

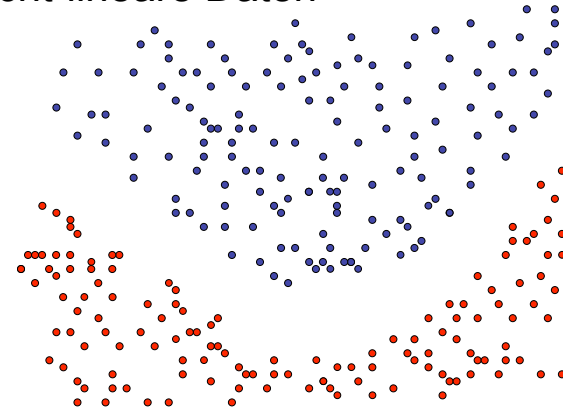


# Kernfunktionen

Wie funktioniert der Kern-Trick?  
 Wann funktioniert der Kern-Trick?  
 Warum?

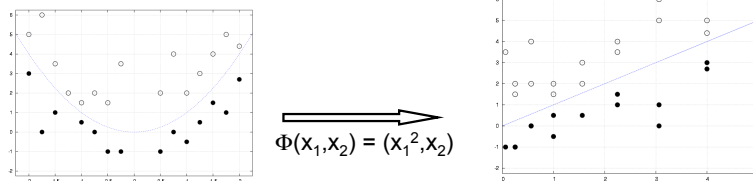


# Nicht-lineare Daten

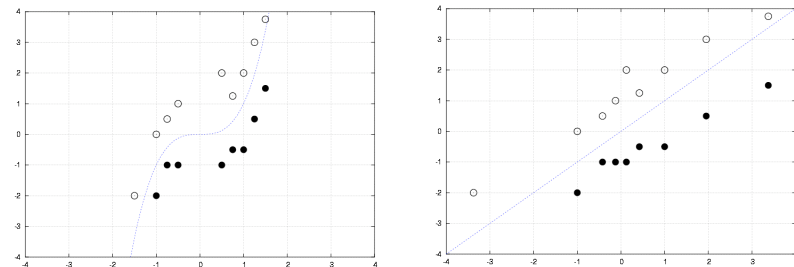


# Nicht-lineare Daten

- Neue SVM-Theorie entwickeln? (Neeee!)
- Lineare SVM benutzen? („If all you’ve got is a hammer, every problem looks like a nail“)
- Transformation in lineares Problem!



# Beispiel



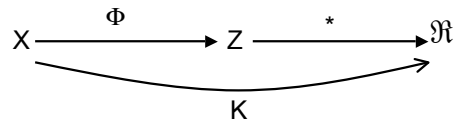
$$\Phi(x_1, x_2) = (x_1^3, x_2)$$



## Kernfunktionen

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

$$f(x) = \sum \alpha_i y_i (x_i * x) + b$$
- SVM hängt von x nur über Skalarprodukt  $x * x'$  ab.
- Ersetze Transformation  $\Phi$  und Skalarprodukt  $*$  durch Kernfunktion  $K(x_1, x_2) = \Phi(x_1) * \Phi(x_2)$



## Der Kern-Trick

- $K(x, x') = x * x'$
- $\Phi: X \rightarrow \mathcal{H}$
- Das Skalarprodukt der Vektoren im Merkmalsraum  $\mathcal{H}$  entspricht dem Wert der Kernfunktion über den Beispielen.
- Welche Funktionen machen  $k(x, x') = \Phi(x) * \Phi(x')$  wahr?



## Polynome

- Ein Monomial hat nur einen Term:  $x^d$
- Alle Produkte von d Vektorkomponenten ergeben Merkmalsraum  $\mathcal{H}$ .  
 $\Phi_2: \mathbb{R}^2 \rightarrow \mathcal{H}^3$   
 $([x]_1, [x]_2) \rightarrow ([x]_1^2, [x]_2^2, [x]_1[x]_2)$
- Sei N die Anzahl der Dimensionen (Buchstaben), d Dimensionen werden daraus verwendet (Länge d. Wortes), dann ist die Anzahl verschiedener Monomials (Wörter):  

$$\binom{d+N-1}{d} = \frac{(d+N-1)!}{d!(N-1)!}$$



## Der Trick

- Der Merkmalsraum ist sehr groß.
- Abbildung der Beispiele in den Merkmalsraum und dann Berechnen der Skalarprodukte zwischen den transformierten Beispielen ist sehr ineffizient.
- Eine Kernfunktion, die auf die Beispiele direkt angewandt dasselbe Ergebnis liefert, wäre effizient!
- Diese Kernfunktion ist hier  $(x * x')^2$



## Skalarprodukt im Polynomraum

- Geordnete Monomials:  
 $\Phi_2: \mathfrak{R}^2 \rightarrow \mathcal{H}^4$   
 $\Phi_2(x) : ([x]_1, [x]_2) \rightarrow ([x]_1^2, [x]_2^2, [x]_1[x]_2, [x]_2[x]_1)$
- Skalarprodukt im Merkmalsraum:  

$$\begin{aligned} \Phi_2(x) * \Phi_2(x') &= [x]_1^2 [x']_1^2 + [x]_2^2 [x']_2^2 + 2[x]_1 [x]_2 [x']_1 [x']_2 \\ &= (x * x')^2 \end{aligned}$$
- Dies gilt allgemein für alle geordneten Produkte d-ten Grades der Komponenten von x:  $\Phi_d(x) * \Phi_d(x') = (x * x')^d$



