O'REILLY®

Merkmals-konstruktion

für Machine Learning

PRINZIPIEN UND TECHNIKEN DER DATENAUFBEREITUNG

Alice Zheng & Amanda Casari

Übersetzung von Thomas Lotze

Merkmalskonstruktion für Machine Learning

Prinzipien und Techniken der Datenaufbereitung

Alice Zheng & Amanda Casari

Deutsche Übersetzung von Thomas Lotze

O'REILLY®

Alice Zheng und Amanda Casari

Lektorat: Alexandra Follenius
Übersetzung: Thomas Lotze
Korrektorat: Sibylle Feldmann, *www.richtiger-text.de*
Satz: III-satz, *www.drei-satz.de*
Herstellung: Stefanie Weidner
Umschlaggestaltung: Karen Montgomery, Michael Oréal, *www.oreal.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

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:
Print 978-3-96009-093-9
PDF 978-3-96010-249-6
ePub 978-3-96010-250-2
mobi 978-3-96010-251-9

1. Auflage

Wieblinger Weg 17
69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.
O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Authorized German translation of the English edition of *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, ISBN 978-1-491-95324-2 © 2018 Alice Zheng, Amanda Casari. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: kommentar@oreilly.de.

5 4 3 2 1 0

Inhalt

Vorwort

Einleitung

Maschinelles Lernen bedeutet, mathematische Modelle an Daten anzupassen, um daraus Erkenntnisse oder Vorhersagen zu gewinnen. Diese Modelle erwarten als Eingabe sogenannte Merkmale. Ein *Merkmal* ist eine numerische Darstellung eines bestimmten Aspekts von Rohdaten. In der Machine-Learning-Pipeline vermitteln Merkmale zwischen Daten und Modellen. *Merkmalskonstruktion* (engl. Feature Engineering) wird der Vorgang genannt, Merkmale aus Rohdaten zu gewinnen und in eine Form zu bringen, die sich für das Machine-Learning-Modell eignet. Sie ist ein entscheidender Schritt in der Machine-Learning-Pipeline, denn die richtigen Merkmale können den schwierigen Vorgang des Modellierens erleichtern und so eine bessere Qualität der Ergebnisse ermöglichen, die die Pipeline ausgibt. Anwender aus der Praxis wissen, dass beim Aufbau einer Machine-Learning-Pipeline die Merkmalskonstruktion und die Datenbereinigung die meiste Zeit benötigen. Trotz seiner Bedeutung wird das Thema jedoch selten eigenständig behandelt. Das mag daran liegen, dass die richtigen Merkmale nur im Kontext sowohl des Modells als auch der Daten definiert werden können; da es so unterschiedliche Daten und Modelle gibt, ist es also schwer, die Merkmalskonstruktion projektübergreifend zu verallgemeinern.

Dennoch geschieht Merkmalskonstruktion nicht einfach aus dem Stegreif. Es gibt zugrunde liegende Prinzipien, die sich am besten an Beispielen zeigen lassen. Jedes Kapitel dieses Buchs widmet sich einer Aufgabe aus der Datenanalyse: der Darstellung von Text- oder Bilddaten, der Dimensionsreduktion automatisch erzeugter Merkmale, der Anwendung von Normierung usw. Stellen Sie sich das Buch als eine Sammlung miteinander verwobener Kurzgeschichten und nicht als einen einzigen langen Roman vor. Jedes Kapitel bietet einen Einblick in die Vielzahl der bekannten Verfahren zur Merkmalskonstruktion. Zusammen veranschaulichen sie die übergreifenden Prinzipien.

Ein Fachgebiet zu beherrschen, bedeutet nicht nur, die Definitionen zu kennen und die Formeln herleiten zu können. Es genügt nicht, zu wissen, wie ein Mechanismus funktioniert und was er vermag – man muss auch verstehen, woher sein Aufbau rührt, wie er sich zu anderen Techniken verhält und welche Vor- und

Nachteile jede Herangehensweise hat. Meisterschaft bedeutet, genau zu wissen, wie etwas gemacht wird, ein Gespür für die Grundprinzipien zu haben und diese in das eigene schon bestehende Wissensgerüst einordnen zu können. Man wird kein Meister, indem man einfach ein Buch liest, obwohl ein gutes Buch neue Türen öffnen kann. Es braucht praktische Erfahrung – die Ideen müssen angewandt werden, was wiederum ein iterativer Vorgang ist. Mit jeder Iteration lernen wir die Ideen besser kennen und werden immer geschickter und kreativer bei ihrer Anwendung. Das Ziel dieses Buchs ist es, die Anwendung der darin vorgestellten Ideen zu erleichtern.

Dieses Buch versucht, immer zuerst die Grundgedanken und danach die Mathematik zu lehren. Statt zu diskutieren, *wie* etwas gemacht wird, wollen wir erklären, *warum* es gemacht wird. Unser Ziel ist es, die *Intuition* hinter den Ideen zu vermitteln, sodass Sie als Leser ein Verständnis dafür bekommen, wie und wann sie anzuwenden sind. Es gibt aber auch Unmengen von Beschreibungen und Bildern für diejenigen, die eher auf andere Weise lernen. Dazu liefern wir die mathematischen Formeln, um der Intuition Genauigkeit zu verleihen und eine Brücke zwischen diesem Buch und anderen Angeboten zu schlagen.

