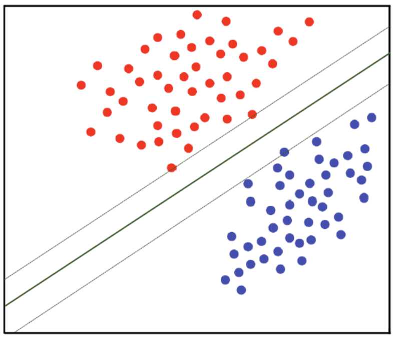
TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
Die Ressourcenuniversität. Seit 1765.

Support Vector Machine



Autorenname: Sophie Merkisch

Studiengang: Business Analytics

Modul: Algorithmik Sommer2021

Betreuer: Prof. Dr. Ingo Schiermeyer

Definition

Heutzutage ist die Support Vector Machine (SVM) eine Standardmethoden zur Klassifikation. Die SVM gehört zu den Machine Learning-Algorithmen:

Lernstil	Modelltyp	Algorithmus	Lernaufgabe
Überwachtes Lernen	Kernmethoden	Support Vector Machine	Klassifikation
		Support Vector Regression	Regression

Tabelle 1: Einordnung der SVM in Machine Learning-Algorithmen

Die SVM konstruiert eine trennende Ebene zwischen den Daten. Die zu analysierenden Objekte sind mittels ihrer Merkmalswerte in einem Merkmalsraum dargestellt. Nahbeieinanderliegende Objekte sind ähnlich. Daten auf der einen Seite der Ebene werden einer Klasse zugeordnet und die Daten auf der anderen Seite der Ebene werden einer anderen Klasse zugeordnet. Die trennende Ebene befindet sich an der Grenze zwischen zwei Klassen. Sie wird auch als Entscheidungsoberfläche bezeichnet.

Support Vector Machines wurden 1992 von Boser, Guyon und Vapnik erarbeitet. 1995 haben Cortes und Vapnik die SVM als eine gängige statistische Methode etabliert.

Mithilfe der SVM lassen sich flexible Trennlinien als Grenzen bestimmen. Diese Grenzen werden mithilfe von mathematischen Verfahren aus der Optimierung ermittelt. Die Klassenhäufigkeit und die Klassenwahrscheinlichkeit sind bei der SVM nicht relevant. Zunächst ist das Verfahren der SVM nur für zwei Klassen geeignet. Eine Übertragung auf mehrere Klassen ist hierbei möglich.

In dem Raum der Eingabedaten lernt eine SVM für die Klassifikation eine Entscheidungsebene, die den maximalen Abstand zu den am nächsten liegenden Datenpunkten aufweist. Die SVM ermittelt mit sogenannten Stützvektoren eine Trennlinie.

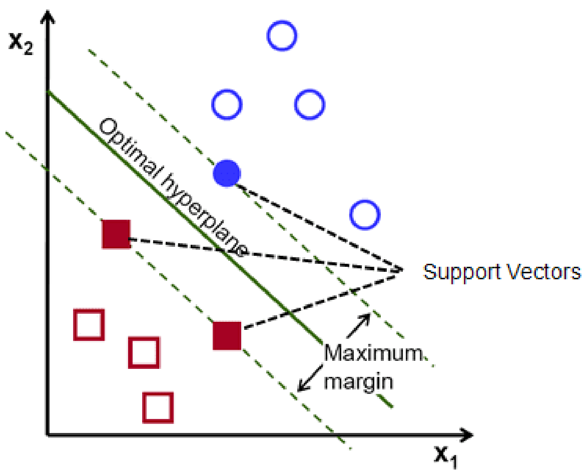


Abbildung 1: Support Vector Machine

Ziel der SVM ist es, n-dimensionale Datenpunkte durch eine optimale Hyperebene in zunächst zwei Klassen zu teilen. Aus bekannten Trainingsdaten, von denen die Klassenzuordnung auch bekannt ist, wird eine optimale Hyperebene konstruiert, sodass die Punkte verschiedener Klassen auf unterschiedlichen Seiten der Ebene liegen. Um die Klasse eines neuen Punktes zu bestimmen, muss lediglich ermittelt werden, auf welcher Seite der Ebene er liegt. Die Hyperebene wird dabei so konstruiert, dass der kleinste Abstand eines Trainingspunktes zu dieser Ebene maximal wird.

