Maschinelles Lernen Vorlesung

SVM – Kernfunktionen, Regularisierung

Katharina Morik

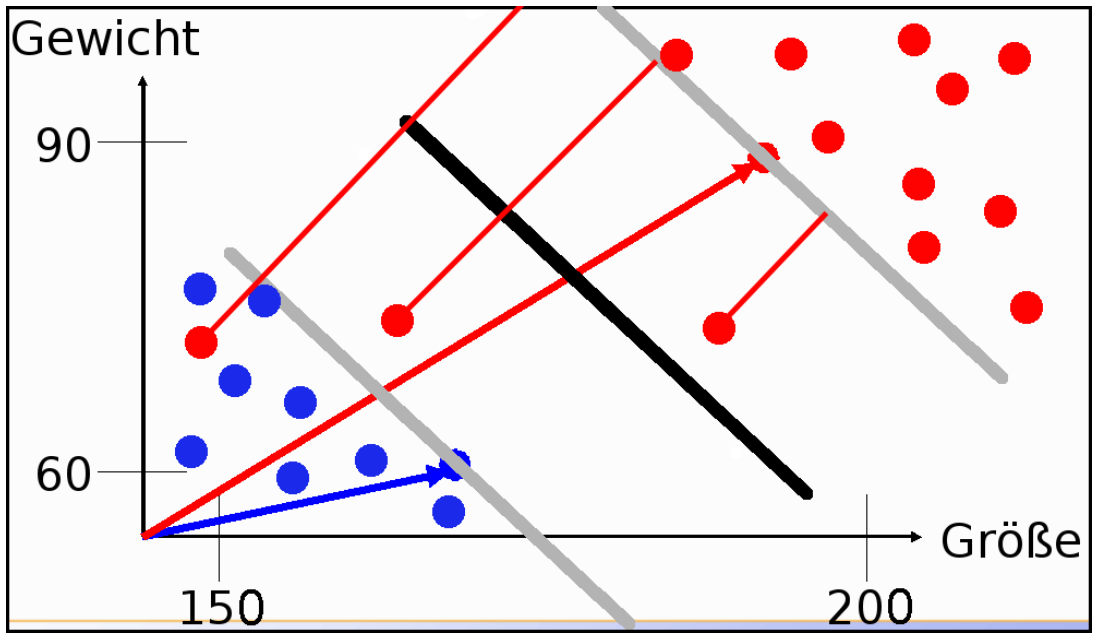
LS 8 Informatik
Technische Universität Dortmund

15.11.2011

Gliederung

SVM mit Ausnahmen

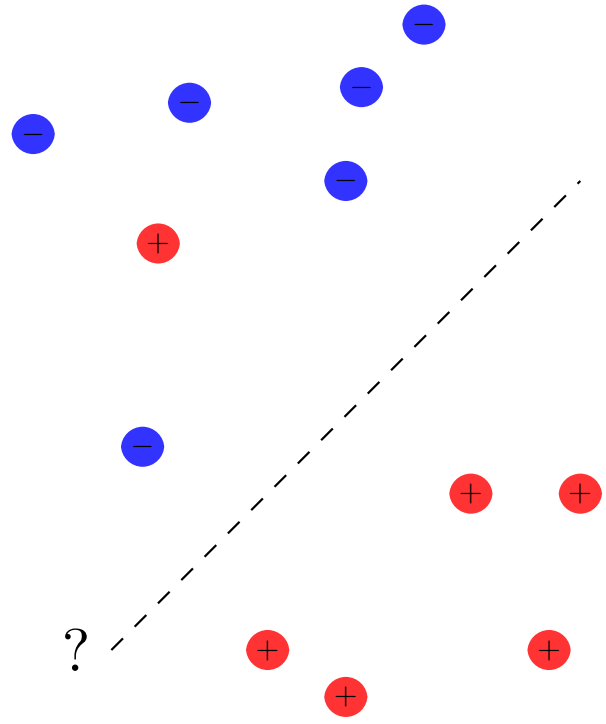
- Was passiert, wenn die Beispiele nicht komplett trennbar sind?



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.



SVM mit Ausnahmen

Ein anderer Ansatz basiert wieder auf einer Relaxation:

- Punkte, die nicht am Rand oder auf der richtigen Seite der Ebene liegen, bekommen einen Strafterm $\xi_j > 0$.
- Korrekt klassifizierte Punkte erhalten eine Variable $\xi_j = 0$.

Dies führt zu folgenden Minimierungsproblem

$$\frac{1}{2}\|\vec{\beta}\|^2 + C\sum_{j=1}^{N}\xi_j \quad \text{für ein festes } C \in \mathbb{R}_{>0} \tag{1}$$

