Statistische Lerntheorie

Theoretische Grundlage der Support-Vektor-Maschinen

Überblick

1. Motivation
2. Begriffe
3. Kapazität und VC-Dimension
4. Fehler, Schranken und ihre Minimierung
5. Structural Risk Minimization
6. Beziehung zu den SVMs
7. Zusammenfassung
8. Literatur

Motivation

- Warum Lerntheorie?
  - Versuch, allgemeine Probleme beim Lernen formal zu fassen und zu lösen
- Hauptproblem (und Thema dieses Vortrags):

  Wie gut ist die Generalisierungsleistung einer Lernmaschine?

  - Wir werden sehen: Hängt zusammen mit ihrer Kapazität
    - Kapazität zu groß: Overfitting
    - Kapazität zu klein: Underfitting

Begriffe

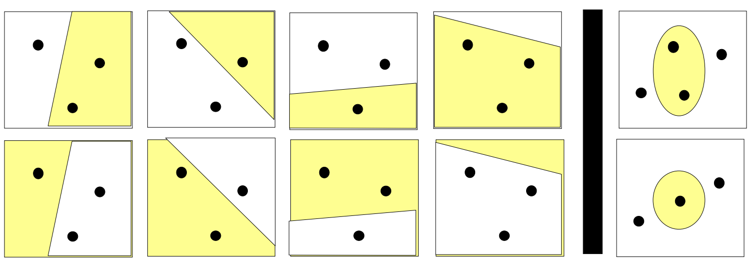
- Trainingsdaten: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \in \mathcal{X} \times \{\pm 1\}$
  - $\mathbf{x}_i$: die Samples, y_i : die Labels
  - der Einfachheit halber nur „Mustererkennungsfall", $y \in \{\pm 1\}$
  - aus unbekannter Wahrscheinlichkeitsverteilung P(x,y)
- Lernmaschine: Eine Menge von Funktionen $\{f(\alpha)\}$ mit $f(\alpha) : \mathcal{X} \rightarrow \{\pm 1\}$
  - α: Menge von Parametern, eine bestimmte Wahl von α generiert eine „Trainierte Maschine"
- Bekanntestes Maß für die Kapazität einer Lernmaschine: Die VC-Dimension

VC-Dimension

- m Samples können auf 2^m Arten „gelabelt" werden.
- Wenn für jede dieser 2^m Arten eine Funktion aus $\{f(\alpha)\}$ existiert, sagt man, die Lernmaschine „zerschmettert" (engl. „to shatter") diese m Punkte
- VC-Dimension h: Die maximale Anzahl von Punkten, die durch die Lernmaschine zerschmettert werden können
  - <u>Achtung</u>: Nicht <u>jede</u> Menge der Mächtigkeit m wird zerschmettert, sondern es existiert mindestens <u>eine</u> Menge von m Punkten, die zerschmettert werden.

Beispiel: Hyperebenen in $\mathbb{R}^2$

- Die Funktionenklasse der trennenden Hyperebenen in $\mathbb{R}^2$ hat eine VC-Dimension von 3



- Allgemein gilt: Die Funktionenklasse der trennenden Hyperebenen in $\mathbb{R}^n$ hat eine VC-Dimension von n+1 (ohne Beweis)

VC-Dimension und die Zahl der Parameter

- Intuitiv könnte man glauben, eine Lernmaschine mit vielen Parametern hat eine hohe VC-Dim, und eine mit wenigen Parametern eine niedrige.
- Dies ist i.a. NICHT der Fall !
- Es gibt Lernmaschinen mit nur <u>einem</u> Parameter und doch unendlicher VC-Dimension
  - Man sagt eine Lernmaschine hat h = ∞, wenn sie m Punkte zerschmettern kann, egal wie groß m ist.