## Beweis

$$\begin{aligned} k(x, x') &= \Phi_d(x) * \Phi_d(x') = (x * x')^d \\ \Phi_d(x) * \Phi_d(x') &= \sum_{j_1=1}^N \dots \sum_{j_d=1}^N [x]_{j_1} \cdot \dots \cdot [x]_{j_d} \cdot [x']_{j_1} \cdot \dots \cdot [x']_{j_d} \\ &= \sum_{j_1=1}^N [x]_{j_1} \cdot [x']_{j_1} \dots \sum_{j_d=1}^N [x]_{j_d} \cdot [x']_{j_d} \\ &= \left( \sum_{j=1}^N [x]_j \cdot [x']_j \right)^d \\ &= (x * x')^d \end{aligned}$$



## Randbemerkung

- Gerade wurden Reihenfolgen unterschieden.
- Üblicherweise ist  $\Phi_d$  aber ohne Doppelte.
- Eine Komponente, die Doppelte enthalten würde, wird durch die Wurzel skaliert.

$$\Phi_2(x) = ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2)$$

- Die genaue Form von  $\Phi_d$  ist aber egal: beide ergeben dieselbe Kernfunktion  $k(x, x') = (x * x')^d$



## Verallgemeinerung

- Die Gram Matrix K für eine Funktion  $k: X^2 \rightarrow \mathfrak{R}$  und Beobachtungen  $x_1, \dots, x_m$  ist eine  $m \times m$  Matrix  
 $K_{ij} := k(x_i, x_j)$
- Eine (reellwertige) positiv definite Matrix ist eine  $m \times m$  Matrix K, für die für alle  $c_i$  in  $\mathfrak{R}$  gilt  

$$\sum_{i,j} c_i c_j K_{ij} \geq 0$$
- Einfaches Beispiel: 1) gelingt nicht, 2) gelingt

$$K = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \quad c_i, c_j = 1 \quad K = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad c_i, c_j = 1$$



## Kernfunktion Definition

- Eine Funktion  $k$  über  $X \times X$ , die sich für alle  $x_i$  in  $X$  als positiv definite Gram Matrix mit symmetrischer Funktion  $k$  darstellen lässt, heißt Kernfunktion oder reproduzierender Kern oder Kovarianzfunktion.

- Bei Hilbert: Eine Funktion  $k$ , die zu dem Operator  $T_k$  führt durch  $(T_k f)(x) = \int k(x, x') f(x') dx'$

heißt Kern von  $T_k$ .



## Konstruieren in 2 Richtungen

- Gegeben eine Kernfunktion  $k$ , konstruiere einen Merkmalsraum, in den  $\Phi$  abbildet. Jede Kernfunktion kann als Skalarprodukt in einem anderen Raum betrachtet werden.
- Gegeben ein Merkmalsraum  $\Phi$  mit Skalarprodukt, konstruiere eine Kernfunktion  $k(x, x') = \Phi(x) \cdot \Phi(x')$ . Gelingt, weil für alle  $c_i$  in  $\mathfrak{R}$  und  $x_i$  in  $X$  gilt:

$$\begin{aligned} \sum_{i,j} c_i c_j k(x_i, x_j) &= \sum_i c_i \Phi(x_i) \cdot \sum_j c_j \Phi(x_j) \\ &= \left\| \sum_i c_i \Phi(x_i) \right\|^2 \geq 0 \end{aligned}$$

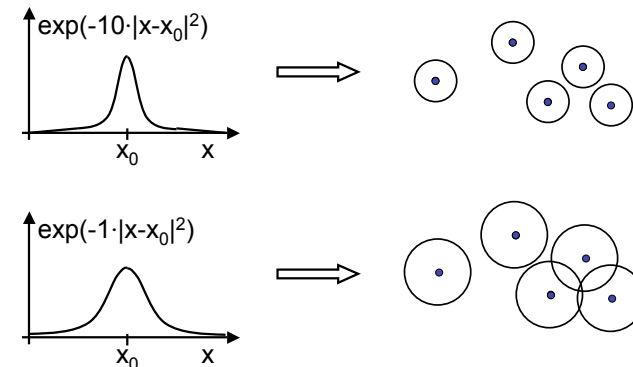


## Kernfunktionen praktisch

- Angabe von  $\Phi$  nicht nötig, einzige Bedingung: Kernmatrix  $(K(x_i, x_j))_{i,j=1 \dots n}$  muss positiv definit sein.
- Polynom:  $K(x, x') = (x \cdot x')^d$
- Radial-Basisfunktion:  $K(x, x') = \exp(-\gamma \|x - x'\|^2)$
- Neuronale Netze:  $K(x, x') = \tanh(\alpha \cdot x \cdot x' + b)$
- Konstruktion von Spezialkernen durch Summen und Produkte von Kernfunktionen, Multiplikation mit positiver Zahl, Weglassen von Attributen



## RBF-Kernfunktion





## Eigenschaften von RBF-Kernen

- Allgemeine Form:  $k(x, x') = f(d(x, x'))$
- Funktion  $f$  kann außer Gauss z.B. auch B-Spline sein.
- Als Metrik  $d(x, x')$  wird auch gewählt  $\|x - x'\| = \sqrt{(x - x')^* (x - x')}$
- Bei  $\|x - x'\|^2$  oder  $\|x - x'\|$  ist die RBF-Kernfunktion invariant bezüglich Drehung und Verschiebung.
- Das bedeutet, dass das Lernergebnis unabhängig von dem Koordinatensystem unserer Daten ist.



