

0.1 Pros

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’s überaus vielseitig. Die Auswahl des Kerns ermöglicht es äußerst flexible Entscheidungsgrenzen zu formen (Kuhn & Johnson, 2013). Dadurch können SVM’s 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’s in der Wissenschaft. Sie haben folglich bewiesen, dass sie für verschiedenste Aufgaben gut funktionieren (Kuhn & Johnson, 2013).

0.2 Cons

Trotz der vielfachen Nutzung von SVM’s, 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’s 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).

0.3 Hypothesen

Die Performance von Support Vector Machines wird anhand verschiedener Datenszenarien untersucht. Dabei werden Logistic Regression und k-nearest Neighbour als Vergleichsalgorithmen hinzugezogen. Genauer gesagt, wird in neun verschiedene Szenarien unterschieden, welche sich durch zwei Unterteilungen ergeben: die Form der Entscheidungsgrenze sowie das Verhältnis zwischen der Anzahl an Dimensionen (p) und der Anzahl an Beobachtungen (n). Dabei werden ausschließlich binäre Klassifikationen untersucht. Es ergibt sich folgende Aufteilung:

Im Folgenden werden Studien hinzugezogen, um eine Einschätzung der Performance in den verschiedenen Szenarien vorzunehmen und Hypothesen abzuleiten.

Es konnte gezeigt werden, dass in einem Szenario, indem erheblich mehr Beobachtungen als Dimensionen und eine lineare Entscheidungsgrenze vorliegen (S1), deutliche Unterschiede zwischen SVM, k-NN und LogR bei

	linear	polynomial	radial
$p \ll n$	S1	S2	S3
$p = n$	S4	S5	S6
$p \gg n$	S7	S8	S9

Tabelle 1: Datensituationen

der Diskriminationsfähigkeit auftreten (Entezari-Maleki et al., 2009). k-NN und lineare SVM zeigen AUC-Werte nahe 1 auf, was für eine nahezu perfekte Differenzierung der Klassen spricht. LogR hingegen hat einen Wert knapp über 0.5, was nur etwas besser als eine Zufallsauswahl ist. Darüber hinaus ist festzustellen, dass die Unterschiede deutlicher werden, je höher die Anzahl an Beobachtungen ist.

Für den Fall einer radialen Entscheidungsgrenze (S3) sind die Ergebnisse ähnlich. So erreicht in diesem Beispiel eine SVM mit radialem Kernel im Vergleich zu einer LogR eine um 34% höhere Genauigkeit (Fávero et al., 2022).

Liegt ein Szenario vor, indem die Anzahl der Dimensionen erheblich größer ist, als die Anzahl der Beobachtungen, mit einer linearen Entscheidungsgrenze (S7), sind die Ergebnisse differenzierter zu betrachten. So schneidet die SVM mit polynomialem Kern am besten unter den genannten Algorithmen ab, jedoch die lineare SVM am schlechtesten (als Kriterium wurde die mittlere Performance über 100 Datensätze evaluiert) (Scholz & Wimmer, 2021). Während k-NN auch in diesem Szenario eine gute Performance hat, schneiden LogR und SVM mit radialem Kern mittelmäßig ab.

Basierend auf den Ergebnissen der genannten Studien können Schlussfolgerungen gezogen werden. Es ist anzunehmen, dass in niedrigdimensionalen Szenarien k-NN und SVM's besser performen als LogR. Jedoch ist zu vermuten, dass die Wahl des Kerns bei SVM's einen großen Einfluss auf die Performance hat.

In hochdimensionalen Szenarien zeigt vermutlich die SVM mit polynomialem oder radialem Kern eine gute Performance, unabhängig von der Form der Entscheidungsgrenze, während die lineare SVM voraussichtlich weniger gut abschneiden wird. Es scheint so, dass auch k-NN und LogR in hochdimensionalen Szenarien zumindest mittelmäßig abschneiden. Hier ist jedoch zu beachten, dass nur eine lineare Entscheidungsgrenze betrachtet wurde und in den Szenarien S8 und S9 andere Ergebnisse möglich sind.

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>
- Burges, 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>
- Entezari-Maleki, R., Rezaei, A., & Minaei-Bidgoli, B. (2009). Comparison of Classification Methods Based on the Type of Attributes and Sample Size. *Journal of Convergence Information Technology*, 4(3), 94–102. <https://doi.org/10.4156/jcit.vol4.issue3.14>
- Fávero, L. P., Belfiore, P., Santos, H. P., dos Santos, M., de Araújo Costa, I. P., & Junior, W. T. (2022). Classification Performance Evaluation from Multilevel Logistic and Support Vector Machine Algorithms through Simulated Data in Python. *Procedia Computer Science*, 214, 511–519. <https://doi.org/10.1016/j.procs.2022.11.206>
- 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
- 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>