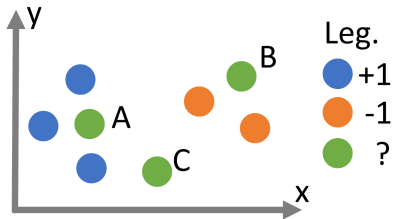


Support Vector Machines mit Kerneln

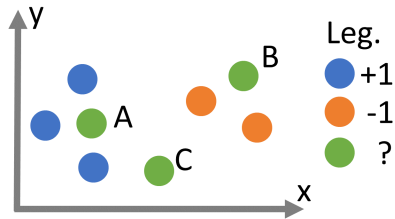
Tim Schlottmann, Hendrik Sieck, Jonas Krug

26.01.2018

Motivation



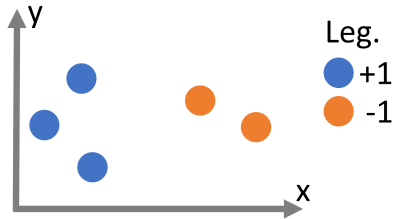
- ▶ Klassifizierung von Objekten
- ▶ Schnell und effizient
- ▶ Möglichst genau



- Support Vector Machine
- Binärer Klassifizierer

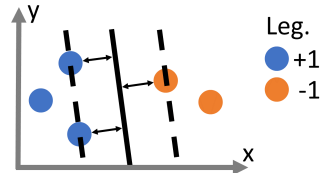
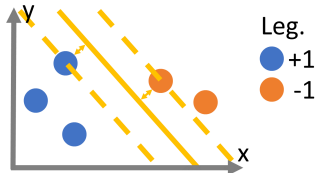
1. Grundlagen von SVM
2. Kernel-Trick
3. Beispiele

Problemstellung



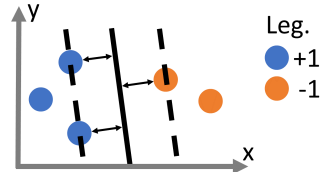
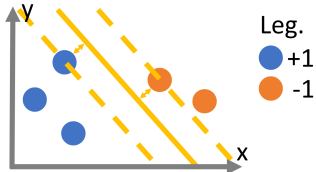
- Wie trenne ich die beiden Klassen voneinander?

Problemstellung



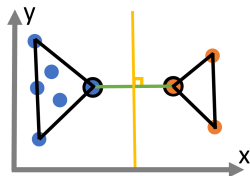
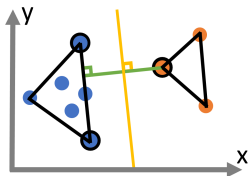
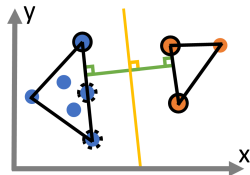
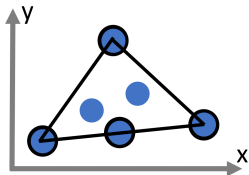
- ▶ Wie lege ich die Hyperebene am besten?
- ▶ Anderer Name von Support Vector Machines: Large Margin Classifier

Problemstellung



- ▶ Wie lege ich die Hyperebene am besten?
- ▶ Anderer Name von Support Vector Machines: Large Margin Classifier

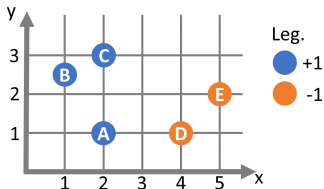
Support vectors (s.v.)



Definitionen

- ▶ $m \in \mathbb{R}$ Datenpunkte
 - ▶ Input $\mathbf{x} \in \mathbb{R}^N$
 - ▶ Output $y \in \{-1, +1\}$
 - ▶ Trainingsset $S \in (\mathbb{R}^N \times \{-1, +1\})^m$
-
- ▶ Hypothese

$$h: \mathbb{R}^N \rightarrow \{+1, -1\}$$
$$\mathbf{x} \mapsto y$$



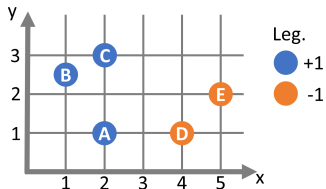
- ▶ $m = 5$
- ▶ Trainingsset S :

$$S = \begin{pmatrix} 1 & 2.5 & +1 \\ 5 & 2 & -1 \\ \vdots & \vdots & \vdots \end{pmatrix}$$