Das Buch setzt Kenntnisse der Grundbegriffe des maschinellen Lernens voraus, etwa was ein Modell oder ein Vektor ist, aber es enthält auch einen Auffrischungskurs, damit wir alle auf demselben Stand sind. Erfahrung mit linearer Algebra, Wahrscheinlichkeitsverteilungen und Optimierung sind hilfreich, aber nicht notwendig.

Python-Bibliotheken

Die Codebeispiele in diesem Buch sind in Python geschrieben und verwenden eine Reihe freier und quelloffener Pakete. Die Bibliothek NumPy (*http://www.numpy.org/*) implementiert numerische Vektor- und Matrixoperationen. Pandas (*http://pandas.pydata.org/*) stellt den DataFrame als Grundbaustein der Datenanalyse in Python zur Verfügung. scikit-learn (*http://scikit-Learn.org/stable/*) ist ein allgemeines Machine-Learning-Paket mit einer umfassenden Sammlung von Modellen und Merkmalstransformationen. Matplotlib (*https://matplotlib.org/*) und die Stilbibliothek Seaborn (*https://seaborn.pydata.org/*) bieten Unterstützung für Diagramme und Visualisierung. Die Beispiele sind als Jupyter-Notebooks in unserem GitHub-Repository (*https://github.com/alicezheng/feature-engineering-book*) zu finden.

Wegweiser durch dieses Buch

Die ersten Kapitel bieten einen gemächlichen Einstieg für Neulinge auf dem Gebiet der Datenanalyse und des maschinellen Lernens. Kapitel 1 stellt die Grundkonzepte in der Machine-Learning-Pipeline vor: Daten, Modelle, Merkmale usw. In Kapitel 2 schauen wir uns die Anfänge der Merkmalskonstruktion für numerische Daten an: Filtern, Klassifikation, Skalierung, logarithmische und Potenztransformationen

sowie Kreuzmerkmale. Kapitel 3 taucht ein in die Merkmalskonstruktion für natürlichen Text und untersucht Techniken wie Bag-of-Words, *n*-Gramme und Phrasenerkennung. Kapitel 4 untersucht den Algorithmus TF-IDF (Begriffshäufigkeit – inverse Dokumentenhäufigkeit, engl. *Term Frequency – Inverse Document Frequency*) als ein Beispiel der Merkmalsskalierung und diskutiert, warum er funktioniert. Das Buch nimmt bei Kapitel 5 Fahrt auf, wenn wir über effiziente Kodierungsverfahren für kategoriale Variablen sprechen, darunter Merkmals-Hashing und Klassenzählung. Spätestens wenn wir in Kapitel 6 bei der Hauptkomponentenzerlegung (PCA, engl. *Principal Component Analysis*) angelangt sind, befinden wir uns tief im Land des maschinellen Lernens. Kapitel 7 betrachtet den *k*-Means-Algorithmus als Verfahren zur Merkmalsgewinnung, wodurch das nützliche Konzept der Stapelung von Modellen aufgezeigt wird. Kapitel 8 beschäftigt sich ganz mit Bildern, die im Hinblick auf Merkmalsgewinnung eine viel größere Herausforderung darstellen als Textdaten. Wir schauen uns zwei Verfahren der Merkmalsgewinnung per Hand an, SIFT und HOG, bevor wir anschließend das Deep Learning als neueste Technik zur Merkmalsgewinnung für Bilder erläutern. Zum Abschluss zeigen wir in Kapitel 9 anhand ausführlicher Beispiele ein paar unterschiedliche Verfahren, indem wir einen Empfehlungsalgorithmus für einen Datensatz von akademischen Aufsätzen erstellen.

Merkmalskonstruktion ist ein weites Feld, und jeden Tag werden neue Verfahren entwickelt, insbesondere auf dem Gebiet des automatisierten Erlernens von Merkmalen. Um das Buch auf einen handlichen Umfang zu beschränken, mussten wir einiges auslassen. So behandeln wir nicht die Fourier-Analyse für Audiodaten, obwohl das ein wunderschönes Thema und nah verwandt mit der Eigenfunktionszerlegung in der linearen Algebra ist, die wir in den Kapiteln 4 und 6 streifen. Auch überspringen wir die Diskussion zufälliger Merkmale, die ebenso eng mit der Fourier-Analyse verknüpft sind. Wir bieten zwar eine Einführung ins Erlernen von Merkmalen für Bilddaten durch Deep Learning, gehen aber nicht näher auf die zahllosen in der Weiterentwicklung befindlichen Deep-Learning-Modelle ein. Weiterführende Forschungsfelder, etwa Zufallsprojektionen, komplexe Modelle zur Merkmalsgewinnung aus Text wie word2vec und Brown-Clustering sowie Latent-Raum-Modelle wie Latente Dirichlet-Allokation und Matrixfaktorisierung, lassen wir ebenfalls aus. Wenn Ihnen diese Begriffe nichts sagen, haben Sie Glück. Sollten Sie sich jedoch für die allerneueste Forschung bei der Merkmalskonstruktion interessieren, dann ist dies vermutlich nicht das richtige Buch für Sie.

In diesem Buch verwendete Konventionen

Die folgenden typografischen Konventionen werden in diesem Buch verwendet:

Kursiv
: Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateiendungen.

