

# 5. Klassifikation

---

## 5.6 Support Vector Machines (SVM)

übernommen von

Stefan Rüping, Katharina Morik, Universität Dortmund

Vorlesung Maschinelles Lernen und Data Mining, WS 2002/03

und

Katharina Morik, Claus Weihs, Universität Dortmund

Wissensentdeckung in Datenbanken, SS 2006

# Funktionslernen

Gegeben:

Beispiele  $X$  in  $LE$

- die anhand einer Wahrscheinlichkeitsverteilung  $P$  auf  $X$  erzeugt wurden und
- mit einem Funktionswert  $Y = t(X)$  versehen sind (alternativ: Eine Wahrscheinlichkeitsverteilung  $P(Y|X)$  der möglichen Funktionswerte - verrauschte Daten).

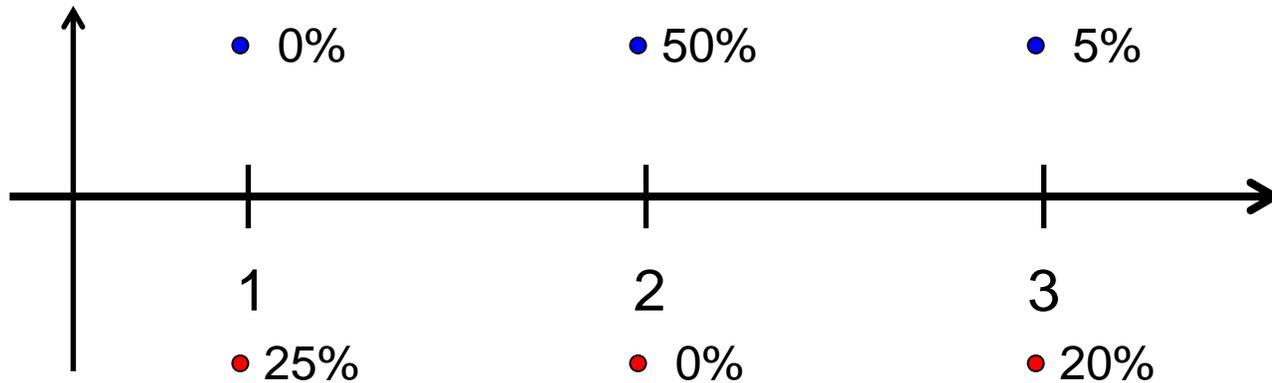
$H$  die Menge von Funktionen in  $LH$ .

Ziel: Eine Hypothese  $h(X) \in H$ , die das erwartete Fehlerrisiko  $R(h)$  minimiert.

Risiko:

$$R(h) = \sum_x Q(x, h)P(x)$$

# Beispiel: Funktionenlernen



$$H = \{ f_a \mid f_a(x) = 1, \text{ für } x \geq a, f_a(x) = -1 \text{ sonst, } a \in \mathcal{R} \}$$

$$R(f_0) = 0,25 + 0 + 0,20 = 0,45$$

$$R(f_{1,5}) = 0 + 0 + 0,20 = 0,20$$

$$R(f_{3,5}) = 0 + 0,5 + 0,05 = 0,55$$

# Reale Beispiele

**Klassifikation:**  $Q(\mathbf{x},\mathbf{h}) = 0$ , falls  $t(\mathbf{x}) = \mathbf{h}(\mathbf{x})$ , **1** sonst

- Textklassifikation ( $x$  = Worthäufigkeiten)
- Handschriftenerkennung ( $x$  = Pixel in Bild)
- Vibrationsanalyse in Triebwerken ( $x$  = Frequenzen)
- Intensivmedizinische Alarmfunktion ( $x$  = Vitalzeichen)

**Regression:**  $Q(\mathbf{x},\mathbf{h}) = (t(\mathbf{x})-\mathbf{h}(\mathbf{x}))^2$

- Zeitreihenprognose ( $x$  = Zeitreihe,  $t(x)$  = nächster Wert)

# Erinnerung: Minimierung des beobachteten Fehlers

---

Funktionslernaufgabe nicht direkt lösbar. Problem:

- Die tatsächliche Funktion  $t(X)$  ist unbekannt.
- Die zugrunde liegende Wahrscheinlichkeit ist unbekannt.