Definitionen

- ▶ $m \in \mathbb{R}$ Datenpunkte
 - ▶ Input $\mathbf{x} \in \mathbb{R}^N$
 - ▶ Output $y \in \{-1, +1\}$
 - ▶ Trainingsset $S \in (\mathbb{R}^N \times \{-1, +1\})^m$
-
- ▶ Hypothese

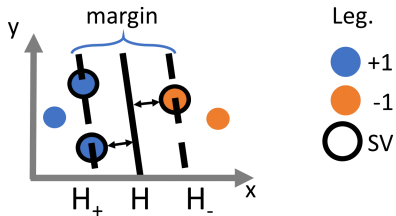
$$h: \mathbb{R}^N \rightarrow \{+1, -1\}$$
$$\mathbf{x} \mapsto y$$



- ▶ $m = 5$
- ▶ Trainingsset S :

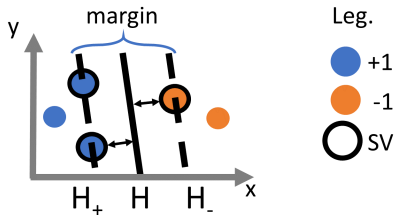
$$S = \begin{pmatrix} 1 & 2.5 & +1 \\ 5 & 2 & -1 \\ & \vdots & \end{pmatrix}$$

Verlauf der Hyperbene



- ▶ Hyperebene: $H' = \mathbf{w}'^T \mathbf{x} + b' = 0$
- ▶ Gutter: H_+ und H_-
- ▶ Hyperbene frei skalierbar

Verlauf der Hyperbene



► Hyperebene: $H' = \mathbf{w}'^T \mathbf{x} + b' = 0$

► Gutter constraint (GC):

$$H_+ := \mathbf{w}^T \mathbf{x}_p + b = +1, \quad \forall \text{ s.v., die auf } H_+ \text{ liegen}$$

$$H_- := \mathbf{w}^T \mathbf{x}_n + b = -1, \quad \forall \text{ s.v., die auf } H_- \text{ liegen}$$

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1 \quad \forall \text{ s.v.}$$

► Um GC zu erfüllen: \mathbf{w} und b werden um $v \in \mathbb{R}$ skaliert:

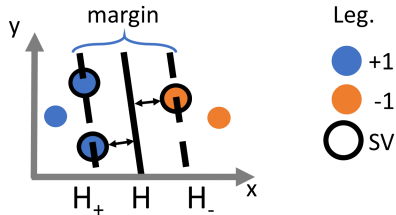
$$\mathbf{w} = v^T \mathbf{w}'$$

$$b = v^T b'$$

$$H = v(\mathbf{w}^T \mathbf{x} + b) = 0$$

► H heißt auch kanonische Hyperebene

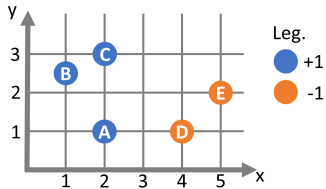
Verlauf der Hyperbene



- ▶ Kanonische Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$
- ▶ Kanonische Hyperebene ermöglicht Klassifizierung:

$$h(\mathbf{x}_i) = \begin{cases} +1 & \text{wenn } \mathbf{w}^T \mathbf{x}_i + b \geq 0 \\ -1 & \text{wenn } \mathbf{w}^T \mathbf{x}_i + b \leq 0 \end{cases}$$

Beispiel I



► Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$

► Gutter constraint:
 $\mathbf{w}^T \mathbf{x}_i + b = y_i, \quad \forall \text{ s. v. } \in \mathbf{x}$

