

## 5. Klassifikation

### 5.6 Support Vector Machines (SVM)

übernommen von

Stefan Rüping, Katharina Morik, Universität Dortmund

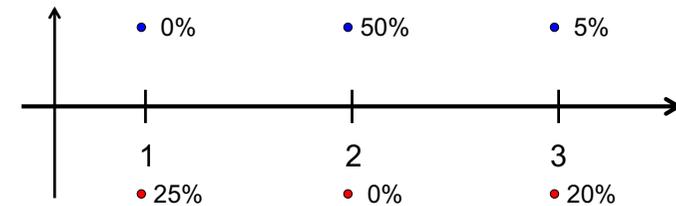
Vorlesung Maschinelles Lernen und Data Mining, WS 2002/03

und

Katharina Morik, Claus Weihs, Universität Dortmund

Wissensentdeckung in Datenbanken, SS 2006

## Beispiel: Funktionenlernen



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

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

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

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

## Funktionslernen

Gegeben:

Beispiele  $X$  in  $L$

- 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:

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

## Reale Beispiele

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

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

**Regression:**  $Q(x, h) = (t(x) - h(x))^2$

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

## Erinnerung: Minimierung des beobachteten Fehlers

Funktionslernaufgabe nicht direkt lösbar. Problem:

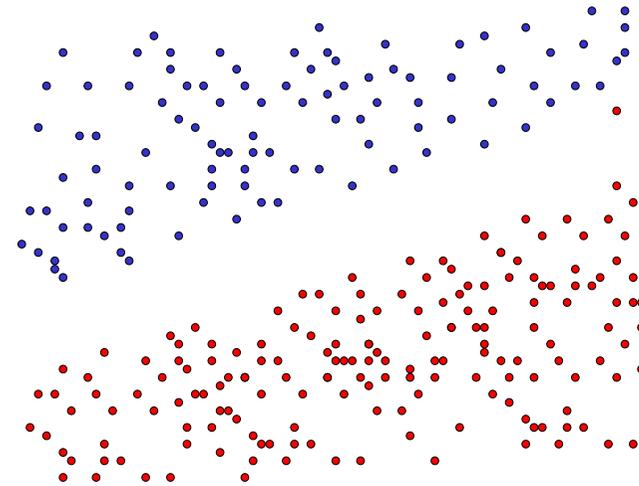
- Die tatsächliche Funktion  $t(X)$  ist unbekannt.
- Die zugrunde liegende Wahrscheinlichkeit ist unbekannt.

Ansatz:

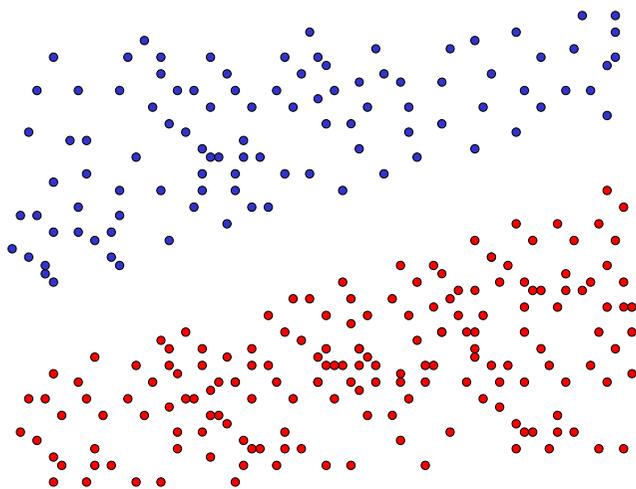
- eine hinreichend große Lernmenge nehmen und für diese den Fehler minimieren.