## Was wissen Sie jetzt?

- Kernfunktionen berechnen das Skalarprodukt der Beobachtungen in einem Merkmalsraum, ohne tatsächlich erst in den Merkmalsraum abzubilden.  $k(x, x') = \Phi(x)^* \Phi(x')$
- Polykern und RBF-Kern als Beispiele.
- Der Kern-Trick:  $k(x, x')$  lässt sich allein aus  $x^* x'$  berechnen.
- Eine Funktion  $X \times X$ , die sich für alle  $x_i$  in  $X$  als positiv definite Gram Matrix mit symmetrischer Funktion  $k$  darstellen lässt, heißt Kernfunktion.
- Die Mercer Bedingung prüft, ob es sich um eine Kernfunktion handelt, also die Matrix positiv definit ist.



## Mercer Bedingung

- Es gibt eine Abbildung  $\Phi$  und eine Kernfunktion

$$k(x, x') = \sum_i \Phi([x])^* \Phi([x'])$$

gdw. für jedes  $g(x)$  mit finitem

$$\int g(x)^2 dx$$

gilt:

$$\int k(x, x') g(x) g(x') dx dx' \geq 0$$

- Wenn die Mercer Bedingung nicht gilt, könnte die Hesse Matrix über den Beispielen indefinit werden.



## Duales weiches Optimierungsproblem

- Maximiere

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

$$\text{u.d. Bedingungen } \sum_{i=1}^m y_i \alpha_i = 0, \forall i : 0 \leq \alpha_i \leq C$$



## Optimierungsproblem mit Kern

- Erst minimierten wir  $w$ , dann maximierten wir das duale Problem, jetzt minimieren wir das duale Problem, indem wir alles mit  $-1$  multiplizieren...
- Minimiere  $L'(\alpha)$  mit

$$L'(\alpha) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y_i y_j K(x_i, x_j) \alpha_i \alpha_j - \sum_{i=1}^m \alpha_i$$

unter den Nebenbedingungen

$$0 \leq \alpha_i \leq C$$

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



## Chunking

- Beispiele  $x_i$  mit  $\alpha_i = 0$  können aus der Matrix gestrichen werden.
  - Finde alle diese Beispiele, lösche sie.
  - Löse das Optimierungsproblem für die verbleibenden.
- Iteratives Vorgehen:
  - Löse das Optimierungsproblem für die  $\alpha_i \neq 0$  aus dem vorigen Schritt und einige Beispiele, die die KKT-Bedingungen verletzen.
- Osuna, Freund, Girosi (1997): feste Matrixgröße.



## Algorithmus für das Optimierungsproblem

- Berechnen wir  $L'(\alpha)$  durch Gradientensuche!
  - Naiver Ansatz berechnet Gradienten an einem Startpunkt und sucht in angegebener Richtung bis kleinster Wert gefunden ist. Dabei wird immer die Nebenbedingung eingehalten. Bei  $m$  Beispielen hat  $\alpha$   $m$  Komponenten, nach denen es optimiert werden muss. Alle Komponenten von  $\alpha$  auf einmal optimieren?  $m^2$  Terme!
  - Eine Komponente von  $\alpha$  ändern? Nebenbedingung verletzt.



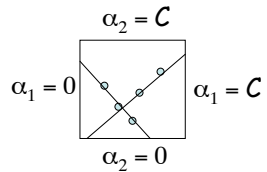
## Sequential Minimal Optimization

- Zwei Komponenten  $\alpha_1, \alpha_2$  im Bereich  $[0, C] \times [0, C]$  verändern!
  - Optimieren von zwei  $\alpha_i$
  - Auswahl der  $\alpha_i$



## KKT-Bedingungen einfach

- Notwendige und hinreichende Bedingungen an die Lösung des Optimierungsproblems: für alle  $i$ 
  - $\alpha_i = 0$  gdw.  $y_i f(x_i) \geq 1$
  - $\alpha_i = C$  gdw.  $y_i f(x_i) \leq -1$
  - $0 < \alpha_i < C$  gdw.  $y_i f(x_i) = -1$



$$\begin{aligned} y_1 &= y_2 \\ \alpha_1 - \alpha_2 & \end{aligned}$$

$$\begin{aligned} 0 &\leq \alpha_i \leq C \\ \sum_{i=1}^m y_i \alpha_i &= 0 \end{aligned}$$



## $\alpha_2$ optimieren

- Maximum der Funktion  $L'(\alpha)$  entlang der Geraden  $g(\alpha_2) = s \alpha_2 + \alpha_1$  mit  $s = y_2/y_1$
- Wenn  $y_1 = y_2$  ist  $s = 1$ , also steigt die Gerade. Sonst  $s = -1$ , also fällt die Gerade.
- Schnittpunkte der Geraden mit dem Bereich  $[0, C] \times [0, C]$ :
  - Falls  $s$  steigt:  $\max(0; \alpha_2 + \alpha_1 - C)$  und  $\min(C; \alpha_2 + \alpha_1)$
  - Sonst:  $\max(0; \alpha_2 - \alpha_1)$  und  $\min(C; \alpha_2 - \alpha_1 + C)$
  - Optimales  $\alpha_2$  ist höchstens max-Term, mindestens min-Term.



## SMO

- Berechne  $\alpha_2$  und gib die Schnittpunkte max, min der Diagonalen mit der Box an.
- 2. Ableitung von  $L'$  entlang der Diagonalen  $\eta = k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2)$
- Wenn  $\eta > 0$ , wird das Minimum für  $\alpha_2$  ausgerechnet, wobei  $E$  der Fehler  $f(x) - y$  ist:

$$\alpha_2^{neu} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}$$

- Beschneiden, so dass  $\min \leq \alpha_2^{neu} \leq \max$
- Berechnen  $\alpha_1^{neu} = \alpha_1 + y_1 y_2 (\alpha_2 - \alpha_2^{neu})$



## Randbemerkung

- Ein nicht positives  $\eta$  kann dann auftreten, wenn
  - zwei Beispiele genau gleich aussehen oder
  - die Kernfunktion nicht der Mercer-Bedingung gehorcht.



