Diese Arbeit wurde vorgelegt am Lehrstuhl für Mathematik (MathCCES)

# Vergleich zweier Methoden zur Bildklassifizierung auf Basis maschineller Lernalgorithmen und ihre Anwendbarkeit in der Vermittlung mathematischer Modellierung

## Comparison of two methods for image classification based on machine learning algorithms and their applicability for conveying mathematical modeling

Masterarbeit Mathematik

Februar 2018


| | |
|---|---|
| Vorgelegt von<br>Presented by | Sarah Schönbrodt<br>[redacted] |
| Erstprüfer<br>First examiner | Prof. Dr. Martin Frank<br>Lehrstuhl für Mathematik (MathCCES)<br>RWTH Aachen University |
| Zweitprüfer<br>Second examiner | Prof. Dr. Sebastian Walcher<br>Lehrstuhl A für Mathematik<br>RWTH Aachen University |
| Koreferent<br>Co-supervisor | Thomas Camminady<br>Lehrstuhl für Mathematik (MathCCES)<br>RWTH Aachen University |

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# 1 Motivation

Das Erkennen von Mustern und Regelmäßigkeiten in Daten ist ein Problem, mit dem sich Wissenschaftlerinnen und Wissenschaftler schon seit Jahrzehnten auseinandersetzen. So spielte die Aufdeckung von Regelmäßigkeiten in den Daten von Atomspektren eine fundamentale Rolle in der Entwicklung der Quantenphysik in den 1920er Jahren (vgl. Bishop, 2006, S. 1). Ein Ziel der Mustererkennung ist die *Klassifizierung*, d. h. die Zuordnung von gegebenen Daten zu verschiedenen Klassen. Die Lösung von solchen *Klassifizierungsproblemen* verlangt heutzutage in den meisten Fällen die Bewältigung riesiger Datenmengen. Um diese tatsächlich verarbeiten zu können, sind effiziente Computeralgorithmen gefragt, welche die Muster in den gegebenen Daten möglichst *automatisiert* erkennen und darauf basierend die Zuordnung zu bestimmten Klassen bzw. Gruppen *lernen*. Ein Forschungsgebiet, das sich unter anderem mit verschiedenen Methoden zur automatisierten Bearbeitung von Klassifizierungsproblemen beschäftigt, ist das sogenannte *maschinelle Lernen*.

Klassifizierungsprobleme treten in zahlreichen Bereichen von Wissenschaft und Forschung auf: Sei es in der Medizin, bei der intelligente Klassifikatoren zur frühen Diagnose und Vorhersage von Krankheiten genutzt werden, zur Erkennung von Kreditkartenbetrug in Echtzeit oder im Bereich des Marketings zur kundenspezifischen Werbung (vgl. Mitchell, 1997, S. 3). Ein fundamentales Problem stellt die Klassifizierung von Bildern dar, die bereits in verschiedenen Kontexten Anwendung findet: Unter anderem bei der automatischen Identifikation von Personen auf Bildern bei Facebook, bei autonom fahrenden Fahrzeugen, die zwischen verschiedenen Verkehrsschildern unterscheiden und bei der Erfassung von handgeschriebenen Ziffern durch Textverarbeitungsprogramme (vgl. Burges, 1998, S. 121). Je nach konkreter Problemstellung und gegebenem Datentyp bieten sich verschiedene Methoden aus dem Bereich des maschinellen Lernens in unterschiedlichem Maße an. Die Bandbreite existierender und bereits vielfach genutzter Methoden reicht von künstlichen neuronalen Netzen, über Naive-Bayes-Klassifikatoren, die k-Nearest-Neighbour-Methode bis hin zu der Support Vektor Maschine und einer Methode die auf der Singulärwertzerlegung basiert. Die beiden letztgenannten Methoden werden in der vorliegenden Arbeit untersucht (vgl. Hastie et al., 2001).

Aufgrund der großen Relevanz der maschinellen Lernmethoden für verschiedene Problemstellungen, die sich hinter diversen Anwendungen aus dem Alltag verbergen, stellen diese Methoden eine vielversprechende Möglichkeit für die authentische mathematische Modellierung mit Schülerinnen und Schülern[1] dar. Wenngleich die verschiedenen Methoden zur Lösung von Klassifizierungsproblemen höchst komplex sind, lassen sie sich bei genauerer Analyse der mathematischen Hintergründe vielfach auf elementarmathematische und z. T. anschauliche Grundkonzepte reduzieren: Sei es die Berech-

[1] Nachfolgend werden Schülerinnen und Schüler unter der Bezeichnung Schüler zusammengefasst. Analog wird mit den Personengruppen Lehrerinnen und Lehrer, Betreuerinnen und Betreuer u.a. verfahren.

nung von Abständen von Punkten zu Hyperebenen in Vektorräumen im Falle der Support Vektor Maschine oder die Bestimmung der besten Approximation von Vektoren in Unterräumen und damit die Berechnung der orthogonalen Projektion im Falle der Methode über die Singulärwertzerlegung. Sowohl aufgrund der elementar-mathematischen Zugänglichkeit der Modelle, als auch aufgrund deren großer Anwendungsbreite für verschiedene lebensnahe Anwendungen liegt es nahe, dass Schülern tatsächlich ein verständiger Zugang zu diesen maschinellen Lernmethoden möglich sein kann. Insbesondere mit Blick auf die Zielsetzung die Vermittlung mathematischer Modellierung motivierend und interessant zu gestalten, bietet sich die Wahl der automatisierten Klassifizierung als authentische und alltagsrelevante Fragestellungen an. Durch die Entwicklung eines Lernmoduls zu der aktuellen Problemstellung der Bildklassifizierung kann eine derartige Vermittlung mathematischer Modellierung möglich gemacht werden, die den Schülern gleichzeitig einen Einblick in aktuelle Forschung gewährt und ein grundlegendes Verständnis für die Arbeitsweise des maschinellen Lernens schafft.

Im folgenden Kapitel wird zunächst ein Überblick über die verschiedenen Lernaufgaben des maschinellen Lernens gegeben sowie die grundlegende Vorgehensweise der maschinellen Lernmethoden bei der Bearbeitung jener Lernaufgaben dargestellt. Anschließend werden in Kapitel 3 die mathematischen Hintergründe der beiden ausgewählten Lernmethoden, die Support Vektor Maschine und die Methode über die Singulärwertzerlegung, dargelegt. Dabei wird stets Bezug zu dem Problem der Bildklassifizierung genommen. Neben der mathematischen Darstellung und der exemplarischen Anwendung beider Methoden kann die vorliegende Arbeit als Grundlage für die Erstellung eines Lernmoduls zur mathematischen Modellierung mit Schülern der Sekundarstufe II betrachtet werden. Deswegen wird bei den Erläuterungen der mathematischen Hintergründe versucht, diese möglichst anschaulich darzustellen, um eine leichtere Zugänglichkeit zu ermöglichen. Außerdem wird an verschiedenen Stellen auf eine detaillierte Beweisführung bzw. auf maximale Allgemeinheit verzichtet.
Kapitel 4 stellt den experimentellen Teil dieser Arbeit dar. Dazu wurden die beiden Klassifizierungmethoden auf den MNIST Datensatz,[2] einer freien Bilddatenbank bestehend aus Bildern mit handgeschriebenen Ziffern, angewendet. Der Klassifizierungserfolg beider Methoden wurde bestimmt und der Einfluss verschiedener Parameter der mathematischen Modelle auf den Klassifizierungserfolg untersucht. Zudem wurden beide Methoden zur Klassifizierung von menschlichen Gesichtern angewendet. Dazu wurde ein eigener Datensatz generiert. Der Fokus des experimentellen Teils dieser Arbeit liegt nicht auf dem Erzielen neuer Highscores hinsichtlich des Klassifizierungserfolgs, sondern auf der Veranschaulichung und Diskussion der zuvor dargelegten mathematischen Hintergründe der beiden Lernmethoden. In Kapitel 5, dem didaktischen Teil dieser Arbeit, werden Ideen für die Gestaltung eines Lernmoduls im Rahmen eines mathematischen Modellierungsworkshops für Schüler dargelegt. Abschließend wird in Kapitel 6 ein Ausblick auf weiterführende Experimente sowie methodisch-didaktische Umsetzungsmöglichkeiten gegeben.

[2] yann.lecun.com/exdb/mnist/, Stand: 10.12.2017

# 2 Klassifizierung auf Basis maschineller Lernalgorithmen

Eine effiziente Herangehensweise an Klassifizierungsprobleme stellt die Verwendung von Methoden aus dem Bereich des maschinellen Lernens dar. Das Konzept des maschinellen Lernens kann in Anlehnung an Mitchell (1997) wie folgt definiert werden:

> *„A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E"(Mitchell, 1997, S. 2).*

Am Beispiel der automatisierten Handschrifterkennung, einem Klassifizierungsproblem, lassen sich die Bezeichnungen aus Mitchells Definition wie folgt übertragen:
Die Lernaufgabe T bezeichne die Erkennung und Klassifizierung von handgeschriebenen Wörtern auf Bildern, das Erfolgsmaß P ist gegeben durch den Prozentsatz der richtig klassifizierten Wörter und die Trainingserfahrung E stellt einen Datensatz bestehend aus Bildern der handgeschriebenen Wörter sowie deren korrekte Klassenzuordnung dar. Allgemeiner formuliert ist das Ziel von Methoden des maschinellen Lernens die Entwicklung von Algorithmen, die aus gegebenen Daten Strukturen extrahieren und die in der Lage sind auf Basis des Gelernten Vorhersagen für neue Daten zu treffen (vgl. Mitchell, 1997, S. 4).

Bei den maschinellen Lernmethoden wird zwischen *überwachtem*, *unüberwachtem* und *bestärkendem Lernen* unterschieden. Eine vielfach auftretende Lernaufgabe ist dem überwachten Lernen (engl. supervised learning) zuzuordnen. Das generelle Vorgehen von überwachten maschinellen Lernmethoden soll nachfolgend am Beispiel der Bildklassifizierung betrachtet werden (vgl. Bishop, 2006, S. 3).
Angenommen es sei ein Datensatz bestehend aus $N$ schwarz-weiß Bildern gegeben, auf denen entweder ein Stoppschild, bezeichnet als Klasse 1, oder ein Vorfahrtschild, bezeichnet als Klasse 2, abgebildet ist. Angenommen diese Bilder haben eine Größe von $16 \times 16$ Pixel, dann könnte der Datensatz aus $N$ Vektoren $\mathbf{x}_1, \ldots, \mathbf{x}_N$[3] der Länge 256 bestehen, deren Einträge die jeweiligen Grauwerte der Pixel, spaltenweise untereinandergeschrieben, repräsentieren. Weiterhin ist die Zuordnung jedes dieser Bilder zu den beiden Klassen durch sog. *Labels* $t_1, \ldots, t_N$ gegeben, die im beschriebenen Beispiel entweder den Wert 1 oder 2 annehmen. Ausgehend von diesem sogenannten *Trainingsdatensatz* soll in der *Trainingsphase* ein Modell, gegeben durch eine Funktion $f(\mathbf{x})$, gelernt werden, welches den Trainingsbildern $\mathbf{x}_n$ die zugehörigen Klassenlabels $t_n$ zuordnet. Das gelernte Modell wird dann in der nachfolgenden *Testphase* auf neue unbekannte Daten, die *Testdaten*, angewendet, um die Generalisierung zu evaluieren.
Im Falle der Bildklassifizierung ist vor der eigentlichen Trainingsphase ggf. eine *Vorverarbeitung* der Eingangsdaten sinnvoll, bei der die Bilder beispielsweise durch Drehung

[3] Im Folgenden werden Variablen die Vektoren repräsentieren dick gedruckt, um sie leichter von Variablen die einzelne Werte bzw. Komponenten repräsentieren zu unterscheiden.

oder Skalierung transformiert werden (vgl. Bishop, 2006, S. 2). Eine mögliche Vorverarbeitung stellt die Standardisierung der Eingangsdaten dar. Dabei werden die einzelnen Dimensionen[4] der Daten so standardisiert, dass der Mittelwert 0 und die Varianz 1 beträgt (vgl. García et al., 2016, S. 11). Die im folgenden Kapitel beschriebenen Methoden, die Support Vektor Maschine und die Singulärwertzerlegung, sind Methoden, bei denen Wissen über die Klassenzuordnungen der Trainingsdaten verwendet wird, um ein entsprechendes Modell zu lernen. Somit fallen diese Methoden in die Kategorie des *überwachten Lernens*. Die allgemeine Funktionsweise von überwachten maschinellen Lernmethoden ist in Abbildung 1 dargestellt.

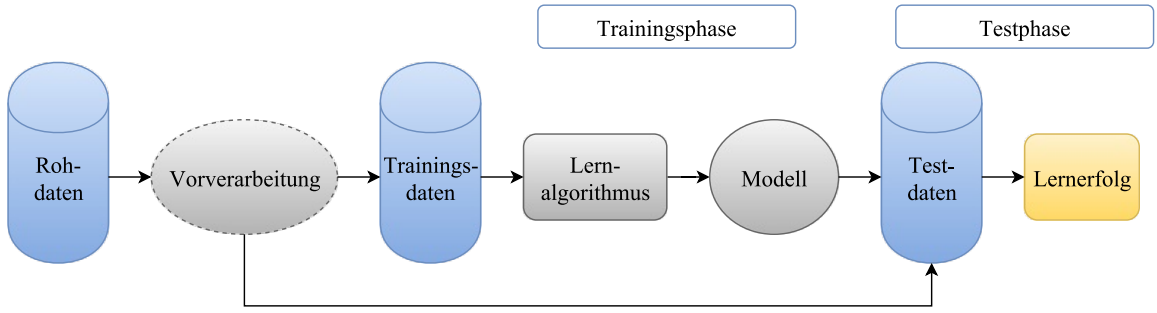


Abbildung 1: Funktionsweise des überwachten maschinellen Lernens

Soll hingegen mit Daten gelernt werden, die keine zugehörigen Labels aufweisen, so spricht man vom *unüberwachten Lernen* (engl. unsupervised learning). Die Methoden des unüberwachten Lernens werden insbesondere zur Beschreibung und Analyse der Strukturen von unklassifizierten Daten genutzt. Das unüberwachte Lernen findet vor allem bei der Wissensentdeckung in Datenbanken, dem sog. Data Mining, Anwendung, bei dem interessante Abhängigkeiten in großen Datensätzen erkannt werden sollen. Beispielsweise könnte ein Onlinehandel daran interessiert sein, welche Abhängigkeiten zwischen dem Kaufverhalten seiner Kunden bestehen, um darauf aufbauend gezielte Werbung bzw. kundenspezifische Produktvorschläge zu schalten (vgl. Wrobel et al., 2013, S. 406). Neben dem Clustern von Daten auf Basis erkannter Strukturen und Ähnlichkeiten findet das unüberwachte Lernen zudem zu Visualisierungszwecken bei der Projektion von Daten eines höher-dimensionalen Raums in den 2- oder 3-dimensionalen Raum Anwendung (vgl. Bishop, 2006, S. 3).

Eine dritte Technik des maschinellen Lernens, deren Funktionsweise aufgrund der Komplexität in diesem Kapitel nicht ausführlich beschrieben werden kann, stellt das sogenannte *bestärkende Lernen* (engl. reinforcement learning) dar. Eine wesentliche Gemeinsamkeit von Problemstellungen aus diesem Bereich ist, dass ganze Abfolgen von Aktionen gelernt werden müssen und dabei, im Gegensatz zum überwachten Lernen, keine Beispiele mit optimalem Ausgabewert vorgegeben sind. Lernmethoden des

[4]Im oben beschriebenen Beispiel der Bildklassifizierung von schwarz-weiß Bildern stellen die Dimensionen die Zeilen der Matrix dar, die die Vektoren der Bilder spaltenweise enthält.

bestärkenden Lernens finden u. a. bei der Steuerung industrieller Anlagen, in der Robotik oder bei Spielen, wie z. B. beim Backgammon, Anwendung (vgl. Wrobel et al., 2013, S. 406).

# 3 Mathematische Hintergründe zweier Methoden zur Bildklassifizierung

## 3.1 Gegebene Daten

Gegeben ist ein *Trainingsdatensatz*

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \cdots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \{1, \ldots, m\}. \tag{3.1}$$

Hier bezeichnet $\mathbb{R}^D$ den Merkmalsraum, aus dem die *Eingangsdaten* $\mathbf{x}_n$, $n = 1, \ldots, N$ stammen, und $t_n$ die zugehörigen *Labels*, welche die Zuordnung der Eingangsdaten zu den Klassen 1 bis $m$ angeben.
Zudem ist ein *Testdatensatz*

$$(\mathbf{x}_{N+1}, t_{N+1}), (\mathbf{x}_{N+2}, t_{N+2}), \cdots, (\mathbf{x}_{N+M}, t_{N+M}) \in \mathbb{R}^D \times \{1, \ldots, m\} \tag{3.2}$$

gegeben, dessen Daten aus demselben Datenraum wie die Trainingsdaten stammen. Dieser Testdatensatz dient dazu den Klassifizierungserfolg der entwickelten Methoden zu überprüfen und damit die Generalisierung des Modells auf unbekannte Daten zu evaluieren.
Im speziellen Fall der Klassifizierung von schwarz-weiß Bildern aus $n \times n$ Pixeln werden die Bilder durch Vektoren der Länge $D = n^2$ repräsentiert, deren Komponenten die Grauwerte der spaltenweise untereinandergeschrieben Pixel wiedergeben.
Anhand der Trainingsdaten soll nun eine Entscheidungsfunktion $f(\mathbf{x})$ gelernt werden, die die Zuordnung der Eingangsdaten zu den Klassen vornimmt - mit möglichst hohem Klassifizierungserfolg.
Neben der Klassifizierung von Bildern könnten auch andere Klassifizierungsprobleme auf gleiche Weise betrachtet werden. Die Einträge der Vektoren würden dann andere Angaben als die Grauwerte widerspiegeln. Ein Beispiel eines solchen Klassifizierungsproblem mit Eingangsdaten aus dem $\mathbb{R}^2$ ist in Kapitel 6 beschrieben. Bei der nachfolgenden Darstellung der mathematischen Hintergründe der beiden ausgewählten Klassifizierungsmethoden wird stets Bezug zur Problemstellung der Bildklassifizierung genommen.

## 3.2 Methode 1: Support Vektor Maschine

Eine *Support Vektor Maschine* (SVM) stellt eine Methode aus dem Bereich des maschinellen Lernens dar, die seit den 1970er Jahren insbesondere durch ihre Anwendbarkeit bei der Lösung von Klassifizierungs- und Regressionsproblemen an Bedeutung gewonnen hat. Das vorliegende Kapitel legt wesentliche mathematische Hintergründe einer Support Vektor Maschine dar. Der Fokus liegt dabei auf den Klassifizierungsproblemen, wobei die Erweiterung auf Regressionsprobleme möglich wäre (vgl. Burges, 1998, S. 121).

Die Auseinandersetzung mit dem Support Vektor Lernen ist aus verschiedenen Gründen interessant. Zum einen basiert es auf elementaren mathematischen Ideen. Es stellt eine lineare Methode in einem hochdimensionalen Merkmalsraum dar, der wiederum nichtlinear mit dem Raum der Eingangsdaten verknüpft ist. Es verdeutlicht zudem, was das Lernen anhand von Trainingsbeispielen charakterisiert. Zum anderen führt es bei der Lösung verschiedener Anwendungsprobleme zu sehr guten Ergebnissen. Der Kernpunkt der SVM, der diese von einfachen linearen Klassifikatoren abhebt, ist, dass komplexe Algorithmen für nichtlineare Klassifizierungsprobleme genutzt werden, die jedoch durch die Verwendung von sog. *Kernfunktionen* als einfache lineare Algorithmen betrachtet und analysiert werden können (vgl. Schölkopf, 1998, S.18). Ein weiterer Vorzug der SVM ist, dass die Modellparameter, die zur Lösung der Klassifizierungsprobleme festzulegen sind, zu einem konvexen Optimierungsproblem gehören. Jede lokale Lösung dieses Problems entspricht damit gleichzeitig einer globalen Lösung (vgl. Bishop, 2006, S. 325).

In ihrer elementaren Form werden SVMs für binäre Klassifizierungsprobleme genutzt, d. h. für solche, die aus nur zwei Klassen bestehen.
Im Folgenden wird ausgehend vom einfachsten Fall, zwei Klassen mit linear separierbaren Daten, ein Algorithmus beschrieben, der die Klassen durch Bestimmung einer *optimalen Hyperebene* voneinander trennt. Ohne Beschränkung der Allgemeinheit werden die Daten der einen Klasse dazu zunächst mit dem Label $+1$, die Daten der anderen Klasse mit dem Label $-1$ versehen. Somit gilt $t_n \in \{\pm 1\}$ für $n = 1, \ldots, N + M$.
Der für linear separierbare Probleme entwickelte Algorithmus wird dann mithilfe des *Kern-Tricks* auf den Fall nichtlinear separierbarer Daten erweitert. Anschließend wird dargestellt, wie durch die Kombination mehrerer binärer SVMs auch die Lösung von Klassifizierungsproblemen mit mehr als zwei Klassen erfolgen kann.

### 3.2.1 Lineare Support Vektor Maschine

Zunächst wird der folgende Fall betrachtet: Die Trainingsdaten zweier Klassen seien im Eingangsdatenraum linear separierbar. Für diese Daten existiert per Definition mindestens eine Hyperebene der Form

$$\mathbf{w}^T\mathbf{x} + b = 0, \tag{3.3}$$

welche die positiv und negativ gelabelten Trainingsdaten voneinander trennt, wobei $\mathbf{w} \in \mathbb{R}^D$ dem Normalenvektor und $\frac{|b|}{\|\mathbf{w}\|}$ dem Abstand der Hyperebene zum Ursprung entspricht. Die *Entscheidungsfunktion* $f$ zur Hyperebene (3.3), welche die Zuordnung eines Datenpunktes $\mathbf{x}$ zu einer der beiden Klassen vornimmt, ist dann gegeben durch

$$\begin{aligned} f : \mathbb{R}^D &\longrightarrow \{\pm 1\} \\ \mathbf{x} \mapsto f(\mathbf{x}) &= \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b). \end{aligned} \tag{3.4}$$

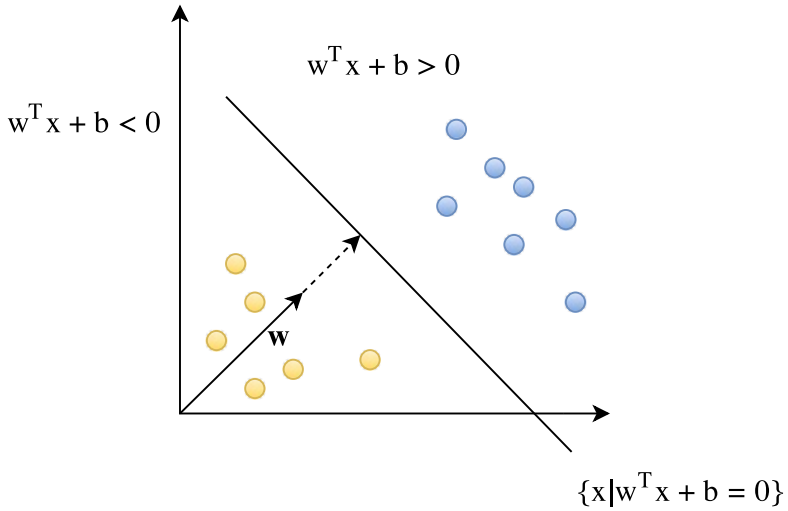


Abbildung 2: Binäres Klassifizierungsproblem linear separierbarer Daten mit trennender Hyperebene ausgedrückt durch den Normalenvektor $\mathbf{w}$ und den Schwellenwert $b$. Die Multiplikation der Parameter $\mathbf{w}$ und $b$ mit der gleichen Konstanten ungleich null liefert die selbe Hyperebene, repräsentiert durch andere Parameter $\mathbf{w}$ und $b$ (in Anlehnung an: Schölkopf & Smola, 2001, S. 191).

Dabei bezeichnet $\operatorname{sgn}(x)$ die Vorzeichenfunktion, die definiert ist als

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{falls } \; x < 0 \\ 0 & \text{falls } \; x = 0 \\ 1 & \text{falls } \; x > 0. \end{cases}$$

Für alle Trainingsdaten $\mathbf{x}_n$, $n = 1, \ldots, N$ gelten damit die folgenden Ungleichungen

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_n + b \geq 0 \;\; \text{für} \;\; t_n = +1 \\ \mathbf{w}^T \mathbf{x}_n + b \leq 0 \;\; \text{für} \;\; t_n = -1. \end{aligned} \tag{3.5}$$

Im 2-dimensionalen Fall liegen die Punkte der positiv-gelabelten Klasse somit oberhalb, die Punkte der negativ-gelabelten Klasse unterhalb der Hyperebene, wie in Abbildung 2 dargestellt.[5] Die Ungleichungen (3.5) können zusammengefasst werden zu

$$t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 0, \quad n = 1, \ldots, N, \tag{3.6}$$

wobei Gleichheit in (3.6) für Punkte gilt, die auf der Hyperebene liegen.

Nun existieren ggf. mehrere Hyperebenen, die die Trainingsdaten korrekt separieren.

[5] Wenngleich die Daten bei Bildklassifizierungsproblemen i. d. R. nicht aus dem $\mathbb{R}^2$ stammen, werden nachfolgend stets Abbildungen herangezogen, die den 2-dimensionalen Fall darstellen, um die mathematischen Ideen und Hintergründe auch visualisieren zu können.

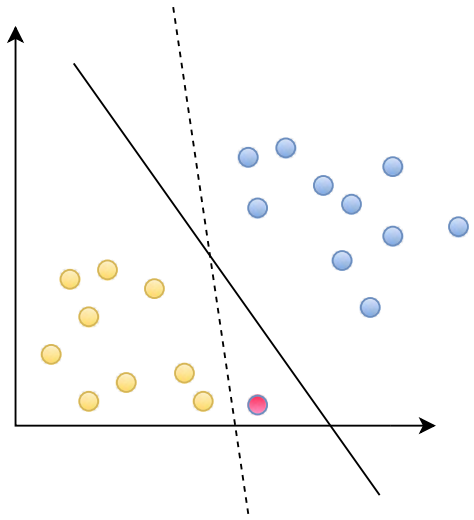

Abbildung 3: Zwei Hyperebenen, die die Trainingsdaten der gelben und der blauen Klasse voneinander separieren. In rot ist ein Testdatenpunkt dargestellt, der von der gestrichelten Hyperebene der blauen Klasse zugeteilt würde, obwohl der Punkt deutlich näher an den Trainingsdaten der gelben Klasse liegt.

Damit ergibt sich die Frage, welche dieser Hyperebenen die *bestmögliche* ist, d. h. welche dieser Hyperebenen neue, unbekannte Datenpunkte mit großer Wahrscheinlichkeit richtig klassifiziert. Bei dem in Abbildung 3 dargestellten Beispiel würde man intuitiv womöglich die Hyperebene wählen, die den größeren Abstand zu allen Trainingsdaten aufweist. Dies entspricht gerade der Vorgehensweise beim Support Vektor Training: Unter allen möglichen Hyperebenen wird diejenige gesucht, die den größten Bereich frei von Datenpunkten hinterlässt. Dieser datenpunktfreie Bereich wird als *Margin* bezeichnet (vgl. Abbildung 4). Der Support Vektor Algorithmus sucht folglich aus der Menge aller möglicher Hyperebenen, die (3.6) erfüllen, jene mit maximalem Margin (vgl. Burges, 1998, S. 128).

Um den Margin in Formeln darstellen zu können, betrachten wir zunächst den Abstand eines beliebigen Datenpunktes $\mathbf{x}_n$ zu der Hyperebene (3.3), der gegeben ist durch

$$\frac{|\mathbf{w}^T\mathbf{x}_n + b|}{\|\mathbf{w}\|}. \tag{3.7}$$

Dieser kann mit (3.6) geschrieben werden als

$$\frac{t_n(\mathbf{w}^T\mathbf{x}_n + b)}{\|\mathbf{w}\|}, \tag{3.8}$$

wobei $\|.\|$ die euklidische Norm bezeichne.

