Zusammenfassung

Maschinelles Lernen

WS 19/20

November 27, 2019

Grundlagen

1.1 Lineare Algebra

1.1.1 Skalarprodukt

- Vektoren $x, y \in \mathbb{R}^n$: $x \circ y = \sum_{i=1}^{n} x_i \cdot y_i = x^T y$
- $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \circ \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 1 \cdot 3 + 2 \cdot 4 = 11$

1.1.2 Vektornorm

$f : \mathbb{R}^n \to \mathbb{R}$ mit

- $f(x) = 0 \Rightarrow x = 0$
- $f(x + y) \leq f(x) + f(y)$ (*Dreiecksgleichung*)
- $f(\alpha x) = |\alpha| f(x)$

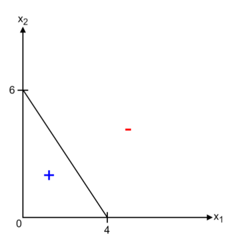
- L_1-Norm: $||x||_1 = \sum_i |x_i|$
- L_2-Norm: $||x||_2 = \sqrt{\sum_i x_i^2}$ (*euklidische Norm*)

1.1.3 Matrizen

- m Zeilen und n *Spalten* A $= \begin{bmatrix} A_{11} & ... & A_{1n} \\ A_{m1} & ... & A_{mn} \end{bmatrix}$, $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$
- $\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \cdot \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} ag + bi + ck & ah + bj + cl \\ dg + ei + fk & dh + ej + fl \end{bmatrix}$, $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $A^{-1} A = I$ (Matrizen mit linear abhängigen Zeilen oder Spalten (niedriger Rang) sind nicht invertierbar)

1.1.4 Hyperebene

- x $\in \mathbb{R}^d$ erfüllen Gleichung $w_0 + w_1 x_1 + w2_x 2 + ... + w_d x_d = 0$ ($w_0 + w^T x = 0$)
- $d = 1$: Skalar ($w_0 + w_1 x_1$), $d = 2$: Gerade ($w_0 + w_1 x_1 + w_2 x_2$), $d = 3$: Ebene
- Für einen Punkt x entscheidet das Vorzeichen $sgn(w_0 + w^T x) \in \{-1, 0, 1\}$ auf welcher Seite der Hyperebene er liegt (bzw. ob er auf ihr liegt)



$1 - \frac{1}{4} x_1 - \frac{1}{6} x_2 = 0$

1.2 Analysis

1.2.1 Kettenregel

- Wenn z von y und y von x abhängt, dann gilt: $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$
- $f(x) = g(h(x)) = \frac{1}{2} \cdot (x_1 - x_2)^2 \rightarrow g(x) = \frac{1}{2}x^2$ und $h(x) = x_1 - x_2$
- $\frac{df}{dx_2} = \frac{dg}{dh}\frac{dh}{d_x 2} = h(x)(-1) = -(x_1 - x_2) = x_2 - x_1$

1.2.2 Partielle Ableitung

$f(x) = 2x_1^3 - 5x_2^2 + 3,\ \frac{df}{dx_1} = 6x_1^2,\ \frac{df}{dx_2} = -10x_2$

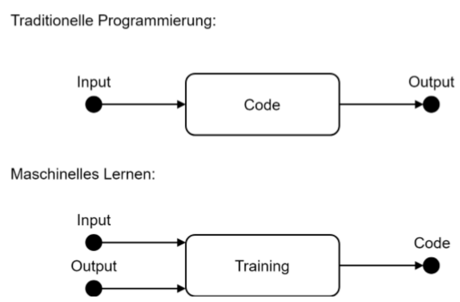
1.2.3 Gradient

$$\nabla f = \begin{bmatrix} \frac{df}{dx_1} \\ . \\ . \\ . \\ \frac{df}{dx_n} \end{bmatrix}, f(x) = 2x_1^3 - 5x_2^2 + 3, \nabla f = \begin{bmatrix} 6x_1^2 \\ -10x_2 \end{bmatrix}$$

1.3 Was ist maschinelles Lernen

1.3.1 Paradigmenwechsel

Es ist schwierig, den entsprechenden Programmcode manuell zu schreiben, daher wird ein anderes Paradigma verwendet:



Drei verschiedene Lernmethoden

- Überwachtes Lernen (*Supervised Learning*)
- Unüberwachtes Lernen (*Unsupervised Learning*)
- Bestärkendes Lernen (*Reinforcement Learning*)

