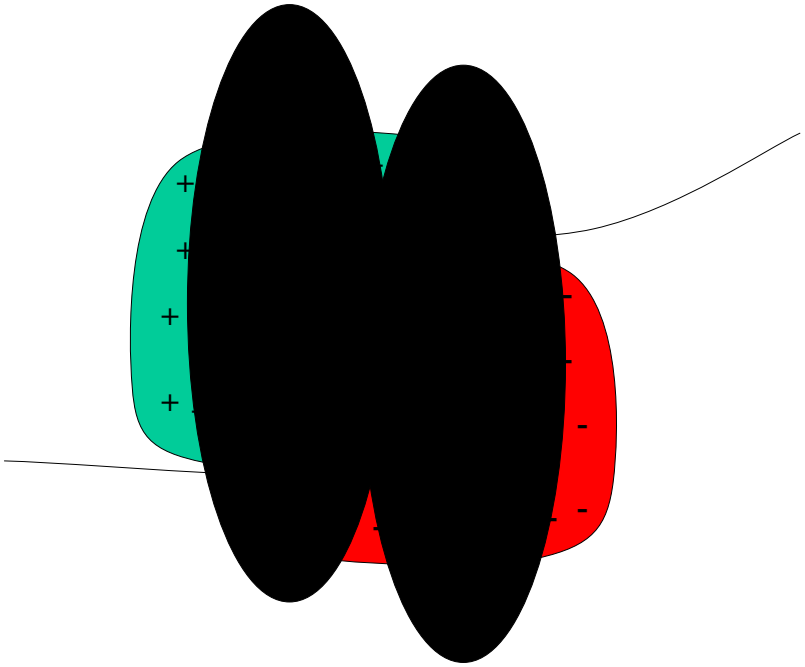
Kapitel 10
Die Support-Vektor-Maschine (SVM)

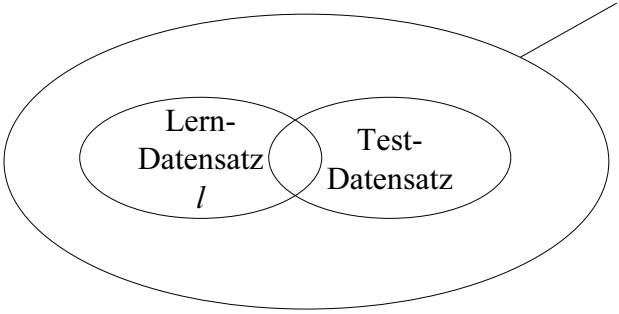
Ein statistischer Ansatz der Lerntheorie zum Entwurf eines optimalen Klassifikators

Inhalt:

Durch Normalverteilungen schlecht darstellbare Klassen



Mustererkennungsexperiment



Stochastischer Prozess mit *unbekannter* Verbundverteilung: $P(\mathbf{x},y)$

Gesucht ist eine Zuordnungsfunktion $f(\mathbf{x},\alpha)$: $\mathbf{x}_i \rightarrow y_i$ welche durch die unbekannten Parameter des Vektors α charakterisiert wird, welche den Erwartungswert des Zuordnungsfehlers minimiert:

$R(\alpha)=E\{R_{test}(\alpha)\}$

Empirischer Fehler $R_{emp}(\alpha)$

Test-Fehler (Generalisierungsfähigkeit) $R_{test}(\alpha)$

Lerntheoretischer Ansatz

Überwachtes Lernen:

- Gegeben: l Beobachtungen (Lernstichprobe) aus dem Mustererkennungs-experiment mit der Verbundverteilung $P(\mathbf{x},y)$ mit den dazugehörigen Klassenzuordnungen (Labels) (Zunächst Beschränkung auf Zweiklassenproblem); ansonsten existiert *keinerlei Vorwissen*:

$$\{(\mathbf{x}_i, y_i) \in P(\mathbf{x}, y)\} \quad i = 1,\ldots,l \quad \text{mit: } \mathbf{x}_i \in \mathbb{R}^N, \quad y_i \in \{+1,-1\}$$

Gesucht:

- deterministische Zuordnungsfunktion, $f(\mathbf{x},\boldsymbol{\alpha})$: $\mathbf{x} \rightarrow y$ aufbauend auf eine Lernstich-probe, welche den *Erwartungswert des Zuordnungsfehlers (expected risk)* minimiert:

$$R(\boldsymbol{\alpha}) = E\{R_{test}(\boldsymbol{\alpha})\} = E\left\{\tfrac{1}{2}\left|y - f(\mathbf{x},\boldsymbol{\alpha})\right|\right\} = \int \tfrac{1}{2}\left|y - f(\mathbf{x},\boldsymbol{\alpha})\right| dP(\mathbf{x},y)$$

$$= \int \tfrac{1}{2}\left|y - f(\mathbf{x},\boldsymbol{\alpha})\right| p(\mathbf{x},y)\, d\mathbf{x}dy \quad \text{falls Verbundverteilungsdichte bekannt}$$

Problem: dieser Ausdruck kann jedoch nicht ausgewertet werden, da $P(\mathbf{x},y)$ nicht zur Verfügung steht und ist somit wenig hilfreich!

