Modell-Selektion bei SVM

Janis Fehr

Institut für Informatik

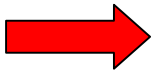
Universität Freiburg

WS2002

Übersicht

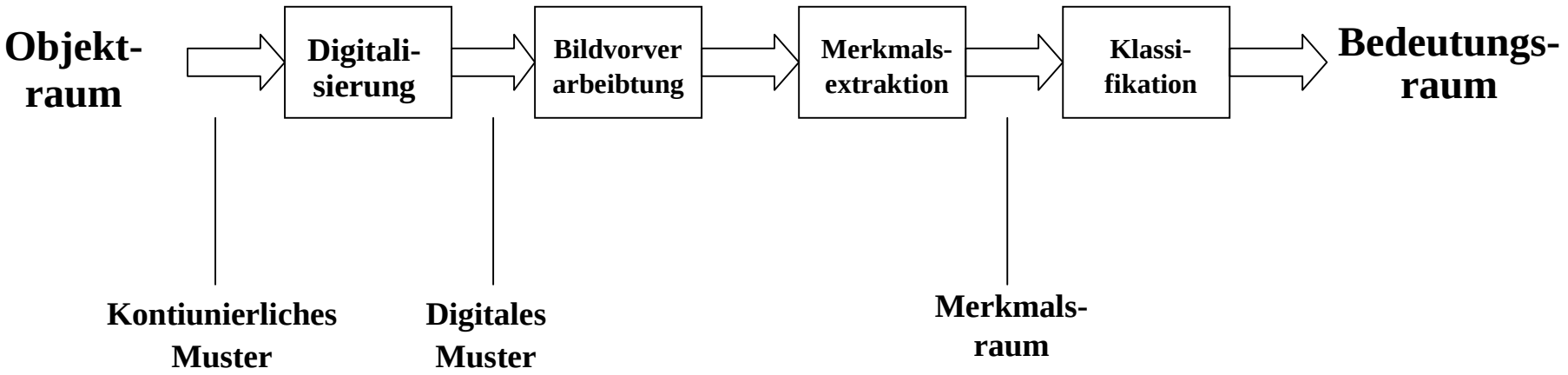
- Motivation
- Maße für die Modell Qualität
- Allgemeine Verfahren
- SVM Verfahren
- Implementierungen
- Literatur