`Nichtproportionalschrift`
: Wird für Programmlistings sowie für Programmelemente in Textabschnitten wie Namen von Variablen und Funktionen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörtern verwendet.

`Nichtproportionalschrift fett`
: Kennzeichnet Befehle oder anderen Text, den der Nutzer wörtlich eingeben soll.

`Nichtproportionalschrift kursiv`
: Kennzeichnet Text, den der Nutzer durch eigene oder zum Kontext passende Werte ersetzen soll.

Das Buch enthält außerdem zahlreiche Gleichungen der linearen Algebra. Wir folgen bezüglich der Notation diesen Konventionen: Skalare werden mit kursiven Kleinbuchstaben geschrieben (z. B. a), Vektoren mit fetten Kleinbuchstaben (z. B. $\mathbf{v}$) und Matrizen mit fetten kursiven Großbuchstaben (z. B. $\boldsymbol{U}$).

Dieses Symbol kennzeichnet einen Tipp oder Vorschlag.

Dieses Symbol kennzeichnet eine allgemeine Bemerkung.

Dieses Symbol kennzeichnet eine Warnung oder einen Rat zur Vorsicht.

Verwendung von Codebeispielen

Zusatzmaterial wie Codebeispiele oder Übungen können Sie von *https://github.com/alicezheng/feature-engineering-book* herunterladen.

Dieses Buch soll Ihnen helfen, Ihre Arbeit zu erledigen. Im Allgemeinen dürfen Sie die Codebeispiele aus diesem Buch in Ihren eigenen Programmen und der dazugehörigen Dokumentation verwenden. Sie müssen uns dazu nicht um Erlaubnis fragen, solange Sie nicht einen wirklich signifikanten Teil des Codes reproduzieren. Beispielsweise benötigen Sie keine Erlaubnis, um ein Programm zu schreiben, in dem mehrere Codefragmente aus diesem Buch vorkommen. Wollen Sie dagegen eine DVD mit Beispielen aus Büchern von O'Reilly verkaufen oder verteilen, benötigen Sie eine Erlaubnis. Eine Frage zu beantworten, indem Sie aus diesem Buch zitieren und ein Codebeispiel wiedergeben, benötigt keine Erlaubnis. Eine beträcht-

liche Menge Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts aufzunehmen, bedarf hingegen einer Erlaubnis.

Wir freuen uns darüber, zitiert zu werden, verlangen es aber nicht. Ein Zitat enthält Titel, Autor, Verlag und ISBN. Ein Beispiel: »*Merkmalskonstruktion für Machine Learning* von Alice Zheng und Amanda Casari (O'Reilly). Copyright 2018 Alice Zheng und Amanda Casari, 978-3-96009-093-9.«

Wenn Sie glauben, dass Sie die Codebeispiele über eine angemessene Nutzung oder die oben gewährte Nutzungserlaubnis hinaus verwenden, dann kontaktieren Sie uns bitte unter *kommentar@oreilly.de*.

Danksagung

Zuallererst möchten wir unseren Lektoren Shannon Cutt und Jeff Bleiel dafür danken, dass sie uns bei unserem ersten Buch durch den uns beiden bis dato unbekannten Marathon einer Buchveröffentlichung geleitet haben. Ohne die enge Zusammenarbeit mit euch hätte dieses Buch nie das Licht der Welt erblickt. Ebenso vielen Dank an Ben Lorica, O'Reilly-Mastermind, dessen Ermutigungen und Bestätigungen das Buch von einer verrückten Idee zu einem tatsächlichen Produkt brachten. Danke an Kristen Brown und das Produktionsteam von O'Reilly für ihre überragende Sorgfalt bei Details und ihre extreme Geduld bei unseren Rückmeldungen.

Wenn es stimmt, dass es ein ganzes Dorf braucht, um ein Kind aufzuziehen, dann braucht es ein ganzes Parlament von Datenanalytikern, um ein Buch zu veröffentlichen. Wir wissen jeden Vorschlag und jeden Hinweis auf mögliche Verbesserungen und alle Nachfragen zu Unklarheiten zu schätzen. Andreas Müller, Sethu Raman und Antoine Atallah nahmen sich kostbare Zeit für technische Überprüfungen. Antoine tat das nicht nur blitzschnell, sondern stellte dabei auch seine dicken Maschinen für Experimente zur Verfügung. Ted Dunnings gewandte Beherrschung von Statistik und seine Meisterschaft im maschinellen Lernen sind legendär. Er ist zudem unglaublich großzügig mit seiner Zeit und seinen Ideen, und von ihm stammen buchstäblich die Methode und das Beispiel, die im Kapitel über den *k*-Means-Algorithmus beschrieben werden. Owen Zhang gewährte einen Blick in seine Kaggle-Schatzkiste zur Verwendung von Antwortraten-Merkmalen, die wir der von Misha Bilenko gesammelten Machine-Learning-Folklore über die Klassenzählung zugesellten. Ein weiteres Dankeschön geht an Alex Ott, Francisco Martin und David Garrison für zusätzliches Feedback.