## Algorithmus

- Äußere Schleife --  $\alpha_1$  wählen
  1. Alle Beispiele durchgehen:  
welche verletzen KKT-Bedingungen?
  2. Non-bound Beispiele suchen ( $\alpha_1$  weder 0 noch C ):  
welche verletzen KKT-Bedingungen?  
Verändern bis alle non-bound Beispiele KKT-Bedingungen erfüllen!
  3. Goto 1
- Innere Schleife --  $\alpha_2$  wählen
  1. Wenn  $E_1 > 0$ , Beispiel mit kleinem  $E_2$  suchen,  
wenn  $E_1 < 0$ , Beispiel mit großem  $E_2$  suchen.
  2.  $L'(\alpha)$  ausrechnen
  3. b ausrechnen

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Satz von Osuna

- Der Algorithmus konvergiert, solange an jedem Schritt 2 Lagrange Multiplikatoren optimiert werden und mindestens einer davon verletzt vorher die KKT-Bedingungen.

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Was wissen Sie jetzt?

- Das Optimierungsproblem wird durch Optimieren je zweier Lagrange-Multiplikatoren gelöst, äußere Schleife wählt ersten, innere zweiten Multiplikator.
- Sei  $\alpha = (\alpha_1, \dots, \alpha_m)$  eine Lösung des Optimierungsproblems. Wir wählen zum update:
 
$$\hat{\alpha}_2 = \alpha_2 + \frac{y_2((f(x_1) - y_1) - (f(x_2) - y_2))}{K(x_1, x_1) - 2K(x_1, x_2) + K(x_2, x_2)}$$
- Optimales  $\hat{\alpha}_1 = \alpha_1 + y_1 y_2 (\alpha_2 - \hat{\alpha}_2)$
- Prinzip des Optimierens: Nullsetzen der ersten Ableitung...
- Der Algorithmus konvergiert, wenn vorher ein  $\alpha$  KKT-Bedingung verletzte.

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Was ist gutes Lernen?

- Fauler Botaniker:  
"klar ist das ein Baum – ist ja grün."
  - Übergeneralisierung
  - Wenig Kapazität
  - Bias
- Botaniker mit fotografischem Gedächtnis:  
"nein, dies ist kein Baum, er hat 15 267 Blätter und kein anderer hatte genau so viele."
  - Overfitting
  - Viel Kapazität
  - Varianz
- Kontrolle der Kapazität!

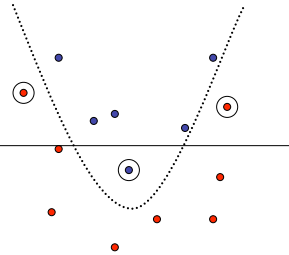
Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008





## Bias-Varianz-Problem

- Zu kleiner Hypothesenraum: Zielfunktion nicht gut genug approximierbar (Bias)
- Zu großer Hypothesenraum: Zuviel Einfluss zufälliger Abweichungen (Varianz)
- Lösung: Minimiere obere Schranke des Fehlers:  
 $R(\alpha) \leq R_{emp}(\alpha) + Var(\alpha)$



## Risikoschranke nach Vapnik

- Gegeben eine unbekannte Wahrscheinlichkeitsverteilung  $P(x,y)$  nach der Daten gezogen werden. Die Abbildungen  $x \rightarrow f(x, \alpha)$  werden dadurch gelernt, dass  $\alpha$  bestimmt wird. Mit einer Wahrscheinlichkeit  $1-\mu$  ist das Risiko  $R(\alpha)$  nach dem Sehen von  $l$  Beispielen beschränkt:

$$R(\alpha) \leq R_{emp}(\alpha) + \underbrace{\sqrt{\frac{\eta(\log(2l/\eta) + 1) - \log(\mu/4)}{l}}}_{VC \text{ confidence}}$$



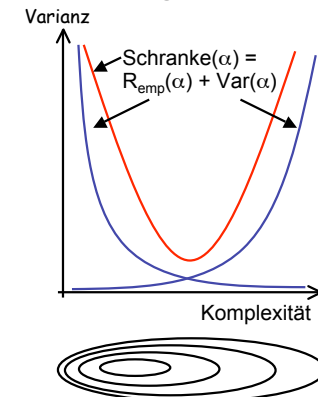
## Strukturelle Risikoschranke

- Unabhängig von einer Verteilungsannahme. Alles, was die Schranke braucht, ist, dass Trainings- und Testdaten gemäß der selben Wahrscheinlichkeitsverteilung gezogen werden.
- Das tatsächliche Risiko können wir nicht berechnen.
- Die rechte Seite der Ungleichung können wir berechnen, sobald wir  $\eta$  kennen.
- Gegeben eine Menge Hypothesen für  $f(x,\alpha)$ , wähle immer die mit dem niedrigsten Wert für die rechte Seite der Schranke ( $R_{emp}$  oder VC confidence niedrig).