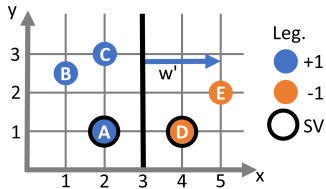
► Graphische Bestimmung der Hyperebenenparameter:

$$x = 3$$

$$\mathbf{w} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 \end{pmatrix}^T \begin{pmatrix} 3 \\ 0 \end{pmatrix} + b = 0 \Rightarrow b = -6$$

Beispiel I



► Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$

► Gutter constraint:
 $\mathbf{w}^T \mathbf{x}_i + b = y_i, \quad \forall \text{ s. v. } \in \mathbf{x}$

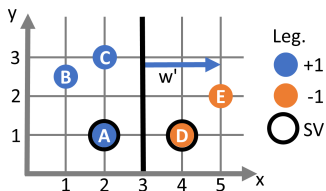
► Graphische Bestimmung der Hyperebenenparameter:

$$x = 3$$

$$\mathbf{w} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 \end{pmatrix}^T \begin{pmatrix} 3 \\ 0 \end{pmatrix} + b = 0 \Rightarrow b = -6$$

Beispiel II



► Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$

► Gutter constraint:
 $\mathbf{w}^T \mathbf{x}_i + b = y_i, \quad \forall \text{ s. v. } \in \mathbf{x}$

► Berücksichtigen der Gutter constraint am Bsp. von Punkt A:

$$c \left(\begin{pmatrix} 2 & 0 \end{pmatrix}^T \begin{pmatrix} 2 \\ 1 \end{pmatrix} - 6 \right) \stackrel{!}{=} +1 \Rightarrow c = -0.5$$

► Somit gilt für die kanonische Hyperebene:

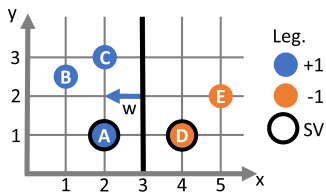
$$\mathbf{w} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$b = 3$$

► Kontrolle mit Punkt D:

$$\begin{pmatrix} -1 & 0 \end{pmatrix}^T \begin{pmatrix} 4 \\ 1 \end{pmatrix} + 3 = -1$$

Beispiel II



► Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$

► Gutter constraint:
 $\mathbf{w}^T \mathbf{x}_i + b = y_i, \quad \forall \text{ s. v. } \in \mathbf{x}$

► Berücksichtigen der Gutter constraint am Bsp. von Punkt A:

$$c \left(\begin{pmatrix} 2 & 0 \end{pmatrix}^T \begin{pmatrix} 2 \\ 1 \end{pmatrix} - 6 \right) \stackrel{!}{=} +1 \Rightarrow c = -0.5$$

► Somit gilt für die kanonische Hyperebene:

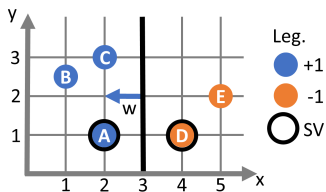
$$\mathbf{w} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$b = 3$$

► Kontrolle mit Punkt D:

$$\begin{pmatrix} -1 & 0 \end{pmatrix}^T \begin{pmatrix} 4 \\ 1 \end{pmatrix} + 3 = -1$$

Beispiel II



► Hyperebene: $H = \mathbf{w}^T \mathbf{x} + b = 0$

► Gutter constraint:
 $\mathbf{w}^T \mathbf{x}_i + b = y_i, \quad \forall \text{ s. v. } \in \mathbf{x}$

► Berücksichtigen der Gutter constraint am Bsp. von Punkt A:

$$c \left(\begin{pmatrix} 2 & 0 \end{pmatrix}^T \begin{pmatrix} 2 \\ 1 \end{pmatrix} - 6 \right) \stackrel{!}{=} +1 \Rightarrow c = -0.5$$

► Somit gilt für die kanonische Hyperebene:

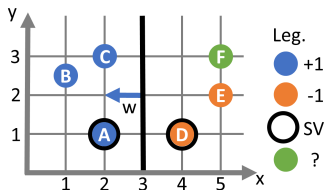
$$\mathbf{w} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$b = 3$$

► Kontrolle mit Punkt D:

$$\begin{pmatrix} -1 & 0 \end{pmatrix}^T \begin{pmatrix} 4 \\ 1 \end{pmatrix} + 3 = -1$$

