

## 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 separiert und als Entscheidungsgrenze funktioniert. Die allgemeine Form einer solchen Hyperplane lautet

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n = 0 \quad (1)$$

oder in Vektorschreibweise

$$\bar{\beta} \cdot \bar{x} + \beta_0 = 0 \quad (2)$$

Die geometrische Interpretation des Vektors  $\beta$  und des Skalars  $\beta_0$  wird in Abbildung 1 im zwei-dimensionalen Fall dargestellt.

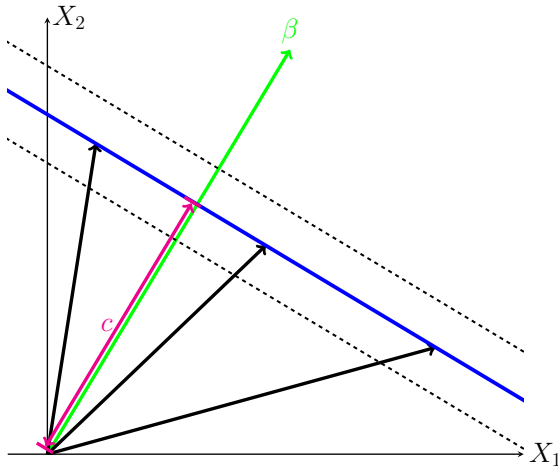


Abbildung 1: Konstruktion der Hyperplane

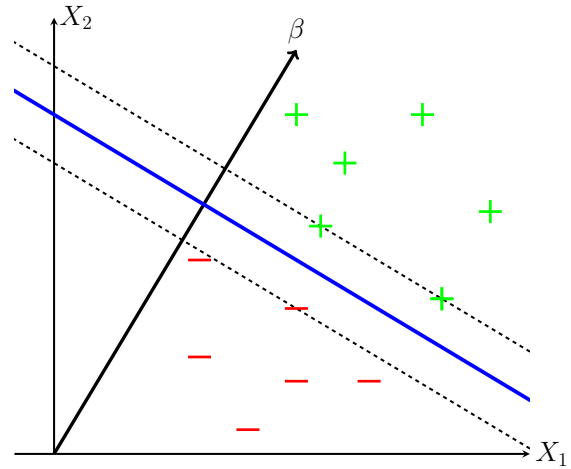


Abbildung 2: Konstruktion der Margins

Die blaue Linie soll die Hyperplane darstellen. Im zweidimensionalen handelt es sich hier um eine Linie. Der  $2 \times 1$  Vektor  $\bar{\beta}$  liegt immer senkrecht zur konstruierten Hyperplane. Würde man alle Vektoren, die auf der Hyperplane landen, auf den  $\bar{\beta}$ -Vektor projizieren, dann hätten alle diese Projektionen die selbe Länge  $c$ . Also gilt für alle Punkte, die auf der Ebene liegen.

$$\frac{\bar{\beta}}{\|\bar{\beta}\|} \cdot \bar{x} = c \Leftrightarrow \bar{\beta} \cdot \bar{x} = c \cdot \|\bar{\beta}\| \quad (3)$$

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

Als nächstes stellt sich jetzt die Frage, welches  $\bar{\beta}$  und  $\beta_0$  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 theoretisch unendlich viele andere Hyperplanes 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 Hyperplane abgesteckt. In Abbildung 2 dargestellt durch die gestrichelten schwarzen Linien, welche man als Schranken bezeichnen könnte. In diesem Bereich sollen keine Datenpunkte liegen und die Schranken sollen immer parallel zur Hyperplane sein 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

$$\bar{\beta} \cdot \bar{x} + \beta_0 = 1 \quad (4)$$

für die Schranke in richtung der grünen Datenpunkte und

$$\bar{\beta} \cdot \bar{x} + \beta_0 = -1 \quad (5)$$

für die Schranke in Richtung der roten Datenpunkte. Aus dieser Beschränkung für die Hyperplane können wir auch ableiten, dass für die positiven Samples  $\bar{x}^+$  immer gilt  $\bar{\beta} \cdot \bar{x}^+ + \beta_0 \geq 1$  und für negative Samples  $\bar{x}^-$  immer gilt  $\bar{\beta} \cdot \bar{x}^- + \beta_0 \leq -1$ . 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

$$y(\bar{\beta} \cdot \bar{x} + \beta_0) \geq 1 \quad (6)$$

Da das Verfahren auch maximum Margin Classifier genannt 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  $\bar{\beta}$  und  $\beta_0$  haben werden. Sie werden **Support-Vektoren** genannt und geben den SVMs ihren Namen.