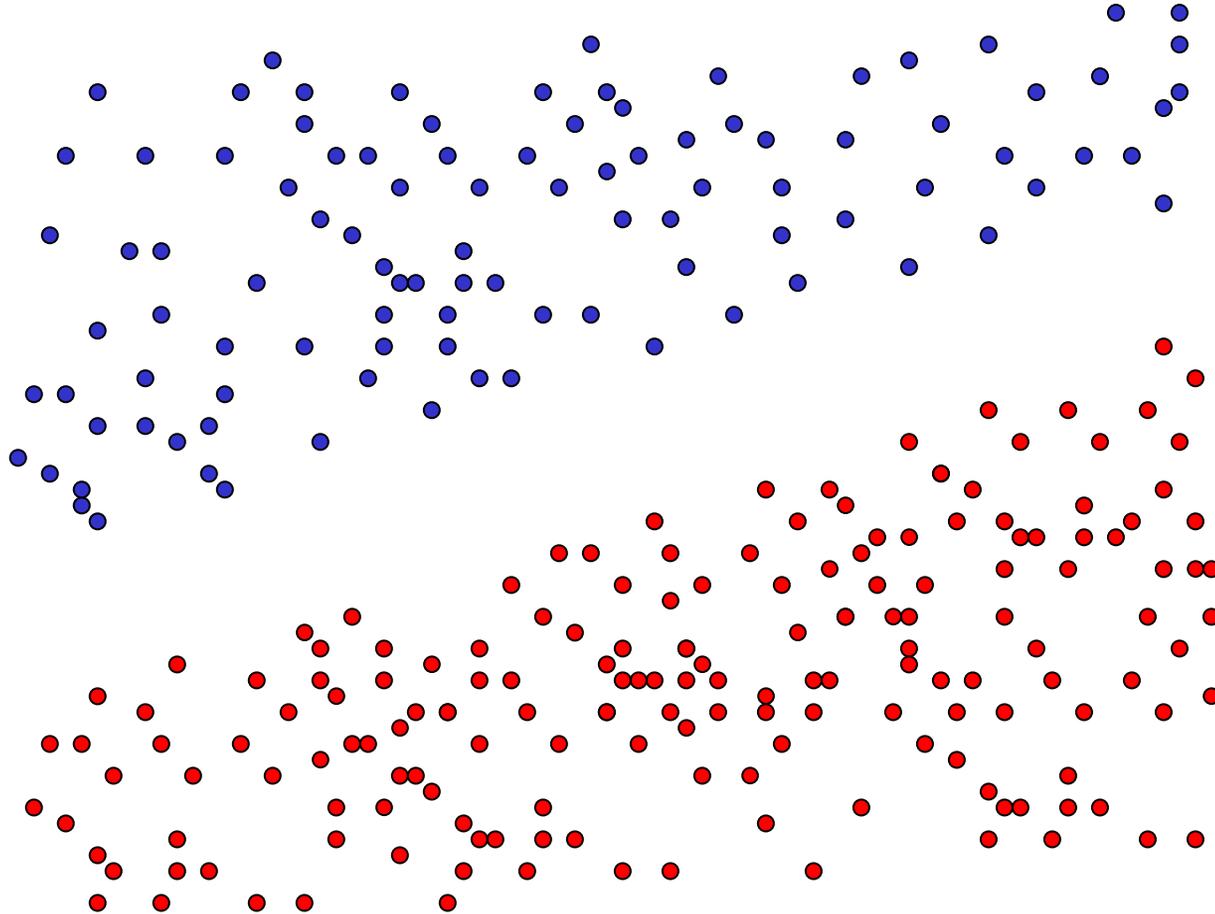
Ansatz:

- eine hinreichend große Lernmenge nehmen und für diese den Fehler minimieren.