Beispiel III



- Klassifizierung:

$$h(x_i) = \begin{cases} +1 & \text{if } \mathbf{w}^T \mathbf{x}_i + b \geq 0 \\ -1 & \text{if } \mathbf{w}^T \mathbf{x}_i + b \leq 0 \end{cases}$$

- Parameter der kanonischen Hyperebene

$$\mathbf{w} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

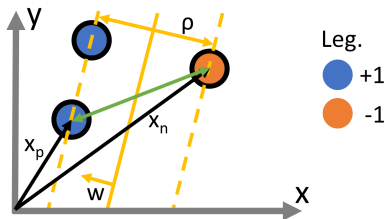
$$b = 3$$

- Klassifizierung von Punkt F:

$$\begin{pmatrix} -1 & 0 \end{pmatrix}^T \begin{pmatrix} 5 \\ 3 \end{pmatrix} + 3 = -2$$

- Punkt F gehört also zur Klasse -1.

Minimierungsproblem



- ▶ Projektionseigenschaft des Skalarprodukts
- ▶ Breite des margin ρ :

$$\rho = (\mathbf{x}_p - \mathbf{x}_n)^T \frac{\mathbf{w}}{\|\mathbf{w}\|} \Leftrightarrow \rho = (\mathbf{x}_p^T \mathbf{w} - \mathbf{x}_n^T \mathbf{w}) \frac{1}{\|\mathbf{w}\|}$$

- ▶ Gutter constraint:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1 \quad \forall \text{ s.v. } \in \mathbf{x}$$

- ▶ $\rho = \frac{2}{\|\mathbf{w}\|}$
- ▶ Ziel einer SVM: Maximiere den margin

$$\max_{\mathbf{w}, b} \frac{2}{\|\mathbf{w}\|} \Leftrightarrow \min_{\mathbf{w}, b} \|\mathbf{w}\| \Leftrightarrow \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

u.d.N. $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0$

Lagrange Multiplikatoren

- ▶ Lagrange Multiplikatoren:

$$L = \frac{1}{2} \|\mathbf{w}\|^2 - \sum \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1]$$

- ▶ Suche nach den Extremum

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = - \sum \alpha_i y_i = 0 \Rightarrow \sum_i \alpha_i y_i = 0$$

- ▶ \mathbf{w} in L eingesetzt:

$$L = \frac{1}{2} \left(\sum_i \alpha_i y_i \mathbf{x}_i \right)^T \left(\sum_j \alpha_j y_j \mathbf{x}_j \right) - \left(\sum_i \alpha_i y_i \mathbf{x}_i \right)^T \left(\sum_j \alpha_j y_j \mathbf{x}_j \right) - \sum_i \alpha_i y_i b + \sum_i \alpha_i$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

- ▶ Ziel: $\max_{\mathbf{w}, b, \alpha} L$
- ▶ Entscheidungsfunktion:

$$h(\mathbf{x}) = \begin{cases} +1 & \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \geq 0 \\ -1 & \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \leq 0 \end{cases}$$

Unterstützungsvariablen α

► Eigenschaften:

$$\alpha \geq 0$$

$$\sum_{\substack{\text{pos. s.v.} \\ p}} \alpha_p = \sum_{\substack{\text{neg. s.v.} \\ n}} \alpha_n$$

$$\alpha_i \begin{cases} > 0 & \text{wenn } x_i \text{ ein support vector ist} \\ = 0 & \text{sonst} \end{cases}$$

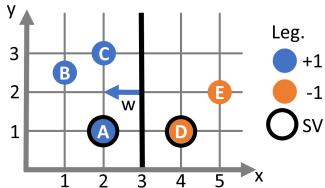
Unterstützungsvariablen α

► Eigenschaften:

$$\alpha \geq 0$$

$$\sum_{\substack{\text{pos. s.v.} \\ p}} \alpha_p = \sum_{\substack{\text{neg. s.v.} \\ n}} \alpha_n$$

$$\alpha_i \begin{cases} > 0 & \text{wenn } x_i \text{ ein support vector ist} \\ = 0 & \text{sonst} \end{cases}$$



