



Mathematische Grundlagen der Künstlichen Intelligenz im Schulunterricht

Chancen für eine Bereicherung des Unterrichts in linearer Algebra

Sarah Schönbrodt  · Thomas Camminady · Martin Frank

Eingegangen: 11. Februar 2021 / Angenommen: 1. September 2021
© Der/die Autor(en) 2021

Zusammenfassung Ein Anspruch des mathematischen Modellierungsunterrichts in der Schule sollte es sein, besonders aktuelle Probleme und interessante neue Technologien aus dem Alltag der Schüler/innen einzubeziehen. Dies gilt insbesondere, wenn sie eine didaktische Reduktion auf elementare (schul-)mathematische Inhalte leicht zulassen. Künstliche Intelligenz (KI) zieht sich durch verschiedene Bereiche von Wissenschaft und Technik und verbirgt sich insbesondere hinter zahlreichen Anwendungen unseres Alltags.

In diesem Beitrag wird diskutiert, wie ein zeitgemäßer Mathematikunterricht durch die Modellierung realer, schülernaher Probleme aus dem Bereich KI bereichert werden kann. Dazu werden zwei Methoden und deren didaktische Reduktion für den Einsatz in einem computergestützten Mathematikunterricht vorgestellt.

Bei der problemorientierten Diskussion beider Methoden werden zwei alltägliche Problemstellungen in den Blick genommen: Zum einen Klassifizierungsprobleme und deren Lösung mithilfe der sogenannten Stützvektormethode (SVM), die auf der Berechnung des Abstandes von Punkten zu Hyperebenen beruht; zum anderen Empfehlungssysteme, die auf einer Matrix-Faktorisierung basieren können.

Zu beiden Problemstellungen wurden digitale Lernmaterialien für Oberstufenschüler/innen entwickelt, die im Rahmen von eintägigen Workshops zur mathematischen Modellierung bereits mehrfach erprobt wurden. Die digitale Umsetzung als Jupyter Notebooks wird abschließend beschrieben und steht den Leser/innen als Open Educational Resources unter einer Creative Commons Lizenz zur Verfügung.

Schlüsselwörter Maschinelles Lernen · Data Science · Künstliche Intelligenz · Mathematische Modellierung · Digitale Lernmaterialien · Mathematikunterricht

Sarah Schönbrodt (✉) · Thomas Camminady · Martin Frank
Karlsruher Institut für Technologie (KIT), Hermann-von-Helmholtz-Platz 1,
76344 Eggenstein-Leopoldshafen, Deutschland
E-Mail: schoenbrodt@kit.edu

1 Künstliche Intelligenz im Mathematikunterricht?

Zahlreiche Anwendungen und Technologien basieren auf Methoden des Maschinellen Lernens (ML): Sei es, wenn Netflix, Amazon und Co ihren Nutzer/innen neue Produkte vorschlagen, wenn Mails automatisch als Spam erkannt werden oder sich unser Smartphone per Gesichtserkennung entsperrt.

Was sich hinter diversen ML-Methoden verbirgt sind mathematische Ideen, Konzepte und Aussagen, die in elementarer Form in zahlreichen Lehrplänen in Deutschland verankert sind. Dies macht das Themengebiet ML zu einem aussichtsreichen Kandidaten für die mathematische Modellierung realer Probleme mit Schüler/innen – und damit auch zu einer hervorragenden Möglichkeit, die Bedeutung von Mathematik für Gesellschaft, Alltag und Technik erfahrbar zu machen. Eine entsprechende Gestaltung von Lerngelegenheiten und Lernmaterialien zu Problemstellungen aus dem Bereich KI kann damit einen Beitrag dazu leisten, insbesondere die erste und dritte Wintersche Grunderfahrung, die der Mathematikunterricht anstreben und verknüpfen sollte, anzusprechen. Die Grunderfahrungen Winters [25] lauten:

„Der Mathematikunterricht sollte anstreben, die folgenden drei Grunderfahrungen, die vielfältig miteinander verknüpft sind, zu ermöglichen:

- (1) 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,
- (2) mathematische Gegenstände und Sachverhalte, repräsentiert in Sprache, Symbolen, Bildern und Formeln, als geistige Schöpfungen, als eine deduktiv geordnete Welt eigener Art kennen zu lernen und zu begreifen,
- (3) in der Auseinandersetzung mit Aufgaben Problemlösefähigkeiten, die über die Mathematik hinaus gehen, (heuristische Fähigkeiten) zu erwerben.

Das Wort Erfahrung soll zum Ausdruck bringen, dass das Lernen von Mathematik weit mehr sein muss als eine Entgegennahme und Abspeicherung von Information, dass Mathematik erlebt (möglicherweise auch erlitten) werden muss.“

Gängige Strategien des Maschinellen Lernens sind *überwachtes*, *unüberwachtes* und *bestärkendes* Lernen. In diesem Beitrag beschäftigen wir uns mit zwei Methoden des überwachten Lernens. Bei dieser Art des Lernens sind nicht nur die Eingangsdaten (*Input*) gegeben, sondern auch die zugehörigen Ergebnisse (*Output*). Am Beispiel der Klassifizierung von Verkehrsschildern könnten die Eingangsdaten Bilder von Stoppschildern (Klasse 1) und Vorfahrtsschildern (Klasse 2) sein. Der zugehörige Output ist die Zuweisung eines jeden Bildes zu seiner entsprechenden Klasse.

Der gesamte Datensatz wird unterteilt in einen Trainings- und einen Testdatensatz. Basierend auf den Trainingsdaten wird ein Modell *gelernt*. Anschließend werden die Testdaten genutzt, um zu validieren, wie gut das Modell für die Anwendung auf neue (unbekannte) Daten geeignet ist. Dazu wird das Modell auf die Testdaten angewandt und das berechnete Ergebnis mit dem tatsächlichen Output verglichen.

Die Anzahl der Publikationen und Projekte, die sich der Aufarbeitung des Themas KI bzw. ML für Schüler/innen von unterschiedlichen Standpunkten widmen, sind im Vergleich zur fachwissenschaftlichen Forschung in diesem Bereich in den vergangenen Jahren nur langsam gestiegen. Bei vielen der Schülerprojekte liegt der Fokus stärker auf der informatischen, der anwendungsbezogenen oder der ethischen Diskussion (s. bspw. [5]). Vorschläge für Lernmaterialien oder gar die Umsetzung dieser für den Mathematikunterricht sind hingegen rar. Erste mathematik-didaktische Annäherungen an das Thema werden u. a. durch das Projekt ProDaBi¹ (Künstliche Neuronale Netze und Entscheidungsbäume [15, 17]) wie auch im Rahmen des Projektes CAMMP² (Heuristiken zur Vorhersage von Nutzereigenschaften in Sozialen Netzwerken, Bildklassifizierung, Empfehlungssysteme [20, 22, 24]) vorangetrieben.

In diesem Beitrag wird an zwei Problemklassen aufgezeigt, welche schulmathematischen Anknüpfungspunkte aufgegriffen werden können, um gemeinsam mit Oberstufenschüler/innen in die Problemlösestrategien des (überwachten) Maschinellen Lernens einzusteigen. Bei dieser exemplarischen Diskussion wird deutlich, dass zwar der Einsatz des Computers unabdinglich ist, um die betrachteten datenlastigen Probleme zu lösen, dass der Kern der ML-Methoden jedoch einfache mathematische Modelle sind; ein Grund warum KI bzw. ML nicht nur (aber auch!) vom informatischen, gesellschaftswissenschaftlichen oder ethischen Standpunkt mit Schüler/innen diskutiert werden sollte, sondern insbesondere auch eine Legitimation und entsprechende didaktische Aufbereitung für die Behandlung im Mathematikunterricht erfahren sollte.

Zu den in den folgenden beiden Abschnitten vorgestellten Problemstellungen und mathematischen Modellen haben die Autoren dieses Beitrags digitales Lernmaterial (umgesetzt als Jupyter Notebooks, s. Abschn. 4) für den Einsatz im Mathematikunterricht oder in interdisziplinären Projekttagen bzw. -zeiten o. ä. entwickelt. Dieses ermöglicht Oberstufenschüler/innen eine aktive Beschäftigung mit dem Thema ML und der problemorientierten Entwicklung ausgewählter mathematischer Modelle. Der Aufbau des Lernmaterials und der Zugriff auf dieses wird in Abschn. 4 beschrieben.

Wie im Folgenden herausgearbeitet wird, bietet die Modellierung beider Problemstellungen diverse Anknüpfungspunkte an schulmathematische Inhalte. Bei der Lösung von Bildklassifizierungsproblemen über die Stützvektormethode (SVM) kommen

- Vektoren,
- Geraden, Ebenen (allgemeiner: Hyperebenen),
- Abstände zwischen Punkten und Geraden / Ebenen,
- das Skalarprodukt,
- Funktionen und
- Optimierungsprobleme

zum Einsatz. Bei der Entwicklung von Empfehlungssystemen basierend auf einer Matrix-Faktorisierungsmethode spielen

¹ www.prodabi.de, letzter Aufruf am 28.12.2020.

² www.cammp.online, letzter Aufruf am 28.12.2020.

- Vektoren und Matrizen³
- das Skalarprodukt,
- lineare Gleichungssysteme,
- Funktionen,
- Differenzierung und Optimierungsprobleme

eine wesentliche Rolle – zahlreiche schulmathematische Konzepte⁴, die eine konkrete Anwendung finden und denen in einem realen Kontext Bedeutung verliehen wird („aha, Mathe ist also doch nützlich“).

Wenngleich der Schwerpunkt des vorliegenden Beitrags und auch des konzipierten Lernmaterials auf Inhalten der Linearen Algebra (Vektoren, Skalarprodukt, Matrizen, (Trenn-)Hyperebenen) liegt, so greifen die entwickelten Modelle dennoch Themen aus den Bereichen Analysis oder Stochastik auf. Durchaus ist eines der Ziele des entwickelten Lernmaterials, dass die Bedeutung von (in der Schule seltener erkennbaren) innermathematischen Vernetzungen für das Lösen realer Probleme erfahrbar gemacht wird.

Die entwickelten Materialien können sowohl zum Einstieg in eine Unterrichtsreihe (bspw. bei der Einführung von Ebenen und Geraden im Raum) oder losgelöst von einem spezifischen mathematischen Thema zur Vertiefung bereits gelernter Inhalte und deren Anwendung Einsatz finden. Es ist auch denkbar, mathematische Inhalte, die den Lernenden noch nicht bekannt sind, bei der Bearbeitung der Materialien problemorientiert einzuführen. Dies hat sich u. a. bei Matrizen (bei der Modellierung von Empfehlungssystemen) oder bei der Einführung der Normalenform von Ebenen (bei der Bearbeitung von Klassifizierungsproblemen) als möglich erwiesen. Um eine verständige Bearbeitung des Materials zu erlauben, werden Vektoren, das Skalarprodukt und ein Verständnis von funktionalen Zusammenhängen jedoch als Vorwissen vorausgesetzt.

Werden bei der Bearbeitung der Problemstellungen neue Inhalte erarbeitet, seien es solche, die noch im Mathematikunterricht zu behandeln sind, oder solche, die darüber hinaus einen Ausblick auf hochschulmathematische Inhalte liefern, so kann das Material lediglich der Einführung und ersten Erkundung dienen. Um neues Wissen bzw. Können auch nachhaltig zu festigen, wären weiterführende Übungen und Anwendungen des Gelernten (bspw. die Bestimmung von Extrema mehrdimensionaler Funktionen oder der Umgang und das Rechnen mit Matrizen) sinnvoll oder gar notwendig. Ziel der bereits durchgeführten Schülerworkshops war es nicht, dass die Lernenden die Anwendung neuer Inhalte unabhängig vom gewählten Kontext beherrschen. Vielmehr sollten Selbstwirksamkeitserwartungen gestärkt und die Bedeutung von Mathematik für die Lebenswelt der Lernenden durch die aktive Bearbeitung der Problemstellung hervorgehoben werden.

³ Matrizen sind nicht (mehr) in allen Mathematiklehrplänen der Sekundarstufe II in Deutschland enthalten. Sie werden im Lernmaterial jedoch problemorientiert eingeführt und stellen keine notwendige Voraussetzung für die verständige Bearbeitung des Materials dar.