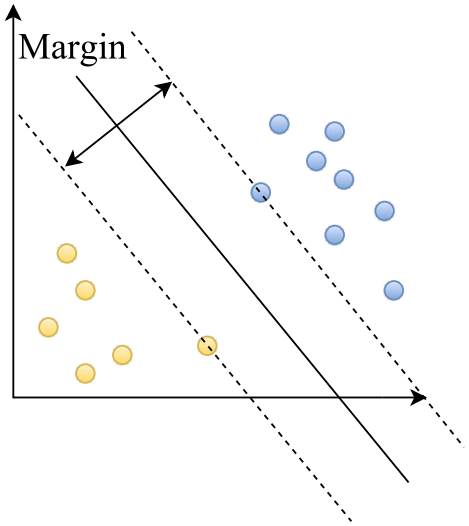


Abbildung 4: Der Margin entspricht dem datenpunktfreien Bereich zwischen den Datenpunkten beider Klassen.

Der Margin[6] ist dann gegeben durch den doppelten Abstand des *nächsten* (bzgl. des Abstands) Datenpunktes $\mathbf{x}_s$ von der Hyperebene. Dies kann ausgedrückt werden durch

$$2 \cdot \min_n \frac{t_n(\mathbf{w}^T\mathbf{x}_n + b)}{\|\mathbf{w}\|}. \tag{3.9}$$

Die Parameter $\mathbf{w}$ und $b$ der Hyperebene sind nun so zu wählen, dass der Margin gegeben durch (3.9) maximiert wird. Damit ergibt sich die optimale Hyperebene als Lösung des Optimierungsproblems

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \cdot \min_n \left[ t_n(\mathbf{w}^T\mathbf{x}_n + b) \right] \right\}. \tag{3.10}$$

Da der Abstand eines beliebigen Punktes $\mathbf{x}_n$ von der Hyperebene gegeben durch (3.7) invariant gegenüber Skalierung von $\mathbf{w}$ und $b$ mit dem selben Faktor (ungleich null) ist, kann dieses Problem in ein äquivalentes einfacheres Problem überführt werden, bei dem die *nächsten* Datenpunkte, wie in Abbildung 5 dargestellt, gerade auf den Hyperebenen $\mathbf{w}^T\mathbf{x} + b = 1$ bzw. $\mathbf{w}^T\mathbf{x} + b = -1$ liegen. Es gilt folglich

$$\min_n t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1. \tag{3.11}$$

Die so skalierte Hyperebene wird auch als *kanonische Form* der Hyperebene bezeichnet (vgl. Schölkopf & Smola, 2001, S. 192).
Das Optimierungsproblem (3.10) lässt sich dann mit (3.11) auf die Maximierung von

$$\|\mathbf{w}\|^{-1}$$

[6]Wird im Folgenden davon gesprochen, dass ein Punkt *auf dem Margin* liegt, so ist gemeint, dass der Punkt auf einer der Hyperebenen parallel zu der separierenden Hyperebene liegt, die einen dem Margin entsprechenden Abstand voneinander aufweisen. In Abbildung 4 entsprechen diese Hyperebenen den gestrichelt dargestellten Geraden.

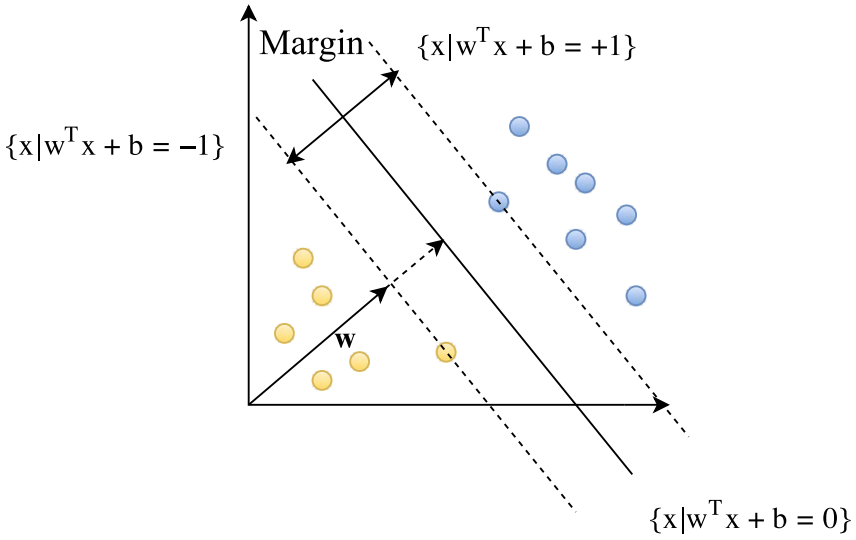


Abbildung 5: Die Parameter der optimalen Hyperebene werden so skaliert, dass die *nächstgelegenen* Datenpunkte auf den Hyperebenen $\mathbf{w}^T\mathbf{x}_n + b = 1$ bzw. $\mathbf{w}^T\mathbf{x}_n + b = -1$ liegen (in Anlehnung an: Schölkopf & Smola, 2001, S. 191).

bzw. äquivalent auf die Minimierung von

$$\frac{1}{2}\|\mathbf{w}\|^2$$

reduzieren, wobei für alle Datenpunkte des Trainingsdatensatzes

$$t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \; n = 1, \ldots, N \tag{3.12}$$

gelten muss, was geschrieben werden kann als

$$t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0, \; n = 1, \ldots, N. \tag{3.13}$$

Um die Hyperebene mit maximalem Margin zu finden, ist somit das folgende Optimierungsproblem unter Nebenbedingungen[7] zu lösen.

$$\underset{\mathbf{w}\in\mathbb{R}^D, b\in\mathbb{R}}{\text{Minimiere}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.14}$$

$$\text{u. d. Nb.} \quad t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0, \; n = 1, \ldots, N. \tag{3.15}$$

[7]Im Folgenden wird der Ausdruck *unter den Nebenbedingungen* mit u. d. Nb. abgekürzt.

Dieses Optimierungsproblem stellt ein quadratisches Optimierungsproblem in seiner *primalen Form* dar, bei dem eine quadratische konvexe Zielfunktion unter Berücksichtigung einer Menge von linearen Ungleichungen als Nebenbedingungen zu minimieren ist. Da alle Punkte, die die Nebenbedingungen erfüllen, eine konvexe Menge bilden (jede lineare Ungleichung definiert eine konvexe Menge und auch der Schnitt der durch die $N$ Ungleichungen erzeugten Mengen ist konvex), liegt ein konvexes Optimierungsproblem vor. Jede lokale Lösung eines solchen Optimierungsproblems entspricht damit gleichzeitig einer globalen Lösung (vgl. Bishop, 2006, S. 325; Hastie et al., 2001, S. 419).

**Lagrange Formulierung**

Im Folgenden wird zu dem Optimierungsproblem (3.14) durch Verwendung von Lagrange Multiplikatoren zunächst die *primale* und anschließend die *duale* Lagrange-Funktion formuliert. Die Darstellung in der dualen Form hat zwei Vorteile: Zum einen werden die Nebenbedingungen (3.13) durch Nebenbedingungen ersetzt, die nur von den Lagrange Multiplikatoren abhängen, was das Lösen des Optimierungsproblems vereinfacht, zum anderen tauchen die Trainingsdaten in der dualen Darstellung lediglich in Form von Skalarprodukten auf, was eine grundlegende Eigenschaft für die Erweiterung auf den nichtlinearen Fall darstellt (vgl. Burges, 1998, S.129).

Zunächst werden für jede der $N$ Nebenbedingungen (3.13) Lagrange Multiplikatoren $a_n \in \mathbb{R}$ mit $a_n \geq 0$, $n = 1, \ldots, N$ eingeführt und die Lagrange Funktion

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \Big[ t_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 \Big] \tag{3.16}$$

mit $\mathbf{a} = (a_1, \cdots a_N)^T$ formuliert (vgl. Bishop, 2006, S. 328; Schölkopf & Smola, 2001, S. 13). Leser, die nicht vertraut mit Lagrange Multiplikatoren und deren Verwendung beim Lösen von Optimierungsproblemen sind, finden bei Bishop (2006, S. 707) eine kurze und für das Verständnis der folgenden Ausführungen hilfreiche Erläuterung.
Die Lösung des Optimierungsproblems ergibt sich nun durch Maximierung der Lagrange Funktion $L$ bezüglich der $a_n$ und Minimierung bezüglich der primalen Variablen $\mathbf{w}$ und $b$. Folglich ist ein Sattelpunkt der Funktion $L(\mathbf{w}, b, \mathbf{a})$ zu bestimmen, in dem die partiellen Ableitungen bezüglich der primalen Variablen verschwinden. Damit folgt:

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \mathbf{a}) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n \tag{3.17}$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \mathbf{a}) = 0 \quad \Rightarrow \quad 0 = \sum_{n=1}^{N} a_n t_n. \tag{3.18}$$

Der Normalenvektor $\mathbf{w}$ der optimalen Hyperebene kann wegen (3.17) als Linearkombination der Trainingsdaten dargestellt werden (vgl. Schölkopf & Smola, 2001, S. 196 ff.).

Durch Berücksichtigung von (3.17) und (3.18) lässt sich (3.16) in die duale Form $L_d$ umformulieren:

$$\begin{aligned} L(\mathbf{w}, b, \mathbf{a}) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \Big[ t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \Big] \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n t_n \mathbf{w}^T\mathbf{x}_n - b\sum_{n=1}^{N} a_n t_n + \sum_{n=1}^{N} a_n \\ &\overset{(3.18)}{=} \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n t_n \mathbf{w}^T\mathbf{x}_n + \sum_{n=1}^{N} a_n \\ &\overset{(3.17)}{=} \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n t_n \sum_{m=1}^{N} a_m t_m \mathbf{x}_m^T\mathbf{x}_n + \sum_{n=1}^{N} a_n \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^T\mathbf{x}_n) + \sum_{n=1}^{N} a_n. \end{aligned} \tag{3.19}$$

Mit $\frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^T\mathbf{w}$ und $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$ erhält man dann

$$L_d(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^T\mathbf{x}_n). \tag{3.20}$$

Die Lösung der Hyperebene mit maximalem Margin ergibt sich schließlich als Lösung des folgenden dualen Optimierungsproblems.

| | | |
|---|---|---|
| Maximiere $\underset{\mathbf{a}\in\mathbb{R}^N}{}$ | $L_d(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^T\mathbf{x}_n)$ | (3.21) |
| u. d. Nb. | $a_n \geq 0,\ n = 1, \ldots, N,$ | (3.22) |
| und | $\sum_{n=1}^{N} a_n t_n = 0.$ | (3.23) |

Während des Support Vektor Trainings wird $L_d$ bezüglich der $a_n$ unter den Nebenbedingungen (3.23) und Positivität der $a_n$ maximiert und eine Lösung der Form

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n \tag{3.24}$$

als Linearkombination der Trainingsdaten erhalten (vgl. Burges, 1998, S.130).
Die in diesem Kapitel für die Lösung von konvexen Optimierungsproblemen mit Nebenbedingungen verwendeten Konzepte, wie die Sattelpunkts-Bedingungen, die Lagrange Formulierungen sowie die primale und duale Optimierung, können aufgrund des begrenzten Rahmens dieser Arbeit nicht ausführlich diskutiert werden. Einen guten

Überblick über das notwendige theoretische Werkzeug zum Lösen von Optimierungsproblemen im Kontext von Lernmethoden mit Kernen findet sich bei Schölkopf & Smola (2001, Kapitel 6). Dort werden überdies notwendige Bedingungen diskutiert, damit die Lösung des dualen Optimierungsproblems mit der Lösung des primalen Problems übereinstimmt.

### Stützvektoren

Für das gegebene konvexe Optimierungsproblem mit Nebenbedingungen stellen die *Karush-Kuhn-Tucker* (KKT) Bedingungen notwendige und hinreichende Optimalitätskriterien dar. Neben den Sattelpunktsbedingungen (3.17) und (3.18) beinhalten diese für das Optimierungsproblem in primaler Form (3.16) folgende weitere Bedingungen:

$$a_n \geq 0, \quad n = 1, \cdots, N \tag{3.25}$$

$$t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 \geq 0, \quad n = 1, \cdots, N \tag{3.26}$$

$$a_n \cdot \Big[t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1\Big] = 0, \quad n = 1, \cdots, N. \tag{3.27}$$

Wegen Bedingung (3.27), die als *Komplementäritätsbedingung* bezeichnet wird, gilt für alle Datenpunkte $\mathbf{x}_n$ entweder $a_n = 0$ oder $t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$. In die Lösung für den Normalenvektor $\mathbf{w}$ der Hyperebene mit maximalem Margin (3.24) gehen wegen (3.27) folglich nur die Trainingsdaten mit Lagrange Multiplikator $a_n > 0$ ein. Diese (Trainings-)Vektoren werden als Stützvektoren bzw. als *Support Vektoren* (SV) bezeichnet (vgl. Bishop, 2006, S. 330).
Da für alle SVs $t_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$ gilt, liegen diese gerade auf dem Margin. Alle übrigen Datenpunkte mit $a_n = 0$ liegen entweder ebenfalls auf dem Margin oder auf der Seite der Hyperebene, sodass die Nebenbedingungen (3.13) strikt erfüllt sind. Betrachten wir das Beispiel der Bildklassifizierung der handgeschriebenen Ziffern von 0 bis 9. Bei der binären Klassifizierung der Bildklasse zur Zahl 1 und der Bildklasse zur Zahl 2 entsprechen die SVs dann gerade den Bildern, die am nächsten an der Entscheidungsgrenze liegen und damit der jeweils anderen Bildklasse *am ähnlichsten sehen*. Die Lösung für den Normalenvektor $\mathbf{w}$ der Hyperebene mit maximalem Margin kann nun geschrieben werden als

$$\mathbf{w} = \sum_{n \in S} a_n t_n \mathbf{x}_n, \tag{3.28}$$

wobei $S$ der Menge aller Indizes der SVs entspricht. Diese Darstellung hat insbesondere den Vorteil einer ökonomischeren Datenspeicherung und einer schnelleren Klassifizierungen unbekannter Datenpunkte.

### Bestimmung des Parameters b

Nachdem das quadratische Progammierungsproblem gelöst und für $\mathbf{a} = (a_1, \ldots a_N)$ eine Lösung gefunden wurde, ist der Wert für den Parameter $b$ zu bestimmen. Berücksichtigt man, dass für jeden SV $\mathbf{x}_s$

$$t_s(\mathbf{w}^T\mathbf{x}_s + b) = 1 \tag{3.29}$$

gilt und setzt (3.28) in (3.29) ein, erhält man

$$t_s\Big(\sum_{n\in S} a_n t_n \mathbf{x}_n^T \mathbf{x}_s + b\Big) = 1. \tag{3.30}$$

Berücksichtigung von $t_s^2 = 1$ und Multiplikation beider Seiten der Gleichung mit $t_s$ liefert

$$b = t_s - \sum_{n\in S} a_n t_n \mathbf{x}_n^T \mathbf{x}_s. \tag{3.31}$$

Der Parameter $b$ kann damit durch Einsetzen eines beliebigen Stützvektors $\mathbf{x}_s$ in (3.31) gefunden werden. Eine numerisch stabilere Lösung ergibt sich jedoch durch Mitteln der Gleichung (3.31) über alle SVs gemäß

$$b = \frac{1}{N_S}\sum_{n\in S}\Big(t_n - \sum_{m\in S} a_m t_m \mathbf{x}_m^T \mathbf{x}_n\Big), \tag{3.32}$$

wobei $N_S$ der Anzahl der SVs entspricht (vgl. Bishop, 2006, S. 330).

### Klassifizierung von Testdaten

Einsetzen der erhaltenen Lösung für die Parameter $b$ und $\mathbf{w}$ der Hyperebene mit maximalem Margin in (3.4) liefert die Entscheidungsfunktion

$$f(\mathbf{x}) = \operatorname{sgn}\Big(\sum_{n\in S} a_n t_n \mathbf{x}_n^T \mathbf{x} + b\Big). \tag{3.33}$$

Abhängig davon, auf welcher Seite der Hyperebene ein zu klassifizierende Testpunkt $\mathbf{x}$ liegt, ergibt sich ein positives oder negatives Vorzeichen, welches dann die Zuordnung zu einer der beiden Klassen festlegt (vgl. Burges, 1998, S.130).

### Nichtlinear separierbare Daten

Bisher wurde der Fall betrachtet, in dem die Trainingsdaten im Merkmalsraum linear separierbar sind, womit diese folglich mit einer Hyperebene exakt getrennt werden können. Was geschieht jedoch, wenn es zu einer Überlappung der beiden Klassen kommt, wie exemplarisch im Abbildung 6 dargestellt? In diesem Fall sollen die *harten* Nebenbedingungen (3.12) abgeschwächt werden. Die SVM wird so modifiziert, dass eine falsche Klassifizierung einzelner Datenpunkte erlaubt ist. Dazu werden für jeden Datenpunkt $\mathbf{x}_n$, $n = 1, \ldots, N$, sog. *Schlupfvariablen*, im Folgenden Slack-Variablen (engl. slack variables) genannt, $\xi_n \geq 0$ eingeführt und die Nebenbedingungen (3.12) in abgeschwächter Form formuliert

$$\begin{aligned} \mathbf{w}^T\mathbf{x}_n + b &\geq +1 - \xi_n \quad \text{für } t_n = +1 \\ \mathbf{w}^T\mathbf{x}_n + b &\leq -1 + \xi_n \quad \text{für } t_n = -1. \end{aligned} \tag{3.34}$$

Diese können zusammengefasst werden zu

$$t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n \quad \text{für } n = 1, \cdots, N. \tag{3.35}$$

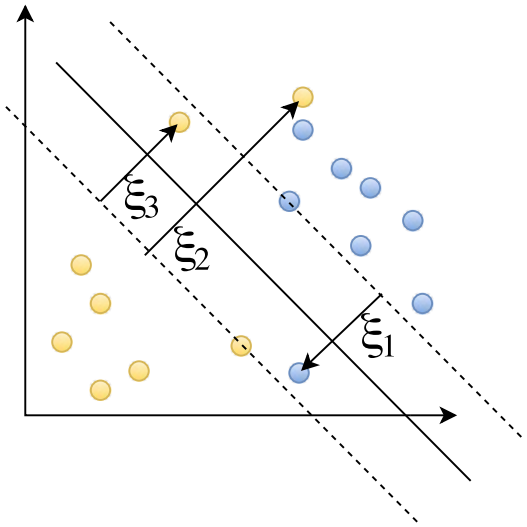


Abbildung 6: Slack-Variablen $\xi_n$ eines binären Klassifizierungsproblems mit überlappenden Verteilungen der beiden Klassen

Punkte mit $\xi_n = 0$ werden richtig klassifiziert und liegen entweder auf dem Margin oder auf der richtigen Seite des Margins. Alle Punkte mit $0 < \xi_n \leq 1$ liegen innerhalb des Margins, jedoch auf der richtigen Seite der Entscheidungshyperebene. Punkte mit $\xi_n > 1$ liegen auf der falschen Seite der Entscheidungshyperebene und werden nicht korrekt klassifiziert. Damit ist $\sum_{n=1}^{N} \xi_n$ eine obere Grenze für die Anzahl der falsch klassifizierten Punkte (vgl. Bishop, 2006, S. 331). Die Klassifizierung mit *weichen* Nebenbedingungen (3.35) wird als *Soft Margin Klassifizierung* bezeichnet, wohingegen die Klassifizierung mit harten Nebenbedingungen (3.12) *Hard Margin Klassifizierung* genannt wird.
Das Optimierungsproblem (3.23) wird nun modifiziert, indem ein weiterer Kostenfaktor $C > 0$ eingeführt und die Slack-Variablen berücksichtigt werden. Dies führt zu der folgenden zu minimierenden Zielfunktion

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n, \quad \text{für } n = 1, \cdots, N. \tag{3.36}$$

Die zugehörige Lagrange Funktion mit neu eingeführten Lagrange Multiplikatoren $\eta_n \geq 0, n = 1, \ldots, N$ lautet dann in primaler Form

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n\Big[t_n(\mathbf{w}^T\mathbf{x}_n + b) - 1 + \xi_n\Big] - \sum_{n=1}^{N}\eta_n\xi_n \tag{3.37}$$

unter den Bedingungen (3.35) und $\xi_n \geq 0$, $a_n \geq 0$. Die KKT-Bedingungen für dieses Optimierungsproblem berücksichtigend, die bei Bishop (2006, S. 233) nachvollzogen werden können, ergibt sich schließlich die duale Form des Soft Margin Klassifizierungsproblems.

$$\underset{\mathbf{a}\in\mathbb{R}^N}{\text{Maximiere}} \quad L_d(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^T \mathbf{x}_n) \tag{3.38}$$

$$\text{u. d. Nb.} \quad 0 \leq a_n \leq C, \; n = 1, \ldots, N, \tag{3.39}$$

$$\text{und} \quad \sum_{n=1}^{N} a_n t_n = 0. \tag{3.40}$$

Dieses quadratische Optimierungsproblem unterscheidet sich lediglich durch die obere Schranke $C$ für die Lagrange Multiplikatoren von dem linear separierbaren Problem (3.21). Weder die Slack-Variablen, noch die neu eingeführten Lagrange Multiplikatoren tauchen in dieser Formulierung auf. Die Lösung von (3.38) ist erneut gegeben durch

$$\mathbf{w} = \sum_{n\in S} a_n t_n \mathbf{x}_n. \tag{3.41}$$

Der Parameter $b$ lässt sich analog zum linear separierbaren Fall bestimmen und ist gegeben durch

$$b = \frac{1}{N_M} \sum_{n\in M} \left( t_n - \sum_{m\in S} a_m t_m \mathbf{x}_m^T \mathbf{x}_n \right), \tag{3.42}$$

wobei $M$ der Menge der Indizes entspricht, deren Datenpunkte $0 < a_n < C$ erfüllen (vgl. Burges, 1998, S.135; Bishop, 2006, S. 334).
Betrachtet man die zu minimierende Zielfunktion (3.36) genauer, so wird der Einfluss der Größe des Kostenfaktors $C$ ersichtlich. Ein großer Wert für $C$ führt bei Minimierung der Zielfunktion (3.36) zu einem kleinen Wert für die Summe $\sum_{n=1}^{N} \xi_n$ der Slack-Variablen. Damit liegen nur wenige Datenpunkte innerhalb des Margins bzw. auf der falschen Seite der Hyperebene. Wird hingegen der Wert von $C$ verkleinert, so wächst die Summe der Slack-Variablen, womit auch die Zahl der Datenpunkte innerhalb des Margins bzw. auf der falschen Seite der Hyperebene wächst. Über den Kostenfaktor $C$ kann demnach zwischen einer Überanpassung (engl. overfitting) der Hyperebene an die konkreten Daten und einer Unteranpassung (engl. underfitting) reguliert werden (vgl. Bishop, 2006, S. 334).

### 3.2.2 Nichtlineare Support Vektor Maschine

Bisher wurde gezeigt, dass die Hyperebene mit maximalem Margin im linear separierbaren Fall einen guten Klassifikator liefert, für dessen Berechnung geeignete Methoden existieren. Nun stellt sich die Frage, wie vorgegangen wird, wenn die Daten im Eingangsraum nicht durch eine lineare Entscheidungsfunktion zu trennen sind, wie in Abbildung 7 dargestellt?

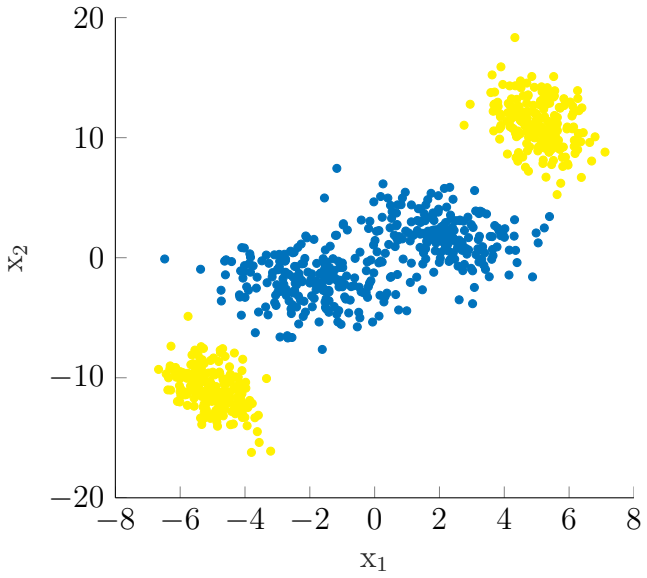


Abbildung 7: Nichtlinear separierbare Daten im 2-dimensionalen Eingangsdatenraum.

Um die beschriebene Technik, die eine optimale Hyperebene für lineare Separationen bestimmt, dennoch nutzen zu können, werden die Eingangsdaten mithilfe einer nichtlinearen Abbildung $\Phi$ in einen höher-dimensionalen euklidischen Merkmalsraum $H$ abgebildet - mit der Idee, die Daten in diesem höher-dimensionalen Raum linear separieren zu können. Diese Idee stützt sich auf den *Satz von Cover*. Dieser besagt, dass die Zahl der theoretisch möglichen linearen Separationen von $N$ Punkten in einem $M$-dimensionalen Raum gemäß

$$2 \cdot \sum_{i=0}^{M-1} \binom{N-1}{i} \tag{3.43}$$

steigt (vgl. Cover, 1965, S. 326; Schölkopf & Smola, 2001, S. 200). Die qualitative Aussage dieses Satzes lautet, dass ein Klassifizierungsproblem mit höherer Wahrscheinlichkeit linear separierbar ist, wenn die Daten von einem niedrig-dimensionalen Merkmalsraum in einen hoch-dimensionalen Merkmalsraum abgebildet werden.
Im nichtlinear separierbaren Fall werden die Datenpunkte $\mathbf{x}_n$ somit zunächst mit einer nichtlinearen Abbildung $\Phi$ mit

$$\begin{aligned} \Phi : \mathbb{R}^D &\longrightarrow H \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}) \end{aligned} \tag{3.44}$$

in einen höher-dimensionalen Merkmalsraum $H$ abgebildet, in dem diese mit höherer Wahrscheinlichkeit durch einen linearen Klassifikator separierbar sind (vgl. Abbildung 8). Mit dem bisherigen Modell wird wiederum ein linearer Klassifikator bestimmt, jedoch in einem anderen Merkmalsraum und auf nichtlinear separierbaren Daten (vgl. Burges, 1998, S. 138).

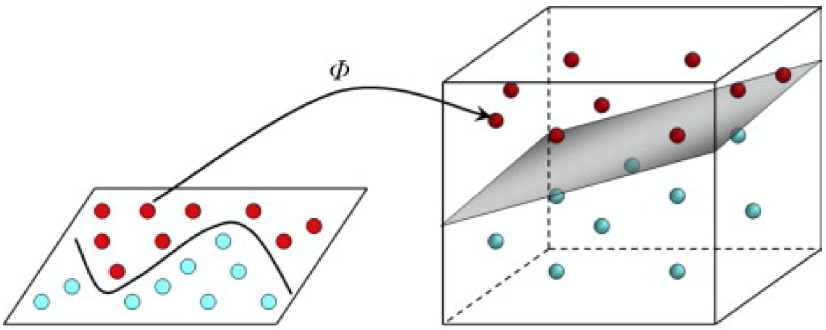


Abbildung 8: Daten, die im 3-dimensionalen Merkmalsraum linear separierbar sind. Der lineare Klassifikator im $\mathbb{R}^3$ kann als nichtlinearer Klassifikator im 2-dimensionalen Eingangsdatenraum interpretiert werden (entnommen aus: Martínez-de Pisón et al., 2008).