## Strukturelle Risikominimierung

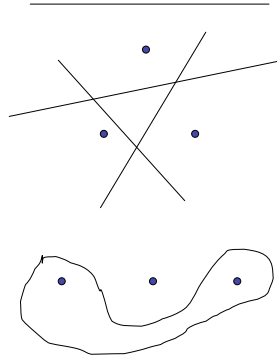
1. Ordne die Hypothesen in Teilmenge gemäß ihrer Komplexität
2. Wähle in jeder Teilmenge die Hypothese mit dem geringsten empirischen Fehler
3. Wähle insgesamt die Hypothese mit minimaler Risikoschranke





## Vapnik-Chervonenkis-Dimension

- Definition: Eine Menge  $H$  von Hypothesen *zerschmettert* eine Menge  $E$  von Beispielen, wenn jede Teilmenge von  $E$  durch ein  $h \in H$  abgetrennt werden kann.
- Definition: Die *VC-Dimension* einer Menge von Hypothesen  $H$  ist die maximale Anzahl von Beispielen  $E$ , die von  $H$  zerschmettert wird.
- Eine Menge von 3 Punkten kann von geraden Linien zerschmettert werden, keine Menge von 4 Punkten kann von geraden Linien zerschmettert werden.



## ACHTUNG

- Für eine Klasse von Lernaufgaben gibt es mindestens eine Menge  $E$ , die zerschmettert werden kann – NICHT jede Menge  $E$  kann zerschmettert werden!
- Zum Beweis der VC Dimension  $n$  muss man also zeigen:
  - Es gibt eine Menge  $E$  aus  $n$  Punkten, die von  $H$  zerschmettert werden kann.  $VCdim(H) \geq n$
  - Es kann keine Menge  $E'$  aus  $n+1$  Punkten geben, die von  $H$  zerschmettert werden könnte.  $VCdim(H) \leq n$

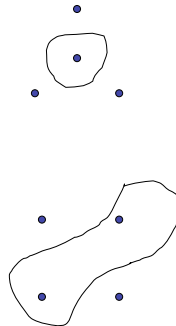


## VC-Dimension von Hyperebenen

Satz: Die VC-Dimension der Hyperebenen im  $\mathbb{R}^n$  ist  $n+1$ .

Beweis:

- $VCdim(\mathbb{R}^n) \geq n+1$ : Wähle  $x_0 = 0$  und  $x_i = (0, \dots, 0, 1, 0, \dots, 0)$ . Für eine beliebige Teilmenge  $A$  von  $(x_0, \dots, x_n)$  setze  $y_i = 1$ , falls  $x_i \in A$  und  $y_i = -1$  sonst. Definiere  $w = \sum y_k x_k$  und  $b = y_0/2$ . Dann gilt  $w x_0 + b = y_0/2$  und  $w x_i + b = y_i + y_0/2$ . Also:  $w x + b$  trennt  $A$ .
- $VCdim(\mathbb{R}^n) \leq n+1$ : Zurückführen auf die beiden Fälle rechts.



## VCdim misst Kapazität

- Eine Funktion mit nur 1 Parameter kann unendliche VCdim haben:  $H$  kann Mengen von  $n$  Punkten zerschmettern, egal wie groß  $n$  ist.
- $H$  kann unendliche VCdim haben und trotzdem kann ich eine kleine Zahl von Punkten finden, die  $H$  nicht zerschmettern kann.
- VCdim ist also nicht groß, wenn die Anzahl der Parameter bei der Klasse von Funktionen  $H$  groß ist.



## VC-Dim. und Anzahl der Parameter

- Setze  $f_\alpha(x) = \cos(\alpha x)$  und  $x_i = 10^{-i}$ ,  $i=1 \dots \ell$ , beliebiges  $\ell$ . Wähle  $y_i \in \{-1, 1\}$ . Dann gilt für  $\alpha = \pi(\sum_{i=1}^{\ell} \frac{1}{2}(1-y_i)10^i)$ :

$$\alpha x_k = \pi \left( \sum_{i=1}^{\ell} \frac{1}{2}(1-y_i)10^i \right) 10^{-k} = \pi \left( \sum_{i=1}^{\ell} \frac{1}{2}(1-y_i)10^{i-k} \right)$$

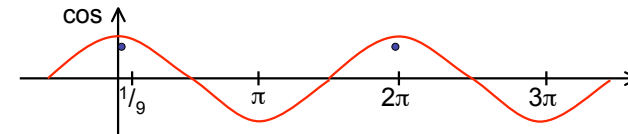
$$= \pi \left( \underbrace{\sum_{i=1}^{k-1} \frac{1}{2}(1-y_i)10^{i-k}}_{0 \leq \sum \dots \leq 10^{-1}+10^{-2}+ \dots = 1/9 \text{ (geometrische Reihe)}} + \frac{1}{2}(1-y_k) + \underbrace{\sum_{i=k+1}^{\ell} \frac{1}{2}(1-y_i)10^{i-k}}_{\text{Vielfaches von 2}} \right)$$

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## VC-Dim. und Anzahl der Parameter

- $\Rightarrow \cos(\alpha x_k) = \cos(\pi z)$  mit  $z \in [0, 1/9]$  für  $y_k = 1$  und  $z \in [1, 10/9]$  für  $y_k = -1$



- $\Rightarrow \cos(\alpha x)$  zerschmettert  $x_1, \dots, x_\ell$
- $\Rightarrow \cos(\alpha x)$  hat unendliche VC-Dimension
- $\Rightarrow$  Die VC-Dimension ist unabhängig von der Anzahl der Parameter!

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## VC-Dimension der SVM

- Gegeben seien Beispiele  $x_1, \dots, x_\ell \in \mathbb{R}^n$  mit  $\|x_i\| < D$  für alle  $i$ . Für die VC-Dimension der durch den Vektor  $w$  gegebenen optimalen Hyperebene  $h$  gilt:  
 $VCdim(h) \leq \min\{D^2 \|w\|^2, n\} + 1$
- Die Komplexität einer SVM ist nicht nur durch die Struktur der Daten beschränkt (Fluch der hohen Dimension), sondern auch durch die Struktur der Lösung!

