Prof. Dr. Matthias Thimm

Modul 64401

Maschinelles Lernen

LESEPROBE

Fakultät für
Mathematik und Informatik

FernUniversität in Hagen

Der Inhalt dieses Dokumentes darf ohne vorherige schriftliche Erlaubnis durch die FernUniversität in Hagen nicht (ganz oder teilweise) reproduziert, benutzt oder veröffentlicht werden. Das Copyright gilt für alle Formen der Speicherung und Reproduktion, in denen die vorliegenden Informationen eingeflossen sind, einschließlich und zwar ohne Begrenzung Magnetspeicher, Computerausdrucke und visuelle Anzeigen. Alle in diesem Dokument genannten Gebrauchsnamen, Handelsnamen und Warenbezeichnungen sind zumeist eingetragene Warenzeichen und urheberrechtlich geschützt. Warenzeichen, Patente oder Copyrights gelten gleich ohne ausdrückliche Nennung. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einführung in Maschinelles Lernen

2.1 Lineare Regression

Version 1.9, 2023/04/18

Prof. Dr. Matthias Thimm
Artificial Intelligence Group, FernUniversität in Hagen

Lineare Regression ist die prinzipiell simpelste Form des maschinellen Lernens und wird üblicherweise auch eher als statistische Methode und nicht als Methode des maschinellen Lernens verstanden. Die Prinzipien der linearen Regression sind allerdings archetypisch für viele Methoden des maschinellen Lernens und es lassen sich viele komplexe Konzepte einfach am Beispiel der linearen Regression erläutern. Aus diesem Grund sind viele der in diesem Abschnitt eingeführten Konzepte für den gesamten Bereich des maschinellen Lernens relevant.

2.1.1 Grundlagen

In seiner einfachsten Form besteht das Problem der linearen Regression in der *optimalen* Anpassung einer Geraden an eine gegebene Menge von Punkten. Insbesondere beschäftigen wir uns hier mit der Anwendung der linearen Regression für das Problem der *Funktionsapproximation*, d. h., der Vorhersage des Funktionswerts einer Instanz, gegeben gewisser *Merkmalsausprägungen* der Instanz. Sei $n \in \mathbb{N}$ fix.

Definition 1. Ein *Datenpunkt* $x = (x_1, \ldots, x_n)^T$ ist ein Punkt $x \in \mathbb{R}^n$.

Wir nennen n die *Dimension* und jedes $i = 1, \ldots, n$ ein *Merkmal* (engl. *feature*). Ist $x \in \mathbb{R}^n$ ein Datenpunkt mit $x = (x_1, \ldots, x_n)^T$, so stellt jedes x_i, $i = 1, \ldots, n$, eine *Ausprägung* des Merkmals i dar. Sei $m \in \mathbb{N}$ fix. Ein Tupel (x, y) mit $y \in \mathbb{R}$ (der Wert der *Zielvariablen*) ist ein *Beispiel* und eine Menge $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ mit Beispielen $(x^{(i)}, y^{(i)})$, $i = 1, \ldots, m$, heißt *Datensatz*.

Beispiel 1. Stellen Sie sich vor, Sie wollen Ihrem oder Ihrer Liebsten einen Ring schenken und möchten, dass dies eine Überraschung wird. Weiterhin ergibt sich nicht die Möglichkeit den Umfang des Ringfingers der Person (heimlich) abzumessen, so dass Sie andere Wege finden müssen, um an diese Information zu kommen. Wie es der Zufall so will, verfügen Sie allerdings über Daten Ihrer besten Freunde, die den Umfang des Ringfingers mit der Körpergröße der jeweiligen Person in Beziehung setzt, siehe Tabelle 1. Abbildung 1 visualisiert die Daten entsprechend. Ignorieren wir die Bezeichner (Personennamen) der Daten (diese sind für die Lernaufgabe nicht von Relevanz) so lassen sich die Daten entsprechend als Datensatz D_{ring} via

$$D_{\text{ring}} = \{((153.3), 47.1), ((158.9), 46.8), ((160.8), 49.3), ((179.6), 53.2), ((156.6), 47.7), \\ ((165.1), 49.0), ((165.9), 50.6), ((156.7), 47.1), ((167.8), 51.7), ((160.8), 47.8)\}$$

repräsentieren. Hierbei ist beispielsweise $((153.3), 47.1)$ ein Beispiel mit Datenpunkt $(153.3) \in \mathbb{R}^1$, wobei 153.3 die Ausprägung des Merkmals „Körpergröße“ ist.

Beispiel 2. Um Ihren persönlichen Aufwand für den Kurs „Einführung in Maschinelles Lernen“ möglichst gering zu halten, haben Sie wieder Ihre Freunde (die den Kurs alle schon besucht haben) befragt, wie viele Stunden sie für den Kurs investiert haben, an wie vielen Sprechstunden sie teilgenommen haben und welche Note am Ende dabei herausgekommen ist, siehe Tabelle 2. Abbildung 2 visualisiert die Daten entsprechend. Ignorieren wir die Bezeichner (Personennamen) der Daten (diese sind für die Lernaufgabe nicht von Relevanz) so lassen sich die Daten entsprechend als Datensatz D_{note} via

$$D_{\text{note}} = \{((253, 3), 2.0), ((301, 6), 1.0), ((211, 1), 3.3), ((103, 3), 5.0), ((353, 4), 1.0), \\ ((250, 2), 2.3), ((98, 4), 4.0), ((150, 4), 3.7), ((63, 1), 5.0), ((282, 5), 1.3)\}$$

repräsentieren. Hierbei ist beispielsweise $((253, 3), 2.0)$ ein Beispiel mit Datenpunkt $(253, 3) \in \mathbb{R}^2$, wobei 253 die Ausprägung des Merkmals „Lernaufwand“ und 3 die Ausprägung des Merkmals „Anzahl Teilnahmen Sprechstunden“ ist.

Person	Körpergröße (in cm)	Ringfingerumfang (in mm)
Anna	153.3	47.1
Bernd	158.9	46.8
Catharina	160.8	49.3
David	179.6	53.2
Edith	156.6	47.7
Frank	165.1	49.0
Gertrud	165.9	50.6
Hans	156.7	47.1
Irma	167.8	51.7
Jochen	160.8	47.8

Tabelle 1: Datensatz zu Beispiel 1. Bitte beachten Sie, dass die Daten rein fiktiv sind.

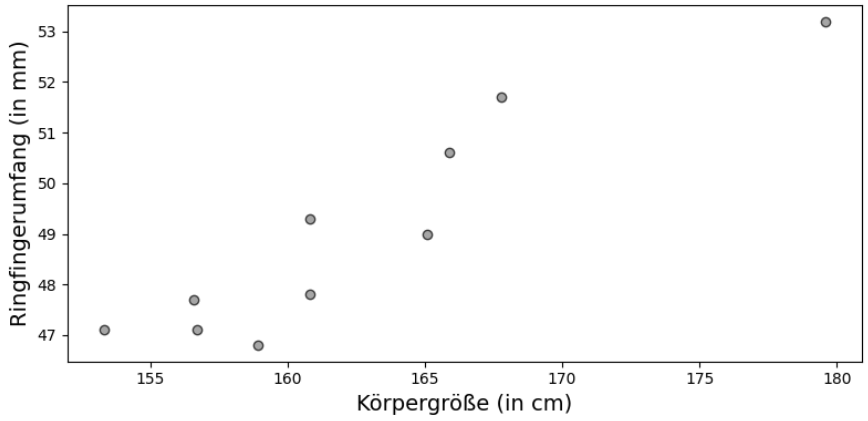

Abbildung 1: Datensatz zu Beispiel 1 (visualisiert).

Person	Lernaufwand (in Std.)	Anzahl Teilnahmen Sprechstunde	Note
Anna	253	3	2.0
Bernd	301	6	1.0
Catharina	211	1	3.3
David	103	3	5.0
Edith	353	4	1.0
Frank	250	2	2.3
Gertrud	98	4	4.0
Hans	150	4	3.7
Irma	63	1	5.0
Jochen	282	5	1.3

Tabelle 2: Datensatz zu Beispiel 2. Bitte beachten Sie, dass die Daten rein fiktiv sind.