⁴ Durchaus ist an dieser Stelle anzumerken, dass nicht alle genannten Inhalte in den Kernlehrplänen sämtlicher Bundesländer verankert sind. Exemplarisch sei an dieser Stelle auf die Lehrpläne Baden-Württembergs [9], Bayerns [21], Nordrhein-Westfalens [10] oder Rheinland-Pfalz [8] verwiesen, die abgesehen von Matrizen alle genannten Inhalte ausweisen.

Unabhängig von möglichen mathematisch-inhaltlichen Lernzielen (Erarbeitung oder Anwendung von Inhalten) wurde das digitale Lernmaterial im Hinblick auf folgende weitere Zielsetzungen entwickelt und eingesetzt:

- Die Lernenden sollen die Bedeutung von Mathematik für das Lösen realer Probleme basierend auf ML-Methoden erfassen, indem sie angeleitet durch kleinere Aufgaben mathematische Formeln entwickeln und problemorientiert Beispiele rechnen. Sie sollen diese Bedeutung auch bei zukünftigen Diskussionen über ML-Anwendungen in Gesellschaft und Medien einordnen können.
- Durch gestufte Hilfen zu einzelnen Aufgaben, vertiefendes Informationsmaterial und Zusatzaufgaben sollen die Lernenden in heterogenen Lerngruppen selbstbestimmt und im eigenen Lerntempo an den Aufgaben arbeiten können. Zur Förderung des eigenständigen Arbeitens werden die Lösungen der Lernenden nach der Eingabe in digitale Arbeitsblätter automatisch überprüft und individuelle Rückmeldungen ausgegeben.
- Die Lernenden bearbeiten die Materialien in Kleingruppen und tauschen sich über Ideen und Fragen bei der durch Aufgaben angeleiteten Entwicklung der Modelle aus. Es soll deutlich werden, dass Teamarbeit für das Lösen realer Problemstellungen hilfreich oder gar notwendig ist. Nicht nur durch den Austausch in den Kleingruppen, sondern auch durch Plenumsdiskussionen, bei denen Ergebnisse gesichert, Fragen aufgeworfen und Ideen der Lernenden diskutiert werden, sollen Team- und Kommunikationsfähigkeit gefördert werden.
- Die Lernenden reflektieren verschiedene Anwendungsmöglichkeiten der beschriebenen Modelle kritisch. Unter anderem werden ethisch bedenkliche Anwendungen und deren gesellschaftliche Auswirkungen aufgezeigt und die Erfahrungen der Lernenden eingebunden.

2 Klassifizierungsprobleme und die Stützvektormethode

Ausgangspunkt stellt das Problem der Zuordnung von unterschiedlichen Objekten zu verschiedenen Klassen dar – eine Problemstellung, die sich hinter zahlreichen Anwendungen verbirgt. Das Beispiel, mit dem wir uns in diesem Beitrag befassen, ist die Klassifizierung von Bildern. Objekte auf Bildern sollen erkannt und ihrer entsprechenden *Klasse* zugewiesen werden: Sei es die Zuordnung von Gesichtern zu Personen oder die Klassifizierung von Verkehrszeichen im Bereich des autonomen Fahrens.

Bei der Entwicklung eines Klassifizierungsmodells und dem anschließenden Validieren des Modells folgen wir der Idee des *überwachten Lernens*:

1. Ein Datensatz, bei dem die Zuordnung der Bilder zu den einzelnen Klassen bekannt ist, wird unterteilt in:
 - Trainingsdaten, die zum Lernen des Zuordnungsmodells genutzt werden und
 - Testdaten, die zur Bewertung der Güte des Modells eingesetzt werden.
2. Basierend auf den Trainingsdaten wird ein Modell (konkret eine Funktion) gelernt. Zum Finden einer geeigneten Funktion kommt die Stützvektormethode (engl. *Support Vector Machine*) zum Einsatz. Die Funktion soll so gewählt werden, dass an-

schließend auch Bilder, deren Klassenzuordnung unbekannt ist, möglichst oft der korrekten Klasse zugewiesen werden.

3. Die gefundene Zuordnungsfunktion wird auf die Testdaten angewandt. Von den Testbildern ist die tatsächliche Klassenzuordnung ebenfalls bekannt. Diese wird jedoch zunächst ausgeblendet und die Testbilder mit dem Zuordnungsmodell klassifiziert. Anschließend werden vorhergesagte und tatsächliche Klassenzuordnung verglichen.
4. Ist die Zuordnungsfunktion ausreichend gut, kann das Modell zur Klassifizierung unbekannter Bilddaten angewendet werden. Andernfalls müssen andere oder mehr Bilder verwendet und/oder das Modell verbessert werden.

Bevor ein Modell für die Bildklassifizierung entwickelt werden kann, muss ein Verständnis für die mathematische Darstellung von Bildern geschaffen werden.

2.1 Wie beschreiben wir ein Bild mathematisch?

Der Einfachheit halber betrachten wir Schwarzweißbilder. Die Bilder bestehen aus einzelnen Pixeln mit Grauwerten. Diese liegen im Bereich von 0 (schwarz) bis 255 (weiß). Ein Bild kann also zunächst als eine Tabelle oder ein Raster (vgl. Abb. 1)

Abb. 1 Schwarzweißbild und Grauwerte der Pixel

224	89	92	75	43	61	32	24	23	23	23	23	23	23	23	23	23
231	100	68	52	48	38	46	73	95	57	51	29	25	26	26		
221	73	66	67	101	126	186	212	206	201	194	116	43	28	30		
225	92	49	123	183	207	219	235	230	223	220	201	137	58	35		
223	59	72	184	218	220	218	220	225	226	230	215	199	120	46		
218	56	123	195	223	218	211	223	212	213	229	231	217	182	76		
217	56	174	208	229	168	123	241	237	115	178	235	222	199	107		
218	81	196	214	228	207	180	222	223	192	215	237	227	208	111		
218	103	209	212	223	214	225	219	220	228	221	231	228	207	110		
219	61	120	157	199	203	220	112	119	221	219	215	165	165	128		
220	59	53	143	179	193	220	88	82	217	214	185	107	56	70		
224	73	63	177	194	200	176	135	136	166	194	200	117	63	86		
227	90	85	193	215	214	196	163	159	196	216	208	116	91	111		
228	98	140	207	227	228	220	209	215	228	231	215	102	111	121		
230	102	182	219	227	226	213	211	225	229	222	201	108	128	128		
230	124	201	225	228	226	218	223	224	224	226	193	124	139	138		
229	174	202	222	218	202	182	184	211	228	226	179	149	151	150		
232	185	216	220	205	174	162	183	223	232	220	179	165	161	160		
235	184	221	223	203	180	171	217	226	228	204	181	171	169	166		
236	176	231	223	190	180	212	232	229	203	192	187	176	172	172		

mit l Zeilen und m Spalten aufgefasst werden. Schreiben wir die Pixelspalten alle untereinander, so liefert dies für jedes Bild eine Repräsentation als k -Tupel bzw. Vektor des k -dimensionalen Raums, wobei $k = l \cdot m$ gilt.

Nun wollen wir nicht behaupten, dass ein Vektor in einem hochdimensionalen Vektorraum ein für Schüler/innen leicht verständliches Konzept darstellt. Reduzieren wir das Problem jedoch auf wenige Dimensionen, so landen wir im bekannten zwei- oder dreidimensionalen Vektorraum; und damit bei einem mit Oberstufenmathematik zugänglichen mathematischen Modell.

Beginnen wir also mit einem Beispiel aus dem zwei- oder dreidimensionalen Raum: Bilder bestehend aus lediglich zwei oder drei Pixeln. Dies ergibt im Hinblick auf reale Bilder zunächst einmal wenig Sinn (diese bestehen schließlich aus deutlich mehr Pixeln), jedoch entspricht es einer gängigen Vorgehensweise in der Mathematik beim Lösen von realen, komplexen Problemen: Ein Problem wird zunächst so weit vereinfacht, dass es mit vorhandenen (schul-)mathematischen Fähigkeiten angehbar wird. Anschließend wird das Modell Stück für Stück erweitert und dem realen Problem angenähert.

2.2 Entwicklung eines Klassifizierungsmodells – die SVM

Das Lernmaterial ist anhand dreier Leitfragen strukturiert, die im Folgenden beantwortet werden [18, 20]:

1. Wie kann ein Klassifizierungsmodell entwickelt werden mit dem Schwarzweißbilder von Ampeln korrekt als „grün“ oder „rot“ klassifiziert werden können? (Daten des 2-dimensionalen Anschauungsraums)
2. Wie kann das Modell erweitert werden, um zudem die gelbe Ampelphase berücksichtigen zu können? (Daten des 3-dimensionalen Vektorraums)
3. Wie lässt sich das entwickelte mathematische Modell auf die Klassifizierung von handgeschriebenen Ziffern oder von Bildern mit Gesichtern erweitern? (Daten eines hoch-dimensionalen Vektorraums)

Wir widmen uns zunächst der ersten Leitfrage und betrachten damit ein Klassifizierungsproblem bestehend aus zwei Klassen. Zudem bleiben wir im zweidimensionalen Anschauungsraum, was am Beispiel der Klassifizierung von Ampelphasen die Betrachtung eines Bildes bestehend aus zwei Pixeln (ein Pixel des oberen Ampellichts und ein Pixel des unteren Ampellichts) bedeuten würde (vgl. Abb. 2).

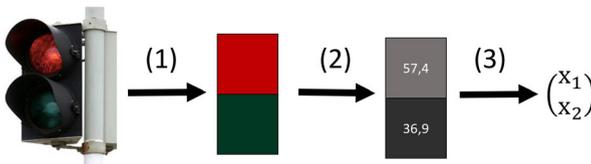
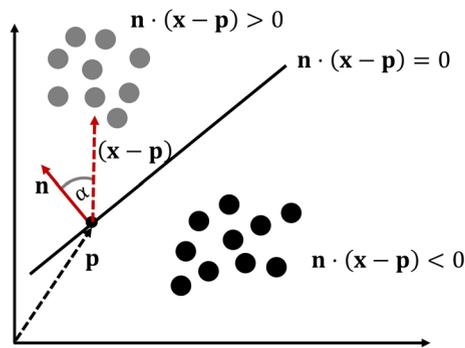


Abb. 2 Bild einer auf rot stehenden Ampel. (1) Die Ampel wird durch zwei Pixel modelliert (ein Pixel des oberen und ein Pixel des unteren Ampellichts). (2) Die farbige Darstellung (RGB-Werte) wird in Graustufen (Grauwerte) überführt. (3) Die Grauwerte werden in einem Vektor gespeichert

Abb. 3 Visualisierung des Winkels, den der Vektor $(\mathbf{x} - \mathbf{p})$ mit dem Normalenvektor \mathbf{n} einschließt



Zudem treffen wir die **Annahme**, dass sich die Datenpunkte \mathbf{x}_i für $i = 1, \dots, N$ der beiden Klassen, wie in Abb. 4 skizziert, linear separieren lassen. In diesem Fall kann per Definition eine Gerade zwischen die Punktwolken der beiden Klassen gelegt werden. Mathematisch kann diese *Trenngerade* in der – in zahlreichen Lehrplänen verankerten⁵ – Normalenform $\mathbf{n} \cdot \mathbf{x} = b$ bzw. $\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0$ mit Normalenvektor \mathbf{n} und einem beliebigen Punkt \mathbf{p} auf der Geraden dargestellt werden (vgl. Abb. 3).

Im entwickelten Schülerworkshop zeichnen die Lernenden zunächst eine beliebige Trenngerade zwischen die gegebenen Punktwolken und beschreiben diese durch eine lineare Funktion f der Form $f(x) = m \cdot x + c$. Anschließend stellen sie diese in der Normalenform dar. Ist den Lernenden der Zusammenhang zwischen den beiden Darstellungsformen noch nicht (oder nicht mehr) bekannt, erhalten sie vertiefendes Lernmaterial. Je nach Vorwissen der Schüler/innen soll so bereits vorhandenes Wissen angewandt und geübt oder Neues erarbeitet werden.

