0.1 Hard Margin Classifier

Um das grundlegende Prinzip der SVMs darzustellen gehen wir zuerst von einer Datensituation aus, in der sich zwei gruppen optimal durch eine lineare entscheidungsgrenze trennen lassen. Das endgültige Ziel ist es eine sogenannte Hyperplane zu finden die diese Daten möglichst gut seperiert und als Entscheidungsgrenze funktioniert. Die allgemeine Form einer solchen hyperplain lautet

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n = 0 \tag{1}$$

oder in Vektorschreibweise

$$\overline{\beta} \cdot \overline{x} + \beta_0 = 0 \tag{2}$$

Die geometrische Interpretation des Vektors β und des Skalars β_0 wird in Abbildung 1 im zwei dimensionalen Fall dargestellt.

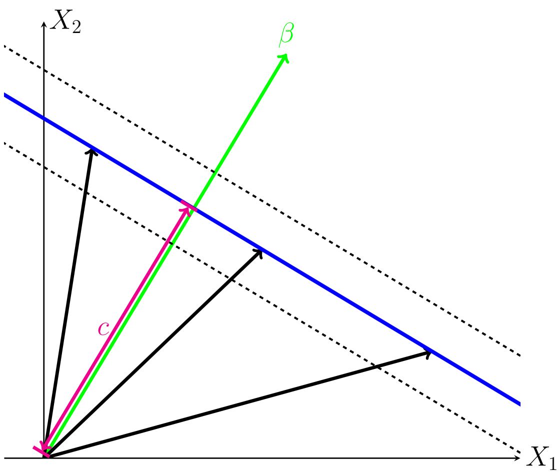

Abbildung 1: Konstruktion der Hyperebene

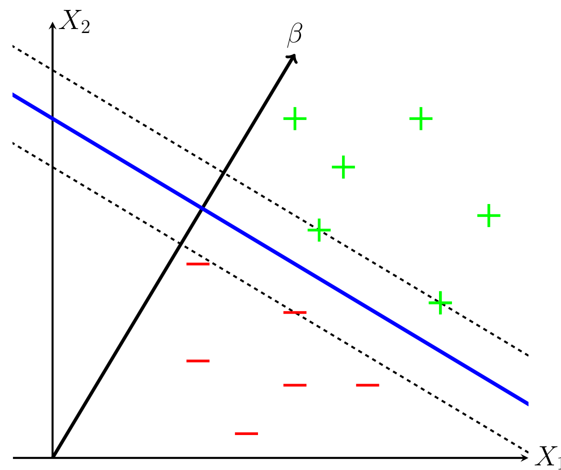

Abbildung 2: Konstruktion der Margins

Die blaue Linie soll die Hyperplain darstellen. Im zweidiemnsionalen handelt es sich hier um eine Linie. Der 2×1 Vektor $\overline{\beta}$ liegt immer senkrecht zur konstruierten Hyperplane. Würde man alle Vektoren, die auf der Hyperplane landen, auf den normierten $\overline{\beta}$-Vektor Projezieren, dann erhält man für alle diese Projektionen den selben Wert c. Also gilt für alle Punkte, die auf der Ebene liegen.

$$\frac{\overline{\beta}}{||\overline{\beta}||} \cdot \overline{x} = c \Leftrightarrow \overline{\beta} \cdot \overline{x} = c \cdot ||\overline{\beta}|| \tag{3}$$

ersetzet man in (3) $c \cdot ||\overline{\beta}||$ mit $-\beta_0$ und zieht dies dann auf die andere Seite, kommen man wieder bei der ursprünglichen Form aus Formel (2) raus (Mavroforakis & Theodoridis, 2006).