BASIS KONZEPT	
Klassentrennung	• Gesucht wird die optimal trennende Hyperebene zwischen den beiden Klassen mittels der Maximierung des Abstands zwischen den nächstgelegenen Punkten der Klassen • Punkte, die auf den Grenzen liegen, werden als Stützvektoren bezeichnet • Mitte der Spanne ist die optimal trennende Hyperebene
Überlappende Klassen	• Datenpunkte auf der "falschen" Seite des Diskriminanzrandes werden nach unten gewichtet, um ihren Einfluss zu reduzieren (soft margin)
Nichtlinearität	• Lässt sich kein lineare Trennlinie finden, werden die Datenpunkte in einen höherdimensionalen Raum projiziert, sodass die Datenpunkte effektiv linear trennbar sind • Projektion erfolgt mittels der Kernel-Techniken
Problemlösung	• Gesamte Aufgabe kann als quadratisches Optimierungsproblem bezeichnet werden, welches mit bekannten Techniken gelöst werden kann
ERWEITERUNGEN	
ν-Klassifikation	• dieses Modell erlaubt eine bessere Steuerung der Anzahl der Support-Vektoren, indem ein zusätzlicher Parameter ν angegeben wird, der den Anteil der Support-Vektoren approximiert
Ein-Klassen-Klassifikation	• dieses Modell versucht, den Support einer Verteilung zu finden und erlaubt so die Erkennung von Ausreißern/Neuheiten
Regression	• hier liegen die Datenpunkte zwischen den beiden Rändern der Marge, die unter geeigneten Bedingungen maximiert wird, um die Einbeziehung von Ausreißern zu vermeiden
ν-Regression	• mit analogen Modifikationen des Regressionsmodells wie im Klassifikationsfall

Prinzip der Trennung

In einem zweidimensionalen Raum wird die Trennlinie der SVM mit Hilfe von Funktionen mit zwei Variablen $f : \mathbf{R}^2 \rightarrow \mathbf{R}$ konstruiert. Die Trennlinie T_c ist die Höhenlinie der Funktion *f.* Die Trennlinie ist demnach eine Linie, die die Punkte mit demselben Funktionswert *c* verbindet. Die Trennlinie lässt sich schreiben als: $T_c = \{(x1, x2) \in \mathbf{R}^2 : f(x1, x2) = c\}$

Für eine gute Trennung der Klassen wird eine Funktion gesucht, für die es eine Höhenlinie T_c gibt, die die Punkte gut voneinander trennt. Das Gütekriterium ist hierbei noch zu definieren. Bildlich ist klar, dass eine möglichst geringe Anzahl von Punkten „auf der falschen Seite der Trennlinie" liegen sollten. Normalerweise wird die Funktion so bestimmt, dass die gesuchte Höhenlinie die Punkte mit dem Funktionswert 0 verbindet: $T_0 = \{(x_1, x_2) \in \mathbf{R}^2 f(x_1, x_2) = 0\}$

Im simpelsten Fall sind die Objekte durch eine Gerade trennbar. Dann spricht man von linearer Trennbarkeit. Wenn Daten nicht linear trennbar sind, können die Klassen mittels anderer Funktionstypen erzeugt werden. Eine Möglichkeit ist die trennende Höhenlinie, die durch Polynome generiert werden kann. Des Weiteren lässt sich mithilfe der Radial-Basis-Funktion eine sehr flexible Funktion *f* finden, für den Fall, dass die Punkte in den Klassen sehr unregelmäßig verteilt sind.

Linear trennbare Daten

Um die Trenngerade soll ein möglichst breiter datenfreier Streifen zwischen den Klassen erzielt werden. Die Breite der sogenannten Margin gibt an, wie zuverlässig die Trennlinie ist, dass die Klassen klar voneinander abgegrenzt werden. Je breiter desto zuverlässiger. Die Suche nach der geeigneten Trennlinie ist somit eine Optimierungsaufgabe. Bei dieser soll die Breite des Trennstreifens maximiert werden.

Die Datenpunkte, die sich exakt auf dem Rand des Streifens befinden, werden Supportvektoren oder Stützvektoren bezeichnet, da sie den Randbereich des Streifens stützen.

Die Bedingung, dass die Supportvektoren den Funktionswert ±1 haben sollen, wird bei der Suche nach der optimalen Trennlinie zur Nebenbedingung.

Optimierungsaufgaben mit Nebenbedingungen lassen sich mit dem Lagrange-Ansatz lösen.