⇒ Empirical Risk Minimization

## Beispiel II



## Beispiel

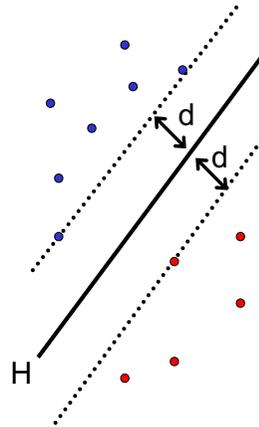


## Probleme der ERM

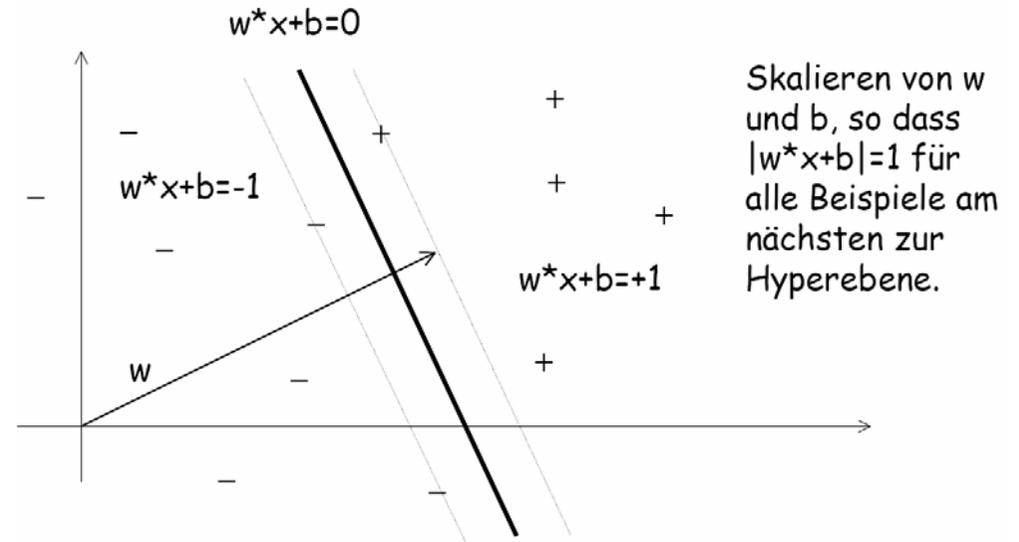
- Aufgabe ist nicht eindeutig beschrieben: Mehrere Funktionen mit minimalem Fehler existieren. Welche wählen?
- Overfitting: Verrauschte Daten und zu wenig Beispiele führen zu falschen Ergebnissen.

## 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.



## Bild



## Grundbegriffe II

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

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

## Separierende Hyperebene

- Beispiele in Form von Vektoren  $x$  aus  $\mathbb{R}^p$  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.

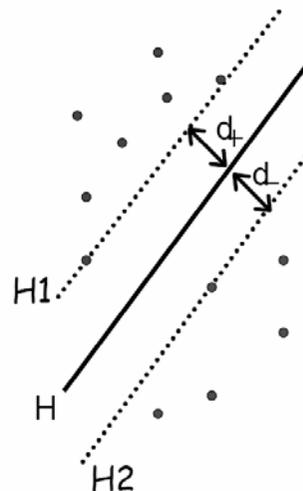
# Margin für separierbare Beispiele

- Abstand  $d_+$  von H zum nächsten positiven Beispiel
- Abstand  $d_-$  von H zum nächsten negativen Beispiel
- Margin:  $d_+ + d_-$
- H1  $x_i * w + b \geq +1$  bei  $y_i = +1$
- H2  $x_i * w + b \leq -1$  bei  $y_i = -1$
- zusammengefasst:  $\forall x_i : y_i (w * x_i + b) - 1 > 0$
- Der Abstand von H1 zum Ursprung ist  $|1-b| / ||w||$
- Der Abstand von H2 zum Ursprung ist  $|-1-b| / ||w||$
- $d_+ = d_- = 1 / ||w||$  und margin =  $2 / ||w||$

## Margin

- H1 und H2 sind parallel, haben denselben Normalenvektor  $w$ .
- Per Konstruktion liegt kein Beispiel zwischen H1 und H2.
- Um  $2 / ||w||$  zu maximieren, müssen wir  $||w||$  minimieren.
- Die Nebenbedingungen müssen eingehalten werden:

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

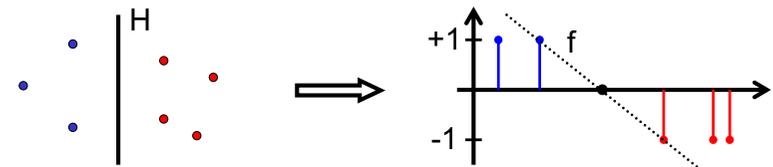


# Minimieren der Länge

- 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.

