren**
  - Grundlage: Gradienten-Vektoren zur „Suche“ nach Minimum einer Funktion
  - Definition des Gradienten:

    Funktion $f(\vec{y})$, Vektor $\vec{y} = (y_1, y_2, \ldots, y_n)^T$

$$\text{grad } f(\vec{y}) = \frac{df(\vec{y})}{d\vec{y}} = \begin{pmatrix} \frac{\partial f}{\partial y_1} \\ \vdots \\ \frac{\partial f}{\partial y_n} \end{pmatrix}$$

d.h. Gradient einer Skalarfunktion eines Vektors ist wiederum ein Vektor

- Eigenschaft: Gradientenvektor entlang der Richtung des z.B. maximalen Anstiegs der Funktion

  $\Rightarrow$ iteratives Verfahren zur Minimumbestimmung einer Funktion

- Lösungsansatz[1]:
  - Ungleichungssystem $\mathbf{X}\vec{w} > \vec{0}$, mit $\vec{w} = (w_1, w_2, \ldots, w_{n+1})^T$ und $X = \begin{pmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_N^T \end{pmatrix}$

z.B.
$$\begin{aligned} w_1 x_{11}^1 + w_2 x_{12}^1 + w_3 &> 0 \\ w_1 x_{21}^1 + w_2 x_{22}^1 + w_3 &> 0 \\ -w_1 x_{11}^2 - w_2 x_{12}^2 - w_3 &> 0 \\ -w_1 x_{21}^2 - w_2 x_{22}^2 - w_3 &> 0 \end{aligned}$$

mit Lösungsvektor $\vec{w} = (w_1, w_2, w_3)^T$

$$\left.\begin{aligned} \omega_1 &= \{\vec{x}_1^1, \vec{x}_2^1\} \\ \omega_2 &= \{\vec{x}_2^1, \vec{x}_2^2\} \\ \vec{x}_1^1 &= (x_{11}^1, x_{12}^1, 1)^T \\ &\text{usw.} \end{aligned}\right\}$$

2-Klassen-Problem (Annahme: ω_i linear separierbar !)

[1]Hinweis: nur für Funktionen mit einem eindeutigen Minimum!

- Wahl einer Funktion $I(\vec{w},\vec{x})$*[1] so, damit $\min(I)$ wenn für alle Ungleichungen des obigen Systems $\vec{w}^T\vec{x}_i > 0, i = 1,\ldots,N$ gilt!

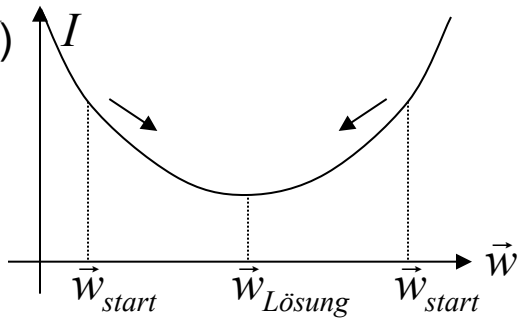
  $\Rightarrow$ Lösungsvektor $\vec{w}$ eindeutig bestimmt

- allgemeine Gradientenabstiegsmethode (‘general gradient descent method ‘)

  Schritt 1: „Inkrementiere“ $\vec{w}$ in Richtung des negativen Gradienten von $I(\vec{w},\vec{x})$

  Schritt 2: wenn Minimum noch nicht erreicht, wiederhole Schritt 1

  d.h. (schematisch für konvexe Funktion I)



- Formalisierung*[3]:

  sei $\vec{w}(k)$ Gewichtsvektor in der k-ten Iteration

$$\vec{w}_{k+1} := \vec{w}_k - c\left\{\frac{\partial I(\vec{w},\vec{x})}{\partial \vec{w}}\right\}_{\vec{w}=\vec{w}(k)} \quad \text{für} \quad \frac{\partial I(\vec{w},\vec{x})}{\partial \vec{w}} \neq 0$$

$$\vec{w}_{k+1} := \vec{w}_k \quad \text{für} \quad \frac{\partial I(\vec{w},\vec{x})}{\partial \vec{w}} = 0$$

mit c: „Größenordnung“ der Korrektur (c>0)

---

*[1] oder: i.a. Funktional

*[2] Hinweis: triviale Lösung für $\vec{w} = 0$ ist hier irrelevant

*[3] Beispiel Fig. 5.3, Seite 34

- Hinweise: i) immer Lösung für konsistente Ungleichungen und Wahl einer "... proper ..." Funktion I

  ii) Problem der „Oszillation" (bei Inkonsistenz)

  iii) Lösung für multiple Minima: siehe z.B. aktuelle Forschung zu neuronalen Netzen (Teil F)

- Perzeptron-Algorithmus* ('revisited ')
  - P.A.: ein spezielles Mitglied aus der Familie der iterativen Gradienten-Verfahren!
  - Wahl der Fkt. I

$$I(\vec{w},\vec{x}) = \frac{1}{2}\left(\left|\vec{w}^T\vec{x}\right| - \vec{w}^T\vec{x}\right)$$

$$\frac{\partial I}{\partial \vec{w}}(\vec{w},\vec{x}) = \frac{1}{2}\left(\vec{x}\cdot \mathrm{sgn}\left(\vec{w}^T\vec{x}\right) - \vec{x}\right)$$

$$\text{mit } \mathrm{sgn}\left(\vec{w}^T\vec{x}\right) = \begin{cases} 1, & \text{falls } \vec{w}^T\vec{x} > 0 \\ -1, & \text{falls } \vec{w}^T\vec{x} \le 0 \end{cases}$$