Zur Klassifizierung eines Datenpunktes ist dann die Entscheidungsfunktion

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \Phi(\mathbf{x}) + b). \tag{3.45}$$

auszuwerten. Nun ergibt sich das Problem, dass die explizite Auswertung von $\Phi(\mathbf{x})$, falls in einen hoch-dimensionalen oder gar unendlich-dimensionalen Merkmalsraum $H$ abgebildet wird, höchst aufwendig sein kann.
An dieser Stelle kommt die Beobachtung zum Tragen, dass die Eingangsdaten lediglich in Form von Skalarprodukten $\mathbf{x}_i^T \mathbf{x}_j$ in dem Trainingsproblem (3.38) - (3.40) erscheinen, womit die transformierten Daten $\Phi(\mathbf{x}_n)$ in der neuen Entscheidungsfunktion ebenfalls nur in Form von Skalarprodukten auftauchen:

$$f(\mathbf{x}) = \text{sgn}\Big( \sum_{n \in S} a_n t_n \phi(\mathbf{x}_n)^T \phi(\mathbf{x}) + b \Big) \tag{3.46}$$

Um die explizite Berechnung von $\Phi$ zu umgehen wird der *Kern-Trick*[8] angewendet. Die grundlegende Idee dieses Tricks ist die Folgende: Tauchen die Eingangsdaten bei einem Algorithmus nur in Form von Skalarprodukten auf, so werden diese Skalarprodukte durch Skalarprodukte eines höher-dimensionalen Raums ersetzt (vgl. Burges, 1998, S. 138).
Dazu wird zunächst eine Funktion

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j), \tag{3.47}$$

eine sog. *Kernfunktion* (vgl. Definition 3.1) definiert, die wiederum ein Skalarprodukt in einem anderen Merkmalsraum $H$ darstellt, und durch die jedes Skalarprodukt des Eingangsdatenraums ersetzt wird (vgl. Bishop, 2006, S. 292).

[8] Dieser Trick findet nicht nur bei SVMs, sondern auch bei weiteren Lernalgorithmen wie bei Nearest-Neighbour-Klassifikatoren Anwendung (vgl. Bishop, 2006, S. 292).

Dieses Vorgehen hat den Vorteil, dass die Kernfunktion lediglich implizit in einen höher-dimensionalen Raum abbildet. Die Funktion $\Phi$ muss nicht explizit angegeben werden und auch die Berechnung von $\Phi(\mathbf{x})$ wird vermieden. Formal kann ein Kern wie folgt definiert werden:

**Definition 3.1.** *Eine Funktion*

$$k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$$

*heißt Kernfunktion oder Kern (engl. kernel), wenn es einen Skalarproduktraum $H$ und eine Abbildung $\Phi : \mathbb{R}^D \to H$ gibt, sodass für alle $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^D$ gilt*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j) \tag{3.48}$$

*(vgl. Shawe-Taylor & Cristianini, 2004, S. 34).*

Nun stellt sich die Frage, welche Eigenschaften eine Funktion $k$ erfüllen muss, damit ein Paar $\{\Phi, H\}$ konstruiert werden kann, sodass (3.48) gilt und $k$ somit eine Kernfunktion darstellt.
Eine notwendige und hinreichende Bedingung liefert das folgende Lemma, welches eng mit einem bekannten Satz der Funktionalanalysis, dem *Satz von Mercer*, verknüpft ist.

**Lemma 3.1.** *Angenommen für die Eingangsdaten $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ und eine Funktion $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ ist die durch $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ definierte Gram-Matrix symmetrisch positiv definit. Dann kann eine Abbildung $\Phi$ in einen $N$-dimensionalen Merkmalsraum $H$ konstruiert werden, sodass*

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j) \tag{3.49}$$

*für alle $i, j = 1, \ldots, N$ gilt (vgl. Schölkopf & Smola, 2001, S. 44).*

Ohne die Funktion $\Phi$ explizit zu bestimmen, liefert dieses Lemma abhängig von den gegebenen Daten eine Aussage darüber, wann eine Funktion ein valider Kern ist. Neben diesem Lemma existieren weitere, insbesondere auch datenunabhängige Aussagen, die allgemeiner formulieren, wann und zudem wie zu einer gegebenen Funktion $k$ die Konstruktion eines Paares $\Phi$ und $H$ möglich ist. Eine ausführliche Diskussion dazu findet man bei Schölkopf & Smola (2001, S. 36 ff.).

Um die Idee des SV-Lernens im nichtlinearen Fall verständlicher zu machen, betrachten wir abschließend folgendes Beispiel: Angenommen die Eingangsdaten stammen aus dem $\mathbb{R}^2$. Die Funktion $k$, definiert durch $k(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2$, soll als Kernfunktion verwendet werden. Als Skalarproduktraum $H$ kann der $\mathbb{R}^3$ und für $\Phi$ die Abbildung

$$\Phi(\mathbf{x}_i) = \Phi \begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix} = \begin{pmatrix} x_{i1}^2 \\ x_{i2}^2 \\ \sqrt{2} x_{i1} x_{i2} \end{pmatrix}$$

gewählt werden. Dann gilt

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i^T \mathbf{x}_j)^2 \\ &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \\ &= (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2})^T (x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}) \\ &= \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j), \end{aligned}$$

womit die Funktion $k$ ein Kern ist (vgl. Schölkopf & Smola, 2001, S. 201).
Drei Beispiele für gängige Kernfunktionen, die zur Klassifizierung mit dem SV-Algorithmus genutzt werden, sind

- der lineare Kern, der dem Skalarprodukt im Eingangsdatenraum entspricht

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y}) \tag{3.50}$$

- Polynomiale Kerne k-ten Grades

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^k \tag{3.51}$$

- Gauß'sche Radiale-Basis-Funktionen (RBF-Kerne)

$$k(\mathbf{x}, \mathbf{y}) = exp\Big\{ - \frac{(\mathbf{x} - \mathbf{y})^2}{2\sigma^2} \Big\} \tag{3.52}$$

(vgl. Burges, 1998, S.142).
Wendet man den Kern-Trick beim Training von nichtlinearen SVMs an, d. h. substituiert man jedes Skalarprodukt $\mathbf{x}_i^T \mathbf{x}_j$ in (3.38) durch $k(\mathbf{x}_i, \mathbf{x}_j)$, so ergibt sich das folgende duale Optimierungsproblem:

$$\underset{\mathbf{a} \in \mathbb{R}^N}{\text{Maximiere}} \quad L_d(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_m, \mathbf{x}_n) \tag{3.53}$$

$$\text{u. d. Nb.} \quad 0 \leq a_n \leq C \tag{3.54}$$

$$\text{und} \quad \sum_{n=1}^{N} a_n t_n = 0. \tag{3.55}$$

Die Lösung dieses Optimierungsproblems und die anschließende Bestimmung des Parameters $b$ analog zum linear separierbaren Fall liefert schließlich einen linearen Klassifikator im Merkmalsraum $H$, der wiederum als nichtlinearer Klassifikator im Eingangsdatenraum $\mathbb{R}^D$ interpretiert werden kann.

Die Entscheidungsfunktion des nichtlinearen Klassifizierungsproblems ist schließlich gegeben durch

$$f(\mathbf{x}) = \operatorname{sgn}\Big( \sum_{n \in S} a_n t_n k(\mathbf{x}_n, \mathbf{x}) + b \Big) \tag{3.56}$$

(vgl. Schölkopf & Smola, 2001, S. 201).

### 3.2.3 Mehrklassen-Klassifizierung

Um das SV-Lernen für Klassifizierungsprobleme mit $m > 2$ Klassen zu verwenden, existieren verschiedene Methoden, bei denen mehrere binäre SVMs kombiniert werden. Ein gängiger Ansatz ist die Methode *One-versus-All* (OVA), bei der $m$ verschiedene SVMs $f^1, \ldots, f^m$ trainiert werden. Die $j$-te SVM wird bestimmt, indem die Daten der Klasse $j$ positiv und die Daten aller übrigen $m-1$ Klassen negativ gelabelt werden. Die $m$ SVMs werden dann kombiniert, indem die Ausgabe aller SVMs vor der Anwendung der Vorzeichenfunktion verglichen wird und der Datenpunkt schließlich der Klasse zugeordnet wird, bei der die Zuordnung *am eindeutigsten* war, d. h. bei der der Ausgabewert maximal ist. Die Zuordnung wird somit gemäß

$$\operatorname*{argmax}_{j=1,\ldots,m} g^j(\mathbf{x})$$

mit

$$g^j(\mathbf{x}) = \sum_{i=1}^{N} t_i a_i^j k(\mathbf{x}_i, \mathbf{x}) + b^j$$

getroffen. Diese Methode hat u. a. den Nachteil, dass die SVMs auf Trainingsdatensätzen mit Klassen von stark unterschiedlicher Größe trainiert werden. Enthält der Datensatz bspw. 10 Klassen mit jeweils 1000 Daten, so besteht die negativ gelabelte Klasse aus 9000 die positiv gelabelte Klasse jedoch nur aus 1000 Daten (vgl. Schölkopf & Smola, 2001, S. 211).
Ein weiterer Ansatz ist die Methode *One-versus-One* (OVO), bei der $m(m-1)/2$ verschiedene binäre SVMs auf allen möglichen Paaren von Klassen trainiert werden. Ein unbekannter Datenpunkt wird von allen $m(m-1)/2$ SVMs einmal klassifiziert. Der Datenpunkt wird dann der Klasse zugeordnet, zu der er am häufigsten zugeteilt wurde. Nachteil dieses Ansatzes ist zum einen, dass wesentlich mehr Berechnungen für die Bestimmung der Klassenzuordnung des Punktes notwendig sind, und zum anderen, dass auch bei dieser Methode die Möglichkeit besteht, dass der Punkt nicht eindeutig einer einzelnen Klasse zugeordnet werden kann, was Abbildung 9 veranschaulicht (vgl. Bishop, 2006, S. 338).
Ein dritter Ansatz basiert auf sogenannten *Error-Correcting-Output-Codes*, einem Fehlerkorrekturverfahren. Die grundlegende Idee ist die Folgende: Genau wie man beim OVA-Ansatz einen binärer Lerner durch Gegenüberstellung einer Klasse gegenüber allen verbleibenden Klassen erhält, können zahlreiche binäre Probleme durch allgemeinere Partitionen der Menge aller Klassen generiert werden. Bspw. kann ein binäres

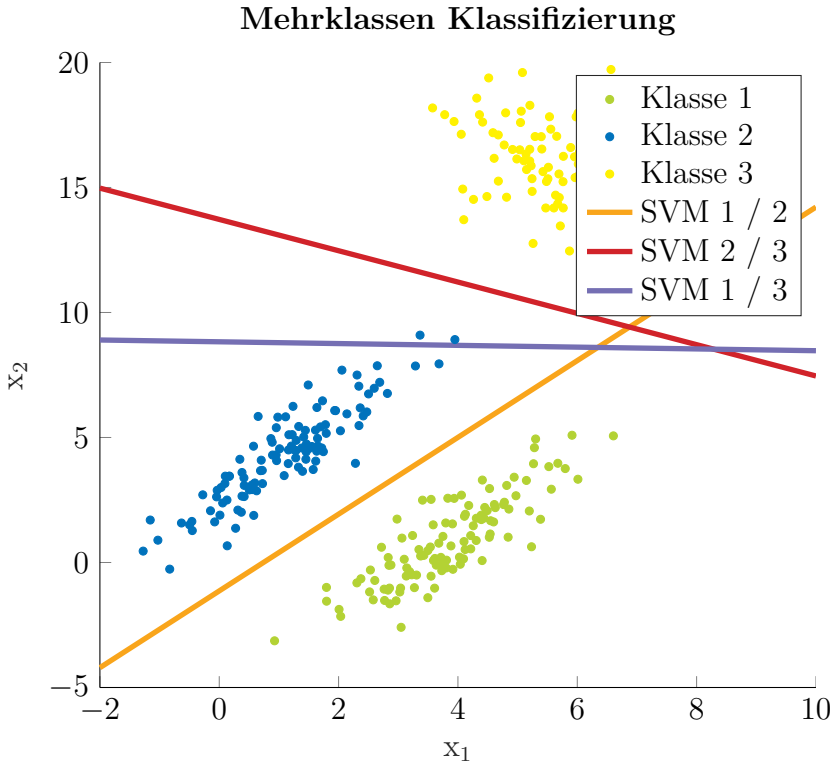


Abbildung 9: Darstellung von drei binären Klassifikatoren berechnet über den One-versus-One-Algorithmus. Punkte, die in dem von den drei Hyperebenen eingeschlossenen Dreieck liegen, haben keine eindeutige Zuordnung zu einer der drei Klassen.

Problem am Beispiel der Klassifizierung handgeschriebener Ziffern von 0 bis 9 durch die Separierung der ungeraden von den geraden Zahlen gewonnen werden. Durch eine geeignete Wahl binärer Klassifikatoren $f^1, \ldots, f^L, L \in \mathbb{N}$ ist es möglich, die Klasse eines Testpunktes vollständig festzulegen. Jede Klasse wird dabei durch einen eindeutigen Vektor $\{\pm 1\}^L$ repräsentiert. Diese Vektoren können für $m$ Klassen zu einer Decoding-Matrix $\{\pm 1\}^{m \times L}$ zusammengefasst werden (vgl. Schölkopf & Smola, 2001, S. 213).
Ist ein gegebener Datensatz *gestört*, so liefert die Klassifizierung eines Testdatenpunktes mit allen $L$ binären Lernern einen Ausgabenvektor $f^1(\mathbf{x}), \ldots, f^L(\mathbf{x})$, der nicht eindeutig einer Zeile der Decoding-Matrix zugeordnet werden kann. Die Zuordnung erfolgt dann zu der Klasse, bei der der Fehler zwischen einer Zeile der Decoding-Matrix und dem Ausgabevektor, gemessen durch den Hamming-Abstand $\Delta(\mathbf{x}, \mathbf{y}) = |\{j \in \{1, \ldots, D\} | x_j \neq y_j\}|$, minimal ist (vgl. Schölkopf & Smola, 2001, S. 213).

## 3.3 Methode 2: Verwendung der Singulärwertzerlegung für Klassifizierungsprobleme

Die zweite Methode nutzt die Singulärwertzerlegung (SVD) (engl.: *Singular Value Decomposition*), eine wichtige Matrixfaktorisierung der linearen Algebra zur Lösung von Klassifizierungsproblemen. Der Grundgedanke bei dieser Lernmethode ist die Betrachtung der Unterräume, die durch die Trainingsdaten einer Klasse, Vektoren des $\mathbb{R}^D$, aufgespannt werden. Die Zuordnung eines unbekannten Testdatenpunktes erfolgt dann zu der Klasse, durch dessen Untervektorraum der Testdatenpunkt *am besten approximiert* werden kann. Wie nachfolgend gezeigt wird, kann die Problemstellung der *besten Approximation* eines Vektors in einen Unterraum auf die Bestimmung der *orthogonalen Projektion* in den Unterraum zurückgeführt werden. Die Verwendung der SVD führt dann zum einen zu einer effizienten Möglichkeit die orthogonale Projektion zu berechnen. Zum anderen ermöglicht sie die Reduktion der Dimension der Unterräume derart, dass diese auf die *wesentlichen Eigenschaften* einer jeden Klasse reduziert werden können.
Die im Folgenden beschriebene Methode eignet sich in der digitalen Bildverarbeitung nicht nur für Klassifizierungen, sondern aufgrund der angesprochenen Dimensionsreduktion insbesondere auch für die Komprimierung von Bildern. Im Bereich der Bildklassifizierung wurde diese Methode neben der Erkennung von handgeschriebener Ziffern u. a. bei der Gesichtserkennung bereits in verschiedenen Projekten erprobt (vgl. Muller et al., 2004, S. 518). Nachfolgend wird schrittweise ein erstes Modell zur Lösung von Klassifizierungsproblemen entwickelt. Da dieses erste Modell angewendet auf verschiedene Datensätze nicht den gewünschten Klassifizierungserfolg liefert, werden anschließend Modellverbesserungen eingebaut.

### 3.3.1 Entwicklung eines ersten mathematischen Modells

Zunächst werden die Trainingsdaten einer Klasse zusammengefasst und die Untervektorräume des $\mathbb{R}^D$, die von den Daten einer jeden Klasse aufgespannt werden, betrachtet. Insgesamt liegen dann $m$ Untervektorräume, was der Anzahl an Klassen entspricht, vor. Bevor die Grundidee hinter der Methode detailliert erläutert wird, versehen wir die in (3.1) beschriebenen Trainingsdaten sowie die zugehörigen Untervektorräume zunächst mit folgender Notation:

$$\begin{aligned} &\text{Daten der Klasse 1: } \{\mathbf{x}_1^1, \ldots, \mathbf{x}_{n_1}^1\} \\ &\text{Daten der Klasse 2: } \{\mathbf{x}_1^2, \ldots, \mathbf{x}_{n_2}^2\} \\ &\qquad\qquad \vdots \\ &\text{Daten der Klasse m: } \{\mathbf{x}_1^m, \ldots, \mathbf{x}_{n_m}^m\} \end{aligned} \tag{3.57}$$

wobei $\sum_{i=1}^{m} n_i = N$.
Die Daten der Klassen 1 bis $m$ spannen dann $m$ Untervektorräume des $\mathbb{R}^D$ auf, die im Folgenden als *Bilddatenräume* bezeichnet werden:

$$W^i = \langle \mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i \rangle \quad \text{für} \quad i = 1, \ldots, m. \tag{3.58}$$

Weiter definieren wir die *Bilddatenmatrizen* $B^i \in \mathbb{R}^{D\times n_i},\ i = 1,\ldots,m$, welche die Eingangsdaten der $i$-ten Klasse spaltenweise enthalten[9]

$$B^i = (\mathbf{x}_1^i \mathbf{x}_2^i \ldots \mathbf{x}_{n_i}^i). \tag{3.59}$$

Es gilt dann

$$W^i = Bild(B^i) \quad \text{für } i = 1,\ldots,m.$$

Auf die Bildklassifizierung handgeschriebener Ziffern von 0 (die Bilder der Ziffer 0 werden im Folgenden der Klasse 10 zugeordnet)[10] bis 9 übertragen, ist der Untervektorraum $W^2$ der Teilraum, der durch sämtliche Trainingsdaten zur Zahl 2 aufgespannt wird, der folglich sämtliche Linearkombinationen aller Trainingsbilder zur Zahl 2 enthält.

### Die beste Approximation eines Testdatenpunktes

Angenommen wir betrachten ein Testbild $\mathbf{b} \in \mathbb{R}^D$, auf dem die Ziffer 1 abgebildet ist. Möchte man dieses durch Kombination von Bildern einer der $m$ Klassen bestmöglich darstellen bzw. approximieren, so würde man intuitiv vermuten, dass dies durch die Bilder der Klasse 1 *am besten* möglich ist. Auf diese Vermutung baut die 2. Klassifizierungsmethode auf.
Die grundlegende Idee ist die Folgende: Für einen beliebigen Testdatenpunkt $\mathbf{b} \in \mathbb{R}^D$ werden die *m besten Darstellungen* als Linearkombination aus sämtlichen Trainingsbildern der Klasse $i$ für $i = 1,\ldots,m$ bestimmt. Das Testbild wird schließlich der Klasse zugeordnet, bei der die beste Darstellung dem tatsächlichen Testbild am nächsten kommt. Dies führt zu der Frage, wie die beste Darstellung bestimmt und mit welchem Maß die Güte dieser Darstellung gemessen werden kann.

Dazu betrachten wir das Ganze in mathematischer Schreibweise:
Für einen beliebigen Testpunkt $\mathbf{b} \in \mathbb{R}^D$ sollen die *besten Approximationen* $\mathbf{b}_{A_1},\ldots,\mathbf{b}_{A_m}$ in jedem der $m$ Untervektorräume $W^1,\ldots,W^m$ bestimmt werden und dann die *Güte* der Approximation gemessen werden. Da die Eingangsdaten aus dem $\mathbb{R}^D$ und damit aus einem euklidischen Vektorraum stammen, stellt der Abstand des Punktes $b$ zu den Approximationen $\mathbf{b}_{A_1},\ldots,\mathbf{b}_{A_m}$ gemessen durch die euklidische Norm ein geeignetes Maß dar. Der Testdatenpunkt wird der Klasse zugeordnet, bei der der Abstand zwischen dem Punkt und seiner besten Approximation in den $m$ Unterräumen minimal ist. Die Klassifizierung wird somit gemäß der *Entscheidungsfunktion*

$$f(\mathbf{b}) = i, \quad \text{wobei } \|\mathbf{b} - \mathbf{b}_{A_i}\|_2 = \min_{j=1,\ldots,m} \|\mathbf{b} - \mathbf{b}_{A_j}\|_2 \tag{3.60}$$

vorgenommen.

[9] An dieser Stelle sei daran erinnert, dass die Eingangsdaten $\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{n_i}^i$ die Bilder der i-ten Klasse repräsentieren.

[10] Dies erwies sich für die konkrete Implementierung des Modells mit der Computersoftware Matlab, deren Indizierung mit 1 beginnt, als sinnvoll.

**Die orthogonale Projektion als beste Approximation**

An dieser Stelle ist zu erörtern, was die beste Approximation eines Vektors, der in diesem Fall ein Bild repräsentiert, in einem Untervektorraum ist, ob eine solche stets existiert und wie diese bestimmt werden kann.

Dazu betrachten wir zunächst nur eine beliebige Klasse $i$ mit zugehöriger Bilddatenmatrix $B = B^i \in \mathbb{R}^{D\times n}$, $n = n_i$.

Die Frage nach der besten Approximation eines Testbildes $\mathbf{b} \in \mathbb{R}^D$ in dem Untervektorraum $W = W^i$ ist dann gleichbedeutend mit der Bestimmung derjenigen Linearkombination $B\mathbf{x}$ mit $\mathbf{x} \in \mathbb{R}^n$ der Spalten von $B$, die einen gegebenen Testdatenpunkt $\mathbf{b}$ bzgl. der euklidischen Norm $\|.\|_2$ am besten approximiert. Man bestimme also das $\mathbf{y}^*$ in

$$W = Bild(B) = \{\mathbf{y} = B\mathbf{x} \ | \mathbf{x} \in \mathbb{R}^n\},$$

für das

$$\|\mathbf{y}^* - \mathbf{b}\|_2 = \min_{\mathbf{y}\in W} \|\mathbf{y} - \mathbf{b}\|_2 = \min_{\mathbf{x}\in\mathbb{R}^n} \|B\mathbf{x} - \mathbf{b}\|_2 \tag{3.61}$$

gilt. Abbildung 10 visualisiert, dass der Abstand $\|B\mathbf{x} - \mathbf{b}\|_2$ genau dann minimal ist, wenn der Differenzvektor $B\mathbf{x} - \mathbf{b}$ senkrecht auf dem Bildraum von $B$ steht.

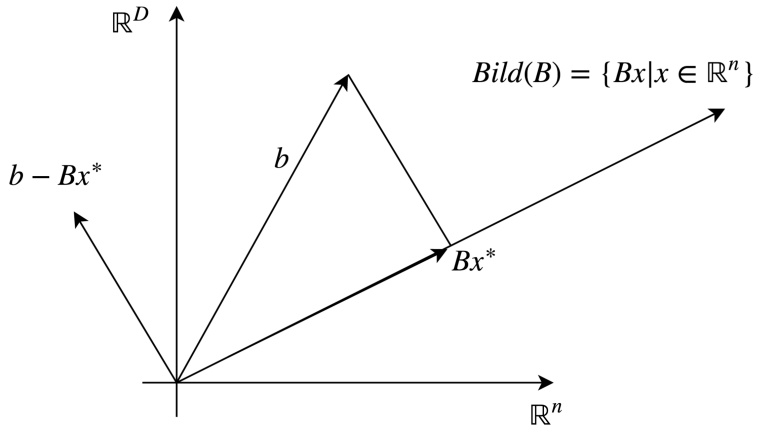


Abbildung 10: Die Lösung des Minimierungsproblems ist durch den Differenzvektor gegeben, der orthogonal auf dem Bild von B steht (in Anlehnung an: Dahmen & Reusken, 2006, S. 123).

Diese von der Anschauung getragene Vermutung wird im $\mathbb{R}^D$ durch den folgenden Satz bestätigt.

**Satz 3.1.** *Sei $W \subset \mathbb{R}^D$ ein endlich-dimensionaler Teilraum des $\mathbb{R}^D$ und $\boldsymbol{b} \in \mathbb{R}^D$. Dann existiert ein eindeutiges $\boldsymbol{y}^* \in W$, das*

$$\|\boldsymbol{y}^* - \boldsymbol{b}\|_2 = \min_{\boldsymbol{y} \in W} \|\boldsymbol{y} - \boldsymbol{b}\|_2 \tag{3.62}$$

*erfüllt. Weiter gilt (3.62) genau dann, wenn der Differenzvektor $(\boldsymbol{y}^* - \boldsymbol{b})$ senkrecht zu $W$ ist. D. h. wenn*

$$(\boldsymbol{y}^* - \boldsymbol{b})^T \boldsymbol{y} = 0 \; \forall \; \boldsymbol{y} \in W \tag{3.63}$$

*gilt (vgl. Dahmen & Reusken, 2006, S. 137).*

*Beweis.* Der Beweis kann bei Dahmen & Reusken (2006, S. 137) nachvollzogen werden. □

Die Lösung des Minimierungsproblems und damit die beste Approximation $\mathbf{b}_A$ ist nach Satz 3.1 durch die orthogonale Projektion von $\mathbf{b}$ auf $W$, bezeichnet mit $\mathbf{b}_W$, gegeben (vgl. Dahmen & Reusken, 2006, S. 137).

Nun ergibt sich die Frage, wie die beste Approximation bzw. die *orthogonale Projektion* des Punktes $\mathbf{b}$ in den Untervektorraum $W = Bild(B)$ berechnet werden kann. Eine Möglichkeit besteht darin, die orthogonale Projektion als Lösung des Minimierungsproblems (3.61) über die Normalgleichungen und unter Verwendung gängiger Verfahren für die Lösung linearer Ausgleichsprobleme, wie bspw. die QR-Zerlegung, zu bestimmen. Dies ist möglich, wenn die Bildmatrizen $B$ vollen Rang haben, was in der Praxis, wenn mit riesigen Datenmengen gearbeitet wird, oft der Fall ist. Diese Herangehensweise wird in Abschnitt 3.3.4 kurz diskutiert. Sie wird jedoch nicht für die weitere Entwicklung des mathematischen Modells gewählt. Stattdessen wird ausgenutzt, dass die Berechnung der orthogonalen Projektion insbesondere dann leicht möglich ist, wenn der Unterraum durch eine *Orthonormalbasis* (ONB) gegeben ist. Dies liefert der folgende Satz:

**Satz 3.2.** *Sei $W \subset \mathbb{R}^D$ ein Teilraum des $\mathbb{R}^D$ und $\{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_l\}$ mit $l = \dim(W)$ eine Orthonormalbasis von $W$. Dann gilt für die orthogonale Projektion $\boldsymbol{b}_W$ eines beliebigen Vektors $\boldsymbol{b} \in \mathbb{R}^D$ in $W$*

$$\boldsymbol{b}_W = \sum_{j=1}^{l} (\boldsymbol{b}^T \boldsymbol{o}_j) \boldsymbol{o}_j \tag{3.64}$$

*(vgl. Dahmen & Reusken, 2006, S. 141).*

*Beweis.*
Offensichtlich liegt $\mathbf{b}_W$ als Linearkombination der Basisvektoren in W. Es bleibt zu zeigen, dass der Differenzvektor $\mathbf{b} - \sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j$ orthogonal auf W steht. Dazu sei $\mathbf{u} = \sum_{i=1}^{l} a_i\mathbf{o}_i \in W$ beliebig.
Dann folgt

$$\begin{aligned}(\mathbf{b} - \sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j)^T\mathbf{u} &= \mathbf{b}^T\mathbf{u} - (\sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j)^T\mathbf{u} = \mathbf{b}^T\mathbf{u} - (\sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j)^T(\sum_{i=1}^{l} a_i\mathbf{o}_i) \\ &= \mathbf{b}^T\mathbf{u} - \sum_{j=1}^{l}\sum_{i=1}^{l} a_i(\mathbf{b}^T\mathbf{o}_j)(\mathbf{o}_j^T\mathbf{o}_i) = \mathbf{b}^T\mathbf{u} - \sum_{j=1}^{l} a_j(\mathbf{b}^T\mathbf{o}_j) \\ &= \mathbf{b}^T\mathbf{u} - \sum_{j=1}^{l}(\mathbf{b}^T a_j\mathbf{o}_j) = \mathbf{b}^T\mathbf{u} - \mathbf{b}^T\sum_{j=1}^{l} a_j\mathbf{o}_j \\ &= \mathbf{b}^T\mathbf{u} - \mathbf{b}^T\mathbf{u} = 0.\end{aligned}$$

Folglich gilt $\sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j \perp W$. Da die orthogonale Projektion nach Satz 3.1 eindeutig ist folgt $\mathbf{b}_W = \sum_{j=1}^{l}(\mathbf{b}^T\mathbf{o}_j)\mathbf{o}_j$. □

Sei nun $O$ die Matrix, welche die orthonormalen Basisvektoren spaltenweise enthält. Dann lässt sich (3.64) schreiben als

$$\mathbf{b}_W = O(O^T\mathbf{b}). \tag{3.65}$$

Bei der Aufgabe der konkreten Berechnung einer ONB $O$ für den Unterraum $W$ kommt nun die Singulärwertzerlegung einer Matrix zum Einsatz, die eben eine solche Basis liefert.