1.4 Überwachtes Lernen

- Ziel: finden einer Funktion $f : X \to Y$ wobei X auch *Features / Prädiktoren* und Y auch *Responses* genannt werden
- $X = \mathbb{R}^d$ (d-dimensionaler Vektorraum) mit $d \in \mathbb{N}$
- Eine perfekte Abbildung ist nicht möglich, es treten *reduzierbare* Fehler (z.B. durch eine bessere Funktion f) und *nicht reduzierbare* Fehler (z.B. Messfehler in Eingabedaten) auf

- *Vorhersage:* $y = f(x)$ optimieren wobei f auch *Blackbox* sein kann
- *Inferenz:* Interpretierbarkeit von f steht im Vordergrund (Welche Prädiktoren sind für welche Response verwantwortlich)
- *Parametrische* Methoden: Annahme einer parametrisierten Struktur von f dessen Parameter mit Hilfe von Daten bestimmt werden
- *Nicht-parametrische* Methoden: Keine Annahme einer Struktur von f sondern möglichst direkte Definition mit Hilfe von Daten

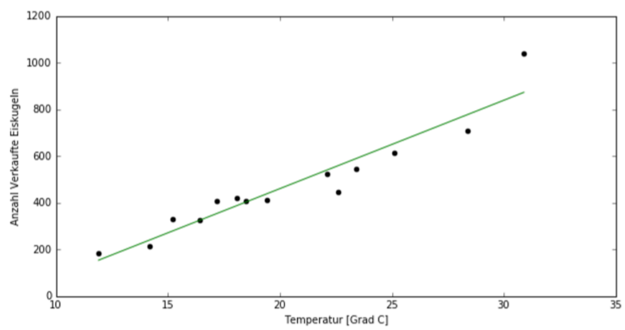
- Menge X und Y bekannt, genaue Abbildung f kann aber nur anhand von Beispielen $D = \{(x^i, y^i) | x^i \in X, y^i \in Y, 1 \leq i \leq n\}$ (*Trainingsdatensatz* bzw. *gelabelte* Daten) erahnt werden

1.4.1 Beispiel Klassifikation

- Wenn Y diskrete Menge $\{C_1, ..., C_k\}$ für $k \in \mathbb{N}$ dann handelt es sich um ein *Klassifikationsproblem*, $C_1, ..., C_k$ sind dann *Klassen / Kategorien*
- $|Y| = 2$ (*Binäre* Klassifikation) mit $f : \mathbb{R} \to \{\text{angenehm, unangenehm}\}$ (Temperaturklassifikation)
- $|Y| = 5$ (*Mehrklassen*-Klassifikation) mit $f : \mathbb{R} \to \{\text{frostig, kalt, angenehm, warm, heiß}\}$

1.4.2 Beispiel Regression

- Wenn Y kontinuierliche Menge, d.h. $Y \subseteq \mathbb{R}$, dann handelt es sich um ein *Regressionsproblem*
- Interesse an *quantitativen* Aussagen



- Ausgabemenge Y kann auch mehrdimensional sein (z.B. $\{$gut, schlecht$\} \times \{$günstig, normal, teuer$\}$)

1.5 Unüberwachtes Lernen

- Mehrwert erhalten ohne Zuhilfenahme von gelabelten Daten
- Man geht von Menge an Daten $D = \{x^i | x^i \in X, 1 \leq i \leq n\}$ aus und versucht mehr über Beschaffenheit von X herauszufinden
- z.B. *Verteilung* von X bei Sprachmodellen, *Dimensionsreduktion* zur Verbesserung von überwachten Lernverfahren

1.6 Datenvisualisierung

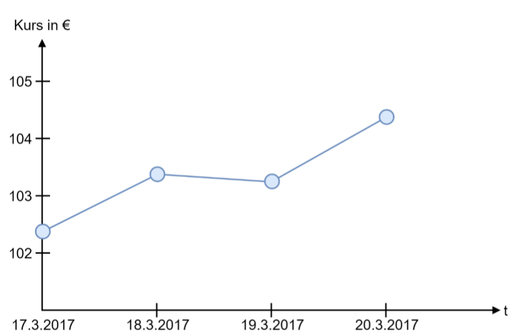


Abbildung 6: Beispiel eines Liniendiagramms.

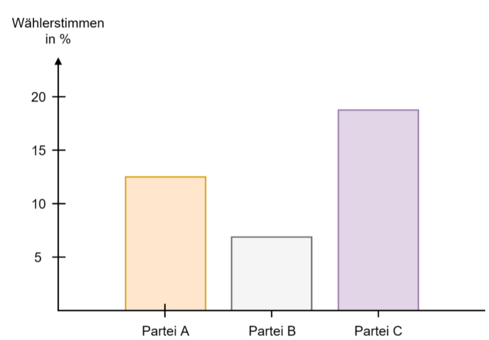


Abbildung 7: Beispiel eines Balkendiagramms.

