Maschinelles Lernen 06

Prof. Dr. David Spieler – david.spieler@hm.edu

Hochschule München

2. Dezember 2019

Support Vector Machines

Support Vector Machines

Support Vector Machines

Support Vector Machines (SVM) ist eine Methode welche in der 1990er Jahren in der Informatik zur

- Klassifikation und
- Regression

entwickelt wurde. Die Methode ist mittlerweile sehr weit verbreitet, da sie oft als sehr gute *out-of-the-box* Methode angesehen wird.

Support Vector Machines

Maximal Margin Klassifikatoren

Ausgangsbasis für die Entwicklung von SVM sind die Maximal Margin Klassifikation. Hier betrachten wir die binäre Klassifikation von Punkten $\mathbf{x}^{(i)} \in \mathbb{R}^d$ mit $y^{(i)} \in \{-1, 1\}$ und nehmen dabei an, dass wir eine separierende Hyperebene im $\mathbb{R}^d$ finden können, welche die beiden Klassen perfekt trennt.

Support Vector Machines

Maximal Margin Klassifikatoren

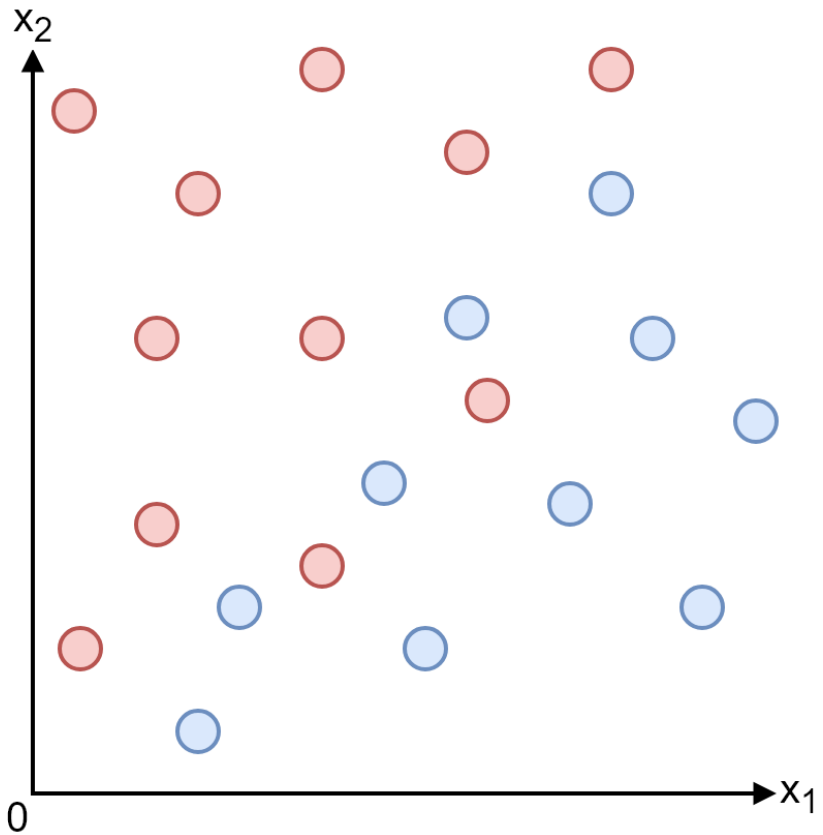

Abbildung 1: Scatterplot eines Datensatzes, welcher nicht durch eine Hyperebene trennbar ist.

Support Vector Machines

Maximal Margin Klassifikatoren

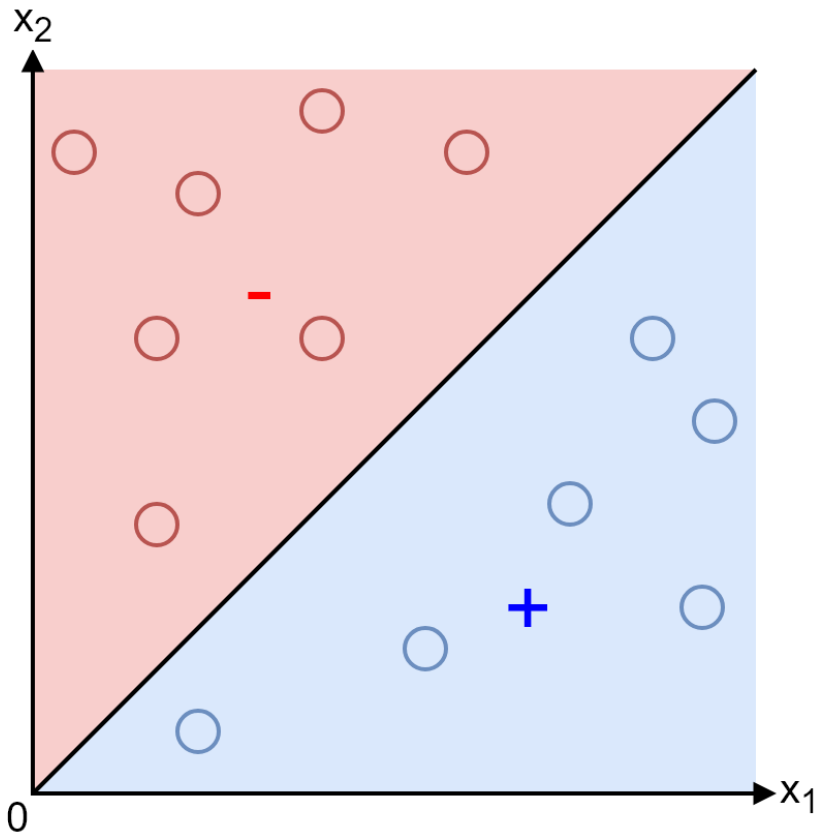

Abbildung 2: Binärer Klassifikator mit Hilfe einer separierenden Hyperebene, welche die beiden Klassen perfekt trennt.

Support Vector Machines

Maximal Margin Klassifikatoren

Eine separierende Hyperebene ist eine Hyperebene mit Parametern $\mathbf{w}_0, \mathbf{w}$ welche die Eigenschaft besitzt, dass für alle Datenpunkte $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{T} \subseteq \mathbb{R}^d \times \{-1, 1\}$ gilt, dass

$$\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)} > 0 \text{ falls } y^{(i)} = 1$$

und

$$\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)} < 0 \text{ falls } y^{(i)} = -1.$$

Diese beiden Ungleichungen ergeben zusammengefasst

$$y^{(i)}(\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)}) > 0.$$

Support Vector Machines

Maximal Margin Klassifikatoren

Wir können also mit Hilfe der Funktion

$$f(\mathbf{x}) = \mathbf{w}_0 + \mathbf{w}^T\mathbf{x}$$

den gesuchten binären Klassifikator definieren als

$$\operatorname{sgn}(f(\mathbf{x})).$$

Aber wir können an $f(\mathbf{x})$ auch ablesen, wie sicher sich der Klassifikator ist.

- Ist $|f(\mathbf{x})| \approx 0$, so befindet sich der Punkt $\mathbf{x}$ sehr nahe an der Hyperebene $\mathbf{w}_0 + \mathbf{w}^T\mathbf{x} = 0$ und ist daher eher ein Wackelkandidat.
- Falls $|f(\mathbf{x})| \gg 0$, so befindet sich der Punkt sehr weit entfernt von der Hyperebene im positiven oder negativen Regime und die Klassifikation ist der relativ sicher.

Support Vector Machines

Maximal Margin Klassifikatoren

Diesen Zusammenhang und dass es meist eine unendlich Anzahl von potentiellen separierenden Hyperebenen im trennbaren Fall gibt, wollen wir bei Maximal Margin Klassifikatoren ausnutzen, um einen Klassifikator zu erstellen, welcher die Unsicherheit minimiert.

