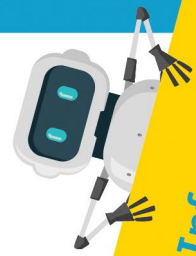


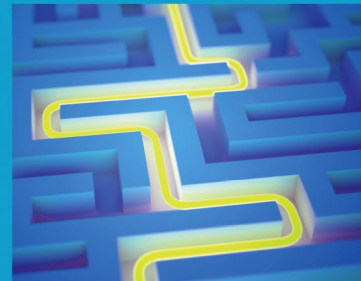
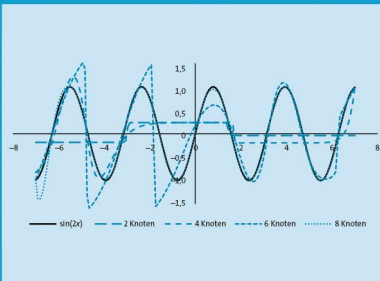
Paul Wilmott



Informatik
verstehen

GRUNDKURS

Machine Learning



- + Mathematische Grundlagen des maschinellen Lernens
- + Alle wichtigen Algorithmen Schritt für Schritt erklärt
- + Inkl. Reinforcement Learning, k-nächste Nachbarn, Neuronale Netze u. v. m.



Rheinwerk
Computing

Liebe Leserin, lieber Leser,

mit Machine Learning haben Sie sich ein spannendes Thema vorgenommen.

Maschinelles Lernen ist angewandte Mathematik. Dieser Grundkurs stellt Ihnen neun wichtige Lernverfahren genau aus dieser Perspektive vor.

Es lädt sie ein, die mathematischen Verfahren nachzuvollziehen – anhand von Beispielen, die für alle Fachrichtungen nachvollziehbar sind. Legen Sie sich neben dem Computer auch Papier und Bleistift zurecht und scheuen Sie die Mühe nicht, einmal Schritt für Schritt durchzuspielen, was Ihnen auch eine Software abnehmen kann.

Sie können dieses Buch von vorne nach hinten lesen und so nach und nach alle Verfahren kennenlernen. Vielleicht interessiert Sie aber auch nur eine Auswahl. Oder Sie sind schon dabei, eine bestimmte Methode in einem Softwareprojekt anzuwenden, und möchten besser verstehen, was dahintersteckt.

Dann spricht nichts dagegen, direkt zu dem Kapitel zu springen, das Sie interessiert. Alle Kapitel sind unabhängig voneinander lesbar, die Beispiele stehen für sich selbst.

Sollten Sie übrigens Fragen zum Buch haben, Lob oder Kritik äußern wollen, wenden Sie sich gern an mich. Ihre freundlichen Kommentare sind uns jederzeit willkommen.

Ihre Almut Poll

Lektorat Rheinwerk Computing

almut.poll@rheinwerk-verlag.de

www.rheinwerk-verlag.de

Rheinwerk Verlag · Rheinwerkallee 4 · 53227 Bonn

Auf einen Blick

1	Einführung	17
2	Allgemeines	35
3	K-nächste Nachbarn	75
4	K-Means Clustering.....	87
5	Naiver Bayes-Klassifikator	107
6	Regressionsmethoden.....	115
7	Support-Vektor-Maschinen	123
8	Selbstorganisierende Karten.....	137
9	Entscheidungsbäume.....	151
10	Neuronale Netze	173
11	Verstärkendes Lernen.....	199

Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

Lektorat Almut Poll

Übersetzung Joachim Steinwendner

Korrektorat Annette Lennartz, Bonn

Herstellung E-Book Norbert Englert

Covergestaltung Julia Schuster

Coverbilder iStock: 1097395654© nd3000; shutterstock: 1188563173

©Konstantin Faraktinov

Satz E-Book Daniel van Soest

Übersetzung der Originalausgabe:

Paul Wilmott: Machine Learning. An Applied Mathematics Introduction

Copyright © 2019, Paul Wilmott

All rights reserved

Original English-language edition first published 2019 by Panda Ohana Publishing.

German-language edition published 2020 by Rheinwerk Verlag GmbH.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8362-7600-9

1. Auflage 2020

© Rheinwerk Verlag GmbH, Bonn 2020

www.rheinwerk-verlag.de

Für JRO, von Ihrem furchtlosen Studenten

Inhalt

Vorwort	13
1 Einführung	17
1.1 Maschinelles Lernen	18
1.2 Lernen ist der Schlüssel	19
1.3 Ein wenig Geschichte	20
1.4 Schlüsselmethodiken in diesem Buch	22
1.5 Klassische mathematische Modellierung	26
1.6 Maschinelles Lernen ist anders	28
1.7 Einfachheit führt zu Komplexität	29
1.8 Weiterführende Literatur	33
2 Allgemeines	35
2.1 Jargon und Notation	35
2.2 Skalierung	37
2.3 Distanzmessung	38
2.4 Fluch der Dimensionalität	39
2.5 Hauptkomponentenanalyse	39
2.6 Maximum-Likelihood-Schätzung	40
2.7 Konfusionsmatrix	44
2.8 Kostenfunktion	47
2.9 Gradientenabstieg	52
2.10 Training, Testen und Validieren	54
2.11 Bias und Varianz	57
2.12 Lagrange-Multiplikatoren	63

2.13	Mehrfachklassen	65
2.14	Informationstheorie und Entropie	67
2.15	Verarbeitung natürlicher Sprache (NLP)	70
2.16	Bayes-Theorem	72
2.17	Was nun?	73
2.18	Weiterführende Literatur	74

3 K-nächste Nachbarn 75

3.1	Wofür können wir die Methode verwenden?	75
3.2	Wie die Methode funktioniert	76
3.3	Der Algorithmus	78
3.4	Probleme mit KNN	78
3.5	Beispiel: Körpergröße und -gewicht	79
3.6	Regression	83
3.7	Weiterführende Literatur	85