Wir analysieren nun die Lage der Punkte der beiden Klassen bezüglich der Trenngeraden. Werden Punkte in die Geradengleichung eingesetzt, die auf der Seite der Geraden liegen in die der Normalenvektor zeigt, so resultiert dies in einen Wert mit positivem Vorzeichen. Einsetzen von Punkten der anderen Klasse liefert hingegen stets einen Wert mit negativem Vorzeichen. Dies wird leicht durch Anwendung der Definition des Skalarprodukts ersichtlich [20].

Ist der Winkel zwischen Normalenvektor und dem Vektor $(\mathbf{x} - \mathbf{p})$ für einen beliebigen Datenpunkt \mathbf{x} kleiner als 90° , liefert dies ein positives Vorzeichen. Der Punkt liegt auf der Seite, in dessen Richtung der Normalenvektor zeigt (vgl. Abb. 3). Bei einem Winkel von 90° liegt der Punkt auf der Trenngeraden. Ist der Winkel hingegen größer als 90° liegt der Punkt auf der anderen Seite der Geraden und das Ergebnis weist ein negatives Vorzeichen auf [20]. Diese Fallunterscheidung wird im Schülerworkshop auch von den Lernenden erarbeitet. Dazu stehen den Lernende verschiedene Skizzen und optionale Hilfekarten zur Verfügung.

Insgesamt haben wir bereits ein Modell entwickelt, welches zur Klassifizierung genutzt werden kann: Wir wählen eine lineare Trennfunktion f , die die Daten der Klassen voneinander trennt, setzen zu klassifizierende Datenpunkte ein, und weisen

⁵ Die Normalenform ist u. a. Bestandteil der Mathematiklehrpläne in Baden-Württemberg [9], Bayern [21], Nordrhein-Westfalen [10] oder Rheinland-Pfalz [8].

die Punkte gemäß ihres Vorzeichen der entsprechenden Klasse zu. Formal kann festgehalten werden:

$$f : \mathbb{R}^2 \longrightarrow \{-1, 0, 1\}$$

$$\mathbf{x} \mapsto f(\mathbf{x}) = \text{sgn}(\mathbf{n} \cdot (\mathbf{x} - \mathbf{p})) = \text{sgn}(\mathbf{n} \cdot \mathbf{x} + b) \tag{1}$$

mit $b = -\mathbf{n} \cdot \mathbf{p}$ und der Vorzeichenfunktion sgn , die definiert ist als

$$\text{sgn}(z) = \begin{cases} -1 & \text{für } z < 0 \\ 0 & \text{für } z = 0 \\ 1 & \text{für } z > 0. \end{cases} \tag{2}$$

Grundsätzlich können diverse Funktionen als Trenngeraden verwendet werden (vgl. Abb. 4). Welche ist jedoch besser geeignet? Anders gefragt: Welche Gerade trennt die Punkte der beiden Klassen *am besten*? Intuitiv würde man womöglich die Gerade auswählen, die den größtmöglichen Abstand zu den beiden Punktwolken lässt, d. h. die einen maximalen datenpunktfreien Bereich, der auch Margin genannt wird, aufweist (vgl. Abb. 5).

Abb. 4 Linear separierbare Daten

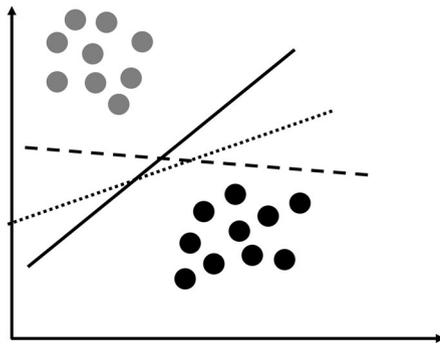
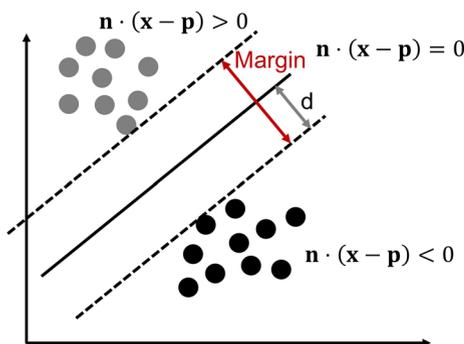


Abb. 5 Skizze des datenpunktfreien Bereichs (Margin)



Wir sind bei einem **Optimierungsproblem unter Nebenbedingungen** angelangt, welches von den Lernenden im Schülerworkshop zunächst in Worten formuliert wird:

Finde die Gerade, die die Daten der beiden Klassen trennt und die zudem den größtmöglichen Abstand zu den nächstgelegenen Datenpunkten beider Klassen aufweist.

Um das Problem mathematisch formulieren und anschließend lösen zu können, wird die Zuordnung der N Datenpunkte zu den beiden Klassen zunächst durch die Variablen t_i für $i = 1, \dots, N$ (sog. *Labels*) beschrieben, wobei $t_i = +1$ für die Datenpunkte der einen Klasse und $t_i = -1$ für alle Datenpunkte der anderen Klasse gelten soll.

Zudem wird der Abstand d zwischen den Datenpunkten und der Trennfunktion durch

$$d = \frac{|\mathbf{n} \cdot \mathbf{x}_i + b|}{\|\mathbf{n}\|_2} \quad (3)$$

mathematisch beschrieben (vgl. Abb. 5). Im Lernmaterial leiten die Schüler/innen die Formel (3) für den Abstand eigenständig her [20]. Dazu wenden sie die Definition des Skalarprodukts auf die Vektoren $(\mathbf{x} - \mathbf{p})$ und \mathbf{n} an:

$$\cos(\alpha) = \frac{\mathbf{n} \cdot (\mathbf{x} - \mathbf{p})}{\|\mathbf{n}\|_2 \cdot \|(\mathbf{x} - \mathbf{p})\|_2}. \quad (4)$$

Hierbei entspricht α dem Winkel, der von den Vektoren $(\mathbf{x} - \mathbf{p})$ und \mathbf{n} eingeschlossen wird. Zudem nutzen sie die Definition des Kosinus im rechtwinkligen Dreieck:

$$\cos(\alpha) = \frac{d}{\|(\mathbf{x} - \mathbf{p})\|_2}. \quad (5)$$

Gleichsetzen der rechten Seiten von Gleichung (4) und (5) und anschließendes Umformen liefert Gleichung (3). Diese Herleitung wird im Lernmaterial durch Skizzen unterstützt. Zudem können die Lernenden auf gestufte Hilfen zugreifen. Ziel ist das problemorientierte Anwenden bereits bekannter mathematischer Konzepte (Skalarprodukt, Kosinus).

Der Margin ist nun durch den doppelten Abstand der *nächstgelegenen* Datenpunkte gegeben:

$$2 \cdot \min_i \frac{|\mathbf{n} \cdot \mathbf{x}_i + b|}{\|\mathbf{n}\|_2}. \quad (6)$$

Ziel ist es den Margin zu maximieren und dabei die Nebenbedingungen (eine Klasse links, die andere Klasse rechts der Trenngeraden) zu berücksichtigen. Formal ist damit das Optimierungsproblem

$$\max_{\mathbf{n}, b} \left\{ \min_i \frac{|\mathbf{n} \cdot \mathbf{x}_i + b|}{\|\mathbf{n}\|_2} \right\}. \quad (7)$$

unter den Nebenbedingungen

$$\begin{aligned} \mathbf{n} \cdot \mathbf{x}_i + b &\geq 0 \text{ für } t_i = +1 \\ \mathbf{n} \cdot \mathbf{x}_i + b &\leq 0 \text{ für } t_i = -1, \end{aligned} \quad (8)$$

die für alle Trainingsdaten \mathbf{x}_i , $i = 1, \dots, N$ gelten müssen, zu lösen. Die Ungleichungen können zusammengefasst werden zu

$$t_i (\mathbf{n} \cdot \mathbf{x}_i + b) \geq 0, \quad i = 1, \dots, N. \quad (9)$$

Zudem können die Parameter \mathbf{n} und b der Gerade so skaliert werden, dass

$$\min_i |\mathbf{n} \cdot \mathbf{x}_i + b| = 1 \quad (10)$$

gilt. Das Optimierungsproblem kann damit vereinfacht werden zu

$$\max_{\mathbf{n}, b} \frac{1}{\|\mathbf{n}\|_2} \quad \text{bzw. äquivalent zu} \quad \min_{\mathbf{n}, b} \frac{1}{2} \|\mathbf{n}\|_2^2 \quad (11)$$

unter den Nebenbedingungen

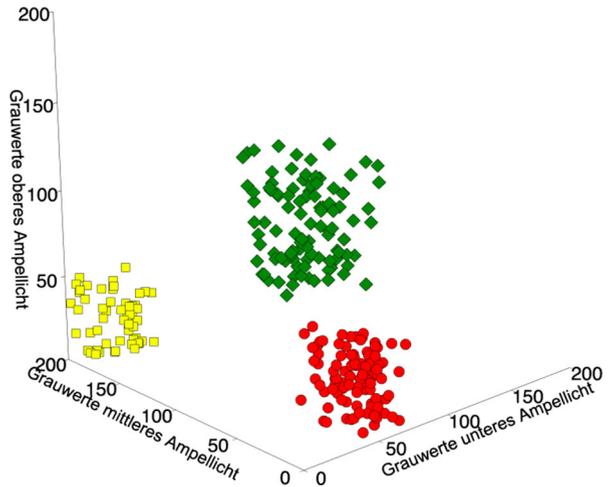
$$t_i (\mathbf{n} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, N. \quad (12)$$

Dabei handelt es sich um ein Optimierungsproblem mit quadratischer Zielfunktion und linearen Ungleichungsnebenbedingungen. Insgesamt liegt ein konvexes Optimierungsproblem vor. Jede lokale Lösung dieses Optimierungsproblems entspricht damit gleichzeitig einer globalen Lösung [1]. Wie dieses Problem in der Praxis gelöst wird, wird in diesem Beitrag und in dem Schülerworkshop nicht weiter thematisiert. Interessierte Leser/innen seien beispielsweise auf Bishop [1], Boyd und Vandenberghe [2] oder Nocedal und Wright [13] verwiesen.

Das Lösen des Optimierungsproblems, welches im Lernmaterial mithilfe dem Computer überlassen wird, liefert eine Darstellung des Normalenvektors als Linearkombination der Trainingsdaten, wobei nur die Daten in die Lösung eingehen, die der Trenngerade am nächsten liegen. Diese Vektoren werden auch Stützvektoren genannt – daher der Name Stützvektormethode.

Die gefundene Trennfunktion kann nun auf Testdaten angewandt und validiert werden. Anschließend wird im Schülerworkshop die Erweiterung auf den dreidimensionalen Fall vorgenommen. Dazu wird am Beispiel der Ampel auch das gelbe Ampellicht hinzugenommen. Ein Datenpunkt ist dann durch einen Vektor des \mathbb{R}^3

Abb. 6 Datenpunkte der drei Ampelphasen grün, gelb und rot

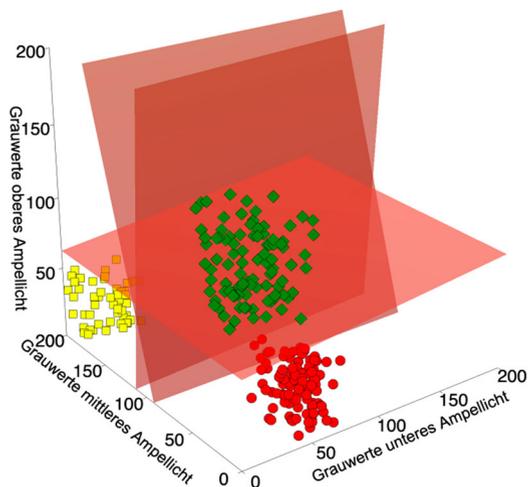


gegeben, dessen Einträge die Grauwerte des oberen, mittleren und unteren Lichts einer Ampel darstellen (vgl. Abb. 6). Die Schüler/innen erweitern ihr Modell für diesen dreidimensionalen Fall (von der Trenngerade zur Trennebene). Da bisher lediglich ein Modell für binäre Klassifikationen entwickelt wurde, werden zunächst weiterhin lediglich zwei Klassen (z. B. rot und nicht-rot) gegenübergestellt.

