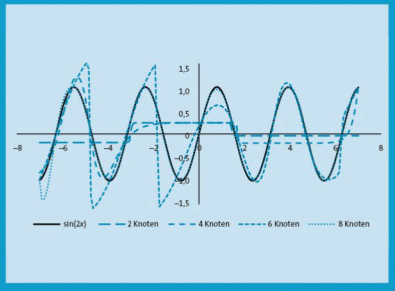
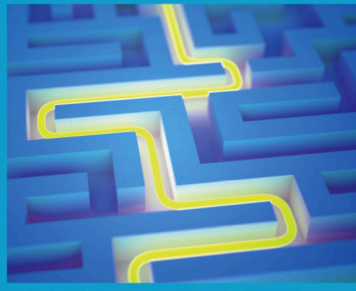
Paul Wilmott

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

Kapitel 6
Regressionsmethoden

Regressionsmethoden sind Methoden des überwachten Lernens. Sie versuchen, abhängige Variablen mittels unabhängiger Variablen zu beschreiben. Die unabhängigen Variablen sind numerisch, und wir passen Geraden, Polynome oder andere Funktionen an, um die abhängige(n) Variable(n) zu bestimmen. Natürlich kann man diese Methode auch für die Klassifikation einsetzen, wobei die abhängigen Variablen dann üblicherweise die Werte 0 oder 1 annehmen.

6.1 Wofür können wir sie verwenden?

Regressionsmethoden werden verwendet für:

- Finden numerischer Beziehungen zwischen abhängigen und unabhängigen Variablen, basierend auf Trainingsdaten
- Klassifizieren von Daten basierend auf einer Menge numerischer Merkmale.

Beispiele:

- Finde den mathematischen Zusammenhang zwischen der Anzahl an Tomaten auf einer Pflanze, der Umgebungstemperatur und der Bewässerungsmenge
- Bestimme die Wahrscheinlichkeit, an Krebs zu erkranken, basierend auf dem gewählten Lebensstil (Alkoholkonsum, Anzahl gerauchter Zigaretten, Anzahl der Amsterdambesuche, etc.).

Sicher kennen Sie die Regression vom Beispiel einer Geraden, die durch eine Punktwolke gelegt wird. Sie haben beispielsweise Werte und Flächen von Häusern, gibt es da einen linearen Zusammenhang zwischen den beiden Größen? Kaum ein Zusammenhang entspricht einer perfekten Linie, was ist also die *beste* Gerade, die durch die Punkte gelegt werden kann? Man kann sich auch in höhere Dimensionen wagen. Gibt es einen linearen Zusammenhang zwischen Wert, Fläche und der Anzahl der Garagenplätze?

Zur Einführung in diese Methode des überwachten Lernens starten wir mit der Suche nach linearen Zusammenhängen in mehreren Dimensionen und gehen dann weiter zur logistischen Regression, die sich sehr gut für Klassifikationsaufgaben eignet.

6.2 Mehrdimensionale lineare Regression

Sie haben bereits in Kapitel 2 eine kurze Zusammenfassung der eindimensionalen Regression gesehen. Der Übergang zu mehreren Dimensionen ist aus mathematischer Sicht recht einfach. Wir haben nun M unabhängige (oft auch *erklärend* genannte) Variablen, die für die Merkmale stehen, und so schreiben wir alle x zusammen als Vektor. Für jeden der N Datenpunkte gibt es eine unabhängige Variable $\mathbf{x}^{(n)}$ (Hausfläche, Anzahl Garagenplätze etc.) und die abhängige Variable $y^{(n)}$ (Immobilienwert). Wir legen nun eine lineare Funktion $h_\theta(\mathbf{x}) = \theta^\mathsf{T}\mathbf{x}$ durch die Punkte y, wobei θ der Vektor des noch unbekannten Parameters ist und $^\mathsf{T}$ für die Transponierte steht.