4 K-Means Clustering 87

4.1	Wofür können wir die Methode verwenden?	87
4.2	Was macht K-Means Clustering?	89
4.3	Scree-Plots	93
4.4	Beispiel: Kriminalität in England, 13 Dimensionen	94
4.5	Beispiel: Volatilität	98
4.6	Beispiel: Zinssatz und Inflation	100
4.7	Beispiel: Zinssätze, Inflation und BIP-Wachstum	103
4.8	Ein paar Kommentare	104
4.9	Weiterführende Literatur	105

5	Naiver Bayes-Klassifikator	107
5.1	Wofür können wir ihn verwenden?	107
5.2	Verwendung des Bayes-Theorems	108
5.3	Anwendung des NBK	108
5.4	In Symbolen	110
5.5	Beispiel: Politische Reden	111
5.6	Weiterführende Literatur	114
6	Regressionsmethoden	115
6.1	Wofür können wir sie verwenden?	115
6.2	Mehrdimensionale lineare Regression	116
6.3	Logistische Regression	117
6.4	Beispiel: Noch einmal politische Reden	119
6.5	Weitere Regressionsmethoden	121
6.6	Weiterführende Literatur	122
7	Support-Vektor-Maschinen	123
7.1	Wofür können wir sie verwenden?	123
7.2	Harte Ränder	123
7.3	Beispiel: Iris (Schwertlilie)	126
7.4	Lagrange-Multiplier-Version	128
7.5	Weiche Ränder	130
7.6	Kernel-Trick	132
7.7	Weiterführende Literatur	136

8 Selbstorganisierende Karten 137

8.1	Wofür können wir sie verwenden?	137
8.2	Die Methode	138
8.3	Der Lernalgorithmus	140
8.4	Beispiel: Gruppierung von Aktien	142
8.5	Beispiel: Abstimmungen im Unterhaus	147
8.6	Weiterführende Literatur	149

9 Entscheidungsbäume 151

9.1	Wofür können wir sie verwenden?	151
9.2	Beispiel: Zeitschriftenabo	153
9.3	Entropie	158
9.4	Überanpassung und Abbruchregeln	161
9.5	Zuschneiden	162
9.6	Numerische Merkmale/Attribute	162
9.7	Regression	164
9.8	Ausblick	171
9.9	Bagging und Random Forest	171
9.10	Weiterführende Literatur	172

10 Neuronale Netze 173

10.1	Wofür können wir sie verwenden?	173
10.2	Ein sehr einfaches Netzwerk	173
10.3	Universelles Approximations-Theorem	174
10.4	Ein noch einfacheres Netzwerk	176
10.5	Die mathematische Manipulation im Detail	177
10.6	Häufige Aktivierungsfunktionen	181

10.7	Das Ziel	182
10.8	Beispiel: Approximation einer Funktion	183
10.9	Kostenfunktion	184
10.10	Backpropagation	185
10.11	Beispiel: Buchstabenerkennung	188
10.12	Training und Testen	190
10.13	Mehr Architekturen	194
10.14	Deep Learning	196
10.15	Weiterführende Literatur	197

11 Verstärkendes Lernen 199

11.1	Wofür können wir es verwenden?	199
11.2	Geländeausfahrt mit Ihrem Lamborghini 400 GT	200
11.3	Jargon	202
11.4	Ein erster Blick auf Blackjack	203
11.5	Der klassische Markow-Entscheidungsprozess für Tic-Tac-Toe	204
11.6	Noch mehr Jargon	206
11.7	Beispiel: Der mehrarmige Bandit	207
11.8	Etwas anspruchsvoller 1: Bekannte Umgebung	211
11.9	Beispiel: Ein Labyrinth	214
11.10	Notation zu Wertefunktionen	218
11.11	Die Bellman-Gleichung	220
11.12	Optimale Policy	221
11.13	Die Bedeutung der Wahrscheinlichkeit	222
11.14	Etwas anspruchsvoller 2	223
11.15	Monte Carlo Policy Evaluation	224
11.16	Temporal-Difference-Lernen	227
11.17	Vor- und Nachteile: MC versus TD	228
11.18	Finden der optimalen Policy	229

11.19 Sarsa	230
11.20 Q-Lernen	232
11.21 Beispiel: Blackjack	233
11.22 Große Zustandsräume	245
11.23 Weiterführende Literatur	245
 Datensätze	 247
Epilog	251
 Index	 253

Vorwort

Dieses Buch soll Ihnen die wichtigsten mathematischen Grundlagen möglichst vieler maschineller Lernverfahren vermitteln. Ich lasse die meiner Einschätzung nach unnötig langweiligen Teile dabei weg. Es unterscheidet sich außerdem absichtlich von den meisten anderen Büchern zu diesem Thema.

Sollten Sie dieses Buch bereits gekauft haben, dann können Sie die nächsten Absätze überspringen. Denn darin erkläre ich, für wen das Buch gedacht ist. Wenn Sie das Buch gekauft haben und es stellt sich heraus, dass Sie das falsche Publikum sind, dann... ups.

Es gibt verschiedene Lesertypen von Lehrbüchern zu maschinellem Lernen. Ihre Zugehörigkeit ist je nach Typ sehr unterschiedlich verteilt.

Gemessen an den via Amazon verfügbaren Büchern, schätze ich, dass es eine Unzahl an Leuten gibt, die nur auf relevanten Programmiercode aus sind, zumeist in Python oder R. Und es gibt in der Tat eine Menge Bücher, die genau das bieten. Noch dazu mit einer beeindruckend kompakt programmierten Repräsentation der Problemlösungen. Diese Gruppe werde ich links liegen lassen, die sind bereits gut versorgt.

Dann gibt es eine ganze Reihe von Büchern, die die *Anwendungen* von Techniken des maschinellen Lernens hervorheben – ausgestattet mit wenig Code und sehr elementarer Mathematik. Diese eignen sich hervorragend, um sich einen Überblick über das Thema zu verschaffen, aber es mangelt ihnen an Details. Man kann sie leicht an der etwas fragwürdigen Schriftart erkennen, die sie für ihre Mathematik verwenden. Und daran, dass sie im Eigenverlag publiziert werden, obwohl daran absolut nichts auszusetzen ist.