2.2.1 Mehrklassen-Klassifizierung

Sollen nun mehrere Klassen (bspw. alle drei Ampelphasen) unterschieden werden, so kann für jede Kombination von zwei Klassen ein binärer Klassifikator berechnet wer-

Abb. 7 Für je zwei Klassen wurden die Trennebenen berechnet und visualisiert



den. Unbekannte Datenpunkte werden dann mit jedem der binären Klassifikatoren klassifiziert und der Klasse zugeteilt, der sie am häufigsten zugewiesen wurden (vgl. Abb. 7). Diese Vorgehensweise wird auch *One-Versus-One*-Klassifizierung genannt [1, S. 338]. Im Schülerworkshop haben die Lernenden zunächst die Möglichkeit selbst eine Strategie für Klassifizierungsprobleme mit mehr als zwei Klassen zu formulieren. Anschließend werden die Strategien der Lernenden diskutiert. Zudem wird das *One-Versus-One*-Verfahren erläutert und auf das Ampelbeispiel angewendet.

2.2.2 Hoch-dimensionale Daten

Im letzten Teil des Lernmaterials wenden die Schüler/innen ihr Modell auf ein Klassifizierungsproblem an, welches in einem höherdimensionalen Vektorraum lebt (Trennebenen werden dann zu Trennhyperebenen). Dazu nehmen sie entweder Bilder ihrer Gesichter auf, erstellen einen eigenen Datensatz und nutzen den entwickelten Lernalgorithmus zur automatischen Gesichtserkennung – eine Anwendung, die sie höchstwahrscheinlich von ihren Smartphones kennen. Alternativ kann ein Satz handgeschriebener Ziffern aufgenommen und der Lernalgorithmus für die Klassifizierung von Zahlen angewandt werden. Die Erweiterung auf Bilder mit k Pixeln und damit auf den k -dimensionalen Raum erfolgt im Lernmaterial problemorientiert und verdeutlicht den Lernenden, dass bekannte Konzepte auf höherdimensionale Räume übertragbar und für das Lösen komplexer Probleme anwendbar sind. Bei beiden Anwendungen – Gesichtsklassifizierung oder Ziffernerkennung – arbeiten die Lernenden mit kleinen Datensätzen (ca. 10–20 Trainingsdatenpunkte/Bilder pro Klasse). Die Klassifizierung funktioniert im Lernmaterial gut, da sich die Testbilder in beiden Fällen kaum von den Trainingsbildern unterscheiden. Bei der Aufnahme von Trainings- und Testdaten für die Gesichtsklassifizierung wird das Gesicht in einem vorgegebenen Kasten positioniert. Bei der Ziffernerkennung wird die Zahl in einem Widget am Computer gezeichnet. Werden die Gesichter oder Ziffern des Testdatensatzes jedoch in leicht anderen Positionen aufgenommen bzw. gezeichnet, so funktioniert die Klassifizierung deutlich schlechter, da die Modelle explizit zur Erkennung von Mustern, wie sie in den Trainingsdaten auftauchen, trainiert wurden. Dies wird im folgenden Abschnitt erneut aufgegriffen.

2.2.3 Modellerweiterungen und kritische Diskussion

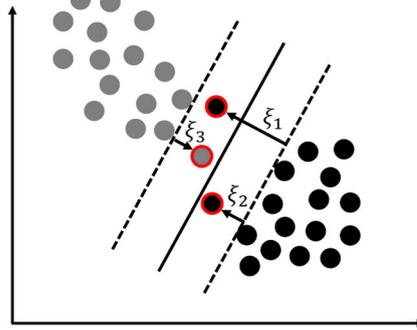
Abschließend lassen sich auf anschaulicher Ebene noch verschiedene Modellerweiterungen, Grenzen sowie Gefahren mit den Lernenden diskutieren:

- **Einbau von Schlupfvariablen:** Um die Anwendung der SVM auch für sich leicht überlappende Punktwolken zweier Klassen zu ermöglichen, kann das Optimierungsproblem modifiziert werden [1, S.330]. Dazu werden die Nebenbedingungen (12) durch den Einbau von Schlupfvariablen ξ_i für $i = 1 \dots N$ zu

$$t_i (\mathbf{n} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{13}$$

abgeschwächt. Diese erlauben einzelnen Datenpunkten innerhalb des Margins oder gar auf der falschen Seite der Hyperebene zu liegen (vgl. Abb. 8). Zudem

Abb. 8 Einführung von Schlupfvariablen für sich leicht überlappende Daten zweier Klassen



wird die Zielfunktion (11) erweitert, indem ein Kostenfaktor $C > 0$ eingeführt wird und die Schlupfvariablen berücksichtigt werden. Insgesamt liefert dies die zu minimierende Funktion

$$\frac{1}{2} \|\mathbf{n}\|_2^2 + C \cdot \sum_{i=1}^N \xi_i. \quad (14)$$

Über den Parameter C kann der Einfluss des „Schlupfes“ (auch Slack genannt) und damit zugleich eine Überanpassung des Modells an die Trainingsdaten reguliert werden. Im Schülerworkshop wird der Einfluss des Parameters C auf eine Unter- bzw. Überanpassung an die gegebenen Daten experimentell erkundet, indem die Lernenden den Parameter systematisch variieren.

- **Der Kern-Trick:** Nichtlinear separierbare Daten werden, wie in Abb. 9 dargestellt, durch eine Kernfunktion in einen höher-dimensionalen Raum überführt, in dem sie mit größerer Wahrscheinlichkeit linear separierbar sind [19, S. 200]. Der Kern-Trick wird im entwickelten Lernmaterial bisher nicht thematisiert.
- **Grenzen des Modells:** Wird ein gefundenes Zuordnungsmodell auf Daten angewandt, die kaum Ähnlichkeit zu den Bildern im Trainingsdatensatz aufweisen, so führt dies zu schlechten Klassifizierungsergebnissen. Die SVM „braucht“ zum

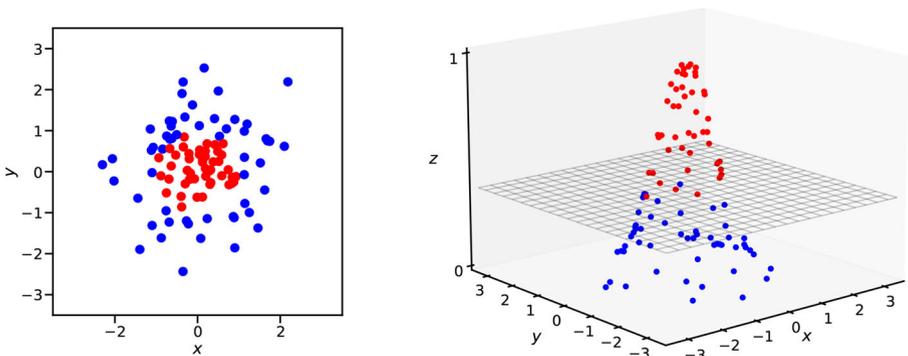


Abb. 9 Der Kern Trick: Überführung der Daten in einen höher-dimensionalen Raum

Lernen die Art der Bilder, die letztlich auch zum Testen bzw. in der späteren Anwendung genutzt werden. Wurden zum Lernen eines Gesichtsklassifizierungsmodells bspw. nur Frontalaufnahmen von Gesichtern verwendet, dann werden Bilder mit Seitenaufnahmen der Gesichter wahrscheinlich nicht korrekt erkannt.

Eine Diskussion über Grenzen des Modells und auch über den Einfluss von verfügbaren Lerndaten (Anzahl, Qualität) sollte auch in der Durchführung mit Lernenden geführt werden. Dabei kann diskutiert werden, dass die Datensätze divers und ausreichend groß sein sollten, um die Modelle robuster bzw. performanter zu machen.

- **Nutzen und Gefahren:** Bildklassifizierungsmodelle sollten mit Schüler/innen auch kritisch diskutiert werden. Dabei kann auf die unterschiedlichen Anwendungen solcher Modelle (Überwachungssysteme, Social-Credit-Systeme, Entsperrung des eigenen Smartphones, ...) eingegangen und ethische Aspekte und Gefahren für den Missbrauch solcher Systeme diskutiert werden.

Welchen Einfluss der Einsatz von ML-Modellen auf großen Datensätzen hat und inwieweit solche Modelle bei einem unreflektierten Einsatz Diskriminierungen und soziale wie auch finanzielle Benachteiligung verstärken können, diskutiert u. a. O'Neil [14] an zahlreichen Beispielen (Vergabe von Krediten, Zulassungen zum College etc.). Solche Diskussionen auch mit Lernenden zu führen scheint in der heutigen Zeit wesentlich.

3 Die Netflix Challenge und Empfehlungssysteme

Netflix, Amazon und zahlreiche weitere Unternehmen versuchen ihre Nutzer/innen durch individuelle Empfehlungen für neue Produkte bzw. Filme an sich zu binden. Um den Geschmack der Nutzer/innen bzw. deren Vorlieben zu modellieren, setzen die Unternehmen Informationen über die Nutzer/innen und insbesondere über ihr bisheriges Kauf- bzw. Nutzungsverhalten ein.

Im Jahr 2006 schrieb der Streamingdienst Netflix einen Wettbewerb aus. Ziel dieses Wettbewerbs war die Entwicklung eines Modells, welches Vorhersagen konnte an welchen Filmen ein/e Nutzer/in zukünftig Interesse haben könnte. Mit anderen Worten musste ein Empfehlungssystem (engl. *Recommender System*) entwickelt werden, welches Nutzerinteressen um 10% besser vorhersagen konnte als Netflix' damaliges System. Im Zuge des Wettbewerbs veröffentlichte Netflix einen (theoretisch⁶) anonymen Datensatz von 480.189 Nutzer/innen, 17.700 Filmen und 100.480.507 zugehörigen Bewertungen von eins (schlechteste Bewertung) bis fünf (beste Bewertung), die die Nutzer/innen für die Filme abgegeben haben [3]. Der Datensatz ist mit lediglich 1,18% bekannten Bewertungen extrem dünn besetzt.

Netflix veröffentlichte nicht nur den genannten Datensatz mit bekannten Bewertungen, sondern zudem einen Quiz-Datensatz mit Nutzer/innen und Filmen deren Bewertungen Netflix zwar kannte, die jedoch nicht öffentlich gemacht wurden. Das Team, dessen Empfehlungssystem die Bewertungen der ausgewiesenen Nutzer/innen

⁶ Im Jahr 2008 zeigten zwei Forscher der Universität Texas, dass die De-Anonymisierung des Datensatzes durch Kombination mit einem weiteren öffentlich verfügbaren Filmdatensatz teilweise möglich ist [11].

im Quiz-Datensatz um 10% (gemessen über die Wurzel der gemittelten Fehlerquadratsumme) genauer vorhersagen konnte als Netflix' eigenes System es vermochte, hatte Chancen auf den Hauptpreis von 1 Million US Dollar. Dieser Preis ging im Jahr 2009 an das Team *BellKor's Pragmatic Chaos* [3, S. 203].

Die nachfolgend vorgestellte und didaktisch reduzierte Methode einer Matrix-Faktorisierung, wurde von dem siegreichen Team (in Kombination mit weiteren Methoden, die hier nicht beschrieben werden) umgesetzt. Der Netflix-Datensatz ist frei im Internet verfügbar⁷ und wurde für die Entwicklung von digitalem Lernmaterial für Schüler/innen heruntergeladen und aufbereitet (vgl. Abschn. 4).