Support Vector Machines

Maximal Margin Klassifikatoren

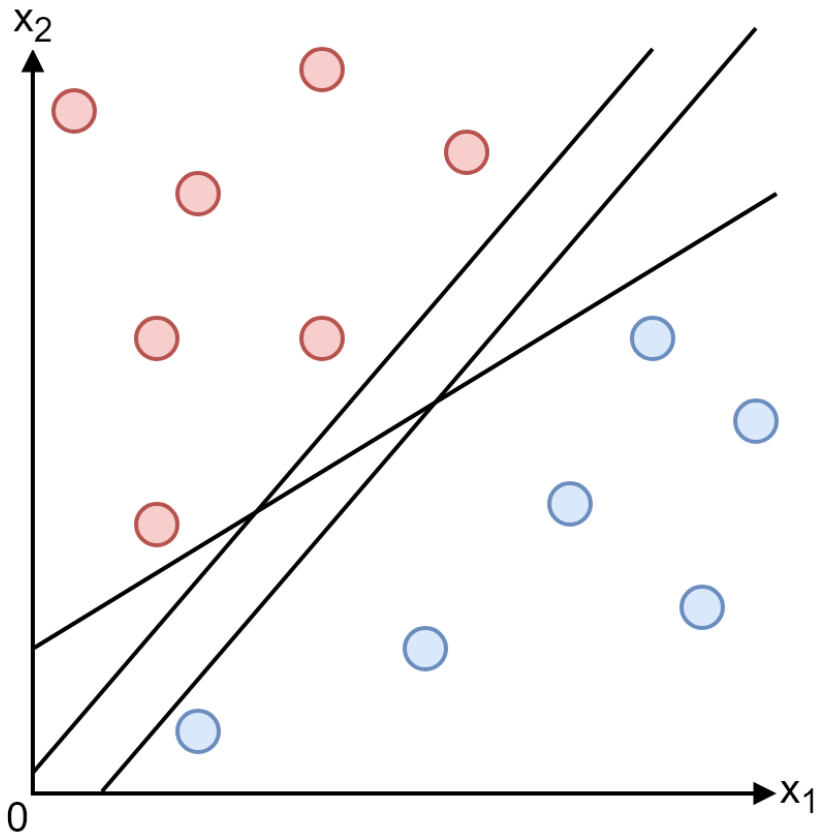

Abbildung 3: Scatterplot eines Datensatzes, welcher durch unendlich viele Geraden trennbar ist. Es sind nur drei Möglichkeiten exemplarisch dargestellt, andere Geraden erhält man durch Rotation und Verschiebung.

Support Vector Machines

Maximal Margin Klassifikatoren

Die Idee ist, die Maximal Margin Hyperebene zu bestimmen, also die separierende Hyperebene, welche am weitesten von den Datenpunkten entfernt ist.

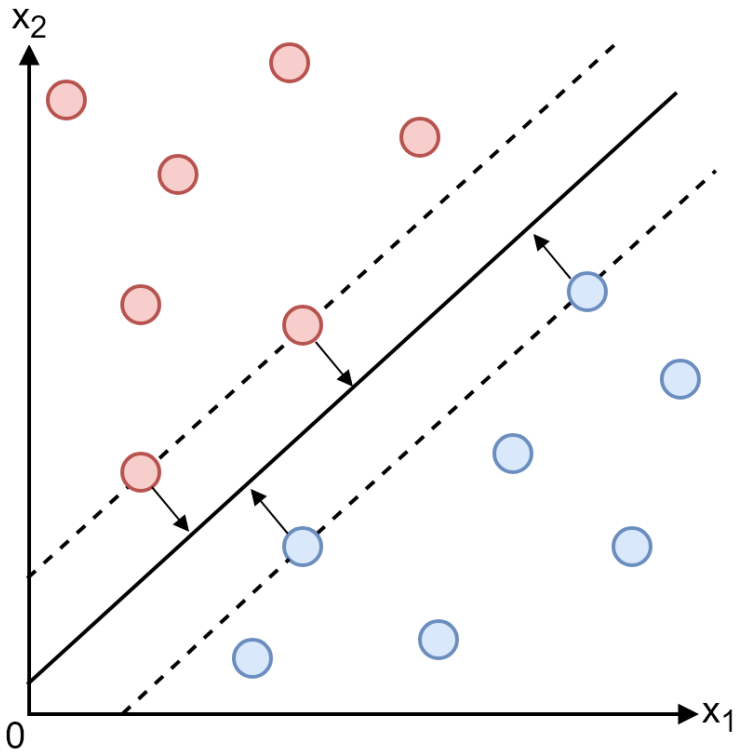

Abbildung 4: Maximum Margin Hyperebene (durchgezogene Gerade) mit dem maximalen Abstand (Margin) von den Datenpunkten (Pfeile).

Support Vector Machines

Maximal Margin Klassifikatoren

In der vorherigen Abbildung gab es genau vier Datenpunkte, welche der Maximal Margin Hyperebene am nächsten liegen. Diese Punkte (2-d Vektoren) nennt man auch Support Vektoren, da sie die Hyperebene *stützen*. Falls sie verschoben werden, ändert sich auch die Hyperebene. Wichtig ist auch, dass die Maximal Margin Hyperebene nur von einer kleinen Anzahl von Support Vektoren gestützt wird.

Support Vector Machines

Maximal Margin Klassifikatoren

Um die Maximal Margin Hyperebene zu berechnen, muss das Optimierungsproblem

$$\arg\min_{\mathbf{w}_0,\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2$$

unter den Nebenbedingungen

$$y^{(i)}(\mathbf{w}_0 + \mathbf{w}^T\mathbf{x}^{(i)}) \geq 1 \ \forall 1 \leq i \leq n$$

gelöst werden.

Support Vector Machines

Maximal Margin Klassifikatoren

Um solche Optimierungsprobleme zu lösen, wenden wir das Karush-Kuhn-Tucker Theorem (KKT) an, welches besagt, dass für die Maxima einer Funktion

$$f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$$

mit n Nebenbedingungen $(1 \leq i \leq n)$

$$g_i(\mathbf{x}) \leq 0$$

bzgl. der Lagrange-Funktion

$$\mathcal{L}(\mathbf{x}) = f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \cdot g_i(\mathbf{x})$$

mit $\lambda_i \in \mathbb{R}_{\geq 0}$ die notwendigen Bedingungen

- $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}) = \mathbf{0}$
- $\lambda_i \cdot g_i(\mathbf{x}) = 0 \quad \forall i \in \{1, \ldots, n\}$

gelten.

Support Vector Machines

Maximal Margin Klassifikatoren

Anstatt $\frac{1}{2}||\mathbf{w}||_2^2$ zu minimieren, können wir auch $-\frac{1}{2}||\mathbf{w}||_2^2$ maximieren und erhalten durch Einführung von Lagrange-Multiplikatoren, wie durch das KKT gefordert die Primale Lagrange-Funktion

$$\mathcal{L}_P(\mathbf{w}_0, \mathbf{w}) = -\frac{1}{2}||\mathbf{w}||_2^2 - \sum_{i=1}^{n} \lambda_i (1 - y^{(i)}(\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)})).$$

Außerdem gilt

$$\nabla_{\mathbf{w}} \mathcal{L}_P(\mathbf{w}_0, \mathbf{w}) = -\mathbf{w} + \sum_{i=1}^{n} \lambda_i y^{(i)} \mathbf{x}^{(i)} \stackrel{!}{=} \mathbf{0},$$

wodurch

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y^{(i)} \mathbf{x}^{(i)}.$$

Support Vector Machines

Maximal Margin Klassifikatoren

Zusätzlich gilt

$$\frac{\partial}{\partial \mathbf{w}_0} \mathcal{L}_P(\mathbf{w}_0, \mathbf{w}) = \sum_{i=1}^{n} \lambda_i y^{(i)} \stackrel{!}{=} 0.$$

Setzen wir alles, was wir nun wissen in $\mathcal{L}_P$ ein und räumen die Gleichung auf (im Detail machen wir das später bei der Support Vector Regression), so erhalten wir die Duale Lagrange-Funktion

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\,T} \mathbf{x}^{(j)},$$

welche unter der oberen Bedingung und $\lambda_i \geq 0$ für alle $i \in \{1, \ldots, n\}$ optimiert wird.