Übersicht

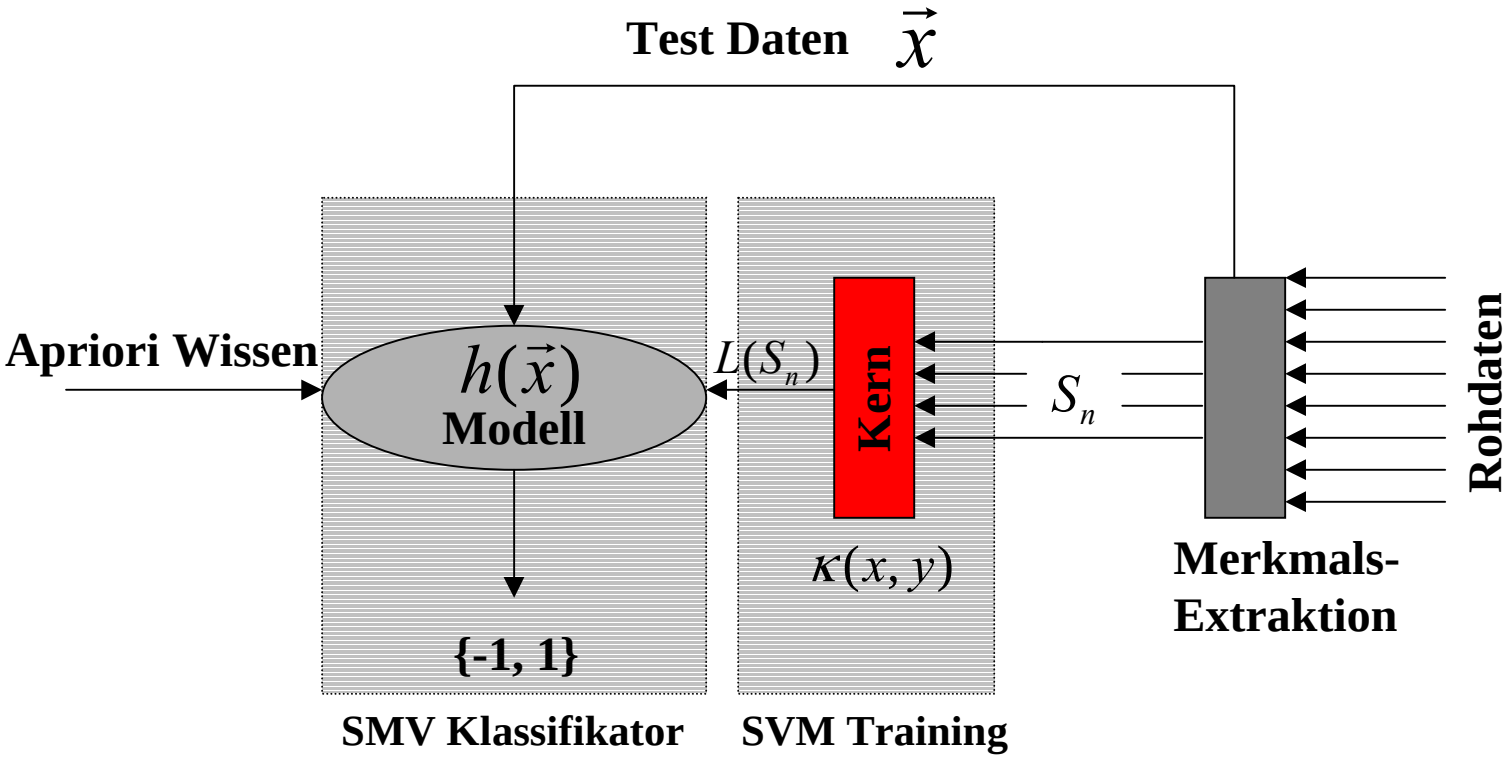


- **Motivation**
- **Maße für die Modell Qualität**
- **Allgemeine Verfahren**
- **SVM Verfahren**
- **Implementierungen**
- **Literatur**

Mustererkennungskette



SVM Modell-Bildung



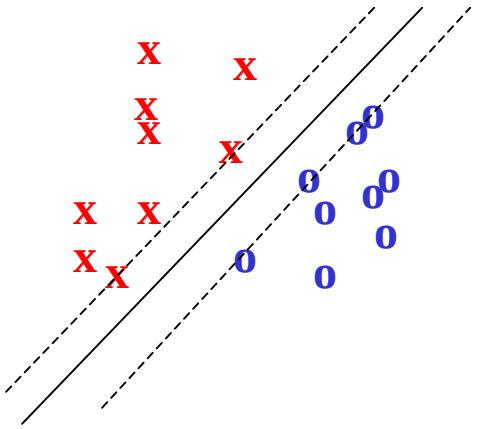
$$S_n = \{(\vec{x}_i, y_i)\}$$

SVM Training : Parameter

Duales Maximierungsproblem :

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

SVM



mit

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

$$C \geq \alpha_i \geq 0$$

Kern Parameter:

Linear $K(x, y) = x \cdot y$

Polynomial $K(x, y) = (x \cdot y + c)^d$

Gauß $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$

Sigmoid $K(x, y) = \tanh(k(x \cdot y) + \theta)$

Nur wenige Parameter beeinflussen das Modell ...

Auswahl der Modell Parameter

Probleme:

- **Was ist ein gutes Modell ?**
- **Lerndaten sind „teuer“ und nur beschränkt verfügbar.**

$$S_n = \{(\vec{x}_i, y_i) \mid i = 1...n, y \in \{-1,1\}\}$$

Was ist ein gutes Modell ?