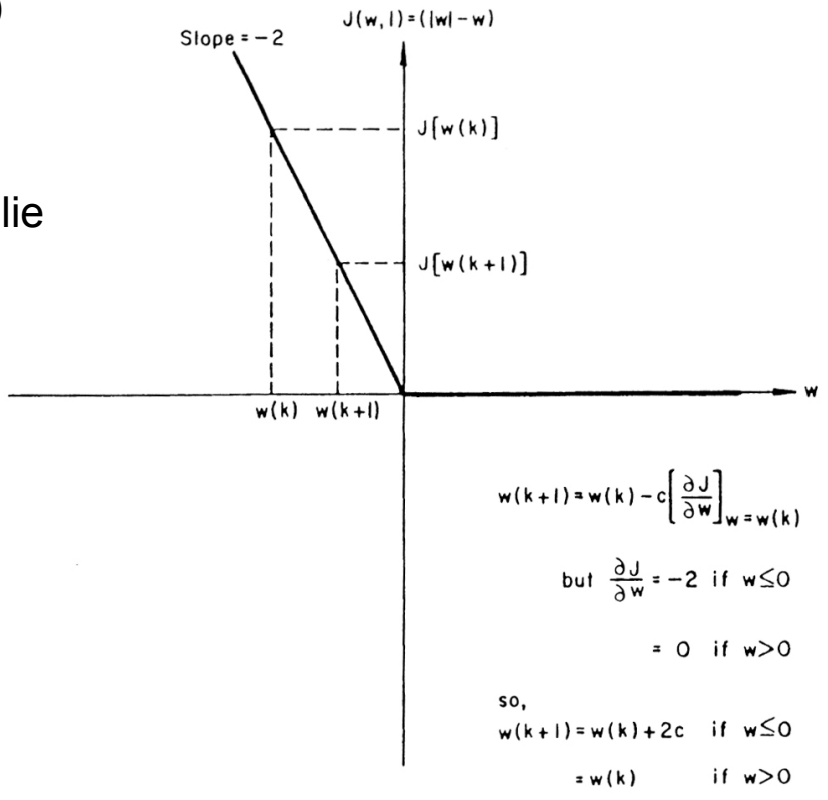


Figure 5.3. Geometrical illustration of the gradient descent algorithm

*: aus Tou & Gonzalez, 1974

- Gradientenabstiegs-Verfahren

$$\vec{w}(k+1) = \vec{w}(k) + \frac{c}{2}\left\{\vec{x}(k) - \vec{x}(k)\cdot \operatorname{sgn}\left[\vec{w}^T(k)\vec{x}(k)\right]\right\} \text{ bzw.}$$

$$\boxed{\vec{w}(k+1) = \vec{w}(k) + c\begin{cases}\vec{0}, & \text{falls } \vec{w}^T(k)\vec{x}(k) > 0\\ \vec{x}(k), & \text{falls } \vec{w}^T(k)\vec{x}(k) \le 0\end{cases}} \quad \text{für } \mathbf{X}\cdot\vec{w} > 0$$

(“reward & punishment“) mit $c > 0$, $\vec{w}(1)$ beliebig

! Konvergenz für durch die Diskriminantenfunktionen separierbare Klassen

- **„Kleinste-Mittlere-Fehlerquadrate“-Algorithmus** (‘least-mean-square-error ‘ = LMSE-Alg.) → oder auch: Ho-Kashyap-Verfahren
  - Motivation:
    - Perzeptron-Algorithmus oszilliert im Fall von (durch die gewählten Diskriminanten-funktionen) nicht separierbaren Klassen der Stichproben
    - Anzahl der Iterationen bis zur Konvergenz nicht vorhersagbar, d.h. lange „Trainingszeit“ (aber: große Zahl von Iterationen muß nicht unbedingt „Oszillation“ bedeuten; s.o.)
  - Eigenschaften des LMSE-Algorithmus
    - Konvergenz für den separierbaren Fall gewährleistet und
    - explizite Bestimmung des Falles der inseparablen Klassen

- Lösungsansatz:
  - anstatt $\mathbf{X} \cdot \vec{w} > \vec{0}$:

    äquivalente Notation $\mathbf{X} \cdot \vec{w} = \vec{b}, \quad \vec{b} = (b_1, b_2, \ldots, b_N)^T, \quad b_i > 0$ (!)

  - Funktional ('criterion function ')

$$I(\vec{w}, \vec{x}, \vec{b}) = \frac{1}{2} \sum_{j=1}^{N} \underbrace{\left(\vec{w}^T \vec{x}_j - b_j \right)^2}_{\text{quadratischer Fehler}} = \frac{1}{2} \left\| \mathbf{X} \cdot \vec{w} - \vec{b} \right\|^2 \rightarrow \text{min}$$

kleinster!

proportional zu mittlerem ≡ gemitteltem quadrat. Fehler

  - Minimierung (bezüglich $\vec{w}$ und $\vec{b}$)

I) $\dfrac{\partial I}{\partial \vec{w}} = \mathbf{X}^T (\mathbf{X}\vec{w} - \vec{b})$

II) $\dfrac{\partial I}{\partial \vec{b}} = -(\mathbf{X}\vec{w} - \vec{b})$

ad I) für $\dfrac{\partial I}{\partial \vec{w}} = 0$ ergibt sich

$$\vec{w} = \underbrace{\left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T}_{\text{verallgem. Inverse von } \mathbf{X}^1} \vec{b} = \mathbf{X}^{\#} \vec{b} \qquad (\vec{w} \text{ beliebig!})$$