► Berechnung von α_A und α_D ergibt:

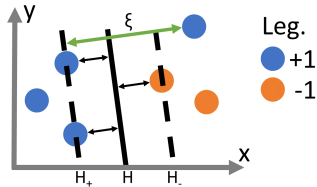
$$\alpha_A = 0.5$$

$$\alpha_D = 0.5$$

► Berechnung von α_C ergibt:

$$\alpha_C = 0$$

Ausgleich eines fehlerhaften Trainingssets



- Schlupfvariable ξ zum Ausgleich falscher Messwerte

- GC:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall \mathbf{x}_i \in S \qquad \xi_i \geq 0$$

- Minimierungsproblem:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i$$

$$\text{u.d.N. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \wedge \xi_i \geq 0$$

- Kanonische Hyperebene:

$$\mathbf{w}^T \mathbf{x} + b = 0$$

- Gesucht: Parameter \mathbf{w} und b
- Lange Multiplikatoren:

$$\max_{\mathbf{w}, b, \alpha} L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

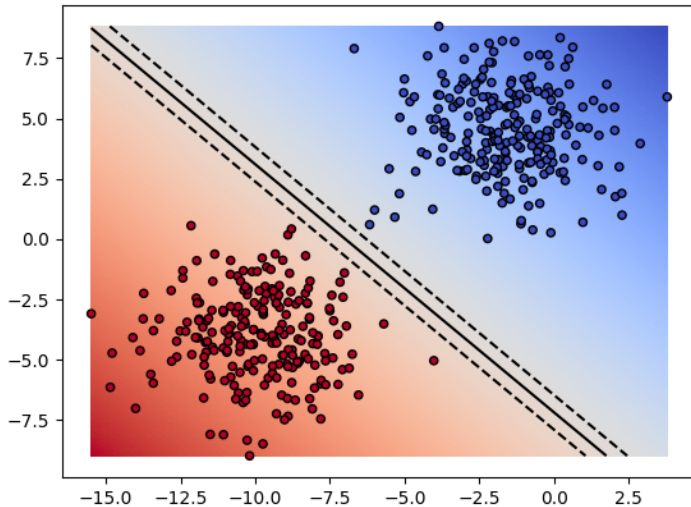
$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\sum_i \alpha_i y_i = 0$$

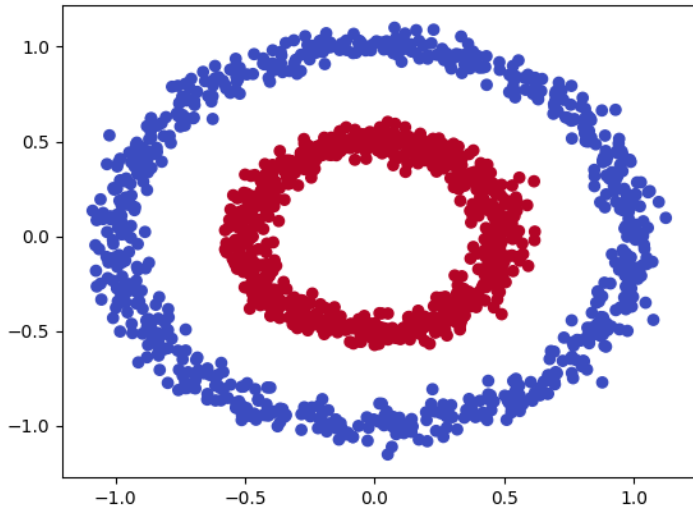
- Entscheidungsfunktion h :

$$h(\mathbf{x}) = \begin{cases} +1 & \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \geq 0 \\ -1 & \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \leq 0 \end{cases}$$