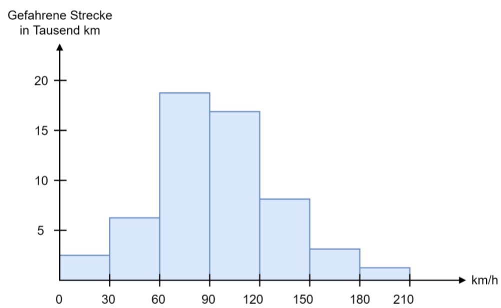


Abbildung 8: Beispiel eines Histogramms – eines speziellen Balkendiagramms.

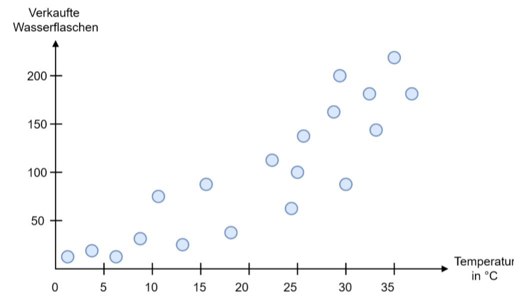


Abbildung 9: Beispiel eines Streudiagramms.

1.7 Datenvorverarbeitung

Bevor ein Modell erstellt und trainiert werden kann, müssen Daten durch

- *Auswahl*: Nur für den Anwendungsfall relevante Daten verwenden
- *Aufbereitung*

  Dateiformat (Tabellen, BigData)

  Bereinigung von unvollständigen oder ungültigen Daten

  Repräsentative Auswahl bei langer Laufzeit / großem Speicheraufwand

- *Transformation*

  Features in geeigneten Wertebereich bringen ([0, 1])

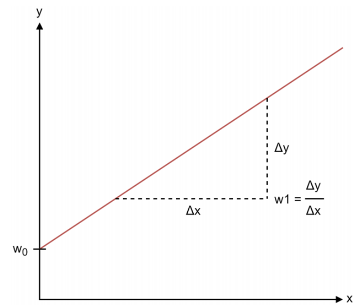
  Zerlegen in sinnvolle Features

  Aggregation mehrerer Features

Lineare Regression

2.1 Lineare Regression im Eindimensionalen

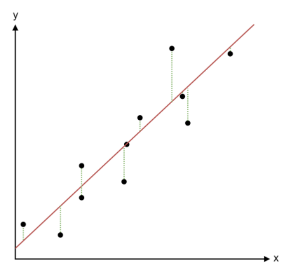
- $f : \mathbb{R} \to \mathbb{R}$ mit $f_w(x) = w_1 x + w_0$
- $w = (w_0, w_1)^T \in \mathbb{R}^2$ sind die *Parameter* des Modells



- Wie mit Daten $D = \{(x^i, y^i) \in \mathbb{R}^2 | 1 \leq i \leq n\}$ die *besten* Parameter von f bestimmen?

2.1.1 Lösungsverfahren

- Quadratischen Fehler (*Residual Sum of Squares*) mit $RSS(w) = \sum_{i=1}^{n}(y^i - f_w(x^i))^2$ bestimmen



- Zur besseren Vergleichbarkeit verwendet man oft die normalisierte Variante *Mean Squared Error*: $MSE(w) = \frac{1}{n} \cdot RSS(w)$ (n = Anzahl Trainingsdaten)
- Die beste Funktion durch Minimierung des Fehlers finden $\Rightarrow w^* = \arg\min E(w) = \arg\min \frac{1}{2} \cdot \sum_{i=1}^{n}(y^i - f_w(x^i))^2$

- Ableitung von $E(w)$ gleich Null setzen und Gleichungssystem lösen
- $\nabla E(w) = \begin{bmatrix} \frac{dE(w)}{dw_0} \\ \frac{dE(w)}{dw_1} \end{bmatrix} = 0$

  $\frac{dE(w)}{dw_0} = -\sum_{i=1}^{n} y^i + w_1 \cdot \sum_{i=1}^{n} x^i + n \cdot w_0$

  $\frac{dE(w)}{dw_1} = -\sum_{i=1}^{n} x^i y^i + w_1 \cdot \sum_{i=1}^{n} x^i x^i + w_0 \cdot \sum_{i=1}^{n} x^i$
- Gleichungssystem mit zwei Gleichungen und zwei Unbekannten lösbar, aber numerisch ungenau bei großen Matrizen