Und dann gibt es da noch die Expertenbücher, von denen sich jedes einem speziellen Teilgebiet widmet und sehr in die Tiefe geht. Zwar sind sie nicht erkennbar an ihrer Schriftart, aber sie sind doch etwas unheimlich. Und für gewöhnlich sehr, sehr teuer.

Nein, ich versuche nicht, mit einem der obengenannten Buchtypen zu konkurrieren.

Ich ziele auf ein kleines Publikum ab, das diesbezüglich etwas unterversorgt ist – ein Publikum, das womöglich minütlich schwindet, während ich diese Zeilen tippe: Angewandte Mathematiker. Menschen, die es lieben, Mathematik zu *machen* und knifflige Probleme zu lösen. Vielleicht alte Hasen, geübt in klassischen Methoden, die daran interessiert sind, mehr über die datengetriebene Revolution zu erfahren, die sie vom Rande aus beobachten. Sie wissen, wer sie sind. Ja, Sie da hinten. Nein, nicht abhauen. Setzen Sie sich hier vorne hin in die erste Reihe, und lassen Sie uns gemeinsam dieses Thema erarbeiten...

Lassen wir Die Maschinen sprechen

Bevor wir so richtig beginnen, lassen Sie uns lesen, was »Die Maschine« über dieses Buch zu sagen hat.¹ (Sehr aufschlussreich. Besonders gefallen hat mir die Empfehlung im letzten Satz.)

Diese Bände sollen in mehreren Abschnitten erscheinen. Sie können sie in diesem Buch an jedem der vier verschiedenen Orte von Anfang bis Ende lesen. Der Hauptpunkt, den Sie hier machen wollen, ist, dass sowohl das Buch als auch der Text über so viele der Techniken, die Sie verwenden werden, gut verstanden werden. Aber was wird getan, um sie alle zugänglich zu machen, und dass sie nicht zu schwer zu verstehen sein werden? Tatsächlich wird jede der Techniken ziemlich leicht zu verstehen sein, und das Material wird sehr gut organisiert sein, so dass Sie in kürzester Zeit (d. h. nach weniger als 40 Seiten) das tun können, was Sie wollen. Einige grundlegende Beispiele für verschiedene Arten des maschinellen Lernens finden sich in dem Buch, aber ich werde mich auf die Hauptpunkte des Buches konzentrieren. Wenn Sie daran interessiert sind, mehr über dieses Thema zu lesen, dann sehen Sie sich mein Buch an.

Danksagung

Ich möchte meinem Co-Autor des Buches *The Money Formula* (erhältlich in allen guten Buchhandlungen), David Orrell, danken für seine speziell auf Tippfehlererkennung gebauten Augen. Das Cover habe ich Liam Larkin zu verdanken, dem Designer meines Magazins (»The world's most expensive magazine«, laut *Esquire*).

Ich bin besonders Thijs van den Berg zu Dank verpflichtet, der mich auf dieses Thema gebracht hat. Er schlug außerdem viele Ergänzungen und Korrekturen zum Buch vor, die ich zu 82,1% auch umgesetzt habe. Thijs und ich geben Schulungen zum Thema maschinelles Lernen, und so möchte ich auch den Kursteilnehmern danken. Ihre Kommentare und Fragen waren eine große Hilfe, um gute Erklärungen für die oft komplexen Zusammenhänge zu finden.

Schließlich muss ich noch meine Frau, meine Kinder, das Finanzamt Ihrer Majestät, IRS, »Game of Thrones«, Sky News und den Brexit erwähnen, ohne die dieses Buch in der Hälfte der Zeit fertig geworden wäre.

¹ Text erzeugt mit Maschine Nr. 1 https://colab.research.google.com/github/ilopezfr/gpt-2/blob/master/gpt-2-playground_.ipynb und von Maschine Nr. 2 (DeePL) vom Englischen ins Deutsche übersetzt.

Über den Autor

Beruflich gesehen ...

Paul Wilmott studierte Mathematik am St. Catherine's College, Oxford, wo er auch sein Doktorat (Dr. phil.) abschloss. Er ist der Autor von *Paul Wilmott Introduces Quantitative Finance* (Wiley 2007), *Paul Wilmott On Quantitative Finance* (Wiley 2006), *Frequently Asked Questions in Quantitative Finance* (Wiley 2009), *The Money Formula: Dodgy Finance, Pseudo Science, and How Mathematicians Took Over the Markets* (mit David Orrell) (Wiley 2017) und anderen Finanzlehrbüchern. Er hat über 100 Forschungsartikel zu Finanzwissenschaften und Mathematik geschrieben. Paul Wilmott war Gründungspartner des Volatilitäts-Arbitrage-Hedgefonds Caissa Capital, der 170 Millionen US\$ verwaltete. Sein Verantwortungsbereich umfasste die Prognose, das Pricing von Derivaten und das Risikomanagement.

Paul ist der Inhaber von www.wilmott.com, der beliebten Website der Community zur quantitativen Finanz und des Magazins *Wilmott*. Er ist der Schöpfer des Certificate in Quantitative Finance, cqf.com, und Präsident des CQF Instituts, cqfinstitute.org.

Andererseits ...

Paul war ein professioneller Jongleur bei der Dab-Hands-Truppe und hat als Undercover-Ermittler für Channel 4 gearbeitet. Er erhielt auch drei »half blues« (Auszeichnung für sportliche Spitzenleistung) von der Oxford University für Gesellschaftstanz. Im Alter von 41 Jahren gewann er schließlich einen Wettbewerb im Sandburgenbau. Er erzeugt seinen eigenen Käse, dessen Geschmack als »gewöhnungsbedürftig« beschrieben wurde.

Paul war der erste Mann in Großbritannien, der sich online scheiden ließ. Er war als Experte an einer TV-Show unter anderem mit der Vorhersage des Namens des königlichen Nachwuchses und des Gewinners des Eurovision Song Contests beauftragt. Er lag vollkommen falsch.

Er spielte Bridge für das D-Team seiner Schule. (Es gab kein E-Team.)

Und er spielt Ukulele. Klar.