Noch eine kleine Verfeinerung: Da wir üblicherweise verhindern wollen, dass der Parameter θ_0 mit einer der unabhängigen Variablen multipliziert wird, schreiben wir $\mathbf{x}$ als $(1, x_1, \ldots, x_M)^\mathsf{T}$, und der Vektor hat somit die Dimension $M + 1$. Die Kostenfunktion bleibt die gleiche wie in Kapitel 2, d. h. die quadratische Funktion:

$$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} \left(h_\theta(\mathbf{x}^{(n)}) - y^{(n)} \right)^2$$

Wie in Kapitel 2 leiten wir die Kostenfunktion ab nach den einzelnen θs, setzen das Resultat gleich 0 und lösen die Gleichung für die θs. Kinderleicht. Obwohl wir rein mathematisch einen analytisch lösbaren Ausdruck für den Vektor θ erhalten, erfordert dies eine Matrixinversion. Sie können daher in der Praxis genauso gut ein Gradientenabstiegsverfahren zur Lösung verwenden.

Numerische Bestimmung der Parameter mittels Gradientenabstieg

Obwohl grundsätzlich keine numerischen Methoden für eindimensionale lineare Regressionsaufgaben notwendig sind, ist es das Werkzeug der Wahl, sobald es komplizierter wird. Batch-Gradientenabstieg und stochastischen Gradientenabstieg haben wir bereits in Kapitel 2 besprochen. Um diese Methoden zu nutzen, brauchen wir:

$$\frac{\partial J}{\partial \theta} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^{(n)} \left(h_\theta(\mathbf{x}^{(n)}) - y^{(n)} \right)$$

Schritt für Schritt: Der Batch-Gradientenabstiegsalgorithmus

1 ***Wiederhole***

Neu θ = Alt $\theta - \beta\ \partial J/\partial\theta$

D. h. Neu θ = Alt $\theta - \beta/N \sum_{n=1}^{N} \mathbf{x}^{(n)} \left(h_\theta(\mathbf{x}^{(n)}) - y^{(n)} \right)$

2 ***Aktualisiere***

Aktualisiere alle θ_k simultan. Wiederhole bis Konvergenz. ■

Lineare Regression ist so bekannt und weit verbreitet, dass ich die Beispiele hier auslasse und wir uns direkt zu etwas anderem bewegen, nämlich zur Regression für Klassifikationsaufgaben.

6.3 Logistische Regression

Nehmen wir an, Sie wollten E-Mails in Spam und Nicht-Spam einteilen. Die unabhängigen Variablen, jeweils x genannt, könnten Merkmale wie die Anzahl an Ausrufzeichen ! oder die Anzahl an Rechtschreibfehlern pro E-Mail, sein. Unsere y-Variablen nehmen dann entweder den Wert 0 für kein Spam oder den Wert 1 für Spam an. Lineare Regression wird in diesem Fall keine gute Wahl sein.

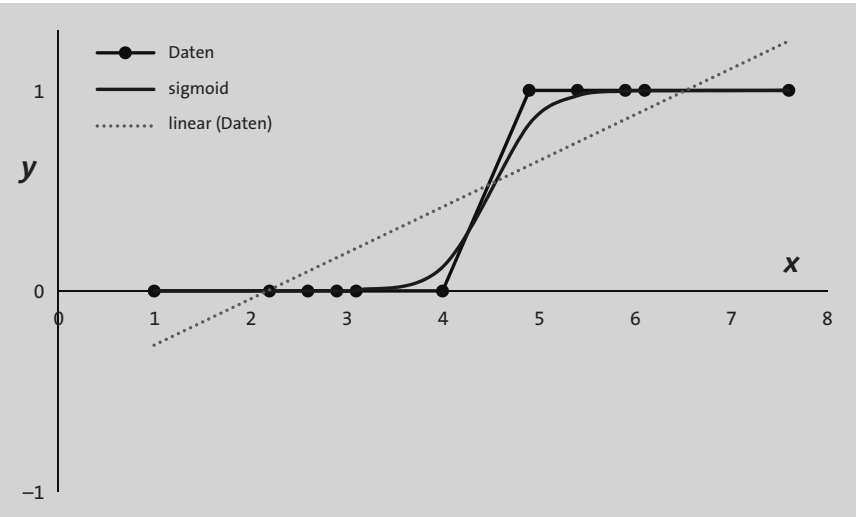


Abbildung 6.1 Regression für ein Klassifikationsproblem

Sehen Sie sich Abbildung 6.1 an: Würden Sie da versuchen wollen, eine Gerade durch diese Punkte zu legen? Das wäre purer Unsinn.