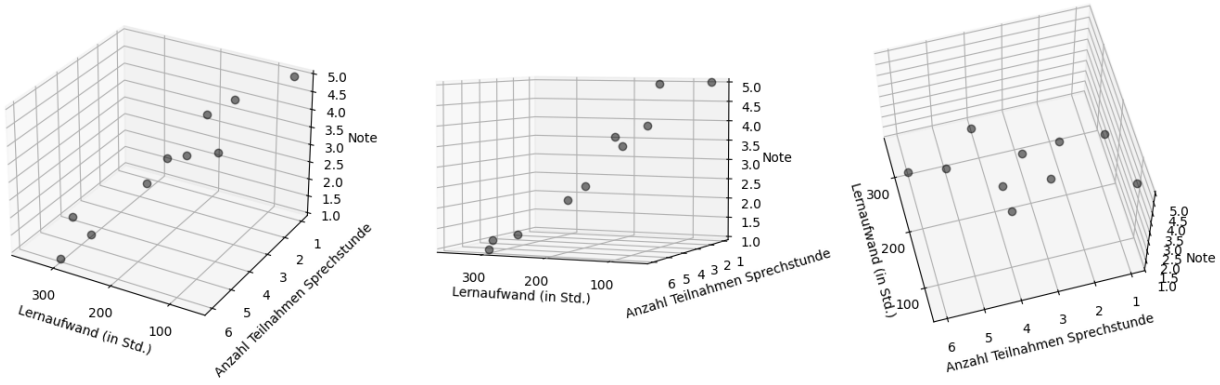

Abbildung 2: Datensatz zu Beispiel 2 (visualisiert).

Wir benutzen lineare Regression zur Funktionsapproximation, d. h., die Aufgabe des zu erlernenden Modells ist die Vorhersage des Funktionswertes $y \in \mathbb{R}$ zu einem beliebigen Datenpunkt $x \in \mathbb{R}^n$. Wir nehmen dazu an, dass die zu suchende Funktion *ähnlich* zu der Funktion ist, die einen gegebenen *Trainingsdatensatz* $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ generiert hat. Bei der linearen Regression nehmen wir zusätzlich an, dass der Zusammenhang zwischen x und y (die *Zielvariable*) *linear* ist.

Definition 2. Sei $\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T \in \mathbb{R}^{n+1}$. Eine *lineares Modell* auf $\mathbb{R}^n$ ist eine Funktion $h_\theta : \mathbb{R}^n \to \mathbb{R}$ mit

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

für alle $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$.

Die Werte $\theta = (\theta_0, \ldots, \theta_n)$ sind die *Parameter* des Modells h_θ. Der Parameter θ_0 stellt hierbei den konstanten Teil der Funktion h_θ dar.

Gegeben ein Trainingsdatensatz $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, ist es nun die Aufgabe, konkrete Werte für die Parameter θ zu finden, so dass $h_\theta(x^{(i)}) \approx y^{(i)}$, für alle $i = 1, \ldots, m$, ist. Dann gilt unter der Annahme, dass zukünftige Beispiele aus der gleichen Verteilung entstammen, auch, dass für einen neuen Datenpunkt $x \in \mathbb{R}^n$ der Wert $h_\theta(x)$ „nahe“ am tatsächlichen Funktionswert liegt.

Beispiel 3. Abbildung 3 visualisiert drei verschiedene mögliche lineare Modelle (unterschiedlicher Güte), die die Daten aus Beispiel 1 (siehe auch Tabelle 1 und Abbildung 1) erklären. Die dargestellten Funktionen sind

$$f_1(z) = 47 + \frac{1}{100} z$$
$$f_2(z) = -4 + \frac{1}{3} z$$
$$f_3(z) = 9 + \frac{1}{4} z$$

Betrachten wir uns f_1 etwas genauer. Die Funktion f_1 kann äquivalent repräsentiert werden durch die Parameter

$$\theta = (47, \frac{1}{100})$$

eines linearen Modells h_θ. Gegeben ein Datenpunkt $x = (x_1)$ ist

$$h_\theta(x) = \theta_0 + \theta_1 x_1$$

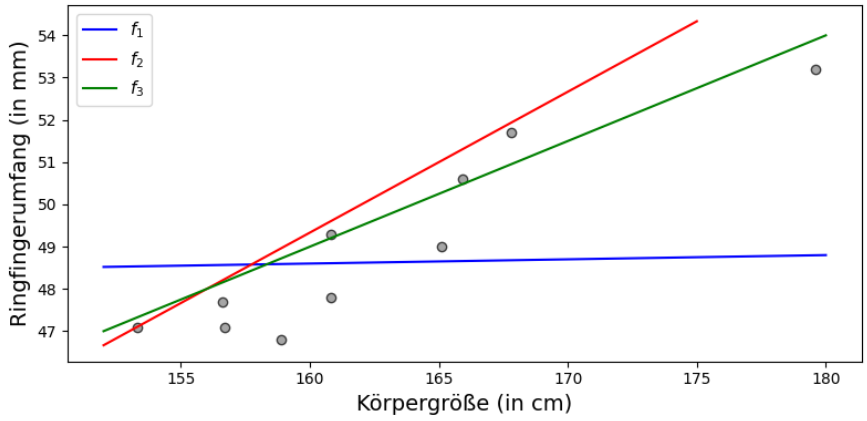

Abbildung 3: Illustration dreier möglicher linearer Modelle zu den Daten aus Beispiel 1.

Beispiel 4. Abbildung 4 visualisiert drei verschiedene mögliche lineare Modelle (unterschiedlicher Güte), die die Daten aus Beispiel 2 (siehe auch Tabelle 2 und Abbildung 2) erklären. Die dargestellten Funktionen sind

$$g_1(y,z) = 6 - \frac{1}{300} * y - \frac{1}{6} * z$$
$$g_2(y,z) = 4 - \frac{1}{200} * y - \frac{1}{6} * z$$
$$g_3(y,z) = 5 - \frac{1}{100} * y - \frac{1}{10} * z$$

Betrachten wir uns g_1 etwas genauer. Die Funktion g_1 kann äquivalent repräsentiert werden durch die Parameter

$$\theta = (6, -\frac{1}{300}, -\frac{1}{6})$$

eines linearen Modells h_θ. Gegeben ein Datenpunkt $x = (x_1, x_2)$ ist

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Das Finden der optimalen Parameter θ, so dass h_θ möglichst gut an die Beispiele in D angepasst ist, kann als Optimierungsproblem modelliert werden. Dazu wählt man zunächst ein geeignetes *Abstandsmaß* (oder *Fehlermaß* oder *Kostenfunktion*), das bewertet, wie nah eine Funktion an die Beispiele D angepasst ist. Das üblichste Fehlermaß im maschinellen Lernen ist der *quadratische Fehler*. Um die Notation zu vereinfachen, repräsentieren wir $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ durch eine Matrix X_D und einen Vektor y_D wie folgt

$$X_D = \begin{bmatrix} 1 & x_1^{(1)} & \ldots & x_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \ldots & x_n^{(m)} \end{bmatrix} \qquad y_D = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Mit anderen Worten, $X_D \in \mathbb{R}^{(n+1)\times m}$ enthält als Zeilenvektoren alle Datenpunkte der Beispiele aus D und eine zusätzlich vorangestellte 1. Letztere dient der einfacheren Darstellung als Vektorprodukt und wird auch *Bias* genannt. Da die Zeilen von X_D später mit $\theta \in \mathbb{R}^{n+1}$ multipliziert werden sollen, stimmen hier nun die Dimensionen und der konstante Teil der Funktion h_θ wird hier einfach mit 1 multipliziert. Der Vektor $y_D \in \mathbb{R}^m$ enthält die zugehörigen Funktionswerte.

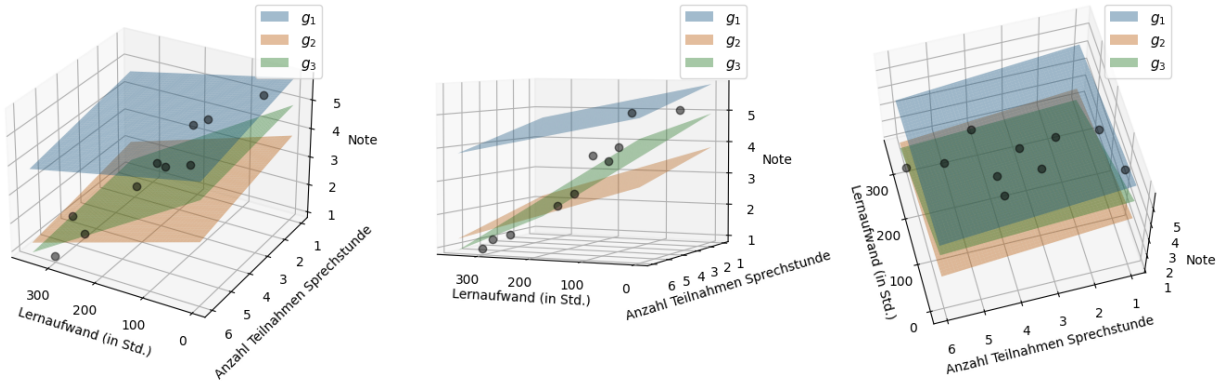

Abbildung 4: Illustration dreier möglicher linearer Modelle zu den Daten aus Beispiel 2.