---

[1] $\mathbf{X}^{\#}$: Notation aus Tou & Gonzalez, 1974

ad II) Randbedingung $b_i > 0$!

$$\boxed{\vec{b}(k+1) = \vec{b}(k) + \delta \cdot \vec{b}(k)}$$ k: Iterationsindex

mit $\delta \cdot b_i = \begin{cases} 2c\left[\mathbf{X}\vec{w}(k) - \vec{b}(k)\right]_i, & \text{falls } \left[\mathbf{X}\vec{w}(k) - \vec{b}(k)\right]_i > 0 \\ 0, & \text{falls } \left[\mathbf{X}\vec{w}(k) - \vec{b}(k)\right]_i \le 0 \end{cases}$

$$\delta \cdot \vec{b}(k) = c\left[\mathbf{X}\vec{w}(k) - \vec{b}(k) + \left|\mathbf{X}\vec{w}(k) - \vec{b}(k)\right|\right]$$

Aus $\vec{w} = \mathbf{X}^{\#}\vec{b}$ und $\vec{b}(k+1) = \vec{b}(k) + \delta \cdot \vec{b}(k)$ folgt:

$$\vec{w}(k+1) = \mathbf{X}^{\#}\vec{b}(k+1) = \mathbf{X}^{\#}\left[\vec{b}(k) + \delta \cdot \vec{b}(k)\right] = \mathbf{X}^{\#}\vec{b}(k) + \mathbf{X}^{\#}\delta \cdot \vec{b}(k)$$

$$= \vec{w}(k) + \mathbf{X}^{\#}\delta \cdot \vec{b}(k) \quad \Rightarrow \text{iterativer Algorithmus}$$

- Algorithmus:

$\vec{w}(1) = \mathbf{X}^{\#}\vec{b}(1),$ mit $b_i(1) > 0,$ aber beliebig

$$\vec{w}(k+1) = \vec{w}(k) + c\mathbf{X}^{\#}\left[\vec{e}(k) + |\vec{e}(k)|\right]$$
$$\vec{b}(k+1) = \vec{b}(k) + c\left[\vec{e}(k) + |\vec{e}(k)|\right]$$

mit : $c > 0$

$\vec{e}(k) = \mathbf{X}\vec{w}(k) - \vec{b}(k)$: ‘error‘ in k-ter Iteration

- Hinweise: i) $\vec{w}(k+1) = \mathbf{X}^{\#}\vec{b}(k+1)$

ii) Konvergenz für $0 < c \leq 1$, wenn für Ungleichungssystem $\mathbf{X}\vec{w} > \vec{0}$ Lösung $\vec{w}$ existiert.
(prinzipiell: in weniger Iterationen als Perzeptron-Algorithmus)

iii) Separabilitätstest: Klassen sind durch Trenn-Hyperebenen nicht separierbar, wenn während einer beliebigen Iteration alle Komponenten von $\vec{e}(k) \leq 0$ (<u>aber</u> nicht alle zugleich null) sind!

Example: (a) Consider again the pattern classes ω_1: $\{(0, 0)', (0, 1)'\}$ and ω_2: $\{(1, 0)', (1, 1)'\}$. Augmenting the patterns and multiplying the patterns of class ω_2 by -1 results in the matrix

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

The generalized inverse $\mathbf{X}^\sharp = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is

$$\mathbf{X}^\sharp = \frac{1}{2}\begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Letting $\mathbf{b}(1) = (1, 1, 1, 1)'$ and $c = 1$, and applying the algorithm of Eqs. (5.3–23), we obtain

$$\mathbf{w}(1) = \mathbf{X}^\sharp\mathbf{b}(1) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

Since

$$\mathbf{X}\mathbf{w}(1) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

we see, according to Eq. (5.3–13), that $\mathbf{w}(1)$ is a solution. In practice, we say that a solution has been achieved whenever $\mathbf{X}\mathbf{w} > \mathbf{0}$.

(b) Consider now the classes ω_1: $\{(0, 0)', (1, 1)'\}$ and ω_2: $\{(0, 1)', (1, 0)'\}$ which are not linearly separable. Letting $c = 1$ and $\mathbf{b}(1) = (1, 1, 1, 1)'$, we obtain

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

$$\mathbf{X}^\sharp = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ \frac{3}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

$$\mathbf{w}(1) = \mathbf{X}^\sharp\mathbf{b}(1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{e}(1) = \mathbf{X}\mathbf{w}(1) - \mathbf{b}(1) = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

The fact that $\mathbf{e}(1)$ is a negative vector indicates that $\mathbf{X}\mathbf{w} > \mathbf{0}$ has no solution.