⇒ Empirical Risk Minimization

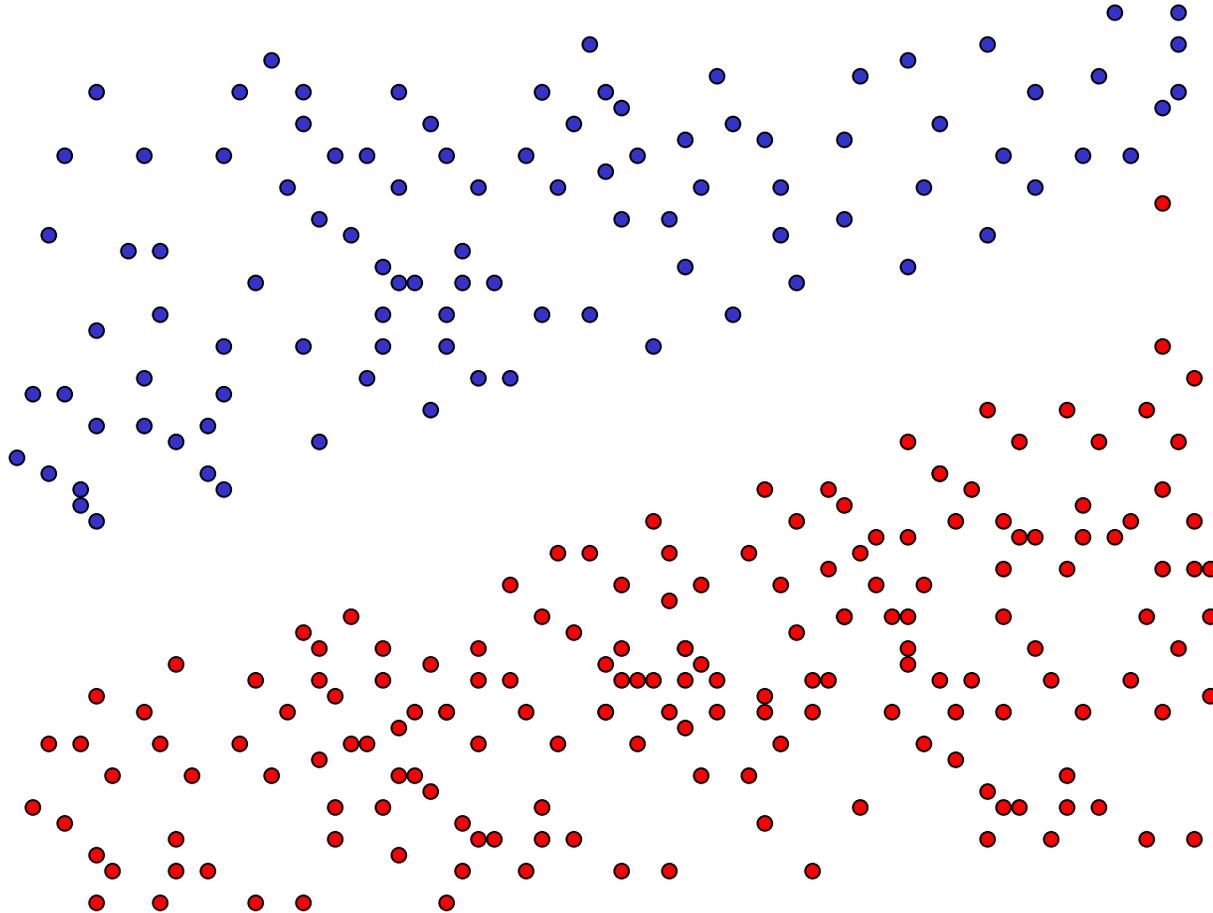
# Beispiel

---



# Beispiel II

---



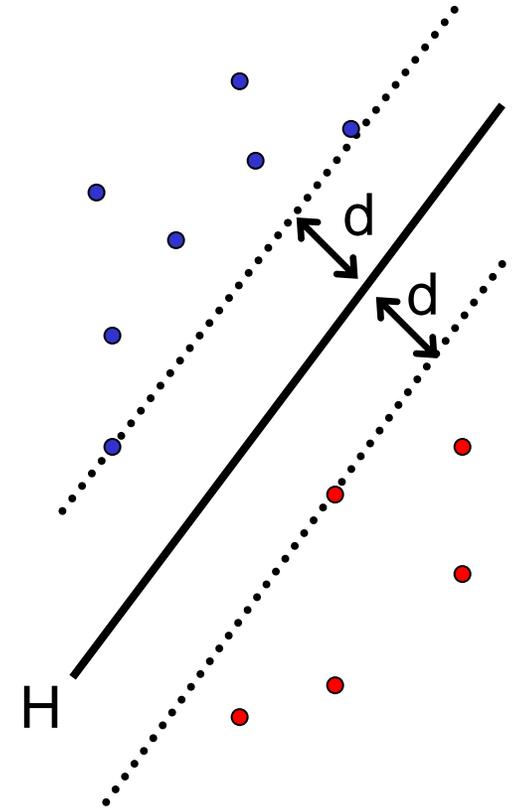
# Probleme der ERM

---

- Aufgabe ist nicht eindeutig beschrieben: Mehrere Funktionen mit minimalem Fehler existieren. Welche wählen?
- Overfitting: Verrauschte Daten und zu wenig Beispiele führen zu falschen Ergebnissen.

# Die optimale Hyperebene

- Beispiele heißen linear trennbar, wenn es eine Hyperebene  $H$  gibt, die die positiven und negativen Beispiele voneinander trennt.
- $H$  heißt optimale Hyperebene, wenn ihr Abstand  $d$  zum nächsten positiven und zum nächsten negativen Beispiel maximal ist.
- Satz: Es existiert eine eindeutig bestimmte optimale Hyperebene.