### 3.3.2 Überblick über die Singulärwertzerlegung

Zur Erinnerung wird im Folgenden zunächst ein kurzer Überblick über die Singulärwertzerlegung einer Matrix sowie bedeutende Eigenschaften dieser Zerlegung für die Anwendung in der Bildklassifizierung gegeben. Die Existenz der Zerlegung für jede beliebige Matrix $B \in \mathbb{R}^{m\times n}$ liefert zunächst der folgende Satz.

**Satz 3.3.** ***Die Singulärwertzerlegung***
*Zu einer beliebigen Matrix $B \in \mathbb{R}^{m\times n}$ existieren orthogonale Matrizen $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$ und eine Diagonalmatrix $\Sigma \in \mathbb{R}^{m\times n}$ mit $\Sigma = diag(\sigma_1, \ldots, \sigma_p)$, $p = min\{m, n\}$ mit*

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0,$$

*sodass*

$$U^T BV = \Sigma \tag{3.66}$$

*gilt (vgl. Dahmen & Reusken, 2006, S. 144).*

*Beweis.*
Ein detaillierter Beweis findet sich bei Dahmen & Reusken (2006, S. 144). Nachfolgend wird lediglich die Grundidee einer induktiven Beweisführung dargelegt, aus der jedoch bereits grundlegende Eigenschaften der SVD hervorgehen.

Sei $B \neq 0$ (der Fall $B = 0$ ist trivial).
Dann definiere man $\sigma_1 := \|B\|_2$. Wegen der Definition der Matrixnorm

$$\|B\|_2 := \max_{\mathbf{x}\neq 0} \frac{\|B\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\|\mathbf{x}\|_2=1} \|B\mathbf{x}\|_2$$

existieren dann Vektoren $\mathbf{u}_1 \in \mathbb{R}^m$ und $\mathbf{v}_1 \in \mathbb{R}^n$ mit $\|\mathbf{u}_1\|_2 = 1$ und $\|\mathbf{v}_1\|_2 = 1$, sodass $\sigma_1\mathbf{u}_1 = B\mathbf{v}_1$ gilt, da das Maximum angenommen wird.
Dann lassen sich $\mathbf{u}_1$ und $\mathbf{v}_1$ mit $\tilde{U}_1 \in \mathbb{R}^{m\times(m-1)}$ und $\tilde{V}_1 \in \mathbb{R}^{n\times(n-1)}$ zu orthogonalen Matrizen[11]

$$U_1 := \begin{pmatrix} \mathbf{u}_1 & \tilde{\mathbf{u}}_2 & \dots & \tilde{\mathbf{u}}_m \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1 & \tilde{U}_1 \end{pmatrix} \in \mathbb{R}^{m\times m}$$
$$V_1 := \begin{pmatrix} \mathbf{v}_1 & \tilde{\mathbf{v}}_2 & \dots & \tilde{\mathbf{v}}_n \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 & \tilde{V}_1 \end{pmatrix} \in \mathbb{R}^{n\times n}$$

erweitern, deren Spalten Basen des $\mathbb{R}^m$ bzw. $\mathbb{R}^n$ bilden. Weiter lässt sich zeigen, dass

$$B_1 := U_1^T B V_1 = \begin{pmatrix} \sigma_1 & \mathbf{w}^T \\ 0 & A \end{pmatrix} \in \mathbb{R}^{m\times n}$$

mit $\mathbf{w} \in \mathbb{R}^{n-1}$, $A \in \mathbb{R}^{(m-1)\times(n-1)}$ gilt. Da

$$\|B_1 \begin{pmatrix} \sigma_1 \\ \mathbf{w} \end{pmatrix} \|_2 = \| \begin{pmatrix} \sigma_1^2 + \mathbf{w}^T\mathbf{w} \\ A\mathbf{w} \end{pmatrix} \|_2 \geq \sigma_1^2 + \mathbf{w}^T\mathbf{w} = \| \begin{pmatrix} \sigma_1 \\ \mathbf{w} \end{pmatrix} \|_2^2$$

folgt

$$\sigma_1 = \|B\|_2 = \|B_1\|_2 \geq \frac{\|B_1 \begin{pmatrix} \sigma_1 \\ \mathbf{w} \end{pmatrix} \|_2}{\| \begin{pmatrix} \sigma_1 \\ \mathbf{w} \end{pmatrix} \|_2} \geq \sqrt{\sigma_1^2 + \|\mathbf{w}\|_2^2}$$

und damit

$$\sigma_1^2 \geq \sigma_1^2 + \|\mathbf{w}\|_2^2$$

sodass $\mathbf{w} = 0$ gelten muss.
Dies liefert die Behauptung für $m = 1$ oder $n = 1$. Per Induktion lässt sich dann die Gesamtaussage des Satzes zeigen (vgl. Dahmen & Reusken, 2006, S. 144; Golub & Van Loan, 1996, S. 70). □

Die $\sigma_i,\ i = 1, \dots, p$ heißen *Singulärwerte* von $B$ und die Spalten von $U$ bzw. $V$ die *Links- bzw. Rechtssingulärvektoren.* Die Spalten von $U$ und $V$ werden mit $\mathbf{u}_j,\ j = 1, \dots, m$ bzw. $\mathbf{v}_i,\ i = 1, \dots, n$ bezeichnet. Ist im Folgenden von *Singulärvektoren* die Rede, so sind damit die Linkssingulärvektoren gemeint.

[11] Die Spalten einer orthogonalen Matrix $A \in \mathbb{R}^{n\times n}$ sind orthonormal.

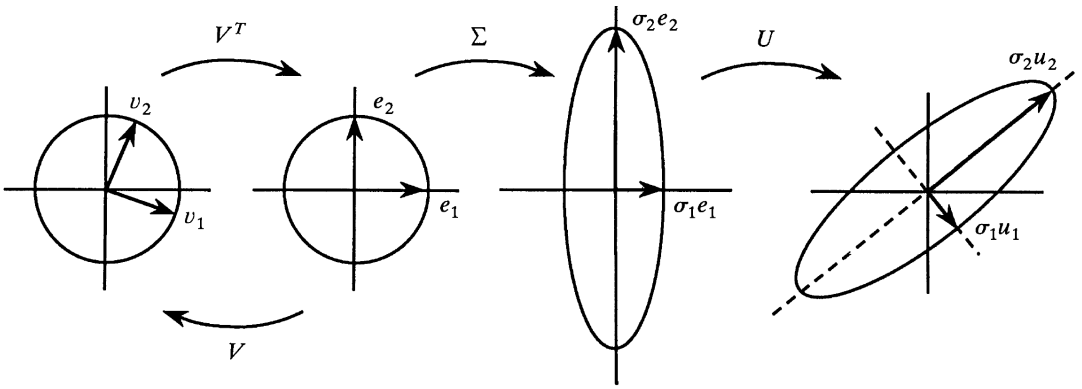


Abbildung 11: Geometrische Interpretation der SVD für $m = n = 2$. Die Abbildung des Einheitskreises unter der Matrixmultiplikation liefert eine Ellipse (entnommen aus: Strang, 1993, S. 854).

**Geometrische Interpretation der SVD**

Geometrisch kann die SVD im Fall $m = n = 2$ wie in Abbildung 11 veranschaulicht werden. Dargestellt ist das Bild des Einheitskreises unter der linearen Abbildung, die durch die Multiplikation mit der (Abbildungs-)Matrix $B$ beschrieben wird. Durch die SVD wird diese Abbildung in drei Teile zerlegt: In eine Rotation oder Spiegelung induziert durch $V^T$, die zu einer Überlagerung von $\mathbf{v}_1$ und $\mathbf{v}_2$ mit den Koordinatenachsen führt,[12] eine Streckung bzw. Stauchung durch $\Sigma$ entlang der Koordinatenachsen und schließlich eine zweite Rotation bzw. Spiegelung durch $U$. Diese zweite Rotation dreht die Basisvektoren $\mathbf{v}_1$ und $\mathbf{v}_2$ so, dass diese auf $\mathbf{u}_1$ bzw. $\mathbf{u}_2$ zu liegen kommen und damit in die gleiche Richtung wie die Hauptachsen der entstandenen Ellipse zeigen. Dies wird zudem ersichtlich, wenn das Bild der Basisvektoren $\mathbf{v}_i$ mit $i = 1, \ldots, n$ unter $B$ betrachtet wird. Dieses kann geschrieben werden als $B\mathbf{v}_i = U\Sigma V^T\mathbf{v}_i = U\Sigma\mathbf{e}_i = U\sigma_i\mathbf{e}_i = \sigma_i\mathbf{u}_i$ mit $\mathbf{e}_i \in \mathbb{R}^n$ dem i-te Einheitsvektor des $\mathbb{R}^n$.
Die geometrische Darstellung verdeutlicht, dass der größte Singulärwert $\sigma_1 = \|B\|_2$ die Länge der *längsten* Hauptachse der Ellipse repräsentiert. Diese Hauptachse weist dabei gerade in die Richtung von $\mathbf{u}_1$, die die *meiste* Information, im Sinne der größten Varianz, über die lineare Transformation beinhaltet. Diese Eigenschaft der SVD wird insbesondere bei der Diskussion der Modellverbesserung in Abschnitt 3.3.3 wieder aufgegriffen (vgl. Muller et al., 2004, S. 521).

Einige wichtige Eigenschaften der SVD, die für die Klassifizierung bzw. die Komprimierung von Bildern von Bedeutung sind und die z. T. bereits aus dem Beweis bzw. der geometrischen Interpretation der SVD hervorgehen, werden im folgenden Lemma zusammengefasst:

[12] Die Multiplikation mit einer orthogonalen Matrix erhält die euklidische Norm. D. h. für eine orthogonale Matrix $A \in \mathbb{R}^{n\times n}$ gilt $\|A\mathbf{x}\|_2 = \|\mathbf{x}\|_2 \ \ \forall\, \mathbf{x} \in \mathbb{R}^n$. Dies verdeutlicht, dass ein beliebiges $\mathbf{x}$ durch $A$ nur gedreht oder gespiegelt werden kann. Weiterhin gilt für eine orthogonale Matrix $A \in \mathbb{R}^{n\times n}$, dass $A^T = A^{-1}$.

**Lemma 3.2.** *Sei $U^T BV = \Sigma$ eine Singulärwertzerlegung von $B$ mit Singulärwerten $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$, $p = min\{m, n\}$. Dann gilt für $i = 1, \ldots, p$*

1. $B\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$.
2. $B^T \boldsymbol{u}_i = \sigma_i \boldsymbol{v}_i$.
3. *Die strikt positiven Singulärwerte von $B$ sind die Wurzeln der strikt positiven Eigenwerte von $B^T B$.*
4. $\|B\|_2 = \sigma_1$.
5. $Bild(B) = \langle \boldsymbol{u}_1, \ldots, \boldsymbol{u}_r \rangle$.
6. $Rang(B) = r$ *(vgl. Dahmen & Reusken, 2006, S. 144).*

*Beweis.*
Sei $U^T BV = \Sigma$ eine SVD von $B$, d. h. $B = U\Sigma V^T$.

1. Siehe Abschnitt zur geometrischen Interpretation der SVD.
2. Sei $i \in \{1, \ldots, p\}$ und $\mathbf{e}_i$ der $i$-te Einheitsvektor des $\mathbb{R}^m$. Dann gilt
$$B^T \mathbf{u}_i = V\Sigma^T U^T \mathbf{u}_i = V\Sigma \mathbf{e}_i = V\sigma_i \mathbf{e}_i = \sigma_i \mathbf{v}_i.$$
3. Es gilt
$$B^T B = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^{-1}$$
und damit sind die Eigenwerte von $\Sigma^T\Sigma$ gerade die Eigenwerte von $B^T B$.[13]
4. Folgt direkt aus dem konstruktiven Beweis zu Satz 3.3.
5. Da $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ eine Basis des $\mathbb{R}^n$ ist, gibt es für jedes $\mathbf{x} \in \mathbb{R}^n$ ein $\mathbf{y} \in \mathbb{R}^n$ mit $\mathbf{x} = V\mathbf{y}$. Sei $\mathbf{e}_i$ der $i$-te Einheitsvektor des $\mathbb{R}^n$. Für das Bild von $B$ gilt dann
$$\begin{aligned} Bild(B) &= \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\} = \{BV\mathbf{y} \mid \mathbf{y} \in \mathbb{R}^n\} \\ &= \{U\Sigma V^T V\mathbf{y} \mid \mathbf{y} \in \mathbb{R}^n\} = \{U\Sigma\mathbf{y} \mid \mathbf{y} \in \mathbb{R}^n\} \\ &= \{U \cdot \sum_{i=1}^{r} \sigma_i y_i \mathbf{e}_i \mid \mathbf{y} \in \mathbb{R}^n\} \\ &= \{\mathbf{u}_1\sigma_1 y_1 + \ldots + \mathbf{u}_r\sigma_r y_r \mid \mathbf{y} \in \mathbb{R}^n\} \\ &= \langle \mathbf{u}_1, \ldots, \mathbf{u}_r \rangle. \end{aligned}$$

[13] An dieser Stelle sei kurz auf die Verwendung der SVD als Kern der Hauptkomponentenanalyse hingewiesen, die u. a. in der multivariaten Statistik Anwendung findet. Aufgrund von Eigenschaft 3. liefert die SVD eine Eigenwertzerlegung für die Kovarianzmatrix $S = \frac{B^T B}{N-1}$ einer Stichprobe (vgl. Hastie et al., 2001, S. 66).

6. Folgt direkt aus Eigenschaft 5., da $Rang(B) = \dim(Bild(B)) = r$. □

Die SVD liefert somit eine ONB $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ für den Bildraum einer Matrix, die aus den ersten $r$ Singulärvektoren besteht, wobei $r$ dem Rang der Matrix entspricht (vgl. Lemma 3.2 Eigenschaften 5. und 6.). Im Folgenden wird die Matrix, welche die orthonormalen Basisvektoren $\mathbf{u}_1, \ldots, \mathbf{u}_r$ spaltenweise enthält als *Basismatrix* $O$ bezeichnet.

Um nun die Klassifizierung eines unbekannten Punktes $\mathbf{b}$ anhand des bisher entwickelten Modells vorzunehmen, sind folgende Schritte zu absolvieren:

**1. Schritt:** Berechnung der ONBs für alle Bildunterräume $W^1, \ldots, W^m$ mithilfe der SVD und speichern der Basismatrizen $O^1, \ldots, O^m$, welche die zugehörigen orthonormalen Basisvektoren spaltenweise enthalten.
**2. Schritt:** Bestimmung der $m$ orthogonalen Projektionen $\mathbf{b}_{W^1}, \ldots, \mathbf{b}_{W^m}$ in jeden der Untervektorräume $W^1, \ldots, W^m$ gemäß (3.65).
**3. Schritt:** Berechnung des Abstands von $\mathbf{b}$ zu allen $m$ Projektionen.
Diese Berechnung lässt sich vereinfachen zu:

$$\begin{aligned}
\|\mathbf{b} - \mathbf{b}_W\|_2^2 &= \|\mathbf{b} \ - OO^T\mathbf{b}\|_2^2 \\
&= (\mathbf{b} - OO^T\mathbf{b})^T(\mathbf{b} - OO^T\mathbf{b}) \\
&= \mathbf{b}^T\mathbf{b} - 2\mathbf{b}^TOO^T\mathbf{b} + (OO^T\mathbf{b})^T(OO^T\mathbf{b}) \\
&= \mathbf{b}^T\mathbf{b} - 2\mathbf{b}^TOO^T\mathbf{b} + \mathbf{b}^TO(O^TO)O^T\mathbf{b} \\
&= \mathbf{b}^T\mathbf{b} - 2\mathbf{b}^TOO^T\mathbf{b} + \mathbf{b}^TOO^T\mathbf{b} \\
&= \mathbf{b}^T\mathbf{b} - \mathbf{b}^TOO^T\mathbf{b} \\
&= \mathbf{b}^T\mathbf{b} - (O^T\mathbf{b})^T(O^T\mathbf{b}) \\
&= \|\mathbf{b}\|_2^2 - \|O^T\mathbf{b}\|_2^2 \qquad (3.67)
\end{aligned}$$

**4. Schritt:** Zuordnung des Testdatenpunktes $\mathbf{b}$ zu der Klasse, bei der der Abstand von $\mathbf{b}$ zum Bildunterraum $W^i$ minimal ist bzw. (3.67) berücksichtigend, für dessen Basismatrix $O^i$ die Norm $\|(O^i)^T\mathbf{b}\|_2^2$ maximal ist. Die Entscheidungsfunktion (3.60) ist damit gegeben durch

$$f(\mathbf{b}) = i, \ \text{ wobei } \ \|(O^i)^T\mathbf{b}\|_2^2 = \max_{j=1,\ldots,m} \|(O^j)^T\mathbf{b}\|_2^2. \qquad (3.68)$$

Klassifiziert man mit dem so entwickelten Modell den MNIST Datensatz, so liefert dies einen denkbar schlechten Klassifizierungserfolg auf dem Testdatensatz. Dies wird in Kapitel 4 ausführlich diskutiert.
Nachfolgend werden Gründe für den schlechten Klassifizierungserfolg erörtert und Modellverbesserungen diskutiert und implementiert.

### 3.3.3 Modellverbesserungen

Bisher wurde angenommen, dass die Bildräume $W^i$ zu den Bildmatrizen $B^i \in \mathbb{R}^{D \times n_i}$ für $i = 1, \ldots, m$ **echte** Teilräume des $\mathbb{R}^D$ sind, dass also insbesondere

$$D > r_i = Rang(B^i) = \dim(Bild(B^i)) = \dim(W^i).$$

gilt. Dies trifft bei Datensätzen zu, bei denen

- die Anzahl der Trainingsbilder pro Klasse kleiner ist als die Anzahl der Pixel, d. h. es gilt $D > n_i$ für $i = 1, \ldots, m$ oder
- die Trainingsbilder sich so *ähnlich* sind, dass die Anzahl linear unabhängiger Bildvektoren einer Klasse klein und insbesondere kleiner als $D$ ist, womit wiederum $D > \dim(W^i)$ gilt.

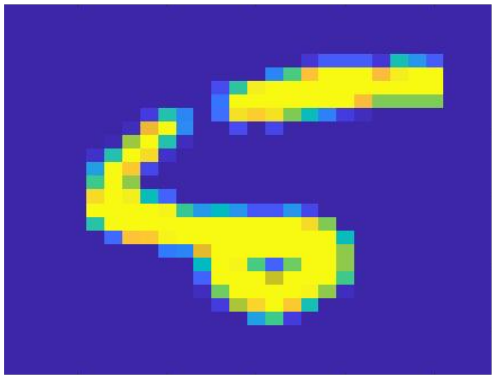

Abbildung 12: Bild einer undeutlich geschriebenen Ziffer 5. Sämtliche schwarz-weiß Bilder werden in der vorliegenden Arbeit mit einer Skala dargestellt, bei dem ein weißer Pixel im Originalbild dem Farbwert blau entspricht und ein schwarzer Pixel dem Farbwert gelb. Das Ziel dieser Darstellung ist eine leichtere visuelle Erfassbarkeit von Unterschieden in den Bildern.

In der Anwendung liegen jedoch zum einen häufig Datensätze mit deutlich mehr Trainingsbildern als Pixeln vor, d. h. $n_i > D$, und zum anderen sind die *Unterschiede* zwischen den Bildern einer Klasse meist so groß, dass die Vektoren der Trainingsbilder einer Klasse den gesamten $\mathbb{R}^D$ aufspannen, und somit $\dim(W^i) = D$ gilt. Jedes beliebige Testbild $\mathbf{b}$ liegt dann in mehreren oder sogar jedem der Bildräume $W^i$, $i = 1, \ldots, m$ und die Klassifizierung über die Entscheidungsfunktion (3.68) kann nicht sinnvoll vorgenommen werden, da $\mathbf{b} = \mathbf{b}_{W^i}$ und damit $\|\mathbf{b} - \mathbf{b}_{W^i}\| = 0$ für mehrere $i = 1, \ldots, m$ gilt.
Ist der Rang von $B^i$ und damit die Dimension des Bildraumes unwesentlich kleiner als $D$, so werden zwar nicht mehr alle beliebigen Testdatenpunkte im Bildraum von

$B^i$ liegen, sie werden sich jedoch gut durch diesen approximieren lassen. Der schlechte Klassifizierungserfolg lässt sich in diesem Fall durch die Tatsache begründen, dass weiterhin nicht nur die *markantesten* Eigenschaften, welche die Bilder einer Klasse auszeichnen, durch die ONB des Bildraumes dargestellt werden, sondern auch *unwichtige* Eigenschaften, d. h. am Beispiel der handgeschriebenen Ziffern, auch solche Eigenschaften, die nur bei unsauber geschriebenen Zahlen auftreten (vgl. Abbildung 12). Dies führt z. B. dazu, dass auch ein Testbild mit der Ziffer 3 durch die Basis der Klasse 1 sehr gut approximiert werden kann (vgl. Abbildung 13).

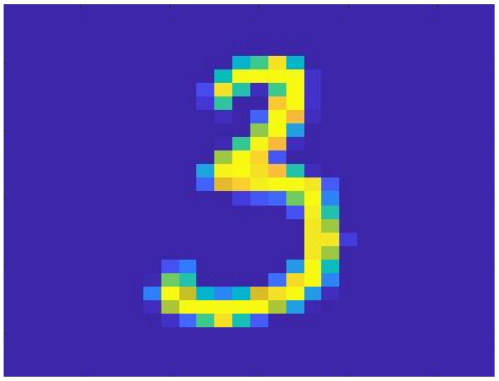

Abbildung 13: Approximation eines Bildes der Klasse 3 durch die ONB zur Klasse 1. Anders ausgedrückt ist die orthogonale Projektion eines Bildes der Klasse 3 in den Unterraum der Klasse 1 dargestellt.

**Niedrigrang-Approximation**

Um den Klassifizierungserfolg zu erhöhen, stellt sich nun die Frage, wie die Bildmatrix $B$ durch eine Matrix $\tilde{B}$ mit $Rang(\tilde{B}) = s < r$ approximiert werden kann, sodass sämtliche *wesentliche* Merkmale der Bildklasse erhalten bleiben, unwichtige Eigenschaften jedoch nicht berücksichtigt werden. Anders ausgedrückt: Welche der Basisvektoren der ONB $O$ sollten gewählt werden, um die bestmögliche Approximation mit Dimension $s < r$ des Bildraumes zu erhalten?
Dazu betrachten wir zunächst die Bildmatrix als Summe von Rang 1 Matrizen:

$$B = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \ldots + \sigma_n \mathbf{u}_n \mathbf{v}_n^T. \tag{3.69}$$

Diese Summe kann vereinfacht werden, indem Summanden mit Singulärwerten gleich null verworfen werden. Dies liefert

$$B = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \ldots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T, \tag{3.70}$$

wobei $r$ der Anzahl Singulärwerte ungleich null und damit dem Rang von $B$ entspricht. Wie verfährt man jedoch mit Singulärwerten nahe null? Zwar haben die Bildmatrizen häufig sehr hohe Ränge, der *effektive Rang* der Matrizen ist jedoch meist niedrig. D. h., dass viele der Singulärwerte sehr klein sind, sie kaum einen Beitrag zu der Summe (3.70) leisten und gleich null gesetzt werden können (vgl. Strang, 2016, S. 367).
Dies führt zu der folgenden Rang $s$ Approximation

$$B_s = \sum_{i=1}^{s} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad \text{mit} \quad s \leq r, \tag{3.71}$$

die die beste Rang $s$ Approximation der Bildmatrix $B$ darstellt, wie der folgende Satz bestätigt (vgl. Muller et al., 2004, S. 522).

**Satz 3.4.** *Sei $U^T B V = \Sigma$ eine Singulärwertzerlegung von $B \in \mathbb{R}^{m \times n}$ mit Singulärwerten $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$, $p = min\{m, n\}$.*
*Die beste Approximation für $B$ vom Rang kleiner gleich $s$ für $0 \leq s \leq p-1$ ist gegeben durch*

$$B_s := \sum_{i=1}^{s} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \in \mathbb{R}^{m \times n}. \tag{3.72}$$

*Weiterhin gilt*

$$\|B - B_s\|_2 = \sigma_{s+1}. \tag{3.73}$$

*Beweis.*
1. Fall: Sei $s = 0$. Dann gilt

$$\|B - B_s\|_2 = \|B\|_2 = \sigma_1 \tag{3.74}$$

was direkt aus der Eigenschaft 4. in Lemma 3.2 folgt.
2. Fall: Sei nun $1 \leq s < r$. Für $B_s$ definiert wie oben gilt dann:

$$\begin{aligned} \|B - B_s\|_2 &= \|U^T (B - B_s) V\|_2 \\ &= \|diag(\sigma_1, \ldots, \sigma_r, 0, \ldots) - diag(\sigma_1, \ldots, \sigma_s, 0, \ldots)\|_2 \\ &= \|diag(0, \ldots, 0, \sigma_{s+1}, \ldots, \sigma_r, 0, \ldots)\|_2 \\ &= \sigma_{s+1} \end{aligned} \tag{3.75}$$

Es bleibt zu zeigen, dass für eine beliebige Matrix $A \in \mathbb{R}^{m \times n}$ mit $Rang(A) \leq s$

$$\|B - B_s\|_2 \leq \|B - A\|_2$$

gilt. Sei also $A \in \mathbb{R}^{m \times n}$ mit $Rang(A) \leq s$.
Dann ist $\dim(Kern(A)) = n - \dim(Bild(A)) = n - Rang(A) \geq n - s$ und somit

$$Kern(A) \cap \langle v_1, \ldots, v_{s+1} \rangle \neq \{0\}$$

Sei $\mathbf{v} = \sum_{i=1}^{s+1} a_i \mathbf{v}_i$ mit $\|\mathbf{v}\|_2 = 1$ ein Vektor aus diesem Durchschnitt. Dann gilt $A\mathbf{v} = 0$ und $1 = \|\mathbf{v}\|_2^2 = \sum_{i=1}^{s+1} a_i^2$. Damit folgt

$$\begin{aligned} \|B - A\|_2^2 \geq \|(B - A)\mathbf{v}\|_2^2 = \|B\mathbf{v}\|_2^2 = \|\sum_{i=1}^{s+1} a_i \sigma_i \mathbf{u}_i\|_2^2 \\ = \sum_{i=1}^{s+1} a_i^2 \sigma_i^2 \geq \sigma_{s+1}^2 \sum_{i=1}^{s+1} a_i^2 = \sigma_{s+1}^2 = \|B - B_s\|_2^2. \end{aligned} \tag{3.76}$$

3. Fall: Sei $r \leq s \leq p - 1$. Dann wähle man $B_s = B$ und es folgt

$$\|B - B_s\|_2 = 0 = \sigma_{s+1}. \tag{3.77}$$

Damit folgt die Behauptung für $0 \leq s \leq p - 1$ (vgl. Dahmen & Reusken, 2006, S. 150). □

Gemäß Satz 3.4 liefert die Wahl der ersten $s$ Singulärvektoren die beste Approximation für die Bildmatrix $B$ vom Rang kleiner oder gleich $s$. Anders ausgedrückt ist der Informationsgehalt der Singulärvektoren über die wesentlichen Eigenschaften einer Bildklasse absteigend. Die strukturierte Bestimmung einer *guten* Niedrigrang-Approximation mit dem Ziel, die Datenmatrix auf wesentliche Informationen zu reduzieren, bzw. mit anderen Worten die theoretische Untersuchung der Anzahl an Singulärvektoren, die zur Beschreibung einer Bildklasse beibehalten werden sollte, stellt eine inverse Problemstellung dar. Derartige Problemstellungen werden in der Theorie der inversen Probleme ausführlich diskutiert und behandelt (vgl. Chu & Golub, 2007, S. 246).
Auch vom numerischen Standpunkt ist das Verwerfen sehr kleiner Singulärwerte und der zugehörigen Singulärvektoren sinnvoll, da es bei der expliziten Berechnung der SVD mit dem Computer zu Rundungsfehlern aufgrund der Maschinengenauigkeit kommt. Wird der Rang mit dem Computer berechnet, so gilt für die mit Rundungsfehlern behafteten Annäherung $\tilde{B} \in \mathbb{R}^{D \times n}$ von $B$ meist

$$Rang(\tilde{B}) > r = Rang(B) \tag{3.78}$$

(vgl. Dahmen & Reusken, 2006, S. 151).
Neben der Anwendung in der Bildklassifizierung ist die Approximierung der Matrizen von Bildern auch mit Blick auf eine ökonomische Datenspeicherung interessant, da die Bilder mithilfe ihrer besten Rang $s$ Approximation komprimiert gespeichert werden können.