Daraus folgt insbesondere

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

Weich trennende Hyperebene

Relaxiertes Optimierungsproblem

Sei $C \in \mathbb{R}$ mit $C > 0$ fest. Minimiere

$$||\vec{\beta}||^2 + C \sum_{i=1}^{N} \xi_i$$

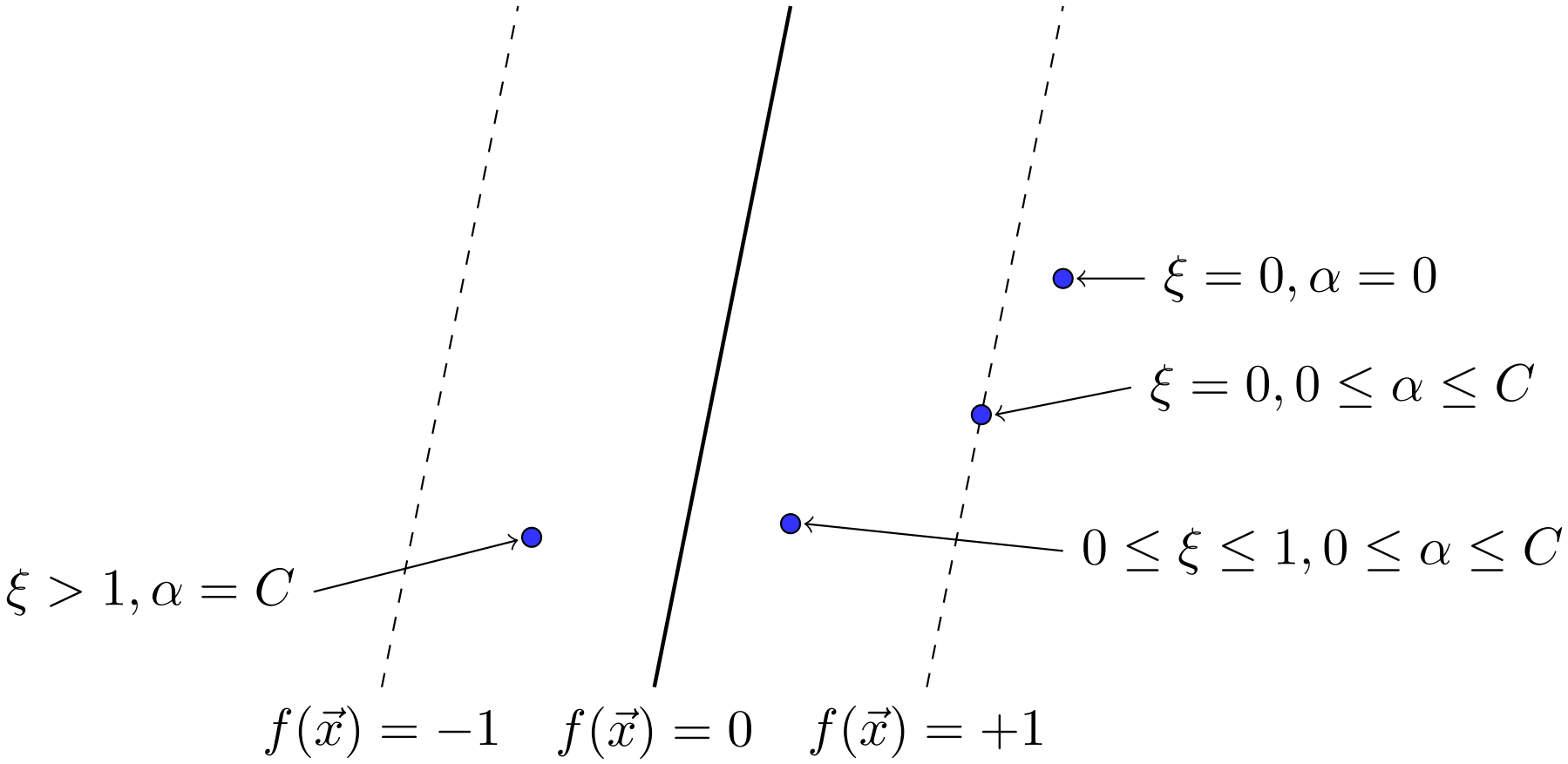
unter den Nebenbedingungen

$$\begin{aligned} \langle \vec{x}_i, \vec{\beta} \rangle + \beta_0 &\geq +1 - \xi_i \quad \text{für } \vec{y}_i = +1 \\ \langle \vec{x}_i, \vec{\beta} \rangle + \beta_0 &\leq -1 + \xi_i \quad \text{für } \vec{y}_i = -1 \end{aligned}$$

Durch Umformung erhalten wir wieder Bedingungen für die Lagrange-Optimierung:

$$y_i(\langle \vec{x}_i, \vec{\beta} \rangle + \beta_0) \geq 1 - \xi_i \quad \forall\, i = 1, \ldots, N$$