Definition 3. Sei D ein Datensatz und $f : \mathbb{R}^n \to \mathbb{R}$ eine beliebige Funktion. Der *quadratische Fehler* L von f bzgl. D ist definiert durch

$$L(D, f) = \sum_{i=1}^{m} (f(x^{(i)}) - y^{(i)})^2$$

Ist $f = h_\theta$ eine lineare Funktion h_θ mit Parametern θ, so ist dies äquivalent zu

$$L(D, \theta) = \|X_D\theta - y_D\|^2$$

wobei $\|\cdot\|$ die *Euklidische Norm* ist.

Beispiel 5. Wir setzen Beispiel 3 fort und wiederholen noch einmal die drei genannten linearen Modelle f_1, f_2 und f_3:

$$f_1(z) = 47 + \frac{1}{100}z$$
$$f_2(z) = -4 + \frac{1}{3}z$$
$$f_3(z) = 9 + \frac{1}{4}z$$

Wenden wir den quadratischen Fehler auf diese Funktionen bzgl. des Datensatzes D_{ring} (siehe Tabelle 1) an, erhalten wir

$$L(D_{\text{ring}}, f_1) \approx 41.705$$
$$L(D_{\text{ring}}, f_2) \approx 21.349$$
$$L(D_{\text{ring}}, f_3) \approx 9.778$$

Wie man sieht erhält die optisch am besten angepasste Funktion f_3 den niedrigsten quadratischen Fehler.

Beispiel 6. Wir setzen Beispiel 4 fort und wiederholen noch einmal die drei genannten linearen Modelle g_1, g_2 und g_3:

$$g_1(y, z) = 6 - \frac{1}{300} * y - \frac{1}{6} * z$$
$$g_2(y, z) = 4 - \frac{1}{200} * y - \frac{1}{6} * z$$
$$g_3(y, z) = 5 - \frac{1}{100} * y - \frac{1}{10} * z$$

Wenden wir den quadratischen Fehler auf diese Funktionen bzgl. des Datensatzes D_{note} (siehe Tabelle 2) an, erhalten wir

$$L(D_{\text{note}}, g_1) \approx 47.0454$$
$$L(D_{\text{note}}, g_2) \approx 9.958$$
$$L(D_{\text{note}}, g_3) \approx 3.397$$

Auch hier sieht man, dass die optisch am besten angepasste Funktion g_3 den niedrigsten quadratischen Fehler erhält.

Damit h_θ die Beispiele in D bestmöglich approximiert, suchen wir Parameter θ, die den quadratischen Fehler L bzgl. D *minimieren*, d h., wir suchen eine Lösung für das folgende Optimierungsproblem:

$$\min_\theta L(D,\theta) = \min_\theta \|X_D\theta - y_D\|^2 \tag{1}$$

Für lineare Regression ist das obige Optimierungsproblem stets eindeutig lösbar, d. h., ein lokales Minimum ist stets das globale Minimum. Methoden wie *Gradient Descent* können hier also das obige Problem beliebig genau lösen. Bei großen Trainingsdatensätzen und/oder vielen Merkmalen ist dies auch die praktikabelste Methode. Allerdings kann das Minimum in (1) durch einfache Methoden der Differentialrechnung auch in geschlossener Form dargestellt und somit direkt berechnet werden.[1] Da das lokale/globale Minimum von $L(D,\theta)$ eindeutig bestimmt ist, kann dieses durch die Nullstelle des Gradienten bzgl. θ von $L(D,\theta)$ charakterisiert werden, d. h., θ ist die optimale Parameterkombination gdw. $\nabla_\theta L(D,\theta) = 0$ ist. Es folgt:

$$\begin{aligned}
& \nabla_\theta L(D,\theta) = 0 \\
\Longrightarrow \quad & \nabla_\theta \|X_D\theta - y_D\|^2 = 0 \\
\Longrightarrow \quad & \nabla_\theta (X_D\theta - y_D)^T (X_D\theta - y_D) = 0 \\
\Longrightarrow \quad & \nabla_\theta (\theta^T X_D^T X_D\theta - 2\theta^T X_D^T y_D + y_D^T y_D) = 0 \\
\Longrightarrow \quad & 2X_D^T X_D\theta - 2X_D^T y_D = 0 \\
\Longrightarrow \quad & (X_D^T X_D)^{-1} X_D^T y_D = \theta
\end{aligned}$$

Da X_D und y_D gegeben ist, kann θ mit obiger Gleichung direkt berechnet werden. Es sollte aber beachtet werden, dass diese Methode die rechenaufwändige Invertierung der Matrix $X_D^T X_D$ beinhaltet. Gerade bei großen Werten für m und n ist die Verwendung von numerischen Optimierungsmethoden (wie *Gradient Descent*) zu bevorzugen.

Beispiel 7. Wir setzen Beispiel 5 fort. Das optimale lineare Modell in diesem Beispiel ist gegeben durch

$$f(z) \approx 5.362 + 0.269 * x$$

Die Funktion f ist visualisiert in Abbildung 5 und verfügt mit

$$L(D_{\text{ring}}, f) \approx 5.91$$

über einen geringeren quadratischen Fehler als die Funktionen aus Beispiel 3.

Beispiel 8. Wir setzen Beispiel 6 fort. Das optimale lineare Modell in diesem Beispiel ist gegeben durch

$$g(y,z) \approx 6.321 - 0.0141 * y - 0.166 * z$$

Die Funktion g ist visualisiert in Abbildung 6 und verfügt mit

$$L(D_{\text{note}}, g) \approx 0.83$$

über einen geringeren quadratischen Fehler als die Funktionen aus Beispiel 4.

[1] Beachten Sie bitte, dass im Folgenden und auch in späteren Abschnitten, einige mathematische Details abstrahiert werden und nicht immer alle notwendigen Vorbedingungen zur Anwendung mathematischer Gleichheiten explizit überprüft werden. Für weitere Details verweisen wir auf weiterführende Literatur.

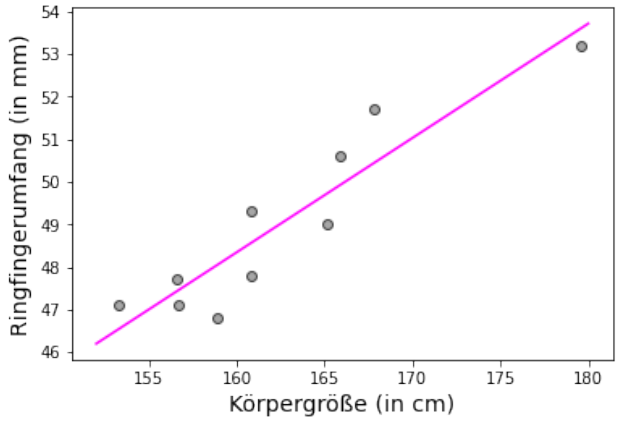

Abbildung 5: Optimal angepasstes lineares Modell für den Datensatz D_{ring}.

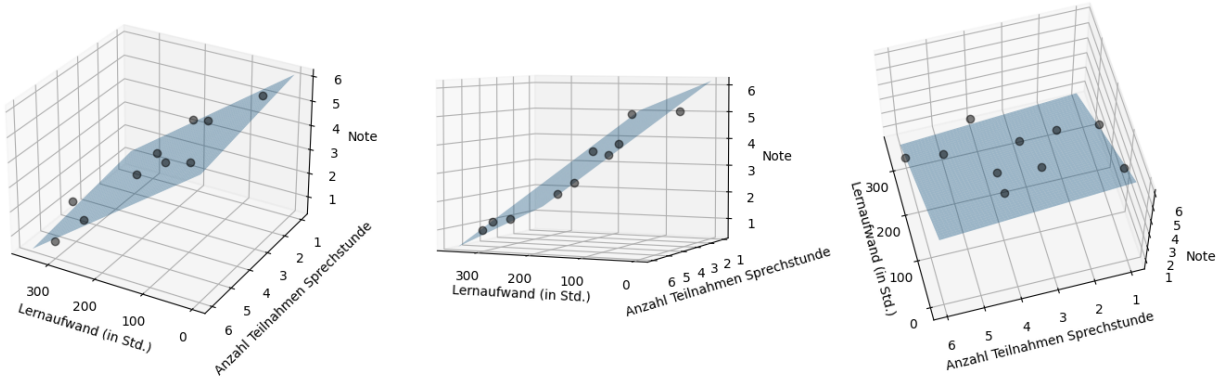

Abbildung 6: Optimal angepasstes lineares Modell für den Datensatz D_{note}.