Als nächstes stellt sich jetzt die Frage, welches $\overline{\beta}$ und β_0 die optimale Hyperplane darstellen. Betrachtet man die Abbildung 2, dann ist zu erkennen, dass die Datenpunkte durch die blaue Linie getrennt werden. Allerdings könnte man theorethisch undendlich viele andere Hyperplains durch rotation oder verschiebung konstruieren, die trotzdem die Daten in ihren Ausprägungen trennen. Um eine eindeutige Lösung zu finden, wird als nächstes ein bereich um die Hperplane abgesteck. In Abbildung 2 dargestell durch die gestrichelten scharzen linien, welche man als Schranken bezeichnen könnte. In diesem Bereich sollen keine Datenpunkte liegen und die Schranken sollen immer parallel zur Hyperplain seien und den gleichen Abstand zu ihr haben. Außerdem dürfen keine positiven Samples unterhalb der oberen Schranke liegen und keine negativen unterhalb. Das gegenteil gilt dementsprechend für die untere Schranke. Als Definition für die beiden Schranken wird festgelegt

$$\overline{\beta} \cdot \overline{x} + \beta_0 = 1 \tag{4}$$

für die Schranke in richtung der grünen Datenpunkte und

$$\overline{\beta} \cdot \overline{x} + \beta_0 = -1 \tag{5}$$

für die Schranke in Richting der roten Datenpunkte. Aus dieser Beschränkung für die Hyperplane können wir auch ableiten, dass für die positiven Samples $\overline{x}^+$ immer gilt $\overline{\beta} \cdot \overline{x}^+ + \beta_0 \geq 1$ und für negative Samples $\overline{x}^-$ immer gilt $\overline{\beta} \cdot \overline{x}^- + \beta_0 \leq -1$. Durch einführen einer weiteren Variable y, welche die eigenschaft hat, dass die den Wert 1 bei einem positiven und den Wert -1 bei einem negativen Sample annimmt, können diese zwei Beschränkungen zu einer zusammengefasst werden (Cortes & Vapnik, 1995).

$$y_i(\overline{\beta} \cdot \overline{x}_i + \beta_0) \geq 1 \tag{6}$$

Da das Verfahren auch maximum Margin Classifier gennant wird, gilt es jetzt noch eine Definition für den Margin also den Abstand zwischen den zwei Schranken zu finden, der schließlich maximiert werden soll. Damit diese Schranken, maximal weit auseinander liegen, muss es zwangsläufig Datenpunkte geben, die genau auf den Schranken liegen. Diese Datenpunkte haben eine wichtige Rolle für die Konstruktion des Margins. Es sind ausschließlich diese Datenpunkte, die einen Einfluss auf die finalen werte von $\overline{\beta}$ und β_0 haben werden. Sie werden **Support-Vektoren** genannt und geben den SVMs ihren Namen.

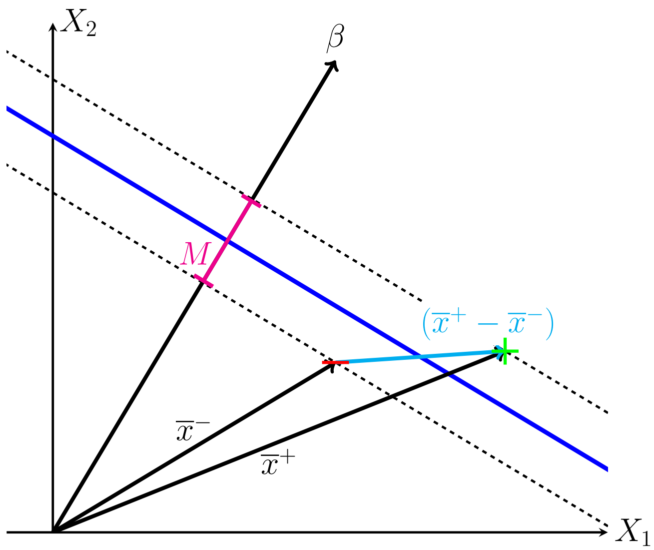

Abbildung 3: Abhängigkeit des Margins von den Support-Vektoren

In Abbildung 3 sind zwei solcher Support-Vektoren zu einem negativen und positiven Sample dargestellt. Der Margin kann dann dargestellt werden als eine Projektion dieser Differenz $(\overline{x}^+ - \overline{x}^+)$ auf den $\overline{\beta}$-Vektor. Damit am Ende die Länge dieses Margins M rauskommt muss $\overline{\beta}$ noch durch seine Länge geteilt werden.