Support Vector Machines

Maximal Margin Klassifikatoren

Wir wollen nun diese Methode an zwei Beispielen ausprobieren. Im ersten Beispiel haben wir zwei Datenpunkte $\mathbf{x}^{(1)} = (1, 2)^T$ und $\mathbf{x}^{(2)} = (2, 1)^T$ mit den Klassen $y^{(1)} = 1$ und $y^{(2)} = -1$.

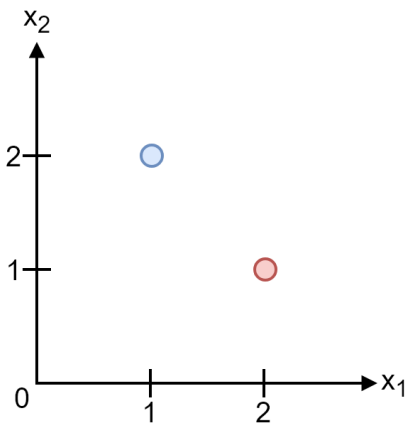

Abbildung 5: Scatterplot des Datensatzes des ersten Beispiels.

Support Vector Machines

Maximal Margin Klassifikatoren

Durch Einsetzen erhalten wir $\mathcal{L}_D(\lambda_1, \lambda_2) =$

$$=\lambda_1 + \lambda_2 - \frac{1}{2}\lambda_1^2 1^2(1^2 + 2^2) - \frac{1}{2}\lambda_2^2(-1)^2(2^2 + 1^2) - \frac{1}{2}\lambda_1\lambda_2 1(-1)(2+2)$$
$$-\frac{1}{2}\lambda_2\lambda_1(-1)1(2+2) = -\frac{5}{2}\lambda_1^2 - \frac{5}{2}\lambda_2^2 + 4\lambda_1\lambda_2 + \lambda_1 + \lambda_2$$

was wir wiederum unter der Nebenbedingung

$$g(\lambda_1, \lambda_2) = \lambda_1 - \lambda_2 = 0$$

optimieren müssen.

Support Vector Machines

Maximal Margin Klassifikatoren

Aufgrund der Nebenbedingung wissen wir nun, dass $\lambda_1 = \lambda_2 = \lambda$ und damit

$$\mathcal{L}_D(\lambda, \lambda) = -\lambda^2 + 2\lambda$$

was das globales Maximum 1 bei $\lambda = 1$ hat.

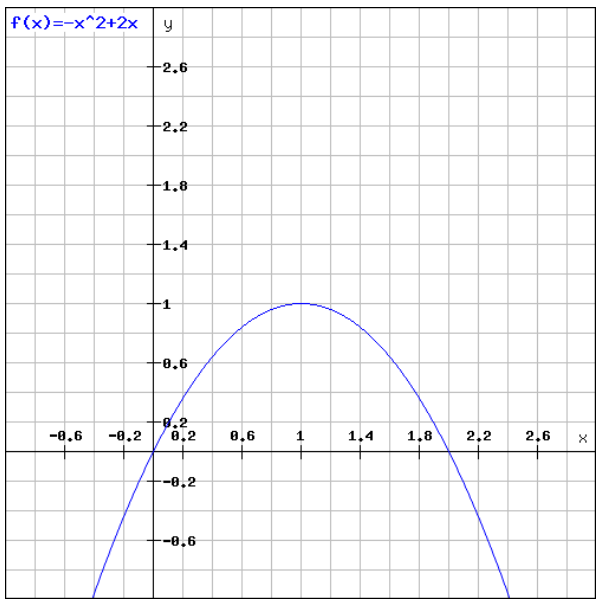

Abbildung 6: Plot der Funktion $f(x) = -x^2 + 2x$.

Support Vector Machines

Maximal Margin Klassifikatoren

Damit ist $\lambda_1 = \lambda_2 = 1$ und

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y^{(i)} \mathbf{x}^{(i)} = (1,2)^T - (2,1)^T = (-1,1)^T.$$

Es fehlt noch $\mathbf{w}_0$, welches wir mit den ursprünglichen Nebenbedingungen

$$y^{(i)}(\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 \ \forall 1 \leq i \leq n$$

in unserem Fall $\mathbf{w}_0 + 1 \geq 1$ und $-\mathbf{w}_0 + 1 \geq 1$ auf $\mathbf{w}_0 = 0$ klären. Somit erhalten wir letztendlich

$$f(\mathbf{x}) = \mathbf{w}_0 + \mathbf{w}^T \mathbf{x} = \mathbf{x}_2 - \mathbf{x}_1.$$

Support Vector Machines

Maximal Margin Klassifikatoren

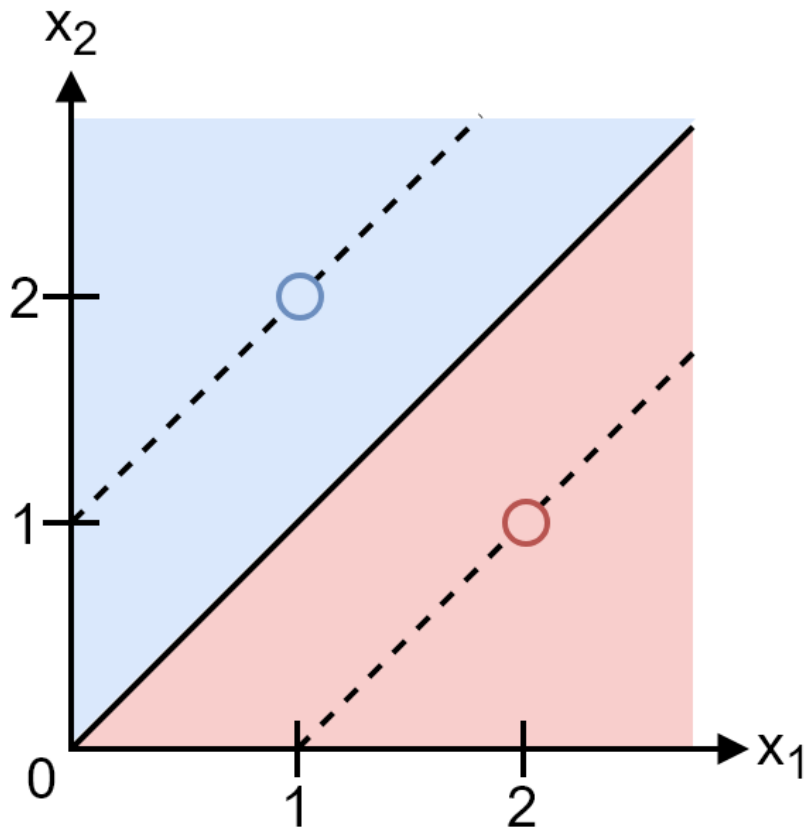

Abbildung 7: Der finale Maximal Margin Klassifikator $f(\mathbf{x}) = \text{sgn}(\mathbf{x}_2 - \mathbf{x}_1)$ mit der Entscheidungsgrenze $\mathbf{x}_2 - \mathbf{x}_1 = 0$.

Support Vector Machines

Maximal Margin Klassifikatoren

Wir möchten ein ähnliches Experiment mit drei Datenpunkten $\mathbf{x}^{(1)} = (1,2)^T$, $\mathbf{x}^{(2)} = (2,4)^T$ und $\mathbf{x}^{(3)} = (2,1)^T$ mit den Klassen $y^{(1)} = 1$, $y^{(2)} = 1$ und $y^{(3)} = -1$ wiederholen.

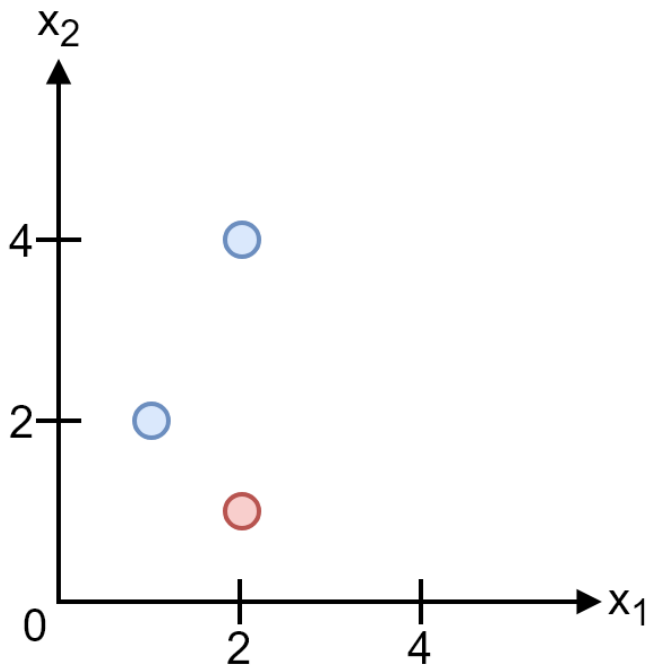

Abbildung 8: Scatterplot des Datensatzes des zweiten Beispiels.