2.1.2 Evaluation

Lineare Regression ist nur eine Methode von vielen im Bereich des maschinellen Lernens. Eine wichtiger Punkt bei der Entwicklung und Anwendung von Methoden des maschinellen Lernens ist daher die *Evaluation*, d. h., die Analyse und Feststellung, wie gut eine bestimmte Methode ein konkretes Problem des maschinellen Lernens löst. Dies hilft insbesondere dabei, die für das konkrete Problem beste Methode auszuwählen.

Um eine Methode des maschinellen Lernens zu evaluieren, teilt man üblicherweise die für das Lernen zur Verfügung stehenden Daten in einen *Trainingsdatensatz* (wie zuvor schon genutzt) und einen *Testdatensatz* auf. Anschließend trainiert man die Methode ausschließlich auf dem Trainingsdatensatz und benutzt den Testdatensatz, um zu evaluieren wie gut das gelernte Modell auf zuvor ungesehenen Daten generalisiert. Üblicherweise geschieht diese Aufteilung in Trainings- und Testdatensatz *zufällig* und so, dass der Trainingsdatensatz ungefähr 75%-90% des Gesamtdatensatzes ausmacht. Als Evaluationsmaß benutzen wir eine normalisierte Variante des quadratischen Fehlers: das *Bestimmtheitsmaß*.

Definition 4. Sei $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ ein Datensatz und $f : \mathbb{R}^n \to \mathbb{R}$ eine beliebige Funktion. Das *Be-*

stimmtheitsmaß R^2 von f bzgl. D ist definiert durch

$$R^2(D,f) = \left(1 - \frac{L(D,f)}{\sum_{i=1}^m (y^{(i)} - \tilde{y})^2}\right) = \left(1 - \frac{\sum_{i=1}^m (f(x^{(i)}) - y^{(i)})^2}{\sum_{i=1}^m (y^{(i)} - \tilde{y})^2}\right)$$

wobei $\tilde{y}$ der Mittelwert der $y^{(i)}$ ist:

$$\tilde{y} = \frac{1}{m}\sum_{i=1}^m y^{(i)}$$

Der Wert $R^2(D,f)$ kann maximal 1 betragen, dies entspricht dem Fall, daß f den Datensatz D perfekt modelliert ($f(x^{(i)}) = y^{(i)}$ für alle $i = 1,\ldots,m$). Je kleiner der Wert $R^2(D,f)$, desto schlechter die Vorhersagequalität von f bezüglich D. Für ein naives Modell f_{naive}, dass stets den Mittelwert $\tilde{y}$ vorhersagen würde, hätten wir $R^2(D, f_{\text{naive}}) = 0$.[2] Modelle f mit $R^2(D,f) < 0$ sind somit noch schlechter als dieses naive Modell.

Gegeben einen Datensatz D und einer Aufteilung von D in einen Trainingsdatensatz D^{train} und einen Testdatensatz D^{test}, so erwarten wir, dass $R^2(D^{\text{train}}, f)$ für das auf D^{train} gelernte Modell f relativ nahe an 1 liegt (da f durch Minimierung des quadratischen Fehlers gelernt wurde und R^2 prinzipiell nur eine skalierte Variante des quadratischen Fehlers ist). $R^2(D^{\text{test}}, f)$ ist üblicherweise kleiner als $R^2(D^{\text{train}}, f)$. Je näher $R^2(D^{\text{test}}, f)$ allerdings an $R^2(D^{\text{train}}, f)$ liegt, desto besser ist die Fähigkeit von f, auf ungesehene Daten zu generalisieren.

Beispiel 9. Wir setzen Beispiel 7 fort. Tabelle 3 enthält noch einmal den Datensatz D_{ring}, diesmal mit Indizes anstatt Personennamen. Wir definieren Trainingsdatensatz $D^{\text{train1}}_{\text{ring}}$ und den Testdatensatz $D^{\text{test1}}_{\text{ring}}$ via

$$D^{\text{train1}}_{\text{ring}} = \{(x^{(1)},y^{(1)}),(x^{(3)},y^{(3)}),(x^{(4)},y^{(4)}),(x^{(5)},y^{(5)}),(x^{(6)},y^{(6)}),(x^{(7)},y^{(7)}),(x^{(8)},y^{(8)}),(x^{(10)},y^{(10)})\}$$
$$D^{\text{test1}}_{\text{ring}} = \{(x^{(2)},y^{(2)}),(x^{(9)},y^{(9)})\}$$

Mit anderen Worten, $D^{\text{test1}}_{\text{ring}}$ enthält die 2. und 9. Zeile aus Tabelle 3 und $D^{\text{train1}}_{\text{ring}}$ enthält alle übrigen Zeilen. Lernen wir ein lineares Modell ausschließlich auf $D^{\text{train1}}_{\text{ring}}$, so erhalten wir das in Abbildung 7 dargestellte Modell f_1 (hier sind die Trainingsdaten auch grau dargestellt, wohingegen die Testdaten rot dargestellt sind). Beachten Sie, dass sich dieses Modell f_1 von dem Modell f aus Beispiel 7 geringfügig unterscheidet, da es nicht auf den exakt selben Daten trainiert wurde. Wollen wir nun evaluieren, wie gut dieses Modell generalisiert, können wir es auf die Testdaten anwenden und vergleichen dazu die Vorhersagegenauigkeit des Modells f_1 mit den vorhandenen Werten. Dazu nutzen wir das Bestimmtheitsmaß als Messgröße, d. h., wir berechnen

$$R^2(D^{\text{test1}}_{\text{ring}}, f_1) \approx 0.691$$

Der Wert 0.691 ist durchaus hoch, insbesondere bei dem recht kleinen Trainingsdatensatz, den wir genutzt haben. Vergleichen wir diesen Wert nun mit dem entsprechenden Wert auf den Trainingsdaten

$$R^2(D^{\text{train1}}_{\text{ring}}, f_1) \approx 0.919$$

so sehen wir, dass das Modell f_1 weitaus besser auf dem Trainingsdatensatz abschneidet (auf dem es ja auch gelernt wurde). Der Abstand $R^2(D^{\text{test1}}_{\text{ring}}, f_1)$ zu $R^2(D^{\text{train1}}_{\text{ring}}, f_1)$ ist relativ groß und man könnte vermuten, dass das Modell f_1 zu stark an die Trainingsdaten angepasst ist und eher schlecht generalisiert (siehe das Problem der *Überanpassung* im nächsten Abschnitt). Vermutlich haben wir hier aber nur Pech mit der Aufteilung in Trainings- und Testdaten gehabt. Wir probieren eine andere Aufteilung:

$$D^{\textit{train2}}_{\text{ring}} = \{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),(x^{(3)},y^{(3)}),(x^{(4)},y^{(4)}),(x^{(5)},y^{(5)}),(x^{(6)},y^{(6)}),(x^{(7)},y^{(7)}),(x^{(10)},y^{(10)})\}$$
$$D^{\textit{test1}}_{\text{ring}} = \{(x^{(8)},y^{(8)}),(x^{(9)},y^{(9)})\}$$

Mit anderen Worten, $D^{\text{test2}}_{\text{ring}}$ enthält die 8. und 9. Zeile aus Tabelle 3 und $D^{\text{train2}}_{\text{ring}}$ enthält alle übrigen Zeilen. Lernen wir ein lineares Modell ausschließlich auf $D^{\text{train2}}_{\text{ring}}$, so erhalten wir das in Abbildung 8 dargestellte Modell f_2 (wieder sind die Trainingsdaten grau und die Testdaten rot dargestellt). Wir berechnen das Bestimmtheitsmaß auf $D^{\text{train2}}_{\text{ring}}$ und $D^{\text{test2}}_{\text{ring}}$:

[2] Im Szenario der Vorhersage der Ringgröße hätten wir beispielsweise $f_{\text{naive}}(x) = 49$, also ein Modell, dass konstant den Mittelwert aller beobachteten Ringgrößen (49) vorhersagt.

Nr.	Körpergröße (in cm)	Ringfingerumfang (in mm)
1	153.3	47.1
2	158.9	46.8
3	160.8	49.3
4	179.6	53.2
5	156.6	47.7
6	165.1	49.0
7	165.9	50.6
8	156.7	47.1
9	167.8	51.7
10	160.8	47.8

Tabelle 3: Datensatz zu Beispiel 7, reproduziert von Tabelle 1.