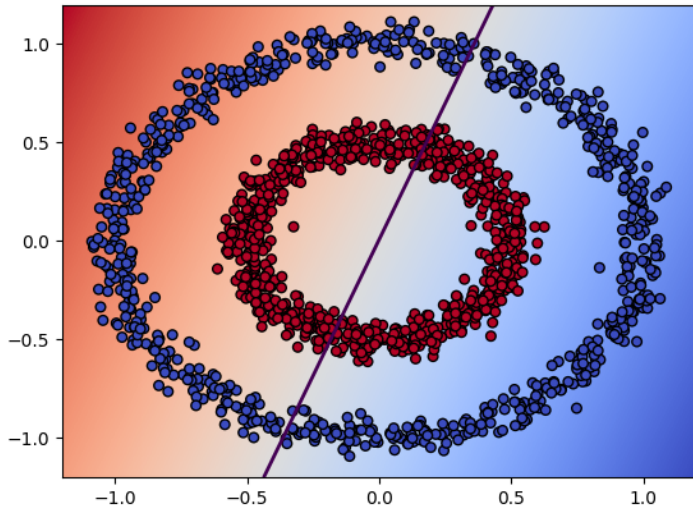
Einführung: Linear separierbare Daten



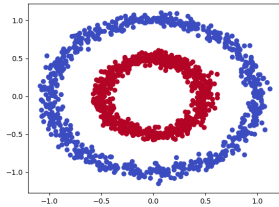
Einführung: Linear nicht separierbare Daten 1



Einführung: Linear nicht separierbare Daten 2

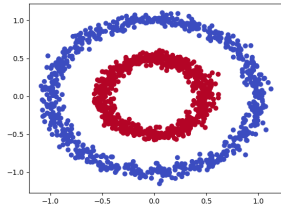


Ansatz: Einführung

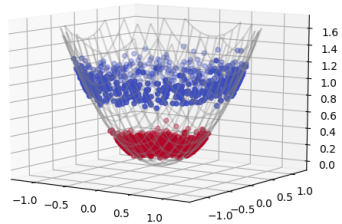


$$\phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$

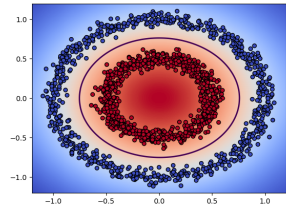
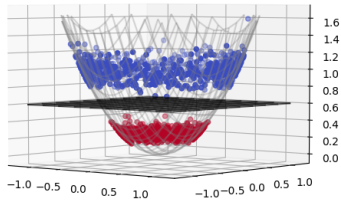
Ansatz: Einführung



$$\phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$



Ansatz: Abbildfunktion $\phi(x)$



Veränderung der Funktionen

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b$$

Veränderung der Funktionen

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b$$

Definition: ϕ

Abbildungsfunktion

$$\begin{aligned}\phi: \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ \mathbf{x} &\mapsto \mathbf{f}\end{aligned}$$

Probleme:

- ▶ $m > n$ höherer Rechenaufwand
Ab einer bestimmten Größe kann damit nicht mehr gerechnet werden
- ▶ Obergrenze für m
- ▶ ϕ nur für Skalarprodukt benötigt

Definition: ϕ

Abbildungsfunktion

$$\begin{aligned}\phi : \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ \mathbf{x} &\mapsto \mathbf{f}\end{aligned}$$

Probleme:

- ▶ $m > n$ höherer Rechenaufwand
Ab einer bestimmten Größe kann damit nicht mehr gerechnet werden
- ▶ Obergrenze für m
- ▶ ϕ nur für Skalarprodukt benötigt

$$\phi(\mathbf{x}) = (1 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ \dots \ x_1^2 \ x_2^2 \ \dots \ \sqrt{2}x_1x_2 \ \sqrt{2}x_1x_3 \ \dots)$$

$$\begin{aligned}\phi(\mathbf{v})^T \phi(\mathbf{w}) &= \sum_j 2v_j w_j + \sum_j v_j^2 w_j^2 + \sum_j \sum_{k>j} 2v_j v_k w_j w_k + \dots \\ &= (1 + \sum_j v_j w_j)^2 \\ &= (1 + \mathbf{v}^T \mathbf{w})^2 \\ &= K(\mathbf{v}, \mathbf{w})\end{aligned}$$

