Support Vector Machines

André Hopfgartner & Matthias Rupp

08.06.2021

Vorarlberg University of Applied Sciences

FH Vorarlberg
University of Applied Sciences

Agenda

Einführung

Intuition

Ziel: lineare Trennung zweier Klassen

Intuition

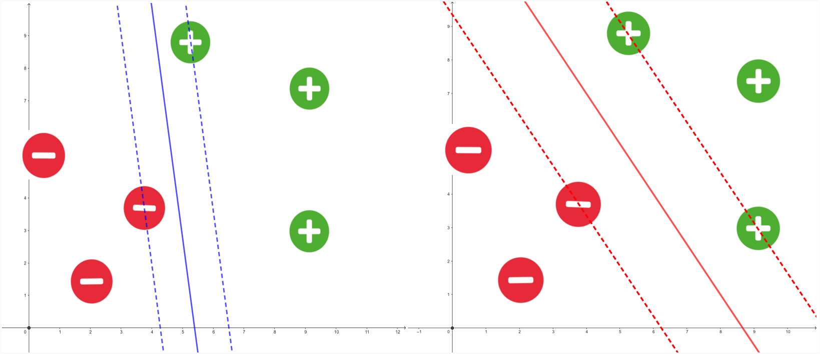
Ziel: lineare Trennung zweier Klassen
Wie?: Definition einer (Hyper-) Ebene

Intuition

Ziel: lineare Trennung zweier Klassen
Wie?: Definition einer (Hyper-) Ebene
Nebenbedingung: Möglichst großer freier Bereich

Intuition

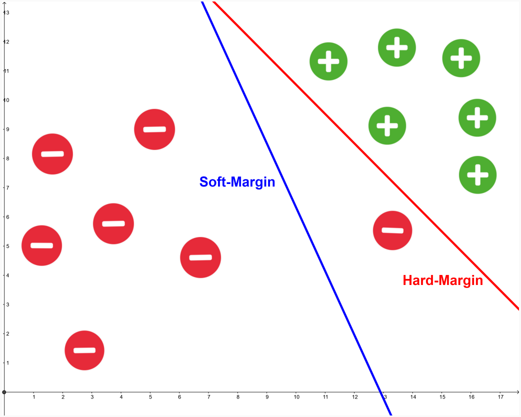
Ziel: lineare Trennung zweier Klassen
Wie?: Definition einer (Hyper-) Ebene
Nebenbedingung: Möglichst großer freier Bereich



Arten von SVM

Arten von SVM:

- *Hard-Margin SVM*: Daten werden 100% korrekt getrennt
- *Soft-Margin SVM*: Einzelne Datenpunkte können falsch klassifiziert werden um insgesamt bessere Trennung zu erhalten



Hard-Margin Support Vector Machine

Mathematische Formulierung

Gegeben sei ein Gewichtsvektor $w \in \mathbb{R}^K$, ein Bias $b \in \mathbb{R}$, ein beliebiger Punkt $x_n \in \mathbb{R}^K$ und ein zugehöriges Label $y_n \in \{-1, +1\}$. Eine Ebene im Raum kann allgemein definiert werden durch:

$$w^T x_n + b = 0$$

Ziel der SVM: w und b bestimmen für optimale Trennung

Klassifikation

Annahme: w und b bereits bekannt

Wie klassifiziert man einen Punkt x_n?

Klassifikation

Annahme: w und b bereits bekannt

Wie klassifiziert man einen Punkt x_n?

Liegt x_n über oder unter Ebene = Vorzeichen:

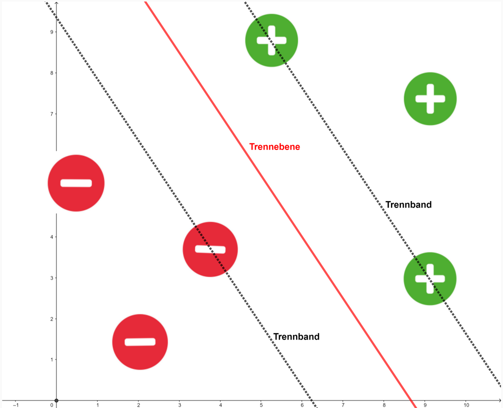
$$
\begin{aligned}
y = sign(w^T x_n + b) &\qquad \text{ist gleichbedeutend mit} \\
w^T x_n + b > 0 &\qquad \text{für } y_n = +1 \\
w^T x_n + b < 0 &\qquad \text{für } y_n = -1
\end{aligned}
$$