Manchmal repräsentiert die vertikale Achse eine Wahrscheinlichkeit, die Klassenzugehörigkeitswahrscheinlichkeit. In diesem Fall wären Zahlen unter 0 oder über 1 ebenfalls unerwünscht, was wir aber bei einer linearen Einpassung erhalten würden.

Langer Rede, kurzer Sinn – für Klassifikationsprobleme brauchen wir etwas Besseres als eine lineare Funktion. Wir tendieren da zu einer sigmoiden Funktion wie der logistischen Funktion

$$h_\theta(x) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x}}$$

oder, da wir sofort zur multiplen Regression übergehen,

$$h_\theta(\mathbf{x}) = \frac{1}{1 + e^{-\theta^\mathsf{T}\mathbf{x}}}.$$

Diese Funktion wird in der Abbildung 6.1 gezeigt. Wir passen also diese Funktion den gegebenen Daten an und könnten dann für einen neuen Datenpunkt (eine eingehende E-Mail) bestimmen, ob es sich um Spam handelt oder eben nicht, abhängig von einem Schwellenwert für h_θ. Bei einem Schwellenwert von 0,5 würde alles über diesem Wert in den Spamordner wandern. Wir könnten auch einen anderen Schwellenwert wählen.

Wenn wir Bedenken haben, dass echte E-Mails in den Spamordner wandern (und wir haben Freunde, die der Rächtschreibung nicht so mechtig sind und ständig Ausrufzeichen verwenden!!), dann können wir den Schwellenwert auf, sagen wir, 0,8 legen.

Die Kostenfunktion

Ein entscheidender Unterschied zwischen logistischer und linearer Regression ist die Wahl der Kostenfunktion.

Folgende Eigenschaften sind für eine solche Kostenfunktion offensichtlich zu fordern: Sie sollte positiv sein, es sei denn, wir haben eine perfekte Lösung, dann sollte sie 0 sein; sie sollte genau ein Minimum haben. Unsere bekannte quadratische Kostenfunktion erfüllt zwar die erste unser Bedingungen, aber wenn h_θ die logistische Funktion ist, hat sie nicht unbedingt ein einziges Minimum.

Die folgende Kostenfunktion hingegen erfüllt all unsere Wünsche:

$$J(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)} \ln\left(h_\theta(\mathbf{x}^{(n)})\right) + (1 - y^{(n)}) \ln\left(1 - h_\theta(\mathbf{x}^{(n)})\right)\right) \tag{13}$$

Warum also funktioniert es mit dieser Kostenfunktion? Wir wissen: y darf nur die Werte 0 oder 1 annehmen. Ist $y = 1$, dann nimmt die Funktion $J(\theta)$ kontinuierlich und monoton ab bis zum Wert 0 bei $h_\theta = 1$. Ist $y = 0$, dann liefert die Kostenfunktion ebenso 0, wenn $h_\theta = 0$.

Aber das Wichtigste ist, dass diese Kostenfunktion eine wunderbare Interpretation im Sinne der Maximum-Likelihood-Schätzung liefert, wie in Kapitel 2 diskutiert.

Hier haben wir keine analytische Lösung für das Minimum der Kostenfunktion, also müssen wir uns mit den Waffen der numerischen Mathematik rüsten. Eleganterweise ergibt sich für die neue Definition für h_θ und die neue Kostenfunktion dennoch folgende Gleichung:

$$\frac{\partial J}{\partial \theta} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}^{(n)} \left(h_\theta(\mathbf{x}^{(n)}) - y^{(n)} \right)$$

Daraus folgt, dass unser Gradientenabstiegsverfahren überraschenderweise unverändert bleibt.