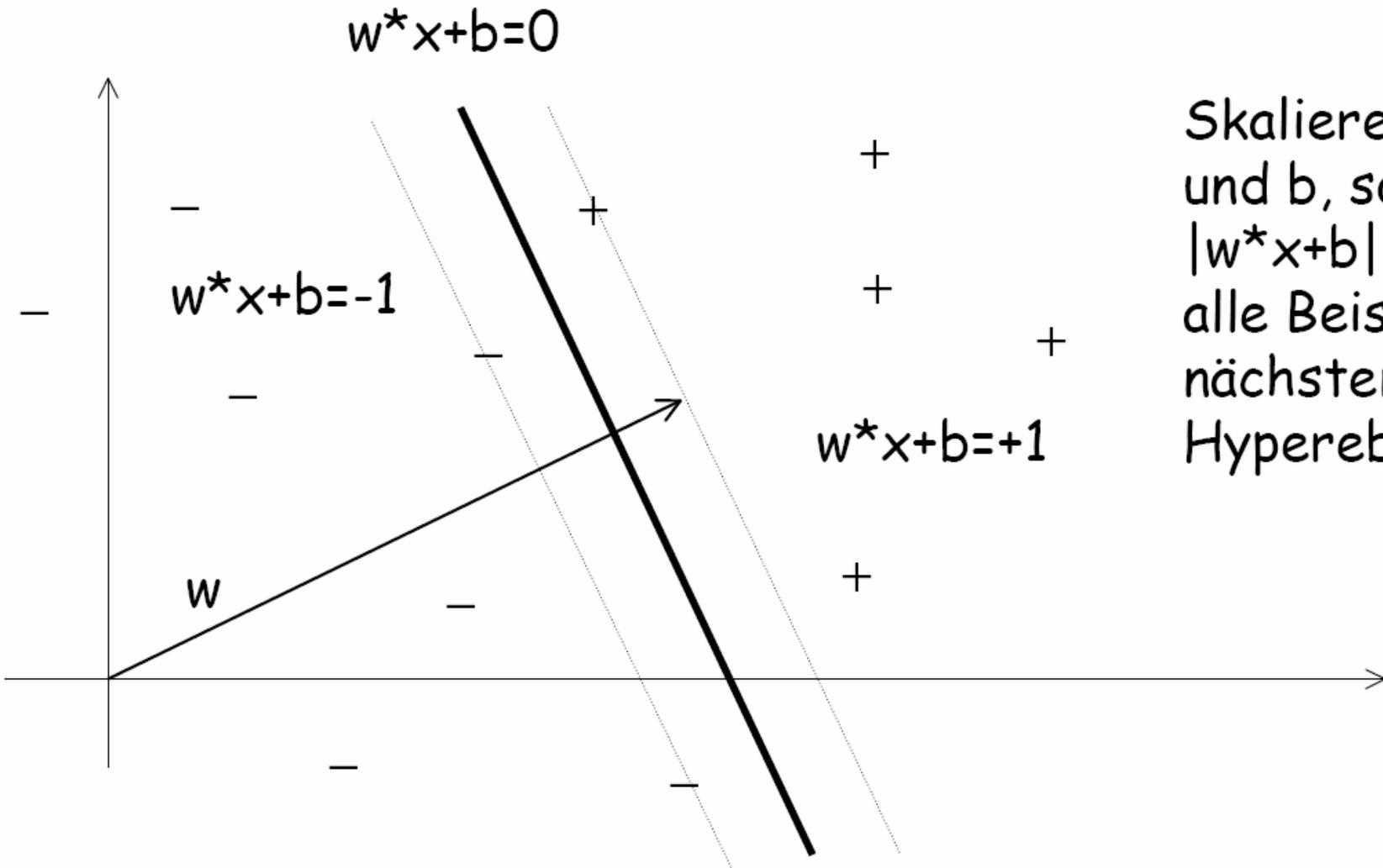


# Grundbegriffe II

- Der Normalenvektor steht senkrecht auf allen Vektoren der Hyperebene. Es gilt:

$$w^* x + b \begin{cases} > 0 \text{ falls } x \text{ im positiven Raum} \\ = 0 \text{ falls } x \text{ auf } H \\ < 0 \text{ falls } x \text{ im negativen Raum} \end{cases}$$

# Bild



Skalieren von  $w$  und  $b$ , so dass  $|w^*x+b|=1$  für alle Beispiele am nächsten zur Hyperebene.

# Separierende Hyperebene

- Beispiele in Form von Vektoren  $x$  aus  $\mathcal{R}^p$  und Klassifikation  $y=+1$  (positive Beispiele) oder  $y=-1$  (negative Beispiele)  
 $E = \{ [x_1, y_1], [x_2, y_2], \dots, [x_m, y_m] \}$
- Separierende Hyperebene  $H$ :  
positive Beispiele im positiven Halbraum,  
negative Beispiele im negativen Halbraum,  
 $x^*w + b = 0$  für Punkte auf der Hyperebene.
- Der Abstand von  $H$  zum Ursprung ist  $b / \|w\|$
- Die Separierbarkeit erfüllen viele Hyperebenen.