Ausgangspunkt für die Entwicklung des Empfehlungssystems ist die Bewertungsmatrix $R \in \mathbb{R}^{N \times M}$ bei der jede Zeile für eine/n Nutzer/in und jede Spalte für einen Film steht. Hat User i den Film j bereits gesehen und eine Bewertung abgegeben, so hat der Eintrag $r_{i,j}$ der Matrix R einen ganzzahligen Wert zwischen 1 und 5, welcher der abgegebenen Bewertung entspricht. Ist die Bewertung des Nutzers i für Film j hingegen nicht bekannt, ist der Eintrag $r_{i,j}$ gleich 0. Ein möglicher Auszug aus dem Datensatz könnte, wie in der folgenden Bewertungsmatrix R dargestellt, aussehen:

$$R = \begin{matrix} & \begin{matrix} \text{🎬}_1 & \text{🎬}_2 & \text{🎬}_3 & \text{🎬}_4 & \text{🎬}_5 & \text{🎬}_6 & \dots \end{matrix} \\ \begin{matrix} \text{👤}_1 \\ \text{👤}_2 \\ \text{👤}_3 \\ \text{👤}_4 \\ \text{👤}_5 \\ \vdots \end{matrix} & \begin{pmatrix} 3 & 0 & 1 & 0 & 1 & 4 & \dots \\ 0 & 2 & 4 & 1 & 3 & 1 & \dots \\ 3 & 1 & 0 & 3 & 0 & 0 & \dots \\ 4 & 3 & 0 & 4 & 4 & 0 & \dots \\ 4 & 0 & 0 & 4 & 0 & 5 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{matrix} \quad (15)$$

In diesem Beispiel wurde vereinfacht ein weniger dünn besetzter Teil des Datensatzes betrachtet. Diese Modellvereinfachung wird auch in der Folge beibehalten, da Zusammenhänge in den Daten so leichter veranschaulicht werden können.

Die Grundidee hinter unserem Empfehlungssystem ist eine Matrix-Faktorisierung: Die Bewertungsmatrix $R \in \mathbb{R}^{N \times M}$ soll durch ein Produkt aus zwei niedrigdimensionalen Matrizen $U \in \mathbb{R}^{N \times F}$ und $M \in \mathbb{R}^{F \times M}$ approximiert werden. Mit anderen Worten werden die Vektoren, die die User (Zeilen der Bewertungsmatrix) bzw. Filme (Spalten der Bewertungsmatrix) repräsentieren, in einen niedrigdimensionalen Eigenschaftsraum überführt [4]. Der Geschmack der Nutzer/innen und die Eigenschaften der Filme werden so basierend auf ihren bisherigen Bewertungen implizit modelliert. Die Methode fällt deswegen unter die Technik des Kollaborativen Filterns [7, S. 31].

Im Folgenden wird ein Überblick über die Daten des Netflix-Datensatzes gegeben und eine Matrix-Faktorisierungsmethode entwickelt. Dabei werden immer wieder Kommentare zur möglichen didaktischen Reduktion der Inhalte und zur Umsetzung in bereits erprobtem Lernmaterial eingestreut.

⁷ Zum Beispiel auf Kaggle: www.kaggle.com/netflix-inc/netflix-prize-data, letzter Aufruf am 02.01.2021.

3.1 Entwicklung eines Empfehlungssystems

Verschaffen wir uns zunächst einen Eindruck von den Daten und möglichen interessanten Zusammenhängen, die wir für die Entwicklung eines Empfehlungssystems ausnutzen können. Dazu betrachten wir folgende beispielhafte Bewertungsmatrix von vier Nutzer/innen und sechs Filmen, deren Bewertungen allesamt bekannt seien:

$$R = \begin{pmatrix} 3 & 1 & 3 & 1 & 1 & 4 \\ 1 & 2 & 1 & 4 & 3 & 1 \\ 3 & 1 & 3 & 1 & 1 & 4 \\ 4 & 3 & 4 & 5 & 4 & 5 \end{pmatrix} \tag{16}$$

Die grundlegende Idee zur Erklärung des Datensatzes ist, dass Filme mit ähnlicher Bewertung gewisse Ähnlichkeiten (bspw. das gleiche Genre, den gleichen berühmten Hauptdarsteller o. ä.) aufweisen. Diese und ähnliche Zusammenhänge in der Bewertungsmatrix können ausgenutzt werden, um ein Modell zu entwickeln, welches unbekannte Bewertungen von Nutzer/innen vorhersagt.

In einem ersten Schritt nehmen wir an, dass sich die Filme leicht bezüglich zweier Eigenschaften einordnen lassen. In dem kleinen Beispiel könnten diese Eigenschaften die Genres **Action** (A) und **Comedy** (C) sein.

Angenommen in unserem Fall beinhalten die sechs Filme folgende Anteile an Action und Comedy (1 = gering bis 5 = hoch):

$$M = \begin{matrix} & \begin{matrix} \text{F}_1 & \text{F}_2 & \text{F}_3 & \text{F}_4 & \text{F}_5 & \text{F}_6 \end{matrix} \\ \begin{matrix} A \\ C \end{matrix} & \begin{pmatrix} 3 & 1 & 3 & 1 & 1 & 4 \\ 1 & 2 & 1 & 4 & 3 & 1 \end{pmatrix} \end{matrix} \tag{17}$$

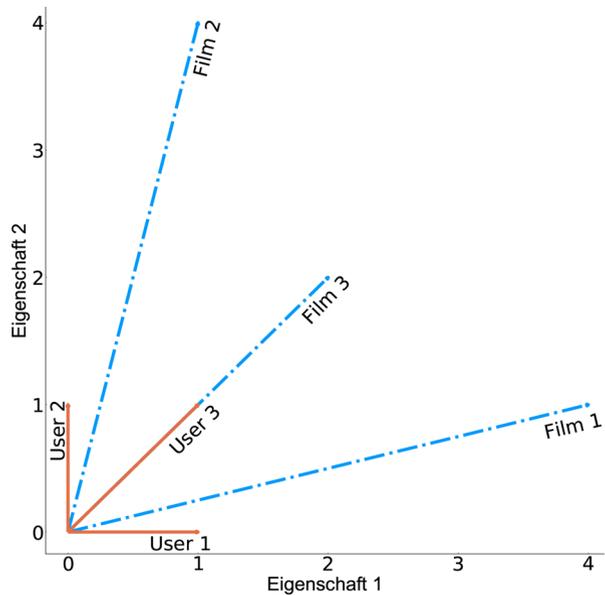
Wir speichern diese Informationen in einer Matrix $M \in \mathbb{R}^{2 \times 6}$, die im Folgenden als Moviematrix bezeichnet wird. Diese Matrix lässt sogleich Rückschlüsse auf den Geschmack der vier Nutzer/innen zu: Wird der Geschmack zunächst als binäres Merkmal aufgefasst, so kann abgeleitet werden, ob ein/e Nutzer/in Comedy bzw. Action mag (= 1) oder nicht (= 0). Dies wird im Lernmaterial auch von den Schüler/innen erwartet. Wir speichern diese Informationen in der Usermatrix $U \in \mathbb{R}^{4 \times 2}$:

$$U = \begin{matrix} & \begin{matrix} A & C \end{matrix} \\ \begin{matrix} \text{N}_1 \\ \text{N}_2 \\ \text{N}_3 \\ \text{N}_4 \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \end{matrix} \tag{18}$$

In diesem Beispiel liefert das Produkt der Usermatrix U und der Moviematrix M die Bewertungsmatrix R . Es gilt: $R = U \cdot M$.

Ist eine derartige Zerlegung der Bewertungsmatrix bekannt, kann diese genutzt werden, um auch unbekannte Bewertungen durch die Berechnung des Skalarpro-

Abb. 10 Darstellung von drei User-Vektoren und drei Movie-Vektoren im Eigenschaftsraum. Eigenschaften könnten beispielsweise Genres wie Action oder Comedy sein



dukts des entsprechenden Zeilenvektors der Usermatrix U und des Spaltenvektors der Moviematrix M vorherzusagen. Mit den Lernenden lässt sich anschaulich herleiten, wann bzw. warum das Skalarprodukt größere bzw. kleinere vorhergesagte Bewertungen liefert. Dazu können die Zeilenvektoren der Usermatrix und die Spaltenvektoren der Moviematrix visualisiert werden (vgl. Abb. 10).

Bei der Entwicklung eines Empfehlungssystems werden wir der Idee der Faktorisierung der Bewertungsmatrix weiter folgen. Dazu sind verschiedene Fragen zu beantworten:

- Die Quantifizierung der Eigenschaften der Filme (bspw. die Genres) und damit die Einträge der Matrix M sind nicht im Netflix-Datensatz enthalten. Wie kann dennoch (automatisiert) eine Zerlegung der Bewertungsmatrix in U und M bestimmt werden?
- Wie viele Eigenschaften sind relevant und sollten bei der Faktorisierung berücksichtigt werden?

Mit Schüler/innen lassen sich verschiedene Zerlegungen einer kleinen Bewertungsmatrix zunächst händisch berechnen, um zu erkennen, dass in manchen Fällen mehrere und in anderen womöglich gar keine Zerlegungen existieren. Dabei wird zugleich deutlich, dass die händische Berechnung äußerst aufwendig ist. Der Einsatz des Computers und die Entwicklung eines Verfahrens ist notwendig. Das zu entwickelnde Verfahren soll zwei Matrizen U und M liefern, deren Produkt die Bewertungsmatrix R ausreichend *gut* annähert.

An dieser Stelle ist zu definieren, wie die Güte einer Zerlegung bewertet und damit auch verschiedene Zerlegungen verglichen werden können. Mit anderen Worten ist

ein Maß zu definieren, welches den Fehler zwischen der Bewertungsmatrix R und der Vorhersage, gegeben durch das Produkt $U \cdot M$, quantifiziert.

3.1.1 Die Wahl der Fehlerfunktion

Durchaus wären bei der Definition eines Fehlermaßes auch seitens der Schüler/innen verschiedene Ansätze denkbar. Zum einen ließe sich die absolute Abweichung zwischen den einzelnen Bewertungen und der Vorhersage gemäß

$$\sum_{(i,j) \in \kappa} |r_{i,j} - \mathbf{u}_{i,\bullet} \mathbf{m}_{\bullet,j}| \tag{19}$$

aufsummieren. Hierbei bezeichnet κ die Menge der Paare (i, j) für die das Rating $r_{i,j}$ bekannt ist, $\mathbf{u}_{i,\bullet}$ bezeichnet den Zeilenvektor i der Matrix U (User-Vektor) und $\mathbf{m}_{\bullet,j}$ den Spaltenvektor j der Matrix M (Movie-Vektor).

Alternativ ließe sich der Fehler als Summe der Fehlerquadrate definieren:

$$\sum_{(i,j) \in \kappa} (r_{i,j} - \mathbf{u}_{i,\bullet} \mathbf{m}_{\bullet,j})^2. \tag{20}$$

Beide Formulierungen unterscheiden sich nicht nur im Hinblick auf ihre Differenzierbarkeit, auch der Einfluss von statistischen Ausreißern wird unterschiedlich gehandhabt: So gewichtet die L_2 Formulierung in Gleichung (20) Ausreißer stärker als es die L_1 Formulierung in Gleichung (19) tut. Die Wahl einer passenden Fehlerfunktion stellt im Hinblick auf die Diskussion mit Schüler/innen daher eine (mathematisch) interessante Modellentscheidung dar, die insbesondere bei dem nachfolgend beschriebenen Optimierungsverfahren erneut aufgegriffen werden kann.

Wir wählen für die weitere Modellierung die Summe der Fehlerquadrate. Die Suche nach einer passenden Faktorisierung kann dann als das folgende nichtlineare Optimierungsproblem formuliert werden

$$\min_{\mathbf{u}_{i,\bullet}, \mathbf{m}_{\bullet,j}} \sum_{(i,j) \in \kappa} (r_{i,j} - \mathbf{u}_{i,\bullet} \mathbf{m}_{\bullet,j})^2. \tag{21}$$

Zur Lösung dieses Problems mit Schüler/innen kann der Computer und ein geeignetes Lösungsverfahren als Black-Box eingesetzt werden. Gängige Lösungsverfahren sind *Stochastic Gradient Descent* (SGD) und *Alternating Least Squares* (ALS) [7, S. 33]. Alternativ kann eines der Verfahren gemeinsam mit leistungsstarken Lernenden entwickelt werden. Im Folgenden wird das ALS-Verfahren, welches verschiedene Anknüpfungspunkte an schulmathematische Inhalte bietet, und eine mögliche didaktische Reduktion dessen, beschrieben⁸.

⁸ Auch SGD scheint geeignet, um den Lernenden einen Einblick in gängige Optimierungsverfahren aus Praxis und Forschung zu geben. Wie bei dem ALS-Verfahren wären verschiedene Anknüpfungspunkte an schulmathematische Inhalte und solche, die darüber hinausgehen, denkbar.