$$M = \frac{\overline{\beta}}{||\overline{\beta}||} \cdot \left(\overline{x}^+ - \overline{x}^+\right) = \frac{\overline{\beta} \cdot \overline{x}^- - \overline{\beta} \cdot \overline{x}^+}{||\overline{\beta}||} \tag{7}$$

Es ist bekannt, dass für postive Supportvektoren gilt $\overline{\beta}\overline{x}^+ + \beta_0 = 1 \Leftrightarrow \overline{\beta}\overline{x}^+ = 1 - \beta_0$ und für negative $\overline{\beta}\overline{x}^- + \beta_0 = -1 \Leftrightarrow \overline{\beta}\overline{x}^- = -1 - \beta_0$. Setzt man dies ein in (7), erhält man als Maximisierungsziel

$$M = \frac{1 - \beta_0 - (-1 - \beta_0)}{||\overline{\beta}||} = \frac{2}{||\overline{\beta}||} \tag{8}$$

Um diesen Maximierungsschritt angenehmer zu gestalten, wird an der Stelle versucht den Ausdruck $\frac{1}{2}||\overline{\beta}||^2 = \frac{1}{2}\overline{\beta}' \cdot \overline{\beta}$ zu minimieren. Was im Endeffekt ebenfalls dazu führt, dass der Ausdruck in (8) maximiert wird.

Dieses Optimierungsproblem mit der Nebenbedingung aus Formel (6) lässt sich am besten über Lagrange-Multiplier lösen

$$\mathcal{L}(\overline{\beta}, \beta_0, \overline{\alpha}) = \frac{1}{2}\overline{\beta}'\overline{\beta} - \sum \alpha_i[y_i(\overline{\beta} \cdot \overline{x_i} + \beta_0) - 1] \tag{9}$$

wobei hier $\overline{\beta}$ und β_0 minimiert und $\overline{\alpha}$ maximiert wird (Vapnik, 2006). Wird dieser Ausdruck partiell abgeleitet und gleich null gesetzt erhält man als zwischen Ergebnis

$$\frac{\partial \mathcal{L}}{\partial \beta} = \beta - \sum \alpha_i y_i \overline{x_i} \stackrel{!}{=} 0 \Rightarrow \beta = \sum \alpha_i y_i \overline{x_i} \tag{10}$$

somit zeigt sich, dass $\overline{\beta}$ als linearkombination der Inputvektoren dargestellt werden kann. Weiterhin gilt für β_0

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = \sum \alpha_i y_i \overline{x_i} \stackrel{!}{=} 0 \tag{11}$$

Setzt man dies in (9) ein erhält man einen neuen Ausdruck, denn es gilt zu minimieren

$$\mathcal{L}(\overline{\alpha}) = -\frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \overline{x}_i \cdot \overline{x}_j + \sum \alpha_i \tag{12}$$

Die Lösung für diesen Ausdruck erfolgt dann über sogenannte „standard non linear optimization algorithms for quadratic forms“ (Boser et al., 1992). Nachem für $\overline{\alpha}$ gelöst wurde, kann dies in (10) eingesetzt werden um das optimale $\overline{\beta}$ zu erhalten. Es kann gezeigt werden, dass die gelösten α_i lediglich für die Supportvektoren werte ungelich Null annehmen. Somit ist der Koeffizientenvektor $\overline{\beta}$ sogar eine Linearkombination von nur den Supportvektoren (Boser et al., 1992). Die letzte unbekannte β_0 kann gelöst werden, indem man mithilfe von einem positiven/negativen Support Vektor (4)/ (5) nach β_0 löst.