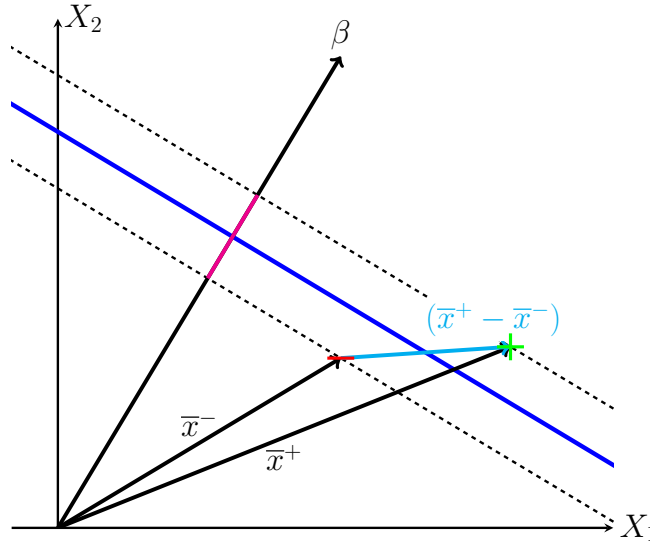


Abbildung 3: Abhängigkeit des Margins von den Support-Vektoren

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  $(\bar{x}^+ - \bar{x}^-)$  auf den  $\bar{\beta}$ -Vektor. Damit am Ende die Länge dieses Margins  $M$  rauskommt muss  $\bar{\beta}$  noch durch seine Länge geteilt werden.

$$M = \frac{\bar{\beta}}{\|\bar{\beta}\|} \cdot (\bar{x}^+ - \bar{x}^-) = \frac{\bar{\beta} \cdot \bar{x}^- - \bar{\beta} \cdot \bar{x}^+}{\|\bar{\beta}\|} \quad (7)$$

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

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

Um diesen Maximierungsschritt angenehmer zu gestalten, wird an der Stelle versucht den Ausdruck  $\frac{1}{2}\|\bar{\beta}\|^2 = \frac{1}{2}\bar{\beta}' \cdot \bar{\beta}$  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

$$\mathcal{L}(\bar{\beta}, \beta_0, \bar{\alpha}) = \frac{1}{2}\bar{\beta}'\bar{\beta} - \sum \alpha_i [y_i(\bar{\beta} \cdot \bar{x}_i + \beta_0) - 1] \quad (9)$$

Wird dieser Ausdruck partiell abgeleitet und gleich null gesetzt erhält man als zwischen ergebnis

$$\frac{\partial \mathcal{L}}{\partial \beta} = \beta - \sum \alpha_i y_i \bar{x}_i \stackrel{!}{=} 0 \Rightarrow \beta = \sum \alpha_i y_i \bar{x}_i \quad (10)$$

somit zeigt sich, dass  $\bar{\beta}$  als linearkombination der Inputvektoren dargestellt werden kann. Weiterhin gilt für  $\beta_0$

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = \sum \alpha_i y_i \bar{x}_i \stackrel{!}{=} 0 \quad (11)$$

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

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

Die Lösung für diesen Ausdruck erfolgt dann über sogenannte „standard non linear optimization algorithms for quadratic forms“ (Boser et al., 1992). Nachdem für  $\bar{\alpha}$  gelöst wurde, kann dies in (10) eingesetzt werden um das optimale  $\bar{\beta}$  zu erhalten. Es kann gezeigt werden, dass die gelösten  $\alpha_i$  lediglich für die Supportvektoren werte ungleich Null annehmen. Somit ist der Koeffizientenvektor  $\bar{\beta}$  sogar eine Linearkombination von nur den Supportvektoren (Boser et al., 1992). Die letzte unbekannte  $\beta_0$  kann gelöst werden, indem man mithilfe von einem positiven/negativen Support Vektor (4)/ (5) nach  $\beta_0$  löst.