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Zusicherungen

- Strukturelle Risikominimierung garantiert, dass die einfachste Hypothese gewählt wird, die noch an die Daten anpassbar ist.
- Strukturelle Risikominimierung kontrolliert die Kapazität des Lernens (weder fauler noch fotografischer Botaniker).
- Die Strukturen von Klassen von Funktionen werden durch die VCdim ausgedrückt. Große VCdim  $\rightarrow$  große VC-confidence.

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Was wissen wir jetzt?

- Kernfunktionen – eine Transformation, die man nicht erst durchführen und dann mit ihr rechnen muss, sondern bei der nur das Skalarprodukt gerechnet wird.
- Idee der strukturellen Risikominimierung:
  - obere Schranke für das Risiko
  - Schrittweise Steigerung der Komplexität
- Formalisierung der Komplexität: VC-Dimension
- SRM als Prinzip der SVM
- Garantie für die Korrektheit der Lernstrategie



## Performanzschätzer

- Welches erwartete Risiko  $R(\alpha)$  erreicht SVM?
- $R(\alpha)$  selbst nicht berechenbar
- Trainingsfehler (zu optimistisch – Overfitting)
- Obere Schranke mittels VC-Dimension (zu locker)
- Kreuzvalidierung / Leave-One-Out-Schätzer (ineffizient)



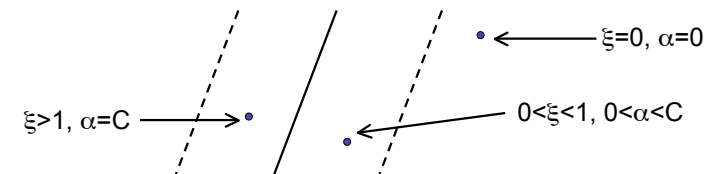
## Performanzschätzer II

- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch  $R_{10} \leq |SV| / n$
- Beweis: Falsch klassifizierte Beispiele werden Stützvektoren. Also: Nicht-Stützvektoren werden korrekt klassifiziert. Weglassen eines Nicht-Stützvektors ändert die Hyperebene nicht, daher wird es auch beim 10-Test richtig klassifiziert.



## Performanzschätzer III

- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch  $R_{10} \leq |\{i : (2\alpha_i D^2 + \xi_i) \geq 1\}| / n$  ( $D$  = Radius des Umkreises um die Beispiele im transformierten Raum).
- Beweis: Betrachte folgende drei Fälle:





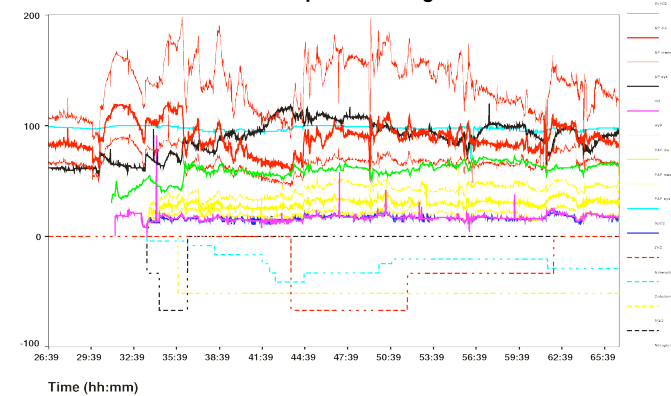
## Fallstudie Intensivmedizin

- Städtische Kliniken Dortmund, Intensivmedizin 16 Betten, Priv.-Doz. Dr. Michael Imhoff
- Hämodynamisches Monitoring, minütliche Messungen
  - Diastolischer, systolischer, mittlerer arterieller Druck
  - Diastolischer, systolischer, mittlerer pulmonarer Druck
  - Herzrate
  - Zentralvenöser Druck
- Therapie, Medikamente:
  - Dobutamine, adrenaline, glycerol trinitrate, noradrenaline, dopamine, nifedipine



## Patient G.C., male, 60 years old

### Hemihepatektomie right



## Wann wird Medikament gegeben?

- Mehrklassenproblem in mehrere 2Klassen-Probleme umwandeln:
  - Für jedes Medikament entscheide, ob es gegeben werden soll oder nicht.
  - Positive Beispiele: alle Minuten, in denen das Medikament gegeben wurde
  - Negative Beispiele: alle Minuten, in denen das Medikament nicht gegeben wurde

Parameter: Kosten falscher Positiver = Kosten falscher Negativer

Ergebnis: Gewichte der Vitalwerte so dass positive und negative Beispiele maximal getrennt werden (SVM).



## Beispiel: Intensivmedizin

$$f(x) = \begin{bmatrix} 0.014 \\ 0.019 \\ -0.001 \\ -0.015 \\ -0.016 \\ 0.026 \\ 0.134 \\ -0.177 \\ \vdots \end{bmatrix} \begin{pmatrix} artsys = 174.00 \\ artdia = 86.00 \\ artmn = 121.00 \\ cvp = 8.00 \\ hr = 79.00 \\ papsys = 26.00 \\ papdia = 13.00 \\ papmn = 15.00 \\ \vdots \end{pmatrix} - 4.368$$

- Vitalzeichen von Intensivpatienten
- Hohe Genauigkeit
- Verständlichkeit?



## Wie wird Medikament dosiert ?

- Mehrklassenproblem in mehrere 2Klassenprobleme umwandeln: für jedes Medikament und jede Richtung (increase, decrease, equal), 2 Mengen von Patienten-daten:
  - Positive Beispiele: alle Minuten, in denen die Dosierung in der betreffenden Richtung geändert wurde
  - Negative Beispiele: alle Minuten, in denen die Dosierung nicht in der betreffenden Richtung geändert wurde.