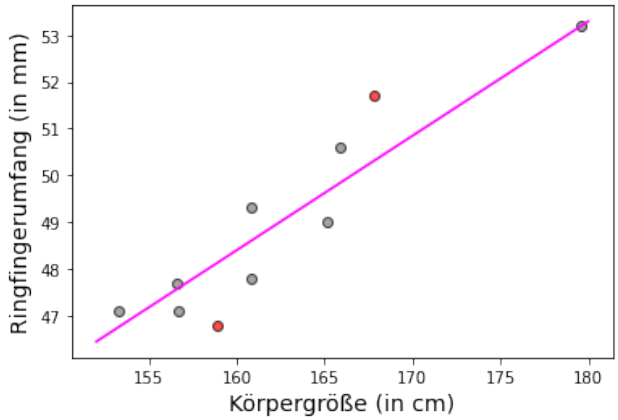

Abbildung 7: Optimal angepasstes lineares Modell für den Datensatz $D_{\text{ring}}^{\text{train1}}$.

$$R^2(D_{\text{ring}}^{train2}, f_2) \approx 0.877$$
$$R^2(D_{\text{ring}}^{test2}, f_2) \approx 0.783$$

Hier sind beide Werte recht hoch und auch recht nah beieinander.

Um das im vorherigen Beispiel auftretende Problem der variierenden Abstände zwischen Bestimmtheit der Trainings- und Testdaten zu adressieren, benutzt man oft die sogenannte *Kreuzvalidierung* (engl. *cross validation*). Gegeben ein Datensatz D und k eine natürliche Zahl (übliche Werte sind 5 oder 10), so teilen wir zunächst D in k ungefähr gleich große Teildatensätze $D_1, \ldots, D_k$ auf. Anschliessend führen wir eine Evaluation auf k verschiedenen Paaren von Trainings- und Testdaten $(D_1^{\text{train}}, D_1^{\text{test}}), \ldots, (D_k^{\text{train}}, D_k^{\text{test}})$ aus, die wie folgt definiert sind:

$$\begin{aligned}
D_1^{\text{train}} &= D_1 \cup \ldots \cup D_{k-1} & D_1^{\text{test}} &= D_k \\
D_2^{\text{train}} &= D_1 \cup \ldots \cup D_{k-2} \cup D_k & D_2^{\text{test}} &= D_{k-1} \\
D_3^{\text{train}} &= D_1 \cup \ldots \cup D_{k-3} \cup D_{k-1} \cup D_k & D_3^{\text{test}} &= D_{k-2} \\
D_4^{\text{train}} &= D_1 \cup \ldots \cup D_{k-4} \cup D_{k-2} \cup \ldots \cup D_k & D_4^{\text{test}} &= D_{k-3} \\
&\vdots & & \\
D_k^{\text{train}} &= D_2 \cup \ldots \cup D_k & D_k^{\text{test}} &= D_1
\end{aligned}$$

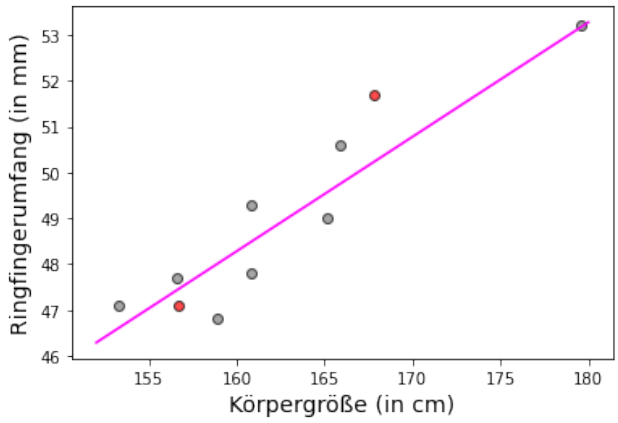

Abbildung 8: Optimal angepasstes lineares Modell für den Datensatz $D_{\text{ring}}^{\text{train2}}$.

Nr.	Fläche (in m^2)	Preis (in EUR)
1	100	210000
2	120	270000
3	160	290000
4	170	470000
5	200	620000
6	210	680000
7	310	1600000
8	330	1900000
9	370	2570000
10	400	3300000

Tabelle 4: Datensatz D_{houses} zu Beispiel 10. Die Zahlen sind rein fiktiv.

Die verschiedenen Paare von Trainings- und Testdaten erhalten wir also, indem wir einen Teildatensatz D_i als Testdatensatz und den Rest als Trainingsdatensatz deklarieren. Auf jedem Paar können dann die entsprechenden R^2-Werte berechnet werden und als finale Evaluationsmetrik berechnet man den Durchschnitt von $R^2(D_1^{\text{test}}, \cdot), \ldots, R^2(D_k^{\text{test}}, \cdot)$.

2.1.3 Nichtlineare Modelle

Lineare Regression ist eine mächtige Methode des maschinellen Lernens, aber fundamental durch die Linearitätsannahme in ihrer Anwendbarkeit eingeschränkt. In vielen realen Anwendungsfällen stehen Merkmale und Zielvariable nicht notwendigerweise in einem linearen Zusammenhang.

Beispiel 10. Betrachten Sie den in Tabelle 4 dargestellten und in Abbildung 9 visualisierten Datensatz D_{houses}, der die Größe eines Hauses zu seinem Kaufpreis in Relation setzt (die Daten sind rein fiktiv). Die Abbildung visualisiert auch das lineare Modell f, das optimal an die Daten angepasst ist (hier wurde der gesamte Datensatz D_{houses} als Trainingsdatensatz benutzt). Wie man sieht, ist eine Anpassung an die Daten nur sehr schlecht möglich, da diese offensichtlich nicht in einem linearen Zusammenhang zueinander stehen.

Die Einbeziehung nichtlinearer Zusammenhänge zwischen Merkmalen und der Zielvariablen kann bei der linearen Regression realisiert werden, indem in einem Vorbereitungsschritt die Beispiele um zusätzliche (nichtlineare) Merkmale ergänzt werden, die aus den schon existierenden Merkmalen berechnet werden.

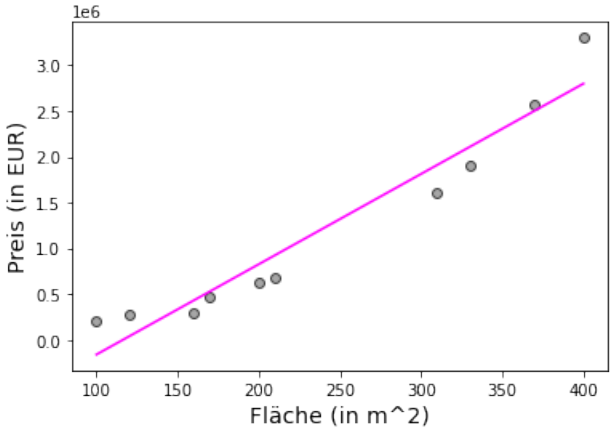

Abbildung 9: Optimal angepasstes lineares Modell für den Datensatz D_{houses}.

Nr.	Fläche (in m^2)	Fläche^2	Preis (in EUR)
1	100	10000	210000
2	120	14400	270000
3	160	25600	290000
4	170	28900	470000
5	200	40000	620000
6	210	44100	680000
7	310	96100	1600000
8	330	108900	1900000
9	370	136900	2570000
10	400	160000	3300000

Tabelle 5: Erweiterter Datensatz D'_{houses} zu Beispiel 11.

Beispiel 11. Eine visuelle Inspektion der Datenpunkte in Abbildung 9 legt nahe, dass ein quadratischer Zusammenhang zwischen der Fläche des Hauses und dem Kaufpreis besteht. Wir erweiterten aus diesem Grund den Datensatz D_{houses} um ein weiteres Merkmal, dass das Quadrat des Merkmals „Fläche“ ist, d. h. bei jedes Beispiel des Datensatzes mit Merkmalsausprägung x des Merkmals „Fläche“ erhält ein weiteres Merkmal „Fläche^2“ mit Ausprägung x^2. Der erweiterte Datensatz D'_{houses} ist in Tabelle 5 dargestellt. Das neue Merkmal „Fläche^2“ hat keine intuitive Bedeutung und deshalb haben wir diesem Merkmal in Tabelle 5 auch keine Einheit zugewiesen.

Die eigentliche formale Maschinerie der linearen Regression kann nun auf einen erweiterten Datensatz genauso angewendet werden wie auf dem originalen Datensatz.