Zusammengefasst kann man dem Problem der Überanpassung des mathematischen Modells an den konkreten Trainingsdatensatz folglich mit der Wahl der *informationsreichsten* Basisvektoren $\mathbf{u}_1, \ldots, \mathbf{u}_s$ begegnen. Wie viele Basisvektoren beibehalten werden hängt dabei stark von dem jeweiligen Datensatz ab. Die experimentell bestimmte *optimale* Anzahl an Singulärvektoren für den MNIST Datensatz wird in Kapitel 4 diskutiert.

### 3.3.4 Exkurs: Lineare Gleichungssysteme bzw. Ausgeichsprobleme

Die dieser Lernmethode zugrundeliegende Idee der Bestimmung der *besten Approximation* eines Vektors in einem Unterraum kann auch aus einem anderen Blickwinkel beleuchtet werden, was in diesem Abschnitt kurz umrissen werden soll.

Das Minimierungsproblem (3.61) lässt sich mit dem Fokus auf der Interpretation als lineares Ausgleichsproblem bzw. lineares Gleichungssystem betrachten. Dazu ist wie folgt zu unterscheiden:
In dem Fall, dass der Rang der Bildmatrix $B$ voll und zudem die Zahl der Trainingsbilder $n$ einer Klasse kleiner als die Anzahl der Pixel $D$ ist (d. h. es gilt $n < D$), stellt das Minimierungsproblem (3.61) ein lineares Ausgleichsproblem dar, welches sich auf die Lösung des Systems der sog. *Normalgleichungen*

$$B^T B\mathbf{x} = B^T\mathbf{b} \tag{3.79}$$

reduzieren lässt. Dieses ist äquivalent zu der Orthogonalitätsrelation aus Satz 3.1:

$$\begin{aligned} B\mathbf{x} - \mathbf{b} \perp Bild(B) &\iff \mathbf{y}^T(B\mathbf{x} - b) = 0 \quad \forall\, \mathbf{y} \in Bild(B) \\ &\iff (B\mathbf{v})^T(B\mathbf{x} - \mathbf{b}) = 0 \quad \forall\, \mathbf{v} \in \mathbb{R}^D \\ &\iff \mathbf{v}^T(B^T B\mathbf{x} - B^T\mathbf{b}) = 0 \quad \forall\, \mathbf{v} \in \mathbb{R}^D \\ &\iff B^T B\mathbf{x} - B^T\mathbf{b} = 0. \end{aligned} \tag{3.80}$$

(vgl. Dahmen & Reusken, 2006, S.123). Eine Lösung $\mathbf{x}^*$ des linearen Ausgleichsproblems lässt sich in diesem Fall über die Cholesky- oder die QR-Zerlegung und damit auch ohne den Umweg über die SVD lösen. Die Lösung nimmt eine Darstellung der Form

$$\mathbf{x}^* = (B^T B)^{-1} B^T\mathbf{b}$$

an.
In dem Fall, dass der Rang der Bildmatrizen nicht voll ist oder $D < n$ gilt, kann die Lösung des Minimierungsproblems über die Pseudoinverse der Matrix $B$ bestimmt werden, deren Berechnung wiederum über die SVD möglich ist (vgl. Dahmen & Reusken, 2006, S.145).
Die hier beschriebenen Möglichkeiten der Lösung des Minimierungsproblems (3.61) wurden nicht gewählt, da zum einen der numerische wie auch der effektive Rang der Matrizen unberücksichtigt bliebe, und da zum anderen die Rangapproximation der Bildmatrizen nicht durchgeführt würde. Dies würde bei verschiedenen Datensätzen wie u. a. MNIST zu der Existenz einer eindeutigen Lösung des Gleichungssystems $B^i\mathbf{x} = \mathbf{b}$ für verschiedene Bildmatrizen $B^i$ führen, womit keine eindeutige Klassenzuordnung vorgenommen werden könnte.

# 4 Anwendung in der Bildklassifizierung

Nachfolgend werden die beiden beschriebenen Methoden zunächst auf den MNIST Datensatz angewendet. Der Klassifizierungserfolg wird abhängig von verschiedenen Parameterwahlen der mathematischen Modelle beider Methoden ermittelt und diskutiert. Das Ziel des experimentellen Teils dieser Arbeit ist es nicht, die Modelle mit Blick auf den Klassifizierungserfolg bestmöglich zu optimieren, sondern vielmehr, die dargelegten mathematischen Hintergründe anhand der Anwendung zu veranschaulichen und den Einfluss verschiedener Parameter auf die Effizienz beider Modelle zu untersuchen. Anschließend werden beide Methoden auf zwei Datensätze aus dem Bereich der Gesichtsklassifizierung angewendet, mit dem Ziel einen Kontext in den Blick zu nehmen, der *alltagsnah* und *interessant* für Schüler ist. Die Anwendung auf diese Datensätze dient lediglich der Anschauung. Eine detaillierte Analyse der Ergebnisse erfolgt nicht. Sämtliche Experimente wurden mit der Computersoftware Matlab[14] auf einem MacBook mit einem 2,7 GHz Intel Core i5 Prozessor durchgeführt.[15]

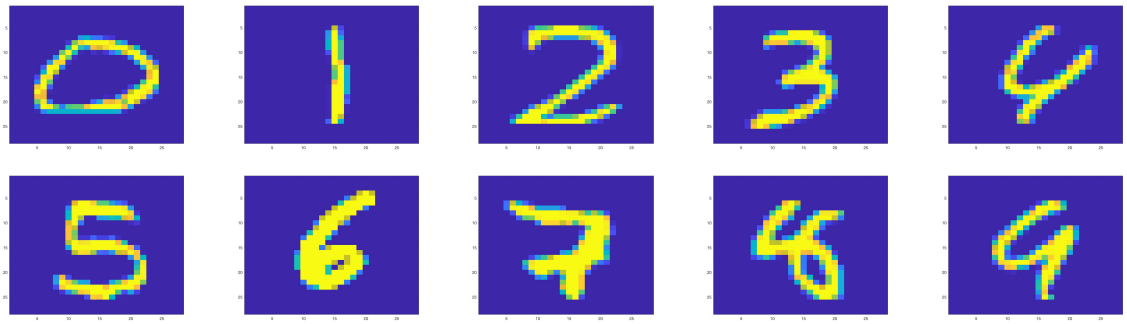

Abbildung 14: Beispielbilder der handgeschriebenen Ziffern von 0 bis 9 des MNIST Datensatzes.

## 4.1 Klassifizierung handgeschriebener Ziffern

Der MNIST Datensatz besteht aus einem Trainingsdatensatz mit 60000 Bildern und einem Testdatensatz mit 10000 Bildern. Die Bilder zeigen die handgeschriebenen Ziffern von 0 bis 9 und haben eine Größe von $28 \times 28$ Pixeln. Die Pixel sind im originalen Datensatz zeilenweise angeordnet und nehmen Werte zwischen 0 und 255 an. Ein Wert von 0 (weiß) entspricht dem Hintergrund und ein Wert von 255 (schwarz) dem Vordergrund. Die Grauwerte der Pixel wurden auf das Intervall $[0, 1]$ normiert. Durch die Berechnung des Massenzentrums der Pixel wurden die Bilder der handgeschriebenen Ziffern zentriert (vgl. LeCun & Cortes, 2010). Zu jeder Zahl ist in Abbildung 14 ein Bild dargestellt. Die Zahlen wurden zum Teil sehr undeutlich geschrieben, wie die Bilder zu den Ziffern 4 und 8 verdeutlichen (vgl. Abbildung 14). Zu jeder Klasse des

[14] `www.mathworks.com/`, Stand: 07.02.2018

[15] Die in Matlab implementierten Codes sind zu finden unter `https://git.rwth-aachen.de/sarah.schoenbrodt/masterarbeit_sarah_schoenbrodt.git`

Trainingsdatensatzes liegen zwischen 5400 und 6750 Bilder vor, zu den Klassen des Testdatensatzes zwischen 890 und 1150 Bilder (vgl. Tabelle 1). Die Zuordnung der Bilder zu den entsprechenden Klassen ist durch einen Vektor gegeben, dessen Einträge die Labels von 0 bis 9 beinhalten.

Tabelle 1: Anzahl der Trainingsbilder (TB) und Testbilder (TE) pro Klasse.

| **Klasse** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **0** |
|---|---|---|---|---|---|---|---|---|---|---|
| **TB** | 6742 | 5958 | 6131 | 5842 | 5421 | 5918 | 6265 | 5851 | 5949 | 5923 |
| **TE** | 1135 | 1032 | 1010 | 982 | 892 | 958 | 1028 | 974 | 1009 | 980 |

### 4.1.1 Anwendung der SVM auf den MNIST Datensatz

Bei der Anwendung der SVM auf den MNIST Datensatz können zahlreiche Kombinationen verschiedener Parameter gewählt und diskutiert werden. Sei es die Wahl der Kernfunktion, die ihrerseits wieder unterschiedliche Parameterwahlen erfordert, der Algorithmus zum Lösen des Optimierungsproblems oder unterschiedliche Mehrklassenalgorithmen, um nur einige mögliche Einflussgrößen zu nennen. Da eine ausführliche Analyse, Optimierung und Kombination sämtlicher Parameter den Rahmen dieser Arbeit überschreiten würde, wurden lediglich ausgewählte Parameter variiert und deren Einfluss auf den Klassifizierungserfolg untersucht.
Die Lösung des quadratischen Optimierungsproblems (3.53) wurde bei allen nachfolgend beschriebenen Experimenten über den iterativen *Sequential Minimal Optimization*-Algorithmus gewonnen (vgl. Platt, 1998). Zudem wurden die binären SVMs bei allen bis auf einem Vergleichsexperiment mit dem OVO Algorithmus trainiert bzw. kombiniert.
Ein Parameter, der bei den durchgeführten Experimenten variiert und dessen Einfluss auf den Klassifizierungserfolg nachfolgend diskutiert wird, ist der Kostenfaktor $C$, der den Slack kontrolliert und über den die Über- bzw. Unteranpassung der SVMs an die gegeben Daten reguliert werden kann. Des Weiteren wurde der Klassifizierungserfolg mit standardisierten sowie nicht-standardisierten Daten[16] durchgeführt.

**Erstes Experiment: Linearer Kern ohne vorherige Standardisierung der Daten**
Bei dem ersten Experiment wurde das Support Vektor Lernen mit einem linearen Kern durchgeführt, wobei die Daten zuvor nicht standardisiert wurden.
Der Klassifizierungserfolg auf den Testdaten über eine Hard Margin Klassifizierung, d.h. es wurde kein Slack zugelassen, betrug 85.28%. Der maximale Klassifizierungserfolg bei der Soft Margin Klassifizierung betrug fast 95%. Dieser wurde mit Werten zwischen $10^{-2}$ und $10^{-1}$ für den Kostenfaktor $C$ erreicht. Für verschiedene weitere

[16] Die Matrix $C$, welche die Trainingsdaten der Klassen zeilenweise enthält, wurde wie in Kapitel 2 beschrieben skaliert, sodass die Spalten einen standardisierten Mittelwert von 0 und eine Varianz von 1 aufweisen.

Werte des Kostenfaktors $C$ ist der Klassifizierungserfolg bei der Soft Margin Klassifizierung in Tabelle 2 aufgeführt.

Tabelle 2: Soft Margin Klassifizierung: Prozentualer Klassifizierungserfolg (KE) auf den Testdaten und Trainingsgenauigkeit (TG) auf den Trainingsdaten. Es wurde ein linearer Kern verwendet. Die Daten wurden nicht standardisiert (TG Hard Margin: 90.42%).

| **C** | $\mathbf{10^{-7}}$ | $\mathbf{10^{-5}}$ | $\mathbf{10^{-3}}$ | $\mathbf{10^{-2}}$ | $\mathbf{10^{-1}}$ | **1** | **10** | $\mathbf{10^{2}}$ | $\mathbf{10^{3}}$ |
|---|---|---|---|---|---|---|---|---|---|
| **KE** | 11.35 | 73.82 | 92.63 | 94.48 | 94.95 | 94.38 | 93.72 | 93.22 | 85.28 |
| **TG** | 11.25 | 73.16 | 92.35 | 94.61 | 96.15 | 97.36 | 98.20 | 98.57 | 90.84 |

Die Ergebnisse unterstreichen den in Kapitel 3.2.1 beschriebenen Einfluss der Wahl des Kostenfaktors $C$: Für sehr kleine Werte von $C$ wird so viel Slack erlaubt, dass lediglich 11.35% der Testdaten korrekt klassifiziert werden. Dieses Ergebnis erschließt sich auch durch die Betrachtung der Klassengrößen des Testdatensatzes (vgl. Tabelle 1).
Die Anzahl der Testbilder der größten Klasse beträgt 1135. Da der gesamte Testdatensatz 10000 Bilder enthält, entspricht der Klassifizierungserfolg von 11.35% damit gerade dem Klassifizierungserfolg, der bei einer Klassifizierung mit dem Zufall zu erwarten wäre. Zudem führen kleine Werte für $C$ zu einer Unteranpassung der SVMs an die Trainingsdaten und damit zu einem hohen Trainingsfehler. Lediglich 11.25% der Trainingsbilder wurden mit den trainierten SVMs richtig klassifiziert.
Werden hingegen sehr große Werte für $C$ gewählt ($\sim C = 10^3$), so nimmt der Klassifizierungserfolg auf dem Testdatensatz ebenfalls ab, da eine Überanpassung der SVMs an die gegeben Trainingsdaten erfolgt. Bereits ab $C = 1000$ liefert die Soft Margin Klassifizierung den gleichen Klassifizierungserfolg wie die Hard Margin Klassifizierung. Die starke Anpassung des gelernten Modells an die Trainingsdaten für große Werte von $C$ geht auch aus dem Trainingsfehler hervor, der für Werte um $C = 100$ nur knapp 1.5% beträgt. Der Fehler auf den Trainingsdaten beträgt bei der Hard Margin Klassifizierung rund 10%, was verdeutlicht, dass die Trainingsdaten nicht exakt mit linearen Klassifikatoren separierbar sind. Die Verteilungen der Klassen überlappen sich folglich.

**Zweites Experiment: Linearer Kern mit vorheriger Standardisierung der Daten**
Beim zweiten Experiment wurden die Daten zunächst standardisiert und das Support Vektor Lernen anschließend mit dem linearen Kern durchgeführt. Bei der Hard Margin Klassifizierung betrug der Klassifizierungserfolg 85%. Die Ergebnisse der Soft Margin Klassifizierung sind in Tabelle 3 aufgeführt. Der maximale Klassifizierungserfolg wurde mit $C = 10^{-2}$ erreicht und lag bei 94.81%. Die Ergebnisse des zweiten Experiments weichen nur leicht von den Ergebnissen ab, die ohne vorherige Standardisierung der Daten erzielt werden konnten. Ein größerer Einfluss der Standardisierung auf den Klassifizierungserfolg wäre u. a. dann zu erwarten, wenn die Pixelwerte der Bilder einer Klasse stark variieren würden, was bspw. auftreten würde, wenn die Bilder eines Datensatzes unter variierenden Lichtverhältnissen aufgenommen werden.

Tabelle 3: Soft Margin Klassifizierung: Prozentualer Klassifizierungserfolg (KE) auf den Testdaten. Es wurde ein linearer Kern verwendet. Die Daten wurden zuvor standardisiert.

| **C** | $\mathbf{10^{-7}}$ | $\mathbf{10^{-5}}$ | $\mathbf{10^{-3}}$ | $\mathbf{10^{-2}}$ | $\mathbf{10^{-1}}$ | **1** | **10** | $\mathbf{10^{2}}$ | $\mathbf{10^{3}}$ |
|---|---|---|---|---|---|---|---|---|---|
| **KE** | 11.35 | 88.82 | 94.44 | 94.81 | 94.26 | 93.63 | 93.04 | 87.79 | 85.00 |

**Drittes Experiment: Vergleich des Klassifizierungserfolgs bei OVO und OVA**
Der Vergleich der Ergebnisse, die mit dem OVO bzw. dem OVA Algorithmus bei Verwendung eines linearen Kerns erhalten wurden, verdeutlichen, dass ein höherer Klassifizierungserfolg mit dem OVO Algorithmus erzielt werden konnte. Zudem wird ersichtlich, dass die Rechenzeiten für sämtliche Wahlen des Kostenfaktors $C$ mit dem OVA Algorithmus deutlich über denen des OVO Algorithmus lagen (vgl. Tabelle 4). Zwar müssen mit dem OVO Algorithmus 45 und mit dem OVA Algorithmus lediglich 10 SVMs trainiert werden, jedoch ist die Lösung des Optimierungsproblems beim OVA deutlich zeitaufwändiger. Der Grund dafür ist, dass bei jedem Support Vektor Training mit dem OVA ca. 6000 negative und 54000 positiv gelabelte Trainingsbeispiele berücksichtigt werden müssen. Bei dem OVO Algorithmus hingegen werden die SVMs nur mit jeweils ca. 6000 positiven und 6000 negativen Beispielen trainiert.

Tabelle 4: Soft Margin Klassifizierung: Prozentualer Klassifizierungserfolg (KE) auf den Testdaten und Rechenzeit in Sekunden mit OVO und OVA. Es wurde ein linearer Kern verwendet. Die Daten wurden nicht standardisiert.

| **C** | $\mathbf{10^{-5}}$ | $\mathbf{10^{-3}}$ | $\mathbf{10^{-1}}$ | **10** | $\mathbf{10^{3}}$ |
|---|---|---|---|---|---|
| **KE (OVO)** | 73.82 | 92.63 | 94.95 | 93.72 | 87.28 |
| **Rechenzeit (OVO)** | $1.87 \cdot 10^3$ | $2.85 \cdot 10^2$ | $2.10 \cdot 10^2$ | $3.36 \cdot 10^3$ | $3.55 \cdot 10^3$ |
| **KE (OVA)** | 79.92 | 90.35 | 92.21 | 92.14 | 37.24 |
| **Rechenzeit (OVA)** | $4.56 \cdot 10^3$ | $1.91 \cdot 10^3$ | $2.73 \cdot 10^3$ | $4.52 \cdot 10^4$ | $1.83 \cdot 10^5$ |

**Fazit**
Die durchgeführten Experimente verdeutlichen, dass bereits mit einem linearen Kern und ohne systematische Optimierung der Parameter Klassifizierungserfolge von bis zu 95% erzielt werden können. Ein kurzer Ausblick auf die Möglichkeiten der Optimierung des Modells mit dem Ziel noch höhere Klassifizierungserfolge zu erreichen wird in Kapitel 6 gegeben.

### 4.1.2 Anwendung der SVD auf den MNIST Datensatz

Gemäß des in Kapitel 3.3 entwickelten Modells wurden zunächst die Singulärwertzerlegungen der Bilddatenmatrizen $B^1, \ldots, B^{10}$ der Klassen 1 bis 10 (die Klasse 10 entspricht der Ziffer 0) und damit die ONBs $O^1, \ldots, O^{10}$ der Bilddatenräume berechnet, die durch die Trainingsbilder einer Klasse aufgespannten werden. Anschließend wurde die Klassifizierung der Testdaten vorgenommen und der Klassifizierungserfolg ermittelt. Die Klassifizierung wurde außerdem mit unterschiedlicher Anzahl an Singulärvektoren, die zur Approximation der Bilddatenräume ausgewählt wurden, durchgeführt. Berücksichtigt man alle 784 Singulärvektoren ergibt sich lediglich ein Klassifizierungserfolg von 21.90%. Dies ist insofern ersichtlich, als dass die Ränge der Bilddatenmatrizen $B^1, \ldots, B^{10}$ knapp über $r = 500$ liegen und damit kleiner als 784 sind. Folglich sind die hinteren Singulärvektoren keine orthonormalen Basisvektoren der Bilddatenräume. Der Einbezug dieser ist nicht sinnvoll und erklärt den schlechten Klassifizierungserfolg. Der beste Gesamtklassifizierungserfolg lag bei 95.85% und wurde mit der Wahl der ersten 23 Singulärvektoren als orthonormale Basisvektoren bei allen Bildklassen erreicht (vgl. Tabelle 5).

Tabelle 5: Prozentualer Klassifizierungserfolg (KE) auf den Testdaten für unterschiedliche Anzahl Singulärvektoren.

| **Singulärvektoren** | **1** | **2** | **3** | **4** | **5** | **10** | **23** |
|---|---|---|---|---|---|---|---|
| **KE** | 81.84 | 87.16 | 90.26 | 91.54 | 92.63 | 94.85 | 95.85 |
| **Singulärvektoren** | **30** | **50** | **100** | **200** | **300** | **500** | **784** |
| **KE** | 95.74 | 95.14 | 92.86 | 89.49 | 83.34 | 52.82 | 21.90 |

Die optimale Anzahl an Singulärvektoren von 23 lässt sich nicht an dem Verlauf der Singulärwerte zu den Klassen 1 bis 10 ablesen. Bei keiner der 10 Klassen ist ein markanter Abfall der Singulärwerte nahe der experimentell ermittelten *optimalen* Anzahl von 23 erkennbar. Exemplarisch sind die ersten 30 Singulärwerte für die Klassen 1 bis 3 in Abbildung 15 dargestellt. Die übrigen Klassen zeigen einen ähnlichen Verlauf.

Die durch Betrachtung der Rang-1-Zerlegung der Bilddatenmatrizen

$$B = \sum_{i=1}^{r} \sigma_i u_i v_i^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T \tag{4.1}$$

ersichtliche und durch Satz 3.4 bewiesene Tatsache, dass der Informationsgehalt bzw. der Beitrag der Singulärvektoren zu den Bildräumen konstant abnimmt, wird durch die Betrachtung aller Singulärwerte einer Klasse unterstrichen. Am Beispiel der Klasse drei verdeutlicht die Darstellung der Singulärwerte 100 bis 784 in Abbildung 15, dass die hinteren Singulärwerte bzw. -vektoren keinen nennenswerten Beitrag zu der Summe (4.1) liefern.

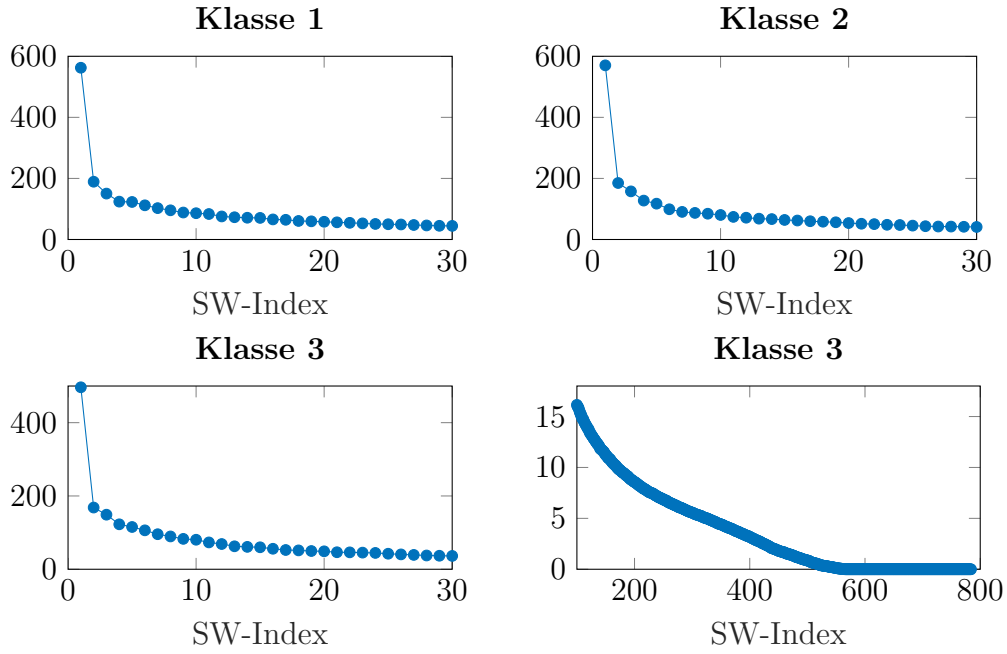


Abbildung 15: Die ersten 30 Singulärwerte $\sigma_i$ (SW) der Klassen 1, 2 und 3 sowie die Singulärwerte 100 bis 784 der Klasse 3.

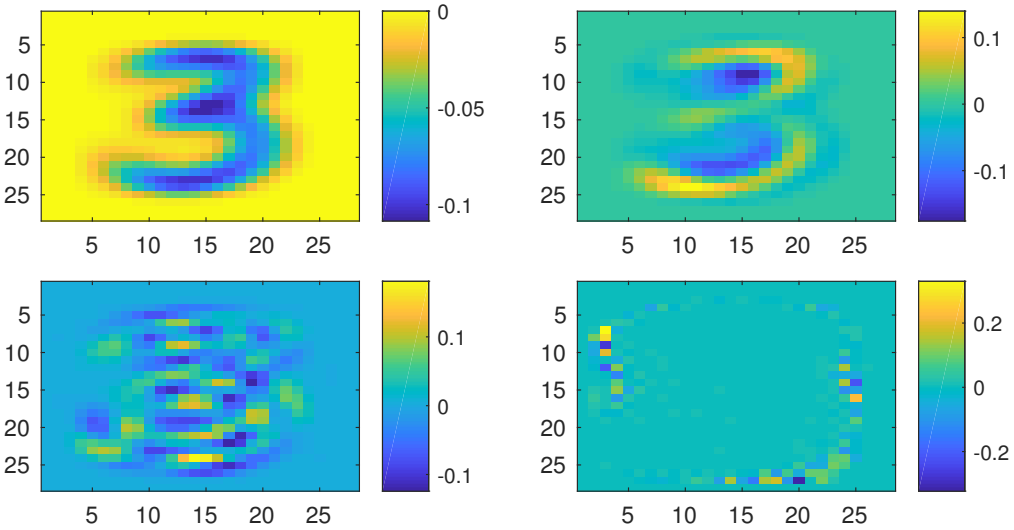

Abbildung 16: Singulärvektoren 1, 5, 50 und 500 der Klasse 3

Dass die Informationen über die markantesten Eigenschaften der Ziffern in den ersten Singulärvektoren enthalten sind, liefert auch die anschauliche Darstellung der Singulärvektoren 1, 5, 50 und 500 der Klasse 3 in Abbildung 16. Die Darstellung der ersten drei Singulärvektoren der Klassen 1 bis 3 in Abbildung 17 veranschaulicht zudem die drei *markantesten* Eigenschaften der Ziffern 1, 2 und 3.