6.4 Beispiel: Noch einmal politische Reden

Im vorherigen Kapitel bediente ich mich einiger Reden und Texte von Politikern, um das Wesen einer noch nicht gesehenen Rede zu bestimmen. Die Operation ist zwar missglückt, aber der Patient lebt, und es war dennoch eine faszinierende Übung, denke ich. Jetzt werde ich die gleichen Daten, die gleichen acht Reden und Texte, auf etwas andere Weise verwenden und wende dafür unsere Regressionsmethoden an.

Wieder nehme ich die Reden/Texte von $N = 8$ Politikern. Jeder Politiker wird mit einer 0 (linkes Lager) oder einer 1 (rechtes Lager) gekennzeichnet. Das sind die $y^{(n)}$ für $n = 1, \ldots, N$.

Anstatt aber auf die Häufigkeit der individuellen Wörter zu sehen, betrachten wir nun die *Worttypen*. Haben wir positive Wörter, negative Wörter, unregelmäßige Verben etc.? Insgesamt bezeichnet M wieder die Anzahl an Merkmalen. Der n^{te} Politiker wird repräsentiert durch $\mathbf{x}^{(n)}$, einem Vektor der Länge $M + 1$. Der erste Eintrag ist wieder 1. Der zweite Eintrag wäre dann der Anteil an positiven Wörtern, die der n^{te} Politiker verwendet, der dritte Eintrag der Anteil an negativen Wörtern usw.

Ja, aber wie weiß ich, ob ein Wort positiv, negativ etc. ist? Dafür verwende ich ein spezielles Wörterbuch, das für diese Art von Textanalysen gebräuchlich ist.

So ein Wörterbuch lässt sich online leicht finden, etwa das Loughran-McDonald-Wörterbuch. Es enthält eine Liste von Wörtern, klassifiziert nach verschiedenen Kategorien.

Diese Kategorien sind: negativ, positiv, Unsicherheit, streitbar, beschränkend, überflüssig, interessant, modal, unregelmäßiges Verb, Harvard IV, Silben. Diese Liste wird oft in Finanzberichten verwendet (daher die Kategorie streitbar). Die Bedeutung der meisten Kategorien ist klar. Modal beschreibt eine Graduierung von Wörtern, wie »immer« (stark modal, 1), »kann« (moderat, 2) bis »könnte« (weak, 3). Harvard IV ist ein psychosoziales Wörterbuch.

Das Resultat der Analyse der Reden finden Sie in Abbildung 6.2. Die Anteile an positiven, negativen etc. Wörtern ist ziemlich klein, da der größte Teil der Wörter im Wörterbuch eben nicht positiv, negativ, etc. ist.

	Benn	Churchill	Corbyn	JFK	M&E	May	Thatcher	Trump
negativ	0,501 %	1,048 %	1,216 %	0,562 %	2,067 %	0,395 %	1,626 %	1,261 %
positiv	0,213 %	0,380 %	0,517 %	0,243 %	0,760 %	0,137 %	1,018 %	1,048 %
Unsicherheit	0,395 %	0,441 %	0,274 %	0,137 %	0,289 %	0,091 %	0,425 %	0,289 %
streitbar	0,152 %	0,061 %	0,274 %	0,213 %	0,441 %	0,091 %	0,304 %	0,228 %
beschränkend	0,000 %	0,030 %	0,137 %	0,122 %	0,304 %	0,122 %	0,046 %	0,122 %
überflüssig	0,000 %	0,000 %	0,000 %	0,000 %	0,030 %	0,000 %	0,000 %	0,000 %
interessant	0,030 %	0,061 %	0,000 %	0,000 %	0,152 %	0,030 %	0,106 %	0,061 %
modal	1,307 %	1,975 %	1,945 %	0,881 %	1,276 %	0,380 %	2,097 %	2,112 %
unregelmäßig	0,228 %	0,745 %	0,608 %	0,334 %	0,988 %	0,289 %	0,942 %	0,699 %
Harvard IV	1,352 %	3,206 %	4,513 %	1,869 %	7,765 %	1,596 %	4,270 %	4,255 %
Silben	1,52	1,44	1,57	1,41	1,68	1,68	1,50	1,49