Support Vector Machines

Maximal Margin Klassifikatoren

Einsetzen der Punkte ergibt das duale Problem der Maximierung von

$$h(\lambda_1, \lambda_2, \lambda_3) = \lambda_1+\lambda_2+\lambda_3-\frac{5}{2}\lambda_1^2-10\lambda_2^2-\frac{5}{2}\lambda_3^2-10\lambda_1\lambda_2+4\lambda_1\lambda_3+8\lambda_2\lambda_3$$

unter der Nebenbedingung

$$g(\lambda_1, \lambda_2, \lambda_3) = \lambda_1 + \lambda_2 - \lambda_3 = 0$$

Dieses Mal können wir das Problem nicht sofort vereinfachen und müssen ein weiteres Mal Lagrange bemühen und den Lagrange-Multiplikator λ^* in der Optimierung von

$$z(\lambda_1, \lambda_2, \lambda_3, \lambda^*) = h(\lambda_1, \lambda_2, \lambda_3) + \lambda^* g(\lambda_1, \lambda_2, \lambda_3)$$

einführen.

Support Vector Machines

Maximal Margin Klassifikatoren

Wir lösen nun das LGS

$$\nabla_{\lambda_1,\lambda_2,\lambda_3} z(\lambda_1,\lambda_2,\lambda_3,\lambda^*) = \begin{bmatrix} 1-5\lambda_1-10\lambda_2+4\lambda_3+\lambda^* \\ 1-20\lambda_2-10\lambda_1+8\lambda_3+\lambda^* \\ 1-5\lambda_3+4\lambda_1+8\lambda_2-\lambda^* \end{bmatrix} = \mathbf{0}$$

mit $g(\lambda_1,\lambda_2,\lambda_3)=0$ und erhalten $\lambda_1=\frac{4}{3}$, $\lambda_2=-\frac{2}{9}$, $\lambda_3=\frac{10}{9}$ und $\lambda^*=-1$. Offensichtlich kann dies mit $\lambda_2<0$ nicht die Lösung sein, wir suchen daher an den Grenzen ($\lambda_i=0$) des Suchraums weiter.

Support Vector Machines

Maximal Margin Klassifikatoren

Wir untersuchen also $g(\lambda_1, \lambda_2, \lambda_3) = \lambda_1 + \lambda_2 - \lambda_3 = 0$ und finden heraus, dass $\lambda_3 \neq 0$, da sonst $\lambda_1 = \lambda_2 = \lambda_3 = 0$.

- $\lambda_1 = 0 \Rightarrow \lambda_2 = \lambda_3$: $h(\lambda_1, \lambda_2, \lambda_3) = -\frac{9}{2}\lambda_3^2 + 2\lambda_3$ mit dem Maximum $\frac{2}{9}$ bei $\lambda_3 = \lambda_2 = \frac{2}{9}$
- $\lambda_2 = 0 \Rightarrow \lambda_1 = \lambda_3$: $h(\lambda_1, \lambda_2, \lambda_3) = -\lambda_3^2 + 2\lambda_3$ mit dem Maximum 1 bei $\lambda_1 = \lambda_3 = 1$, was auch das globale Maximum ist.

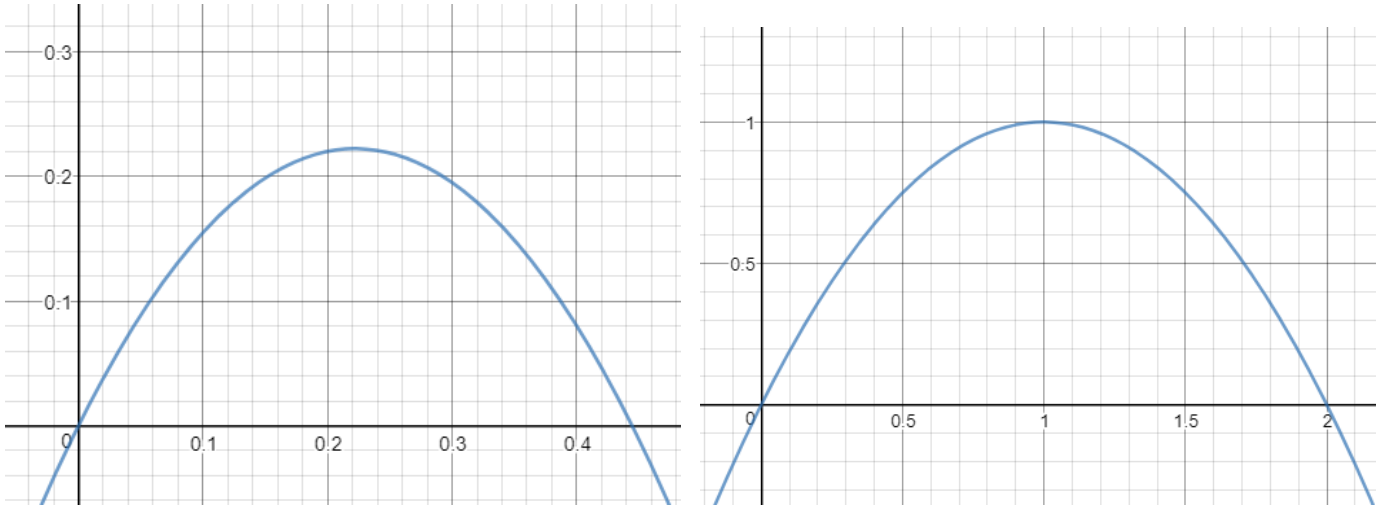

Abbildung 9: Plots der beiden zu optimierenden Funktionen.

Support Vector Machines

Maximal Margin Klassifikatoren

Wir erhalten mit $\lambda_1 = 1$, $\lambda_2 = 0$ und $\lambda_3 = 1$ den gleichen Klassifikator wie im vorherigen Beispiel, jedoch ist an dessen Definition Datenpunkt $\mathbf{x}^{(2)}$ nicht mehr beteiligt.

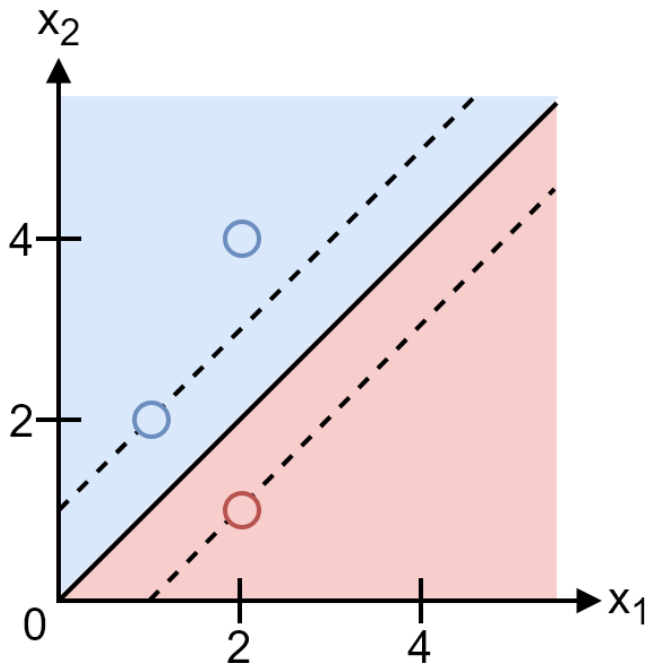

Abbildung 10: Der finale Maximal Margin Klassifikator $f(\mathbf{x}) = \text{sgn}(\mathbf{x}_2 - \mathbf{x}_1)$ mit der Entscheidungsgrenze $\mathbf{x}_2 - \mathbf{x}_1 = 0$.