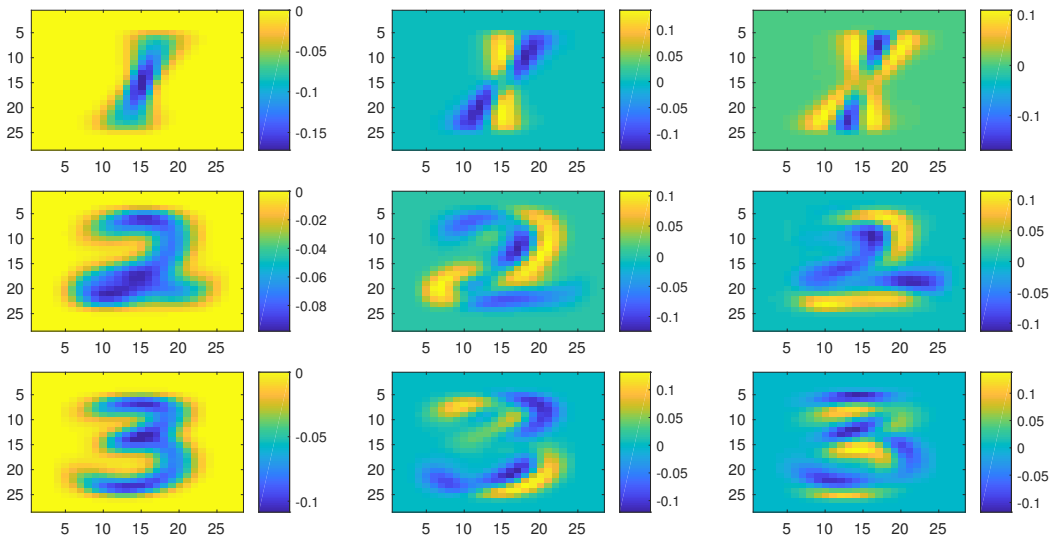

Abbildung 17: Die ersten drei Singulärvektoren der Klassen 1, 2 und 3

Wählt man die ersten 500 Singulärvektoren als orthonormale Basisvektoren der Bildklassen, so sinkt der Klassifizierungserfolg bereits auf knapp 50%. Das Bild einer beliebigen Klasse lässt sich *zu gut* durch die Bildbasen der anderen Klassen darstellen. Dies verdeutlicht auch die Approximation eines Bildes der Klasse 3 durch die Basen der Klassen 1, 2 und 3 bei einer Wahl von 23, von 100 und von 500 Singulärvektoren (vgl. Abbildung 18). Mit 100 Singulärvektoren ist die Zahl bereits als Ziffer 3 identifizierbar und mit 500 Singulärvektoren lässt sich für das menschliche Auge bereits kein Unterschied mehr zwischen dem Originalbild und den Approximationen ausmachen.

### 4.1.3 Vergleich der Ergebnisse auf dem MNIST Datensatz

Beide Methoden liefern ohne eine systematische Optimierung der Parameter gute Ergebnisse bei der Klassifizierung des Testdatensatzes von 94.95% im Falle der SVM und von 95.85% im Falle der SVD.
Anhand der bisher durchgeführten Experimente eine Einschätzung vorzunehmen, welche der beiden Methoden sich besser für die Bildklassifizierung eignet, kann gemessen am Klassifizierungserfolg an dieser Stelle nicht sinnvoll getroffen werden. Insbesondere aufgrund der Tatsache, dass beide Methoden noch verschiedene Optimierungsmöglichkeiten zur Erhöhung des Klassifizierungserfolgs bieten, auf die in Kapitel 6 ein Ausblick gegeben wird.
Zum Vergleich des Rechenaufwands bietet sich eine Gegenüberstellung der benötigten Berechnungszeiten für Trainings- und Testphase der beiden Methoden an.[17]
Dazu wurde die benötigte Rechenzeit für Trainings- und Testphase insgesamt be-

[17] Bei diesem Vergleich sei darauf hingewiesen, dass bei der Implementierung die Funktionalität der Algorithmen im Fokus stand und nicht deren maximale Effizienz hinsichtlich der notwendigen Rechenschritte. Eine weitere Optimierung der implementierten Algorithmen wäre möglich und würde vermutlich zu kürzeren Rechenzeiten führen.

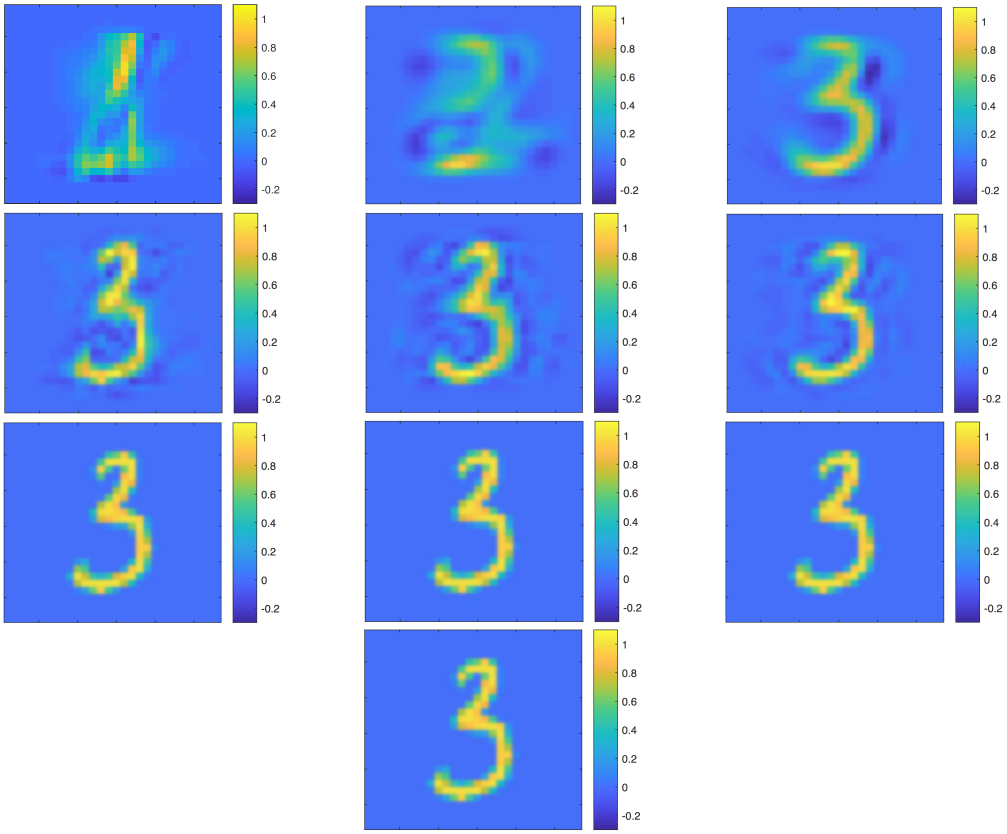

Abbildung 18: Ein Bild der Klasse 3 dargestellt durch die approximierten Basen der Klassen 1, 2, 3 (von links nach rechts) mit 23 (1. Reihe), 100 (2. Reihe), 500 Singulärvektoren (3. Reihe) sowie das Originalbild (4. Reihe)

stimmt, wobei bei beiden Methoden die Parametereinstellungen gewählt wurden, die bei den durchgeführten Experimenten zu maximalem Klassifizierungserfolg geführt haben.
Die Ergebnisse zeigen, dass die Trainings- und Testphase insgesamt bei der SVM mit 1001 Sekunden deutlich länger dauert als bei der SVD mit 255 Sekunden. Ein wesentlicher Vorteil der SVM ist jedoch, dass die Testphase nach einmaliger Entwicklung des mathematischen Modells in der Trainingsphase deutlich schneller verläuft als bei der SVD. Die Klassifizierung der Testdaten über die SVM in der Testphase ist mit 5 Sekunden deutlich kürzer als bei der SVD mit 92 Sekunden.

## 4.2 Klassifizierung von Gesichtern

Neben der bedeutenden Anwendung der Bildklassifizierung im Bereich der Schrift- bzw. Ziffernerkennung ist auch die automatisierte Gesichtserkennung eine höchst relevante und aktuelle Anwendung der Bildklassifizierung. Neben der bereits angesprochenen Klassifizierung bzw. dem Clustern von Gesichtern in verschiedenen sozialen Netzwerken spielt diese Technik bspw. auch bei der Erkennung von gefährlich eingestuften

Personen, wie Terroristen, eine immer größere, wenn auch kontrovers diskutierte Rolle. So wird seit Mitte 2017 ein Pilotprojekt an einem Berliner Bahnhof durchgeführt bei dem die Gesichter von Passanten von einer Kamera erfasst werden. Die aufgenommenen Bilder werden dann mit einer bestehenden Datenbank aus Bildern von knapp 300 Testpersonen verglichen (Reuters, 2017).

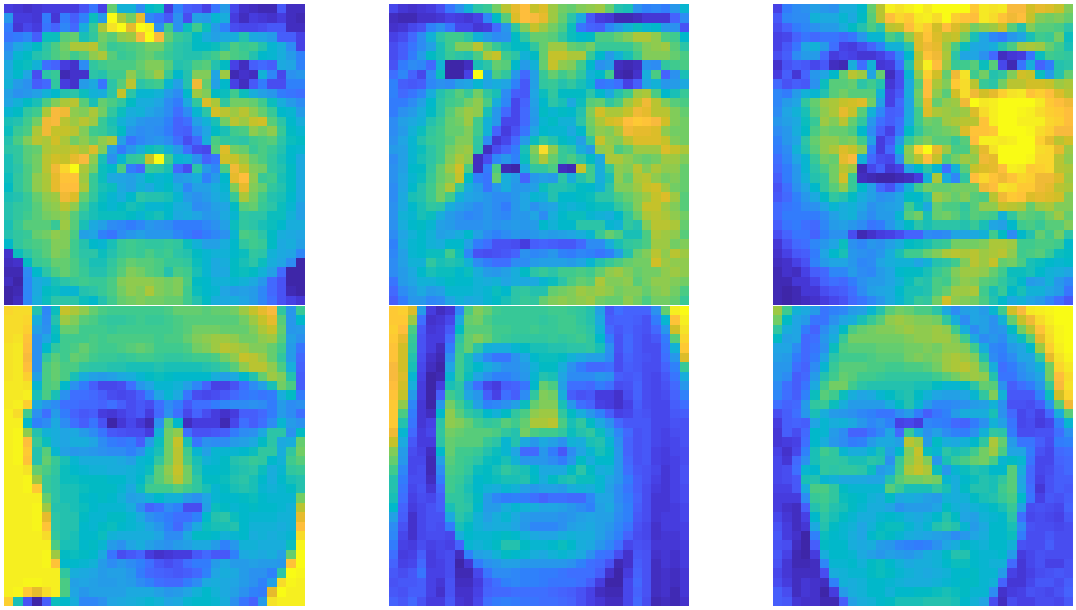

Abbildung 19: Drei beliebige Bilder verschiedener Klassen des Yale B Datensatzes (oben) und drei Bilder des eigenen Datensatzes (unten)

Mit Blick auf das Ziel dieser Arbeit, eine Grundlage für die Entwicklung eines Lernmoduls für Schüler zu der Thematik der automatisierten Bildklassifizierung zu schaffen, wurden die beiden beschriebenen Methoden auf diesen für Schüler womöglich interessanteren Anwendungskontext der automatisierten Gesichtserkennung angewendet. Dazu wurde zum einen ein online verfügbarer Datensatz, der Yale B Datensatz,[18] bestehend aus Klassen von 38 Individuen mit je 50 Trainings- und 14 Testbildern, zum anderen ein eigens generierten Datensatz von 8 Individuen mit ebenfalls 50 Trainings- und 14 Testbildern verwendet. Die schwarz-weiß Bilder beider Datensätze haben eine Größe von $32 \times 32$ Pixeln und zeigen die Frontalansicht der Gesichter der Personen. Einzelne Beispielbilder sind in Abbildung 19 dargestellt. Alle 64 Bilder zu einer Person aus dem eigenen Datensatz weisen nur minimale Variationen[19] auf und wurden jeweils unter den gleichen Lichtverhältnissen aufgenommen. Der Yale B Datensatz hingegen zeigt unterschiedliche Gesichtsausdrücke der einzelnen Personen und auch die Lichtverhältnisse wurden variiert. Bei der nachfolgenden Diskussion der Ergebnisse wird auf eine systematische und ausführliche Analyse der erhaltenen Ergebnisse verzichtet.

[18] www.cad.zju.edu.cn/home/dengcai/Data/FaceData.htm, Stand: 20. Oktober 2017
[19] Die Bilder wurden in Sekundenbruchteilen hintereinander aufgenommen.

Vielmehr soll ein Eindruck verschafft werden, inwieweit die Algorithmen beider Methoden auf die Gesichterdatensätze anwendbar sind und an welchen Stellen Probleme auftreten, die Verbesserungen der entwickelten Modelle erfordern.

### 4.2.1 Verwendung der SVM zur Gesichtsklassifizierung

Die Klassifizierung des Yale B Datensatzes mit der SVM führt bei Verwendung eines linearen Kerns und sowohl ohne als auch mit vorheriger Standardisierung der Daten mit verschiedene Werten für den Kostenfaktor $C$ zu einem schlechten Klassifizierungserfolg von unter 30%. Der Datensatz aus allen 38 Klassen scheint nicht linear separierbar. Möglichkeiten zur Erhöhung des Klassifizierungserfolgs wären zum einen neben der Standardisierung der Daten weitere Schritte der Vorverarbeitung durchzuführen, um markante Eigenschaften der Bildklassen herauszufiltern. Zum anderen könnten verschiedene Kernfunktionen erprobt werden, die die Daten in einen höher dimensionalen Merkmalsraum überführen. Interessant ist zu sehen, dass bei bei einer Reduktion des Yale B Datensatz auf die ersten 8 Klassen ein deutlich höherer Klassifizierungserfolg von 98.21% ($C = 0.1$) ohne vorherige Standardisierung und von 100% mit vorheriger Standardisierung der Daten ($C = 0.1$) erzielt werden kann.
Die Anwendung der SVM auf den eigenen Datensatz liefert bei der Soft Margin Klassifizierung mit $C = 0.1$ und bei der Hard Margin Klassifizierung unter Verwendung des linearen Kerns einen Klassifizierungserfolg von 100%. Dieses Ergebnis wird sowohl mit als auch ohne Standardisierung erzielt.

### 4.2.2 Verwendung der SVD zur Gesichtsklassifizierung

Bei der Klassifizierung des Yale B Datensatzes mit der Methode der SVD wurden die in Tabelle 6 dargestellten Klassifizierungserfolge erzielt. Das beste Ergebnisse von 95.53% konnte mit einer beliebigen Anzahl zwischen 34 und 41 der ersten Singulärvektoren als Basisvektoren der approximierten Bilddatenräume erreicht werden. Die Anwendung auf den eigenen Datensatz liefert für jede Anzahl an Singulärvektoren zwischen 1 und 50 einen Klassifizierungserfolg von 100%.

Tabelle 6: Prozentualer Klassifizierungserfolg (KE) über die Methode der SVD auf dem Yale B Testdatensatz für unterschiedliche Anzahl Singulärvektoren.

| **Singulärvektoren** | 1 | 2 | 3 | 4 | 5 | 10 | 30 | 50 |
|---|---|---|---|---|---|---|---|---|
| **KE** | 36.97 | 69.26 | 73.54 | 90.27 | 88.72 | 95.14 | 95.53 | 95.33 |

Die Betrachtung der ersten Singulärvektoren der einzelnen Klassen verdeutlicht, welches die *markantesten* Eigenschaften eines Gesichts auf einem schwarz-weiß Bild sind. Dies wird durch die exemplarische Darstellung der ersten drei Singulärvektoren von drei Individuen veranschaulicht (vgl. Abbildung 20).

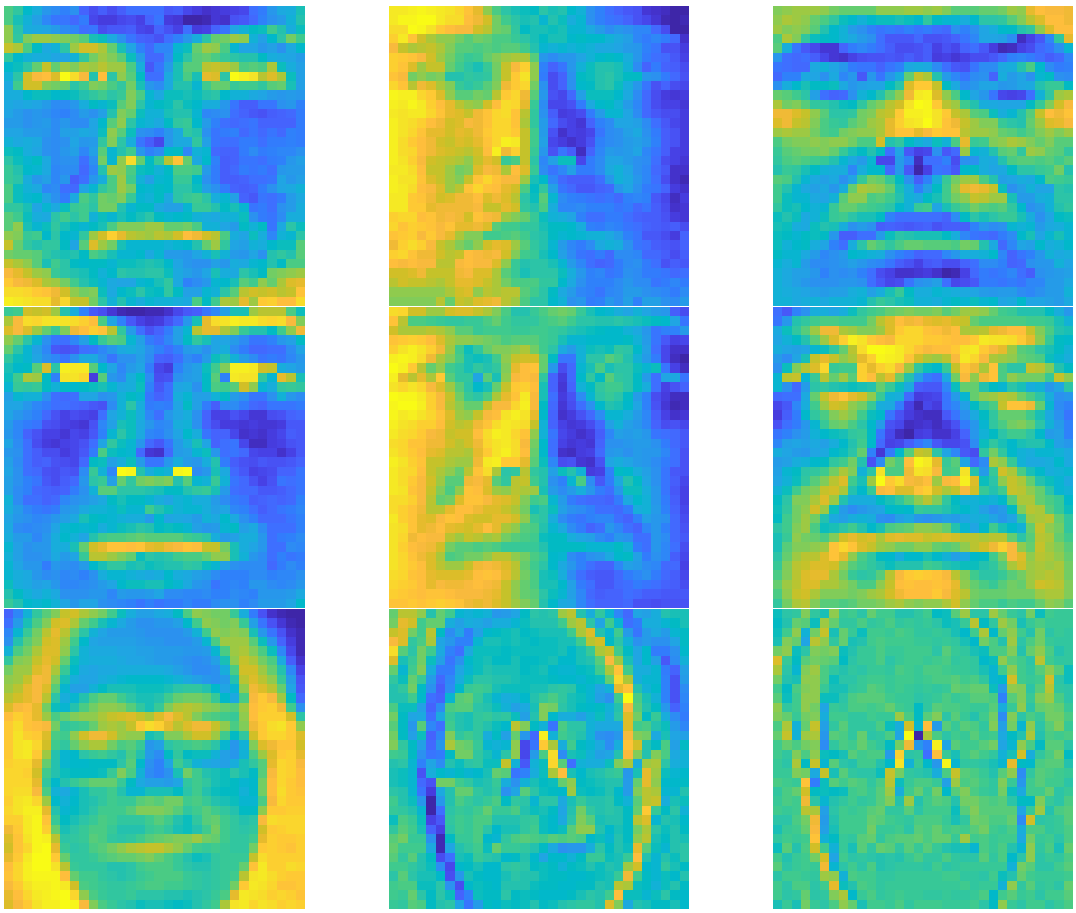

Abbildung 20: Darstellung des 1., 2. und 3. Singulärvektors (von links nach rechts) von zwei Klassen des Yale B Datensatzes (Reihe 1 und 2) und von einer Klasse des eigenen Datensatzes (Reihe 3)

Beim Yale B Datensatz stellen Nasenlöcher, Augen und Mund die markantesten, der Kontrast von rechter und linker Gesichtshälfte die zweit-markanteste und Nasenrücken sowie Wangenknochen die dritt-markantesten Merkmale der Gesichter dar. Bei dem eigenen Datensatz wurde ein größerer Bereich des Kopfes aufgenommen, sodass hier die Silhouette des Kopfes als informationsreiche Eigenschaft herausgefiltert wird.

# 5 Maschinelles Lernen in der mathematischen Modellierung mit Schülerinnen und Schülern

Eines der Ziele in der Vermittlung (angewandter) mathematischer Modellierung ist die Wahl von realen und lebensnahen Problemstellungen, durch deren Bearbeitung die Relevanz der Mathematik für Alltag, Wissenschaft und Forschung erfahrbar gemacht werden soll und die zudem die mathematische Modellierungskompetenz in besonderem Maße fördert (vgl. Greefrath et al., 2013, S. 21). In diesem Kapitel wird diskutiert, inwieweit die automatisierte Bildklassifizierung eine geeignete Fragestellung darstellt, um diesem Ziel Rechnung zu tragen. Dazu wird zunächst ein kurzer Überblick über wichtige Begriffe und die damit verbundenen theoretischen und didaktischen Hintergründe der mathematischen Modellierung gegeben sowie die wesentlichen Schritte des Modellierungsprozesses diskutiert. Des Weiteren wird die Kompetenz der mathematischen Modellierung, die während der mathematischen Ausbildung in der Sekundarstufe I und II in der Schule erworben werden soll, beschrieben. Im Anschluss werden Möglichkeiten der Gestaltung eines Lernmoduls zu der Problemstellung der Bildklassifizierung unter besonderer Berücksichtigung der gewählten Methoden, SVD und SVM, diskutiert. Die Überlegungen und Umsetzungsideen umreißen ein mögliches Grundgerüst für die didaktisch-methodische Entwicklung und Umsetzung eines solchen Lernmoduls. Dieses könnte im Rahmen eines computergestützten mathematischen Modellierungstages[20] mit Schülern der Sekundarstufe II durchgeführt werden.

## 5.1 Theoretischer Hintergrund der mathematischen Modellierung

Der Fokus der mathematischen Modellierung, die einen Teilaspekt der angewandten Mathematik darstellt, liegt auf dem Prozess des Lösens von Problemen aus der Realität (vgl. Greefrath et al., 2013, S. 11). Der Modellierungsprozess kann idealisiert in Form eines Kreislaufes dargestellt werden. Ein siebenschrittiger Modellierungskreislauf, der von Blum und Leiss (2005) visualisiert wurde, ist in Abbildung 21 dargestellt. Dieser beschreibt detailliert, welche Schritte vom Lernenden beim Lösen von Modellierungsaufgaben absolviert werden (können). Diese werden im Folgenden kurz erläutert.

Im 1. Schritt muss die Realsituation, d. h. eine Problemstellung oder eine Aufgabe aus der realen Welt, zunächst *verstanden* werden, um ein mentales Modell der Ausgangssituation, das sogenannte Situationsmodell, zu konstruieren. Da das Situationsmodell vielfach noch Angaben enthält, die irrelevant, überflüssig oder zu komplex sind, müssen die gegebenen Daten und Informationen im 2. Schritt *strukturiert* und *vereinfacht* sowie idealisierende Annahmen getroffen werden. Das weniger komplexe Realmodell stellt das Problem schließlich in vereinfachter Form dar. Bei der Vereinfachung und Idealisierung des Modells ist darauf zu achten, dass das Problem durch die getroffenen

[20] Einen möglichen Rahmen für die Durchführung eines Lernmoduls stellen die computergestützten mathematischen Modellierungstage bzw. -wochen des Schülerlabors CAMMP (Computational And Mathematical Modeling Program) dar. `www.cammp.rwth-aachen.de`, Stand: 12.01.2018

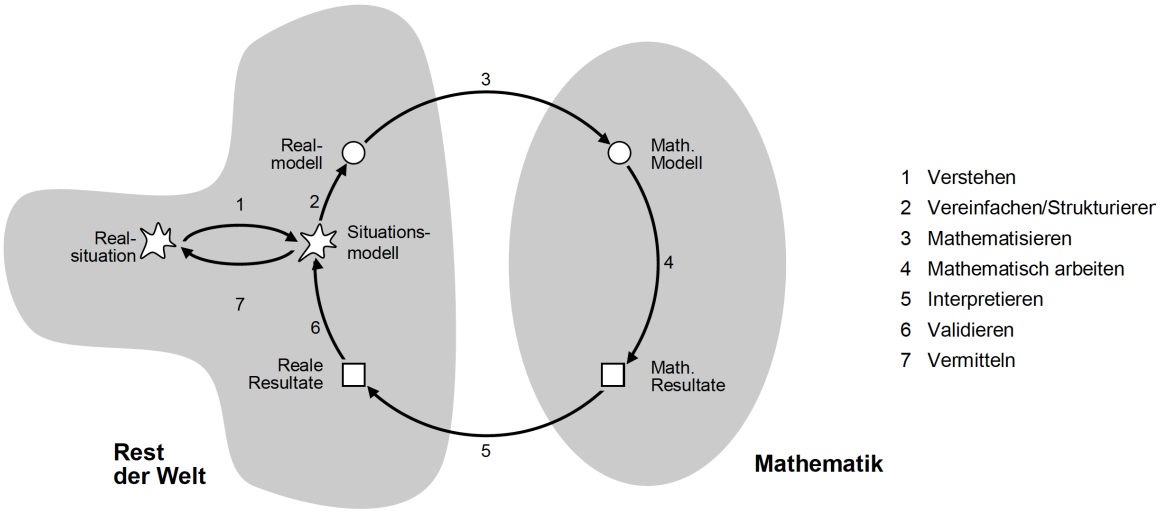


Abbildung 21: Siebenschrittiger Modellierungskreislauf (entnommen aus: Blum, 2006, S. 9)

Annahmen nicht *zu einfach* und dadurch falsch abgebildet wird. Das Realmodell wird im 3. Schritt durch *Mathematisierung* in ein mathematisches Modell übersetzt, auf das im 4. Schritt *mathematische Methoden* angewendet werden können, um schließlich ein mathematisches Resultat zu erhalten (vgl. Büchter & Leuders, 2011, S. 157). Die im Modell gewonnenen mathematischen Resultate müssen dann im 5. Schritt auf die Realsituation übertragen und *interpretiert* werden. Die so erhaltenen realen Resultate werden schließlich hinsichtlich ihrer Angemessenheit im Situationsmodell *validiert* (6. Schritt). Liefert dieser erste Durchgang durch den Modellierungskreislauf bereits gute Ergebnisse, so müssen diese im 7. und letzten Schritt verständlich kommuniziert und *vermittelt* werden, um die reale Problemstellung zu beantworten. Da jedoch nach dem einmaligen Durchlauf der Modellierungsschritte 2 bis 6 vielfach noch keine problemadäquate Lösung gefunden wird, ist der Modellierungskreislauf erneut zu durchlaufen. Dazu werden getroffene Annahmen und Vereinfachungen auf Gültigkeit geprüft, weitere Informationen einbezogen oder weggelassen, Daten auf ihrer Richtigkeit überprüft und das mathematische Modell modifiziert bzw. verbessert. Dieser iterative Prozess wird so oft wiederholt, bis eine zufriedenstellende Lösung gefunden wurde (vgl. Greefrath et al., 2013, S. 19).

Neben dem hier vorgestellten siebenschrittigen Kreislauf existieren verschiedene weitere Modellierungskreisläufe, die zu unterschiedlichen Zwecken entwickelt wurden. Der oben beschriebene Kreislauf schlüsselt alle Schritte auf, die bei der Bearbeitung von Modellierungsaufgaben idealtypischer Weise vom Lernenden absolviert werden. Er kann damit insbesondere dem Lehrenden als Orientierung dienen, wie Lernende mit Modellierungsaufgaben umgehen, und ist damit vor allem diagnostisch hilfreich.

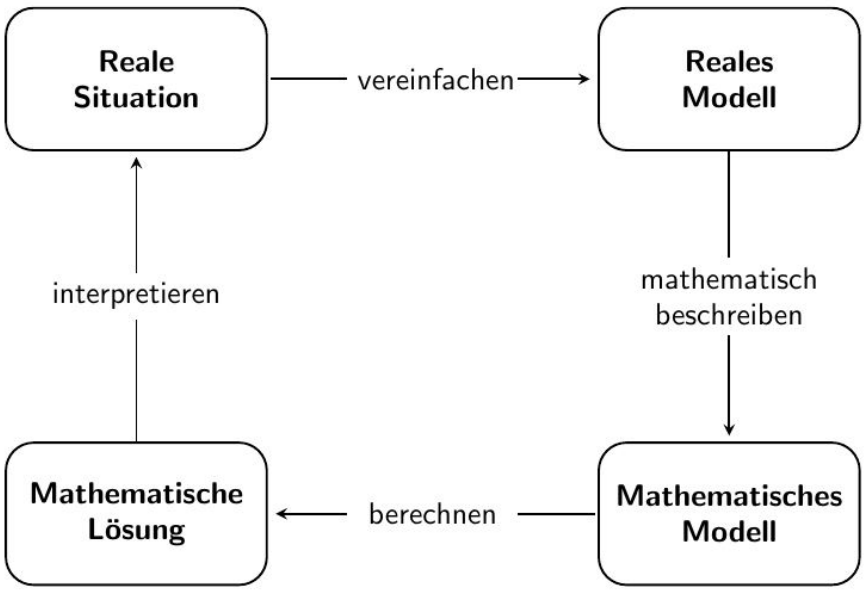


Abbildung 22: Vereinfachter Modellierungskreislauf angelehnt an Blum (1985) (vgl. Greefrath et al., 2013, S. 17).

