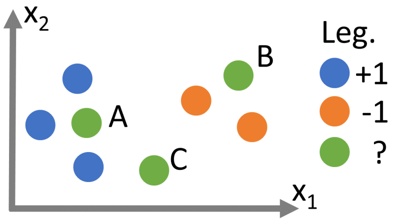
Support Vector Machines mit Kerneln

Tim Schlottmann, Hendrik Sieck, Jonas Krug

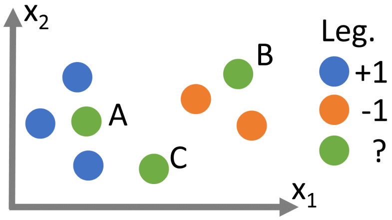
26.01.2018

Motivation



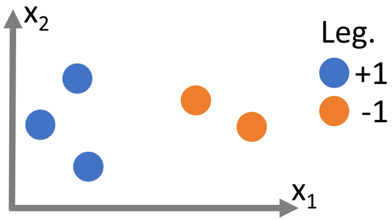
- Klassifizierung von Objekten
- Schnell und effizient
- Möglichst genau

Themenvorstellung



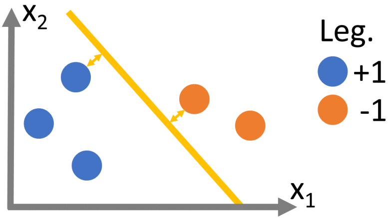
- Support Vector Machines (deutsch: Stützvektor Maschine)
- Binärer Klassifierierer

Lineare Klassifizierung



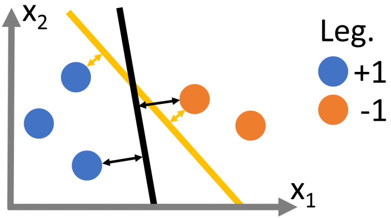
- Wie trenne ich die beiden Klassen voneinander?

Lineare Klassifizierung



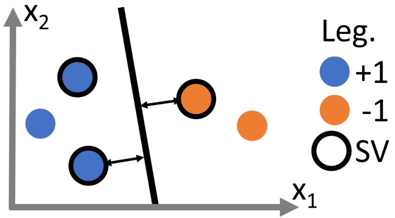
- Wie lege ich die Hyperebene am besten?

Lineare Klassifizierung



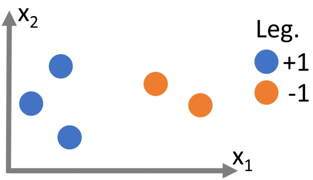
- Anderer Name von Support Vector Machines: Large Margin Classifier

Lineare Klassifizierung



- Anderer Name von Support Vector Machines: Large Margin Classifier

Definitionen



- $m = 5$
- Trainingsset:

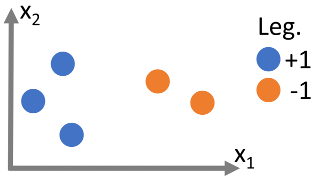
$$S \in (\boldsymbol{x} \times \boldsymbol{y})^m$$
$$= \begin{pmatrix} 1 & 0.5 & +1 \\ 3.5 & 1 & -1 \\ & \vdots & \end{pmatrix}$$

- Anzahl an Trainingspunkten $m \in \mathbb{R}$
- Input $\boldsymbol{x} \in \mathbb{R}^N$
- Output $y \in \{-1, +1\}$
- Trainingsset $S \in (\mathbb{R}^N \times \{-1, +1\})^m$

- Hypothese

$$\begin{aligned} h : \boldsymbol{x} &\longrightarrow \boldsymbol{y} \\ \boldsymbol{x}_i &\longmapsto \{+1, -1\} \end{aligned}$$