(aus Tou & Gonzalez, 1974, chapter 5, pp. 175-177)

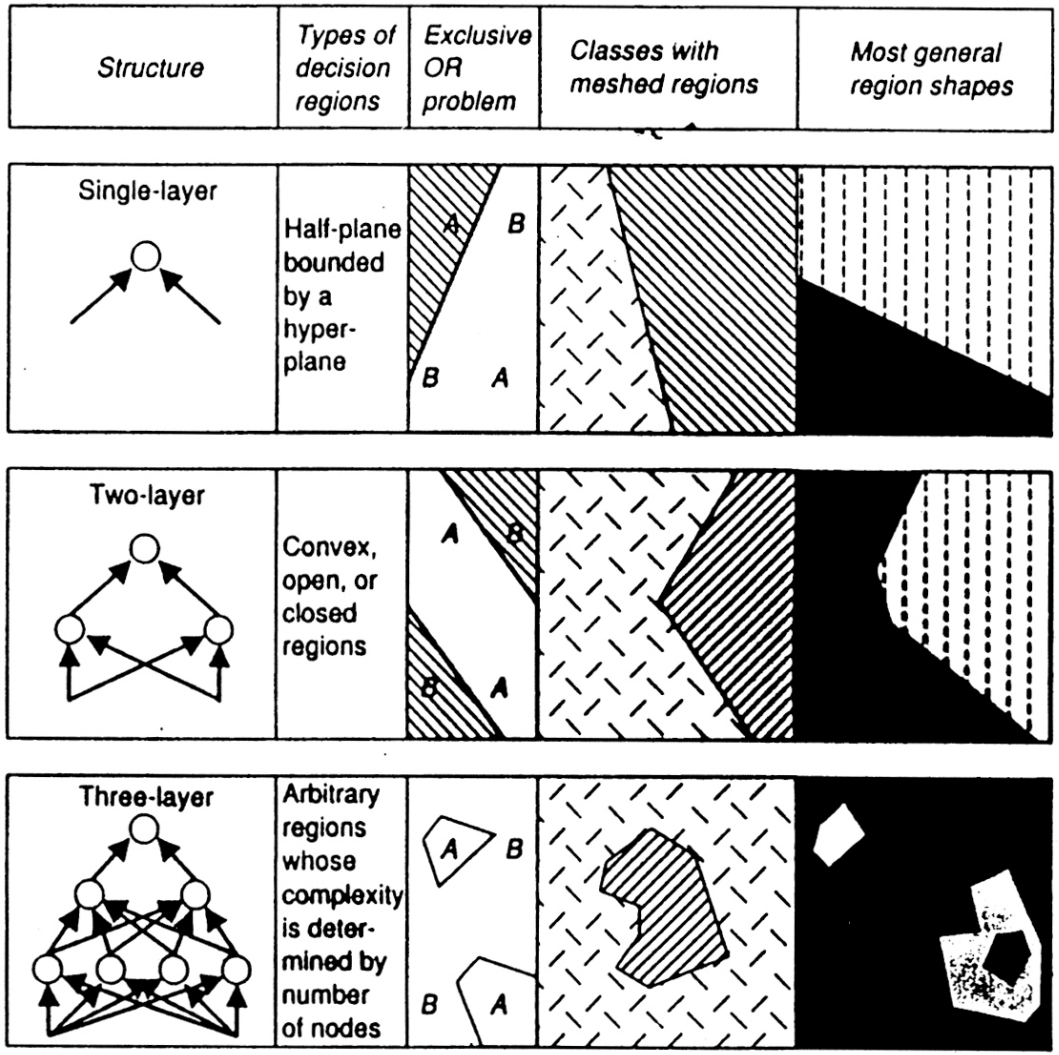


Figure 3.8 ***Types of decision regions that can be formed by single- and multilayer perceptrons with one and two layers of hidden units and two inputs.***

Different shading is used to denote the decision regions for classes *A* and *B*. Smooth closed contours bound input distributions for classes *A* and *B*. Nodes in all nets use hard-limiting nonlinearities.

Reprinted with permission from R. Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, April 1987. © 1987 IEEE.

aus: Tarun Khanna (1989)
Foundation of Neural Networks,
Addiseon Wesley

- **Methode der Potentialfunktionen** (einfaches physikalisches Modell)
  - 2-Klassen-Problem (ω_1 und ω_2): (siehe Fig. 5.12)

    Vektoren der Stichprobe, $\vec{x}_k$, als „Energiequellen“,

    d.h. Punkte mit elektrischen (z.B. Einheits-)Ladungen, so dass
    - Potential an $\vec{x}_k$ Extremalwert annimmt und in Umgebung $\vec{x}_k \pm \Delta\vec{x}$ stark abfällt
    - positive Ladung für $\vec{x}_k \in \omega_1$ und vice versa, d.h. Polarität pro Klasse

    $\Rightarrow$ Trenn-Hyperfläche als resultierendes elektrostatisches Potential

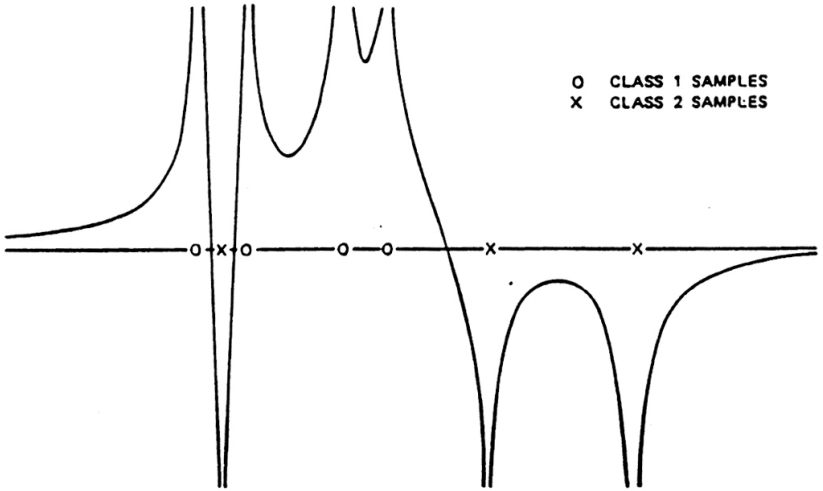


FIGURE 5.12. The potential field as a discriminant function.