Kapitel 1

Einführung

Hallo, mein Name ist Paul Wilmott.¹ Ich gehöre der Profession der angewandten Mathematiker an und habe viele Jahre in der Wissenschaft gearbeitet, dies und das erforscht, Studierende betreut und Arbeiten für wissenschaftliche Journale geschrieben. Der Beginn meiner Forschung lag im Bereich der angewandten/industriellen Mathematik. Damit ging einher, dass ich mit Leuten aus der Industrie zu tun hatte und ihre Aufgaben, die oft mit physikalischen Prozessen zu tun hatten, in mathematische Modelle zu transferieren versuchte. Die Lösung dieser mathematischen Modelle ergab interessante Einblicke in die gestellten Probleme. Diese Vorgehensweise hatte viele Vorteile gegenüber physikalischen Experimenten, nicht zuletzt den, dass wir viel billiger waren. Über die Jahre entwickelt man dabei eine Reihe von Techniken, die helfen, fast jede Situation modellieren zu können. Manchmal hat man physikalische Gesetze, wie etwa das Gesetz der Massenerhaltung, auf dem man aufbauen kann, viel öfter aber eben nicht. Hat man solide physikalische Gesetze zur Verfügung, dann führt das zu einem *quantitativ* guten Modell. Ansonsten bleibt nur ein Modell übrig, das lediglich *qualitativ* gut ist. Letzteres bezeichnet man oft auch als Spielzeugmodell. (Okay, zugegeben: Manchmal waren die Modelle richtiggehend schlecht, die Ergebnisse haben wir dann tendenziell eher nicht publiziert – normalerweise.)

Ich erzähle Ihnen das, weil dieser Werdegang großen Einfluss auf meine Sichtweise auf diesen relativ jungen Wissenschaftszweig, das maschinelle Lernen, hat. Ich betrachte ihn durchaus mit einer gewissen Skepsis, die ich auch in dieses Buch einfließen lasse. Dennoch hoffe ich, dass der Spaß an diesem Thema nicht allzu sehr darunter leidet. Ich werde etwas später in diesem Kapitel noch einmal darauf zurückkommen. Außerdem werde ich die klassische mathematische Modellierung und das maschinelle Lernen einander gegenüberstellen. Für zwei Bereiche, die das gleiche Ziel verfolgen, ist es doch überraschend, wie groß die Unterschiede sind.

Ich bringe auch meine Philosophie als angewandter Mathematiker mit ein, der gerne Probleme aus der echten Welt löst. Das bedeutet, man schreibt schnell mal ein mathe-

¹ Das Kapitel heißt ja schließlich Einführung, *Introduction*, deshalb stelle ich mich vor, *I introduce myself*.

matisches Modell auf, ohne alles auch mathematisch beweisen zu müssen. Die Tatsache, dass etwas nicht bewiesen ist, bedeutet noch lange nicht, dass es falsch ist. Man will nur so schnell wie möglich zum Fleisch kommen, ohne dass ganze Gemüse essen zu müssen. (Nach all dem bin ich sicher, dass es einige Leser darauf anlegen werden, so viele Fehler wie möglich in diesem Buch zu finden. Wenn Sie welche finden, dann schreiben Sie mir zunächst eine E-Mail, über den Verlag an almut.poll@rheinwerk-verlag.de, bevor Sie mich bei Amazon rezensieren. Feedback ist mir willkommen.)

Dieses Buch enthält absolut keinen Programmiercode. Dies ist gewollt, und zwar aus zwei Gründen: Zum einen bin ich als Programmierer nicht zu gebrauchen. Zum anderen ist es mit dem Vorteil verbunden, dass ich die Methoden so detailliert vermitteln muss, dass Sie sie selbst umsetzen können. Ich kann keine Abkürzungen nehmen, wie zum Beispiel eine mathematisch schwer nachvollziehbare Beschreibung einer Technik, gefolgt von den Worten: »Nun schreiben Sie diese zwei Zeilen Code, drücken die Eingabetaste und schauen Sie sich die hübschen Bilder an.«

Natürlich gibt es viele Programmbibliotheken, die Sie zur Implementierung der von mir beschriebenen Methoden verwenden können, und Sie brauchen nicht zu wissen, wie sie funktionieren, um sie anzuwenden – und das ist in Ordnung. Genauso, wie mich meine völlige Ahnungslosigkeit darüber, wie ein Vergaser funktioniert, nicht davon abhält, mein Jensen Interceptor Cabriolet zu genießen. Und es mir wahrscheinlich trotzdem eine Menge Zeit und Kosten ersparen würde, etwas über Vergaser zu wissen.

Schließlich bietet dieses Buch eine Fülle von Beispielen, fast immer mit realen Daten. Dies ist jedoch kein dicker Forschungswälzer, deshalb dienen diese Beispiele nur der Veranschaulichung. Wäre dies ein Buch, das den neuesten Stand der Forschung beschreibt, dann bräuchte es viel mehr Datenvorbereitung, Tests und Validierungsaufgaben. Gleichzeitig müsste ich auf manche Erklärung zu den Grundlagen verzichten, und auch auf Hinweise, wo etwas schief laufen kann.

Lassen Sie uns aber zum eigentlichen Thema kommen.

1.1 Maschinelles Lernen

Eine Definition aus Wikipedia:

Maschinelles Lernen ist ein Zweig der Computerwissenschaften, der sich damit beschäftigt, mittels statistischer Methoden ein Computersystem zu befähigen, aus Daten zu »lernen« (zum Beispiel durch schrittweise Verbesserung der Aufgabenlösung), ohne explizit dafür programmiert zu werden.