Beispiel

- Sei $x \in \mathbb{R} : \{\theta(x) = 1\ \forall x > 0;\ \theta(x) = -1\ \forall x \leq 0\}$
  $f(x,\alpha) \equiv \theta(\sin(\alpha x)),\quad x,\alpha \in \mathbb{R}$
- Für beliebiges m wähle man m Punkte
  $x_i = 10^{-i},\quad i = 1,\ldots,m$
- Und beliebige Labels $y_1, y_2, \ldots, y_m,\quad y_i \in \{\pm 1\}$
- Dann gibt f(α) diese Labels für

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

also ist h({f (α)}) = ∞

Fehler und ihre Minimierung

- „Actual Risk": Erwarteter Fehler für trainierte Lernmaschine $$R(\alpha) = \int \frac{1}{2} \left| y - f(\mathbf{x}, \alpha) \right| dP(\mathbf{x}, y)$$

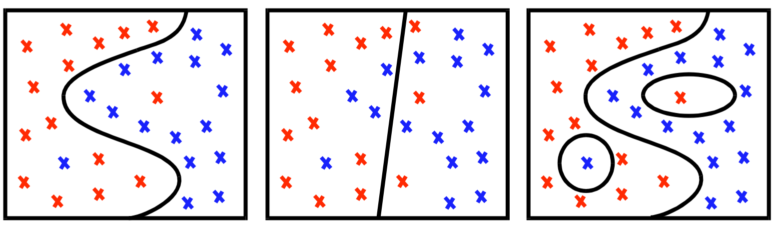
$\rightarrow$ Diesen wollen wir minimieren, aber: leider nicht berechenbar, da P(x,y) unbekannt

- Was können wir berechnen?
- „Empirical Risk": $$R_{emp}(\alpha) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left| y_i - f(\mathbf{x}_i, \alpha) \right|$$

$\rightarrow$ Fehler auf den Trainingsdaten

ERM allgemein nicht ausreichend

- $R_{emp}(\alpha)$ minimieren führt aber nicht automatisch zu kleinem $R(\alpha)$!



- Ziel ist also, eine Lernmaschine von angemessener Kapazität zu finden

Eine Schranke für R(α)

- Statistische Lerntheorie gibt einige Schranken, um R(α) nach oben abzuschätzen.
- Eine wichtige ist: $R(\alpha) \leq R_{emp}(\alpha) + \phi(h,m,\delta)$

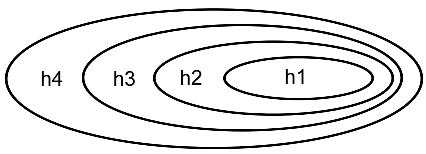
  (Vapnik, 1995)

  wobei $\phi(h,m,\delta) = \sqrt{\frac{1}{m}\left(h\left(\ln\frac{2m}{h}+1\right)+\ln\frac{4}{\delta}\right)}$

  - φ(h,m,δ) wird VC-Konfidenz genannt.
  - Die Schranke hält mit Wahrscheinlichkeit δ
  - Sie ist unabhängig von P(**x**,y)
  - Sie ist leicht zu berechnen, wenn wir h kennen

Structural Risk Minimization

- Wie können wir nun aber $R_{emp}(\alpha)$ + φ(h,m,δ) (den „Risk Bound") minimieren?
- Die VC-Konfidenz ist abhängig von der Funktionsklasse, nicht von einer bestimmten Wahl von α, kann also nicht so einfach minimiert werden
- Idee: Wir führen eine Struktur auf {f(α)} ein

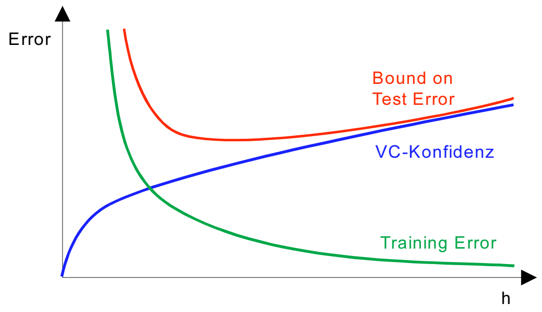


Ineinander geschachtelte Untermengen, mit absteigender VC-Dimension

h4 > h3 > h2 > h1

Structural Risk Minimization 2

- Jetzt können wir den Risk Bound minimieren, indem wir für jede Unterklasse R_{emp} minimieren



- Dieses Prinzip heißt Structural Risk Minimization (SRM)

Leider kein Allheilmittel...