Mit den gelösten Werten zur optimalen Hyperplane kann jetzt auch eine Entscheidungsregel für ungelabelte Datenvektoren  $\bar{x}_u$  konstruiert werden. Bedenkt man also wenn man einen Vektor der nicht auf der Hyperplane liegt in (3) einsetzt erhält man also  $\frac{\bar{\beta}}{\|\bar{\beta}\|} \cdot \bar{x}_u = c + k$ . 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 folgender Entscheidungsregel

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

## 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 positive 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 Hyperplane 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 sogenannte Slackvariable  $\varepsilon$  eingeführt (James et al., 2021). Setzt man diese in diese in (6) lautet die neuen Nebenbedingung

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

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

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

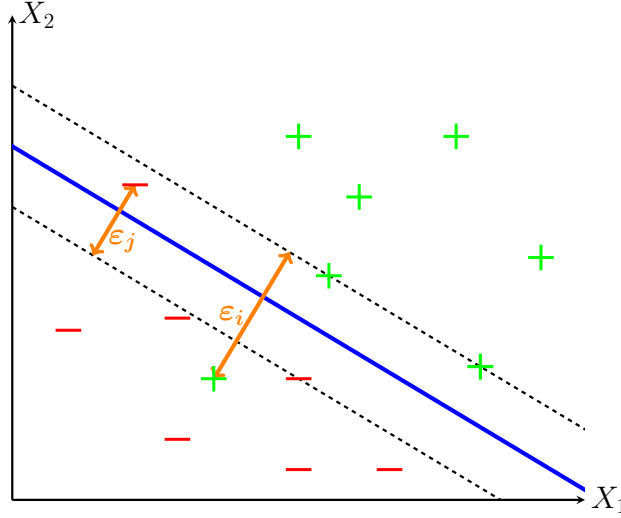


Abbildung 4: Funktion der Slack Variable

Wenn dieser Ausdruck wie beim Hardmargin Classifier gelöst wird und die Ergebnisse eingesetzt werden erhält man wieder den Ausdruck aus (12) mit der zusätzlichen Einschränkung  $0 \leq \alpha_i \leq C$ . Dieses Maximierungsproblem wird dann genauso aufgelöst wie bei dem Hard Margin Classifier und die Entscheidungsregel ist ebenfalls gleich.

### 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 Interaktionsterme erweitert werden. Dies führt dazu, dass die Entscheidungsgrenze 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 einem starken Anstieg des Rechenaufwands, da die Möglichkeiten der Merkmalerweiterung endlos sind (James et al., 2021).

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

$$f(x_u) = \sum \alpha_i y_i x_i \cdot x_u + \beta_0 \quad (16)$$

Es zeigt sich also, dass die Entscheidungsfunktion im Wesentlichen aus einer Linearkombination von Punktprodukten aus dem Vektor  $x_u$  mit allen Trainingsvektoren  $x_i$  ergibt. Dieses Punktprodukt kann als Ähnlichkeitsmaß zwischen dem neuen Datenpunkt und dem jeweiligen Trainingsdatenpunkt interpretiert werden. Es ist nun möglich auf diese Produkte eine Funktion anzuwenden, 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 angewendet 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:

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

Es gibt eine ganze Reihe an Kernelfunktionen, die bei SVMs Anwendung finden. Die Grundlage ist der lineare Kernel, wobei dieser lediglich das Punktprodukt beschreibt, also praktisch genau das macht, was bei einer