Beispiel Kernel

$$\phi(\mathbf{x}) = (1 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 \quad \dots \quad x_1^2 \quad x_2^2 \quad \dots \quad \sqrt{2}x_1x_2 \quad \sqrt{2}x_1x_3 \quad \dots)$$

$$\begin{aligned}\phi(\mathbf{v})^T \phi(\mathbf{w}) &= \sum_j 2v_j w_j + \sum_j v_j^2 w_j^2 + \sum_j \sum_{k>j} 2v_j v_k w_j w_k + \dots \\ &= (1 + \sum_j v_j w_j)^2 \\ &= (1 + \mathbf{v}^T \mathbf{w})^2 \\ &= K(\mathbf{v}, \mathbf{w})\end{aligned}$$

Einführung von Kernen

$$K(\mathbf{v}, \mathbf{w}) = \phi(\mathbf{v})^T \phi(\mathbf{w})$$

$$K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Einführung von Kernen

$$K(\mathbf{v}, \mathbf{w}) = \phi(\mathbf{v})^T \phi(\mathbf{w})$$

$$K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$h(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Mercer's Theorem: 1

$$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$$
$$\mathcal{K}_{i,j} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$$\begin{aligned}\mathcal{K}_{i,j} &= K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ &= \phi(\mathbf{x}^{(i)})^T \phi(\mathbf{x}^{(j)}) \\ &= \phi(\mathbf{x}^{(j)})^T \phi(\mathbf{x}^{(i)}) \\ &= K(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) \\ &= \mathcal{K}_{j,i}\end{aligned}$$

Mercer's Theorem: 1

$$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$$
$$\mathcal{K}_{i,j} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$$\begin{aligned}\mathcal{K}_{i,j} &= K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ &= \phi(\mathbf{x}^{(i)})^T \phi(\mathbf{x}^{(j)}) \\ &= \phi(\mathbf{x}^{(j)})^T \phi(\mathbf{x}^{(i)}) \\ &= K(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) \\ &= \mathcal{K}_{j,i}\end{aligned}$$

Mercer's Theorem: 2

$$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$$
$$\mathcal{K}_{i,j} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

Wähle \mathbf{z} beliebig:

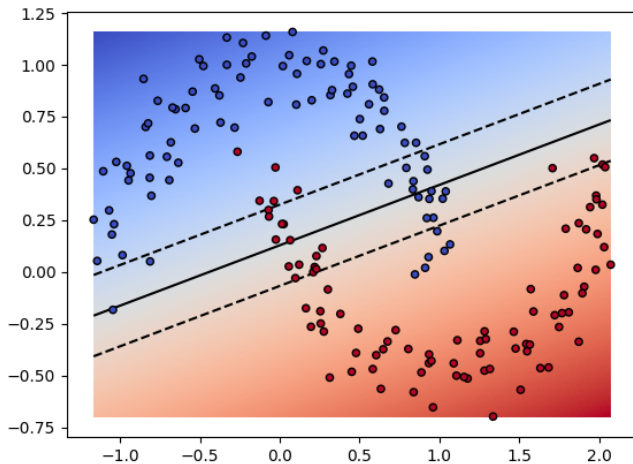
$$\begin{aligned}\mathbf{z}^T \mathcal{K} \mathbf{z} &= \sum_i \sum_j \mathbf{z}_i \mathcal{K}_{i,j} \mathbf{z}_j \\&= \sum_i \sum_j \mathbf{z}_i \phi(\mathbf{x}^{(i)})^T \phi(\mathbf{x}^{(j)}) \mathbf{z}_j \\&= \sum_i \sum_j \mathbf{z}_i \sum_k \phi_k(\mathbf{x}^{(i)}) \phi_k(\mathbf{x}^{(j)}) \mathbf{z}_j \\&= \sum_k \sum_i \sum_j \mathbf{z}_i \phi_k(\mathbf{x}^{(i)}) \phi_k(\mathbf{x}^{(j)}) \mathbf{z}_j \\&= \sum_k \left(\sum_i \mathbf{z}_i \phi_k(\mathbf{x}^{(i)}) \right)^2 \\&\geq 0\end{aligned}$$

Verschiedene Kernel in der Praxis

- ▶ Linearer Kernel
- ▶ Polynomiell
- ▶ Gauß'schen Kernel
- ▶ Esotherische Kernel