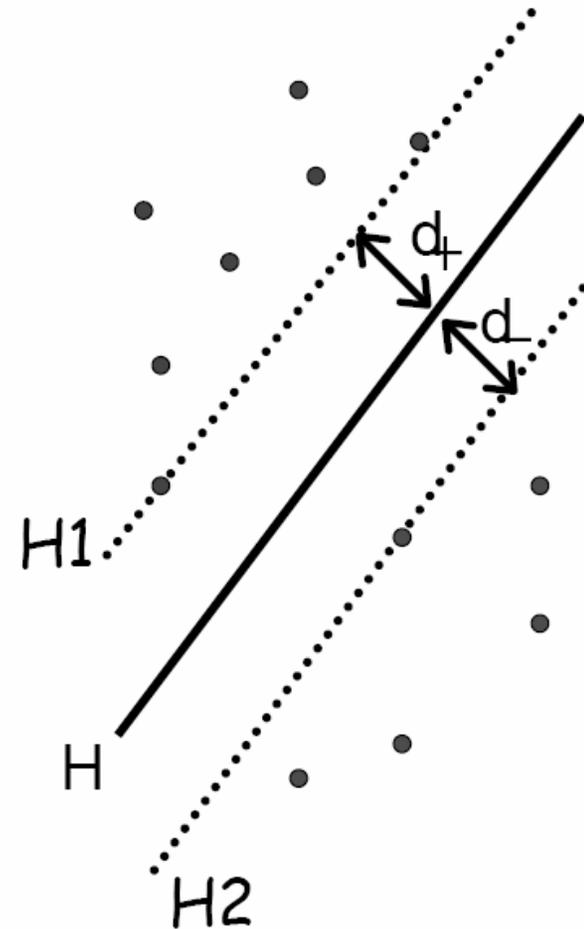
# Margin für separierbare Beispiele

- Abstand  $d_+$  von  $H$  zum nächsten positiven Beispiel
- Abstand  $d_-$  von  $H$  zum nächsten negativen Beispiel
- Margin:  $d_+ + d_-$
- $H_1$   $x_i * w + b \geq +1$  bei  $y_i = +1$
- $H_2$   $x_i * w + b \leq -1$  bei  $y_i = -1$
- zusammengefasst:  $\forall x_i : y_i (w * x_i + b) - 1 > 0$
- Der Abstand von  $H_1$  zum Ursprung ist  $|1-b| / ||w||$
- Der Abstand von  $H_2$  zum Ursprung ist  $|-1-b| / ||w||$
- $d_+ = d_- = 1 / ||w||$  und margin =  $2 / ||w||$

# Margin

- $H_1$  und  $H_2$  sind parallel, haben denselben Normalenvektor  $w$ .
- Per Konstruktion liegt kein Beispiel zwischen  $H_1$  und  $H_2$ .
- Um  $2 / ||w||$  zu maximieren, müssen wir  $||w||$  minimieren.
- Die Nebenbedingungen müssen eingehalten werden:

$$\forall i: y_i(x_i * w + b) - 1 \geq 0$$

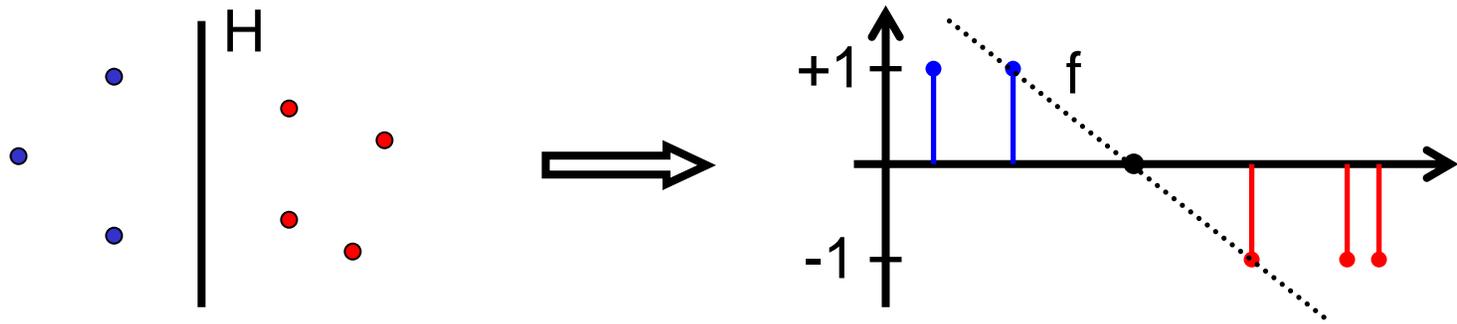


# Minimieren der Länge

- Um die geometrische Breite  $\frac{1}{\|w\|}$  zu maximieren, müssen wir die Länge von  $w$  minimieren.  
Wir können genauso gut  $w^*w$  minimieren.
- So finden wir nun eine eindeutige Hyperebene aus den vielen möglichen trennenden.
- Für alle Beispiele ist sie richtig:  $f(x_i) > 0$  gdw.  $y_i > 0$
- Wir können sie anwenden, um neue unklassifizierte Beobachtungen zu klassifizieren:  
 $f(x) = w^*x + b$   
das Vorzeichen gibt die Klasse an.

# Berechnung der opt. Hyperebene

- Hyperebene  
 $H = \{x \mid w^*x + b = 0\}$
- $H$  trennt  $(x_i, y_i)$ ,  $y_i \in \{\pm 1\}$
- $H$  ist optimale Hyperebene
- Entscheidungsfunktion  $f(x) = w^*x + b$
- $f(x_i) > 0 \Leftrightarrow y_i > 0$
- $\|w\|$  minimal und  
 $f(x_i) \geq 1$ , wenn  $y_i = 1$   
 $f(x_i) \leq -1$ , wenn  $y_i = -1$



# Optimierungsaufgabe der SVM

- Minimiere  $\|w\|^2$
- so dass für alle  $i$  gilt:  
 $f(x_i) = w^*x_i + b \geq 1$  für  $y_i = 1$  und  
 $f(x_i) = w^*x_i + b \leq -1$  für  $y_i = -1$
- Äquivalente Nebenbedingung:  $y_i * f(x_i) \geq 1$
- Konvexes, quadratisches Optimierungsproblem  $\Rightarrow$  eindeutig in  $O(n^3)$  lösbar.
- Satz:  $\|w\| = 1/d$ ,  $d =$  Abstand der optimalen Hyperebene zu den Beispielen.

# Lagrange-Funktion

- Sei das Optimierungsproblem gegeben,  $f(w)$  zu minimieren unter der Nebenbedingung  $g_i(w) \geq 0$   $i=1, \dots, m$ , dann ist die Lagrange-Funktion

$$L(w, \alpha) = f(w) - \sum_{i=1}^m \alpha_i g_i(w)$$

- Dabei muss gelten  $\alpha_i \geq 0$
- Für Ungleichheitsbedingungen werden  $\alpha$ -Multiplikatoren eingeführt, Gleichheitsbedingungen werden direkt eingesetzt.
- Es ist leichter, Vektor  $\alpha$  zu bestimmen, als direkt nach der Erfüllung der Bedingungen zu suchen.