Beispiel 12. Wir lösen das Optimierungsproblem aus Gleichung (1) bezüglich D'_{houses} und erhalten Parameter θ, die folgender linearen Funktion f_{houses} entsprechen:

$$f_{\text{houses}}(x_1, x_2) \approx -7297.676x_1 + 34.189x_2 + 647307.78$$

wobei x_1 eine Ausprägung des Merkmals „Fläche“ und x_2 eine Ausprägung des Merkmals „Fläche^2“ ist. Da $x_2 = x_1^2$ für jedes Beispiel ist, können wir f_{houses} auch vereinfacht darstellen durch

$$f_{\text{houses}}(x_1) \approx -7297.676x_1 + 34.189x_1^2 + 647307.78$$

und somit ist f auch keine lineare Funktion mehr, sondern ein Polynom. Abbildung 10 visualisiert f_{houses}.

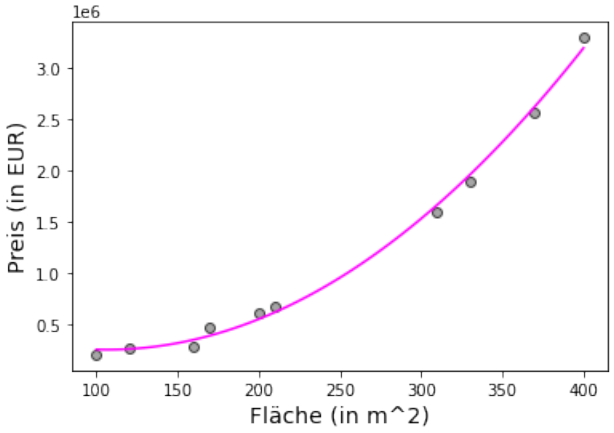

Abbildung 10: Optimal angepasstes „lineares“ Modell für den Datensatz D'_{houses}.

Im vorherigen Beispiel haben wir nur ein einziges polynomielles Merkmal ergänzt. Bei größeren Regressionsproblemen (oder auch Klassifikationsproblemen) ist es meist nicht klar, welche polynomiellen Merkmale hinzugefügt werden und bis zu welchem Grade. Üblicherweise wählt man einen Maximalgrad als Parameter aus, fügt alle möglichen Polynome bis zu diesem Grad als zusätzliche Merkmale ein und „hofft“, dass überflüssige Merkmale bei der Optimierung ignoriert werden (also dass der entsprechende Parameter in θ nahe bei 0 liegt; wir werden uns mit dieser Problematik aber in den nächsten beiden Abschnitten noch genauer beschäftigen).

Beispiel 13. Angenommen wir haben einen Datensatz D mit Beispielen der Form $(x^{(i)}, y^{(i)})$, wobei $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) \in \mathbb{R}^3$ und $y^{(i)} \in \mathbb{R}$. Die *polynomielle Merkmalserweiterung* von D für den Maximalgrad 2 besteht entsprechend aus den Beispielen $(\hat{x}^{(i)}, y^{(i)})$ mit

$$\hat{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_1^{(i)}x_2^{(i)}, x_2^{(i)}x_3^{(i)}, x_1^{(i)}x_3^{(i)}, (x_1^{(i)})^2, (x_2^{(i)})^2, (x_3^{(i)})^2)$$

Der Datensatz D'_{houses} aus Beispiel 12 ist also eine polynomielle Merkmalserweiterung für den Maximalgrad 2. Da dort die Beispiele nur über ein Merkmal verfügten, musste auch nur ein weiteres polynomielles Merkmal bei der Erweiterung hinzugefügt werden. Beispiel 13 macht deutlich, dass bei mehreren Merkmalen bei der polynomiellen Erweiterung eine ganze Reihe zusätzlicher Merkmale hinzukommen können.

Lineare Regression auf polynomiellen Erweiterungen nennt man auch direkt *polynomielle Regression*. Es ist wichtig hervorzuheben, dass wir auch nach Einführung polynomieller Merkmale zur Berechnung des am besten angepassten Modells immer noch Gleichung (1) benutzen und damit ein *lineares Modell* berechnen. Dieses Modell ist allerdings nur linear bezüglich des erweiterten Merkmalsraums und erscheint polynomiell bei Projektion auf den Ursprungsraum (wie in Abbildung 10).

2.1.4 Über- und Unteranpassung

Die Möglichkeit, zusätzliche Merkmale durch beliebige Verknüpfung vorhandener Merkmale zu generieren, erlaubt es, beliebig komplexe Modelle zu konstruieren und damit beliebig komplexe Funktionen zu approximieren. Dies führt zu der Frage, warum man statt eines linearen Modells nicht direkt ein maximal komplexes Modell nimmt, das damit auch beliebig genau an den Trainingsdatensatz angepasst werden kann. Zunächst ergeben sich dadurch ressourcenspezifische Probleme, da das Lernen unter Umständen signifikant mehr Zeit benötigt. Viel signifikanter dabei ist allerdings das Problem der *Überanpassung* (engl. *overfitting*), d. h., das gelernte Modell ist aufgrund seiner Komplexität so stark an die Trainingsdaten angepasst, dass es nicht mehr gut auf ungesehene Daten generalisiert. Das dazu entgegengesetzte Problem ist die sogenannte *Unteranpassung* (engl. *underfitting*), d. h., das gelernte Modell ist nicht ausdrucksstark genug, um sowohl die Trainings- als auch die Testdaten vernünftig zu modellieren. Das Erkennen von Unter- und

x	y
1	17
2	27
3	43
4	73
5	80
6	82
7	95
8	92
9	99
10	104

Tabelle 6: Datensatz D^* zu Beispiel 14.

Überanpassung eines Modells und das richtige Abwägen zwischen diesen beiden Aspekten ist eine der wichtigsten Aufgaben in der Entwicklung eines adäquaten Modells für eine bestimmte überwachte Lernaufgabe.

Beispiel 14. Betrachten wir den Datensatz D^* in Tabelle 6. Abbildung 11 zeigt lineare Modelle für den Datensatz D^* und dessen polynomielle Erweiterungen bis Grad 9. Das Modell für $d = 1$ entspricht dabei dem normalen linearen Regressionsmodell auf dem Ursprungsdatensatz D^* und die Modelle der übrigen Abbildungen nehmen je ein höheres Polynom auf dem Merkmal x mit auf, bis zu x^9 in der letzten Abbildung, d. h. die Funktion f^9 in der letzten Abbildung ist von der Form

$$f^9(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_9 x^9$$

Die Komplexität der Modelle nimmt mit der Hinzunahme von Termen höherer Grade weiter zu und komplexere Modelle sind somit in der Lage, die Trainingsdaten beliebig genau zu approximieren. Offensichtlich ist das Modell für $d = 1$ zu einfach, um die Trainingsdaten korrekt zu erklären: dieses Modell ist *unterangepasst*. Auf der anderen Seite ist das Modell für $d = 9$ *überangepasst*. Es repräsentiert die Trainingsdaten zwar perfekt, ist allerdings nicht robust gegenüber möglichen Störungseinflüssen in den Trainingsdaten und modelliert die Daten in unintuitiver Weise. Gleiches gilt für die Modelle für $d = 4$ bis $d = 8$. Die beiden Modelle für $d = 2$ und $d = 3$ erscheinen am passendsten, auch wenn diese die Trainingsdaten nicht perfekt modellieren.

Das Problem der Abwägung von Über- und Unteranpassung wird auch *Verzerrung-Varianz-Dilemma* genannt (engl. *bias-variance tradeoff*). Im Allgemeinen ist der *Verzerrungsfehler* eines Modells (engl. *bias error*) der Fehler, der durch zu einfache Annahmen in der Modellstruktur entsteht und damit eventuell zu Unteranpassung führt. Der *Varianzfehler* eines Modells (engl. *variance error*) ist der Fehler, der durch zu starke Anpassung an potentiell verzerrte Trainingsdaten entsteht und damit eventuell zu Überanpassung führt.