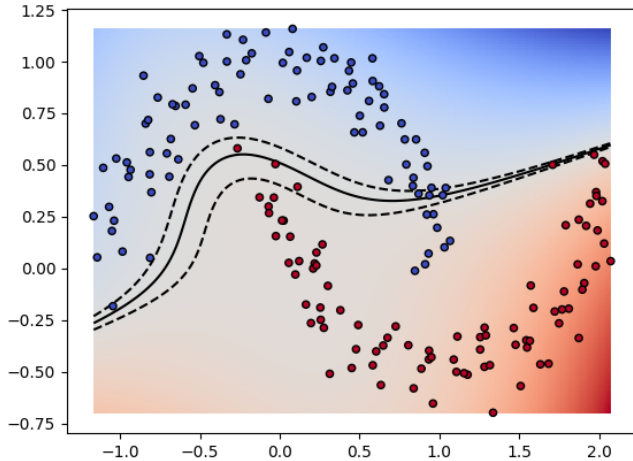
Verschiedene Kernel in der Praxis: Linear

$$K(\mathbf{v}, \mathbf{w}) = \mathbf{v}^T \mathbf{w}$$



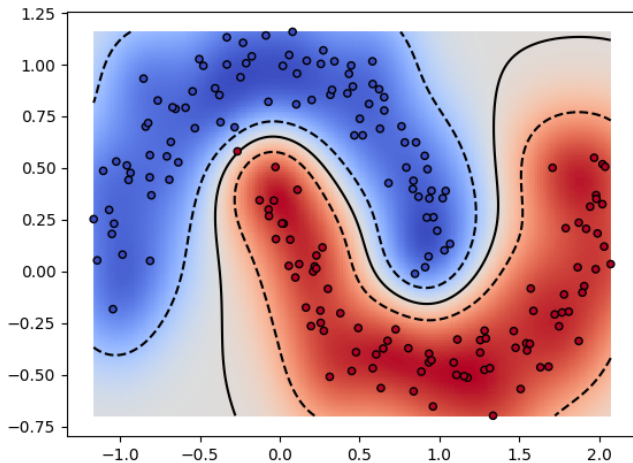
Verschiedene Kernel in der Praxis: Polynomiell

$$K(\mathbf{v}, \mathbf{w}) = (\mathbf{v}^T \mathbf{w} + c)^d$$



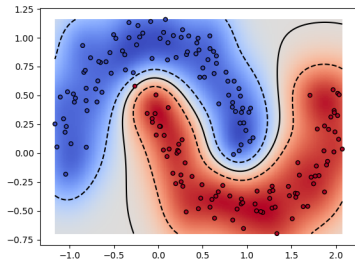
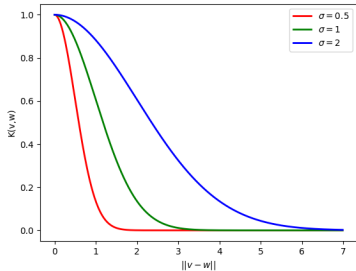
Verschiedene Kernel in der Praxis: Gauß

$$K(\mathbf{v}, \mathbf{w}) = \exp\left(-\frac{\|\mathbf{v}-\mathbf{w}\|^2}{2\sigma^2}\right)$$



Verschiedene Kernel in der Praxis: Gauß

$$K(\mathbf{v}, \mathbf{w}) = \exp\left(-\frac{\|\mathbf{v}-\mathbf{w}\|^2}{2\sigma^2}\right)$$



Verschiedene Kernel in der Praxis: Esoterisch

$$K: D \times D \rightarrow \mathbb{R}$$

Beispiel: String Kernel

- Misst Ähnlichkeit von zwei Strings

- Vergleicht verschiedene Aspekte

 - e.g. Subsequenzen, gemeinsame Wörter, Länge, ...

- ▶ SVMs in ihrer Standardform haben Probleme nicht linear trennbare Datensätze zu klassifizieren
- ▶ Mit $\phi(x)$ Daten in einen Raum abbilden, wo dies möglich ist
- ▶ Kernel nutzen, um die daraus folgende Berechnung zu vereinfachen