Mit den gelösten Werten zur optimalen Hyperplane kann jetzt auch eine Entscheidungsregel für ungelabelte Datenvektoren $\overline{x}_u$ konstruiert werden. Bedenkt man also wenn man einen Vektor der nicht auf der Hypeplane liegt in (3) einsetzt erhält man also $\frac{\overline{\beta}}{||\overline{\beta}||} \cdot \overline{x}_u = c + k$. Wenn k positiv ist, liegt der neue Datenpunkt oberhalb der Hyperplane liegt und somit als positives Sample gewertet wird. Wenn k negativ ist, dann liegt der Datenpunkt unterhalb der Hyperplane und wird als negativ gewertet. Mit der gleichen Umformung wie weiter oben schon beschrieben kommt man zu einer Entscheidungsfunktion $f(\overline{x}_u)$ und einer daraus resultierenden Stufenfunktion $g(\overline{x}_u)$, die dann die Kategorisierung der neuen Beobachtung vornimmt.

$$g(\overline{x}_u) = \begin{cases} \text{positiv} & \text{wenn } f(\overline{x}_u) = \overline{\beta} \cdot \overline{x}_u + \beta_0 > 0 \\ \text{negativ} & \text{wenn } f(\overline{x}_u) = \overline{\beta} \cdot \overline{x}_u + \beta_0 < 0 \end{cases} \tag{13}$$

0.2 Soft Margin Classifier

Dass die Daten sich perfekt linear trennen lassen ist zwar ein gut um die Vorgehensweise zu veranschaulichen, tritt aber in realen Situationen so gut wie nie auf. Falls sich postitive und negative Samples im Raum überlappen, ist die Konstruktion einer Hyperplane wie beim Hard Margin Classifier unmöglich. Man müsste also entweder auf eine nicht lineare Hypeplane ausweichen oder man erweicht die vorgaben für die Konstruktion der Hyperplane. Zweiteres ist genau das, was durch die Soft Margin Classifier erreicht wird. Die Vorgabe für die Konstruktion der Schranken ermöglicht es einzelnen Datenpunkten auf der falschen Seite der Schranke, ja sogar der Entscheidungsgrenzen zu liegen. Dafür wird für die Einschränkungen eine sogennannte Slackvariable ε eingeführt. Wie dieser sich auswirkt ist in Abbildunt 4 aufgeführt. (James et al., 2021). Setzt man diese in diese in (6) lautet die neuen Nebenbedingung

$$y_i(\beta \cdot \overline{x}_i - \beta_0) > 1 - \varepsilon_i \tag{14}$$

Jetzt könnte man versuchen diese neue Nebenbedingung einfach in das zuvor angewandte Optimisierungsverfahren einzufügen. Allerdings besteht hier das Problem, dass ε einfach immer maximal groß gewählt wird und so die Bedingung immer erfüllt wird. Um das Ausmaß der Verletzung der usptünglichen Annahmen zu begrenzen, aber trotzdem noch gewisse Abweichung zuzulassen, wird ein weitere Parameter C eingeführt, als regularisierender Parameter für ε. Leitet daraus zusammen mit der Restriktion $\varepsilon \geq 0$ wieder einen Lagrangefunktion her erhält man

$$\mathcal{L}(\overline{\beta}, \beta_0, \overline{\alpha}, \overline{\varepsilon}, \overline{\lambda}) = \frac{1}{2}\overline{\beta}' \cdot \overline{\beta} + C\sum_{i=1}^{n} \varepsilon_i - \underbrace{\sum \alpha_i [y_i(\overline{\beta} \cdot \overline{x_i} + \beta_0) - 1 + \varepsilon_i]}_{\text{für } y_i(\overline{\beta} \cdot \overline{x}_i - \beta_0) > 1 - \varepsilon_i} - \underbrace{\sum \lambda_i \varepsilon_i}_{\substack{\text{für} \\ \varepsilon_i \geq 0}} \tag{15}$$