Bedeutung von ξ und $\vec{\alpha}$

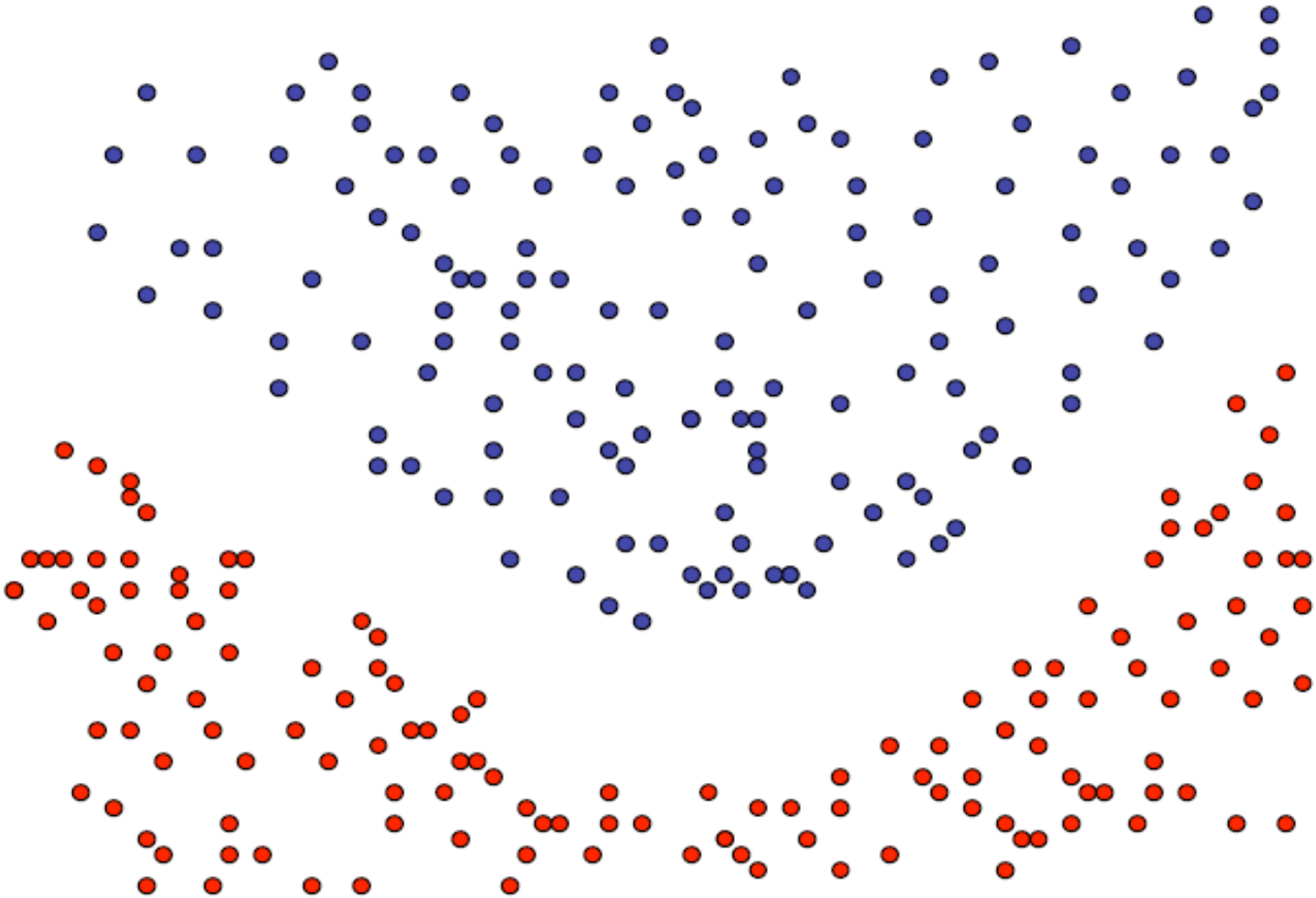


Beispiele $\vec{x}_i$ mit $\alpha_i > 0$ sind Stützvektoren.

Wo sind wir?

- Maximieren der Breite einer separierenden Hyperebene (maximum margin method) ergibt eindeutige, optimale trennende Hyperebene.
- Transformation des Datenraums durch Kernfunktion behandelt Nichtlinearität.
  - Das kam nur einmal am Rande vor. Wir sehen es nachher genauer.
- Regularisierung minimiert nicht nur den Fehler, sondern auch die Komplexität des Modells.
  - Später!

Nicht-lineare Daten

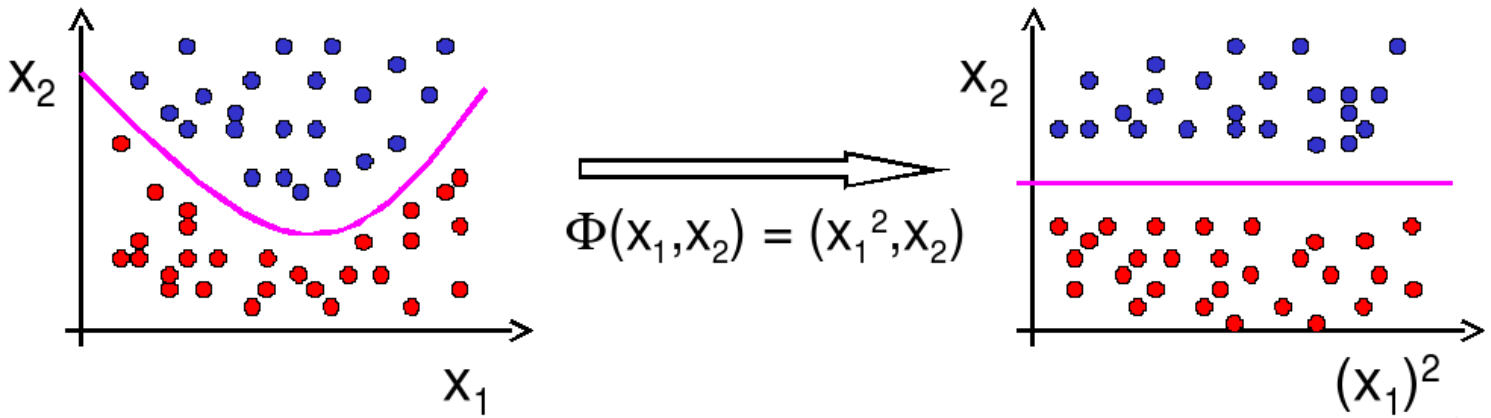


Nicht-lineare Daten

- 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!



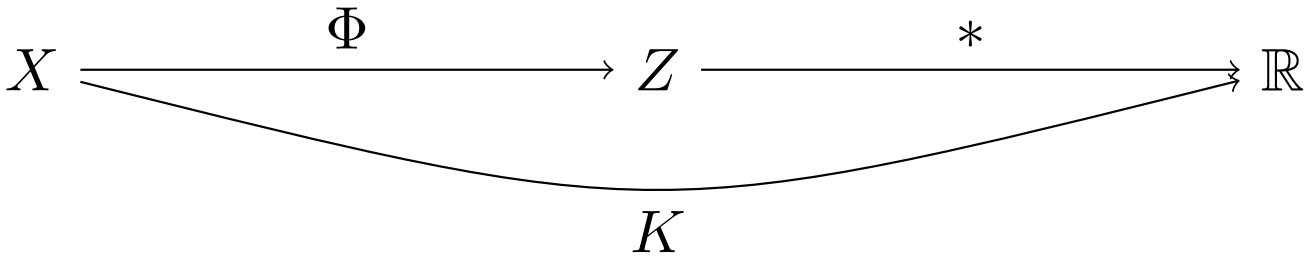
Kernfunktionen

- Erinnerung:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \langle \vec{x_i}, \vec{x_j} \rangle$$

$$f(\vec{x}) = \sum \alpha_i y_i \langle \vec{x_i}, \vec{x} \rangle + \beta_0$$

- SVM hängt von $\vec{x}$ nur über Skalarprodukt $\langle \vec{x}, \vec{x'} \rangle$ ab.
- Ersetze Transformation Φ und Skalarprodukt durch Kernfunktion $K(\vec{x_1}, \vec{x_2}) = \langle \Phi(\vec{x_1}), \Phi(\vec{x_2}) \rangle$