Besonderer Dank von Alice

Ich möchte der Familie GraphLab/Dato/Turi für ihre großzügige Unterstützung in der ersten Phase dieses Projekts danken. Die Idee erwuchs aus dem Umgang mit unseren Anwendern. Beim Bau einer ganz neuen Machine-Learning-Plattform für

Datenanalytiker war uns aufgefallen, dass die Welt ein systematischeres Verständnis von Merkmalskonstruktion braucht. Danke an Carlos Guestrin für die Freistellung vom geschäftigen Start-up-Leben, damit ich mich aufs Schreiben konzentrieren konnte.

Danke an Amanda, die als technische Gutachterin begann und später einsprang, um diesem Buch zum Leben zu verhelfen. Du bringst Dinge über die Ziellinie! Nachdem dieses Buch nun fertig ist, müssen wir ein neues Projekt finden, und sei es nur, um unsere Arbeitssitzungen bei Tee und Kaffee und Sandwiches und leckerem Essen fortzusetzen.

Ein besonderes Dankeschön an meine Freundin und Heilerin Daisy Thompson für ihre beständige Unterstützung während aller Phasen dieses Projekts. Ohne deine Hilfe hätte ich viel länger gebraucht, um mich darauf einzulassen, und mich über den Marathon geärgert. Du hast, wie immer mit deiner Arbeit, Licht und Leichtigkeit in dieses Projekt gebracht.

Besonderer Dank von Amanda

Da dies ein Buch und keine Auszeichnung fürs Lebenswerk ist, will ich versuchen, meinen Dank auf das vorliegende Projekt zu beschränken.

Vielen, vielen Dank an Alice dafür, dass sie mich als technische Lektorin und dann Mitautorin eingebracht hat. Ich lerne ständig so viel von dir, unter anderem wie man bessere mathematische Scherze macht und komplexe Konzepte verständlich erklärt.

Nur der Reihenfolge nach zuletzt geht ein ganz besonderer Dank an meinen Mann Matthew für das nahezu unmögliche Kunststück, mir Halt zu geben, mich auf dem Weg zu meinem nächsten Ziel zu bestärken und nie zuzulassen, dass ein Konzept vage abgetan wird. Du bist der beste Partner und mein Lieblingskomplize. In den größten wie den kleinsten sonnigen Momenten spornst du mich an, dich stolz zu machen.

KAPITEL 1

Die Machine-Learning-Pipeline

Bevor wir uns mit Merkmalskonstruktion beschäftigen, wollen wir uns die Machine-Learning-Pipeline als Ganzes anschauen, um unseren Platz im Gesamtsystem zu finden. Zu diesem Zweck betrachten wir zunächst Grundbegriffe wie *Daten* und *Modelle*.

Daten

Als *Daten* bezeichnen wir Beobachtungen realer Phänomene. So können Daten von Aktienmärkten Beobachtungen der täglichen Aktienpreise, Gewinnankündigungen einzelner Firmen und sogar Meinungsartikel von Fachleuten umfassen. Persönliche biometrische Daten wären unter anderem minütliche Messungen von Pulsfrequenz, Blutzuckerspiegel, Blutdruck usw., und Daten zur Kundenanalyse sind beispielsweise Aussagen wie »Alice hat am Sonntag zwei Bücher gekauft«, »Bob hat diese Seiten der Website angesehen« und »Charlie hat auf den Link zum Sonderangebot aus der letzten Woche geklickt«. Wir könnten endlos Beispiele aus ganz unterschiedlichen Anwendungsgebieten finden.

Jedes Einzelteil dieser Daten gewährt Einblick in einen kleinen Aspekt der Wirklichkeit. Die Gesamtheit aller dieser Beobachtungen liefert uns ein Bild des Ganzen. Aber das Bild ist chaotisch, weil es aus Tausenden kleinen Teilen zusammengesetzt ist und wir es immer mit Messrauschen und fehlenden Teilen zu tun haben.

Aufgaben

Warum sammeln wir Daten? Es gibt Fragen, die wir mithilfe von Daten beantworten können – Fragen wie »In welche Aktien sollte ich investieren?« oder »Wie kann ich gesünder leben?« oder »Wie kann ich den wechselnden Geschmack meiner Kunden verstehen, damit ich sie besser bedienen kann?«.

Der Pfad von Daten zu Antworten ist gespickt mit falschen Fährten und Sackgassen (siehe Abbildung 1-1). So mancher vielversprechende Ansatz wird nicht aufgehen, während ein vages Bauchgefühl zur besten Lösung führen kann. Die Arbeit mit

Daten ist oftmals ein mehrstufiger, iterativer Prozess. Aktienpreise werden beispielsweise an der Börse beobachtet, in einer Datenbank gespeichert, von einer Firma gekauft, in einen Hive-Store auf einem Hadoop-Cluster umgewandelt, von einem Skript aus dem Store geholt, von einem anderen Skript ausgedünnt, aufbereitet und bereinigt, in eine Datei geschrieben und in ein Format überführt, das Sie mit der Modellierungsbibliothek Ihrer Wahl in R, Python oder Scala ausprobieren können. Die Vorhersagen werden dann wiederum in eine CSV-Datei geschrieben und von einem Auswertungsprogramm gelesen. Das Modell durchläuft mehrere Iterationen, wird von Ihrer Produktionsabteilung in C++ oder Java neu geschrieben und auf der gesamten Datenmenge laufen gelassen, bevor die fertigen Vorhersagen in eine weitere Datenbank gefüllt werden.