Um ein Modell zu bestimmen, das weder zu stark unterangepasst noch zu stark überangepasst ist, hilft es, die Verläufe der Kostenfunktionswerte (oder des Bestimmtheitsmaßes) bei Trainings- und Testdaten mit steigender Modellkomplexität zu betrachten. Abbildung 12 zeigt typische Verläufe des Bestimmtheitsmaßes in diesem Kontext. Wie wir bereits zuvor gesehen haben, nimmt die Bestimmtheit eines Modells mit steigender Komplexität auf den Trainingsdaten zu: je ausdrucksstärker das Modell ist, desto besser wird es an die Trainingsdaten angepasst. Bei den Testdaten ist der Verlauf etwas komplexer. Üblicherweise nimmt die Bestimmtheit zunächst zu: solange das Modell unterangepasst ist, können weder Trainings- noch Testdaten gut modelliert werden, aber je näher man an das „korrekte“ Modell kommt, desto besser werden insbesondere auch die Vorhersagen auf den Testdaten. Steigt die Modellkomplexität aber weiter, so wird das Modell überangepasst und die Vorhersagequalität auf den Testdaten sinkt wieder. Die optimale Modellkomplexität liegt also in diesem Fall am Scheitelpunkt des Kurvenverlaufs zu den Testdaten. Hier ist die Bestimmtheit sowohl bei den Trainings- als auch bei den Testdaten relativ hoch und beide Werte relativ nah beieinander. Modelle, deren Komplexität links davon liegen sind unterangepasst und Modelle, die rechts von diesem Punkt liegen, sind überangepasst.

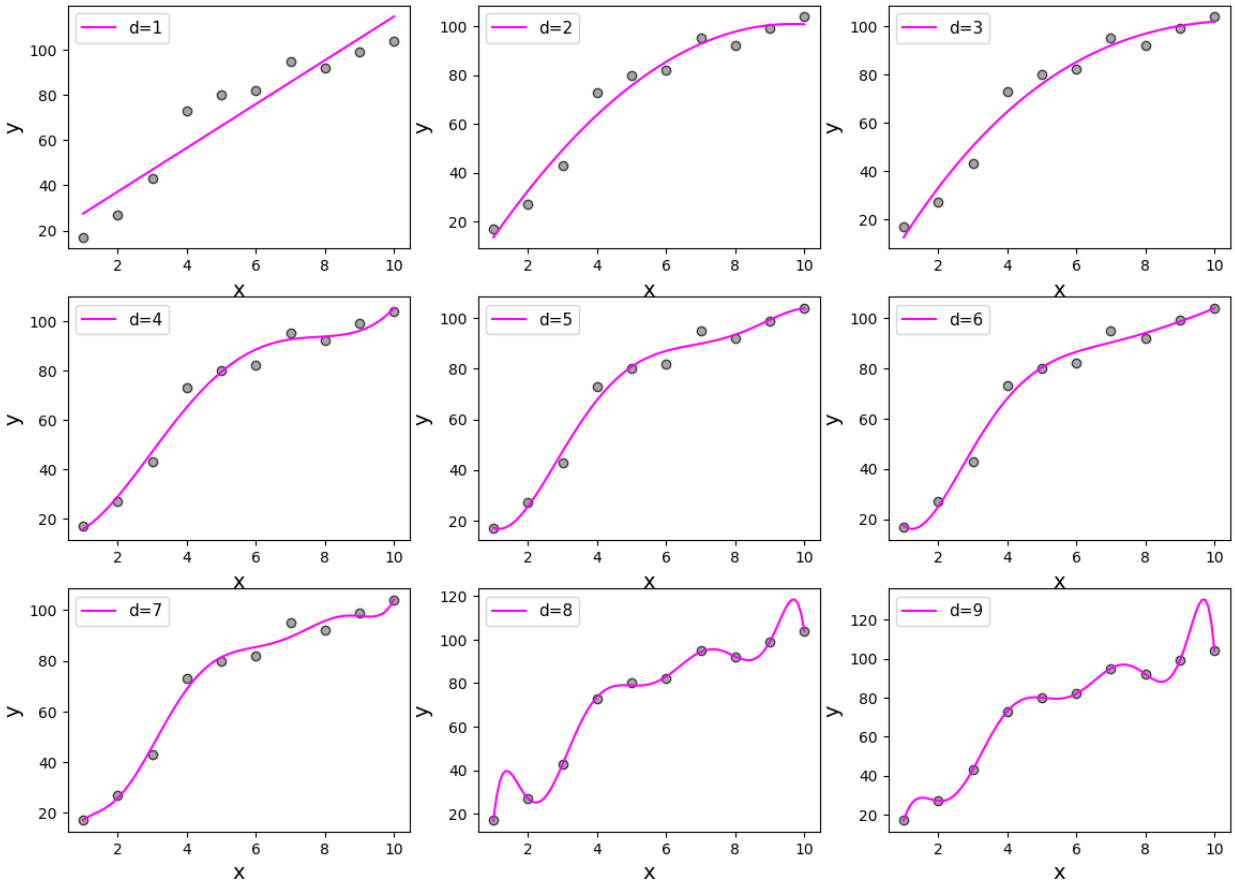

Abbildung 11: Optimal angepasste lineare Modelle für den Datensatz D^* (links oben) und dessen polynomielle Erweiterungen bis Grad 9.

Beispiel 15. Wir führen Beispiel 14 fort und benutzen weiterhin Datensatz D^* als Trainingsdaten. Wir betrachten noch einen weiteren Datensatz D^*_{test}, den wir als Testdatensatz benutzen und der zusammen mit Datensatz D^* in Abbildung 13 visualisiert ist. Es sollte offensichtlich sein, dass D^* und D^*_{test} derselben Verteilung entstammen. Die Berechnung der einzelnen Bestimmtheitswerte auf D^* und D^*_{test} bezüglich der verschiedenen Regressionsmodelle aus Beispiel 14 ergibt die in Abbildung 14 dargestellten Verläufe. Die Verläufe bestätigen den Eindruck aus Beispiel 14. Das Modell mit $d = 1$ ist unterangepasst, wohingegen die Modelle mit $d = 4$ bis $d = 9$ überangepasst sind. Die Modelle mit $d = 2$ und $d = 3$ erscheinen optimal.

2.1.5 Regularisierung

Eine Möglichkeit um bei der linearen Regression das Verzerrung-Varianz-Dilemma (semi-automatisch) zu lösen, ist die *Regularisierung*. Regularisierung beschreibt eine Technik, die dafür sorgt, dass ein zu komplexes Modell beim Lernvorgang bestraft wird.

Definition 5. Sei $L(D, f)$ eine beliebige Kostenfunktion mit $f : \mathbb{R}^n \to \mathbb{R}$. Eine Funktion[3] $R : (\mathbb{R}^n \to \mathbb{R}) \to \mathbb{R}^{\geq 0}$ heißt *Regularisierer* und für $\lambda > 0$ heißt

$$L_{R,\lambda}(D, f) = L(D, f) + \lambda R(f)$$

[3] $\mathbb{R}^{\geq 0}$ bezeichne die Menge der nicht-negativen reellen Zahlen.

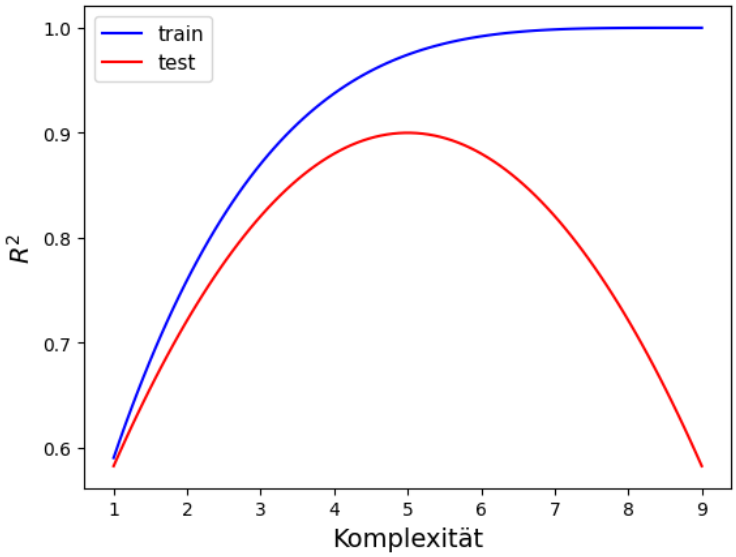

Abbildung 12: Typischer Zusammenhang der Bestimmtheit eines Regressors auf Trainings- und Testdaten bei steigender Komplexität.

die mit R und λ *regularisierte Kostenfunktion*.

Im allgemeinen soll ein Regularisierer R ein Maß für die Komplexität von f implementieren und durch die Einbeziehung des Terms $\lambda R(f)$ in die Kostenfunktion L wird bei Erlernen der Funktion f dessen Komplexität entsprechend berücksichtigt. Je nach Größe von λ (der *Regularisierungsparameter*) werden komplexere Funktionen mehr oder weniger stark bei der Minimierung von $L_{R,\lambda}(D,f)$ berücksichtigt. Schauen wir uns einen konkreten Regularisierer für die lineare Regression an.