Definitionen



- $m = 5$
- Trainingsset:

$$S \in (\boldsymbol{x} \times \boldsymbol{y})^m$$

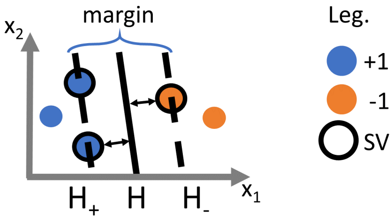
$$= \begin{pmatrix} 1 & 0.5 & +1 \\ 3.5 & 1 & -1 \\ & \vdots & \end{pmatrix}$$

- Anzahl an Trainingspunkten $m \in \mathbb{R}$
- Input $\boldsymbol{x} \in \mathbb{R}^N$
- Output $y \in \{-1, +1\}$
- Trainingsset $S \in (\mathbb{R}^N \times \{-1, +1\})^m$

- Hypothese

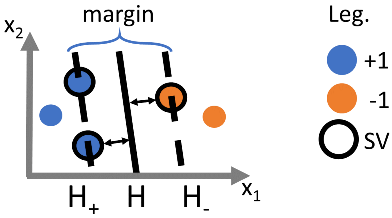
$$h : \boldsymbol{x} \rightarrow \boldsymbol{y}$$
$$\boldsymbol{x}_i \mapsto \{+1, -1\}$$

Large Margin Classifier



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

Large Margin Classifier



- Hyperebene: $H' = \boldsymbol{w}'^T \boldsymbol{x} + b' = 0$
- Gutter constraint (GC):

$$\boldsymbol{w}^T \boldsymbol{x}_i + b = y_i, \quad \forall \text{ support vectors } \in \boldsymbol{x}$$

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) = 1 \quad \forall \text{ support vectors } \in \boldsymbol{x}$$

- Um GC zu erfüllen: $\boldsymbol{w}$ und b werden um $c \in \mathbb{R}$ skaliert:

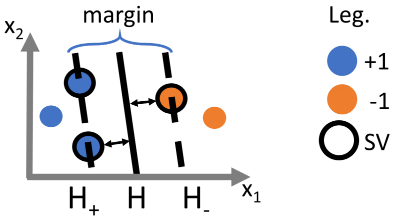
$$H = c(\boldsymbol{w}^T \boldsymbol{x} + b) = 0$$

$$\boldsymbol{w} = c^T \boldsymbol{w}'$$

$$b = c^T b'$$

- H heißt auch kanonische Hyperebene

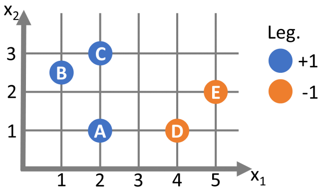
Large Margin Classifier



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

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

Beispiel I



- Hyperebene: $H = \boldsymbol{w}^T \boldsymbol{x} + b = 0$
- Gutter constraint: $\boldsymbol{w}^T \boldsymbol{x}_i + b = y_i, \quad \forall$ s. v. $\in \boldsymbol{x}$

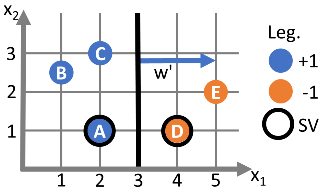
- Graphische Bestimmung der Hyperebenenparameter:

$$x = 3$$

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

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

Beispiel I



- Hyperebene: $H = \boldsymbol{w}^T \boldsymbol{x} + b = 0$
- Gutter constraint: $\boldsymbol{w}^T \boldsymbol{x}_i + b = y_i, \quad \forall$ s. v. $\in \boldsymbol{x}$

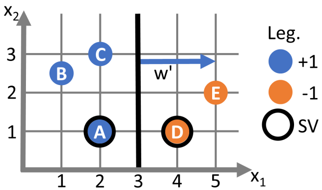
- Graphische Bestimmung der Hyperebenenparameter:

$$x = 3$$

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

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

Beispiel II



- Hyperebene: $H = \boldsymbol{w}^T \boldsymbol{x} + b = 0$
- Gutter constraint:
  $\boldsymbol{w}^T \boldsymbol{x}_i + b = y_i, \quad \forall$ s. v. $\in \boldsymbol{x}$