## Steigern von Dobutamine

ARTEREN: -0.05108108119  
 SUPRA: 0.00892807538657973  
 DOBUTREX: -0.100650806786886  
 WEIGHT: -0.0393531801046265  
 AGE: -0.00378828681071417  
 ARTSYS: -0.323407537252192  
 ARTDIA: -0.0394565333019493  
 ARTMN: -0.180425080906375  
 HR: -0.10010405264306  
 PAPSYS: -0.0252641188531731  
 PAPDIA: 0.0454843337112765  
 PAPMN: 0.00429504963736522  
 PULS: -0.0313501236399881

Vektor w für k Attribute



## Anwendung des Gelernten

- Patientwerte
  - pat46, artmn 95, min. 2231
  - ...
  - pat46, artmn 90, min. 2619
- Gelernte Gewichte für Dobutamin
  - artmn -0,18
  - ...

$$svm\_calc = \sum_{i=1}^k w_i x_i \quad decision = sign(svm\_calc + b)$$

svm\_calc (pat46, dobutrex, up,min.2231,39) svm\_calc (pat46, dobutrex, up,min.2619, 25)

b=-26, i.e. increase in minute 2231,  
 not increase in minute 2619.



## Steigern von Glyceroltrinitrat

$$sign \begin{pmatrix} 0.014 & artsys174.00 \\ 0.019 & artdia86.00 \\ -0.001 & artmn121.00 \\ -0.015 & cvp8.00 \\ -0.016 & hr79.00 \\ 0.026 & papsys26.00 \\ 0.134 & papdia13.00 \\ -0.177 & papmn15.00 \\ -9.543 & nifedipine0 \\ -1.047 & noradrenaline0 \\ -0.185 & dobutamie0 \\ 0.542 & dopamie0 \\ -0.017 & glyceroltrinitrate0 \\ 2.391 & adrenaline0 \\ 0.033 & age77.91 \\ 0.334 & emergency0 \\ 0.784 & bsal.79 \\ 0.015 & broca1.02 \end{pmatrix} - 4.368$$

Jedes Medikament hat einen Dosierungsschritt.  
 Für Glyceroltrinitrat ist es 1,  
 für Suprarenin (adrenalin) 0.01.  
 Die Dosis wird um einen Schritt erhöht oder gesenkt.

Vorhersage:  
 pred\_interv(pat49, min.32,nitro, 1.0)



## Evaluierung

- Blind test über 95 noch nicht gesehener Patientendaten.
  - Experte stimmte überein mit tatsächlichen Medikamentengaben in 52 Fällen
  - SVM Ergebnis stimmte überein mit tatsächlichen Medikamentengaben in 58 Fällen

| Dobutamine      | Actual up     | Actual equal   | Actual down    |
|-----------------|---------------|----------------|----------------|
| Predicted up    | <b>10 (9)</b> | 12 (8)         | 0 (0)          |
| Predicted equal | 7 (9)         | <b>35 (31)</b> | 9 (9)          |
| Predicted down  | 2 (1)         | 7 (15)         | <b>13 (12)</b> |



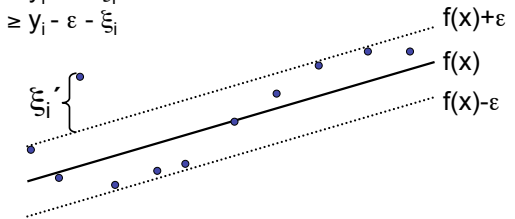
## SVMs für Regression

- Minimiere  $\|w\|^2 + C \left( \sum_{i=1}^n \xi_i + \sum_{i=1}^n \xi_i' \right)$

- so dass für alle i gilt:

$$f(x_i) = w^*x_i + b \leq y_i + \varepsilon + \xi_i' \text{ und}$$

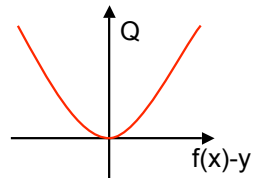
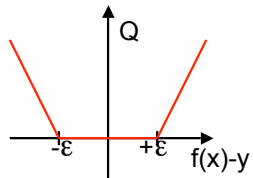
$$f(x_i) = w^*x_i + b \geq y_i - \varepsilon - \xi_i$$



## Verlustfunktion

lineare Verlustfunktion

quadratische Verlustfunktion



## Duales Optimierungsproblem

- Maximiere

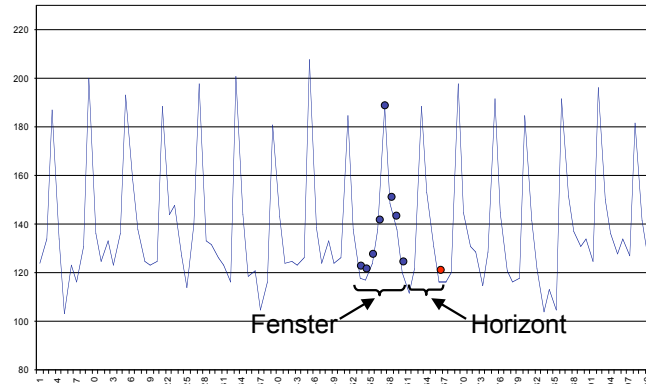
$$W(\alpha) = \sum_{i=1}^n y_i (\alpha_i' - \alpha_i) - \varepsilon \sum_{i=1}^n (\alpha_i' + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i' - \alpha_i) (\alpha_j' - \alpha_j) K(x_i, x_j)$$

- unter  $0 \leq \alpha_i, \alpha_i' \leq C$  für alle i und  $\sum \alpha_i' = \sum \alpha_i$

- Mit  $y_i \in \{-1, +1\}$ ,  $\varepsilon = 0$  und  $\alpha_i = 0$  für  $y_i = 1$  und  $\alpha_i' = 0$  für  $y_i = -1$  erhält man die Klassifikations-SVM!