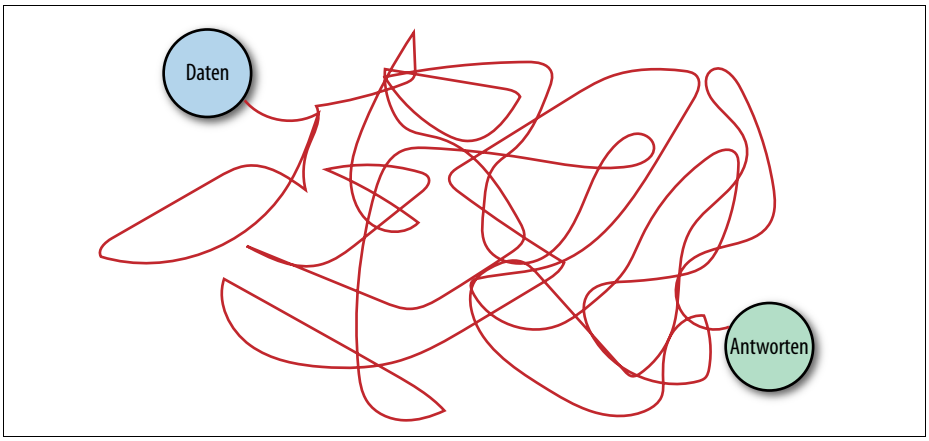

Abbildung 1-1: Der Garten der verschlungenen Pfade von Daten zu Antworten

Wenn wir jedoch das Chaos der Werkzeuge und Systeme für einen Moment ausblenden, können wir erkennen, dass an dem Vorgang zwei mathematische Größen beteiligt sind, die das tägliche Brot des maschinellen Lernens darstellen: *Modelle* und *Merkmale*.

Modelle

Zu versuchen, die Welt durch Daten zu verstehen, ist so, als wolle man die Wirklichkeit aus einem verrauschten, unvollständigen Puzzle mit ein paar überschüssigen Teilen zusammensetzen. Hier kommt die mathematische – insbesondere die statistische – Modellierung ins Spiel. Die Sprache der Statistik kennt Begriffe für viele häufig auftretende Eigenschaften von Daten, darunter *falsch*, *redundant* und *fehlend*. Falsche Daten ergeben sich aus Messfehlern, redundante Daten enthalten ein und dieselbe Information mehrfach: So kann ein Wochentag als kategoriale Variable mit den Ausprägungen »Montag«, »Dienstag«, ..., »Sonntag« und zugleich noch einmal als ganze Zahl zwischen 0 und 6 vorliegen. Ist diese Information über

den Wochentag für einige Datenpunkte nicht vorhanden, haben wir es wiederum mit fehlenden Daten zu tun.

Ein *mathematisches Modell* von Daten beschreibt die Beziehungen zwischen verschiedenen Aspekten der Daten. Beispielsweise könnte ein Modell, das Aktienpreise vorhersagt, aus einer Formel bestehen, die die bisherigen Gewinne einer Firma, frühere Aktienpreise und die Branche auf die Vorhersage für den Aktienpreis abbildet. Ein Modell für Musikempfehlungen könnte anhand der Hörgewohnheiten von Anwendern eine Ähnlichkeit zwischen ihnen messen und denjenigen, die sich viele ähnliche Titel angehört haben, dieselben Künstler empfehlen.

Mathematische Formeln stellen Beziehungen zwischen numerischen Größen her. Aber Rohdaten sind oft nicht numerisch. (Die Aussage »Alice kaufte am Mittwoch die Trilogie *Der Herr der Ringe*« ist ebenso wenig numerisch wie die Buchbesprechung, die sie später schreibt.) Es muss also etwas geben, das die beiden Welten verbindet. An dieser Stelle kommen Merkmale ins Spiel.

Merkmale

Ein *Merkmal* ist eine numerische Darstellung von Rohdaten. Man kann Rohdaten auf vielerlei Weise in numerische Messungen verwandeln, weshalb Merkmale alles Mögliche sein können. Natürlich müssen sich Merkmale aus den vorhandenen Daten ableiten lassen. Weniger offensichtlich ist vielleicht, dass sie auch ans Modell gebunden sind; manche Modelle eignen sich besser für bestimmte Arten von Merkmalen und umgekehrt. Die richtigen Merkmale zeichnen sich dadurch aus, dass sie relevant für die zu lösende Aufgabe und leicht in das Modell einzuspeisen sind. *Merkmalskonstruktion* ist der Vorgang, diejenigen Merkmale zu formulieren, die sich für die gegebenen Daten, das Modell und die zu lösende Aufgabe am besten eignen.

Die Anzahl der Merkmale ist ebenfalls von Bedeutung. Ohne ausreichend viele aussagekräftige Merkmale wird das Modell die gestellte Aufgabe nicht bewältigen. Hat man zu viele oder größtenteils irrelevante Merkmale, wird es aufwendiger und schwieriger sein, das Modell anzulernen, und beim Anlernen könnte irgendetwas schiefgehen, sodass das Modell an Leistungsfähigkeit verliert.