# Optimierungsfunktion als Lagrange

- Minimiere  $L(w, b, \alpha)$ !

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (y_i (x_i * w + b) - 1)$$

- Eine optimale Lösung zeichnet sich durch die folgenden notwendigen Bedingungen an  $\alpha$  aus:

$$w = \sum_{i=1}^m \alpha_i y_i x_i \quad \sum_{i=1}^m \alpha_i y_i = 0$$

- $L$  soll bezüglich  $w$  und  $b$  minimiert, bezüglich  $\alpha$  maximiert werden.

# Karush-Kuhn-Tucker Bedingungen

- Für das primale Optimierungsproblem gelten die KKT Bedingungen gdw.  $w, b, \alpha$  die Lösung ist.

$$\frac{\partial}{\partial w_v} L(w, b, \alpha) = w_v - \sum_i \alpha_i y_i x_{i,v} = 0 \quad v = 1, \dots, d$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = - \sum_i \alpha_i y_i = 0$$

$$y_i (x_i * w + b) - 1 \geq 0$$

$$\forall i : \alpha_i \geq 0$$

$$\forall i : \alpha_i (y_i (w * x_i + b) - 1) = 0$$

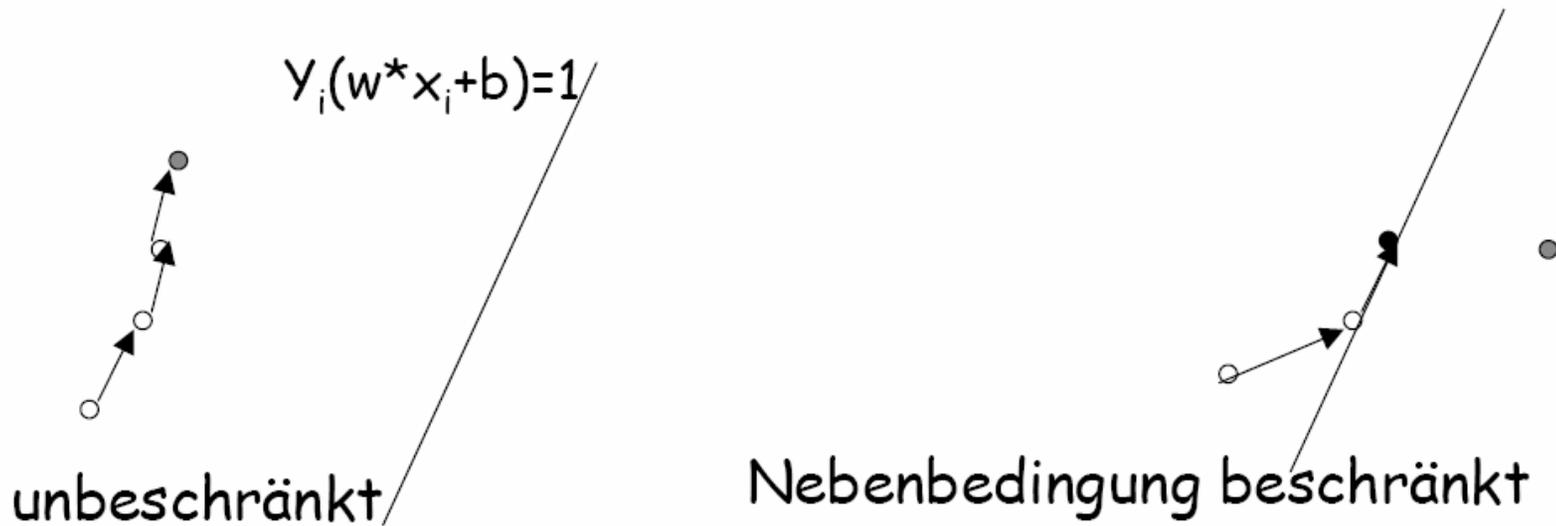
$i$  Beispiele,  $v$  Attribute der Beispiele=Komponenten der Vektoren

# Duales Problem

- Die Gleichheitsbedingungen werden in  $L(w,b,\alpha)$  eingesetzt.
- Der duale Lagrange-Ausdruck  $L(\alpha)$  soll maximiert werden.
- Das Minimum des ursprünglichen Optimierungsproblems tritt genau bei jenen Werten von  $w,b,\alpha$  auf wie das Maximum des dualen Problems.

# Anschaulich?

- Wir wollen  $w$  minimieren, also  $\Delta w = 0$ , also Minimum von  $w$  in Richtung des Gradienten suchen.
- Die Nebenbedingungen sind entweder weit ab oder der auf ihnen liegende nächste Punkt zum Minimum gibt das Minimum unter Einhaltung der Nebenbedingungen an.



