

Support Vector Machines haben ein hohes Ansehen unter den Machine Learning Algorithmen, da sie einige Vorteile mit sich bringen. Aufgrund der Idee einer Soft Margin und des “Kernel Tricks” ist die Methode sehr flexibel und kann für spezielle Anwendungsbereiche angepasst werden (Bennett & Campbell, 2000). Dazu sind die Ergebnisse stabil und reproduzierbar, was sie von anderen Methoden wie beispielsweise Neural Networks abhebt. Auch die Anwendung ist vergleichsweise einfach, da es eine überschaubare Anzahl an Parametern gibt (wie beispielsweise bei der *SVM* mit radialem Kern nur der gamma- und cost-Parameter festzulegen ist).

Durch die Möglichkeit der Nutzung verschiedener Kerne sind *SVM* überaus vielseitig. Die Auswahl des Kerns ermöglicht es äußerst flexible Entscheidungsgrenzen zu formen (Kuhn & Johnson, 2013). Dadurch können *SVM* an verschiedene Datensituationen angepasst werden.

Ein weiterer Vorteil ist, dass die Methode weitgehend robust gegenüber overfitting ist (Kuhn & Johnson, 2013). Dafür verantwortlich ist der Cost-Parameter, anhand dessen der Fit an die Daten kontrolliert werden kann. Jedoch birgt dies auch Probleme (Erläuterungen im folgenden Abschnitt).

Diese Vorteile resultieren in einer allgemein häufigen Nutzung von *SVM* in der Wissenschaft, wobei sie schon oft bewiesen haben, dass sie für verschiedenste Aufgaben gut funktionieren (Kuhn & Johnson, 2013).

Trotz der vielfachen Nutzung von *SVM*, bringen sie auch Nachteile mit sich. Das wohl größte Problem liegt in der Modellselektion (Bennett & Campbell, 2000). Wie bereits im vorherigen Abschnitt erwähnt, ist die Auswahl der Parameter von hoher Bedeutung bei der Performance und dem Fit an die Daten. So kontrollieren die kernspezifischen Parameter und der Cost-Parameter einerseits die Komplexität und andererseits den Fit an die Daten (Kuhn & Johnson, 2013). Dabei kann die Wahl der Parameter sowohl zu einem underfit als auch zu einem overfit führen. Jedoch haben nicht nur die Parameter einen Einfluss auf die Performance sondern bereits die Wahl des Kerns kann entscheidend sein (Burgess, 1998). Je nach Datensituation können *SVM* mit verschiedenen Kernen äußerst unterschiedliche Ergebnisse liefern. Dies zeigt die Sensitivität der Methode gegenüber der Wahl des Kerns und der Parameterabstimmung.

Ein weiterer Nachteil ist, dass die Methode weniger intuitiv und aufwendiger anzuwenden ist als andere Algorithmen (Bennett & Campbell, 2000). So ist es zum Beispiel schwer Informationen aus Support Vektoren zu ziehen und es gibt keine Koeffizienten die interpretiert werden können.

Zuletzt ist zu erwähnen, dass die Methode bei einer hohen Anzahl an Beobachtungen besonders rechenintensiv ist. So konnte beispielsweise gezeigt werden, dass insbesondere die *SVM* mit polynomialem und radialem Kern eine hohe Rechenzeit aufweisen (Scholz & Wimmer, 2021). Dabei konnten andere Methoden wie die *logistische Regression* oder *k-nearest Neighbour* deutlich besser abschneiden. Dies liegt daran, dass die Lösung des *SVM*-Optimierungsproblems, die Behebung eines quadratischen Programmierungsproblems erfordert. Da die Anzahl der zu optimierenden Parameter mit der Anzahl der Daten quadratisch zunimmt, führt dies zu einer hohen Rechenkomplexität (Kecman, 2005).

## 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>
- Burgess, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2), 121–167. <https://doi.org/10.1023/A:1009715923555>
- Kecman, V. (2005, April). Support Vector Machines – An Introduction. In J. Kacprzyk & L. Wang (Hrsg.), *Support Vector Machines: Theory and Applications* (S. 1–47, Bd. 177). Springer Berlin Heidelberg. [https://doi.org/10.1007/10984697\\_1](https://doi.org/10.1007/10984697_1)
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer New York. <https://doi.org/10.1007/978-1-4614-6849-3>
- Scholz, M., & Wimmer, T. (2021). A Comparison of Classification Methods across Different Data Complexity Scenarios and Datasets. *Expert Systems with Applications*, 168, 114217. <https://doi.org/10.1016/j.eswa.2020.114217>