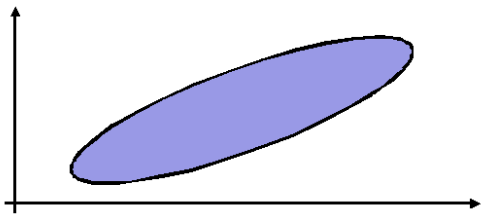


Kernfunktionen II

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

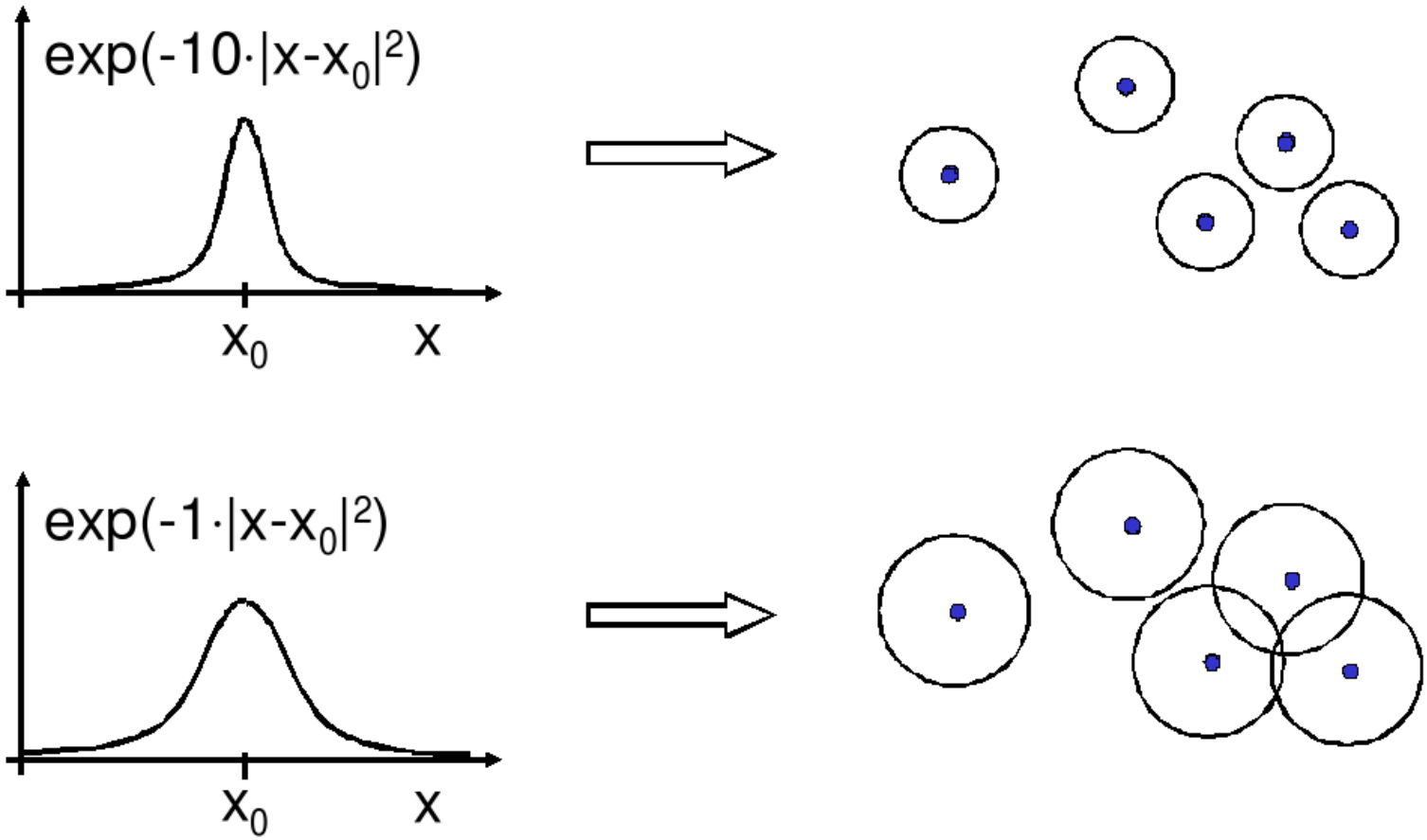
Polynom-Kernfunktionen

- $K_d(\vec{x_i}, \vec{x_j}) = \langle \vec{x_i}, \vec{x_j} \rangle^d$
- Beispiel: $d = 2, \vec{x_i}, \vec{x_j} \in \mathbb{R}^2$.



$$
\begin{aligned}
K_2(\vec{x_i}, \vec{x_j}) &= \langle \vec{x_i}, \vec{x_j} \rangle^2 \\
&= ((x_{i_1}, x_{i_2}) * (x_{j_1}, x_{j_2}))^2 = (x_{i_1} x_{j_1} + x_{i_2} x_{j_2})^2 \\
&= x_{i_1}^2 x_{j_1}^2 + 2 x_{i_1} x_{j_1} x_{i_2} x_{j_2} + x_{i_2}^2 x_{j_2}^2 \\
&= (x_{i_1}^2, \sqrt{2} x_{i_1} x_{i_2}, x_{i_2}^2) * (x_{j_1}^2, \sqrt{2} x_{j_1} x_{j_2}, x_{j_2}^2) \\
&=: \langle \phi(\vec{x_i}), \phi(\vec{x_j}) \rangle
\end{aligned}
$$

RBF-Kernfunktion



Kernfunktionen

- Die Kernfunktionen werden nicht als Vorverarbeitungsschritt durchgeführt.
- Man muss lediglich bei der Berechnung des Skalarprodukts die Kernfunktion berücksichtigen.
- Allerdings kann $\vec{\beta}$ jetzt nicht mehr so einfach interpretiert werden als Bedeutung der Variablen (Merkmale) X_i.