The method originally developed from the idea that if the samples $\mathbf{x}_i$ were thought of as points in space, and if electrical charges q_i were placed at these points, positive if $\mathbf{x}_i$ were labelled ω_1 and negative if $\mathbf{x}_i$ were labelled ω_2, then perhaps the resulting electrostatic potential would serve as a useful discriminant function (see Figure 5.12). If the potential at a point $\mathbf{x}$ due to a unit charge at a point $\mathbf{x}_i$ is $K(\mathbf{x}, \mathbf{x}_i)$, then the potential due to n charges is given by

$$g(\mathbf{x}) = \sum_{i=1}^{n} q_i K(\mathbf{x}, \mathbf{x}_i). \tag{85}$$

The *potential function* $K(\mathbf{x}, \mathbf{x}_i)$ of classical physics varies inversely with $\|\mathbf{x} - \mathbf{x}_i\|$, but many other functions are equally suitable. There is a clear analogy between $K(\mathbf{x}, \mathbf{x}_i)$ and the Parzen-window function $\varphi[(\mathbf{x} - \mathbf{x}_i)/h]$, and the behavior of the discriminant function $g(\mathbf{x})$ is generally similar to the behavior of the difference of the Parzen-window estimates of two densities.

(aus: Duda & Hart, 1973, chapter 5, p. 172)

- Potentialfunktion am Ort $\vec{x}_k$ des Merkmalraums

$$K(\vec{x},\vec{x}_k)=\sum_{i=1}^{\infty}\lambda_i^2\varphi_i(\vec{x})\varphi_i(\vec{x}_k)$$

mit $\varphi_i(\vec{x})$: orthonormale Funktion

λ_i : reelle Zahl $\lambda_i \neq 0$, so dass $K(\vec{x},\vec{x}_k)$ beschränkt ist, d.h. $K \leq \text{const}$ für $\vec{x}\in\omega_1\cup\omega_2$

- Trenn-Hyperfläche $d(\vec{x})$
  - kumulative Erzeugung aus „Sequenz“ von Potentialfunktionen $K(\vec{x},\vec{x}_1),K(\vec{x},\vec{x}_2),\ldots$ gemäß der „sequentiellen“ Betrachtung bzw. Hinzunahme der Stichproben $\vec{x}_1,\vec{x}_2,\ldots$*

i) Schritt 1: Betrachtung von $\vec{x}_1$

$$K_0(\vec{x})=0$$

$$K_1(\vec{x})=\begin{cases}K_0(\vec{x})+K(\vec{x},\vec{x}_1), & \text{falls } \vec{x}_1\in\omega_1\\ K_0(\vec{x})-K(\vec{x},\vec{x}_1), & \text{falls } \vec{x}_1\in\omega_2\end{cases}$$

$$\Rightarrow K_1(\vec{x})=\left.\begin{cases}K(\vec{x},\vec{x}_1), & \text{falls } \vec{x}_1\in\omega_1\\ -K(\vec{x},\vec{x}_1), & \text{falls } \vec{x}_1\in\omega_2\end{cases}\right\}\text{ initiale Potentiale an } \vec{x}_1$$

---

*Vektoren bzw. Daten in vektorieller Repräsentation aus einer vorklassifizierten Stichprobe, d.h. überwachtes Lernen!

ii) Schritt 2: Betrachtung von $\vec{x}_2$

a) $K_2(\vec{x}) = K_1(\vec{x})$, falls $\left(\vec{x}_2 \in \omega_1 \wedge K_1(\vec{x}_2) > 0\right) \vee \left(\vec{x}_2 \in \omega_2 \wedge K_1(\vec{x}_2) < 0\right)$

b) falls $\left(\vec{x}_2 \in \omega_1 \wedge K_1(\vec{x}_2) \leq 0\right)$: $K_2(\vec{x}) = K_1(\vec{x}) + K(\vec{x}, \vec{x}_2)$

$= K(\vec{x}, \vec{x}_1) + K(\vec{x}, \vec{x}_2)$

c) falls $\left(\vec{x}_2 \in \omega_2 \wedge K_1(\vec{x}_2) \geq 0\right)$: $K_2(\vec{x}) = K_1(\vec{x}) - K(\vec{x}, \vec{x}_2)$

$= \pm K(\vec{x}, \vec{x}_1) - K(\vec{x}, \vec{x}_2)$

iii) Schritt 3: Betrachtung von $\vec{x}_3$

usw.

d.h. wenn $\vec{x}_k$ gemäß der durch $K_{k-1}(\vec{x})$ definierten Trenn-Hyperebene falsche Polarität hat+, wird $K_k(\vec{x})$ als kumuliertes Potential „modifiziert“ (abgesenkt bzw. angehoben).

$\Rightarrow$ allgemein: $d_{k+1}(\vec{x}) = d_k(\vec{x}) + r_{k+1} K(\vec{x}, \vec{x}_{k+1})$

$\equiv K_{k+1}(\vec{x}) = K_k(\vec{x}) + r_{k+1} K(\vec{x}, \vec{x}_{k+1})$

mit $r_{k+1} = \begin{cases} 0, & \text{für } \vec{x}_{k+1} \in \omega_1 \wedge K_k(\vec{x}_{k+1}) > 0 \\ 0, & \text{für } \vec{x}_{k+1} \in \omega_2 \wedge K_k(\vec{x}_{k+1}) < 0 \\ 1, & \text{für } \vec{x}_{k+1} \in \omega_1 \wedge K_k(\vec{x}_{k+1}) \leq 0 \\ -1, & \text{für } \vec{x}_{k+1} \in \omega_2 \wedge K_k(\vec{x}_{k+1}) \geq 0 \end{cases}$ d.h. $r_{k+1} = 0$ für korrekte Klassifikation (s. “reward & punishment“-Konzept)

(Hinweis: alternative rekursive Form $d_{k+1}(\vec{x}) = \sum_{i=1}^{\infty} c_i(k+1)\varphi_i(\vec{x}) = K_{k+1}(\vec{x})$)

+ Fall der Fehlklassifikation!