Modellbewertung

Merkmale und Modelle sind das Bindeglied zwischen Rohdaten und gesuchten Erkenntnissen (siehe Abbildung 1-2). Zum Arbeitsablauf beim maschinellen Lernen gehört es, nicht nur das Modell, sondern auch die Merkmale auszuwählen. Das ist ein Balanceakt: Beides beeinflusst einander. Gute Merkmale vereinfachen den nachfolgenden Modellierungsschritt und sorgen dafür, dass das daraus entstehende Modell die gewünschte Aufgabe besser erfüllen kann. Schlecht gewählte

Merkmale erfordern ein viel komplizierteres Modell, um dasselbe Ergebnis zu erreichen. Im Rest dieses Buchs besprechen wir verschiedene Arten von Merkmalen und diskutieren ihre Vor- und Nachteile in Bezug auf die unterschiedlichen Arten von Daten und Modellen. Fangen wir also ohne Umschweife an!

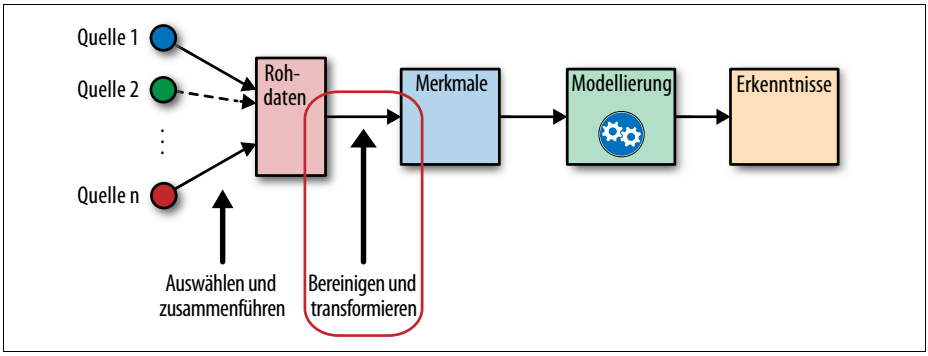

Abbildung 1-2: Der Platz der Merkmalskonstruktion im Arbeitsablauf beim maschinellen Lernen

KAPITEL 2

Tricksereien mit einfachen Zahlen

Bevor wir in die Welt komplexer Datentypen wie Text und Bilder eintauchen, wollen wir mit dem Einfachsten beginnen: mit numerischen Daten. Diese können aus vielfältigen Quellen stammen: geografische Orte eines Gebäudes oder einer Person, Einkaufspreise, Messungen eines Sensors, Verkehrszählungen usw. Numerische Daten liegen bereits in einer Form vor, die sich leicht in mathematische Modelle einspeisen lässt. Das macht die Merkmalskonstruktion jedoch keineswegs überflüssig: Gute Merkmale sollten nicht nur aussagekräftige Aspekte der Daten wiedergeben, sondern auch zu den Annahmen des Modells passen. Daher sind oftmals noch Transformationen notwendig. Numerische Verfahren der Merkmalskonstruktion sind etwas Grundlegendes; sie finden immer dann Anwendung, wenn Daten in numerische Merkmale umgeformt werden.

Die erste Frage bei einer Plausibilitätsprüfung für numerische Daten betrifft ihre Größe. Müssen wir lediglich wissen, ob sie positiv oder negativ sind? Oder interessiert uns vielleicht nur eine ganz grobe Vorstellung von ihrer Größenordnung? Diese Fragen sind besonders wichtig bei automatisiert gesammelten Daten wie Zählungen – den täglichen Besuchszahlen einer Website, der Anzahl von Kritiken für ein Restaurant usw.

Als Nächstes ist der Wertebereich der Merkmale von Bedeutung. Wie groß sind die größten und die kleinsten Werte? Umfassen sie mehrere Größenordnungen? Modelle, die aus glatten Funktionen der eingegebenen Merkmale bestehen, sind empfindlich für die Größe ihrer Eingangswerte. Beispielsweise ist $3x + 1$ eine einfache lineare Funktion der Eingangsgröße x, und der Wert ihrer Ausgabe hängt direkt vom Wert der Eingabe ab. Weitere Beispiele sind k-Means-Clustering, Nächste-Nachbarn-Methoden, radiale Basisfunktionen (RBF-Kerne) und alles, was mit dem euklidischen Abstand zu tun hat. Für diese Modelle und Modellierungskomponenten bietet es sich häufig an, die Merkmale zu *normieren*, sodass die Ausgaben in einer erwarteten Größenordnung liegen.

Logische Funktionen sind hingegen unempfindlich bezüglich der Größe von Merkmalswerten. Ihre Ausgabe ist für alle Arten von Eingangsgrößen stets binär. Beispielsweise nimmt das logische UND zwei beliebige Variablen und gibt genau dann 1 aus,

wenn beide Eingangswerte wahr sind. Ein anderes Beispiel einer logischen Funktion ist die Stufenfunktion (etwa die Entscheidung, ob der Eingangswert x größer als 5 ist). Entscheidungsbaummodelle bestehen aus Stufenfunktionen von Eingangsmerkmalen. Daher sind Modelle auf der Grundlage von Raumpartitionierungsbäumen (Entscheidungsbäume, gradientenverstärkte Maschinen, Random Forests) nicht wertebereichsempfindlich. Die einzige Ausnahme tritt auf, wenn der Wert der Eingangsgröße mit der Zeit wächst, was bei Merkmalen der Fall ist, die eine fortlaufende Zählung darstellen – irgendwann werden sie über den Bereich hinauswachsen, auf dem der Baum angelernt wurde. Wenn damit zu rechnen ist, kann es nötig werden, die Eingangswerte regelmäßig neu zu skalieren. Eine andere Lösung stellt die Methode der Klassenzählung aus Kapitel 5 dar.