Bisher: Punkte können genau auf der Grenze liegen wenn $w^T x_n + b = 0$

Einführung eines Trennbandes

Striktere Regel: Um Ebene soll Band frei bleiben

$$
\begin{aligned}
w^T x_n + b &\geq +1 \qquad \text{für } y_n = +1 \\
w^T x_n + b &\leq -1 \qquad \text{für } y_n = -1
\end{aligned}
$$



Einführung eines Trennbandes

Beidseitige Multiplikation mit y_n

$$\begin{aligned} y_n(w^T x_n + b) &\geq 1 \qquad \text{für } y_n = +1 \\ y_n(w^T x_n + b) &\geq 1 \qquad \text{für } y_n = -1 \end{aligned}$$

Einführung eines Trennbandes

Beidseitige Multiplikation mit y_n

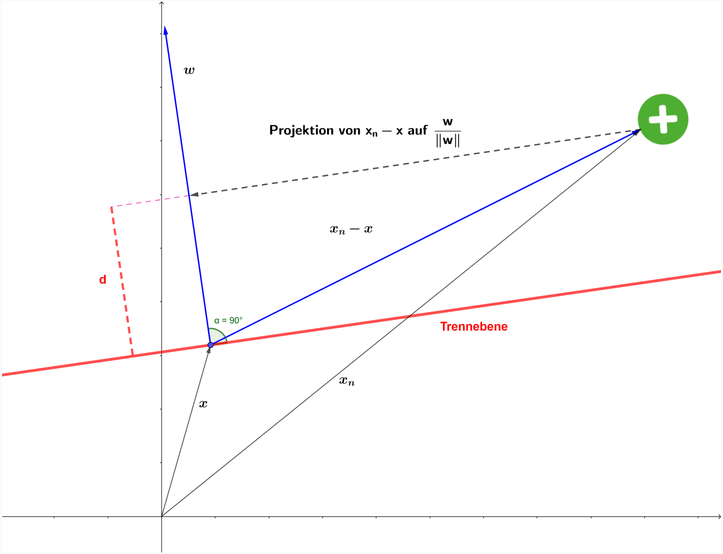
$$\begin{aligned} y_n(w^T x_n + b) &\geq 1 \qquad \text{für } y_n = +1 \\ y_n(w^T x_n + b) &\geq 1 \qquad \text{für } y_n = -1 \end{aligned}$$

Für den Fall, dass $x_n = \hat{x}$ genau an der Grenze des Trennbands liegt, gilt somit:

$$y_n(w^T \hat{x} + b) = 1$$

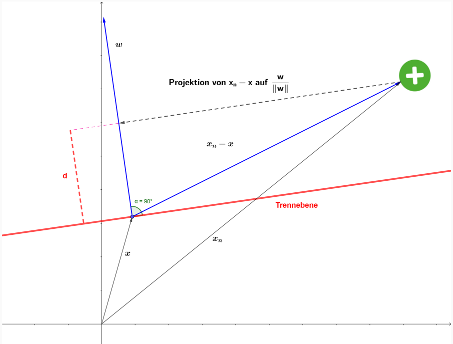
Normalabstand eines Punktes zur Ebene

Gesucht: Normalabstand d eines Punktes $x_n \in \mathbb{R}^K$ zur Ebene



Normalabstand eines Punktes zur Ebene

$$
\begin{aligned}
d &= \left|\frac{w^T}{\|w\|}(x_n - x)\right| = \\
&= \frac{1}{\|w\|}\left|(w^T x_n - w^T x)\right| = \\
&= \frac{1}{\|w\|}\left|(w^T x_n + b - (w^T x + b))\right|
\end{aligned}
$$