- Wahl der Potentialfunktion
  - „praktischer“ Ansatz: endliche Reihe orthonormaler Funktionen φ_i

$$K(\vec{x},\vec{x}_k)=\sum_{i=1}^{m}\varphi_i(\vec{x})\varphi_i(\vec{x}_k)$$

→ Typ 1–Potentialfunktionen, z.B. Hermite-Polynome:

$$\mathrm{H}_0(x)=1; \qquad \mathrm{H}_1(x)=2x;$$
$$\mathrm{H}_2(x)=4x^2-2; \qquad \mathrm{H}_3(x)=8x^3-12x;$$
$$H_4(x)=16x^4-48x^2+12; \quad \text{etc.}$$

mit $m=4$: $\varphi_1(\vec{x})=\varphi_1(x_1,x_2)=\mathrm{H}_0(x_1)\mathrm{H}_0(x_2)=1$

$$\varphi_2(\vec{x})=\varphi_2(x_1,x_2)=\mathrm{H}_1(x_1)\mathrm{H}_0(x_2)=2x_1$$

$$\varphi_3(\vec{x})=\varphi_3(x_1,x_2)=\mathrm{H}_0(x_1)\mathrm{H}_1(x_2)=2x_2$$

$$\varphi_4(\vec{x})=\varphi_4(x_1,x_2)=\mathrm{H}_1(x_1)\mathrm{H}_1(x_2)=4x_1x_2$$

$$\Rightarrow K(\vec{x},\vec{x}_k)=1+4x_1x_{k_1}+4x_2x_{k_2}+16x_1x_2x_{k_1}x_{k_2}$$

damit: finale Form von $d(\vec{x})$ ist vorgegeben durch (z.B. quadratische) Potentialfunktion

- zweiter „praktischer“ Ansatz:
  symmetrische Funktion zweier Variablen $\vec{x}$ und $\vec{x}_k$ (siehe Fig 5.4 und 5.5)

z.B. $K(\vec{x},\vec{x}_k) = \exp\left\{-\alpha \left\|\vec{x}-\vec{x}_k\right\|^2\right\}$

$$K(\vec{x},\vec{x}_k) = \frac{1}{1+\alpha \left\|\vec{x}-\vec{x}_k\right\|^2}$$

$$K(\vec{x},\vec{x}_k) = \left|\frac{\sin \alpha \left\|\vec{x}-\vec{x}_k\right\|^2}{\alpha \left\|\vec{x}-\vec{x}_k\right\|^2}\right|$$

→ Typ 2–Potentialfunktionen, z.B. mit $\alpha = 1$

$$K(\vec{x},\vec{x}_k) = \exp\left\{-\left\|\vec{x}-\vec{x}_k\right\|^2\right\}$$

$$\updownarrow$$

$$K(\vec{x},\vec{x}_k) = \exp-\left[\left(\vec{x}_1-\vec{x}_{k_1}\right)^2+\left(\vec{x}_2-\vec{x}_{k_2}\right)^2\right]$$ (Fig 5.7)

Hinweis: finale Form von $d(\vec{x})$ abhängig von Zahl der „Modifikationen“ des kumulierten Potentials (im Gegensatz zu Typ 1–Potentialfunktionen)

Beispiele für Potentialfunktionen

$$K(\mathbf{x}, \mathbf{x}_k) = \exp\{-\alpha||\mathbf{x} - \mathbf{x}_k||^2\} \qquad (5.6\text{–}25)$$

$$K(\mathbf{x}, \mathbf{x}_k) = \frac{1}{1 + \alpha||\mathbf{x} - \mathbf{x}_k||^2} \qquad (5.6\text{–}26)$$

$$K(\mathbf{x}, \mathbf{x}_k) = \left| \frac{\sin \alpha||\mathbf{x} - \mathbf{x}_k||^2}{\alpha||\mathbf{x} - \mathbf{x}_k||^2} \right| \qquad (5.6\text{–}27)$$