Womit beginnen wir also ... mit Computern? Oder mit Statistik? Oder mit den Daten? Dies ist sicher nicht der geeignete Ort für eine detaillierte Geschichte des maschinellen Lernens. Ich könnte das auch nie so gut wie Google. (Ich habe außerdem dieses persönliche Vorurteil, dass jeder Mathematiker, der sich mit der Geschichte seines Fachs auskennt, kein richtiger Mathematiker ist.) Sollten Sie aber ein paar knackige Sätze für eine Dinnerparty-Unterhaltung brauchen (bevor Sie zu spannenderen Themen wie Religion oder Politik wechseln), dann bitte:

- ▶ Wahrscheinlichkeitsrechnung und Statistik gehen zurück auf das Jahr 1654, als Schlüsselereignis gilt ein Briefwechsel zwischen den französischen Mathematikern Blaise Pascal und Pierre de Fermat zu einem Würfelspiel. Ihr Problem bestand darin, herauszufinden, wie eine Wette gesetzt werden kann, um besser abzuschneiden als der Spielgegner. Dies galt als die Geburtsstunde des Begriffs und Konzepts der statistischen Erwartung.
- ▶ Wem die Erfindung des ersten Taschenrechners zuzuschreiben ist, ist umstritten. Es könnte wieder Pascal oder aber Wilhelm Schickard um 1623 gewesen sein. So oder so war es kaum mehr als eine einfache Additionsmaschine. Der erste *programmierbare* Computer wurde jedoch 1837 von Charles Babbage, oft als Vater der Computerwissenschaften bezeichnet, erfunden – allerdings nie gebaut. Sein Computer sollte Lochkarten für die Programmierung verwenden. Solche Systeme der Programmierung existierten bis in die jüngste Zeit. Für manchen Leser werden vertraute, wenn auch schmerzhaft Erinnerung geweckt.
- ▶ Viel früher schon wurden jedoch Daten gesammelt. Für das maschinelle Lernen brauchen wir Daten – und zwar Unmengen davon. Historisch gesehen wurden Daten oft in Form von Fragebögen und Umfragen erhoben. Das berühmte, rund 900 Jahre alte »Domesday Book« war eine von Wilhelm dem Eroberer in Auftrag gegebene Erhebung in England und Wales. Es diente unter anderem dazu, bei der Steuereintreibung zu helfen. *Plus ça change, plus c'est la même chose.*

Man nehme also Daten und Statistik und bewerfe damit einen Computer – und voilà, erhält man maschinelles Lernen.

1.2 Lernen ist der Schlüssel

Natürlich ist es mehr als das, aber die Crux steckt im Wort »Lernen«. In der klassischen Modellierung würden Sie sich mit einem Stück Papier, einem Bleistift und einem Single Malt hinsetzen und ... dem Algorithmus alles beibringen, was Sie oder besser wir über zum Beispiel das Schachspiel wissen. Sie würden eine Codeanweisung aufschreiben wie »WENN weiße Dame ETWAS ETWAS schwarzer Läufer DANN ETWAS ETWAS.« (Ich habe

seit vierzig Jahren kein Schach mehr gespielt, also entschuldigen Sie bitte den Mangel an Präzision.) Und Sie würden sehr, sehr viele Zeilen mit WENN-, UND- und ODER- Co-deanweisungen brauchen, die dem Programm sagen, was in komplexen Situationen zu tun ist, mit Nuancen in den Positionen der Schachfiguren. Oder ... Sie würden eine Differenzialgleichung aufschreiben, die Ihrer Meinung nach erfasst, wie sich die Inflation zu einer Veränderung der Zinssätze verhält. Die Inflation steigt, also reagiert eine Zentralbank mit einer Erhöhung der Zinssätze. Die Erhöhung der Zinssätze verursacht ...

Sie würden sich die Daten genauer ansehen, vielleicht unter Verwendung von grundlegender Statistik, aber hauptsächlich wäre das Ergebnis ein Produkt des eigenen Gehirns.

Wie auch immer das Problem, Sie würden das Modell selbst erstellen.

Beim maschinellen Lernen sind Ihre Möglichkeiten zu modellieren viel eingeschränkter. Sie entscheiden sich für ein Gerüst, eine Methode, zum Beispiel ein neuronales Netz oder eine Support-Vektor-Maschine, und die Daten erledigen den Rest. (Es wird nicht mehr lange dauern, und die Maschinen kümmern sich auch um die Daten.) Der Algorithmus *lernt*.

1.3 Ein wenig Geschichte

Wie schon erwähnt, werde ich in diesem Buch Geschichtliches kurz halten. Das allein deshalb, weil sich die Geschichte in diesem Bereich rasend schnell ändert durch Ergänzungen und andere Sichtweisen. Ich möchte nur auf zwei Aspekte des maschinellen Lernens hinweisen, die ich persönlich sehr interessant finde.

Lassen Sie mich zunächst Donald Michie vorstellen. Während des Zweiten Weltkrieges, hat Donald Michie zusammen mit Alan Turing daran gearbeitet, Codes zu knacken. Nach dem Krieg promovierte er, und als Professor in Edinburgh widmete er sich Anfang der 60er Jahre der Frage, wie man einen Computer trainieren könnte, um Tic-Tac-Toe alias »Drei gewinnt« zu spielen. Das Wort *trainieren* werden Sie noch ziemlich oft lesen. Nun, eigentlich ging es hierbei nicht so sehr um einen Computer, sondern vielmehr eine Reihe von Streichholzschachteln – ohne Streichhölzer. 304 Schachteln, die so angeordnet sind, dass sie die Phasen eines Tic-Tac-Toe-Spiels repräsentieren. Ich werde hier nicht die Regeln dieses Spiels erklären, so wie ich weiter unten auch nicht die Regeln des Spiels Go erklären werde. Aber jede Phase wird durch ein Gitter dargestellt, in das ein Spieler jeweils ein X und abwechselnd der andere Spieler ein O schreibt mit dem Ziel, drei gleiche Symbole in einer Reihe zu erhalten. (Verflixt, jetzt habe ich die Spielregeln doch erklärt. Aber bitte, erwarten Sie keine Anleitung für das Spiel Go, allein schon, weil ich dieses Spiel selbst nicht zu 100 % verstanden habe.)