Normalabstand eines Punktes zur Ebene

$$d = \frac{1}{\|w\|}|(w^T x_n + b - (w^T x + b))|$$

Weil der Punkt x auf der Ebene liegt gilt $w^T x + b = 0$ und somit für den Normalabstand eines beliebigen Punktes x_n:

$$d = \frac{1}{\|w\|}|(w^T x_n + b)|$$

Breite des Trennbands

$$d = \frac{1}{\|w\|}|(w^T x_n + b)|$$

Annahme: $x_n = \hat{x}$ ist der am nächsten zur Ebene liegende Punkt auf der Grenze des Trennbands
Weil $y_n(w^T\hat{x} + b) = 1 = |w^T\hat{x} + b|$ gilt ergibt sich der minimale Normalabstand D:

$$D = \frac{1}{\|w\|}$$

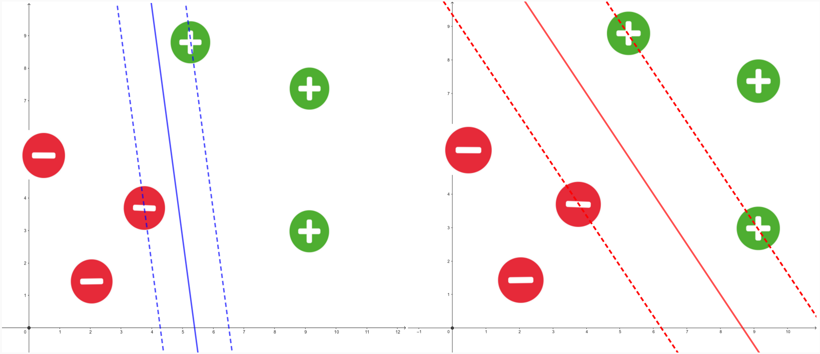
Weil D der minimale Normalabstand zur Ebene ist, ist $2D$ die Breite des freien Trennbands.

Reminder

Ziel: lineare Trennung mit möglichst breitem, freien Trennband

Entspricht Maximierung:

$$\max_w(2D) = \max_w \frac{2}{\|w\|} = \max_w \frac{1}{\|w\|}$$



Optimierungsproblem

$$\max_{w} \quad \frac{1}{\|w\|}$$

$$\text{mit} \quad \min_{n=1..N} |w^T x_n + b| = 1$$

$\min_{n=1..N} |w^T x_n + b| = 1$ ist der am nächsten zur Ebene liegende Punkt $\hat{x}$

Beidseitige Multiplikation mit y_n zur Vermeidung des Betrags:

$$|w^T x_n + b| = y_n(w^T x_n + b)$$

Optimierungsproblem

Nach Umformung (Maximierung in Minimierung) und Verallgemeinerung der Nebenbedingung auf beliebige Punkte x_n:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2} w^T w \\ \text{mit} \quad & y_n(w^T x_n + b) \geq 1 \text{ für } n = 1..N \end{aligned}$$

Bemerkungen:

- Faktor $\frac{1}{2}$ wird so gewählt weil dieser später wegfällt
- $w^T w$ und $\|w\|$ sind aus Optimierungssicht gleichbedeutend, Problem ist in dieser Form aber besser optimierbar

Lagrange Optimierung

Optimierungsproblem mit Ungleichung als Nebenbedingung

Umformen der Nebenbedingung:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2} w^T w \\ \text{mit} \quad & y_n(w^T x_n + b) - 1 \geq 0 \text{ für } n = 1..N \end{aligned}$$

Aufstellen der Lagrange Gleichung

Ungleichung wird von zu optimierender Funktion abgezogen und Lagrange Multiplikatoren eingeführt:

$$\min_{w,b} \qquad \mathcal{L}(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n (y_n(w^T x_n + b) - 1)$$

$$\max_{\alpha_n} \qquad \alpha_n \geq 0 \text{ für } n = 1..N$$

Lösung durch 0 setzen der partiellen Ableitungen:

$$\nabla_w \mathcal{L} \stackrel{!}{=} \vec{0}$$

$$\frac{\partial}{\partial b}\mathcal{L} \stackrel{!}{=} 0$$

Lösen der Lagrange Gleichung

Nach w:

$$\nabla_w \mathcal{L} = w - \sum_{n=1}^{N} \alpha_n y_n x_n \stackrel{!}{=} \vec{0}$$

$$w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

Nach b:

$$\frac{\partial}{\partial b} \mathcal{L} = - \sum_{n=1}^{N} \alpha_n y_n \stackrel{!}{=} 0$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0$$