Abbildung 6.2 Die Analyseergebnisse: Häufigkeiten nach Kategorien

Da ich nur Reden von acht Politikern für das Training verwendete, habe ich auch nur einen Teil der elf Kategorien für die Regression eingesetzt. Hätte ich alle verwendet, wäre eine nicht sehr brauchbare Lösung dabei herausgekommen (zwölf Unbekannte und acht Gleichungen). Daher habe ich die Merkmale auf positiv, negativ, unregelmäßige Verben und Silben eingeschränkt.

Ich habe bemerkt, dass die θ-Werte für positive Wörter, unregelmäßige Verben und Anzahl Silben alle positiv, die θ-Werte für negative Wörter aber negativ waren. Die offensichtliche Interpretation überlasse ich Ihnen. Allerdings sollten Sie meine Vorbehalte bereits kennen!

Interessehalber wollte ich einmal einen meiner eigenen Texte klassifizieren. Also habe ich einen Ausschnitt dieses überaus bekannten politischen Textes verwendet: *Paul Wilmott On Quantitative Finance, Zweite Ausgabe*. Ergebnis: Ich bin, wie Margaret Thatcher, ein Rechter. Was immer man auch daraus lernen mag.

Aber nun ernsthaft, für eine realitätsnahe Analyse gäbe es einige offensichtliche Verbesserungen:

- mehr Trainingsdaten, d. h. viel, viel mehr Reden und Texte von einer viel größeren Menge an Politikern
- ein besseres Wörterbuch – eines, das besser für die Klassifikation von Politikerreden geeignet ist als für Reden über den Umsatzzuwachs von irgendwelchen Dingen

6.5 Weitere Regressionsmethoden

Zu Regressionsmethoden gibt es natürlich noch viel, viel mehr zu sagen. Nur kurz:

- **Die üblichen Verdächtigen**
  Es gibt eine Menge bekannter Basisfunktionen, die für Anpassung bzw. Approximation von Funktionen eingesetzt werden. Zweifellos haben Sie selber auch schon welche verwendet. Beispiele wären: Fourierreihen, Legendre-Polynome, Hermite-Polynome, radiale Basisfunktionen, Wavelets etc. Alle mögen ihre Verwendung für Regressionsaufgaben haben. Manche sind sehr angenehm und benutzerfreundlich, weil sie orthogonal sind (d. h., das Integral ihres Produkts über einer Domäne, eventuell auch mit Gewichtsfunktion, ist gleich null).

- **Polynomregression**
  Polynome sind eine offensichtliche Wahl. Allerdings gilt, je höher der Polynomgrad, desto größer die Gefahr der Überanpassung.

  Sie können die Methode der kleinsten Quadrate (KQM) verwenden, um die Parameter zu finden, dabei behandeln sie jeden nicht linearen Term so, als ob es eine weitere unabhängige Variable wäre. Eine eindimensionale Polynomapproximation wird so zu einer linearen Regression in mehreren Dimensionen. Das bedeutet, statt

  $$\theta_0 + \theta_1 x + \theta_2 x^2$$

  denken wir uns

  $$\theta_0 + \theta_1 x_1 + \theta_2 x_2.$$

  Durch die offensichtliche Korrelation von x und x^2 ist es nicht so einfach, die Bedeutung der Polynomkoeffizienten zu interpretieren.

- **Ridge-Regression**
  Ich habe in Kapitel 2 erwähnt, dass man manchmal einen Regularisierungsterm zur KQM hinzufügt:

  $$J(\theta) = \frac{1}{2N}\left(\sum_{n=1}^{N}\left(h_\theta(x^{(n)}) - y^{(n)}\right)^2 + \lambda|\bar{\theta}|^2\right)$$

  In ähnlicher Form macht man das auch zu anderen Kostenfunktionen. Mit $\bar{\theta}$ bezeichne ich den Vektor, dessen erster Eintrag 0 ist (d. h., es gibt kein θ_0). Und zum Vergleich mit unten, hier haben wir es mit der L^2 oder euklidischen Norm zu tun. Dieser zusätzliche Strafterm hat den Zweck der Reduktion der Anzahl der Koeffizienten.