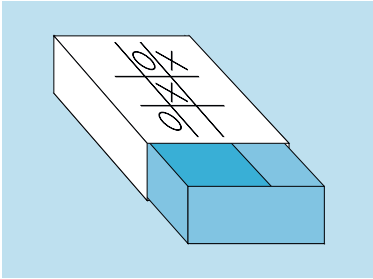


Abbildung 1.1 Eine Streichholzschachtel, die einen Tic-Tac-Toe-Zustand darstellt

In Abbildung 1.1 sehen wir die Skizze einer Streichholzschachtel von Professor Michie. Hier ist Spieler X an der Reihe. Offensichtlich gibt es einen Zug, der zum Gewinn führt (eigentlich zwei davon). Ich weiß das, Sie wissen das, aber würde das ein Computer ebenso wissen? Traditionellerweise würde man einem Computer so programmieren, dass er von allen möglichen Zuständen des Spiels den besten nächsten Spielzug macht. Das ist noch machbar für Tic-Tac-Toe, aber ganz sicher nicht für ein nicht-triviales Spiel. Stattdessen füllte Professor Michie jede der 304 Streichholzschachteln mit farbigen Glaskügelchen, wobei jede Farbe eine der leeren Zellen im Gitter repräsentiert. (Dabei machte er sich gewisse Symmetrien zunutze, sonst hätte man mehr als 304 Schachteln gebraucht.) Sind wir am Zug, dann entnehmen wir der Streichholzschachtel ein zufällig gewähltes Kügelchen, dessen Farbe uns zeigt, wo wir unser X hinzeichnen sollen. Dann ist unser Gegner an der Reihe. Das Spiel geht weiter, und wir bewegen uns von Spielzustand zu Spielzustand, also von Schachtel zu Schachtel. Gewinnen wir, dann erhält jede Schachtel in dem gewonnenen Spiel ein zusätzliches Kügelchen in genau der Farbe, dessen Gitterplatz wir in der Spielaktion ausgewählt haben. Verlieren wir, wird das Kügelchen dieser Farbe hingegen entfernt. Schlussendlich füllen sich die Streichholzschachteln mit Glaskügelchen jener Farben, die den erfolgreichsten Zug in jedem Status repräsentiert. Diese Maschine, die Michie MENACE getauft hat (die Abkürzung steht für *Machine Educable Noughts And Crosses Engine*), hat Tic-Tac-Toe spielen gelernt. Einige wichtige Begriffe sind hier bereits gefallen: *Training*, *Zustand*, *Aktion*, *Belohnung*, *Lernen*. Sie werden noch mehr davon hören, und wir werden auch auf Donalds Streichholzschachteln zurückkommen, und zwar im Kapitel über verstärkendes Lernen. Die nächste spannende Geschichte ist die von Google DeepMind, in der einem Computer das Go-Spiel beigebracht worden war, der daraufhin prompt Lee Sedol schlug, einen professionellen Go-Spieler des 9. Dan. (Nein, ich wusste nicht, dass man den 9. Dan nur durch Sitzen auf dem eigenen Hintern erreichen kann. Das gibt uns allen noch Hoffnung.) AlphaGo verwendete eine Reihe von Techniken, darunter auch neuronale Netze. Wieder lernte es das Spiel ohne explizite Programmierung der optimalen Spielzüge. Kurioserweise ließ man AlphaGo nicht nur

gegen menschliche Gegner, sondern auch gegen sich selber spielen. Um eine Vorstellung davon zu bekommen, wie schwierig Go ist und folglich auch die Programmierung für AlphaGo, stellen wir uns vor, wir würden Donald Michies Streichholzschachtel-Methode verwenden, um die Spielzustände zu umfassen. Man würde wesentlich mehr als 304 Schachteln brauchen, die notwendige Anzahl würde das uns bekannte Universum vollständig mit Streichholzschachteln auffüllen. Ich empfehle an dieser Stelle, sich den Film mit dem kreativen Namen »The AlphaGo Movie« zu Gemüte zu führen. Beobachten Sie dabei die Gesichtszüge aller involvierten Personen. Lee Sedol startete das Spiel durchaus selbstsicher, wirkte bisweilen ziemlich ausgeblutet und gab schließlich auf. Er wirkte auch psychisch angeschlagen. Die meisten Entwickler hingegen jubelten vor Freude ob des weiteren Spielverlaufs. Obwohl unter ihnen einer war, der etwas entsetzt wirkte, so als ob er dächte »Oh Gott, was haben wir nur getan?«

Zuweilen machte der Computer eigenartige Spielzüge, die kein professioneller Go-Spieler je in Erwägung ziehen würde. Im Film wurde darüber diskutiert, ob es sich dabei jeweils um einen Programmierfehler, zu wenig Training oder einen Geniestreich des Computers handelte. Es war immer letzteres! Man sagte auch, das Spiel Go habe sich aufgrund dessen, was Menschen von der Maschine gelernt haben, weiterentwickelt. (Läuft es Ihnen auch gerade kalt den Rücken herunter? Also mir schon.)

Ich versuche, Sie in diesem Buch durch einige der populärsten Techniken des maschinellen Lernens zu leiten. Und zwar auf einem Niveau, auf dem Sie es sofort ausprobieren können. Der Rest dieses Kapitels widmet sich einem kurzen Überblick einiger dieser Verfahren.

Dieses Werk ist nicht als starres Lehrbuch gedacht, sondern es soll Ihr Interesse wecken, und idealerweise sitzen Sie mit einer geöffneten Excel-Datei (oder Ähnlichem) am PC und experimentieren. Es ist die Art Buch, die ich mir gewünscht hätte, als ich begann, mich mit diesem Bereich auseinanderzusetzen.

Noch eine Warnung: Dieses Buch wird zeigen, wie schnell man süchtig nach maschinellen Lernen werden kann, weil es so gottverdammten Spaß macht. Selbstüberschätzung ist jedoch nicht angebracht, wenn man sich mit einem Thema auseinandersetzt, das nach derzeitigem Kenntnisstand die Weltherrschaft übernehmen wird.

1.4 Schlüsselmethodiken in diesem Buch

Wir unterscheiden drei Hauptkategorien des maschinellen Lernens, wobei sich zwei davon durch den Grad der Unterstützung der Maschine unterscheiden. Bei der dritten geht es darum, einer Maschine beizubringen, bestimmte Dinge zu tun.