Rücksubstitution in Lagrange Gleichung

Aufteilen der Summe:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n (y_n (w^T x_n + b) - 1) =$$

$$= \frac{1}{2}w^T w - [\sum_{n=1}^{N} \alpha_n y_n b - \sum_{n=1}^{N} \alpha_n + \sum_{n=1}^{N} \alpha_n y_n w^T x_n]$$

Aus Ableitung nach b wissen wir $\sum_{n=1}^{N} \alpha_n y_n = 0$:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}w^T w - [-\sum_{n=1}^{N} \alpha_n + \sum_{n=1}^{N} \alpha_n y_n w^T x_n]$$

Rücksubstitution in Lagrange Gleichung

Vergleicht man den Term $\sum_{n=1}^{N} \alpha_n y_n w^T x_n$ mit dem Ergebnis der partiellen Ableitung nach w ($w = \sum_{n=1}^{N} \alpha_n y_n x_n$) erkennt man, dass gilt:

$$\begin{aligned}\sum_{n=1}^{N} \alpha_n y_n w^T x_n = w^T w = \\ &= \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m\end{aligned}$$

Eingesetzt in Lagrange Gleichung:

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m$$

Maximierung ohne Nebenbedingung

Quadratic Programming Problem ($x_n^T x_m$):

$$
\begin{aligned}
\max_{\alpha} \quad & \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m \\
\text{mit} \quad & \alpha_n \geq 0 \text{ für } n = 1..N \\
& \sum_{n=1}^{N} \alpha_n y_n = 0 \text{ für } n = 1..N
\end{aligned}
$$

Lösung mittels QP-Solver

Ergebnis: α Vektor mit α_n Lagrange-Multiplikatoren

Schlupfterm

Reminder Ausgangsproblem:

$$\min_{w,b} \quad \mathcal{L}(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{n=1}^{N} \alpha_n(y_n(w^T x_n + b) - 1)$$

$$\max_{\alpha_n} \quad \alpha_n \geq 0 \text{ für } n = 1..N$$

$\alpha_n(y_n(w^T x_n + b) - 1)$ („Schlupf“) wird 0 wenn:

- $\alpha_n = 0$ oder
- $(y_n(w^T x_n + b) - 1) = 0$

Umgekehrt: Alle x_n mit $\alpha_n \neq 0$ haben Schlupf 0, liegen also am nächsten zur Trennebene.
Diese Vektoren werden **Stützvektoren** genannt.

Bestimmung Gewichtsvektor

α Vektor mit α_n Faktoren ist bekannt aus QP-Solver

Viele α_i werden 0 sein, die $\alpha_i \neq 0$ gehören zu den Stützvektoren x_i.

Damit kann Formel für w

$$w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

vereinfacht werden:

$$w = \sum_{n \text{ ist Sützvektor}} \alpha_n y_n x_n$$

Die Bezeichnung Stützvektor ergibt sich, weil die Ebene durch diese Vektoren „gestützt “wird. Alle Vektoren mit $\alpha_n = 0$ haben keinen Einfluss!

Bestimmung Bias

$y_n(w^T x_n + b) = 1$ gilt für Stützvektoren, daher kann mit beliebigem Stützvektor x_n der Bias bestimmt werden:

$$\begin{aligned} b &= \frac{1}{y_n} - w^T x_n = \\ &= y_n - w^T x_n \end{aligned}$$

Lösung mittels QP-Solver

Lösung mittels QP-Solver

Standardform von QP-Problemen:

$$\min_{x} = \frac{1}{2}x^T Qx + cx + d$$

Umformung Maximierung in Minimierung weil $\max -f(x) = \min f(x)$:

$$\min_{\alpha} \mathcal{L}(\alpha) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

Problem in QP-Standardform

$$\min_{\alpha} \mathcal{L}(\alpha) = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

In QP-Standardform $\rightarrow$ Lösungs-Frameworks:

$$\begin{aligned}
\min_{\alpha} \quad & \mathcal{L}(\alpha) = \frac{1}{2} \alpha^T Q \alpha + (-1^T)\alpha \\
\text{mit} \quad & Q = \begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 & \dots & y_1 y_N x_1^T x_N \\ y_2 y_1 x_2^T x_1 & y_2 y_2 x_2^T x_2 & \dots & y_2 y_N x_2^T x_N \\ \vdots & \vdots & \vdots & \vdots \\ y_N y_1 x_N^T x_1 & y_N y_2 x_N^T x_2 & \dots & y_N y_N x_N^T x_N \end{bmatrix}
\end{aligned}$$