Ein an Blum (1985) angelehnter vereinfachter Kreislauf, der sich auf die vier Schritte *vereinfachen*, *mathematisch beschreiben, berechnen* und *interpretieren* beschränkt, ist in Abbildung 22 dargestellt. Dieser eignet sich insbesondere für die Diskussion des Modellierungsprozesses mit den Schülern und kann somit im Bereich der Metakognition verortet werden (vgl. Greefrath et al., 2013, S. 14). Die in der didaktischen Diskussion gängige Darstellung des Modellierungsprozesses als Kreislauf lässt sich auf die Beschreibung durch eine Spirale erweitern. Eine Modellierungsspirale des Schülerlabors CAMMP, die an die *Solution Helix of Maths* von der Initiative *Computer-Based Math*[21] angelehnt ist und die sich in ähnlicher Form auch bei Büchter & Leuders (2011, S. 77) wiederfindet, ist in Abbildung 23 dargestellt. Die Modellierungsspirale greift die vier Modellierungsschritte des vereinfachten Kreislaufes von Blum (1985) auf, verdeutlicht jedoch darüber hinaus, dass die wiederholte Durchführung der Modellierungsschritte zu einer Annäherung an die *optimale Lösung* führt. Der Lösungsfortschritt wird veranschaulicht. Die dargestellte Modellierungsspirale nimmt zudem, wie von Greefrath & Weitendorf (2013) vorgeschlagen, den Einsatz des Computers als hilfreiches und, je nach Komplexität der Problemstellung, notwendiges Werkzeug in den Modellierungsprozess mit auf.

Aufgrund der großen Bedeutung der mathematischen Modellierung, die in zahlreichen Disziplinen der Ingenieurs- und Naturwissenschaften Anwendung findet, fast immer, wenn reale Probleme mithilfe von Mathematik gelöst werden, stellt das Modellieren

[21] `http://computerbasedmath.org/`, Stand: 09.02.2018

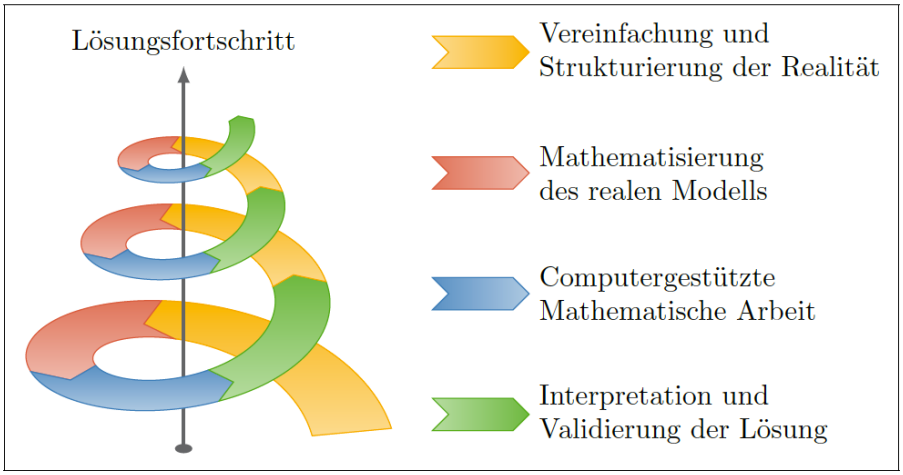


Abbildung 23: Computergestützte Modellierungsspirale des Schülerlabors CAMMP

eine der zentralen Kompetenzen des Mathematikunterrichts dar (vgl. Roeckerath et al., 2017, S. 2). Diese Kompetenz kann als die Fähigkeit, die oben dargelegten Prozessschritte der mathematischen Modellierung „problemadäquat ausführen zu können sowie gegebene Modelle analysieren und vergleichen zu können“(Blum, 2007, S. 6), beschrieben werden. Insbesondere durch die Einführung der länderübergreifenden Bildungsstandards (2003) ist diese Kompetenz als eine der sechs allgemeinen mathematischen Kompetenzen in den Vordergrund gerückt und stellt einen verbindlichen Bestandteil des Mathematikunterrichts dar (vgl. Blum, 2006, S. 8). Durch die Auswahl von lebensnahen und relevanten Problemstellungen soll nicht nur die Ausbildung der mathematischen Modellierungskompetenz gefördert, sondern gleichermaßen auch die große Bedeutung der Mathematik für Wissenschaft, alltägliches Leben oder Industrie hervorgehoben werden. Dieses (inhaltsbezogene) Ziel authentischer mathematischer Modellierung geht insbesondere mit der ersten der drei Winter'schen Grunderfahrungen, die ein allgemeinbildender Mathematikunterricht Schülern ermöglichen sollte, einher: „Erscheinungen der Welt um uns, die uns alle angehen oder angehen sollten, aus Natur, Gesellschaft und Kultur, in einer spezifischen Art wahrzunehmen und zu verstehen“(Winter, 1995, S.37).
Schülerrelevante Problemstellungen, die nur mithilfe von mathematischen Modellen gelöst werden können, finden sich hinter zahlreichen lebensnahen Anwendung, wie der Suchmaschine Google, dem globalen Navigationssatellitensystem GPS oder der Musikerkennungsapp Shazam,[22] um nur einige Beispiele zu nennen (vgl. Roeckerath et al., 2017, S. 2). Auch die Problemstellung der automatisierten Bildklassifizierung verbirgt sich hinter verschiedenen Anwendungen aus dem alltäglichen Leben (vgl. Kapitel 1) und macht diese somit zu einem vielversprechenden Kandidaten für die mathematische Modellierung mit Schülern. Inwieweit diese Problemstellung und das mathemati-

[22]Zu den genannten Themen wurden bereits Lernmodule im Rahmen des Schülerlabors CAMMP erstellt und vielfach mit Schülern im Zuge von eintägigen Workshops durchgeführt (vgl. `https://blog.rwth-aachen.de/cammp/angebote`, Stand: 08.02.2018).

sche Modell zur Lösung dieser Problemstellungen über die gewählten Methoden SVM und SVD mit dem Schulwissen der Schüler zugänglich gemacht werden kann, wird nachfolgend diskutiert und verschiedene Anknüpfungspunkte an die Schulmathematik dargelegt.

## 5.2 Grundstruktur eines Lernmoduls

Die hier beschriebene Grundstruktur des Lernmoduls orientiert sich an dem in Abbildung 22 dargestellten, vereinfachten Modellierungskreislauf. Dieser kann im Rahmen eines Lernmoduls auch explizit mit den Schülern thematisiert werden, um ihnen so eine Orientierungshilfe und einen Überblick über die Prozessschritte der mathematischen Modellierung zu geben.

Bei der Entwicklung eines Lernmoduls zu Klassifizierungsproblemen auf Basis maschineller Lernalgorithmen können unterschiedliche Schwerpunktsetzungen verfolgt werden. Zum einen kann der Fokus auf der mathematischen Beschreibung, der Ver- und Bearbeitung sowie der Klassifizierung von Bildern liegen. Zum anderen können zunächst Klassifizierungsprobleme im Allgemeinen in den Blick genommen und die Bildklassifizierung lediglich als Spezialfall betrachtet werden. Die nachfolgenden Ausführungen beziehen sich auf den erstgenannten Schwerpunkt, dem Problem der Bildklassifizierung. Auf mögliche schülerrelevante Klassifizierungsprobleme, die nicht aus dem Bereich der Bildklassifizierung stammen, wird in Kapitel 6 ein Ausblick gegeben.

### 5.2.1 Mathematische Modellbildung

Die nachfolgende Diskussion der notwendigen Modellierungsschritte bei der Bearbeitung von Klassifizierungsproblemen bezieht sich zunächst auf beide maschinelle Lernmethoden gleichermaßen. Eine Differenzierung wird erst in Abschnitt 5.2.2 und 5.2.3 bei der expliziten Entwicklung der mathematischen Modelle sowie der Diskussion der möglichen mathematischen Anknüpfungspunkte an das Schulwissen der Schüler vorgenommen.

#### Reale Situation

Zunächst wird die reale Problemstellung betrachtet: *Wie können Bilder automatisiert klassifiziert werden?* Die Relevanz dieser Fragestellung und die Notwendigkeit der Entwicklung von automatisiert und intelligent klassifizierenden Methoden lässt sich insbesondere durch die Bewältigung von riesigen Datenmengen motivieren. Dazu kann mit den Schülern diskutiert werden, dass hunderte von Bildern beim autonomen Fahren in Sekundenbruchteilen erfasst und klassifiziert werden müssen. Auch können eigene Ideen und Erfahrungen der Schüler, bei welchen Anwendungen ebenfalls Probleme der Bildklassifizierung auftreten, gesammelt werden.

Im Anschluss wird ein Überblick über die gegebene Daten- und Informationslage geschaffen und die Grundidee des überwachten maschinellen Lernens besprochen. Diese

Grundidee kann leicht über die Betrachtung einer konkreten Anwendung aus dem Bereich der Bildklassifizierung zugänglich gemacht werden:
Es liegt ein Datensatz bestehend aus Bildern verschiedener Bildklassen vor. Die Klassenzuordnung aller dieser Bilder sei bekannt. Dieser Datensatz wird zunächst aufgeteilt in einen großen Trainingsdatensatz und einen kleineren Testdatensatz. Das Ziel ist es unter Berücksichtigung der Trainingsdaten ein Modell zu entwickeln, welches sich zur Vorhersage unbekannter Bilder eignet. Damit jedoch die *Güte* des Modells überprüft werden kann, d.h. inwieweit das entwickelte Modell tatsächlich zum gewünschten Klassifizierungserfolg führt, wird dieses auf den Testdatensatz angewendet. Die Klassenzuordnung der Testbilder ist zwar bekannt, diese wird jedoch zunächst ausgeblendet. Die Testbilder werden dann mit dem entwickelten Modell klassifiziert und anschließend wird die bestimmte mit der tatsächliche Klassenzuordnung verglichen und evaluiert, inwieweit das Modell *gut* funktioniert. Als schülernahe und interaktive Anwendung ließe sich die konkrete Problemstellung der Gesichtsklassifizierung in den Blick nehmen. Zu Beginn des Schülerworkshops könnten die Schüler einen eignen Datensatz aufnehmen. Das Bild eines beliebigen Schülers wird geschwärzt und das übergeordnete Ziel des Workshops wäre es, ein Modell zur Klassifizierung zu entwickeln, durch das dieses geschwärzte Bild korrekt zugeordnet werden kann. Eine derartige Gestaltung des Lernmoduls sollte vor dem Hintergrund einer realen Anwendung, wie bspw. der erwähnten Überwachung in öffentlichen Bahnhöfen, durchgeführt werden.

### Reales Modell

Da die reale Situation abhängig von dem betrachteten Klassifizierungsproblem sehr komplex sein kann, werden zunächst verschiedene Vereinfachungen vorgenommen und idealisierende Annahmen getroffen. Diese sollten gemeinsam mit den Schülern diskutiert werden. Zu diesen Vereinfachungen zählt die ausschließliche Betrachtung von schwarz-weiß Bildern, die Beschränkung auf Klassifizierungsprobleme mit zunächst nur zwei Klassen sowie die Annahme, dass die gegebenen Klassenzuordnungen der Bilder korrekt sind. Anschließend kann zu der Frage übergeleitet werden, die sogleich zum 1. Schritt bei der Entwicklung der mathematischen Modelle führt: *Wie lassen sich Bilder mathematisch beschreiben?*

### Mathematisches Modell und Mathematische Lösung

Die Entwicklung der mathematischen Modelle bei beiden dargestellten maschinellen Lernmethoden basiert auf einem zentralen Strukturbegriff der Mathematik: dem Vektorbegriff. Die Vektoren repräsentieren bei den Klassifizierungsproblemen Objekte verschiedener Klassen, wobei die Komponenten der Vektoren verschiedene Eigenschaften dieser Objekte, im Falle der Bildklassifizierung die Grauwerte der Pixel, widerspiegeln. Auch in der Schule stellt der Vektorbegriff ein zentrales Konzept des Mathematikunterrichts dar. Meist werden Vektoren dort jedoch primär geometrisch als Verschiebungen in Ebene und Raum eingeführt. Sie werden dann als *Pfeilklassen* aufgefasst, was einen sehr anschaulichen Zugang zum Vektorbegriff ermöglicht, jedoch gleichzeitig die Gefahr der Begriffseinengung und -verzerrung birgt (vgl. Henn & Filler, 2015, S. 88).

Ein Lernmodul zu der Thematik der Bildklassifizierung über die beschriebenen Methoden bietet die Möglichkeit, den Fokus auf eine stärker arithmetisch-algebraische Betrachtung von Vektoren als $n$-Tupel (auch für $n \geq 3$) zu legen und den Nutzen dieser Betrachtung durch die Anwendbarkeit von bekannten Konzepten der metrischen analytischen Geometrie hervorzuheben. Auch erleichtert die Betrachtung der Vektoren als $n$-Tupel die Erweiterung von Problemen des bekannten 2- bzw. 3- dimensionalen Raums auf den $n$-dimensionalen Fall. Verschiedene Kernaussagen der metrischen analytischen Geometrie, die im Rahmen eines Lernmoduls Anwendung finden könnten, werden in Abschnitt 5.2.2 und 5.2.3 diskutiert. Didaktische Überlegungen, wie die Brücke zwischen geometrischen Konzepten und einer nicht geometrischen Fragestellung geschlagen werden kann, finden sich bei Heitzer (2012). Dort wird anhand verschiedener Problemstellungen, wie der Bildkompression, diskutiert, inwieweit „sonst auf die Geometrie beschränkte Begriffe wie Vektor, Linearkombination, Unterraum, Lineare Unabhängigkeit, Erzeugnis, Basis, Dimension, Skalarprodukt, Länge bzw. Norm, Abstand bzw. Metrik, Orthogonalität oder Projektion [...] in einen größeren Zusammenhang gestellt und mit neuem Leben erfüllt [werden können]“(Heitzer, 2012, S. 174). Bei der konkreten Entwicklung des Lernmoduls können diese Ausführungen als methodisch-didaktische Orientierungshilfe dienen.

Im Rahmen eines Lernmoduls zur Bildklassifizierung kann anhand von Beispielen zunächst die Betrachtung von Bildern als Raster, die aus kleinen Rechtecken bestehen, eingeführt werden. Diese Rechtecke stellen die Pixel dar, die durch Grauwerte repräsentiert werden. Ein Bild lässt sich dann als Matrix oder Tabelle interpretieren, deren Einträge die Grauwerte der Pixel angeben. Diese werden anschließend durch Aneinanderreihung der Spalten zu Vektoren umgeschrieben, was nicht nur eine übersichtlichere Darstellung liefert, sondern insbesondere auch die Anwendung bekannter Konzepte der analytischen Geometrie ermöglicht. Diese Vorteile könnten zum Abschluss des Lernmoduls explizit mit den Schülern reflektiert werden. Abhängig davon, wie detailliert sich mit Bildern und deren digitaler Repräsentation auseinander gesetzt werden soll, können verschiedene Transformationen, wie Drehungen, Spiegelungen sowie Kombinationen von Bildern, betrachtet und diskutiert werden, um ein tieferes Verständnis für die Darstellung von digitalen Bildern zu schaffen.

Bei der Entwicklung der mathematischen Modelle mit den Schülern kann hinsichtlich des Abstraktionsniveaus variiert und unterschiedliche Abstufungen der Analyse und Diskussion der mathematischen Hintergründe gewählt werden. So kann verstärkt auf der Anschauungsebene gearbeitet und mathematische Begrifflichkeiten und Verfahren, die den Schülern nicht bekannt sind, lediglich als *hilfreiche Techniken* zur Entwicklung des mathematischen Modells präsentiert werden. Andernfalls könnten diese neuen Begriffe und Verfahren auch formal betrachtet und eingeführt werden. Beispiele dafür stellen die Entwicklung von Orthonormalbasen sowie die Dimension von Unterräumen oder die Diskussion der Optimalitätsbedingungen bei der Lösung von Minimierungsproblemen dar.

Um eine anschauliche Heranführung an die Problemstellung zu ermöglichen und Schwierigkeitsgrad bzw. Abstraktionsniveau langsam wachsen zu lassen, kann zunächst ein kleiner Datensatz aus nur zwei Klassen und Daten des $\mathbb{R}^2$ oder des $\mathbb{R}^3$ betrachtet werden. Die Problemstellung ist dann im geometrischen Raum darstellbar, dessen Strukturen mitsamt der geltenden Zusammenhänge den Schülern bekannt sind. Im Falle der Bildklassifizierung ließe sich ein einfacher Datensatz verwenden, bei dem entweder Bilder bestehend aus nur 2 bzw. 3 Pixeln betrachtet werden oder bestimmte Eigenschaften der Bilder extrahiert und diese als Komponenten von Vektoren des $\mathbb{R}^2$ bzw. des $\mathbb{R}^3$ gespeichert werden. Beispielsweise könnte der erste Eintrag den durchschnittlichen Grauwert über alle Pixel repräsentieren, der zweite Eintrag die Anzahl der weißen und der dritte Eintrag die Anzahl der schwarzen Pixel. Im Falle der Gesichtsklassifizierung könnte die erste Komponente den Abstand zwischen den Augen und die zweite bzw. dritte Komponente die Breite bzw. Höhe des Kopfes widerspiegeln.[23] Im geometrischen Anschauungsraum ließe sich ein erstes Modell zur Klassifizierung entwickeln, welches schließlich durch die Abstraktion vom 2- bzw. 3- dimensionalen Fall auf den $n$-dimensionalen Fall erweitert werden kann.

Bei der expliziten Entwicklung der mathematischen Modelle zu den Methoden SVM oder SVD und den möglichen Anknüpfungspunkten an das Wissen der Schüler müssen dann abhängig von der gewählten maschinellen Lernmethode unterschiedliche Wege eingeschlagen werden, was in Abschnitt 5.2.2 und 5.2.3 differenziert diskutiert wird.
Ein zentraler Aspekt, der jedoch für beide Methoden von übergeordneter Bedeutung ist, stellt die Verwendung des Computers als digitales Werkzeug dar. Berechnungen, die zu aufwendig mit Stift und Papier durchzuführen sind oder die weit über das Schulwissen hinausgehen und womöglich für Schüler zu komplex sind, können dem Computer überlassen werden. Auch ermöglicht der Computereinsatz eine schnelle Visualisierung der Ergebnisse, die experimentelle Betrachtung vieler Beispiele in kurzer Zeit sowie eine direkte Überprüfung und Rückmeldung zu den Ergebnissen der Schüler (vgl. Heitzer, 2012, S. 176). Durch den Einsatz des Computers sollen die Schüler von aufwendiger Rechenarbeit entlastet werden, damit die hinter den Berechnungen liegende Bedeutung in der Vordergrund rückt (vgl. Leuders, 2011, S. 207). Der Fokus des Lernmoduls kann so auf die kreative Bearbeitung der anderen Modellierungsschritte gelegt werden.
Im Falle beider Methoden ist das *finale* mathematische Modell durch eine *Entscheidungsfunktion* gegeben, in die neue unbekannte Datenpunkte (Vektoren) eingesetzt werden und deren Auswertung zu der mathematischen Lösung führt: Eine Zahl, welche die Zuordnung zu einer Klasse angibt. Die Entscheidungsfunktion stellt damit einen Anknüpfungs- bzw. Erweiterungspunkt des Lernmoduls an den aus der Schule bekannten Funktionenbegriff dar, wobei der inhaltliche Schwerpunkt von Funktionen als mathematische Modelle betont wird. Zudem geht die Entscheidungsfunktion über die in der Schule primär betrachteten Funktionen, bei denen Definitions- und Werte-

[23] Inwieweit diese Betrachtungen tatsächlich sinnvoll sind, hängt von dem konkret verwendeten Datensatz und dem vorliegenden Klassifizierungsproblem ab. Zudem ist abzuwägen, inwieweit eine derartige Vorverarbeitung noch der Grundidee der maschinellen Lernmethoden, die Klassifizierung *automatisiert* vorzunehmen, genüge tut.

bereich i. d. R. aus demselben euklidischen Vektorraum stammen, hinaus (vgl. KLP Mathematik Sek. II, S. 33).

**Modellverbesserungen**

Werden die Modelle zunächst auf kleine Datensätze angewendet, bei denen mit den mathematischen Modellen in ihrer einfachsten Form gute Klassifizierungsergebnisse erzielt werden, kann durch die Anwendung auf komplexere Datensätze die Notwendigkeit der Verbesserung der Modelle angeregt werden. Komplexere Datensätze könnten bspw. solche sein, die aus mehr als nur zwei Klassen bestehen oder die sich überschneidende Klassenverteilungen aufweisen. Die Modellverbesserungen wären dann von den Schülern in das bestehende Modell einzubauen und die einzelnen Schritte des Modellierungsprozesses erneut zu durchlaufen.

### 5.2.2 Gestaltungsideen und mathematische Anknüpfungspunkte - SVM

Bei der Entwicklung des mathematischen Modells im Fall der SVM bietet es sich an die Schüler zunächst mit Stift, Papier und Lineal an einem kleinen Datensatz zweier linear separierbarer Klassen mit Daten aus dem $\mathbb{R}^2$ die *beste Gerade* konstruieren zu lassen. Auf diese Weise kann die Idee des *maximalen Margins* motiviert werden. Die gleichen Überlegungen lassen sich im nächsten Schritt auf den $\mathbb{R}^3$ übertragen.

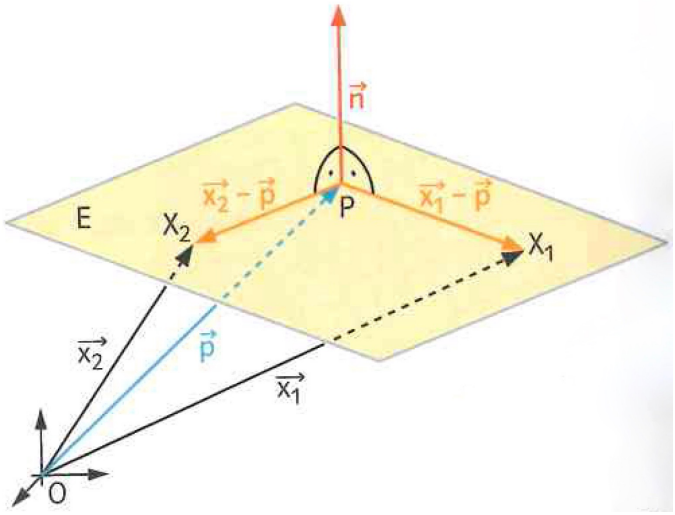


Abbildung 24: Darstellung einer Ebene mit zugehörigem Normalenvektor aus einem Schulbuch der Sekundarstufe II (entnommen aus: Baum et al., 2011, S. 214)

Die Berechnung des maximalen datenpunktfreien Bereichs an dem konkreten Beispiel führt sogleich zu der Berechnung des Abstands von Punkten zu Geraden bzw. Ebenen. Im $\mathbb{R}^3$ wird in der Schule[24] für die Berechnung des Abstands $d$ eines Punktes $R \in \mathbb{R}^3$ zu

[24]Die Darstellung von Ebenen in Normalenform stellt nur noch für den Leistungskurs einen verpflichtenden inhaltlichen Bestandteil der Oberstufe dar (vgl. KLP Mathematik Sek. II, S. 33).

einer Ebene $E$ eine Formel verwendet, für die die Ebene in Hesse'scher Normalenform

$$E : (\vec{x} - \vec{p}) \cdot \vec{n} = 0$$

mit $\vec{x}, \vec{p} \in \mathbb{R}^3$ und Normalenvektor $\vec{n} \in \mathbb{R}^3$ bestimmt werden muss (vgl. Abbildung 24). Für den Abstand gilt dann

$$d = \left\| (\vec{r} - \vec{p}) \frac{\vec{n}}{\|\vec{n}\|} \right\|,$$

wobei $\vec{r}$ dem Ortsvektor des Punktes $R$ entspricht (vgl. Baum et al., 2011, S. 214). Damit liegt bereits die Darstellung der Hyperebene bzw. des Abstandes vor, wie sie auch im allgemeineren Fall beim Support Vektor Lernen genutzt wird (vgl. Kapitel 3.2.1).
Durch eine entsprechende Wahl der Daten des Einstiegsbeispiels sollte ersichtlich werden, dass für die Bestimmung der (Hyper-)Ebene lediglich die Datenpunkte bzw. die Vektoren, die *am nächsten* an der separierenden Ebene liegen, betrachtet werden müssen, um die Definition der *Stützvektoren* zu motivieren.
Im nächsten Schritt lässt sich mit den Schülern die Frage erörtern, wie die bestimmte Gerade bzw. Ebene zur Klassifizierung neuer Datenpunkte genutzt werden kann. Anschaulich ist klar, dass die Punkte abhängig davon, ob sie oberhalb oder unterhalb der Gerade bzw. Ebene liegen, zugeordnet werden können. Doch wie lässt sich diese Lagebestimmung anhand der berechneten Hyperebene festlegen? Zu der Erkenntnis, dass dies durch das Vorzeichen, welches sich beim Einsetzen des Punktes in die Gleichung der Hyperebene ergibt, festgelegt werden kann, können die Schüler auf verschiedenen Wegen gelangen. Zum einen können sie dies experimentell feststellen, indem sie verschiedene Punkte in die Ebenengleichung einsetzen.
Zum anderen kann die Klassenzuordnung über das Vorzeichen auch zunächst anschaulich und anschließend unter Einbezug des aus der Schule bekannten Skalarprodukts auch formal hergeleitet werden.
Die zunächst anschauliche Betrachtung einer Ebene im $\mathbb{R}^3$ sollte verdeutlichen, dass für den Winkel $\alpha$ zwischen $\vec{x} - \vec{p}$ und $\vec{n}$ mit $\vec{x}$ als zu klassifizierendem Punkt gilt:

- $0° \leq \alpha < 90° \Leftrightarrow \vec{x} - \vec{p}$ zeigt in die gleiche Richtung bzgl. der Hyperebene wie $\vec{n}$
- $\alpha = 90° \Leftrightarrow \vec{x} - \vec{p}$ liegt in der Hyperebene
- $90° < \alpha \leq 180° \Leftrightarrow \vec{x} - \vec{p}$ zeigt in die entgegengesetzte Richtung bzgl. der Hyperebene wie $\vec{n}$

Diese Fallunterscheidung kann von den Schülern anschließend über das Skalarprodukt hergeleitet werden. Folgender Zusammenhang für den Winkel $\alpha$ zwischen zwei Vektoren $\vec{a}$ und $\vec{b}$ ist aus der Schule bekannt (KLP Mathematik Sek. II, S. 29):

$$\vec{a} \cdot \vec{b} = \frac{\cos(\alpha)}{\|a\| \cdot \|b\|}.$$

Die Berücksichtigung dieses Zusammenhangs liefert schließlich folgendes Ergebnis:

$$(\vec{x}-\vec{p})\cdot\vec{n}=\frac{\cos(\alpha)}{\|(\vec{x}-\vec{p})\|\cdot\|\vec{n}\|}=\begin{cases}>0 & \text{für } 0^\circ\leq\alpha<90^\circ\\ =0 & \text{für } \alpha=90^\circ\\ <0 & \text{für } 90^\circ<\alpha\leq 180^\circ\end{cases}$$

Die Zuordnung eines Punktes $\vec{x}$ zu einer der beiden Klassen gemäß des Vorzeichens beim Einsetzen in die (Hyper-)Ebenengleichung führt die Schüler zu der Entscheidungsfunktion im binären linear separierbaren Fall und damit zu einem ersten mathematischen Modell.
Durch die Wahl eines Datensatzes, bei dem nicht direkt erkennbar ist, welches die *nächsten Datenpunkte* sind, kann aufgezeigt werden, dass die direkte Berechnung einer Hyperebene über die nächstgelegenen Vektoren nicht immer so leicht möglich ist wie im Einstiegsbeispiel. Die allgemeinere Betrachtung und Festlegung der Problemstellung gemäß

$$(\vec{x}-\vec{p})\cdot\vec{n}=\begin{cases}>0 & \text{falls Trainingspunkt } \vec{x} \text{ aus Klasse 1}\\ <0 & \text{falls Trainingspunkt } \vec{x} \text{ aus Klasse 2}\end{cases}$$

wird motiviert. Unter der Berücksichtigung, dass der kleinste Abstand zwischen den Datenpunkten und der Hyperebene maximal werden soll, kann dies von den Schülern formuliert werden als:

Finde den Normalenvektor $\vec{n}$ und einen Vektor $\vec{p}$, sodass das Minimum der Abstandsfunktion

$$d(\vec{x})=\left\|(\vec{x}-\vec{p})\frac{\vec{n}}{\|\vec{n}\|}\right\|. \tag{5.1}$$

für alle Datenpunkte $\vec{x}$ aus dem Trainingsdatensatz maximal ist.