Prinzipielle Kategorien

► Überwachtes Lernen

Beim *überwachten Lernen* sind Eingabe und Ausgabe gegeben. Die Eingabe liegt als Vektor vor, d. h. mit typischerweise mehr als einer Variablen. Nehmen wir z. B. digitale Bilder von Hunden, die zu Vektoren transformiert werden, die die Pixelfarben repräsentieren. Diese Bilder wurden als Jagdhund, Pudel, Terrier etc. klassifiziert. Wir trainieren nun unseren Algorithmus auf Basis dieser Bilder in der Hoffnung, dass ein neues Bild korrekt klassifiziert wird. Natürlich können wir überwachtes Lernen auch für Regressionsaufgaben verwenden, in denen für jede Eingabe numerische Werte für die jeweilige Ausgabe vorhergesagt werden (zum Beispiel besteht die Eingabe aus Temperatur und Tageszeit, und wir versuchen, den Stromverbrauch vorherzusagen). Überwachtes Lernen umfasst viele verschiedene Algorithmen, von denen wir einige im Folgenden erwähnen.

► Unüberwachtes Lernen

Man spricht von *unüberwachtem Lernen*, wenn die Eingabedaten beim Training nicht mit passenden Ausgabewerten versehen sind. Das bedeutet, wir haben nur Eingabedaten, keine Ausgabedaten (die auch zuweilen als *Label* oder *Kennzeichnung* bezeichnet werden und die Daten als *gekennzeichnet*). Der Algorithmus ist gewissermaßen auf sich allein gestellt. Dieser Algorithmus hat eher die Aufgabe, Beziehungen oder Muster für Sie zu finden. Wir zeigen dem Computer digitale Hundebilder, und dieser versucht, diese Bilder zu gruppieren oder zu bündeln auf Basis irgendwelcher Eigenschaften, die *er* für am wichtigsten hält. Möglicherweise erhalten wir als Ergebnis Gruppen oder Cluster, die wir dann als Jagdhund, Pudel oder Terrier ... oder vielleicht als schwarz, braun, weiß ... bezeichnen. Oder der Computer liefert Gruppen nach vollkommen unterschiedlichen Kriterien, die wir Menschen nicht mal bemerken würden. Unüberwachtes Lernen mag auf den ersten Blick etwas eigenartig erscheinen, aber um es zu verstehen, können Sie sich vorstellen, ich würde zu Ihnen sagen: »Hier ist der Inhalt meiner Schreibtischlade. Bitte sortiere die Sachen.« Auch für das unüberwachte Lernen gibt es viele verschiedene Algorithmen, die wir uns später noch genauer ansehen.

► Verstärkendes Lernen

Beim *verstärkenden Lernen* lernt ein Algorithmus, etwas zu *tun*. Dabei wird der Algorithmus für erfolgreiches Vorgehen belohnt und/oder für misslungenes Verhalten bestraft. (Womit wir wieder bei den Hunden wären!) Das zuvor erwähnte MENACE war dafür ein einfaches Beispiel, Belohnungen und Bestrafungen bestanden dabei aus Zugabe oder Wegnahme von Streichhölzern zu oder von den Schachteln. Computer erfolgreich Brettspiele oder Videospiele aus den 80ern spielen zu lassen, scheint

gerade hochmodern zu sein. Für diese Aufgaben werden Algorithmen des verstärkenden Lernens verwendet.

Wichtige Verfahren

Ich beschreibe im Folgenden viele der wichtigsten Verfahren, die auf den zuvor erwähnten prinzipiellen Kategorien aufbauen. Die Reihenfolge orientiert sich hauptsächlich an ihrer mathematischen Komplexität. Die meisten dieser Verfahren können mit einigen wenigen Zeilen Python-Code implementiert werden, meine Aufgabe sehe ich allerdings nicht darin, solchen Code vorzugeben, sondern vielmehr in der Erklärung dessen, was der Code eigentlich macht. In der Folge sehen Sie eine Liste der in diesem Buch beschriebenen Verfahren des überwachten und unüberwachten Lernens. Das verstärkende Lernen wird separat behandelt, und zwar ganz am Ende des Buches.

Ich persönlich sehe einige dieser Techniken nicht unbedingt als maschinelles Lernen, sie haben zu viel Ähnlichkeit mit klassischen statistischen Methoden. Aber ich habe sie trotzdem hier aufgenommen, weil sie allgemein als maschinelles Lernen akzeptiert sind – und, ja, sie haben mit maschinellem Lernen gemein, viele, viele Daten zu verwenden.

- ▶ **K-nächste-Nachbarn:** *K-nächste-Nachbarn* ist eine Technik des überwachten Lernens, bei der wir Abstände zwischen jedem neuen Datenpunkt (in beliebig vielen Dimensionen) und den K am nächsten gelegenen unserer bereits klassifizierten Daten messen und so schließen, zu welcher Klasse unser neuer Datenpunkt gehört. Ein einfaches Beispiel: Wir sind auf einer Party, und von den fünf Personen, die uns am nächsten stehen, tragen drei eine Brille, also sind wir wahrscheinlich auch Brillenträger. Diese Methode kann auch zur Regression verwendet werden.
- ▶ **K-Means Clustering:** *K-Means Clustering* ist eine Technik des unüberwachten Lernens. Wir haben viele Daten in Vektorform (also mehrdimensionale Information für jeden Punkt), und wir haben K Zahlen, Punkte oder Zentroide im Raum. Jeder Datenpunkt ist mit dem nächstgelegenen Zentroid verbunden, und das definiert, zu welcher Kategorie jeder Datenpunkt gehört. Dann finden wir die optimale Position für diese Zentroide, die uns die beste Einteilung in die K-Klassen ermöglicht. Schließlich experimentieren wir mit der Zahl K herum, um zu sehen, welche Auswirkung das auf die Einteilung der Daten in Kategorien hat.
- ▶ **Naiver Bayes-Klassifikator:** Der *naive Bayes-Klassifikator* ist ein Vertreter des überwachten Lernens, bei dem das Bayes-Theorem verwendet wird, um die Wahrscheinlichkeit dafür zu berechnen, dass neue Datenpunkte in verschiedenen Klassen liegen. Das Wort *naiv* bezieht sich auf vereinfachende Annahmen, die dabei getroffen werden, obwohl sie in der Praxis selten zutreffen, was aber dennoch zu guten Ergebnissen führt.