Support Vector Machines

Support Vector Klassifikation

Die Maximum Margin Klassifikation hat zwei Hauptprobleme, denn sie

- funktioniert nur im perfekt separierbaren Fall und
- ist nicht robust gegenüber einzelnen Datenpunkten.

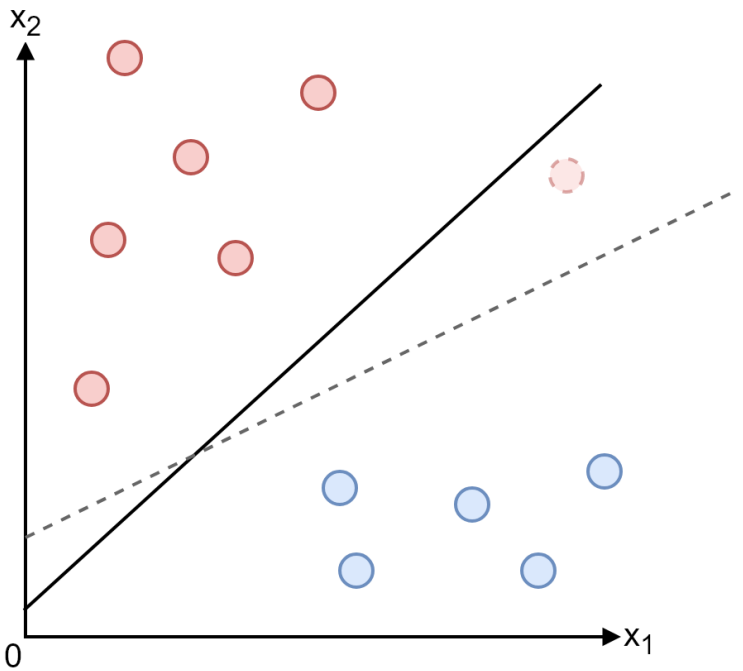

Abbildung 11: Maximum Margin Klassifikation ist nicht sehr robust z.B. bei Hinzunahme von Datenpunkten.

Support Vector Machines

Support Vector Klassifikation

Wir wollen nun die Einschränkung der perfekten Separierbarkeit der Klassen aufgeben und sehen, was wir im allgemeinen Fall tun können. Die Idee ist also, dass wir einigen Datenpunkten erlauben

- innerhalb des Margins oder sogar
- auf der falschen Seite der Hyperebene

zu sein. Letzteres ist auch die Grundvoraussetzung, um überhaupt eine trennende Hyperebene definieren zu können.

Support Vector Machines

Support Vector Klassifikation

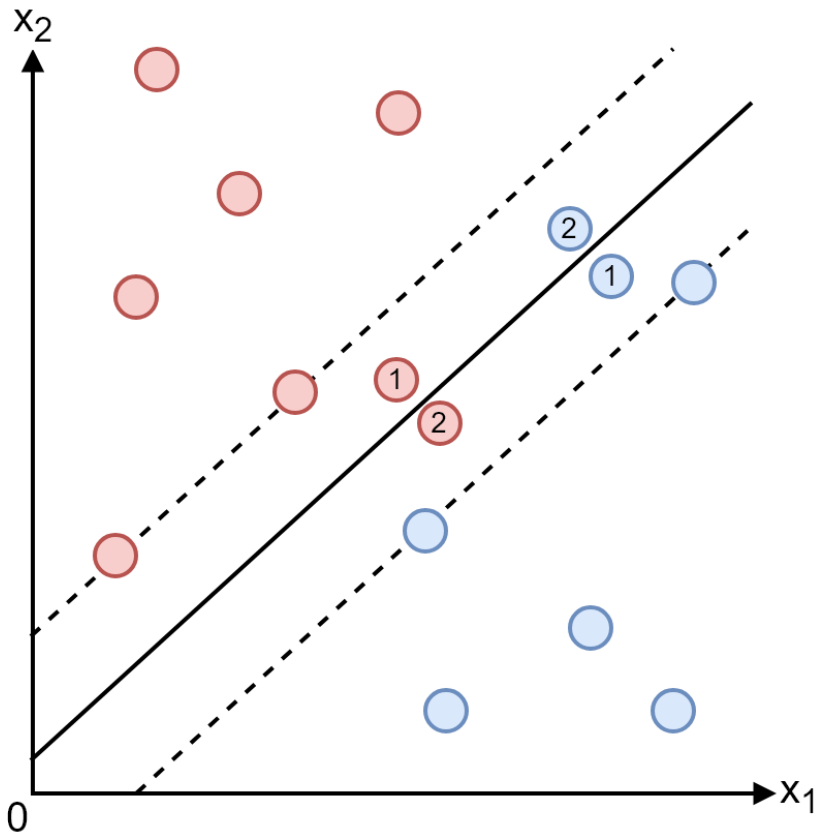

Abbildung 12: Support Vector Klassifikatoren erlauben es manchen Datenpunkten innerhalb des Margins zu sein (1) oder sogar auf der falschen Seite der Hyperebene (2).

Support Vector Machines

Support Vector Klassifikation

Das resultierende Optimierungsproblem für die Support Vector Klassifizierung (SVK) ist nun

$$\arg \max_{\mathbf{w}_0, \mathbf{w}, \epsilon_1, \ldots, \epsilon_n} M$$

unter den Bedingungen

$$\sum_{j=1}^{d} \mathbf{w}_j^2 = 1,$$

$$y^{(i)}(\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq M(1 - \epsilon_i), \quad \epsilon_i \geq 0,$$

für alle $1 \leq i \leq n$ und

$$\sum_{i=1}^{n} \epsilon_i \leq C,$$

wobei $C \in \mathbb{R}_{\geq 0}$ ein Hyperparameter ist. Auch hierfür gibt es effiziente Lösungsverfahren, welche wir uns erst später ansehen.

Support Vector Machines

Support Vector Klassifikation

Interpretation: Die Schlupfvariablen ϵ_i geben Auskunft darüber, wo sich der i-te Datenpunkt befindet.

- $\epsilon_i = 0$: $\mathbf{x}^{(i)}$ befindet sich auf der richtigen Seite der Hyperebene.
- $\epsilon_i \in (0, 1)$: $\mathbf{x}^{(i)}$ verletzt den Mindestabstand zur Hyperebene (Margin), befindet sich jedoch noch auf der richtigen Seite.
- $\epsilon_i > 1$: $\mathbf{x}^{(i)}$ befindet sich auf der falschen Seite der Hyperebene.

Der Hyperparameter C begrenzt hierbei den Grad der Verletzungen. Ist $C = 0$ und damit $\epsilon_1 = \cdots = \epsilon_n = 0$ haben wir einen Maximum Margin Klassifikator vor uns. Generell dürfen nicht mehr als C Datenpunkte auf der falschen Seite liegen.