Eine weitere wichtige Eigenschaft numerischer Merkmale ist ihre Verteilung. Die Verteilung fasst die Wahrscheinlichkeiten dafür zusammen, dass bestimmte Werte angenommen werden. Auf die Verteilung von Eingangsmerkmalen kommt es bei manchen Modellen mehr, bei anderen weniger an. Beispielsweise wird beim Anlernen eines linearen Regressionsmodells angenommen, dass Vorhersagefehler nach einer Gauß-Kurve (*http://mathworld.wolfram.com/NormalDistribution.html*) verteilt sind. Das ist meistens ein guter Ansatz, es sei denn, das Vorhersageziel umspannt mehrere Größenordnungen. In diesem Fall kann man wahrscheinlich nicht mehr von einer gaußschen Fehlerverteilung ausgehen. Ein möglicher Ausweg besteht darin, das Ausgabeziel zu transformieren, um das Ausmaß des Wachstums zu bändigen. (Streng genommen wäre das eine Zielkonstruktion, keine Merkmalskonstruktion.) Logarithmische Transformationen, die zu den *Potenztransformationen* gehören, bringen die Verteilung der Variablen näher an eine Gauß-Kurve.

Man kann Merkmale nicht nur auf die Annahmen des Modells oder des Anlernvorgangs hin zuschneiden, man kann auch mehrere von ihnen zu komplexeren Merkmalen zusammensetzen. Dabei hofft man, dass komplexe Merkmale wichtige Informationen in den Rohdaten prägnanter darstellen können. »Ausdrucksstärkere« Eingangsmerkmale erlauben einfachere Modelle, die leichter anzulernen und zu bewerten sind und bessere Vorhersagen treffen. Treibt man das auf die Spitze, können komplexe Merkmale selbst Ausgaben statistischer Modelle sein. Dieses als *Stapelung von Modellen* bekannte Konzept besprechen wir in den Kapiteln 7 und 8 viel ausführlicher. In diesem Kapitel stellen wir nur das einfachste Beispiel komplexer Merkmale vor: die *Kreuzmerkmale*.

Kreuzmerkmale sind einfach zu formulieren, aber die Kombination von Merkmalen führt dazu, dass viel mehr davon in das Modell eingegeben werden. Um den Rechenaufwand zu verringern, muss man üblicherweise die Eingangsmerkmale mithilfe automatischer *Merkmalsauswahl* ausdünnen.

Wir beginnen mit den Grundkonzepten der Skalare, Vektoren und Räume und besprechen danach Wertebereich, Verteilung, Kreuzmerkmale und Merkmalsauswahl.

Skalare, Vektoren und Räume

Bevor wir weitermachen, müssen wir zunächst einige Grundbegriffe definieren, auf denen der Rest des Buchs aufbaut. Ein einzelnes numerisches Merkmal heißt *Skalar*. Eine geordnete Liste von Skalaren wird *Vektor* genannt. Vektoren leben in einem *Vektorraum*. Bei den allermeisten Anwendungsfällen maschinellen Lernens werden die Eingangsdaten für ein Modell gewöhnlich als numerischer Vektor dargestellt. Der Rest dieses Buchs wird bewährte Strategien besprechen, um Rohdaten in Vektoren von Zahlen zu verwandeln.

Ein Vektor kann als Punkt im Raum veranschaulicht werden. (Gelegentlich wird eine Linie oder ein Pfeil vom Ursprung zu diesem Punkt gezeichnet. In diesem Buch werden wir zumeist nur den Punkt verwenden.) Nehmen wir beispielsweise an, wir hätten einen zweidimensionalen Vektor **v** = [1, –1]. Der Vektor enthält zwei Zahlen: In der ersten Richtung, d_1, hat der Vektor den Wert 1, und in der zweiten Richtung, d_2, hat er den Wert –1. Wir können **v** in einem 2-D-Diagramm darstellen (siehe Abbildung 2-1).