Intuitiv: **„Ein Modell, dass alle Testdaten richtig klassifiziert“**

Formal: **Nicht trivial, da S_n endlich. Menge M der möglichen Testdaten aber in der Regel unendlich oder zumindest sehr viel größer.**

Abschätzung:

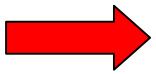
$$S_n \rightarrow h(M)$$

Generalisierung

Definition: Modell-Selektion

„Wahl der Modellparameter zur Optimierung der geschätzten Modell-Güte von $h(M)$ anhand der zur Verfügung stehenden Lerndatensätze S_n .“

Übersicht



- **Motivation**
- **Maße für die Modell Qualität**
  - **Rec(h)**
  - **Prc(h)**
  - **F1**
  - **VC-Dimension**
- **Allgemeine Verfahren**
- **SVM Verfahren**
- **Implementierungen**
- **Literatur**

Maße für Modell Qualität

Geg. : **2 Klassen Problem mit** $h(\vec{x}) = \{-1,1\}$

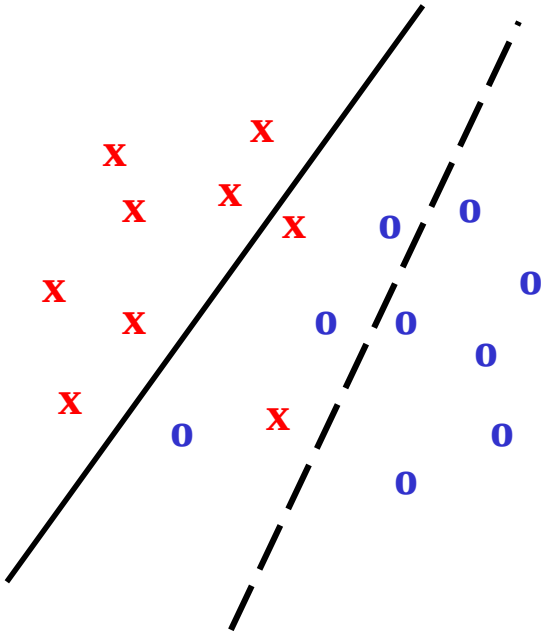
Recall :Rec(h) := **Wahrscheinlichkeit für** $(\vec{x},1) \Rightarrow h(\vec{x}) = 1$

Precision: Prec(h) := **Wahrscheinlichkeit für** $h(\vec{x}) = 1 \Rightarrow (\vec{x},1)$

Sensitivität / Spezifität

Trade off: Rec(h) vs. Prec(h)

Geg. : 2 Klassen Problem mit $h(\vec{x}) = \{-1,1\}$ **x=(-1) o=1**



Rec(h) gut,
Prec(h) schlecht.

Rec(h) schlecht,
Prec(h) gut.

F1 - Wert

- **Prec(h) und Rec(h) alleine sind nur wenig aussagekräftig.**

- **F1 - Wert als geometrisches Mittel**

$$F_1 = \frac{\mathrm{Prec}(h) + \mathrm{Rec}(h)}{2}$$

Neben F_1 als Modell-Quallität, wird auch die Fehlerrate $Err(h)$ eines Modells zur Modell-Selektion herangezogen.

Maß für die Mächtigkeit des Modells (eng. model power)

Wie hoch ist der Fehler der erwarteten Modell-Qualität ?

Ist das Modell mächtig genug für die Komplexität des Problems ?

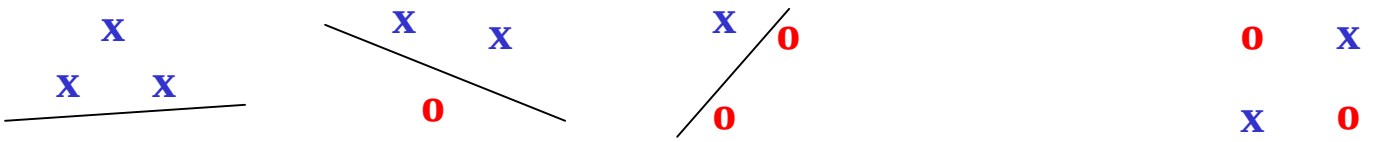
Wie Mächtig ist das Modell ?