Support Vector Machines

Support Vector Klassifikation

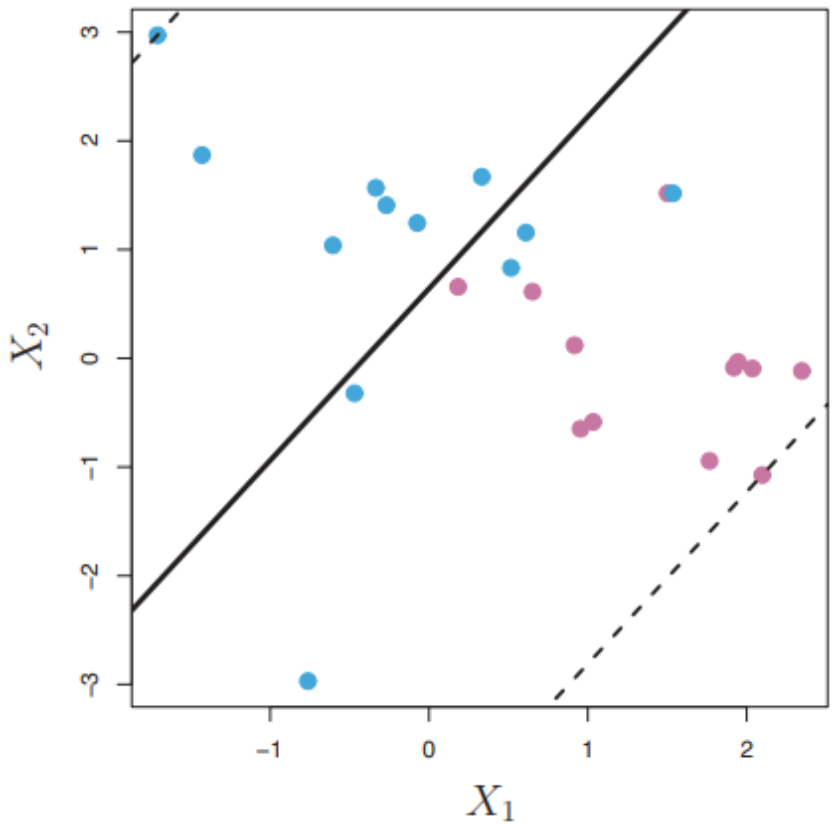

Abbildung 13: SVK mit großem C. Der Klassifikator zeigt eine große Toleranz gegenüber Falsch-Klassifikationen. Auch der Margin ist relativ groß. Abbildung entnommen aus [JWHT14].

Support Vector Machines

Support Vector Klassifikation

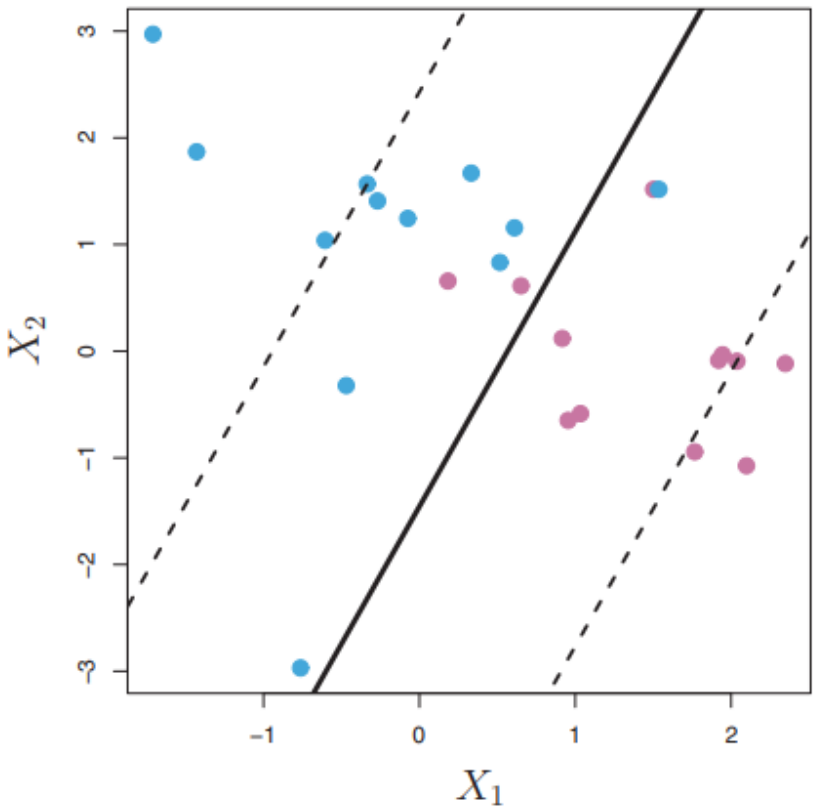

Abbildung 14: Die folgenden Abblidungen zeigen immer kleinere Werte für C. Abbildung entnommen aus [JWHT14].

Support Vector Machines

Support Vector Klassifikation

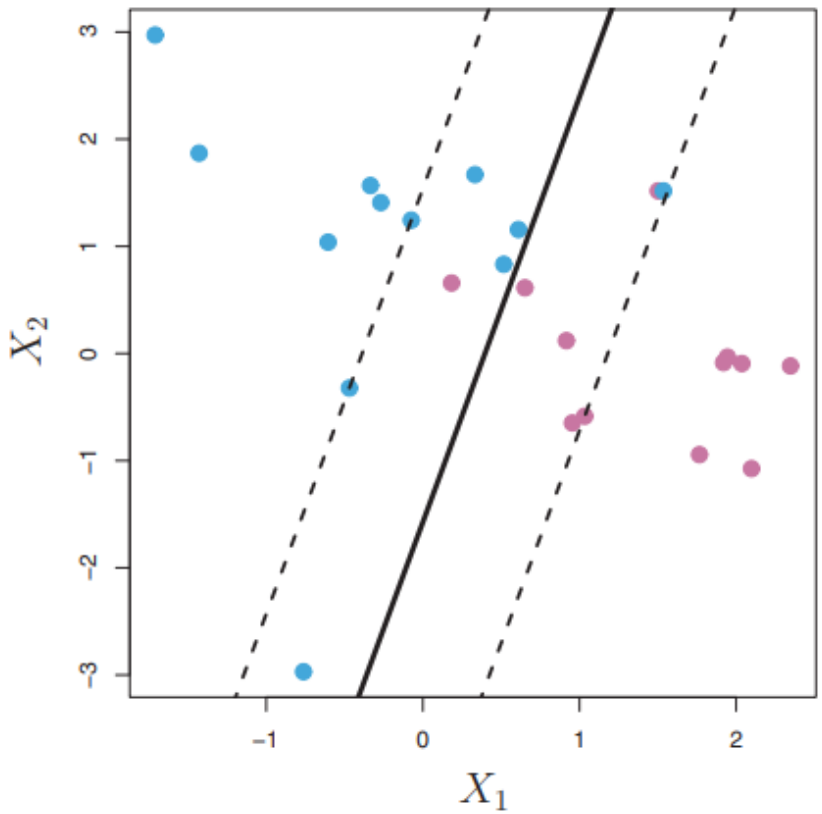

Abbildung 15: Die Anzahl der Falsch-Klassifikationen sinkt mit wachsendem C. Abbildung entnommen aus [JWHT14].

Support Vector Machines

Support Vector Klassifikation

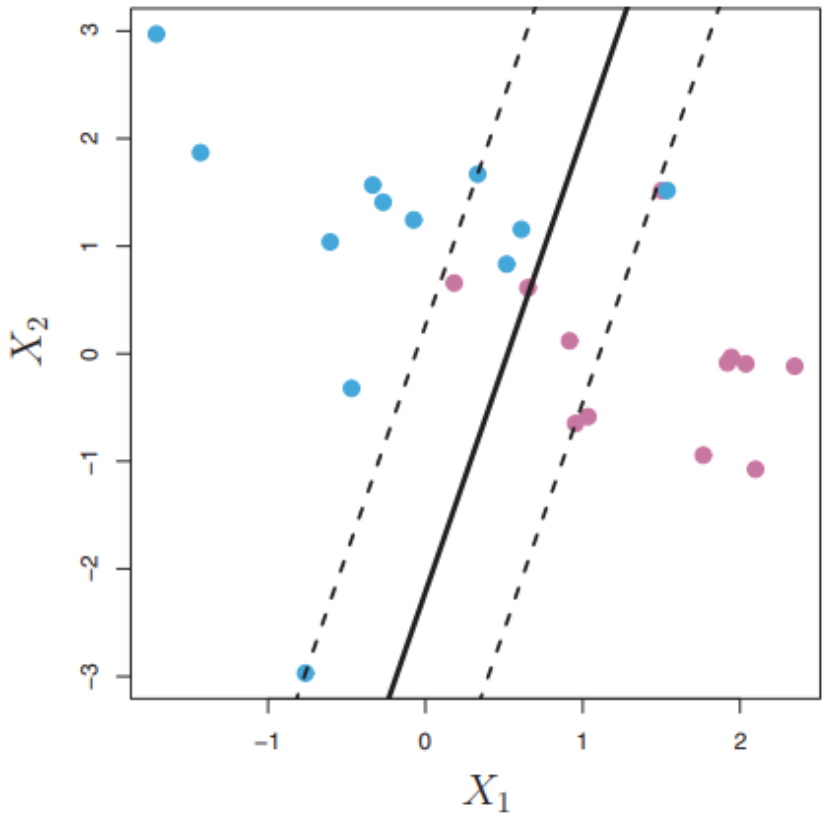

Abbildung 16: Auch der Margin wird immer kleiner. Abbildung entnommen aus [JWHT14].