## Beispiel: Prognose von Zeitreihen



Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



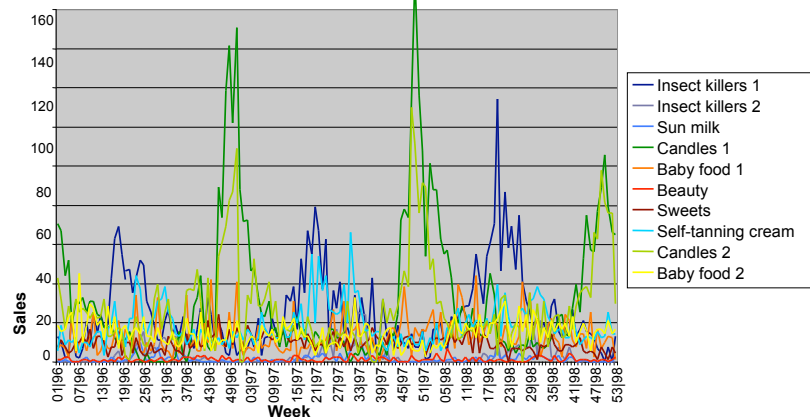
## Prognose von Zeitreihen

- Trend
- Zyklen
- Besondere Ereignisse (Weihnachten, Werbung, ...)
- Wieviel vergangene Beobachtungen?
- Ausreißer

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Abverkauf Drogerieartikel



Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008



## Vorhersage Abverkauf

Gegeben Verkaufsdaten von 50 Artikeln in 20 Läden über 104 Wochen

Vorhersage Verkäufe eines Artikels, so dass

Die Vorhersage niemals den Verkauf unterschätzt,

Die Vorhersage überschätzt weniger als eine Faustregel.

Beobachtung: 90% der Artikel werden weniger als 10 mal pro Woche verkauft.

Anforderung: Vorhersagehorizont von mehr als 4 Wochen.

Prof. Dr. Katharina Morik | Wissensentdeckung in Datenbanken SoSe 2008





## Verkaufsdaten

| Shop | Week | Item1 | ... | Item50 |
|------|------|-------|-----|--------|
| Dm1  | 1    | 4     | ... | 12     |
| Dm1  | ...  | ...   | ... | ...    |
| Dm1  | 104  | 9     | ... | 16     |
| Dm2  | 1    | 3     | ... | 19     |
| ...  | ...  | ...   | ... | ...    |
| Dm20 | 104  | 12    | ... | 16     |

$LE_{DB1} : I: T_1 A_1 \dots A_{50}$ ; Menge multivariater Zeitreihen



## Vorverarbeitung

- Multivariat nach univariat

$L_{E1} : i:t_1 a_1 \dots t_k a_k$

For all shops for all items:  
Create view Univariate as  
Select shop, week, item<sub>i</sub>  
Where shop="dm<sub>j</sub>"  
From Source;

|            |   |       |     |    |
|------------|---|-------|-----|----|
| Dm1_Item1  | 1 | 4...  | 104 | 9  |
| ...        |   |       |     |    |
| Dm1_Item50 | 1 | 12... | 104 | 16 |

- Multiples Lernen

....

|             |   |       |     |    |
|-------------|---|-------|-----|----|
| Dm20_Item50 | 1 | 14... | 104 | 16 |
|-------------|---|-------|-----|----|



## Vorverarbeitung II

- Viele Vektoren aus einer Reihe gewinnen durch Fenster

$L_{H5} : i:t_1 a_1 \dots t_w a_w$

bewege Fenster der Größe w um m Zeitpunkte

|                 |     |       |     |    |
|-----------------|-----|-------|-----|----|
| Dm1_Item1_1     | 1   | 4...  | 5   | 7  |
| Dm1_Item1_2     | 2   | 4...  | 6   | 8  |
| ...             |     |       |     |    |
| Dm1_Item1_100   | 100 | 6...  | 104 | 9  |
| ...             |     |       |     |    |
| ...             |     |       |     |    |
| Dm20_Item50_100 | 100 | 12... | 104 | 16 |



## SVM im Regressionfall

- Multiples Lernen:  
für jeden Laden und jeden Artikel, wende die SVM an. Die gelernte Regressionsfunktion wird zur Vorhersage genutzt.
- Asymmetrische Verlustfunktion :
  - Unterschätzung wird mit 20 multipliziert, d.h. 3 Verkäufe zu wenig vorhergesagt -- 60 Verlust
  - Überschätzung zählt unverändert, d.h. 3 Verkäufe zu viel vorhergesagt -- 3 Verlust

(Stefan Rüping 1999)



## Vergleich mit Exponential Smoothing

| Horizont | SVM    | exp. smoothing |
|----------|--------|----------------|
| 1        | 56.764 | 52.40          |
| 2        | 57.044 | 59.04          |
| 3        | 57.855 | 65.62          |
| 4        | 58.670 | 71.21          |
| 8        | 60.286 | 88.44          |
| 13       | 59.475 | 102.24         |

Verlust



## Was wissen wir jetzt?

- Anwendung der SVM für die Medikamentenverordnung
- Idee der Regressions-SVM
- Anwendung der SVM für die Verkaufsvorhersage
  - Umwandlung multivariater Zeitreihen in mehrere univariate
  - Gewinnung vieler Vektoren durch gleitende Fenster
  - Asymmetrische Verlustfunktion