- 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:

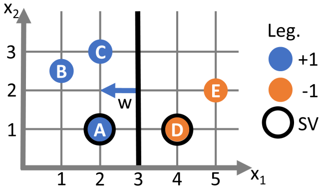
$$\boldsymbol{w} = \begin{pmatrix}-1\\0\end{pmatrix}$$

$$b = 3$$

- Kontrolle mit Punkt D:

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

Beispiel II



- Hyperebene: $H = \boldsymbol{w}^T\boldsymbol{x} + b = 0$
- Gutter constraint:
  $\boldsymbol{w}^T\boldsymbol{x}_i + b = y_i, \quad \forall$ s. v. $\in \boldsymbol{x}$

- 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:

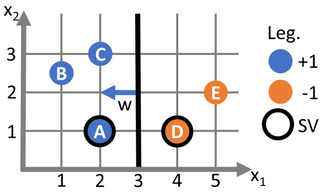
$$\boldsymbol{w} = \begin{pmatrix}-1\\0\end{pmatrix}$$

$$b = 3$$

- Kontrolle mit Punkt D:

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

Beispiel II



- Hyperebene: $H = \boldsymbol{w}^T \boldsymbol{x} + b = 0$
- Gutter constraint:
  $\boldsymbol{w}^T \boldsymbol{x}_i + b = y_i, \quad \forall$ s. v. $\in \boldsymbol{x}$

- 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:

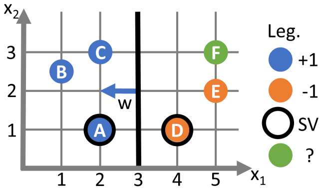
$$\boldsymbol{w} = \begin{pmatrix}-1\\0\end{pmatrix}$$

$$b = 3$$

- Kontrolle mit Punkt D:

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

Beispiel III



- Klassifizierung:

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

- Parameter der kanonischen Hyperebene

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

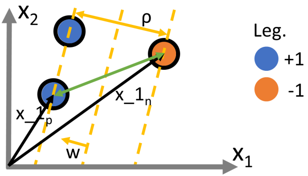
$$b = 3$$

- 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:

$$\rho = (\boldsymbol{x_p} - \boldsymbol{x_n})^T \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

- Gutter constraint:

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

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

$$\max \frac{2}{\|\boldsymbol{w}\|} \Leftrightarrow \min \|\boldsymbol{w}\| \Leftrightarrow \min \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{u.d.N. } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \geq 0$$

Lagrange multipliers

- Langrange multipliers:

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

- Suche nach den Extremum

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

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

- $\boldsymbol{w}$ in L eingesetzt:

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

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

- Ziel: $\max L$
- Entscheidungsfunktion:

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

Supportiveness values α

- Eigenschaften:

$$\boldsymbol{\alpha} \geq 0$$

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

- $\sum_{\substack{\text{s.v.} \\ i}} \alpha_i y_i = 0 \Leftrightarrow \sum_{\substack{\text{pos. s.v.} \\ p}} \alpha_p = \sum_{\substack{\text{neg. s.v.} \\ n}} \alpha_n$
- $\sum_i \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{w} \Leftrightarrow \boldsymbol{w} = \sum_p \alpha_p \boldsymbol{x}_p - \sum_n \alpha_n \boldsymbol{x}_n$

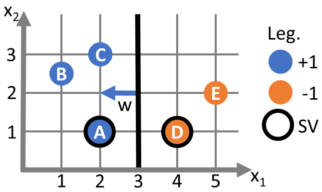
Supportiveness values α

- Eigenschaften:

$$\boldsymbol{\alpha} \geq 0$$

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

- $\sum\limits_{\substack{\text{s.v.} \\ i}} \alpha_i y_i = 0 \Leftrightarrow \sum\limits_{\substack{\text{pos. s.v.} \\ p}} \alpha_p = \sum\limits_{\substack{\text{neg. s.v.} \\ n}} \alpha_n$
- $\sum\limits_i \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{w} \Leftrightarrow \boldsymbol{w} = \sum\limits_p \alpha_p \boldsymbol{x}_p - \sum\limits_n \alpha_n \boldsymbol{x}_n$