=> VC-Dimension als Maß der Mächtigkeit eines Modells

VC - Dimension (Vapnik, Cherrovenkis)

Intuitiv: **„Die VC-Dimension gibt die Anzahl der vom Modell (unabhängig von der Belegung) garantiert separierbaren Datensätze an.“**

Beispiel: **geg. linearer Kern im** R^2



$$VC - Dim(R^2) = 3$$

VC - Dimension (Vapnik, Cherrovenkis)

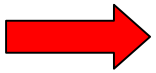
Formal : **Sei** $\Psi(\vec{p}) := \{\psi(\vec{x}, \vec{p}) \mid \psi(\vec{x}, \vec{p}) = \{-1,1\}\}$ **Die Menge aller Möglichen Funktionen eines Modells, mit den Testdaten** $\vec{x}$ **und Parametern** $\vec{p}$

$$VC - Dim(\Psi(\vec{p})) := |\, i \,| \text{ mit } 1..i : \forall \vec{x}_i \exists \psi(\vec{x}_i, \vec{p}) = \{-1,1\}$$

Lemma:

linear	**nicht linear**
für $R^n : VC - Dim(\Psi(\vec{p})) = n+1$	für $H : VC - Dim(\Psi(\vec{p})) = Dim(H)+1$

Übersicht



- Motivation
- Maße für die Modell Qualität
- Allgemeine Verfahren
  - Training Error
  - Hold-Out Testing
  - Cross-Validation
  - Leave-One-Out (loo)
- SVM Verfahren
- Implementierungen
- Literatur

Allgemeine Verfahren zur Modell-Selektion

Einige Definitionen:

$Err^n(h)$ **=: die „wahre“ Fehlerrate eines Modells** h **bei n Trainingssätzen.**

$$S_n := ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$$

Trainingssatz

$$L(h(\vec{x}), y) =: \begin{cases} 1 & h(\vec{x}, y) \neq y \\ 0 & \text{sonst} \end{cases}$$

0/1-loss Funktion.

Training - Error

Trainiere Modell h mit allen n Trainingsdatensätzen und setze sie mit der Anzahl der falsch klasifizierten $\vec{x}$ ins Verhältnis.

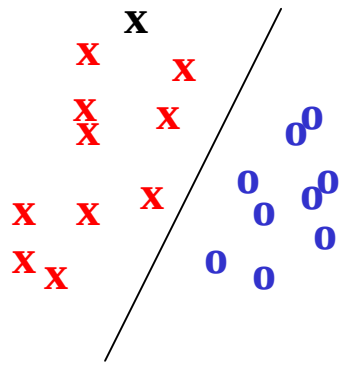
$$Err_{emp}^{n}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\vec{x}_i), y_i)$$

Keine Generalisierung !

Wie gut ist die Schätzung von $Err^n(h)$ durch $Err^n_{emp}(h)$?

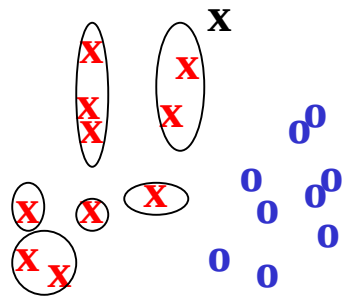
Qualität hängt von der Mächtigkeit des Modells und der Anzahl der Trainingssätze ab.