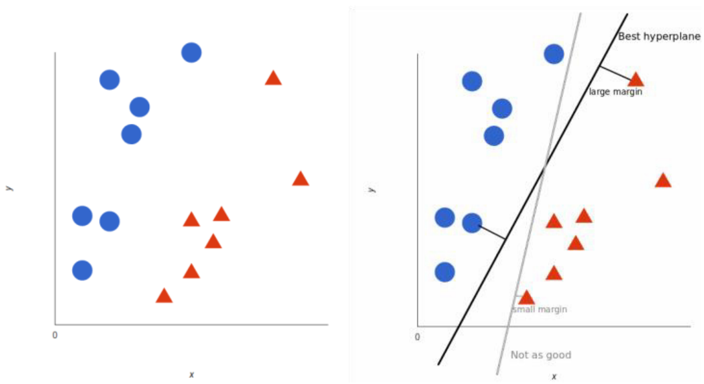


Abbildung 2: lineare Trennbarkeit

Lineare Trennbarkeit mit Ausnahmepunkten

Es kann vorkommen, dass die Punkte der Lerndaten so stark streuen, dass maximal ein sehr schmaler datenfreier Streifen gefunden werden kann. Fällt auf, dass die Punkte grundsätzlich durch einen breiteren Trennbereich getrennt werden könnten, der jedoch nicht komplett datenfrei wäre und einige Datenpunkte enthält, ist es meistens sinnvoller den breiteren Trennbereich zu wählen. Ein breiterer Trennbereich hat eine bessere Prognosegüte zur Folge im Vergleich zu einer schmalen perfekten Trennlinie. Mit einem breiteren Trennbereich kann verhindert werden, dass der Trennbereich sich nicht übermäßig an die Lerndaten anpasst und Overfitting zu keiner Problematik wird. Ein weiterer Nachteil eines schmaleren Trennbereichs ist, dass minimale Änderungen der Daten zu einer ganz anderen Trennlinie führen können.

Als Nebenbedingung der Optimierung kann mittels des sogenannten Regulierungsparameters C die Suche nach der optimalen Trennfunktion f gesteuert werden. Diese Schlupfvariable erlaubt, dass sich einzelne Datenpunkte auch dichter oder gar auf der falschen Seite der Trennebene befinden können. C gibt an, wie stark der Trennbereich überschritten werden darf. Je größer C ist, desto stärker werden Überschreitungen nicht toleriert und desto schmaler wird somit der Trennbereich. Daraus resultiert, dass weniger Trainingsdaten zu den Support-Vektoren gehören.

Nicht linear trennbare Daten

Nicht linear trennbare Daten lassen sich klassifizieren, indem der Raum auf einer höheren Dimension abgebildet wird. Die Berechnung dieser Transformation ist jedoch in vielen Fällen sehr rechenintensiv. Es kann dazu kommen, dass es zu einer Vielzahl von neuen Dimensionen kommt und jede ggf. eine aufwendige Berechnung erfordert. Diese aufwendige Rechnung müsste für jeden Vektor des Datensatzes erfolgen.

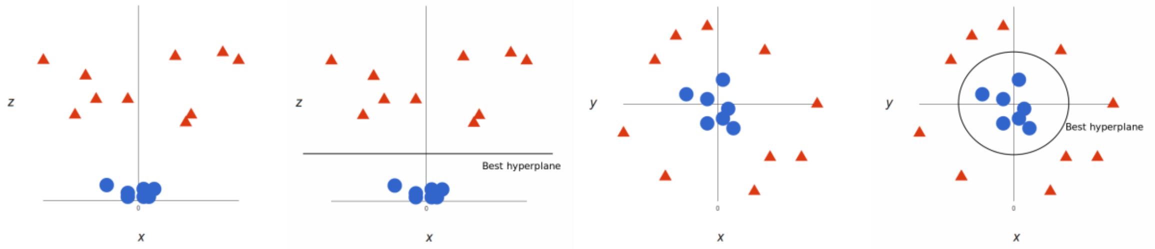


Abbildung 3: nicht lineare Trennbarkeit – Transformationsfunktion

In diesem Beispiel ist sofort deutlich, dass es keine lineare Trennlinie gibt. Dennoch ist zu erkennen, dass die Vektoren deutlich voneinander zu trennen sind und es scheint simple, diese zu trennen. In der nächsten Grafik wird eine dritte Dimension hinzugefügt und die Vektoren in diesem Beispiel mittels der Gleichung für Kreise berechnet. Daraufhin lässt sich eine Hyperebene parallel zur x-Achse bestimmen. Zuletzt werden die Daten wieder in einen zweidimensionalen Raum transformiert.