Problem in QP-Standardform

$$\min_{\alpha} \mathcal{L}(\alpha) = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

In QP-Standardform $\rightarrow$ Lösungs-Frameworks:

$$\min_{\alpha} \qquad \mathcal{L}(\alpha) = \frac{1}{2}\alpha^T Q \alpha + (-1^T)\alpha$$

$$\text{mit} \qquad Q = \begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 & \dots & y_1 y_N x_1^T x_N \\ y_2 y_1 x_2^T x_1 & y_2 y_2 x_2^T x_2 & \dots & y_2 y_N x_2^T x_N \\ \vdots & \vdots & \vdots & \vdots \\ y_N y_1 x_N^T x_1 & y_N y_2 x_N^T x_2 & \dots & y_N y_N x_N^T x_N \end{bmatrix}$$

Q ist $N \times N$ Matrix. Problematisch bei großen Trainingssets!

Lösung mittels QP-Solver

Ergebnis des QP-Solvers: $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$
Berechnung von w und b wie zuvor gezeigt:

$$w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

Mit beliebigem Stützvektor x_k:

$$b = \frac{1}{y_k} - w^T x_k$$

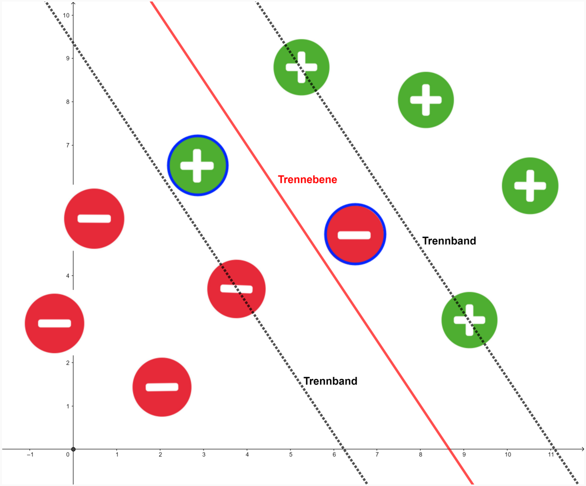
Klassifikation neuer Eingaben x:

$$y = sign(w^T x + b)$$

Soft-Margin Support Vector Machine

Einführung Soft-Margin SVM

Annahme bisher: Daten linear trennbar ohne Fehler



Einführung von Fehlervariablen

Problem: bisheriger Algorithmus terminiert nicht bei Fehlern
Lösung: Einführung von positiven Fehlervariablen $\xi_n \in \mathbb{R}^K, \xi_n \geq 0$:

$$\begin{aligned} w^T x_n + b &\geq +1 - \xi_n \qquad \text{für } y_n = +1 \\ w^T x_n + b &\leq -1 + \xi_n \qquad \text{für } y_n = -1 \end{aligned}$$

Wann kann einzelne Fehlklassifikation auftreten? Wenn $\xi_n > 1$
Obere Grenze Anzahl Fehler:

$$E = C(\sum_{n=1}^{N} \xi_n)$$

$C \in \mathbb{R}, C \geq 0$: „Straffaktor"für Fehler

Erweiterung Optimierungsproblem um Fehlerterm

Ziel: Optimales w mit möglichst wenig Fehlern:

$$\begin{array}{ll} \min\limits_{w} & \frac{1}{2} w^T w + C(\sum_{n=1}^{N} \xi_n) \\ \text{mit} & y_n(w^T x_n + b) - 1 \geq 0 \text{ für } n = 1..N \end{array}$$

Ableiten, 0 setzen und lösen wie zuvor...