# Umformung

$$\begin{aligned} & \frac{1}{2} w^* w - \sum_{i=1}^m \alpha_i [y_i (x_i^* w + b) - 1] \\ = & \frac{1}{2} w^* w - \sum_{i=1}^m \alpha_i y_i (x_i^* w + b) + \sum_{i=1}^m \alpha_i \\ = & \frac{1}{2} w^* w - \sum_{i=1}^m \alpha_i y_i x_i^* w - \sum_{i=1}^m \alpha_i y_i b + \sum_{i=1}^m \alpha_i \\ = & \frac{1}{2} w^* w - \sum_{i=1}^m \alpha_i y_i x_i^* w + \sum_{i=1}^m \alpha_i \end{aligned}$$

Bei gutem  $\alpha$  muss gelten  $0 = \sum_{i=1}^m \alpha_i y_i$

# Umformung II

- Es gilt für optimalen Vektor  $\alpha$   $w = \sum_{i=1}^m \alpha_i y_i x_i$  wir ersetzen

$$\begin{aligned}
 & \frac{1}{2} w^* w && - \sum_{i=1}^m \alpha_i y_i x_i^* w && + \sum_{i=1}^m \alpha_i \\
 = & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^* x_j && - \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^* x_j && + \sum_{i=1}^m \alpha_i \\
 = & + \sum_{i=1}^m \alpha_i && - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^* x_j
 \end{aligned}$$

- Mit den Nebenbedingungen:

$$0 = \sum_{i=1}^m \alpha_i y_i \quad \text{und} \quad \alpha_i \geq 0$$

# SVM Optimierungsproblem

- Maximiere

unter  $0 \leq \alpha_i$  für alle  $i$  und  $\sum \alpha_i y_i = 0$

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j (x_i * x_j)$$

- Für jedes Beispiel gibt es ein  $\alpha$  in der Lösung.
  - $0 = \alpha_i$  heißt, dass das Beispiel  $x_i$  im passenden Halbraum liegt.
  - $0 < \alpha_i$  heißt, dass das Beispiel  $x_i$  auf  $H_1$  oder  $H_2$  liegt (Stützvektor).
- Es gilt  $w = \sum \alpha_i y_i x_i$ ,
  - Also  $f(x) = \sum \alpha_i y_i (x_i * x) + b$
  - Also ist der beste Normalenvektor  $w$  eine Linearkombination von Stützvektoren ( $\alpha_i \neq 0$ ).

# Optimierungsalgorithmus

```
s = Gradient von W( $\alpha$ ) //  $s_i = \sum \alpha_j (x_j * x_i)$ 
while(nicht konvergiert(s)) // auf  $\epsilon$  genau
    WS = working_set(s) // suche k „gute“ Variablen
     $\alpha'$  = optimiere(WS) // k neue  $\alpha$ -Werte
    s = update(s,  $\alpha'$ ) // s = Gradient von W( $\alpha'$ )
```

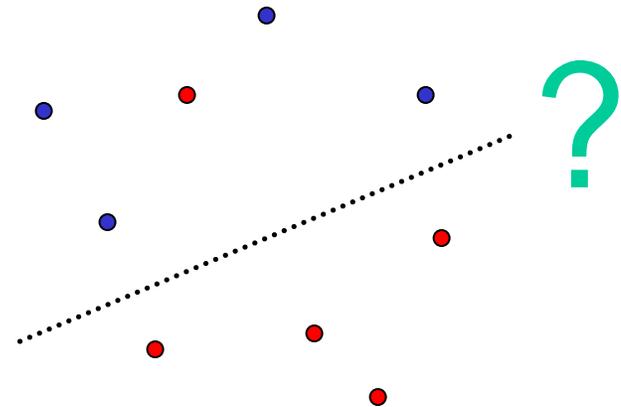
- Gradientensuchverfahren
- Trick: Stützvektoren allein definieren Lösung
- Weitere Tricks: Shrinking, Caching von  $x_i * x_j$

# Was wissen wir jetzt?

- Maximieren des Margins einer Hyperebene ergibt eine eindeutige Festlegung der optimalen trennenden Hyperebene.
- Dazu minimieren wir die Länge des Normalenvektors  $w$ .
  - Formulierung als Lagrange-Funktion
  - Formulierung als duales Optimierungsproblem
- Das Lernergebnis ist eine Linearkombination von Stützvektoren.
- Mit den Beispielen müssen wir nur noch das Skalarprodukt rechnen.

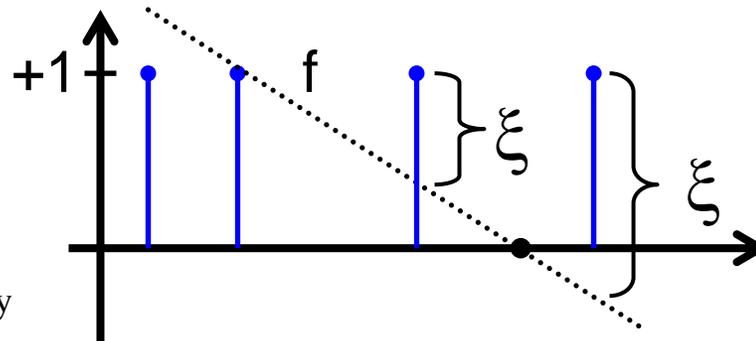
# Nicht linear trennbare Daten

- In der Praxis sind linear trennbare Daten selten.
- 1. Ansatz: Entferne eine minimale Menge von Datenpunkten, so dass die Daten linear trennbar werden (minimale Fehlklassifikation).
- Problem: Algorithmus wird exponentiell.