  Warum in aller Welt sollten wir das tun? Im Allgemeinen verwendet man das im Fall von stark zusammenhängenden Merkmalen. Eine Regression mit Körpergröße und Alter kann problematisch werden, da diese Parameter sehr stark korrelieren. Eine

Optimierung ohne Strafterm wäre schwierig, da es im Fall einer perfekten Korrelation keine eindeutige Lösung gäbe. Regularisierungsmethoden vermeiden das und finden eine Balance zwischen den Koeffizienten korrelierter Merkmale.

- **LASSO-Regression**
  LASSO steht für Least Absolute Shrinkage and Selection Operator. Diese Methode ist der Regularisierung sehr ähnlich, allerdings ist der Strafterm hier die L^1-Norm, d. h. die Summe der Absolutwerte, statt der Summe der Quadrate.

  Die LASSO-Regression verkleinert nicht nur die Koeffizienten, sondern tendiert auch dazu, einige Koeffizienten ganz auf null zu setzen und so das Modell zu vereinfachen. Um letzteren Punkt zu überprüfen, zeichnen Sie $|\bar{\theta}| =$ konstant im θ-Raum (wir bleiben bei zwei Dimensionen!) und stellen Sie Vergleiche zwischen der Minimierung der Verlustfunktion mit Strafterm und der Minimierung der Verlustfunktion mit einer Bedingung an (siehe Abschnitt 2.12, »Lagrange-Multiplikatoren«).

6.6 Weiterführende Literatur

Applied Regression Analysis, dritte Ausgabe von Norman Draper und Harry Smith, veröffentlicht im Wiley Verlag, deckt alles zu linearer Regression ab. Es ist aber nicht billig!

Haben Sie die lineare Regression gemeistert, dann fahren Sie mit einem weiteren Wiley-Buch fort: *Nonlinear Regression Analysis and Its Applications* von Douglas Bates und Donald Watts.

GRUNDKURS

Machine Learning

Die mathematische Grundausbildung zum maschinellen Lernen

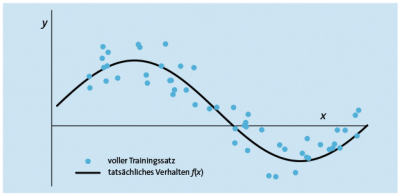
Maschinelles Lernen ist in aller Munde. Dieses Lehrbuch lenkt den Blick direkt auf den Kern der Sache – und das ist, aller Software-Frameworks zum Trotz, die Mathematik.

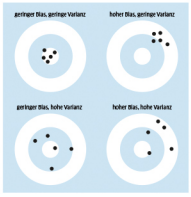
Jedes Lernverfahren wird Schritt für Schritt erklärt. Mit praktischen Tipps, vielen Zwischenschritten und einer Prise Humor. Die Mathematik ist für Studienanfänger nachvollziehbar.

Beim Trainieren der Modelle steckt der Teufel im Detail. Paul Wilmott zeigt, worauf es ankommt. Mit anschaulichen Beispielen von Natural Language Processing bis zum Abstimmungsverhalten im Parlament.

Die Lernverfahren:

+ k-nächste Nachbarn
+ k-Means-Clustering
+ Naiver Bayes-Klassifikator
+ Lineare und logistische Regression
+ Support-Vektor-Maschinen
+ Selbstorganisierende Karten
+ Entscheidungsbäume
+ Neuronale Netze
+ Reinforcement Learning





Paul Wilmott vermittelt angewandte Mathematik – mit Kultstatus! Seine unverwechselbaren Einführungen bringen seit Jahrzehnten Licht in finanzmathematische Modelle und KI.

Rheinwerk Computing

9 783836 275989

€ 29,90 [D]
€ 30,80 [A]

ISBN 978-3-8362-7598-9
Künstliche Intelligenz