Einfaches Modell,



$$VC - Dim(h) = 3$$

Komplexes Modell



Overfitting !

$$Err^n_{emp} << Err^n$$

Abschätzung von Err^n durch Err^n_{emp}

$$Err^n(h) \leq Err^n_{emp}(h) + 2\sqrt{\frac{VC - Dim(\ln\frac{2n}{Vc - Dim(h)} + 1) - \ln\frac{\eta}{4}}{n}}$$

Obere Schranke nach [Wapnik, Tscherwonenkis, 1979]

Hold-Out-Testing

Teile die zur Verfügung stehenden Datensätze in Trainingssätze und Testsätze auf.

$$S_l^{train} \bigcup S_k^{test} = S_n$$

$$Err_{ho}^{l,k}(h) = \frac{1}{k} \sum_{(\vec{x}_i, y_i) \in S_k^{test}} L(h(\vec{x}_i, y_i))$$

Wie gut ist die Schätzung von $Err^n(h)$ durch $Err_{ho}^{l,k}(h)$?

Allgemein: $Err^n(h) < Err_{ho}^{l,k}(h)$.

Auf Grund der geringeren Anzahl der Trainingssätze, wird Hold-Out ein schlechteres Modell liefern.

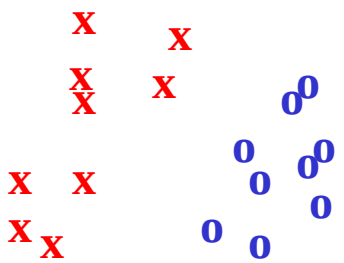
Was ist eine gute Wahl für l und k ?

Cross - Validation

Idee: **führe Hold-Out-Testing nicht nur 1 mal, sondern m=n/k mal aus.**

$$\forall \delta = 1...m: \qquad S_{l,\delta}^{train} \bigcup S_{k,\delta}^{test} = S_n$$

$$Err_{ho}^{k,\delta}(h) = \frac{1}{k} \sum_{(\vec{x}_i, y_i) \in S_\delta^{test}} L(h(\vec{x}_i, y_i))$$



$$\Rightarrow Err_{ho}^{k}(h) = \frac{1}{m} \sum_{\delta=1}^{m} Err_{ho}^{k,\delta}(h)$$

Cross - Validation(2)

- **Verfahren liefert eine gute Schätzung von $Err^n(h)$.**
- **Implementiert in *libSVM***

Aber :

- **Sehr rechenintensiv. Modell muss m mal trainiert werden.**

Alternative: Vereinfachung von Cross-Validation zu Leave-One-Out

Leave-One-Out (loo)

Idee : **Cross-Validation mit k=1, d.h. entferne jeweils den i-ten Trainingssatz aus S_n und trainiere mit dem Rest das Modell $h^{\setminus i}$**

$$Err_{loo}^{n}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h^{\setminus i}(\vec{x}_i, y_i))$$

Eigenschaften:

- **Liefert sehr gute Schätzung von $Err^n(h)$. Da Cross-Validation approximiert wird .**
- **Scheinbar hohe Berechnungskomplexität: Modell wird n-mal berechnet !**

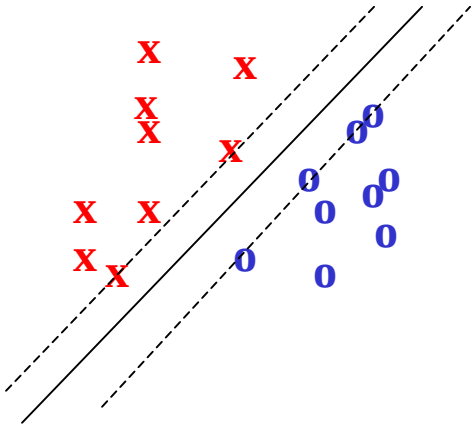
Übersicht

- **Motivation**
- **Maße für die Modell Qualität**
- **Allgemeine Verfahren**
- **SVM Verfahren**
  - **loo**
  - **Vector-Span**
  - $\alpha\xi$ - Estimator
- **Implementierungen**
- **Literatur**

loo bei SVM

loo: $$Err^n_{loo}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h^{\setminus i}(\vec{x}_i, y_i)) \text{ für } \forall \vec{x}_i \in S_n$$

SVM



$$Err^n_{loo-SVM}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h^{\setminus i}(\vec{x}_i, y_i))$$

$$\text{für } \forall \vec{x}_i \in S_n : \alpha_i > 0$$

Teste nur noch die Support Vektoren !