- Berechnung von α_A und α_D ergibt:

$$\alpha_A = 0.5$$
$$\alpha_D = 0.5$$

- Berechnung von α_C ergibt:

$$\alpha_C = 0$$

Zusammenfassung

- Kanonische Hyperebene:

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

- Gesucht: Parameter $\boldsymbol{w}$ und b
- Supportive values α:

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

$$\sum_i \alpha_i y_i = 0$$

- Entscheidungsfunktion H:

$$h(\boldsymbol{x}) = \sum_j \alpha_j y_j \boldsymbol{x}_j^T \boldsymbol{x} + b \begin{cases} \geq 0 & \Rightarrow y_x = +1 \\ \leq 0 & \Rightarrow y_x = -1 \end{cases}$$

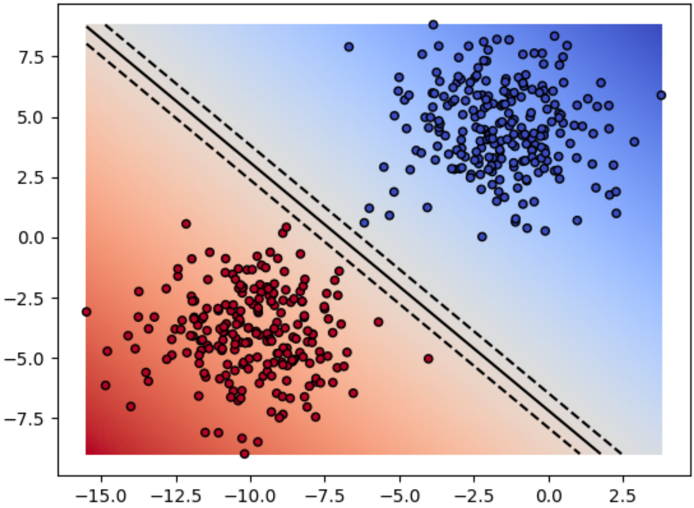
- Supportvektoren

$$y_i(\sum_j \alpha_j y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + b) = 1 \quad \forall \text{ s.v. } \boldsymbol{x}_i \in \boldsymbol{x}$$

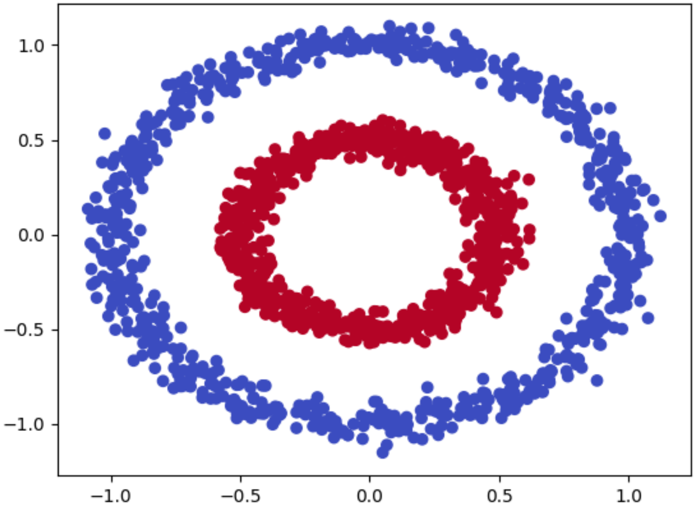
$$\alpha_i \begin{cases} > 0 & \text{wenn } x_i \text{ ein support vector ist} \\ = 0 & \text{sonst} \end{cases}$$