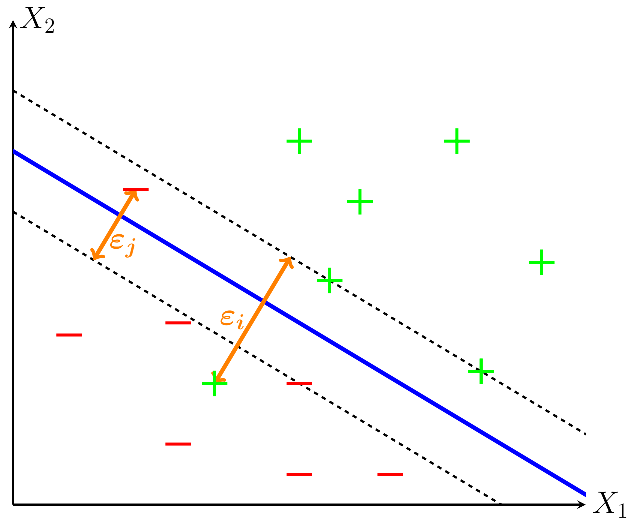

Abbildung 4: Bedeutung der Slack Variable

Wenn dieser Ausdruck wie beim Hardmargin Classifier gelöst wird und die Ergebnisse eingesetzt werden erhält man wieder den Ausruck aus (12) mit der zusätzlichen Einschränkung $0 \leq \alpha_i \leq C$ (Bennett & Campbell, 2000). Dieses Maximierungsproblem wird dann genauso aufgelöst wie bei dem Hard Margin Classifier und die Entscheidungsregel ist ebenfalls gleich. Erst durch den einsatz dieser Technik wurde von der Methode auch wirklich als Support Vector Machine gesprochen (Vapnik, 2006).

0.3 Der Kernel Trick

Auch wenn eine lineare Entscheidungsgrenze Vorteile in Sachen Generalisierbarkeit bietet, ist sie doch nicht für jede Datensituation geeignet.In Abbildung 5 ist es sehr gut zu erkennen, dass in diesem Fall eine lineare Grenze zwischen den Klassen keinen Sinn Ergeben würde und eine elliptische Form wahrscheinlich besser geeignet wäre. Eine Lösung für dieses Problem, wäre den Merkmalsraum zu erweitern. So könnte die angenommene Formel für die lineare Hyperebene in (1) durch polynomterme der Merkmale X_i oder durch Interaktionstherme erweitert werden. Dies führt dazu, dass die Entscheidungsgreze in diesem vergrößerten Merkmalsraum immer noch linear ist, aber die Trennung möglich ist (siehe Abbildung 6). Transformiert man diese dann wieder in den ursprünglichen Merkmalsraum ist die Entscheidungsgrenze dann nicht mehr linear. Allerdings führt diese Herangehensweise zu eine starken Anstieg des Rechenaufwands, da die Möglichkeiten der Merkmalserweiterung endlos sind (James et al., 2021).

Die Lösung für das Problem sind sogenannte Kernel Funktionen. Betrachtet man die entscheidungsfunktion (13) und setzt für $\overline{\beta}$ die Gleichung aus (10) erhält man