Support Vector Machines

Support Vector Klassifikation

Auch dieses Optimierungsproblem wird mit Hilfe des KKT gelöst und reduziert sich auf das duale Problem

$$\arg \max_{\lambda_1,\dots,\lambda_n} \mathcal{L}_D(\lambda)$$

mit

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\,T} \mathbf{x}^{(j)}$$

unter den Nebenbedingungen

- $\sum_{i=1}^{n} \lambda_i y^{(i)} = 0$ und
- $0 \leq \lambda_i \leq C$ für alle $i \in \{1, \dots, n\}$.

Support Vector Machines

Support Vector Klassifikation

Das Optimierungsproblem für Support Vector Machines hat die Eigenschaft, dass die Hyperebene nur von Datenpunkten, innerhalb des Margins bzw. auf der falschen Seite bestimmt wird, den Support Vektoren.

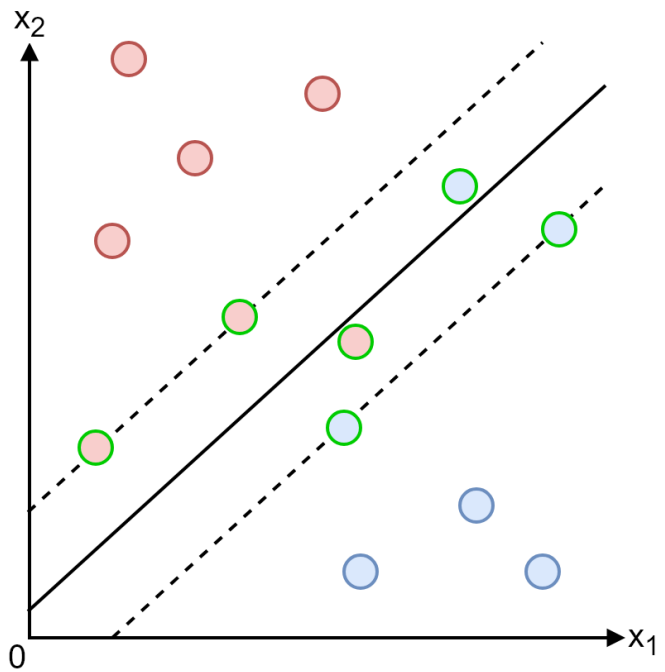

Abbildung 17: SVM mit seinen Support Vektoren (grün).

Support Vector Regression

Support Vector Regression

Support Vector Regression

Wie auch bei der lineare Regression, gehen wir bei der Support Vector Regression davon aus, dass die Funktion f die Form

$$f(\mathbf{x}) = \mathbf{w}_0 + \mathbf{w}^T\mathbf{x}$$

mit $\mathbf{w} \in \mathbb{R}^d$ und $\mathbf{w}_0 \in \mathbb{R}$ hat.

Support Vector Regression

Zusätzlich fordern wir, dass es einen ϵ-großen Margin um f gibt, in welchem möglichst alle Datenpunkte liegen. Da dies nur selten gewährleistet ist, erlauben wir jedoch Überschreitungen $\epsilon + \xi_i$ und Unterschreitungen $\epsilon + \xi_i^*$ des gesamten Margins.

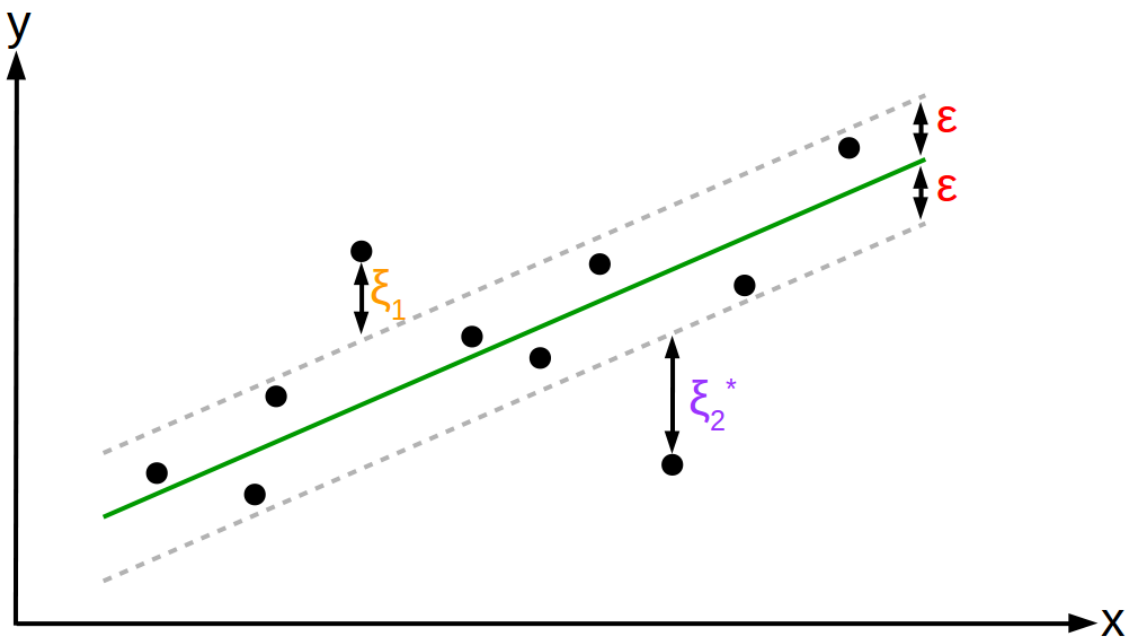

Abbildung 18: Prinzip der Support Vector Regression.

Support Vector Regression

Das resultierende Optimierungsproblem ist für eine feste Wahl von $\epsilon, C \in \mathbb{R}_{\geq}0$ gegeben durch

$$\arg \min_{\mathbf{w}_0, \mathbf{w}, \xi_1, \xi_1^*, \ldots, \xi_n, \xi_n^*} \frac{1}{2}||\mathbf{w}||^2 + C \cdot \sum_{i=1}^{n} (\xi_i + \xi_i^*)$$

unter den Nebenbedingungen

- $y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} - \mathbf{w}_0 \leq \epsilon + \xi_i$
- $\mathbf{w}^T \mathbf{x}^{(i)} + \mathbf{w}_0 - y^{(i)} \leq \epsilon + \xi_i^*$
- $\xi_i, \xi_i^* \geq 0$

für alle $i \in \{1, \ldots, n\}$. Wir bestrafen also nur alle Abweichungen um ξ_i bzw. ξ_i^* außerhalb des ϵ-Margins. Hierbei handelt es sich um ein quadratisches Optimierungsproblem unter Nebenbedingungen (Ungleichungen).