Quellen

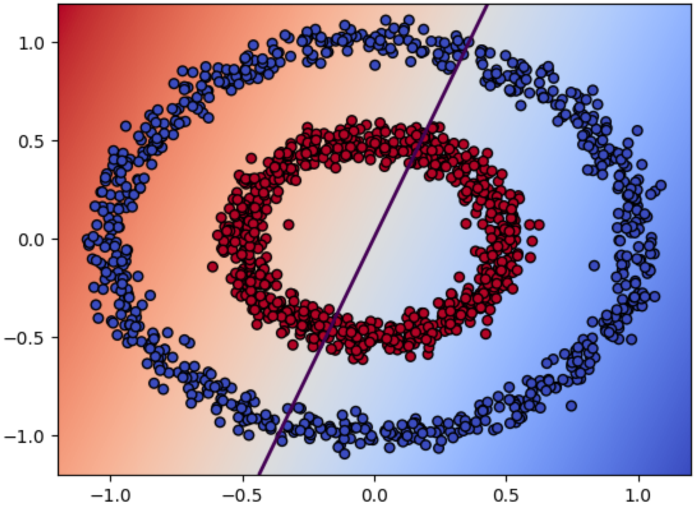
Einführung: Linear seperierbare Daten



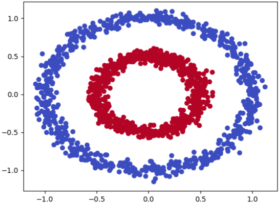
Einführung: Linear nicht seperierbare Daten 1



Einführung: Linear nicht seperierbare Daten 2

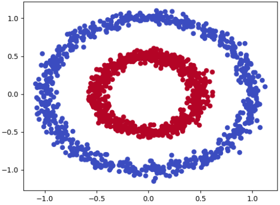


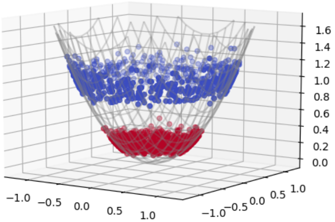
Ansatz: Einführung



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

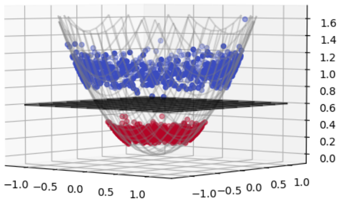
Ansatz: Einführung

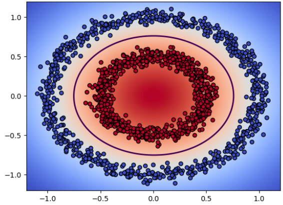




$$\phi(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$

Ansatz: Abbildfunktion $\phi(x)$





Veränderung der Funktionen

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

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

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

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

Veränderung der Funktionen

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

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

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

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

Genauere Klärung von ϕ

Abbildfunktion

$$\begin{aligned} \phi : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ \boldsymbol{x} &\longmapsto \boldsymbol{f} \end{aligned}$$

Probleme:

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

Genauere Klärung von ϕ

Abbildfunktion

$$\begin{aligned} \phi : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ \boldsymbol{x} &\longmapsto \boldsymbol{f} \end{aligned}$$

Probleme:

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

Beispiel Kernel

$$\phi(\boldsymbol{x}) = (1 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ \ldots \ x_1^2 x_2^2 \ \ldots \ \sqrt{2}x_1 x_2 \ \sqrt{2}x_1 x_3 \ \ldots)$$

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

Beispiel Kernel

$$\phi(\boldsymbol{x}) = (1 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ \ \ldots \ \ x_1^2 x_2^2 \ \ldots \ \sqrt{2}x_1 x_2 \ \sqrt{2}x_1 x_3 \ \ \ldots)$$

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

Einführung von Kerneln

$$K(\boldsymbol{v}, \boldsymbol{w}) = \phi(\boldsymbol{v})^T \phi(\boldsymbol{w})$$
$$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

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

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

Einführung von Kerneln

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

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

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

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

Mercer's Theorem: 1

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

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

Mercer's Theorem: 1

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

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

Mercer's Theorem: 2

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

Wähle $\boldsymbol{z}$ beliebig:

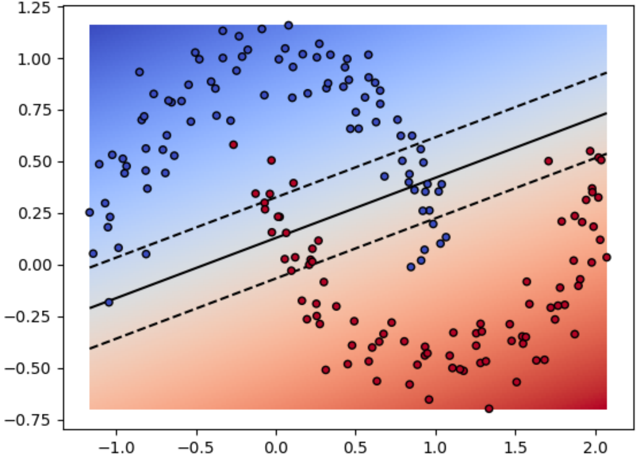
$$
\begin{aligned}
\boldsymbol{z}^T \mathcal{K} \boldsymbol{z} &= \sum_i \sum_j \boldsymbol{z}_i \mathcal{K}_{i,j} \boldsymbol{z}_j \\
&= \sum_i \sum_j \boldsymbol{z}_i \phi(\boldsymbol{x}^{(i)})^T \phi(\boldsymbol{x}^{(j)}) \boldsymbol{z}_j \\
&= \sum_i \sum_j \boldsymbol{z}_i \sum_k \phi_k(\boldsymbol{x}^{(i)})^T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

Verschiedene Kernel in der Praxis

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

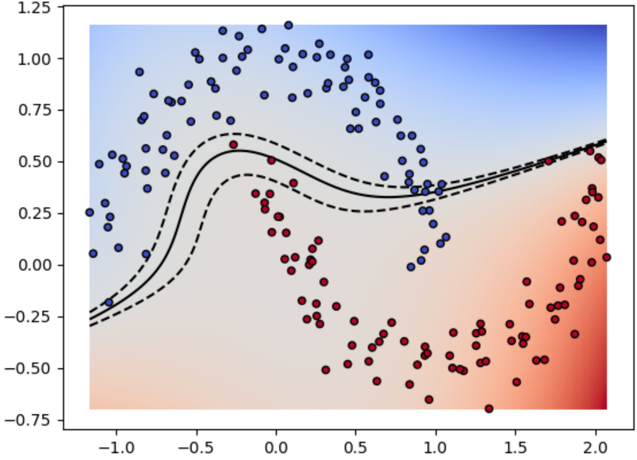
Verschiedene Kernel in der Praxis: Linear

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



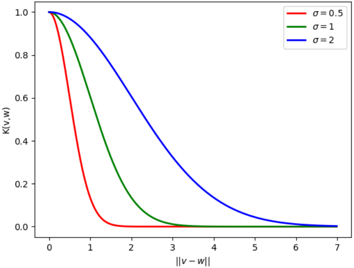
Verschiedene Kernel in der Praxis: Polynomiell

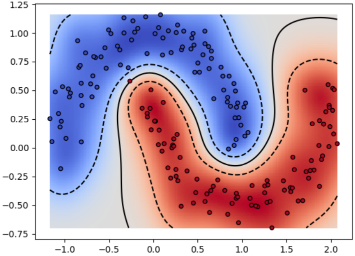
$$K(\boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{v}^T \boldsymbol{w} + c)^d$$



Verschiedene Kernel in der Praxis: Gauß

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





Verschiedene Kernel in der Praxis: Ësotherisch"

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

Beispiel: String Kernel

- Misst Ähnlichkeit von zwei Strings
- Vergleicht verschiedene Aspekte
    - e.g. Subequenzen, gemeinsame Wörter, Länge, ...

Zusammenfassung

- SVMs in ihrer Standartform haben Probleme nicht liear 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