3.1.2 Entwicklung eines Optimierungsverfahrens

Wie der Name verrät, wird bei der ALS-Methode alternierend ein lineares Ausgleichsproblem gelöst. Dazu werden iterativ zwei Schritte wiederholt:

Die User-Vektoren $\mathbf{u}_{i,\bullet}$ werden festgehalten und das erhaltene quadratische Optimierungsproblem nach den Movie-Vektoren $\mathbf{m}_{\bullet,j}$ gelöst. Die gefundene Lösung für die Movie-Vektoren wird festgehalten. Das quadratische Optimierungsproblem wird nun nach den User-Vektoren gelöst. Dies wird solange fortgeführt, bis der Fehler ausreichend klein ist oder die Optimierung auskonvergiert ist; wobei die Konvergenz gegen die global optimale Faktorisierung nicht gesichert ist [23, S. 99].

Bei der Entwicklung des Verfahrens mit Schüler/innen lässt sich an schulmathematisches Wissen aus dem Bereich der Analysis (Ableitung, Extremwertbestimmung) anknüpfen. Inhaltliche Voraussetzung für die Entwicklung und Anwendung des Verfahrens im Schülerworkshop ist Vorwissen zu linearen Gleichungssystemen und zur Berechnung von Extremwerten über die Nullstellen der Ableitung. Die Erarbeitung des Verfahrens mit den Lernenden zielt darauf ab, einen problemorientierten Einstieg in mehrdimensionale Funktionen, Gradienten und die Berechnung von Minima bzw. Maxima zu geben. Zugleich wird das Anwenden von Ableitungsregeln bei der Bestimmung der Gradienten und das Lösen von linearen Gleichungssystemen an einer konkreten Anwendung geübt.

Vor der Entwicklung des Verfahrens erhalten die Lernenden zunächst vertiefendes Informationsmaterial zu mehrdimensionalen Funktionen und Gradienten, welches die Übertragung von bekannten Konzepten aus der Differentialrechnung von Funktionen einer Variablen auf Funktionen mehrerer Variablen unterstützt. Es sei erneut erwähnt, dass dieser Teil des Lernmaterials für leistungsstarke Schüler/innen konzipiert wurde.

An dem folgenden Minimalbeispiel wird verdeutlicht, wie das Verfahren mit Lernenden erarbeitet werden kann:

3.1.3 Beispiel

Gesucht ist eine Zerlegung der Bewertungsmatrix R mit⁹

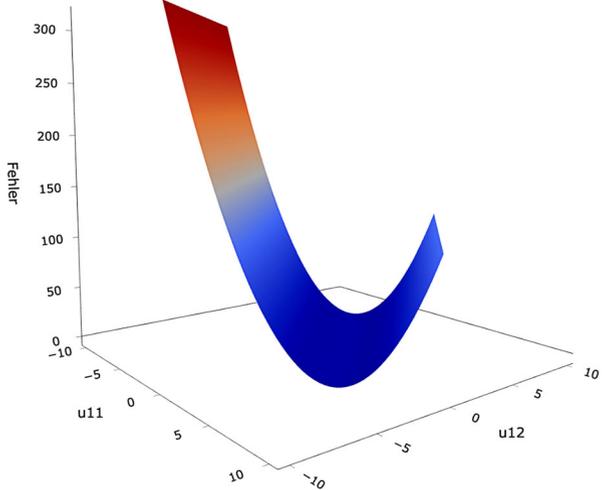
$$R = \begin{pmatrix} 2 & 3 \end{pmatrix} \quad (22)$$

in eine Usermatrix $U \in \mathbb{R}^{1 \times 2}$ und Moviematrix $M \in \mathbb{R}^{2 \times 2}$. Die Moviematrix kann im ersten Iterationsschritt beliebig festgelegt werden. In diesem Beispiel wird sie auf

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad (23)$$

⁹ In diesem Fall wäre $U = R$ und $M = \mathbb{I}$ eine triviale Zerlegung. Die hier durchgeführte Berechnung dient daher nur als Erklärung der allgemeinen Vorgehensweise. Die Dimensionalität ist bewusst klein gewählt, damit die Zielfunktion visualisierbar bleibt.

Abb. 11 Darstellung der Zielfunktion (24). Es existiert kein eindeutiges globales Minimum



gesetzt und das Minimum der Funktion f mit

$$f(u_{11}, u_{12}) = (2 - (u_{11} + u_{12}))^2 + (3 - (u_{11} + u_{12}))^2 \tag{24}$$

bestimmt (vgl. Abb. 11). Dazu bestimmen die Lernenden zunächst die beiden partiellen Ableitungen und damit den Gradienten. Die Berechnung des Minimums wird dann über die Nullstellen des Gradienten

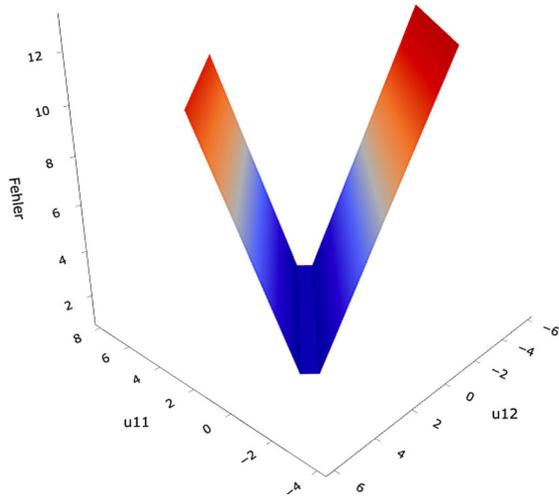
$$\nabla f = \begin{pmatrix} \text{Ableitung von } f \text{ bzgl. } u_{11} \\ \text{Ableitung von } f \text{ bzgl. } u_{12} \end{pmatrix} \tag{25}$$

$$= \begin{pmatrix} -2 \cdot (2 - (u_{11} + u_{12})) - 2 \cdot (3 - (u_{11} + u_{12})) \\ -2 \cdot (2 - (u_{11} + u_{12})) - 2 \cdot (3 - (u_{11} + u_{12})) \end{pmatrix} \stackrel{!}{=} 0 \tag{26}$$

angegangen. Dies liefert ein lineares Gleichungssystem, welches von den Lernenden händisch gelöst wird. Die Lösung ist in diesem Beispiel nicht eindeutig. Es existieren unendlich viele Lösungen. Durch die interaktive Betrachtung der dreidimensionalen graphischen Darstellung der Zielfunktion (24) (vgl. Abb. 11) können die Lernenden anschaulich erkunden, dass die Funktion kein eindeutiges Minimum besitzt. Die Lernenden wählen eine Lösung des Gleichungssystem (z.B. $u_{11} = 1$ und $u_{12} = 1.5$) und legen so die Werte für die Einträge der Usermatrix im nächsten Iterationsschritt fest. Dann führen die Lernenden analog die Optimierung der Einträge der Moviematrix durch. Sei erhalten die Zielfunktion für diesen Optimierungsschritt, bestimmen den Gradienten und berechnen dessen Nullstellen durch erneutes Lösen eines linearen Gleichungssystems. In diesem kleinen Beispiel führt bereits der zweite Iterationsschritt zu einer exakten Zerlegung der Bewertungsmatrix R .

Anmerkung: Bei der Entwicklung des Optimierungsverfahrens kann zudem problemorientiert diskutiert werden, welche Vorteile die Wahl der Summe der Fehlerquadrate gegenüber der Summe der absoluten Abweichungen für die Lösbarkeit des

Abb. 12 Darstellung der Zielfunktion (27) bei Wahl der Summe der absoluten Abweichungen als Fehlerfunktion



Optimierungsproblems liefert. Wurde die Summe der absoluten Abweichungen zur Berechnung des Fehlers gewählt, wird die Zielfunktion (24) zu

$$f(u_{11}, u_{12}) = |2 - (u_{11} \cdot 1 + u_{12} \cdot 1)| + |3 - (u_{11} \cdot 1 + u_{12} \cdot 1)|. \quad (27)$$

Mit Schüler/innen kann am Graphen dieser Funktion (vgl. Abb. 12) diskutiert werden, dass diese nicht überall differenzierbar ist, womit die Berechnung des Minimums komplexer wird.

Fernab von der schulmathematischen Formulierung des Optimierungsverfahrens kann das Problem in den einzelnen Iterationsschritten in Matrix-Vektor-Notation überführt werden. Exemplarisch wird dies an dem Iterationsschritt beschrieben, in dem die Moviematrix M optimiert wird. Das Problem kann in diesem Fall als folgendes lineares Ausgleichsproblem mit fester Usermatrix U notiert werden:

$$\min_{\mathbf{m}_{\bullet,j}} \| W_j \cdot (U \mathbf{m}_{\bullet,j} - \mathbf{r}_{\bullet,j}) \|_2^2. \quad (28)$$

wobei $W_j \in \mathbb{R}^{N \times N}$ definiert ist als

$$W_j = \text{diag}(d_1, \dots, d_N) \text{ mit } d_i = \begin{cases} 1 & \text{falls } (i, j) \in \kappa \\ 0 & \text{sonst,} \end{cases}$$

womit wir sicherstellen, dass in dem Minimierungsproblem nur Einträge der Bewertungsmatrix berücksichtigt werden, die bekannt sind (Null-Einträge von $\mathbf{r}_{\bullet,j}$ werden eliminiert, da die entsprechenden Ratings nicht bekannt sind).

Die Bestimmung des Minimums und damit die Berechnung der Nullstellen der Ableitung entspricht der Lösung des folgenden Normalgleichungssystems:

$$(\tilde{U}^T \tilde{U}) \mathbf{m}_{\bullet,j} = \tilde{U}^T \mathbf{r}_{\bullet,j} \tag{29}$$

mit $\tilde{U} = W_j \cdot U$.

Hat die Matrix \tilde{U} vollen Spaltenrang, so ist $(\tilde{U}^T \tilde{U})$ invertierbar. Die Lösung für einen beliebigen Movie-Vektor $\mathbf{m}_{\bullet,j}$ kann dann geschrieben werden als [23, S. 124]

$$\mathbf{m}_{\bullet,j} = (\tilde{U}^T \tilde{U})^{-1} \tilde{U}^T \mathbf{r}_{\bullet,j}. \tag{30}$$

3.1.4 Regularisierung

Um das lineare Ausgleichsproblem für Matrizen \tilde{U} ohne vollen Spaltenrang zu lösen, kann die Pseudoinverse von \tilde{U} berechnet werden. Stattdessen wird üblicherweise ein Regularisierungsterm in das Optimierungsproblem eingebaut. Das Optimierungsproblem (20) wird dann zu

$$\min_{\mathbf{u}_{i,\bullet}, \mathbf{m}_{\bullet,j}} \sum_{(i,j) \in \kappa} (r_{i,j} - \mathbf{u}_{i,\bullet} \cdot \mathbf{m}_{\bullet,j})^2 + \lambda \cdot (\|\mathbf{u}_{i,\bullet}\|_2^2 + \|\mathbf{m}_{\bullet,j}\|_2^2) \tag{31}$$

mit Regularisierungsparameter $\lambda > 0$. Die Normalgleichungen lauten

$$(\tilde{U}^T \tilde{U} + \lambda \mathbb{I}) \mathbf{m}_{\bullet,j} = \tilde{U}^T \mathbf{r}_{\bullet,j}, \tag{32}$$

wobei $\mathbb{I} \in \mathbb{R}^{N \times N}$ der Einheitsmatrix entspricht. Die Matrix $(\tilde{U}^T \tilde{U} + \lambda \mathbb{I})$ ist invertierbar und die Lösung für $\mathbf{m}_{\bullet,j}$ ist gegeben durch [23, S. 124]

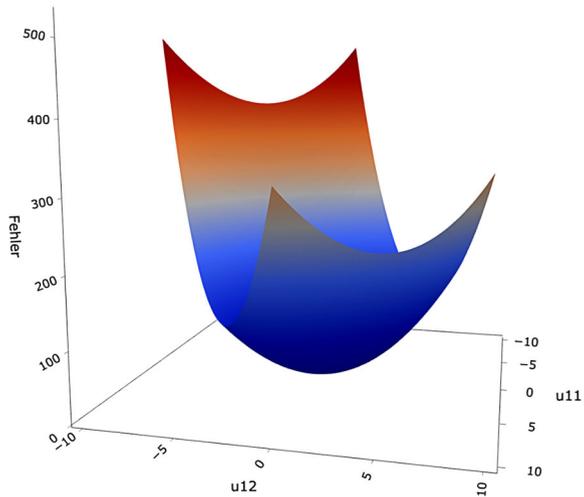
$$\mathbf{m}_{\bullet,j} = (\tilde{U}^T \tilde{U} + \lambda \mathbb{I})^{-1} \tilde{U}^T \mathbf{r}_{\bullet,j}. \tag{33}$$

Analog kann bei der Optimierung der User-Vektoren $\mathbf{u}_{i,\bullet}$ vorgegangen werden.