Definition 6. Sei $\theta \in \mathbb{R}^{n+1}$ und $h_\theta : \mathbb{R}^n \to \mathbb{R}$ mit

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

für alle $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ ein lineares Modell und $L(D,\theta)$ mit

$$L(D,\theta) = \|X_D\theta - y_D\|^2$$

die entsprechende Kostenfunktion (der quadratische Fehler). Der *Tikhonov-Regularisierer* $R_T : \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ is definiert via

$$R_T(\theta_1, \ldots, \theta_n) = \sum_{i=1}^{n} \theta_i^2$$

und für $\lambda > 0$

$$L_T(D,\theta) = \|X_D\theta - y_D\|^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

die entsprechende regularisierte Kostenfunktion.[4]

[4]Beachten Sie, dass der konstante Term θ_0 des linearen Modells nicht bei der Regularisierung beachtet wird.

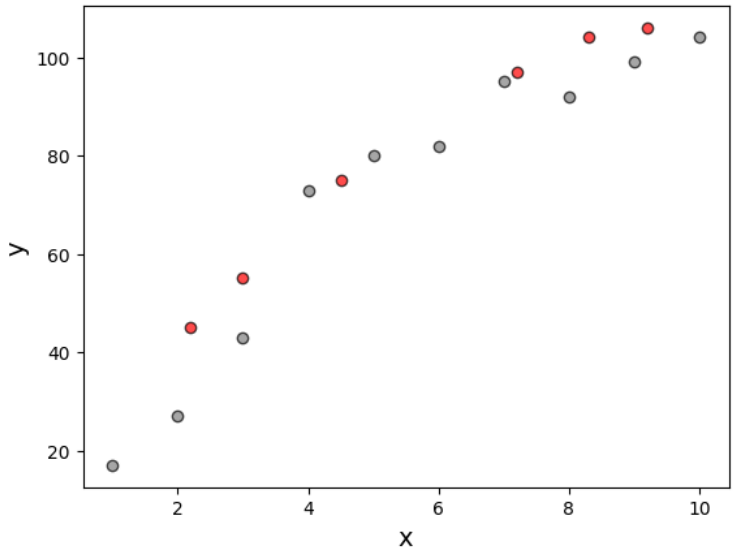

Abbildung 13: Trainingsdatensatz D^* (in grau) und Testdatensatz D^*_{test} (in rot) aus Beispiel 15.

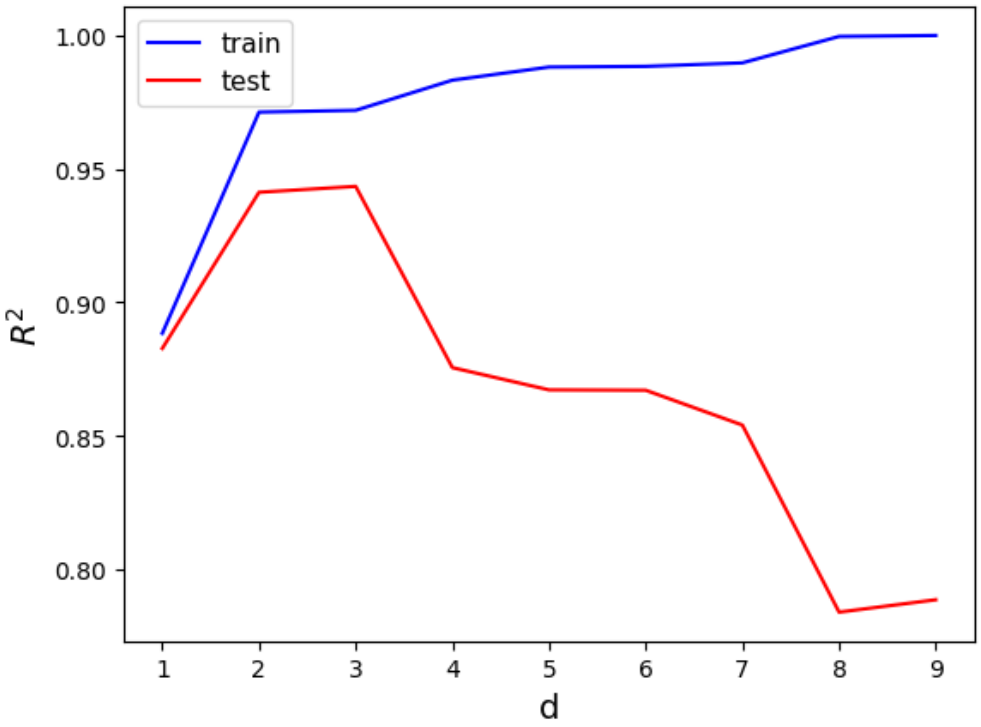

Abbildung 14: Bestimmtheit verschiedener polynomieller Regressionsmodelle auf Trainings- und Testdaten aus Beispiel 15.

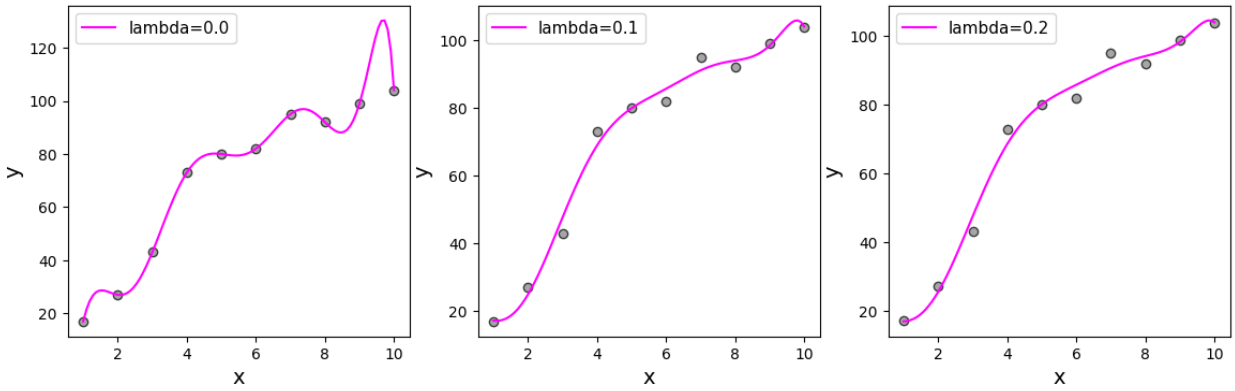

Abbildung 15: Ridge-Regressionsmodelle aus Beispiel 16.

Die lineare Regression mit Kostenfunktion L_T nennt man allgemein auch *Ridge-Regression* (engl. *ridge regression*). Der Term $\lambda \sum_{i=1}^{n} \theta_i^2$ bestraft hohe Werte der einzelnen Elemente von θ und ist minimal, wenn $\theta_1 = \ldots = \theta_n = 0$ ist, also die Funktion h_θ eine Parallele zur ersten Achse darstellt. Solch eine Funktion beschreibt natürlich das einfachste Modell für eine gegebene Lernaufgabe. Der Term $\lambda \sum_{i=1}^{n} \theta_i^2$ wird umso größer, je mehr Elemente von θ ungleich 0 sind und je größer sie sind. Ist dies der Fall, so ist die gelernte Funktion h_θ umso komplexer, da viele Merkmale (insbesondere solche, die zusätzlich durch die in Abschnitt 2.1.3 angesprochenen Techniken hinzugefügt wurden) in die Anpassung einbezogen werden. Mit anderen Worten erzwingt die Einbeziehung des Terms $\lambda \sum_{i=1}^{n} \theta_i^2$ eine Fokussierung auf möglichst einfache Funktionen. Durch den Parameter λ wird gesteuert, ob die gelernte Funktion eher überangepasst wird (bei kleinen Werten von λ) oder eher unterangepasst wird (bei großen Werten von λ). Das Finden eines geeignetes Wertes von λ ist hier also die Kernaufgabe.

Beispiel 16. Abbildung 15 zeigt drei Ridge-Regressionsmodelle für den mit Maximalgrad 9 erweiterten Datensatz D^* aus Tabelle 6 mit drei verschiedenen Parametern $\lambda \in \{0, 0.1, 0.2\}$. Das Modell für $\lambda = 0$ ist somit das normale polynomielle Regressionsmodell und identisch mit dem letzten Modell in Abbildung 11. Bei den anderen Modellen sieht man einen eindeutigen Trend der Überanpassung des ersten Modells entgegenzuwirken. Mit anderen Worten, obwohl die anderen Modelle über den selben Merkmalen definiert sind, werden bei der Optimierung die Parameter vieler (hoch-polynomieller) Merkmale sehr klein gewählt, um den Regularisierungsterm nicht zu groß werden zu lassen.