# 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:

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

# 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  $0 = \sum_{i=1}^m \alpha_i y_i$

# 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(α)           // s_i = ∑α_j(x_j*x_i)
while(nicht konvergiert(s))     // auf ε genau
    WS = working_set(s)         // suche k „gute“ Variablen
    α' = optimiere(WS)          // k neue α-Werte
    s = update(s, α')           // s = Gradient von W(α')
    
```

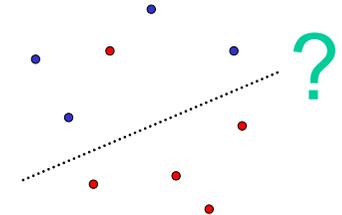
- Gradientensuchverfahren
- Trick: Stützvektoren allein definieren Lösung
- Weitere Tricks: Shrinking, Caching von  $x_i * x_j$

## 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.

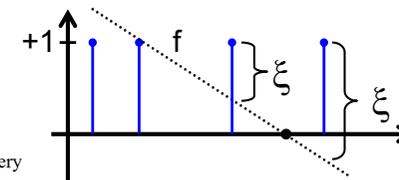


## Weich trennende Hyperebene

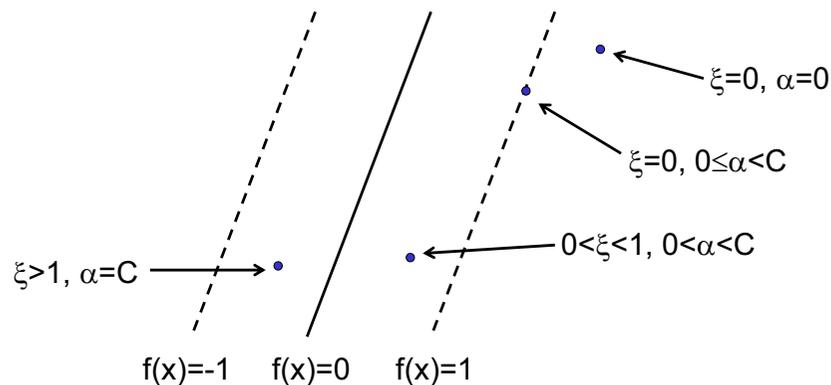
Praktikable Lösung, wenn Daten „fast“ linear separabel sind:

- Wähle  $C \in \mathbb{R}_{>0}$  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

## Kernel-Methoden

- Problem: Daten sind oft nicht linear trennbar.
- Lösungsidee:
  1. Nicht-lineare Einbettung in einen höher-dimensionalen Raum
  2. Dort Suchen der optimal trennenden Hyperebene
  3. Rücktransformation

## Kernel-Methoden

Beispiel:

$$\phi: \mathbf{R}^3 \rightarrow \mathbf{R}^6 \text{ mit } \phi_1(\mathbf{x})=x_1, \phi_2(\mathbf{x})=x_2, \phi_3(\mathbf{x})=x_3, \\ \phi_4(\mathbf{x})=(x_1)^2, \phi_5(\mathbf{x})=x_1 x_2, \phi_6(\mathbf{x})=x_1 x_3, \\ \text{für } \mathbf{x}=(x_1, x_2, x_3) \in \mathbf{R}^3.$$

Eine trennende Hyperebene im  $\mathbf{R}^6$  hat die Form  $d(\mathbf{z}) = \mathbf{w}\mathbf{z} + b$ .

Wir lösen das Problem für  $\mathbf{w}$  und  $b$  und übersetzen die Lösung zurück:

$$d(\mathbf{z}) = w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + b \\ = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 (x_1)^2 + w_5 x_1 x_2 + w_6 x_1 x_3 + b$$

## Kernel-Methoden

Probleme:

- Welche Einbettung verwenden?
- Das Skalarprodukt im hochdimensionalen Raum ist teuer.

Lösung: Kernel-Funktion:

- Die Trainingsdaten tauchen nur in der Form  $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$  auf.
- Eine Kernelfunktion  $K(\mathbf{x}_i, \mathbf{x}_j)$  ist eine Funktion mit  $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ .
- Sie kann im niedrigdimensionalen Raum berechnet werden!

Typische Kernelfunktionen:

- $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^h$ , für  $h \in \mathbf{N}$
- $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$
- $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j - \delta)$