Zentrales Problem der statistischen Lerntheorie: Wann führt ein niedriger Trainingsfehler zu einem niedrigen echten Fehler?

- Das empirische Risiko, nämlich die Fehlerrate für einen gegebenen Trainingsdatensatz lässt sich leicht berechnen gemäß:

$$R_{emp}(\boldsymbol{\alpha}) = \tfrac{1}{2l}\sum_{i=1}^{l}\left|y_i - f(\mathbf{x}_i,\boldsymbol{\alpha})\right|$$

- Der Ansatz mit einem Neuronalen Netz und dem Backpropagation Learning z.Bsp. begnügt sich mit einer Minimierung des empirischen Risikos (ERM)
- Hier nun die Frage: Wie nah ist man nach l Trainingsbeispielen am *echten* Fehler? Und: Wie gut kann aus dem empirischen Risiko das echte Risiko abgeschätzt werden (Structural Risc Minimization (SRM) anstatt Empirical Risc Minimization (ERM)) ? Dies beinhaltet die Generalisierungsfähigkeit!
- Eine Antwort darauf gibt die Lerntheorie von Vapnik-Chervonenkis!

Eine Obergrenze für die Generalisierungsfähigkeit des Lernens mit der VC-Theorie

(Vapnik/Chervonenkis)

Mit der Wahrscheinlichkeit (1-η) gilt die folgende obere Abschätzung für den tatsächlichen Fehler (d.h. z.Bsp. mit η = 0,05 und damit mit einer Wahrscheinlichkeit von 95%):

$$R(\boldsymbol{\alpha}) \le R_{emp}(\boldsymbol{\alpha}) + \underbrace{\Phi(h,l,\eta)}_{\text{VC-Konfidenz}}$$

$$\text{mit: } \Phi(h,l,\eta) = \sqrt{\frac{h\left(\log\frac{2l}{h}+1\right) - \log\left(\frac{\eta}{4}\right)}{l}}$$

- l Anzahl der Trainingsbeispiele
- h VC-Dimension des verwendeten Hypothesenraums

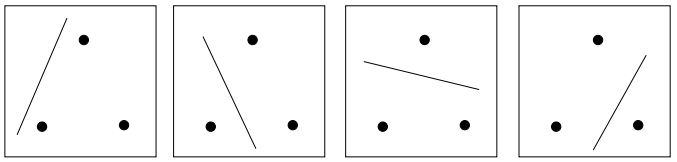
Bemerkenswert ist, dass dieser Ausdruck unabhängig ist von der zugrundeliegenden Verbundverteilung P(x,y) ! (vorausgesetzt, die Trainings- und Testdatensätze werden statistisch unabhängig gezogen). D.h. falls man die Auswahl zwischen verschiedenen Lernmaschinen hat (einhergehend mit speziellen Familien von Abbildungsfunktionen f(**x**,α)), entscheidet man sich für die Maschine, welche den geringsten Wert bei gegebener VC-Dimension für die Konfidenz Φ liefert.

Die VC-Dimension ist eine Eigenschaft für gegebene Funktionenklassen $\{f(\alpha)\}$

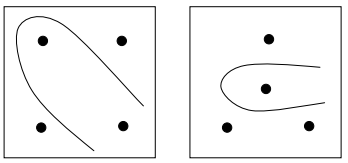
- Eine gegebene Menge von l Punkten kann für den Zweiklassen-Fall in 2^l mögliche Klassen aufgeteilt werden. Die VC-Dimension einer Menge von Funktionen $\{f(\alpha)\}$, ist definiert als die maximale Anzahl von Trainingspunkten, welche durch diese Klasse von Funktionen in allen möglichen Zuordnungen separiert werden können.
- Liefert Maße für „Kapazität“ von Funktionenklassen
  - Funktionenklassen z.B.
    - Menge der linearen Klassifikatoren
    - Menge der Klassifikatoren, die ein NN realisieren kann
    - Menge der Klassifikatoren, die eine SVM realisieren kann
- Die VC-Konfidenz wächst monoton mit h: Demnach würde man bei einer Auswahl von Lernmaschinen, deren empirisches Risiko 0 ist, sich für diejenige entscheiden, deren assoziierte Menge von Abbildungsfunktionen minimale VC-Dimension besitzt.

VC-Dimension von Hyperebenen in $\mathbb{R}^2$

- Drei Punkte in $\mathbb{R}^2$ lassen sich mit Hyperebenen (Geraden) in allen Konstellationen trennen



- Vier Punkte in $\mathbb{R}^2$ lassen sich nicht mehr mit Hyperebenen in allen Konstellationen trennen



  - => Hyperebenen in $\mathbb{R}^2$: VCdim=3
  - Allgemein: Hyperebenen in $\mathbb{R}^N$: VCdim=N+1
  - die VC-Dimension bei Polynomen wächst mit ihrem Grad!