Interpretation der SVM

- Wenn das Skalarprodukt als Kernfunktion gewählt wird, entspricht jede Komponente des $\vec{\beta}$ einem Gewicht des Merkmals und jedes α dem Gewicht eines Beispiels $\vec{x}$, $\phi(\vec{x}) = \vec{x}$.
- Wenn nicht, wie finden wir zu jedem $\phi(\vec{x})$ den Ursprung $\vec{x}$?

$$
\begin{aligned}
f(\vec{x}) &= \sum_{i=1}^{N} \alpha_i K(\vec{x_i}, \vec{x}) \\
&= \sum_{i=1}^{s} \alpha_i \phi(\vec{x_i}) \cdot \phi(\vec{x}) \\
&= \left(\sum_{i=1}^{N} \alpha_i \phi(\vec{x_i}) \right) \cdot \phi(\vec{x}) \\
&=: \vec{\beta} \cdot \phi(\vec{x})
\end{aligned}
$$

Pre-Image Problem von Mika et al. 1998

Mika, Schölkopf, Smola, Müller, Scholz, Rätsch (1998) Kernel PCA and de-noising in feature spaces, in: NIPS, vol 11.
Rüping (2006) Learning Interpretable Models, Diss. TU Dortmund

Pre-Image Problem

Gegeben die Abbildung $\phi : X \to \mathcal{X}$ und ein Element aus dem Merkmalsraum, $\vec{\beta} \in \mathcal{X}$,
finde ein $\vec{x} \in X$, so dass $\phi(\vec{x}) = \vec{\beta}$.

Approximatives Pre-Image Problem

Gegeben die Abbildung $\phi : X \to \mathcal{X}$ und ein Element aus dem Merkmalsraum, $\vec{\beta} \in \mathcal{X}$,
finde ein $\vec{x} \in X$ mit minimalem Fehler $\| \vec{\beta} - \phi(\vec{x}) \|^2$.

Den Ursprung im Merkmalsraum suchen

Weil wir die genaue Abbildung ϕ nicht kennen, müssen wir den quadratischen Fehler im Merkmalsraum minimieren, um $\vec{x}$ zu finden.

$$\begin{aligned}\vec{x} &= argmin \parallel \vec{\beta} - \phi(\vec{x}) \parallel^2 \\ &= argmin\langle\beta,\beta\rangle - \langle 2\beta, \phi(\vec{x})\rangle + \langle\phi(\vec{x}),\phi(\vec{x})\rangle \\ &= argmin\langle\beta,\beta\rangle - 2f(\vec{x}) + K(\vec{x},\vec{x})\end{aligned}$$

Minimum von $K(\vec{x},\vec{x}) - 2f(\vec{x})$ (Gradientenabstieg) kann das Pre-Image von $\vec{\beta}$ liefern (oder ein lokales Minimum).

Pre-Images lernen!

Wenn wir häufiger für den selben Merkmalsraum den Ursprung $\vec{x}$ von $\phi(\vec{x})$ bestimmen wollen, dann lohnt es sich, die umgekehrte Abbildung $\Gamma : \mathcal{X} \rightarrow X$ zu lernen.
Allerdings müssen wir dann für den Merkmalsraum eine geeignete Basis finden, z.B. durch eine Hauptkomponentenanalyse mit Kernfunktion.
Auf dieser Basis wird dann für eine kleinere Menge $\vec{x_i}$ die Abbildung Γ approximiert.
Bair, Weston, Schölkopf (2003) Learning to find pre-images, in: NIPS, vol. 16

Reduced Set Problem

Reduced Set Problem

Gegeben die Abbildung $\phi : X \to \mathcal{X}$ und eine natürliche Zahl s, finde $\vec{z_1}, ..., \vec{z_s} \in X$ und Koeffizienten $\gamma_1, ..., \gamma_s$ so dass $\| \vec{\beta} - \sum_{i=1}^{s} \gamma_i \phi(\vec{z_i}) \|^2$ minimal ist.

Das gelernte $\vec{\beta} = \sum_{i=1}^{N} \alpha_i \phi(\vec{x})$ ist eine Linearkombination der Stützvektoren. Diese sind die erste Lösung des Problems.
Wir wollen aber nicht alle Daten bearbeiten, sondern nur $s << N$!
Wir wollen $\vec{\gamma}$ aus weniger Beispielen lernen. Das ist möglich, weil hier nicht die Nebenbedingungen gelten wie bei dem Optimierungsproblem der SVM.

Neues Optimierungsproblem

SVM liefert $\vec{\beta} = \sum_i^N \alpha_i \phi(\vec{x_i})$
Wir wollen die Distanz der Approximation zur originalen SVM minimieren:

$$\| \sum_{i=1}^{N} \alpha_i \phi(x_i) - \sum_{i=1}^{N} \gamma_i \phi(x_i) \|^2 + \lambda \sum |\gamma_i|$$

und γ soll spärlich besetzt sein. $\lambda > 0$ gewichtet die Spärlichkeit gegen die Präzision.