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

loo bei SVM (2)

- **Beschleunigung von loo durch Ausnutzen der Eigenschaften der SVM.**
- **Liefert bekannt gute Ergebnisse.**
- **ABER: für große n kann es auch sehr viele Support Vektoren geben. Rechenaufwand immer noch groß.**

Lösung : Anstelle $Err^{n}_{loo-SVM}(h)$ **zu berechnen, Schätzt man** $Err^{n}_{loo-SVM}(h)$ **ab.**

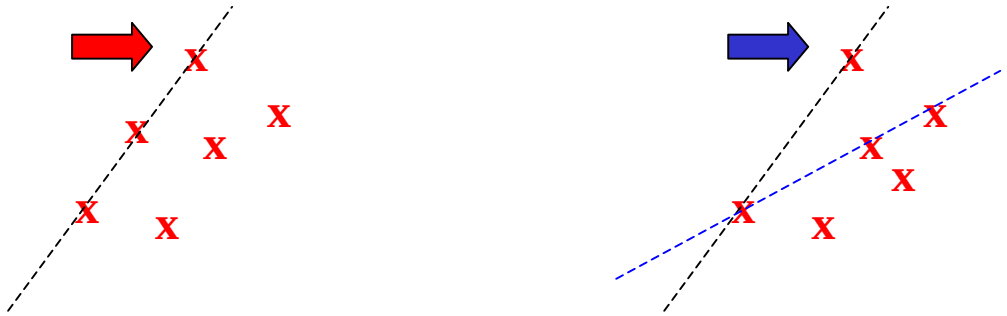
Schätzen von $Err^n_{loo-SVM}(h)$

Obere Schranke : $$Err^n_{loo-SVM}(h) \leq \frac{1}{n} Anz(\vec{x}_i) : \alpha_i > 0$$

Die Anzahl der Fehlklasifikationen ist kleiner gleich der Anzahl der Support Vektoren

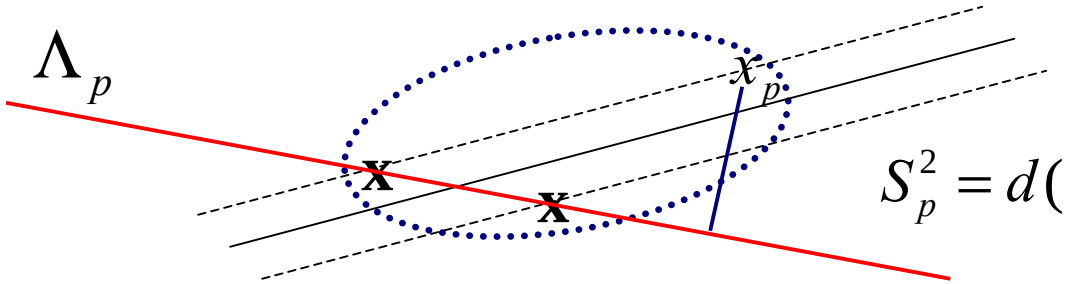
Aber: **Nicht alle SV haben den gleichen Einfluß auf die Modell-Bildung.**

Bsp.:



Support Vector Span

$$\Lambda_p = \left\{ \sum_{\{i \neq p \,/\, 0<\alpha_i<C\}}^{l} \lambda_i \; x_i, \quad \sum_{i=1, i\neq p}^{l} \lambda_i = 1, \; 0 \leq \alpha_i + y_i y_p \alpha_p \lambda_i \leq C \right\}$$