VC-Dimension von SVM

– Aussagen über VC-Dimensionen bei SVM möglich!
  - VC-Dimension der Menge der Hyperebenen in $\mathbb{R}^M$

$$h \le \min(R^2 A^2, M) + 1$$

$$\|w\| \le A,\ K(x_i, x_i) \le R^2$$

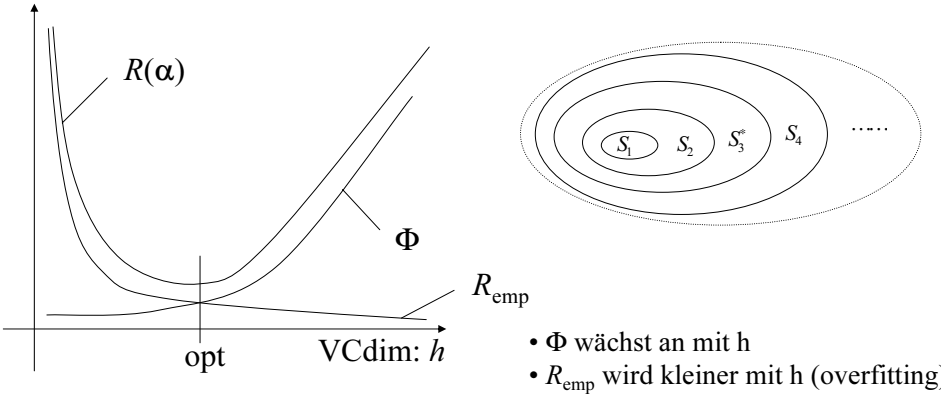
  - SVM mit Polynomkernen $K(x,y)=(x \cdot y)^p$

$$h = \binom{N+p-1}{p} + 1$$

– Hiermit “Structural Risk Minimization“ möglich

Structural Risk Minimization (SRM)

- SRM ist Minimieren der Abschätzung für $R(\alpha)$ über anwachsende Funktionenklassen S_i . Diese bilden Hypothesenräume mit anwachsender VC-Dimension VCdim=h_i ($h_1<h_2<h_3<...$).
- Kompromiss zwischen empirischem Risiko und Generalisierungsfähigkeit



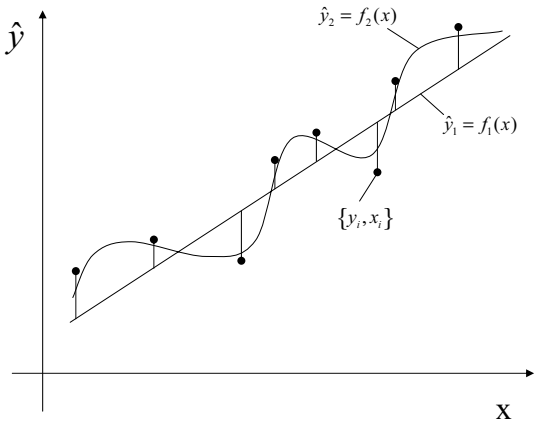
- Φ wächst an mit h
- R_{emp} wird kleiner mit h (overfitting)

Entwurfshinweise

Mehrere Ausdrücke für das gleiche Phänomen:

- Bias/Varianz-Kompromiss
- Generalsierungs/Overfitting-Kompromiss
- Empirischer Fehler/VC-Dimensions-(Kapazitätskontrolle) -Kompromiss

Bias-Varianz-Tradeoff



f_1 : Bias $\frac{1}{n}\sum_{i=1}^{n}(y_i - f_1(x_i))$ klein; Varianz $\frac{1}{n}\sum_{i=1}^{n}(y_i - f_1(x_i))^2$ groß

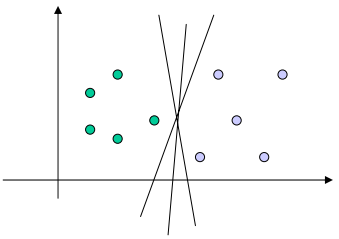
gute Generalisierung (einfaches Modell)!

f_2 : Bias $\frac{1}{n}\sum_{i=1}^{n}(y_i - f_2(x_i))$ groß; Varianz $\frac{1}{n}\sum_{i=1}^{n}(y_i - f_2(x_i))^2$ klein

schlechte Generalisierung (overfitting)!

Linear separierbare Klassen

- Daten in $\mathbb{R}^2$



  - Allgemeine Hyperebene $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$
  - Klassifikation via $f(x) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$
  - z.B. Rosenblatt´s Perceptron (1956)

    Iteratives Lernen, Korrektur nach jeder Fehlklassifikation

    ->keine Eindeutigkeit der Lösung