Support Vector Regression

Um das KKT nun auf das Optimierungsproblem der SVR anwenden zu können bilden wir zunächst die Primale Lagrange-Funktion

$$\begin{aligned}\mathcal{L}_P(\mathbf{w}, \mathbf{w}_0, \lambda, \lambda^*) = &-\frac{1}{2}||\mathbf{w}||^2 - C \cdot \sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &- \sum_{i=1}^{n} \lambda_i \left[y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} - \mathbf{w}_0 - \epsilon - \xi_i \right] \\ &- \sum_{i=1}^{n} \lambda_i^* \left[\mathbf{w}^T \mathbf{x}^{(i)} + \mathbf{w}_0 - y^{(i)} - \epsilon - \xi_i^* \right]\end{aligned}$$

Support Vector Regression

Durch Anwendung des KKT erhalten wir

$$\nabla_{\mathbf{w}} \mathcal{L}_P(\mathbf{w}, \mathbf{w}_0, \lambda, \lambda^*) = -\mathbf{w} + \sum_{i=1}^{n} (\lambda_i - \lambda_i^*) \cdot \mathbf{x}_i \stackrel{!}{=} \mathbf{0}$$

und daher

$$\boxed{\mathbf{w} = \sum_{i=1}^{n} (\lambda_i - \lambda_i^*) \cdot \mathbf{x}^{(i)}}.$$

Außerdem erhalten wir noch

$$\frac{\partial \mathcal{L}_P}{\partial \mathbf{w}_0} = \sum_{i=1}^{n} (\lambda_i - \lambda_i^*) \stackrel{!}{=} 0.$$

Support Vector Regression

Wir setzen diese Erkenntnisse nun in $\mathcal{L}_P$ ein und erhalten die Duale Lagrange-Funktion $\mathcal{L}_D(\lambda, \lambda^*) =$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)\mathbf{x}^{(i)\,T}\mathbf{x}^{(j)} + \epsilon\sum_{i=1}^{n}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{n} y^{(i)}(\lambda_i^* - \lambda_i)$$

welche wir unter den Nebenbedingungen (KKT) für alle $1 \leq i \leq n$:

- $\sum_{i=1}^{n}(\lambda_i - \lambda_i^*) = 0$
- $0 \leq \lambda_i, \lambda_i^* \leq C$
- $\xi_i(C - \lambda_i) = 0,\ \xi_i^*(C - \lambda_i^*) = 0$

optimieren müssen.

Support Vector Regression

Die zusätzlichen KKT Bedingungen

- $\lambda_i \left[y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} - \mathbf{w}_0 - \epsilon - \xi_i \right] = 0$
- $\lambda_i^* \left[\mathbf{w}^T \mathbf{x}^{(i)} + \mathbf{w}_0 - y^{(i)} - \epsilon - \xi_i^* \right] = 0$

können zur Berechnung von $\mathbf{w}_0$ benutzt werden.

Zusätzlich wird klar, dass nur ein Teil der Datenpunkte $\mathbf{x}^{(i)} \in \mathcal{S} \subseteq \mathcal{D}$ Lagrange-Multiplikatoren liefert. mit $\lambda_i, \lambda_i^* > 0$, die Support Vektoren. Wie bei der SVK verletzen diese $\mathbf{x}^{(i)}$ den Margin.

Support Vector Regression

Durch $\mathbf{w} = \sum_{i \in \mathcal{S}} (\lambda_i - \lambda_i^*) \cdot \mathbf{x}^{(i)}$ erhalten wir

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{w}_0 = \sum_{i \in \mathcal{S}} (\lambda_i - \lambda_i^*) \cdot \mathbf{x}^{(i)\,T} \mathbf{x} + \mathbf{w}_0$$

Wir verallgemeinern das Skalarprodukt $\mathbf{x}^{(i)\,T}\mathbf{x}$, indem wir einen Kernel k für eine Basistransformation ϕ definieren als

$$k(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^T \phi(\mathbf{v})$$

und erhalten

$$f(\mathbf{x}) = \sum_{i \in \mathcal{S}} (\lambda_i - \lambda_i^*) \cdot k(\mathbf{x}^{(i)}, \mathbf{x}) + \mathbf{w}_0.$$

Die ursprüngliche Formulierung erhalten wir mit $\phi(\mathbf{u}) = \mathbf{u}$ und somit $k(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$.

Support Vector Regression

Wenn wir diese Verallgemeinerung auch in die duale Lagrange-Funktion propagieren, erhalten wir $\mathcal{L}_D(\lambda, \lambda^*) =$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*) k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \epsilon \sum_{i=1}^{n} (\lambda_i + \lambda_i^*) + \sum_{i=1}^{n} y^{(i)} (\lambda_i^* - \lambda_i)$$

Der Vorteil ist, dass die Berechnung von $k(\mathbf{u}, \mathbf{v})$ effizient und ohne die genaue Kenntnis von ϕ erfolgen kann. Das Verfahren wird Kerntrick genannt. Beispiele:

- Lineare Kernel: $k(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$
- Polynomiale Kernel: $k(\mathbf{u}, \mathbf{v}) = (1 + \mathbf{u}^T \mathbf{v})^p$
- Radiale Basisfunktionen: $k(\mathbf{u}, \mathbf{v}) = e^{-\frac{||\mathbf{u} - \mathbf{v}||^2}{2\sigma^2}}$

Support Vector Regression

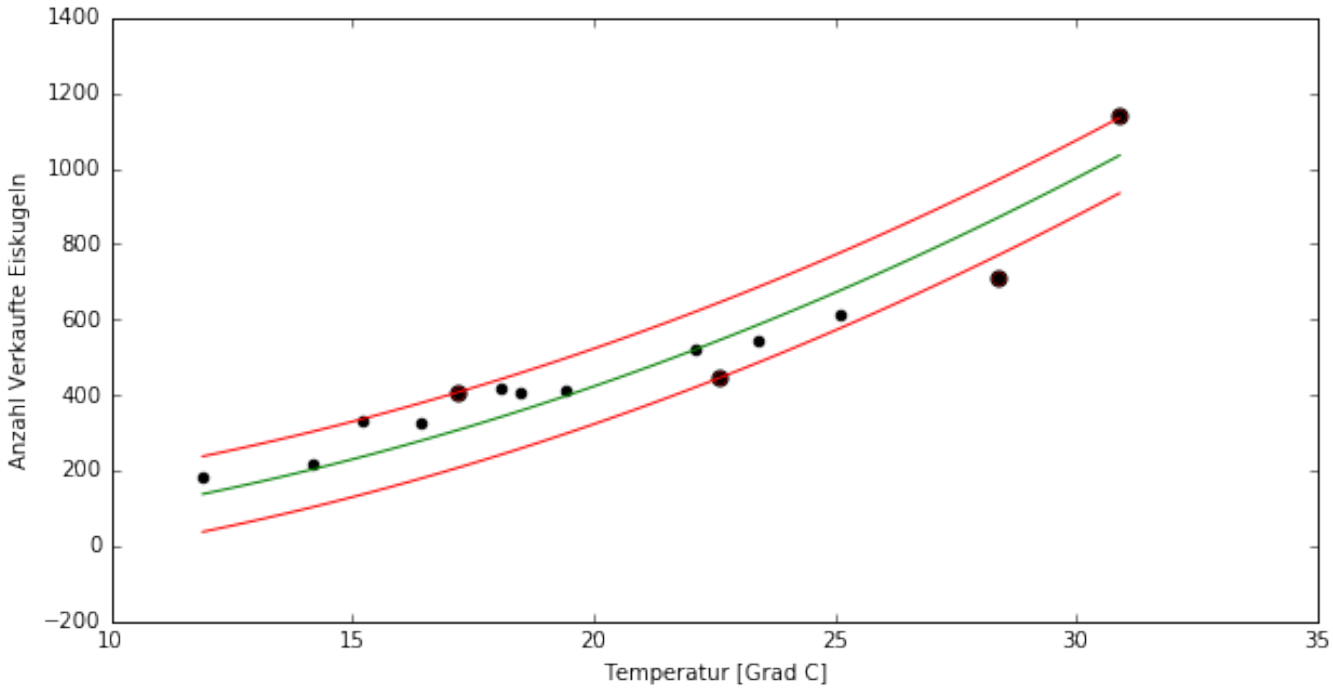

Abbildung 19: Mit dem Kernel $k(u, v) = (1 + u \cdot v)^2$ nehmen wir einen quadratischen Zusammenhang zwischen der Temperatur und der Anzahl der verkauften Kugeln Eis an. Zusätzlich erlauben wir einen Fehlerspielraum von $\epsilon = 100$ und setzen $C = 1$.

Support Vector Regression

Zum Abschluss sei gesagt, dass der Kerntrick natürlich auch bei der Support Vector Klassifikation verwendet wird, so kann auch hier die Duale Lagrange-Funktion

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\,T} \mathbf{x}^{(i)}$$

verallgemeinert werden zu

$$\mathcal{L}_D(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}).$$

Support Vector Regression

References

G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning: With applications in r*, Springer Publishing Company, Incorporated, 2014.