Der Regularisierungsparameter liefert neben der leichten Lösbarkeit des linearen Ausgleichsproblems über die Inverse von $(\tilde{U}^T \tilde{U} + \lambda \mathbb{I})$ zudem die Möglichkeit eine (Über-)Anpassung (Stichwort *overfitting*) der gesuchten Faktorisierung an die Trainingsdaten zu regulieren.

Die obigen Ausführungen zur Regularisierung sind in dieser Form mit Schüler/innen nicht denkbar. Mit den Lernenden kann der Einfluss des Regularisierungsterms jedoch sowohl auf visuelle, wie auch auf experimentelle Art erkundet werden. Zum einen kann der Einfluss des Regularisierungsterms auf die Zielfunktion f in dem Minimalbeispiel von oben visualisiert werden. Es wird ersichtlich, dass der Regularisierungsterm zu einem eindeutigen Optimum der Zielfunktion führt (vgl. Abb. 13). Alternativ können die Lernenden den Einfluss des Regularisierungsparameters auf den Trainings- bzw. Testfehler im Rahmen einer Parameterstudie untersuchen. Wählen die Lernenden einen großen Wert für λ , so liefert dies einen größeren Fehler

Abb. 13 Darstellung der Zielfunktion (24) erweitert um einen Regularisierungsterm: $f(u_{11}, u_{12}) + \lambda \cdot (u_{11}^2 + u_{12}^2)$ mit $\lambda = 2$. Es existiert ein eindeutiges globales Minimum



auf den Trainingsdaten. Der Fehler auf den Testdaten wird ggf. kleiner, da eine Überanpassung des Modells an die Trainingsdaten vermieden wird.

3.1.5 Verzerrungen in den Daten (Bias)

Das Bewertungsverhalten einzelner Nutzer/innen kann nicht nur hinsichtlich der Vorlieben für verschiedene Filme, sondern auch bezüglich deren Tendenz eher euphorisch oder reserviert zu bewerten, variieren. So ist es denkbar, dass Nutzer/in A Bewertungen von 4 und 5 abgibt, wenn ihm/ihr ein Film gefällt, wohingegen der/die kritische Nutzer/in B, dem/der die gleichen Filme gefallen, Bewertungen von 3 und 4 abgibt. Dies führt zu einer Verzerrung in den Bewertungen (auch Bias genannt), die in der Modellierung der Interaktion zwischen Nutzer/innen, Filmen und deren Eigenschaften über das Skalarprodukt $\mathbf{u}_i \cdot \mathbf{m}_{\bullet, j}$ nicht berücksichtigt werden.

Ähnliche Effekte können auch bei den Filmen auftreten. So ist es denkbar, dass ein Film aufgrund eines besonders berühmten Schauspielers im Mittel um 0,5 Bewertungspunkte besser bewertet wird, als der Durchschnitt.

Um solche Effekte zu berücksichtigen, kann die Vorhersage $\hat{\mathbf{r}}_{i, j}$ einer Bewertung unterteilt werden: in einen Term, der Effekte wie User- und Movie-Bias berücksichtigt, und einen Term, der die tatsächliche Interaktion von Nutzer/innen und Filmen hinsichtlich der Eigenschaften modelliert. Dies liefert als Vorhersage $\hat{\mathbf{r}}_{i, j}$ einer Bewertung $\mathbf{r}_{i, j}$ [6]

$$\hat{\mathbf{r}}_{i, j} = \mathbf{u}_{i, \bullet} \cdot \mathbf{m}_{\bullet, j} + \mu + b_i + b_j. \quad (34)$$

Hierbei entspricht μ dem Mittelwert über alle bekannten Bewertungen im Trainingsdatensatz, b_i beschreibt den Bias von User i und b_j den Bias eines beliebigen Filmes j . Für die Festlegung der Parameter b_i und b_j sind unterschiedliche Ansätze denkbar. Eine Möglichkeit ist, den User-Bias b_i als Abweichung des mittleren Ratings

des Nutzers i von dem Mittelwert μ und analog b_j als Abweichung des mittleren Ratings des Filmes j vom Mittelwert μ zu definieren.

Insgesamt ist damit das folgende Optimierungsproblem zu lösen:

$$\min_{\mathbf{u}_{i,\bullet}, \mathbf{m}_{\bullet,j}} \sum_{(i,j) \in \kappa} (r_{i,j} - \mathbf{u}_{i,\bullet} \cdot \mathbf{m}_{\bullet,j} - \mu - b_i - b_j)^2 + \lambda \cdot (\|\mathbf{u}_{i,\bullet}\|_2^2 + \|\mathbf{m}_{\bullet,j}\|_2^2). \quad (35)$$

Anhand von geeignet Beispielen und unterstützt durch Visualisierungen könnten Verzerrungen in den Daten auch mit Schüler/innen diskutiert werden. In den bisher durchgeführten Schülerworkshops wurde dies jedoch noch nicht umgesetzt.

Lösen des Optimierungsproblems (35) basierend auf den Bewertungsdaten $r_{i,j}$ des Trainingsdatensatzes liefert eine Zerlegung der Bewertungsmatrix R . Diese kann genutzt werden, um Bewertungen des Testdatensatzes vorherzusagen und die Anwendbarkeit des Modells auf unbekannte Daten zu validieren. Die Entwicklung eines Empfehlungssystems über eine Matrix-Faktorisierung ist damit, wie die oben beschriebene Stützvektormethode, dem *überwachten Lernen* zuzuordnen.

Anmerkung: Neben der Entwicklung von Empfehlungssystemen über eine Matrix-Faktorisierung finden vielfach auch sog. *Nachbarschaftsmethoden* Einsatz. Bei diesen werden Ähnlichkeiten zwischen Nutzer/innen (Nutzer-basierter Ansatz) oder zwischen Filmen (Film-basierter Ansatz) über Ähnlichkeitsmaße modelliert. Basierend auf gefundenen Ähnlichkeiten zwischen den Nutzer/innen bzw. Filmen werden die Filme festgelegt, die einem/einer Nutzer/in sehr wahrscheinlich gefallen. Auch zu den Nachbarschaftsmethoden wurde Lernmaterial für Oberstufenschüler/innen entwickelt. Dieses ermöglicht es den Lernenden selbstständig Ähnlichkeitsmaße zu definieren und auf dem Netflix-Datensatz zu validieren [16].

Weiterführende Punkte, die mit Lernenden diskutiert werden könnten oder gar sollten, sind:

- Das sogenannte *Cold-Start-Problem*: Wie gehen wir mit Nutzer/innen um, die sich neu registriert haben und noch keine einzige Bewertung abgegeben haben (Nullzeile in der Bewertungsmatrix)?
- Der Vorteil der Matrix-Faktorisierung für das Speichern der Bewertungsdaten: Statt ursprünglich 100.480.507 Bewertung im Netflix-Datensatz müssen bei einer Zerlegung, bei der 40 Eigenschaften berücksichtigt werden, lediglich $17.770 \cdot 40 + 40 \cdot 480.189 = 19.918.360$ Daten gespeichert werden. Dies entspricht ca. 20 % des ursprünglichen Speicherbedarfs.
- Die persönliche Einschätzung der Lernenden, ob sie die Empfehlungssysteme für sich persönlich als sinnvoll und hilfreich empfinden. Beispielsweise kann kritisiert werden, dass man als Nutzer/in keine Vorschläge für andersartige Filme erhält (Stichwort: *Filterblase*).
- Die Manipulation von Empfehlungssystemen (bspw. durch Fake-User).
- Die kritische Diskussion, inwieweit eine De-Anonymisierung des Datensatzes für Nutzer/innen problematisch sein kann.
- Die Diskussion zur Wichtigkeit der Vorhersage abhängig vom Kontext in dem ein Modell angewendet werden soll: So lässt sich differenzieren, dass falschen Vorhersagen im Bereich Unterhaltung (z. B. Musik-, Film-, Buchempfehlungen) eine

ganz andere Bedeutung zukommt als falschen Krebs-Diagnosen. Diese Diskussion ließe sich auch an die Problemstellung zur (Bild-)Klassifizierung angliedern.

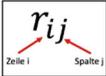
4 Umsetzung des Lernmaterials und Erfahrungen mit Schüler/innen

Maschinelles Lernen wird in der Regel eingesetzt, um Problemstellungen basierend auf großen Datenmengen, die ohne Computer nicht zu verarbeiten wären, zu lösen. Auch mit den Lernenden kommt im Sinne der Entwicklung von *authentischem* Lernmaterial der Computer zum Einsatz, um (a) Daten zu visualisieren, (b) die Lernenden von aufwendigen Rechenschritten zu entlasten und damit den Fokus auf den Problemlöseprozess zu legen, und (c) das Lösen von (Optimierungs-)Problemen, die schulmathematisches Wissen überschreiten, dem Computer zu überlassen.

(1) Teil c

Ziel ist es, die Vorhersage der Ratings basierend auf einer gefundenen Zerlegung in U und M zu verallgemeinern und mathematisch zu formulieren, um dies später auch auf große Datensätze anwenden zu können. Wir stellen dazu in dieser Aufgabe eine Formel zur Berechnung eines beliebigen Eintrags der Ratingmatrix R aus der Usermatrix U und der Moviematrix M auf. Zunächst bezeichnen wir die Einträge von R , U und M allgemein wie folgt:

$$R = \begin{matrix} & \begin{matrix} \text{f1} & \text{f2} & \text{f3} & \text{f4} & \text{f5} & \text{f6} \end{matrix} \\ \begin{matrix} \text{u1} \\ \text{u2} \\ \text{u3} \\ \text{u4} \\ \text{u5} \end{matrix} & \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} & r_{15} & r_{16} \\ r_{21} & r_{22} & r_{23} & r_{24} & r_{25} & r_{26} \\ r_{31} & r_{32} & r_{33} & r_{34} & r_{35} & r_{36} \\ r_{41} & r_{42} & r_{43} & r_{44} & r_{45} & r_{46} \\ r_{51} & r_{52} & r_{53} & r_{54} & r_{55} & r_{56} \end{pmatrix} \end{matrix}$$

$$U = \begin{matrix} & \begin{matrix} A & C \end{matrix} \\ \begin{matrix} \text{u1} \\ \text{u2} \\ \text{u3} \\ \text{u4} \\ \text{u5} \end{matrix} & \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ u_{41} & u_{42} \\ u_{51} & u_{52} \end{pmatrix} \end{matrix} \quad M = \begin{matrix} & \begin{matrix} \text{m1} & \text{m2} & \text{m3} & \text{m4} & \text{m5} & \text{m6} \end{matrix} \\ \begin{matrix} A \\ C \end{matrix} & \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \end{pmatrix} \end{matrix}$$


Der Eintrag r_{34} (Variable: r_{34}) entspricht dem Eintrag r_{ij} mit Zeile $i = 3$ und Spalte $j = 4$. Genauso funktioniert es für die Usermatrix und die Moviematrix. Zum Beispiel liefert u_{12} (Variable: u_{12}) den Eintrag der Usermatrix in Zeile i und in Spalte 2 .

 & 

- Stelle eine Formel auf, mit der der Eintrag r_{34} in **Zeile 3 und Spalte 4** der Ratingmatrix R berechnet werden kann. Gib deine Formel für das erste `NaN` ein.
- Stelle eine Formel auf, mit der ein beliebiger Eintrag r_{ij} in **Zeile i und Spalte j** der Ratingmatrix R berechnet werden kann. Gib deine Formel für das zweite `NaN` ein.
Notiere deine Formeln auch auf dem Antwortblatt.