Soft-Margin SVM Optimierungsproblem

Soft-Margin Optimierungsproblem:

$$\max_{\alpha} \quad \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m$$

$$\text{mit} \quad 0 \leq \alpha_n \leq C \text{ für } n = 1..N$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \text{ für } n = 1..N$$

Einziger Unterschied zu Hard-Margin: Beschränkung $\alpha_n \leq C$
(Hard-Margin: $\alpha_n \leq \infty$)

Soft-Margin SVM Optimierungsproblem

Soft-Margin Optimierungsproblem:

$$\max_{\alpha} \quad \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} y_n y_m \alpha_n \alpha_m x_n^T x_m$$

$$\text{mit} \quad 0 \leq \alpha_n \leq C \text{ für } n = 1..N$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \text{ für } n = 1..N$$

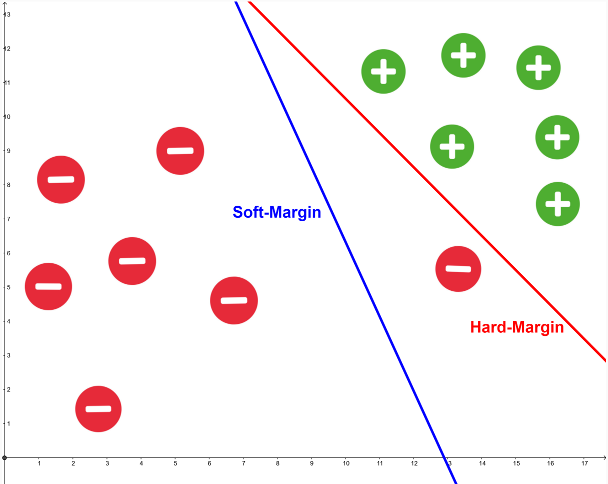
Einziger Unterschied zu Hard-Margin: Beschränkung $\alpha_n \leq C$
(Hard-Margin: $\alpha_n \leq \infty$)
Umgekehrt: Soft-Margin mit $C \to \infty$ entspricht Hard-Margin
Lösung: Wie zuvor gezeigt mit QP-Solver

Vergleich Hard- & Soft-Margin Support Vector Machine

Vergleich Hard- & Soft-Margin SVM

Hard-Margin: einzelne Ausreißer bestimmen Lage der Ebene

Soft-Margin: Fehlklassifikationen zugunsten besserer Gesamt-Trennung



A test with images

- Some
- text
- on left side of slide here..
- Abb. 1 zeigt blabla.

A test with images

- Some
- text
- on left side of slide here..
- Abb. 1 zeigt blabla.

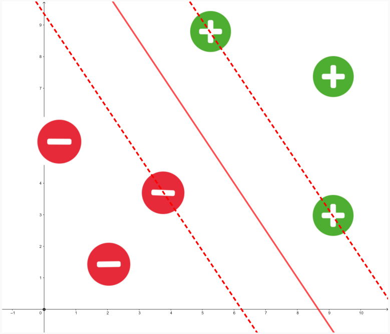


Abbildung 1: Abhängig von der Lage der Trennebene entstehen schmale (blau) oder breite (rot) Trennbänder. Ziel ist die Maximierung der Breite des Trennbands durch die Ermittlung der optimalen Lage der Trennebene.

citation tests

$$
\begin{aligned}
y = sign(w^T x + b) &\qquad \text{gleichbedeutend mit} && (14a)\\
w^T x + b > 0 &\qquad \text{für } y = +1 && (14b)\\
w^T x + b < 0 &\qquad \text{für } y = -1 && (14c)
\end{aligned}
$$

In Gleichung (14) wird ..
Footcite example[1]
Burges (1998)

[1] Platt 1998.

Nichtlineare Trennung

Pseudocode und Beispiele

Pseudocode Hard-Margin SVM

Hard-Margin SVM

Initialisiere x, y

$Q = (yy^T)K$

$c = (-1, -1, \dots, -1)^T$

$A = diag\,(-1, -1, \dots, -1)$

$b = (0, 0, \dots, 0)^T$

$A_{eq} = y^T$

$b_{eq} = 0$

$lb = (0, 0, \dots, 0)^T$

$ub = C * (1, 1, \dots, 1)^T$

$\alpha = QPSolver\,(Q, c, A, b, A_{eq}, b_{eq})$

$w = \sum_{n=SV} \alpha_n y_n x_n$

$bias = \frac{1}{y_n} - w^T x_n$

Anmerkungen

Fragen?

Literatur

Burges, Christopher J.C. (1. Juni 1998). „A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2.2, S. 121–167. ISSN: 1573-756X. DOI: 10.1023/A:1009715923555. URL: https://doi.org/10.1023/A:1009715923555 (besucht am 06. 03. 2021).

Platt, John (Apr. 1998). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* MSR-TR-98-14, S. 21. URL: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.