$$f(x_u) = \sum \alpha_i y_i x_i \cdot x_u + \beta_0 \tag{16}$$

Es zeigt sich also, dass die Entscheidungsfunktion im Wesentlichen aus einer Linearkombiantion von Punktprodukten aus dem Vektor x_u mit allen Trainingsvektoren x_i ergibt. Diese Punktprodukt kann als Ähnlichkeitsmaß zwischen dem neuen Datenpunkt und dem jeweiligen Trainingsdatenpunkt interpretiert werden. Es ist nun Möglich diese Produkte durch eine Funktion zu ersetzen, welche die Ähnlichkeiten von Datenpunkten anders bewertet. Diese sogenannte Kernel Funktion $K(x_i, x_j)$ ermöglicht es eine flexiblere Entscheidungsgrenze zu implementieren. Der Vorteil ist dabei, dass die Kernel Funktion nur auf alle Punktprodukte angewndet wird und es dabei nicht nötig ist den Merkmalsraum zu Erweitern, was wiederum Rechenzeit spart(James et al., 2021). Die Entscheidungsfunktion wird dann mithilfe dieser Kernelfunktionen berechnet:

$$f(x_u) = \sum \alpha_i y_i K(x_i, x_u) + \beta_0 \tag{17}$$

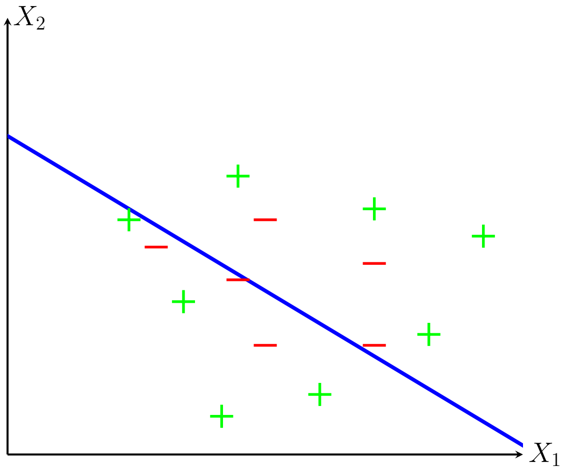

Abbildung 5: nicht linear getrennte Daten

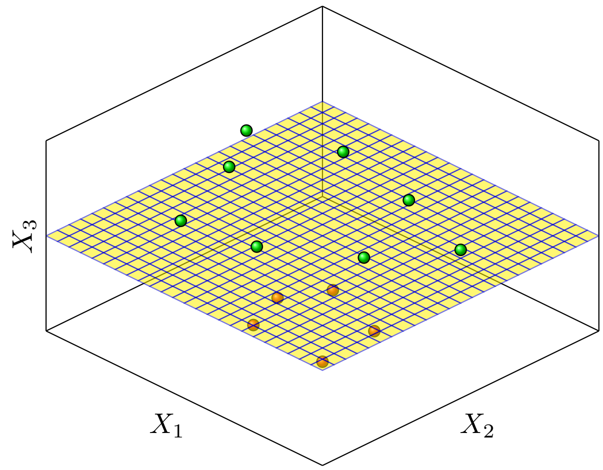

Abbildung 6: Feature Erweiterung

Es gibt eine ganze Reihe an Kernelfunktionen, die bei SVMs Anwedung finden. Die Grundlage ist der lineare Kernel, wobei dieser lediglich das Punktprodukt beschreibt, also praktisch genau das macht, was bei einer linearen Entscheidungsgrenze gemacht wird. Deweiteren gibt es den Polynomial Kernel. Dieser hat die Form

$$K(x_i, x_u) = (1 + x_i \cdot x_u)^d \tag{18}$$

Die Verwendung von diesem Kernel führt dazu, dass die Entscheidungsgrenze sich ähnlich Verhält, wie als würde man zu Beginn eine Merkmalserweiterung mit Polynomen vom Grad d durchführen (siehe Abbildung 7. Eine weitere Kernel Funktion ist der Radial Basis Function Kernel(RBF) mit der Form

$$K(x_i, x_u) = \exp\left(-\gamma ||x_i - x_u||^2\right) \tag{19}$$

Für diesen Kernel wird die quadrierte euklidische Distanz als Ähnlichkeitsmaß verwendet, was dazu führt, dass für diejenigen x_i die näher an x_u liegen, in der Entscheidungsfunktion einen größeren Einfluss haben. Der Parameter γ legt dann fest, wie stark der Einfluss der Distanz sein soll. Die projektion die der Kernel hier macht ist eine, in einen uendlich großen Merkmalsraum . Daher könnte man selbst durch vorgeriges Erweitern des Merkmalsraums nicht das Ergebnis eines RBF Kernel replizieren und dies führt auch zu einer sehr flexiblen Entscheidungsgrenze (siehe Abbildung 8)