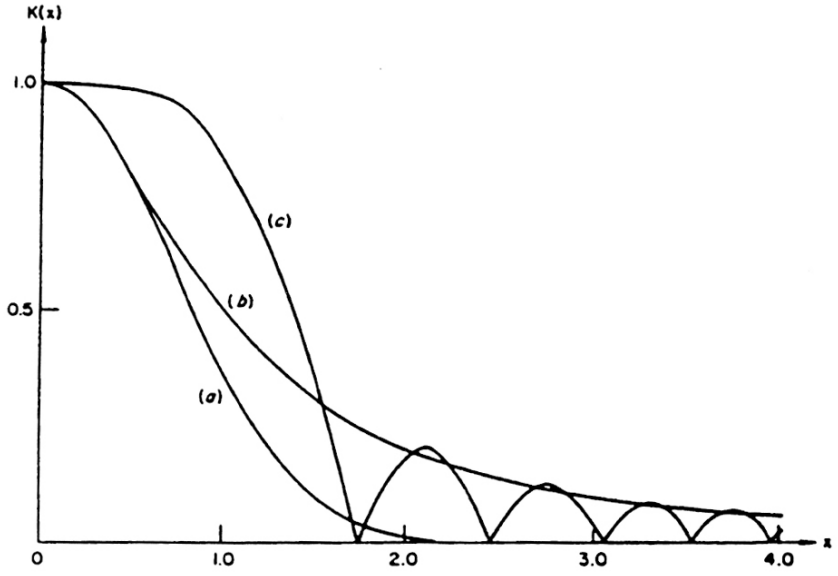


Figure 5.4. Examples of one-dimensional potential functions: (a) plot of Eq. (5.6–25); (b) plot of Eq. (5.6–26); (c) plot of Eq. (5.6–27). In all three cases $\alpha = 1$ and $x_k = 0$

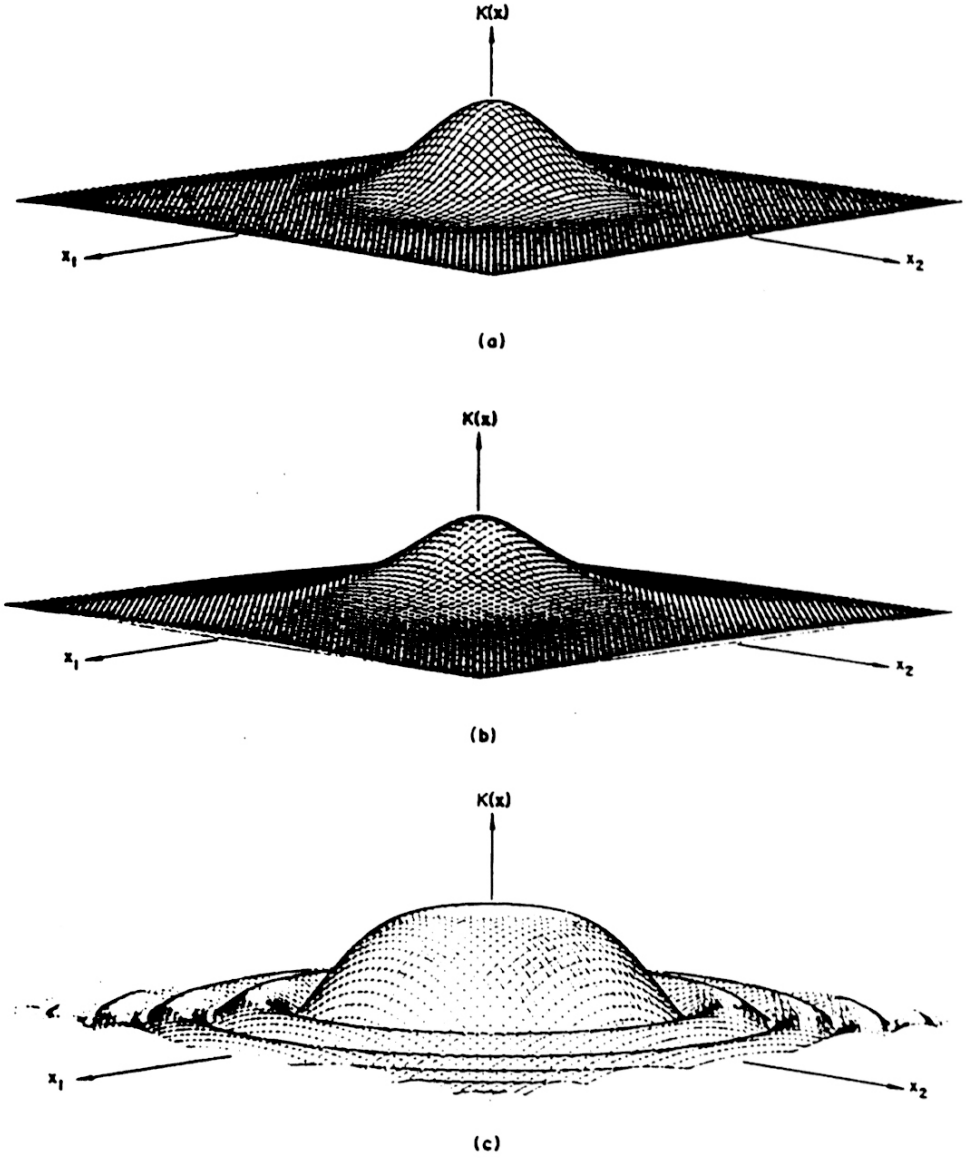
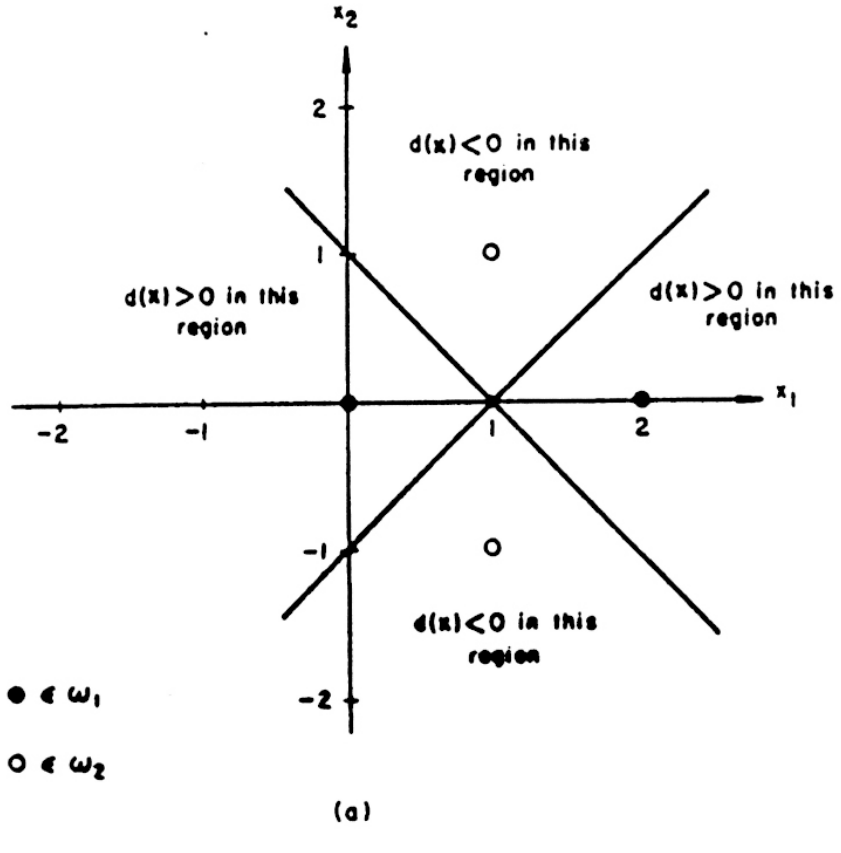