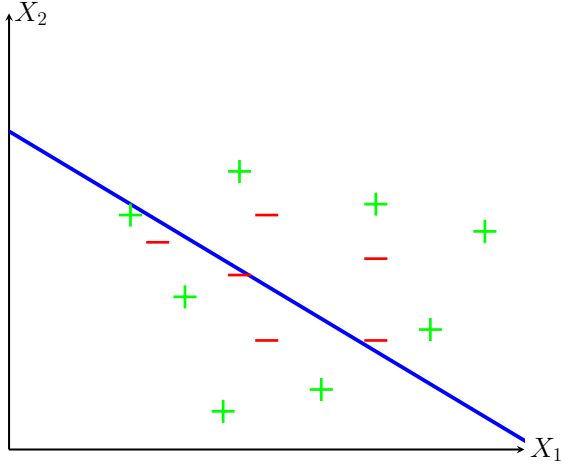


Abbildung 5: nicht linear getrennte Daten

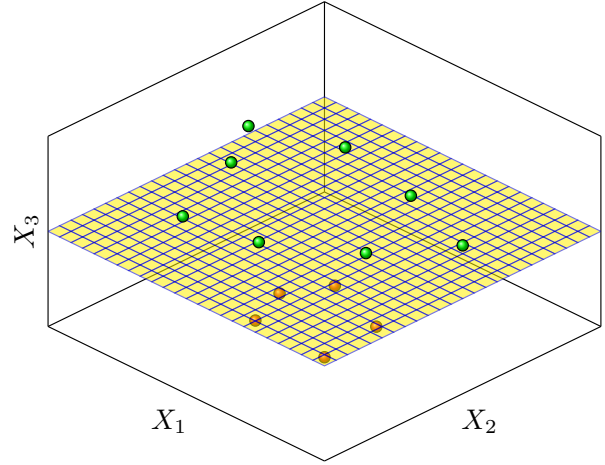


Abbildung 6: Feature Erweiterung

linearen Entscheidungsgrenze gemacht wird. Deweiteren gibt es den Polynomial Kernel. Dieser hat die Form

$$f(x_u) = \sum \alpha_i y_i K(x_i, x_u) + \beta_0 \quad (18)$$

wobei  $d$  ein Integer  $> 0$  ist. 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

$$K(x_i, x_u) = \exp(-\gamma ||x_i - x_u||^2) \quad (19)$$

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  $\gamma$  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)

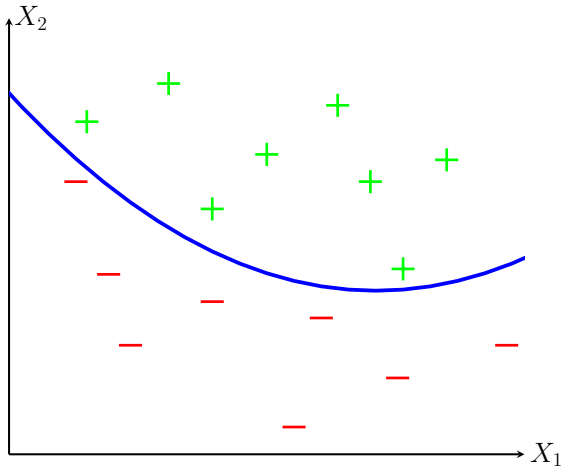


Abbildung 7: Mögliche Entscheidungsgrenze für polynomial Kernel

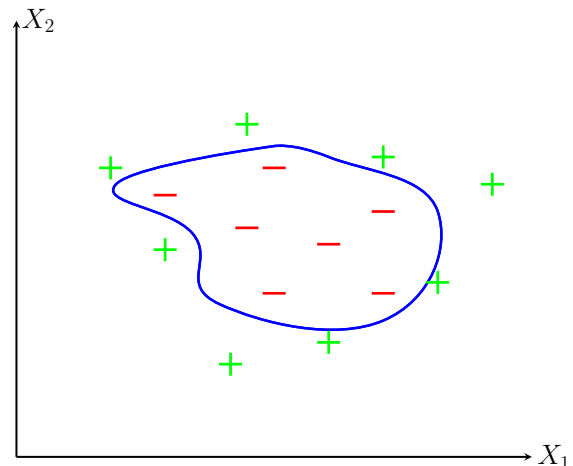


Abbildung 8: Mögliche Entscheidungsgrenze für RBF Kernel

Es gibt noch eine Reihe weiterer Kernel, die auf unterschiedlichen Ähnlichkeitsmaßen beruhen, aber eher seltener oder nur in speziellen Zusammenhängen 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  $\gamma$  eingeführt, die bei der Model selection ebenfalls beachtet werden müssen.

overfitting Problem

## Literatur

- 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>
- 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>