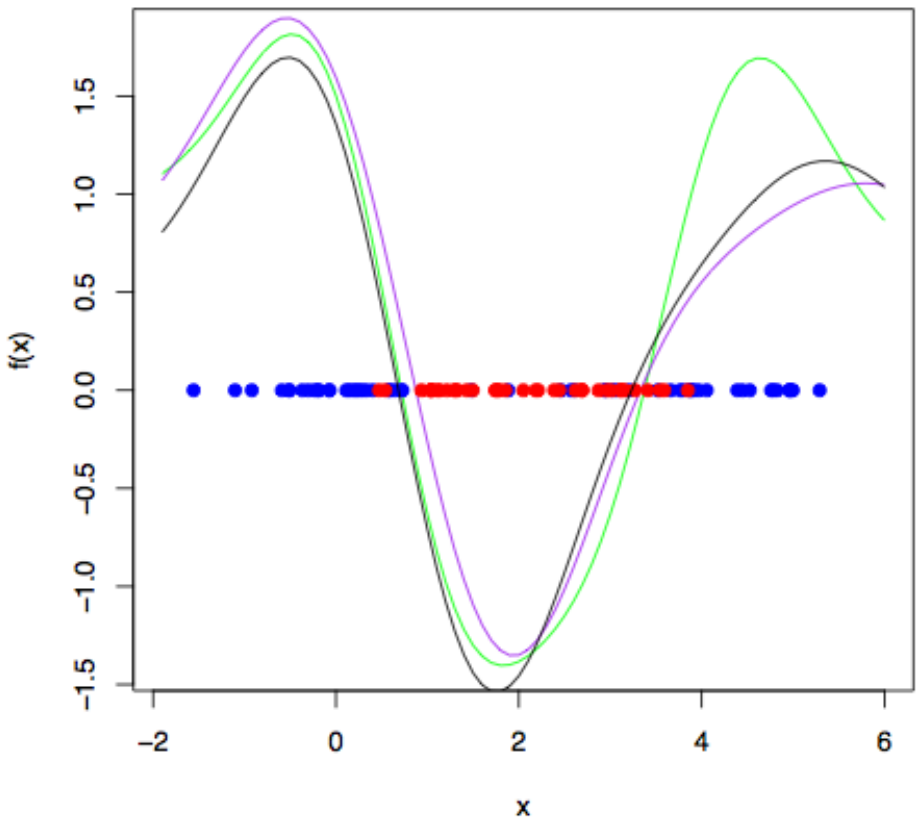
Schölkopf, Mika, Burges, Knirsch, Müller, Rätsch, Smola (1999)
Input space versus feature space in kernel-based methods.
IEEE Neural Networks Vol.10, No. 5

Iterativer Algorithmus – Skizze

1. k=:0; $Z_k =: \{\}$
2. finde z_k, so dass gilt $\mathbf{K}^{zx}\alpha = \mathbf{K}^{z}\gamma$
   Kernmatrix $\mathbf{K}^{zx} = \langle \phi(\vec{z_i}), \phi(\vec{x_i}) \rangle$,
   Kernmatrix $\mathbf{K}^{z} = \langle \phi(\vec{z_i}), \phi(\vec{z_j}) \rangle$ der neue Punkt $\vec{z}$ mit γ verhält sich wie mit β bei allen Beispielen.
3. k=: k+1; $Z_k =: Z_{k-1} \cup z_k$
4. berechne $\gamma = (\mathbf{K}^{z})^{-1}\mathbf{K}^{zx}\alpha$
5. Wenn $\| \sum_{i=1}^{N} \alpha_i \phi(x_i) - \sum_{i=1}^{N} \gamma_i \phi(x_i) \|^2 + \lambda \sum |\gamma_i| < \theta$, stop, sonst Schritt 2!

Reduced Set Approximation – Bild

Bei einem eindimensionalen Datensatz mit Klassen blau und rot, sieht man die Funktionswerte der tatsächlichen Funktion (grün), die Approximation lt. Schölkopf et al (1999) (lila) und die Approximation lt. Rüping (2006) (schwarz):



Was wissen Sie jetzt?

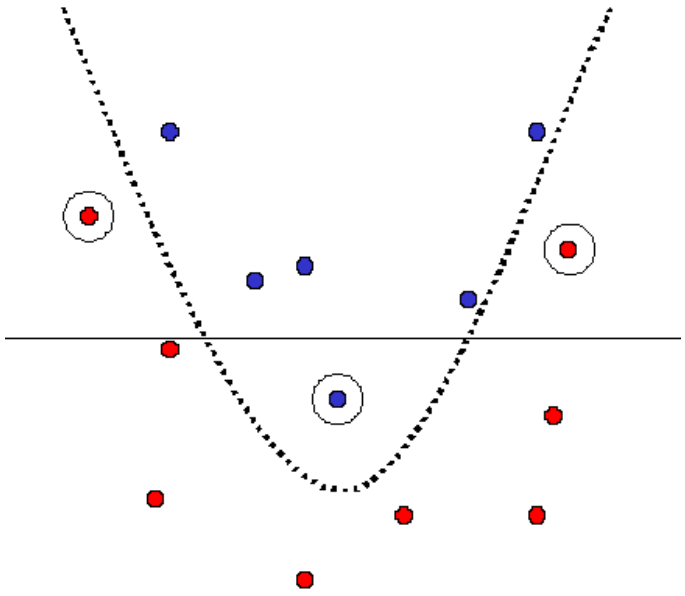
- Lineare SVM sind leicht zu interpretieren: α gewichtet Beispiele, β gewichtet Merkmale.
- Bei Kernfunktionen wissen wir für gegebene Wert $\phi(\vec{x})$ nicht, welches $\vec{x}$ dahinter steht.
- Ansatz: zu einer SVM noch eine Approximation der SVM lernen!
  - Die gelernte SVM klassifiziert mit max margin.
  - Die Approximation gibt eine Vorstellung von der Funktion.
  - Das Reduced Set Problem findet eine Approximation für wenige Beispiele mit γ statt β auf der Grundlage eines gelernten Modells.