Figure 5.5. Two-dimensional potential functions: (a) plot of Eq. (5.6–25); (b) plot of Eq. (5.6–26); (c) plot of Eq. (5.6–27). In all three cases the range on the coordinates of $\mathbf{x} = (x_1, x_2)'$ is from -3 to 3, $\alpha = 1$, and $\mathbf{x}_k = 0$

(aus Tou & Gonzalez, 1974, chapter 5, pp. 194-195)



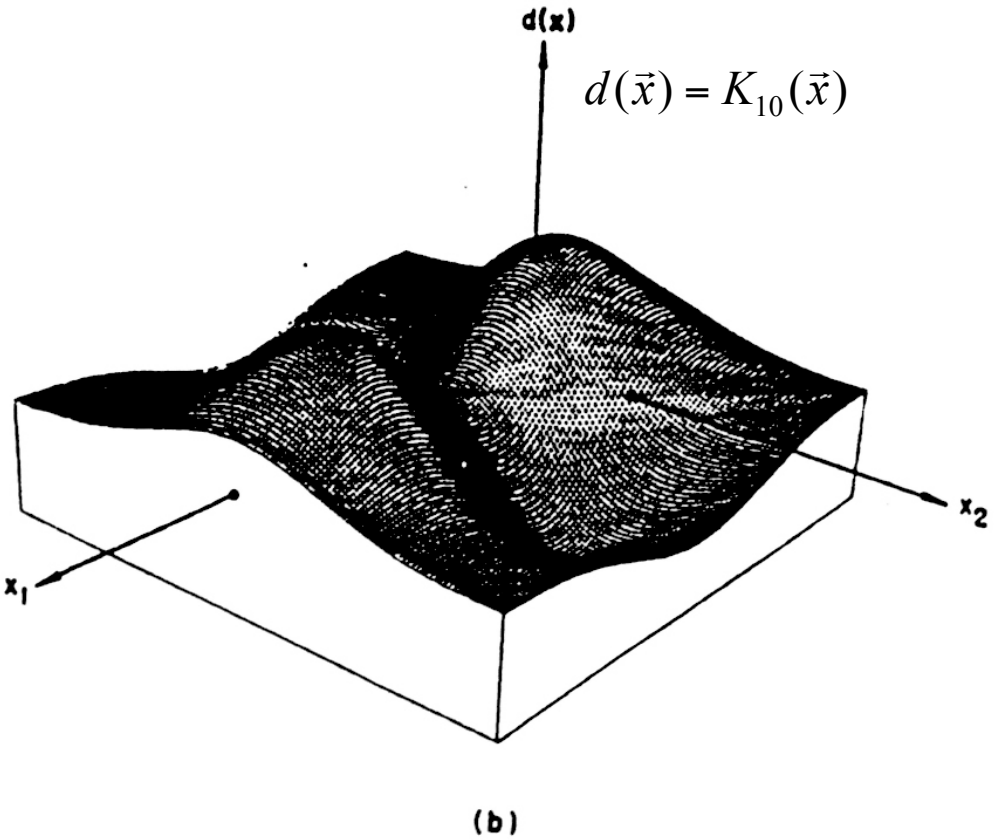


Figure 5.7. Patterns used in illustrating the potential function algorithm. (a) Patterns and decision surface. (b) Plot of $d(\mathbf{x})$ in the range $-1 \leqslant x_1 \leqslant 3$ and $-2 \leqslant x_2 \leqslant 2$

$$\text{für } K(\vec{x},\vec{x}_k) = \exp\left\{-\left\|\vec{x}-\vec{x}_k\right\|^2\right\}$$

$$= \exp-\left\{\left(x_1 - x_{k_1}\right)^2 + \left(x_2 - x_{k_2}\right)^2\right\}$$

(aus Tou & Gonzalez, 1974, chapter 5, p. 200)

- **Konklusio** (deterministische Lernalgorithmen)
  - Performanzüberprüfung nur durch “trial & error“
  - gewisse Zahl von Fehlklassifikationen muss toleriert werden in praxi
  - algorithmisch einfacher als lineare Programmierung