- Risk Bound $R_{emp}(\alpha) + \phi(h,m,\delta)$ theoretisch schön, aber praktisch nicht immer nützlich
  - Macht oft keine nichttrivialen Aussagen
  - Gilt nicht für Lernmaschinen mit $h = \infty$
  - Trotzdem können Lernmaschinen mit $h = \infty$ eine gute Performance haben: Beispiel k-Nearest Neighbour Klassifikator mit k=1
    - VC-Dimension unendlich
    - $R_{emp} = 0$ (falls Trainingsmenge widerspruchsfrei)
- VC-Dimension als Kapazitätsmaß oft ein bisschen grob

Was es sonst noch gibt

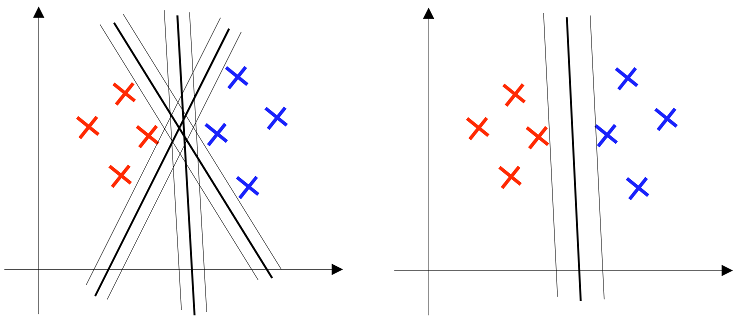
- Es gibt noch andere, engere Bounds
  - Benutzen teilweise andere Kapazitätsmaße
    - VC-Entropie
    - Annealed Entropy
    - Growth Function
  - Sind teilweise nicht mehr verteilungsunabhängig
  - Geht über den Rahmen dieses Vortrags hinaus...
- Stattdessen: Bezug der Theorie zu SVMs

Beziehung zu den SVMs:

- SVMs und Neurnonale Netze als Instanzen von Lernmaschinen:
  - SVMs halten R_{emp} fest und versuchen, die VC-Konfidenz zu minimieren
  - Im Gegensatz dazu: Neuronale Netze halten VC-Konfidenz konstant (durch die Wahl der Struktur) und versuchen, R_{emp} zu minimieren
- Funktionenklasse die SVMs zugrunde liegt, sind die separierenden Hyperebenen im $\mathbb{R}^n$
- SVMs machen SRM, indem sie die optimal (mit maximalem Rand) separierende Hyperebene bestimmen
- Warum ist das so? Formal kompliziert, daher nur eine Anschauung...

Optimal separierende Hyperebene

- Indem ein Constraint über die Breite des Randes auf die Funktionenklasse der separierenden Hyperebenen gelegt wird, beschränkt man deren Kapazität



Zusammenfassung

- Wir haben das allgemeine Konzept der Lernmaschine und ihrer Trainingsdaten formal gefasst.
- Wir haben ein wichtiges Maß für die Kapazität einer Lernmaschine kennengelernt, die VC-Dimension.
- Wir haben gesehen, dass der empirische Fehler im Allgemeinen kein guter Indikator für den zu erwartenden Fehler ist.

Zusammenfassung 2

- Wir haben den zu erwartenden Fehler, den empirischen Fehler und die Kapazität der Lernmaschine in Beziehung gesetzt und eine allgemeine Abschätzung für den zu erwartenden Fehler, den Risk Bound, kennengelernt.
- Mit dem induktiven Prinzip der Structural Risk Minimization haben wir eine Methode kennengelernt, den Risk Bound zu minimieren
- Wir haben eine Intuition erhalten, wie SVMs auf der Statistischen Lerntheorie aufsetzen und wie die optimal separierenden Hyperebenen zum SRM in Beziehung stehen

Literatur

- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining, 2(2), 1998.
- V. Vapnik: The Nature of Statistical Learning Theory, Springer-Verlag New York, 1995.
- B. Schölkopf, A. Smola: Learning With Kernels, MIT-Press Cambridge, 2002.