Kleineste Ellipse die alle SV enthält mir Durchmesser D

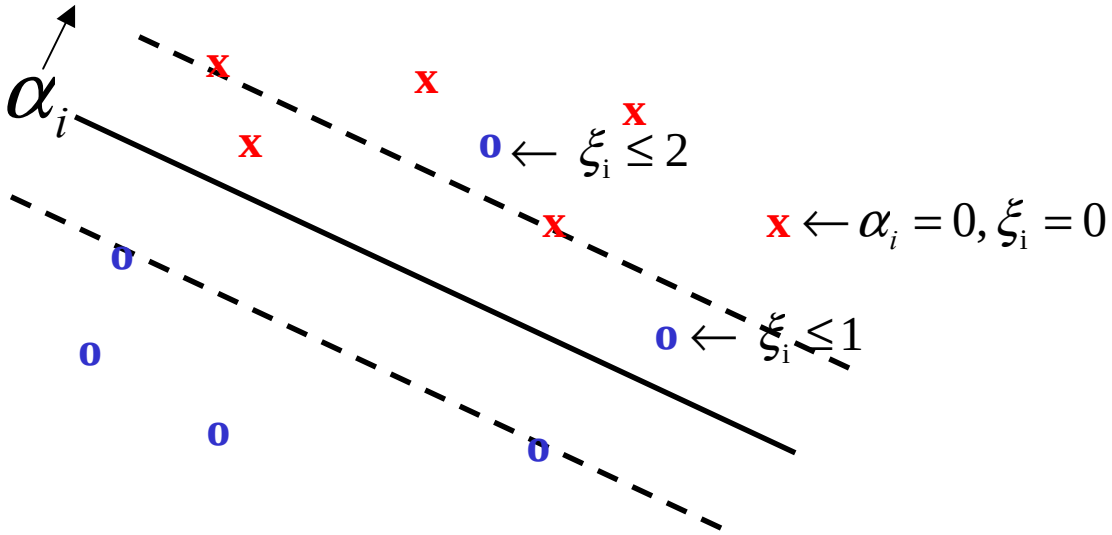
Vektor Span : S_p^2

$$Err_{loo-vs}^{n}(h) \leq \sum_{p=1}^{n} \vec{x}^{\,p} \quad \text{mit} \quad \alpha_p \geq \frac{1}{S_p \max(D, \frac{1}{\sqrt{C}})}$$

$\alpha\xi$ – Estimator

Weiterer Ansatz zur Schätzung von loo. Nach T.Joachims, implementiert in *SVMlight.*

Obere Schranke: $Err^n_{\xi\alpha}(h) = \frac{d}{n}$ mit $d = \left|\left\{i : (2\alpha_i R^2_\Delta + \xi_i \right\} \geq 1\right|$



Übersicht

- Motivation
- Maße für die Modell Qualität
- Allgemeine Verfahren
- SVM Verfahren
- Implementierungen
- Literatur

Übersicht

- Motivation
- Maße für die Modell Qualität
- Allgemeine Verfahren
- SVM Verfahren
- Implementierungen
- Literatur

Literatur

T. Joachims: Estimating the Generalization Performance of a SVM Efficiently, 1999
(Allgemeine Verfahren und loo Schätzung)
http://citeseer.nj.nec.com/joachims00estimating.htm

C.J.C. Burges: A Tutorial on Support Vector Machines for Patter Recognition
(VC-Dim)
http://www.ai.mit.edu/people/polina/Papers/miccai00.ps.gz

O. Chapelle, V. Vapnik: Model Selection for Support Vector Machines
(Vector Span)
http://citeseer.nj.nec.com/chapelle00model.html

P. Bannett: A Pattern Search Method for Model Selection of Support Vector Regression
(loo)
http://www.siam.org/meetings/sdm02/proceedings/sdm02-16.pdf

T.L. Bailey, C. Elkan: Estimating the Accuracy of Learned Concepts

http://citeseer.nj.nec.com/bailey93estimating.htm