Es gibt noch einen Reihe wietere Kernel, die auf unterschiedlichen Ähnlichkeitsmaßen beruhen, aber eher seltener oder nur in speziellen Zusammenhänge angewendet werden. Wichtig ist anzumerken, dass die Verwendung von Kernels zwar die Flexibilität der Entscheidungsgrenze erhöht, damit aber auch die Gefahr von overfitting einhergeht. Zusätzlich werden mit den Kernels auch neue Hyperparameter wie d oder γ eingeführt, die bei der Model selektion ebenfalls beachtet werden müssen.

Literatur

Bennett, K., & Campbell, C. (2000). Support Vector Machines: Hype or Hallelujah? *ACM SIGKDD Explorations Newsletter*, *2*, 1–13. https://doi.org/10.1145/380995.380999

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152. https://doi.org/10.1145/130385.130401

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: With Applications in R*. Springer US. https://doi.org/10.1007/978-1-0716-1418-1

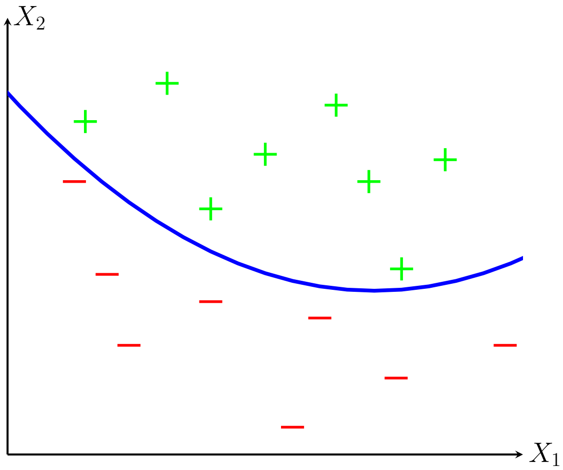

Abbildung 7: Mögliche Entscheidungsgrenze für polynomial Kernel

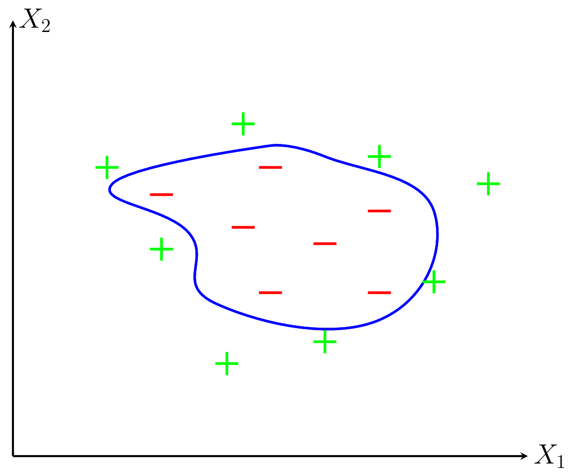

Abbildung 8: Mögliche Entscheidungsgrenze für RBF Kernel

Mavroforakis, M. E., & Theodoridis, S. (2006). A Geometric Approach to Support Vector Machine (SVM) Classification. *IEEE transactions on neural networks*, *17*(3), 671–682. Zugriff 28. August 2024 unter https://ieeexplore.ieee.org/abstract/document/1629090/

Vapnik, V. (2006). *Estimation of Dependences Based on Empirical Data*. Springer. https://doi.org/10.1007/0-387-34239-7