Die konkrete Lösung dieses Optimierungsproblems kann dann einer Computersoftware überlassen werden.
Ausgehend von dem einfachen Einstiegsbeispiel mit exakt linear separierbaren Daten lässt sich anschließend das Problem der sich überlappenden Klassenverteilungen betrachten. Dazu kann die Fehlerschranke $C$ und deren Einfluss auf die Lage der Hyperebene diskutiert werden. Diese Diskussion kann anschaulich auf Basis der Visualisierung von Hyperebene und Datenpunkten erfolgen, wie in Abbildung 25 dargestellt.

Im nächsten Schritt lassen sich bei der Erstellung des Lernmoduls verschiedene Wege einschlagen. Eine Möglichkeit ist, nun die Abstraktion vom 2- bzw. 3-dimensionalen Fall hin zum $\mathbb{R}^n$ zu vollführen und dabei beliebig detailliert zu diskutieren, dass die gleichen Rechenregeln wie im $\mathbb{R}^3$ angewandt werden können. Ist diese Abstraktion

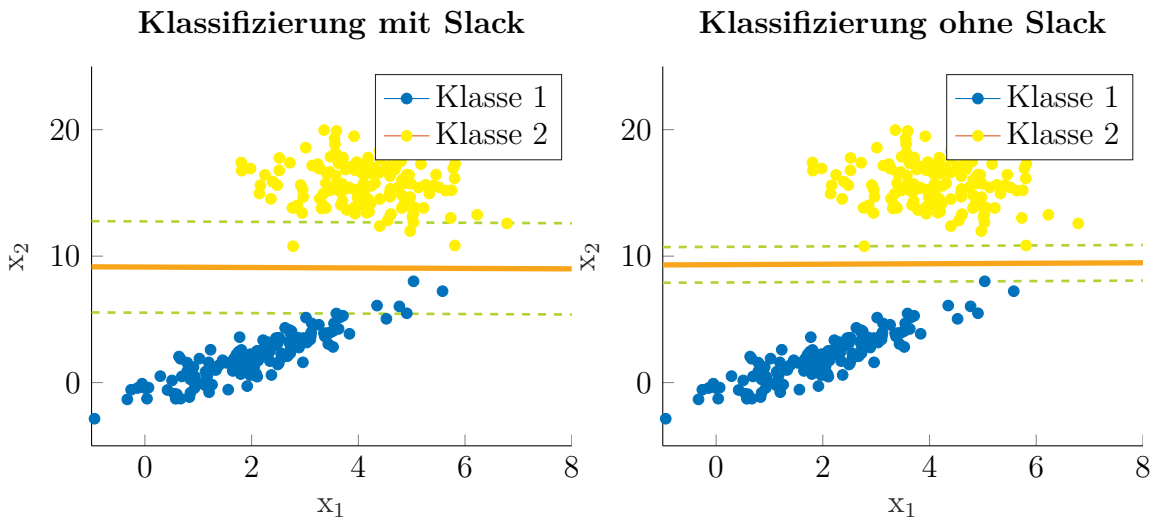


Abbildung 25: Klassifizierung eines binären linear separierbaren Datensatzes mit Slack (C=0.01) und ohne Slack.

erfolgt, lassen sich Klassifizierungsprobleme mit Bilddaten der Größe $r \times r$ betrachten, indem die oben beschriebene Schreibweise von Bildern als $n$-Tupel eingeführt wird. Die Diskussion von Kernfunktionen und deren Verwendung im nichtlinearen Fall scheint für die mathematische Modellierung mit Schülern eher ungeeignet. Es ist jedoch möglich diesen Fall exemplarisch, an einem konkreten Beispiel zu betrachten. Was wiederum durchaus mit den Schülern untersucht werden kann, ist die Kombination mehrerer SVMs bei Datensätzen aus mehr als 2 Klassen. Eine Möglichkeit besteht darin, die OVO Klassifizierung an einem Beispiel im $\mathbb{R}^2$ inklusive auftretender Probleme, wie eine womöglich Uneindeutigkeit der Zuordnung, zu diskutieren (vgl. Kapitel 3.2.3).

### 5.2.3 Gestaltungsideen und mathematische Anknüpfungspunkte - SVD

Die mathematischen Hintergründe der Klassifizierungsmethode über die SVD basieren auf wesentlichen Konzepten der linearen Algebra wie Basis, Dimension sowie lineare Unabhängigkeit von Vektoren - Konzepte, die nicht im Kernlehrplan der Sekundarstufe II in NRW verankert sind, und die in der Schule vermutlich kaum oder nur am Rande Anwendung finden (vgl. KLP Mathematik Sek. II). Die Entwicklung eines Lernmoduls zu dieser Methode beinhaltet somit die Schwierigkeit, ein Gleichgewicht zwischen der Vermittlung bzw. Erarbeitung neuen Wissens und neuer Konzepte und dem Fokus auf dem Prozess der mathematischen Modellierung zu finden. Zugleich bietet ein Lernmodul zu dieser Methode aber die Chance, Begriffe wie Basis, Dimension und lineare Unabhängigkeit anhand einer konkreten Problemstellung einzuführen und zu veranschaulichen. Gerade mit Blick auf die Anwendung in der Gesichtsklassifizierung stellt die Betrachtung der Basen einer Gesichterklasse bzw. der Anzahl notwendiger Basisvektoren zur Approximation eines Gesichts eine interessante Herangehensweise an das für die meisten Schüler neue Konzept der Basis dar.

Bei der Entwicklung des Lernmoduls lässt sich, wie bei der Methode der SVM, zunächst ein kleines Klassifizierungsproblem mit Daten aus dem $\mathbb{R}^2$ oder $\mathbb{R}^3$ betrachten. Die Erweiterung von diesem einfachen Problem aus dem geometrischen Anschauungsraum und die Übertragung bekannter Strukturen und Zusammenhänge auf höherdimensionale Räume sowie die Einführung neuer Konzepte wird im Rahmen dieser Arbeit nicht ausführlich didaktisch diskutiert. Zahlreiche Anregungen finden sich u. a. bei Heitzer (2012), die diese Erweiterung detailliert diskutiert.

Eine mögliche Schrittfolge bei der weiteren Entwicklung des mathematischen Modells über die Methode der SVD wäre die Folgende:

1. Zunächst wird die Idee diskutiert, dass die Darstellung eines Datenpunktes durch die Daten seiner Klasse am besten möglich ist und zu der Fragestellung der besten Approximation übergeleitet.

2. Der Zusammenhang zwischen orthogonaler Projektion und der besten Approximation wird an einem konkreten Beispiel in der Ebene oder dem Raum betrachtet.

3. Die orthogonalen Projektion wird an dem ausgewählten Beispiel bestimmt, indem der Lotfußpunkt eines Datenpunktes auf eine Gerade bzw. Ebene und anschließend der Abstand zwischen Lotfußpunkt und Datenpunkt berechnet wird. Dies ist inhaltlicher Bestandteil des Mathematikunterrichts in Grund- und Leistungskursen der Sekundarstufe II (vgl. KLP Mathematik Sek. II, S. 29). Der berechnete Abstand wird dann als Maß für die Güte der besten Approximation herangezogen.

4. Unterräume, die durch sämtliche lineare Kombinationen der Datenpunkte einer Klasse entstehen, werden eingeführt. Die Betrachtung der Unterräume kann am Beispiel des $\mathbb{R}^3$ erfolgen, in dem diese entweder einen Punkt, eine Gerade, eine Ebene oder sogar den ganzen Raum darstellen.

5. Beliebig detailliert wird die Diskussion zu Basen und Dimensionen von Unterräumen geführt, als neue Konzepte die im Laufe des Lernmoduls erarbeitet und die nicht als Schulwissen vorausgesetzt werden können.

6. Die leichtere Berechnung der orthogonalen Projektion als besten Approximation eines Datenpunktes über ONBs kann zur Motivation der Berechnung eben solcher Basen herangezogen werden. An dieser Stelle lässt sich erneut beliebig detailliert auf die Existenz und auf die Berechnung von ONBs eingehen. Liegt der Schwerpunkt stärker auf der direkten Problembearbeitung und weniger auf der Diskussion neuer mathematischer Ideen und Konzepte, so lässt sich hier bereits die SVD als hilfreiches Werkzeug zur Berechnung von ONBs einführen. Die

*interessante* Eigenschaft der bestimmten Basisvektoren, einen abnehmenden Informationsgehalt über eine Bildklasse zu beinhalten, kann dann durch Betrachtung verschiedener bildlich dargestellter Basisvektoren experimentell erfahrbar gemacht werden.

7. Durch die Anwendung des bis hierhin entwickelten Modells auf einen komplexeren Datensatz, bestehend aus mehr als zwei Klassen, wird schließlich die Modellverbesserung der Reduktion der Basisvektoren motiviert und durchgeführt. Dies kann anhand der Darstellung der *zu guten* Approximation eines Datenpunktes durch die Basen anderer Klassen veranschaulicht werden.

Die hier beschriebene Schrittfolge bezieht sich insbesondere auf den geometrischen Anschauungsraum und damit auf Eingangsdaten aus dem $\mathbb{R}^2$ oder $\mathbb{R}^3$. Die Abstraktion auf höher dimensionale Räume kann zu verschiedenen Zeitpunkten erfolgen. Sie bietet sich vor allem vor der experimentellen Phase in Schritt 6 an, um Bilder aus mehr als drei Pixeln betrachten zu können.

### 5.2.4 Vergleich und Fazit der Anwendbarkeit beider Lernmethoden in der Vermittlung mathematischer Modellierung

Die Diskussion zu den möglichen mathematischen Anknüpfungspunkten an das Schulwissen der Schüler verdeutlicht, dass die mathematischen Hintergründe der Lernmethode SVM mehr Überschneidungen mit den laut Lehrplan des Landes Nordrhein-Westfalen zu vermittelnden mathematischen Inhalten der Sekundarstufe II aufweisen als die Methode über die SVD (KLP Mathematik Sek. II). Die Durchführung eines Lernmoduls basierend auf der SVM bedarf damit weniger Vermittlung neuen Wissens. Der Fokus kann auf die Anwendung von bereits bekannten mathematischen Konzepten bei der Bearbeitung der konkreten Problemstellung gelegt werden. Mit Blick auf das Ziel eines Lernmoduls, den Prozess der mathematischen Modellierung in den Vordergrund zu stellen und die kreative Entwicklung von Modellen sowie die Anwendung mathematischer Methoden zu fördern, erscheint die Verwendung der Lernmethode SVM bei der Konzipierung eines Workshops vielversprechender einsetzbar.

Insgesamt konnte aufgezeigt werden, dass ein nicht geringer Anteil der mathematischen Hintergründe der untersuchten Lernmethoden auf mathematischen Ideen und Konzepten beruht, die den Schülern zumindest im Anschauungsraum bekannt sein sollten. Damit stellt die Problemstellung der automatisierten Klassifizierung und deren Bearbeitung mithilfe maschineller Lernmethoden einen geeigneten Kandidaten für die computergestützte mathematische Modellierung mit Schülern dar - sowohl aufgrund ihrer Zugänglichkeit mit Schulwissen als auch aufgrund ihrer Relevanz für zahlreiche Anwendungen und Fragestellungen aus dem Alltag. Kritisch zu hinterfragen bleibt, inwieweit die konkrete Problemstellung der Bildklassifizierung als Einstiegsproblem für ein Lernmodul sinnvoll einsetzbar ist, da sich aus diesem Bereich womöglich schwer reale, authentische Daten des $\mathbb{R}^2$ bzw. $\mathbb{R}^3$ finden lassen. Auf mögliche Klassifizierungsprobleme aus anderen Anwendungsbereichen wird nachfolgend ein Ausblick gegeben.

# 6 Ausblick

Dieses Kapitel gibt zunächst einen Ausblick auf Möglichkeiten, wie die mathematischen Modelle der untersuchten maschinellen Lernmethoden weiter optimiert werden können - mit dem Ziel im Bereich der Bildklassifizierung höhere Klassifizierungserfolge zu erzielen. Im Rahmen dessen werden weiterführend sinnvolle und interessante Experimente aufgezeigt. Da diese Arbeit als Grundlage für die Entwicklung eines Lernmoduls zur Klassifizierung auf Basis maschineller Lernmethoden dienen soll, wird anschließend ein Klassifizierungsproblem in den Blick genommen, das nicht aus dem Bereich der Bildklassifizierung stammt, welches jedoch für die mathematische Modellierung mit Schülern vielversprechend erscheint. Zudem werden weitere Gestaltungs- und Umsetzungsideen für die Erstellung eines Lernmoduls umrissen.

Wie in Abschnitt 4.1.3 erwähnt, bestehen verschiedene Möglichkeiten, die beiden untersuchten maschinellen Lernmethoden zu optimieren. Bei der SVM kann dazu eine systematische Optimierung des Parameters $C$ durchgeführt werden, um das Modell unter Verwendung des linearen Kerns zu verbessern. Zudem könnten Experimente mit verschiedenen Mehrklassenalgorithmen durchgeführt werden. Weiterhin könnten andere Kernfunktionen, wie der Gauß'sche RBF Kern oder ein polynomialer Kern, eingesetzt werden. Dies scheint insbesondere mit Blick auf die Literatur vielversprechend, da dort Klassifizierungserfolge von über 98% dokumentiert sind, die unter Verwendung von polynomialen Kernfunktionen 9-ten Grades sowie dem RBF Kern erzielt wurden (vgl. Decoste & Schölkopf, 2002, S. 178).

Bei der Methode der SVD könnte vor der eigentlichen Klassifizierung eine Vorverarbeitung der Bilder durchgeführt und deren Einfluss auf den Klassifizierungserfolg ermittelt werden. Bspw. könnten die Bilder, wie bei den Experimenten zur Methode der SVM geschehen, standardisiert werden. Weiterhin kann die Anzahl an Singulärvektoren, die für die Approximation der Bildbasen der einzelnen Klassen verwendet werden, derart optimiert werden, dass eine unterschiedliche Anzahl an Singulärvektoren für jede der Bildklassen zugelassen würde. Damit wären sämtlich Kombinationen der Anzahl an Singulärvektoren der einzelnen Klassen zu untersuchen und die Kombination mit dem maximalen Klassifizierungserfolg zu wählen. Überdies könnte die Aufgabe der Bestimmung der *optimalen* Anzahl an Singulärvektoren auch zunächst auf der Grundlage von Theorien aus dem Bereich der inversen Probleme bearbeitet und das mathematische Modell anschließend entsprechend der theoretischen Ergebnisse angepasst werden.

Im Bereich der Gesichtsklassifizierung mit eigens generierten Datensätzen wären Experimente mit stärker variierenden Bildern einer Gesichterklasse interessant. Ein solcher Datensatz kann generiert werden, indem sich die Personen auf den Bildern bewegen oder indem die Belichtung verändert wird. Auch wären Manipulationen der Bilder, wie Drehungen, Spiegelungen oder Verzerrungen, möglich.
Da bei der Verwendung der SVM in der Gesichtsklassifizierung auf dem Yale B Datensatz keine guten Ergebnisse erzielt wurden, stellt eine weitere Vorverarbeitung der

Bilder eine interessante Untersuchung dar. Im Rahmen dessen könnten vor dem Support Vektor Training Methoden zum Einsatz kommen, die wesentliche Eigenschaften der Bilder einer Klasse extrahieren. Eine Möglichkeit stellt eine Extraktionsmethode dar, bei der die Bilder als Histogramme orientierter Gradienten (engl. histogram of oriented gradients feature extraction) dargestellt werden (vgl. Chapelle et al., 1999, S. 1). Weiterführend interessant wäre im Rahmen der Herausarbeitung wesentlicher Bildeigenschaften die Kombination von SVD und SVM. Mittels SVD könnten wichtige bzw. *informationsreiche* Eigenschaften der Bildklassen extrahiert und anschließend die Klassifizierung mit der SVM vorgenommen werden.

Mit Blick auf die Entwicklung eines Lernmoduls für Schüler bieten sich neben der Bildklassifizierung auch andere Klassifizierungsprobleme an, die einen hohen Lebensweltbezug bzw. eine große Relevanz für den Alltag (von Schülern) aufweisen. Eine Möglichkeit wäre die Anwendung der maschinellen Lernmethoden auf Daten aus sozialen Netzwerken. Einen vielversprechenden Ansatzpunkt stellt der frei verfügbare Datensatz des Netzwerks *friendster*[25] dar, zu dem von Sube (2016) bereits ein Lernmodul entwickelt wurde. Anknüpfend an das bereits bestehende Lernmodul, welches Klassifizierungsprobleme mithilfe von Heuristiken bearbeitet, ließen sich die Lernmethoden SVM oder SVD anwenden. Basierend auf gegebenen Nutzerdaten könnten Klassifikatoren entwickelt werden, die die Zuordnung der Nutzer zu verschiedenen Klassen hinsichtlich interessierender Eigenschaften vornehmen. Solche interessierenden Eigenschaften könnten das Alter oder die politische, sexuelle oder religiöse Orientierung der Nutzer darstellen.

Insbesondere da sich reale Klassifizierungsprobleme mit authentischen Daten aus dem 2- oder 3-dimensionalen Anschauungsraum womöglich leichter im Bereich der sozialen Netzwerke als in der Bildklassifizierung finden lassen, bietet sich die Konzipierung eines Lernmoduls in diesem Kontext an. Ein Klassifizierungsproblem zu einem realen Datensatz des $\mathbb{R}^2$ aus dem Bereich der sozialen Netzwerke könnte wie folgt lauten:
Angenommen es soll die Klassifizierung hinsichtlich der *sexuellen Orientierung* vorgenommen werden. Ein Nutzer wird dann durch einen Vektor $x \in \mathbb{R}^2$ repräsentiert, dessen Komponenten bestimmte Eigenschaften dieses Nutzers widerspiegeln. Beispielsweise könnte der erste Eintrag den Anteil der homosexuellen Freunde des Nutzers darstellen und der zweite Eintrag den Anteil der homosexuellen Freunde der Freunde.[26] Ein auf Basis dieser Daten entwickeltes Modell kann dann genutzt werden, um die sexuelle Orientierung auch bei Nutzern vorherzusagen, die diese nicht explizit in ihrem Profil angegeben haben. Eine solche Problemstellung zu Daten von sozialen Netzwerken würde damit zugleich die sensible Thematik der Datensicherheit aufgreifen.

[25] `www.friendster.com/`, Stand: 10.02.2018

[26] Diese Eigenschaften werden auch bei dem bereits bestehenden Lernmodul von Sube (2016) untersucht und führen unter Verwendung der auf Heuristiken basierenden Lernmethode zu Klassifizierungserfolgen von über 90%.

Neben der Wahl anderer lebensnaher und relevanter Klassifizierungsprobleme könnten zudem verschiedene weitere maschinelle Lernmethoden betrachtet, experimentell untersucht und auf ihre Eignung für die Vermittlung mathematischer Modellierung mit Schülern sowie der Anknüpfungspunkte an das Schulwissen diskutiert werden. Kandidaten dafür wären u. a. die k-Nearest-Neigbour-Methode oder Neuronale Netze.

Abschließend sei auf einen weiteren Durchführungs- und Gestaltungsrahmen eines mathematischen Modellierungsworkshops zu der Problemstellung der automatisierten (Bild-)Klassifizierung mit Schülern hingewiesen: eine computergestützte mathematische Modellierungswoche, wie bspw. die CAMMP week[27] des Schülerlabors CAMMP der RWTH Aachen. Unterstützt von wissenschaftlichen Betreuern könnten kleine Schülerteams die Problemstellung *unangeleitet*, d. h. ohne zuvor didaktisch-methodisch ausgearbeitete Aufgabenstellungen zu einer explizit gewählten maschinellen Lernmethode bearbeiten. Mit fachlicher Unterstützung sollen sie kreativ ihre eigene Herangehensweise an die Problemstellung finden und eigenständig ein Modell zur Klassifizierung entwickeln. Es wäre durchaus interessant zu sehen, wie sie ihr mathematisches Wissen einsetzen, um Klassifizierungsprobleme dieser Art anzugehen und zu lösen.

[27] `https://blog.rwth-aachen.de/cammp/angebote/`, Stand: 10.02.2018

# Literatur


Baum, M., Bellstedt, M. & et al. (2011). *Lambacher Schweizer Mathematik Qualifikationsphase Leistungskurs/Grundkurs.* Stuttgart: Ernst Klett Verlag.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics).* Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Blum, W. (2006). Modellierungsaufgaben im Mathematikunterricht – Herausforderung für Schüler und Lehrer. In A. Büchter, H. Humenberger, S. Hußmann & S. Prediger (Hrsg.), *Realitätsnaher Mathematikunterricht – vom Fach aus und für die Praxis. Festband für Hans-Wolfgang Henn zum 60. Geburtstag.* Hildesheim: Verlag Franzbecker.

Blum, W. (2007). Mathematisches Modellieren - Zu schwer für Schüler und Lehrer? In *Beiträge zum Mathematikunterricht 2007.* Hildesheim, Berlin: Verlag Franzbecker.

Büchter, A. & Leuders, T. (2011). *Mathematik selbst entwickeln: Lernen fördern - Leistung überprüfen* (Cornelsen, Hrsg.).

Burges, C. J. C. (1998, Juni). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*, *2* (2), 121–167. Zugriff am 12.10.2017 auf `https://doi.org/10.1023/A:1009715923555`

Chapelle, O., Haffner, P. & Vapnik, V. N. (1999, Sep). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, *10* (5), 1055-1064.

Chu, M. T., Funderlic, R. E. & Plemmons, R. J. (2003). Structured low rank approximation. *Linear Algebra and its Applications*, *366*, 157 - 172. Zugriff am 12.02.2018 auf `http://www.sciencedirect.com/science/article/pii/S0024379502005050`

Chu, M. T. & Golub, G. H. (2007). *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications.* Oxford University Press.

Cover, T. M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, *EC-14* (3), 326-334.

Dahmen, W. & Reusken, A. (2006). *Numerik für Ingenieure und Naturwissenschaftler.* Berlin: Springer-Verlag.

Decoste, D. & Schölkopf, B. (2002, Jan). Training Invariant Support Vector Machines. *Machine Learning*, *46* (1), 161–190. Zugriff am 02.01.2018 auf `https://doi.org/10.1023/A:1012454411458`

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M. & Herrera, F. (2016, Nov). Big data preprocessing: methods and prospects. *Big Data Analytics*, *1* (1), 9. Zugriff am 10.02.2018 auf https://doi.org/10.1186/s41044-016-0014-0

Golub, G. H. & Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press.

Greefrath, G., Kaiser, G., Blum, W. & Borromeo Ferri, R. (2013). Mathematisches Modellieren – Eine Einführung in theoretische und didaktische Hintergründe. In R. Borromeo Ferri, G. Greefrath & G. Kaiser (Hrsg.), *Mathematisches Modellieren für Schule und Hochschule.* Wiesbaden: Springer-Verlag.

Greefrath, G. & Weitendorf, J. (2013). Modellieren mit digitalen Werkzeugen. In R. Borromeo Ferri, G. Greefrath & G. Kaiser (Hrsg.), *Mathematisches Modellieren für Schule und Hochschule.* Wiesbaden: Springer-Verlag.

Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc.

Heitzer, J. (2012). *Orthogonalität und Approximation. Vom Lotfällen bis zum JPEG-Format. Von der Schulmathematik zu modernen Anwendungen*. Wiesbaden: Springer Spektrum.

Henn, H.-W. & Filler, A. (2015). *Didaktik der Analytischen Geometrie und Linearen Algebra*. Berlin, Heidelberg: Springer Spektrum.

Kultusministerkonferenz. (2012). *Bilungsstandards im Fach Mathematik für die Allgemeine Hochschulreife.* Zugriff am 12.12.2017 auf www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2012/2012_10_18-Bildungsstandards-Mathe-Abi.pdf

LeCun, Y. & Cortes, C. (2010). *MNIST handwritten digit database.* Zugriff am 09.02.2018 auf http://yann.lecun.com/exdb/mnist/

Leuders, T. (2011). Chancen und Risiken des Computereinsatzes im Mathematikunterricht. In T. Leuders (Hrsg.), *Mathematikdidaktik Praxishandbuch für die Sekundarstufe I und II.* Berlin: Cornelsen Verlag.

Martínez-de Pisón, F. J., Barreto, C., Pernía, A. & Alba, F. (2008). Modelling of an elastomer profile extrusion process using support vector machines (SVM). *Journal of Materials Processing Technology*, *197* (1), 161 - 169. Zugriff am 14.02.2018 auf http://www.sciencedirect.com/science/article/pii/S0924013607006036

Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen. (2014). *Kernlehrplan für die Sekundarstufe II Gymnasium / Gesamtschule in Nordrhein-Westfalen – Mathematik.* Zugriff am 05.01.2018 auf https://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/m/KLP_GOSt_Mathematik.pdf

Mitchell, T. M. (1997). *Machine Learning* (1. Aufl.). New York, NY, USA: McGraw-Hill, Inc.

Muller, N., Magaia, L. & Herbst, B. M. (2004). Singular Value Decomposition, Eigenfaces, and 3D Reconstructions. *SIAM Review*, *46* (3), 518-545. Zugriff am 10.12.2017 auf `https://doi.org/10.1137/S0036144501387517`

Platt, J. C. (1998). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines* (Bericht). Adavances in kernel methods - Support Vector Learning. Zugriff am 10.02.2018 auf `www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.pdf`

Reuters. (2017). *De Maizière verlängert umstrittene Tests zur Gesichtserkennung.* Zugriff am 09.02.2018 auf `www.spiegel.de/netzwelt/netzpolitik/bahnhof-berlin-suedkreuz-testlauf-zur-gesichtserkennung-wird-verlaengert-a-1183528.html`

Roeckerath, C., Schönbrodt, S., Richter, P. & Frank, M. (2017). Wie funktioniert eigentlich GPS? - Ein Computergestützter Modellierungsworkshop. (unveröffentlicht)

Schölkopf, B. (1998, Juli). SVMs — a practical consequence of learning theory. *IEEE Intelligent Systems and their Applications*, *13* (4), 18-21.

Schölkopf, B. & Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press.

Strang, G. (1993). The Fundamental Theorem of Linear Algebra. *The American Mathematical Monthly*, *100* (9), 848-855. Zugriff am 12.02.2018 auf `http://www.jstor.org/stable/2324660`

Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press. Zugriff am 14.02.2018 auf `http://math.mit.edu/~gs/linearalgebra/`

Sube, M. (2016). Wie sicher ist meine Privatsphäre in sozialen Netzwerken? ... und was hat das mit Mathe zu tun? *Masterarbeit, RWTH Aachen.*

Winter, H. (1995). Mathematikunterricht und Allgemeinbildung. *Mitteilungen der Gesellschaft für Didaktik der Mathematik*, *61*, 37 - 46. Zugriff am 31.01.2018 auf `https://doi.org/10.1515/dmvm-1996-0214`

Wrobel, S., Joachims, T. & Morik, K. (2013). Maschinelles Lernen und Data Mining. In G. Görz, J. Schneeberger & U. Schmid (Hrsg.), *Handbuch der Künstlichen Intelligenz* (S. 1–18). Oldenbourgh Verlag.