Falls du Hilfe benötigst, kannst du dir hier einen Tipp ansehen.

Hinweis: Die Usermatrix U und die Moviematrix M wurden bereits gespeichert. Du kannst die Einträge von U und M in deiner Formel verwenden.

(2)

```
# Formel für den Eintrag r34
compute_r34 = u31 * m14 + u32 * m24;

# Formel für den Eintrag rij
compute_rij = NaN;

# Hier nichts ändern!
checkCompute_rij(compute_r34, compute_rij)
```

not executed at 2021-01-09 16:24:56 in 16s

(3)

✓ Deine Formel für r34 ist korrekt.

! Deine Formel für rij ist noch nicht korrekt.

Abb. 14 Auszug aus einem digitalen Arbeitsblatt mit (1) Textfeldern für Erläuterungen und Aufgabenstellungen, (2) Code-Feldern für die Eingabe der Lösungen und (3) die Rückmeldung zu den Code-Eingaben

Ein Werkzeug, mit dem interaktives, digitales Lernmaterial entwickelt werden kann, sind Jupyter Notebooks¹⁰. Diese erlauben die Entwicklung von digitalen Arbeitsblättern bei denen *Textfelder* mit *Code-Feldern* kombiniert werden können (vgl. Abb. 14) und sind mittlerweile ein Industrie-Standard für KI-Anwendungen.

Die Textfelder können für Erläuterungen zur Problemstellung sowie für Aufgabenstellungen genutzt werden. Grafiken, aber auch Erklärvideos, können direkt in die Textfelder eingebunden werden. Zudem kann Differenzierungsmaterial (bspw. Zusatzaufgaben und Hilfekarten) leicht integriert und an den entsprechenden Aufgabenteilen verlinkt werden. So soll das Material auch in heterogenen Lerngruppen einsetzbar sein. Die Code-Felder eignen sich für Berechnungen und Visualisierungen von Daten, Funktionsgraphen etc. Durch versteckten Hintergrund-Code können die Formeln, Modelle und Ergebnisse der Schüler/innen zudem automatisch überprüft und individuelle Rückmeldung zu einzelnen Aufgabenteilen ausgegeben werden (vgl. Abb. 14). Dies bietet die Möglichkeit auf typische Schülerfehler entsprechende Rückmeldungen zu geben. Selbstbestimmtes Lernen im eigenen Lerntempo soll unterstützt werden.

Jupyter Notebooks werden vielfältig in Industrie und Forschung eingesetzt, womit nicht nur die Lerninhalte (Maschinelle Lernmethoden), sondern zugleich das digitale Werkzeug einen authentischen Einblick in aktuelle Problemlösestrategien und technische Umsetzungen in der angewandten Mathematik bzw. allgemeiner in MINT-Bereichen liefert.

Die beiden in diesem Beitrag vorgestellten Problemstellungen wurden als Jupyter Notebooks umgesetzt. Die Code-Felder sind als Lückentexte aufgebaut, in denen die Lernenden ihre Lösungen oder Formeln an ausgewiesenen Stellen eintragen können (vgl. Abb. 14). Die Code-Eingaben, die von den Lernenden getätigt werden müssen, gehen dabei nicht über die Bedienung eines grafikfähigen Taschenrechners hinaus. Programmierkenntnisse sind damit keine Voraussetzung für die Bearbeitung des Lernmaterials.

Die Arbeitsblätter wurden basierend auf der Programmiersprache *Julia*¹¹ entwickelt, deren Syntax nah an Pseudocode ist. Alternativ kann als Programmiersprache u. a. auch *Python* oder *R* verwendet werden.

Um die Bearbeitung des Materials problemlos in Projektzeiten, AGs oder im Rahmen einer Unterrichtsreihe vor Ort in der Schule durchführen zu können, liegt das Material auf einer cloud-basierten Workshop-Plattform zum direkten Unterrichtseinsatz bereit. Für den Einsatz des Materials ist keine Installation von Software, sondern lediglich ein Webbrowser notwendig. Da sämtliche Berechnungen auf Servern des KIT ausgeführt werden, kann das Material auf jedem beliebigen Endgerät mit Internetzugang bearbeitet werden.

Interessierte Leser/innen sind eingeladen auf das digitale Lernmaterial unter <https://workshops.cammp.online> zuzugreifen¹². Das Lernmaterial liegt dort unter einer Creative Commons Lizenz (CC-BY-SA) bereit.

¹⁰ <https://jupyter.org/>, letzter Aufruf am 01.12.2020.

¹¹ <https://julialang.org>, letzter Aufruf am 01.12.2020.

¹² Zugangsdaten werden per Anfrage an cammp@scc.kit.edu jederzeit zur Verfügung gestellt.

Die Lernmaterialien wurden bereits im Rahmen von Unterrichtsreihen (à 4–5 Doppelstunden) oder eintägigen Workshops mit insgesamt mehr als 200 Schüler/innen ab Klasse 10 eingesetzt. Bei den bisherigen Durchführungen, die zum Teil in Präsenz und zum Teil online stattgefunden haben, stachen die vielfältigen Argumente der Schüler/innen bei der kritischen Reflexion der Modelle wie auch zu verschiedenen Anwendungsmöglichkeiten und Gefahren heraus. In den sehr regen Diskussionen zu beiden Themen (Bildklassifizierung und Empfehlungssysteme) zeigte sich, dass die Lernenden ein großes Interesse an den Problemstellungen hatten.

Danksagung Ein herzlicher Dank gilt Lars Schmidt, Jannick Wolters und Marcel Marnitz, die bei der Umsetzung des Lernmaterials mitgewirkt haben. Auch danken wir den beiden Gutachter/innen dieses Beitrags für die konstruktiven und sehr wertvollen Verbesserungsvorschläge.

Funding Open Access funding enabled and organized by Projekt DEAL. Diese Arbeit wurde gefördert durch das Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg im Rahmen des Projekts „Simulierte Welten“.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

1. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2006)
2. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
3. Feuerberger, A., He, Y., Khatri, S.: Statistical significance of the Netflix challenge. *Stat. Sci.* **27**(2), 202–231 (2012)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. 2008 Eighth IEEE International Conference on Data Mining., S. 263–272 (2008) <https://doi.org/10.1109/ICDM.2008.22>
5. Janssen, D.: Machine Learning in der Schule – Ein praxisorientierte Einführung in künstliche neuronale Netze, Gesichtserkennung und Co. 1. Aufl. Science On Stage Deutschland, Berlin (2020)
6. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* **4**(1), 1–24 (2010). <https://doi.org/10.1145/1644873.1644874>
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
8. Ministerium für Bildung, Wissenschaft, Weiterbildung und Kultur des Landes Rheinland-Pfalz: Lehrplan Mathematik – Grund- und Leistungsfach in der gymnasialen Oberstufe (2015). <https://lehrplaene.bildung-rp.de/?keyword=mathematik>, Zugegriffen: 6. Mai 2021
9. Ministerium für Kultus, Jugend und Sport Baden-Württemberg: Bildungsplan Mathematik (2016). www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_M.pdf, Zugegriffen: 6. Mai 2021
10. Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen: Kernlehrplan für die Sekundarstufe II Gymnasium / Gesamtschule in Nordrhein-Westfalen – Mathematik (2014).

- www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/m/KLP_GOST_Mathematik.pdf, Zugegriffen: 6. Mai 2021
11. Narayanan, A., Shmatikov, V.: How to break anonymity of the Netflix prize dataset (2006). <https://arxiv.org/abs/cs/0610105>, Zugegriffen: 30. Nov. 2020
 12. Nilsson, N.J.: Introduction to machine learning. An early draft of a proposed textbook (1998). <https://ai.stanford.edu/people/nilsson/MLBOOK.pdf>, Zugegriffen: 28. Dez. 2020. Stanford University
 13. Nocedal, J., Wright, J.S.: Numerical Optimization. Springer Series in Operations Research. Springer, New York (2006)
 14. O’Neil, C.: Weapons of Math Destruction – How Big Data Increases Inequality and Threatens Democracy. Crown, New York (2016)
 15. Opel, S., Schlichtig, M., Schulte, C., Biehler, R., Frischemeier, D., Podworny, S., Wassong, T.: Entwicklung und Reflexion einer Unterrichtssequenz zum Maschinellen Lernen als Aspekt von Data Science in der Sekundarstufe II. In: Pasternak, A. (Hrsg.) Proceedings zur 18. GI-Fachtagung Informatik und Schule “Informatik für Alle”, S. 285–294. Gesellschaft für Informatik, Bonn (2019)
 16. Rantza, L.: Empfehlungssysteme basierend auf Nachbarschaftsmethoden – mathematisch-fachliche Diskussion und Entwicklung digitalen Lernmaterials zur Netflix Challenge für Schüler*innen der Sekundarstufe II (2021). www.cammp.online/Netflix-BA.pdf, Zugegriffen: 10. Mai 2021. Bachelorarbeit, KIT
 17. Schlichtig, M., Opel, S., Schulte, C., Biehler, R., Frischemeier, D., Podworny, S., Wassong, T.: Maschinelles Lernen im Unterricht mit Jupyter Notebook. In: Pasternak, A. (Hrsg.) Proceedings zur 18. GI-Fachtagung Informatik und Schule “Informatik für Alle”, S. 385. Gesellschaft für Informatik, Bonn (2019)
 18. Schmidt, L.: Machine Learning: automatische Bilderkennung mit Mathematik?! - Ein Lehr-Lern-Modul im Rahmen eines mathematischen Modellierungstages für Schülerinnen und Schüler der Sekundarstufe II (2019). www.cammp.online/Masterthesis4druck.pdf, Zugegriffen: 30. Nov. 2020. Masterarbeit, RWTH Aachen
 19. Schölkopf, B., Müller, K.R., Smola, A.: Lernen mit Kernen. Inform Forsch Entw **14**, 154–163 (1999). <https://doi.org/10.1007/s004500050135>
 20. Schönbrodt, S.: Maschinelle Lernmethoden für Klassifizierungsprobleme – Perspektiven für die mathematische Modellierung mit Schülerinnen und Schülern. Springer Spektrum, Wiesbaden (2019)
 21. Staatsinstitut für Schulqualität und Bildungsforschung München: Lehrplan Mathematik Jahrgangsstufen 11/12 (2004). https://www.gym8-lehrplan.bayern.de/contentserv/3.1.neu/g8.de/id_26192.html, Zugegriffen: 6. Mai 2021
 22. Steffen, N.: Sicherheit der Privatsphäre in sozialen Netzwerken – Wie Mathematik die Nutzer ausspioniert. Ein Lehr-Lern-Modul im Rahmen eines mathematischen Modellierungstages für Schülerinnen und Schüler der Sekundarstufe I (2019). www.cammp.online/MasterarbeitNils.pdf, Zugegriffen: 28. Dez. 2020. Masterarbeit, RWTH Aachen
 23. Strang, G.: Linear Algebra and Learning from Data. Wellesley-Cambridge Press, Wellesley (2019)
 24. Sube, M.: Wie sicher ist meine Privatsphäre in sozialen Netzwerken? ...und was hat das mit Mathe zu tun? (2016). www.cammp.online/Masterarbeit%20Maike%20Sube.pdf, Zugegriffen: 28. Dez. 2020. Masterarbeit, RWTH Aachen
 25. Winter, H.: Mathematikunterricht und Allgemeinbildung (1995). <https://ojs.didaktik-der-mathematik.de/index.php/mgdm/article/view/69/80>, Zugegriffen: 16. Jan. 2021. Mitteilungen der Gesellschaft für Didaktik der Mathematik 61:37–46
 26. Wohak, K., Sube, M., Schönbrodt, S., Frank, M., Roeckerath, C.: Authentische und relevante Modellierung mit Schülerinnen und Schülern an nur einem Tag?! (2021). In: Bracke, M., Ludwig, M., Vorhölter, K. (Hrsg.) Modellierungsprojekte mit Schülerinnen und Schülern. Realitätsbezüge im Mathematikunterricht, S. 37–50. Springer Spektrum, Wiesbaden