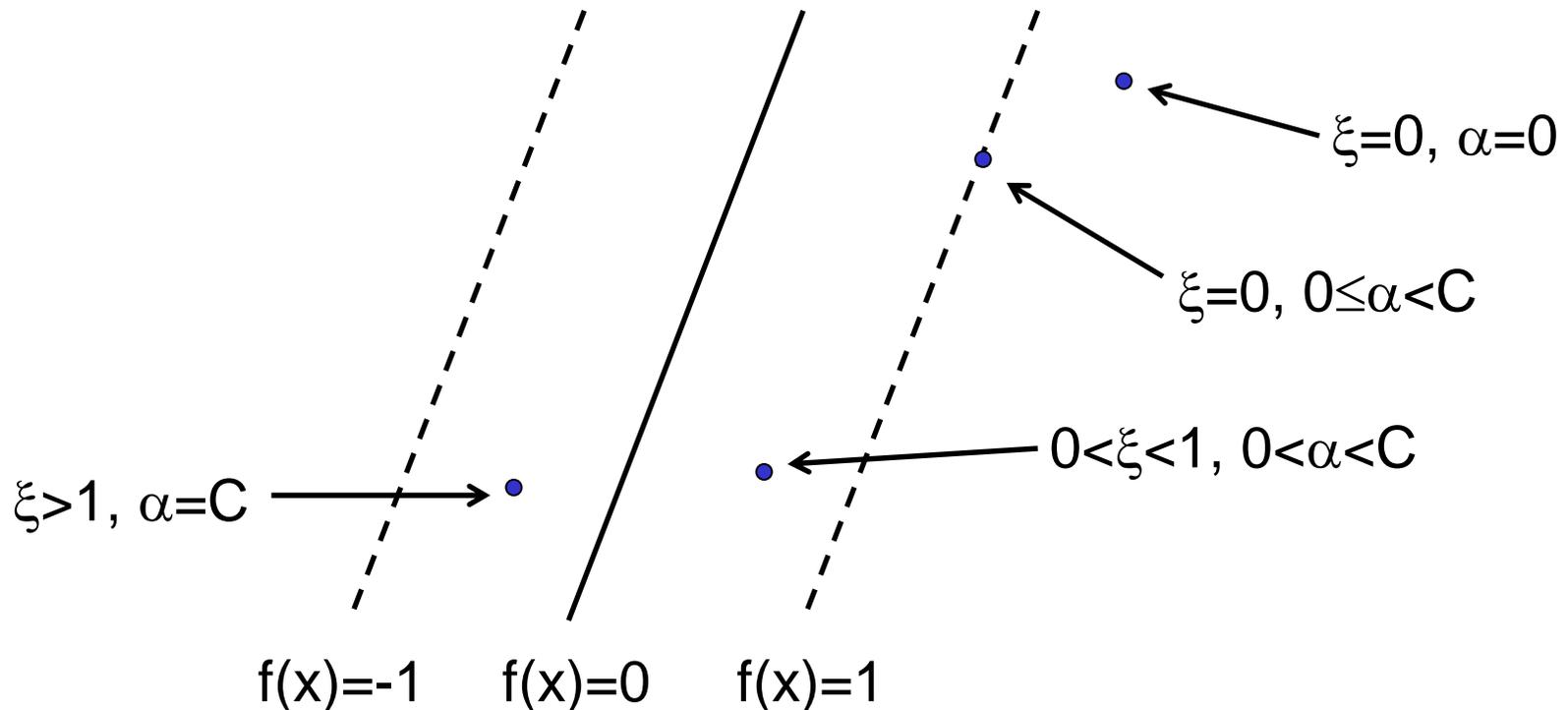


# Weich trennende Hyperebene

- Wähle  $C \in \mathcal{R}_{>0}$  und minimiere 
$$\|w\|^2 + C \sum_{i=1}^n \xi_i$$
- so dass für alle  $i$  gilt:  
 $f(x_i) = w^*x_i + b \geq 1 - \xi_i$  für  $y_i = 1$  und  
 $f(x_i) = w^*x_i + b \leq -1 + \xi_i$  für  $y_i = -1$
- Äquivalent:  $y_i * f(x_i) \geq 1 - \xi_i$



# Bedeutung von $\xi$ und $\alpha$



Beispiele  $x_i$  mit  $\alpha_i > 0$  heißen Stützvektoren  $\Rightarrow$  SVM

# Was wissen wir jetzt über SVM's?

---

- Funktionslernen als allgemeine Lernaufgabe
- Minimierung des empirischen Risikos als Lösungsstrategie
- Optimale Hyperebene präzisiert die ERM
- Praxis: weich trennende Hyperebene
- Berechnung mittels SVM und dualem Problem
- **Offene Fragen:**
  - Generelles Prinzip hinter der optimalen Hyperebene?
  - Nicht lineare Daten? → Kernel-Methoden