Kerneltrick

Eine komplizierte Transformationsfunktion kann zu einem großen Rechenaufwand führen. Es gibt Funktionen, die dasselbe Ergebnis liefern wie das Skalarprodukt von zwei transformierten Vektoren. Der große Vorteil ist die relativ einfache Berechnung. Diese Funktionen werden Kernfunktionen genannt. Die Kernelfunktion bildet Vektoren auf höheren Dimensionen ab und vermeidet direkte Vektoren im hochdimensionalen Raum. Mithilfe des Kerneltricks lassen sich auch nichtlineare Entscheidungsebenen recheneffizient lernen. Dies erfolgt durch das Bestimmen der Ebene in einem implizierten höherdimensionalen Raum.

Die Berechnung der Skalarprodukte der Vektoren werden als Kernel-Funktion bezeichnet. Dieser Weg erleichtert die Berechnung der Transformationsfunktionen deutlich.

Dies ist als Kernel-Trick bekannt, der den Merkmalsraum vergrößert, um eine nicht-lineare Grenze zwischen den Klassen zu berücksichtigen.

Gängige Arten von Kernel, die zur Trennung nicht-linearer Daten verwendet werden:

- lineare Kernel
- Polynom-Kernel
- Radial-Basis-Kernel

All diese Kernel Varianten transformieren die Daten so, dass eine lineare Hyperebene entsteht und die Daten somit klassifiziert werden können. Die Radial-Basis-Funktion findet am häufigsten Anwendung.

Auch bei den nichtlinearen Trennfunktionen ist es entscheidend, dass es einen möglichst breiten datenfreien Streifen gibt. Ausnahmepunkte können hierbei zugelassen sein. Wie bei den linearen Trennfunktionen wird auch bei den nichtlinearen Trennfunktionen mit dem Strafterm C die Anzahl und Lage der Ausnahmepunkte gesteuert. Auf diesem Wege werden auch hier die Anzahl der Stützvektoren und die Breite des Trennbereichs bestimmt. Hat C einen größeren Wert, werden weniger Ausnahmepunkte toleriert, der Trennbereich verkleinert sich und daraus resultieren weniger Stützvektoren.

Vorteile	Nachteile
• Praktikabel, kostengünstig und effiziente Methode zur Klassifizierung von Objekten • SVM muss lediglich mit den Support-Vektoren verglichen werden → schnelles System bei der Klassifizierung • Unterstützung von linearer und nicht-linearer Klassifizierung (Kernelfunktion) • Hohe Dimensionalität • Speichereffizienz • Vielseitigkeit • Es leidet nicht unter dem Problem der Multikollinearität • Geringer Implementierungsaufwand der SVM • Geeignet für: Handschriften- und Bilderkennung, Genomanalyse und Astrophysik, Text- oder Bildklassifikation	• Trainingsphase braucht viel Zeit, da es aufwendig ist, die Hyperebene zu finden • braucht viel Zeit bei großen Datensätzen • Nicht-probabilistisch • gibt keine direkten Wahrscheinlichkeitsschätzungen zurück • lineare Kernel ist der logistischen Regression im Falle von linear trennbaren Daten sehr ähnlich • Auswahl der Kernel-Parameter

Literaturverzeichnis

Eitrich, Tatjana (2003) Support-Vektor-Maschinen und ihre Anwendung auf Datensätze aus der Forschung.

Florian Dumpert/Katja von Eschwege/Martin Beck (2016) Einsatz von Support Vector Machines bei der Sektorzuordnung von Unternehmen.

Fraunhofer (2018) Maschinelles Lernen – Ergebnisbericht.

Hamel, Lutz (Hrsg.) (2009) Knowledge Discovery with Support Vector Machines, Hoboken, NJ, USA.

Hude, Marlis von der (2020) Predictive Analytics und Data Mining, Wiesbaden.

Hüftle, Mike (2006) Methoden der Klassifikation.

Kersting, Kristian/Lampert, Christoph/Rothkopf, Constantin (2019) Wie Maschinen lernen.

Meyer, David (2021) Support Vector Machines. The Interface to libsvm in package e1071.

Pönisch, Jens (2019) Grundlagen von Support Vector Machines.

Vojdani, Nina/Erichsen, Björn (2020) Anwendungspotenziale von maschinellem Lernen in der Produktion und Logistik.

Welsch, Andreas/Eitle, Verena/Buxmann, Peter (2018) Maschinelles Lernen, in: HMD Praxis der Wirtschaftsinformatik, 55. Jg., Nr. 2, S. 366–382.