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!

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:
  $R(\alpha) \leq_\eta R_{emp}(\alpha) + Var(\alpha)$



Risikoschranke nach Vapnik

Strukturelles Risiko

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

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

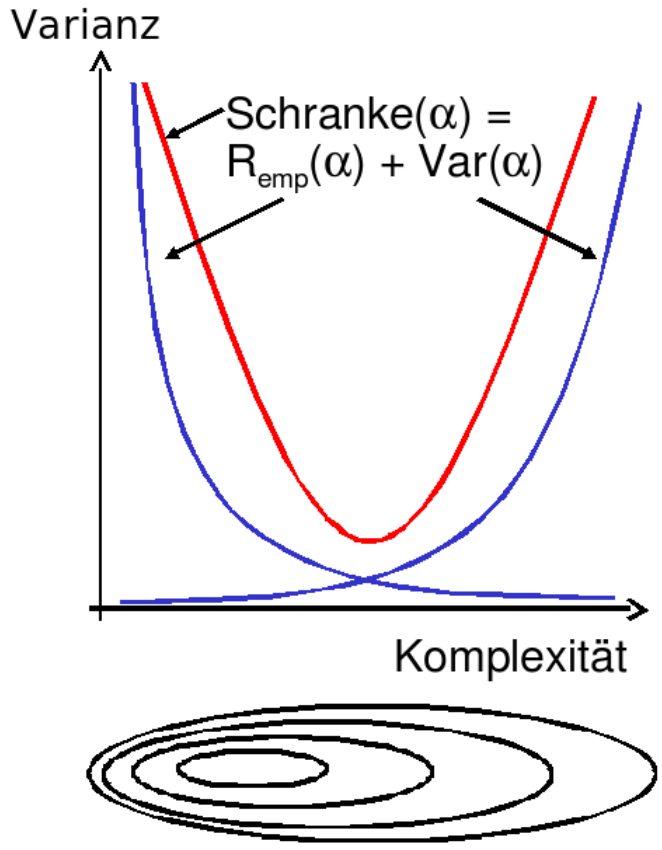
Bevor wir η ergründen (Vapnik-Chervonenkis-Dimension), erst einmal festhalten, was die Bedeutung dieser Schranke ist!

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 η kennen, die Vapnik-Chervonenkis-Dimension.
- Gegeben eine Menge Hypothesen für $f(\vec{x}, \vec{\alpha})$, wähle immer die mit dem niedrigsten Wert für die rechte Seite der Schranke (R_{emp} oder VC confidence niedrig).

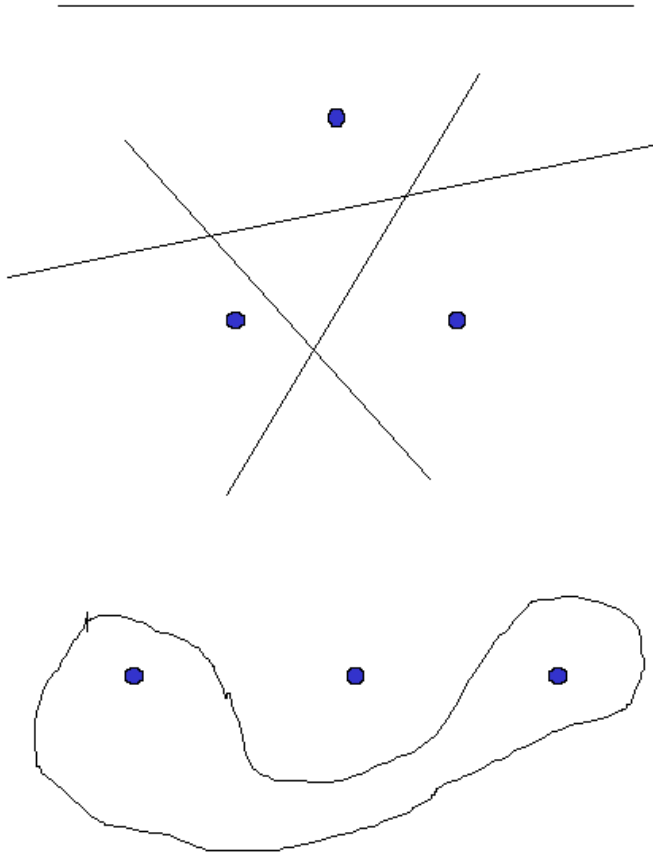
Strukturelle Risikominimierung

1. Ordne die Hypothesen in Teilmengen 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.



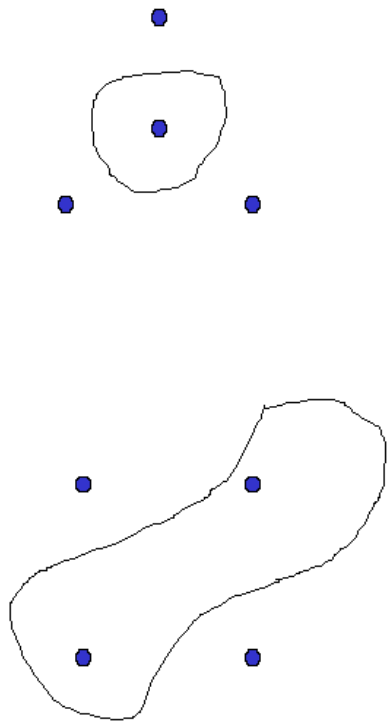
ACHTUNG

- 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 R^p ist $p+1$.
Beweis:

- $VCdim(R^p) \geq p+1$: Wähle $\vec{x_0} = 0$ und $\vec{x_i} = (0, \ldots, 0, 1, 0, \ldots 0)$. Für eine beliebige Teilmenge A von $(\vec{x_0}, \ldots, \vec{x_n})$ setze $y_i = 1$, falls $\vec{x_i} \in A$, sonst $y_i = -1$.
  Definiere $\vec{\beta} = \sum y_k \vec{x_k}$ und $\beta_0 = \frac{y_0}{2}$.
  Dann gilt $\vec{\beta}\vec{x_0} + \beta_0 = \frac{y_0}{2}$ und $\vec{\beta}\vec{x_i} + \beta_0 = y_i + \frac{y_0}{2}$.
  Also: $\vec{\beta}\vec{x} + \beta_0$ trennt A.
- $VCdim(R^p) \leq p+1$: 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-Dimension der SVM

- Gegeben seien Beispiele $\vec{x_1}, \ldots, \vec{x_N} \in \mathcal{R}^p$ mit $\|\vec{x_i}\| < D$ für alle i. Für die VC-Dimension der durch den Vektor $\vec{\beta}$ gegebenen optimalen Hyperebene H gilt:

$$VCdim(H) \leq \min\left\{D^2\|\vec{\beta}\|^2, p\right\} + 1$$

- Die Komplexität einer SVM ist auch durch die Struktur der Lösung begrenzt!
- Die SVM minimiert nicht nur das empirische Risiko, sondern auch das strukturelle – Regularisierung.

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.
- Wir haben nun also ein Verfahren, dass ohne zusätzlichen Aufwand die Komplexität regularisiert, wie wir es bei der Modellselektion für lineare und lokale Modelle mal wollten.

Performanzschätzer

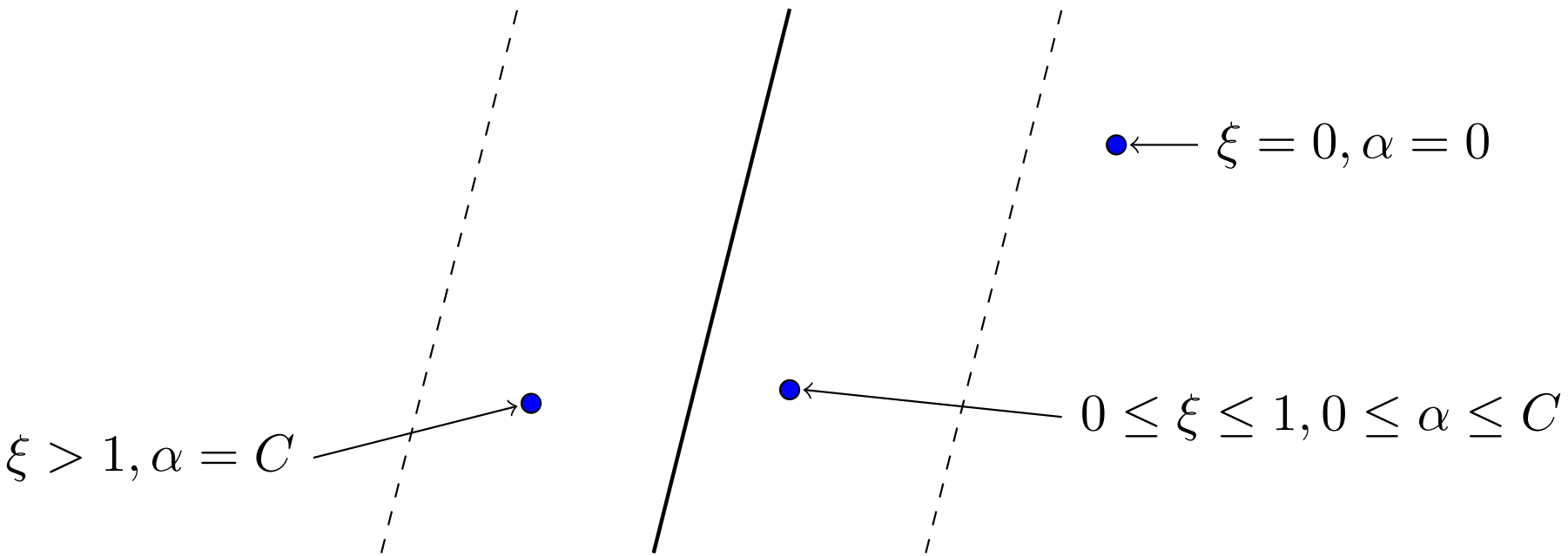
- Welches erwartete Risiko $R(\alpha)$ erreicht SVM?
- $R(\vec{\alpha})$ selbst nicht berechenbar
- Trainingsfehler (zu optimistisch - Overfitting)
- Obere Schranke mittels VC-Dimension (zu locker)
- Kreuzvalidierung / Leave-One-Out-Schätzer (ineffizient)

Performanzschätzer II

- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch $R_{l1o} \leq \frac{|SV|}{N}$
- Beweis (Skizze):
  - Falsch klassifizierte Beispiele werden Stützvektoren (SV).
  - Also: Nicht-Stützvektoren werden korrekt klassifiziert. Weglassen eines Nicht-Stützvektors ändert die Hyperebene nicht, daher wird es auch beim $l1o$-Test richtig klassifiziert.
  - Nur der Anteil der Stützvektoren an den Beispielen macht den Fehler aus.

Performanzschätzer III

- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch $R_{l1o} \leq \frac{|\{i:(2\alpha_i D^2+\xi_i)\geq 1\}|}{N}$ (D = Radius des Umkreises um die Beispiele im transformierten Raum).
- Beweis: Betrachte folgende drei Fälle:



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 Regularisierung:
  - obere Schranke für das Risiko
  - Schrittweise Steigerung der Komplexität
- Formalisierung der Komplexität: VC-Dimension
- Regularisierung als strukturelle Risikominimierung der SVM
- Garantie für die Korrektheit der Lernstrategie