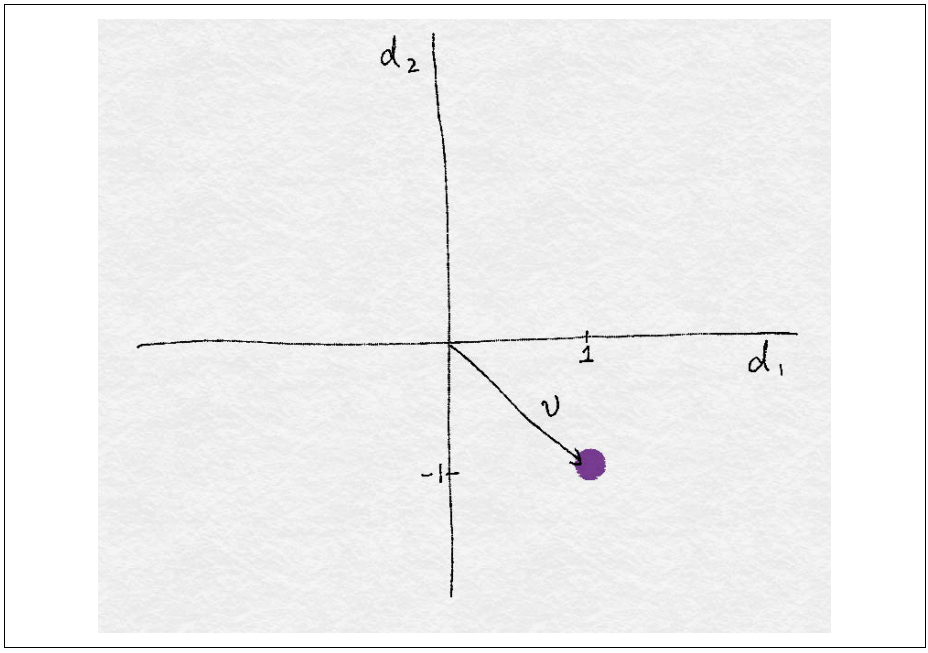

Abbildung 2-1: Ein einzelner Vektor

In der Welt der Daten haben ein abstrakter Vektor und seine Merkmalsdimensionen eine tatsächliche Bedeutung. Ein Vektor kann beispielsweise die Vorlieben einer Person für Musikstücke darstellen. Jedes Lied ist dabei ein Merkmal, wobei ein Wert 1 Gefallen bedeutet und ein Wert –1 Missfallen. Der Vektor **v** stellt beispielsweise die Vorlieben des Hörers Bob dar. Bob mag »Blowin' in the Wind« von

Bob Dylan und »Poker Face« von Lady Gaga. Andere Menschen haben andere Vorlieben. Zusammen genommen, kann eine Datensammlung im *Merkmalsraum* als Punktwolke veranschaulicht werden.

Umgekehrt kann ein Musiktitel durch die individuellen Vorlieben einer Gruppe von Personen repräsentiert werden. Angenommen, es gäbe nur zwei Hörer, Alice und Bob. Alice mag »Poker Face«, »Blowin' in the Wind« und »Hallelujah« von Leonard Cohen, nicht jedoch Katy Perrys »Roar« und Radioheads »Creep«. Bob mag »Roar«, »Hallelujah« und »Blowin' in the Wind«, kann aber »Poker Face« und »Creep« nicht ausstehen. Jedes Lied ist ein Punkt im Raum der Hörer. Ebenso, wie wir Daten im Merkmalsraum darstellen können, können wir Merkmale im *Datenraum* abbilden. Abbildung 2-2 zeigt das an diesem Beispiel.

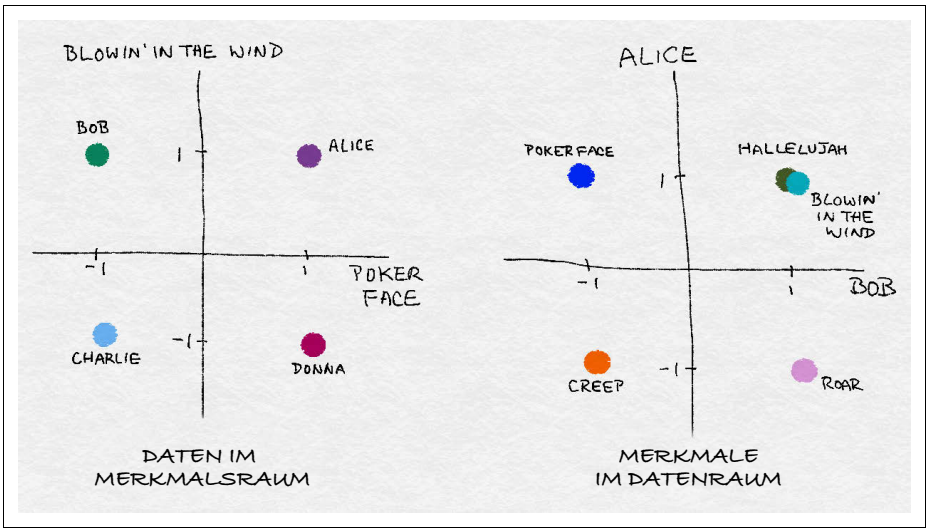

Abbildung 2-2: Veranschaulichung von Merkmalsraum und Datenraum

Der Umgang mit Zählern

Im Zeitalter von Big Data können Zähler rasch über alle Grenzen wachsen. Ein Nutzer kann einen Musiktitel oder einen Film in Endlosschleife abspielen oder ein Skript verwenden, um regelmäßig nachzuschauen, ob es Eintrittskarten für eine gefragte Vorstellung gibt, wodurch der Zähler fürs Abspielen bzw. der Besucherzähler der Website schnell steigt. Wenn Daten in großem Umfang oder hoher Geschwindigkeit erzeugt werden können, enthalten sie höchstwahrscheinlich ein paar extreme Werte. Es empfiehlt sich dann, den Wertebereich anzusehen und zu entscheiden, ob man die Daten als rohe Zahlen behält, in binäre Werte übersetzt, um ein Vorhandensein anzuzeigen, oder sie in gröbere Klassen einteilt. Schauen wir uns einige Beispiele an, um diese Konzepte zu veranschaulichen.