- **Regression:** *Regressionsverfahren* sind überwachte Lerntechniken, die versuchen, eine numerisch abhängige Variable in Form von unabhängigen Variablen zu erklären. Wahrscheinlich kennen Sie den Begriff Regression bereits von einem bekannten Vertreter, der linearen Regression. Aber mit komplizierteren Regressionstypen können Sie viel komplexere Aufgaben lösen.
- **Support-Vektor-Maschinen:** Ein weiteres Verfahren des überwachten Lernens sind die sogenannten *Support-Vektor-Maschinen*, eine Technik, die Daten in Klassen einteilt, je nachdem, auf welcher Seite einer Hyperebene im Merkmalsraum ein Datenpunkt liegt.
- **Selbstorganisierende Karten:** Eine *selbstorganisierende Karte* ist ein Beispiel eines unüberwachten Lernverfahrens, das hochdimensionale Daten in schöne, typischerweise zweidimensionale Bilder zur Visualisierung von Beziehungen zwischen Datenpunkten umwandelt. Stellen Sie sich ein Schachbrett vor, und setzen Sie in die Felder Datenpunkte mit ähnlichen Eigenschaften. Sie allerdings spezifizieren diese Merkmale nicht, sie werden als Ergebnis des Algorithmus gefunden.
- **Entscheidungsbäume:** Wiederum ein Verfahren des überwachten Lernens. Ein *Entscheidungsbaum* ist wie ein Flussdiagramm. »Wie viele Beine hat es? Zwei, vier, mehr als vier?« Vier. Als Nächstes: »Hat es einen Schwanz? Ja oder nein.« Ja. Und so weiter. Wie ein Spiel mit 20 Fragen. Aber kann eine Maschine auf diese Weise lernen und bessere Entscheidungsbäume aufbauen als Menschen? Gibt es eine beste Reihenfolge, in der die Fragen gestellt werden, damit die Klassifizierung schnell und genau erfolgt? Überraschenderweise kann dieses Verfahren auch für Regressionsaufgaben verwendet werden.
- **Neuronale Netzwerke:** Ein *neuronales Netz* (manchmal auch als *künstliches neuronales Netz* bezeichnet) ist ein maschinelles Lernverfahren, das versucht, die Funktionsweise unseres Gehirns zu modellieren. Dabei wird ein Eingangsvektor transformiert, typischerweise durch Multiplikation mit einer Matrix. So weit, so linear. Dann wird das Ergebnis in eine nicht lineare *Aktivierungsfunktion* eingesetzt. Das bezeichnen wir dann als eine *Schicht*. Das Ergebnis wird dann der nächsten Schicht übergeben und wieder transformiert usw. In dieser Form kann man es als Anwendung einer nicht linearen Funktion der ursprünglichen Daten sehen. Aber es wird noch komplizierter, wenn die Daten nicht nur in eine Richtung gehen, sondern durch eine Rückkopplung in die entgegengesetzte Richtung laufen. Die Parameter im neuronalen Netz werden durch *Training* gefunden. Die Technik kann sowohl für überwachtes als auch für unüberwachtes Lernen verwendet werden.

1.5 Klassische mathematische Modellierung

Ich möchte kurz erklären, worin sich das maschinelle Lernen vom klassischen mathematischen Modellieren unterscheidet. Ich muss betonen, dass ich nicht über statistisches Modellieren oder Ökonometrie spreche, obwohl es durchaus einige Ähnlichkeiten mit dem maschinellen Lernen gibt. Zu einem gewissen Grad entspringt das meiner persönlichen Sicht und meinem Erfahrungsschatz in der angewandten mathematischen Modellierung.

Es gibt da einen anerkannten, oftmals beschrittenen Pfad, dem (wir) mathematische Modellierer folgen, wenn wir uns einem neuen Problem stellen. Er geht in etwa so:

Zuerst bestimme deine Variablen

Man unterscheidet zwei Typen von Variablen: unabhängige und abhängige Variablen. Unabhängige Variablen beschreiben den Raum, in dem unser Problem definiert ist. Typische Vertreter für unabhängige Variablen wären zum Beispiel Raum- und Zeitkoordinaten x , y , z und t .

Im Gegensatz dazu stehen die abhängigen Variablen, für die wir eine Lösung suchen. Sie könnten zum Beispiel eine Wärmetransportaufgabe lösen wollen, in der die Temperatur typischerweise die abhängige Variable darstellen würde. Als weiteres Beispiel betrachten wir die Schwingung einer Cello- (oder Gitarren-) Saite für jene auf den billigen Plätzen. Die abhängige Variable wäre hier die Distanz von der Saite in Ruhestellung. (Hier hätten wir also eine Distanz als abhängige Variable, aber auch die unabhängigen Variablen enthalten eine Distanz – nämlich den Abstand entlang der Saite – und natürlich die Zeit.)

Auf jeden Fall wollen wir mit einer Minimalzahl an Variablen auskommen, um ein System und sein Verhalten zu spezifizieren. Einstein soll einmal gesagt haben: »Man muss die Dinge so einfach wie möglich machen. Aber nicht einfacher.« Das ist wahrscheinlich die Regel Nummer 1 für die mathematische Modellierung. Regel Nummer 2 lautet: »Willst du es komplizierter, dann nimm eine Eigenschaft nach der anderen dazu.« Da haben wir bereits einen Riesenunterschied zu maschinellem Lernen, wo man dazu tendiert, zu Beginn einfach alles hineinzuwurfen, und die Küchenspüle noch dazu.

In diesem Stadium ist das Ziel der klassischen mathematischen Modellierung, das kleinstmögliche nicht triviale mathematische Problem zu definieren, das von einem gewissen Interesse ist.

Finde die hauptsächlichen Treiber deines Systems heraus

Was treibt Ihr System? Gibt es Kräfte, die etwas schieben oder ziehen? Prallen Moleküle zufällig aufeinander? Bewegen Käufer und Verkäufer den Milchpreis hinauf oder hinun-