2.1.2 Gradientenabstiegsverfahren

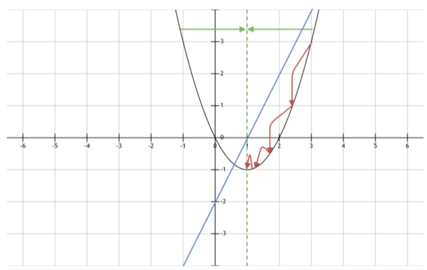


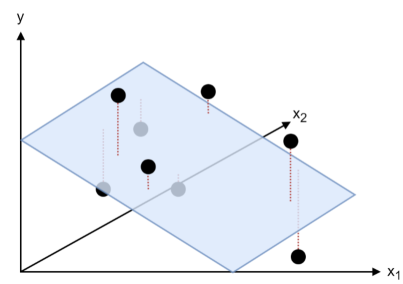
Abbildung 5: Gradientenabstiegsverfahren auf $f(x) = x(x-2)$

- Iterativ einem Bruchteil der negativen Ableitung: $-\eta f'(x) = \eta \cdot (2-2x)$ folgen
- *Lernrate* η hat direkten Einfluss auf Konvergenz (zu klein $\Rightarrow$ viele Schritte, zu groß $\Rightarrow$ Oszillation)

```
w0 = 0, w1 = 0
for (x, y) in D
  dw0 += -y + w1*x + w0
  dw1 += -xy + w1*x*x + w0*x
end for
w0 += -eta*dw0
w1 += -eta*dw1
```

2.2 Mehrdimensionale Lineare Regression

- $X = \mathbb{R}^d$ und $f : \mathbb{R}^d \to \mathbb{R}$ sowie $f_w(x) = \sum_{i=1}^{d} w_i x_i + w_0$ mit Parametern $w = (w_0, w_1, ..., w_d)^T \in \mathbb{R}^{d+1}$ - Kompaktere Schreibweise mit $x_0 = 1$: $f_w(x) = \sum_{i=1}^{d} w_i x_i + w_0 = w^T x$



- Im Mehrdimensionalen wird eine Hyperebene, im dreidimensionalen eine Ebene, im Raum so positioniert, dass der Abstand zu den Datenpunkten minimiert wird

- Angepasste Fehlermetrik $E(w) = \frac{1}{2} \cdot \sum_{i=1}^{n}(y^i - f(x^i))^2$ mit $\nabla E(w) = \begin{bmatrix} \frac{dE(w)}{dw_0} \\ \frac{dE(w)}{dw_1} \\ \dots \\ \frac{dE(w)}{dw_d} \end{bmatrix}$

```
dw = 0
for (x, y) in D
  dw += -(y - f(x) * gradF(x))
end for
w += -eta*dw
```

wobei gradF(x) $= \nabla f(x) = \begin{bmatrix} 1 \\ x_1 \\ \dots \\ x_d \end{bmatrix}$

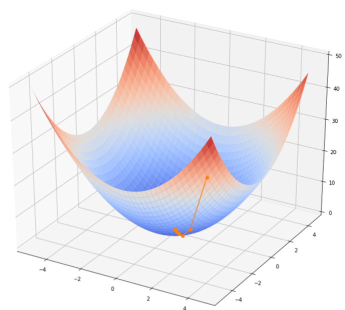


Abbildung 9: Gradientenabstiegsverfahren im mehrdimensionalen Raum bei der Funktion $f(\mathbf{x}) = \mathbf{x}_1^2 + \mathbf{x}_2^2$.

2.3 Genauigkeit

- Wie gut ist das durch das Gradientenabstiegsverfahren gefundene Modell?
$\Rightarrow$ Quadratischer Fehler RSS oder mittlerer quadratischer Fehler MSE
- Letzterer ist unabhängig von der Anzahl an Trainingsdaten allerdings gibt es keine allgemein gültige Skala da diese vom Wertebereich der y-Werte abhängt

2.3.1 R^2 Statistik

- Definiert über den quadratischen Gesamtfehler $TSS = \sum_{i=1}^{n}(y^i - \bar{y})^2$
- $\bar{y} = \frac{1}{n} \cdot \sum_{i=1}^{n} y^i \Rightarrow R^2(w) = \frac{TSS - RSS(w)}{TSS} = 1 - \frac{RSS(w)}{TSS}$
- TSS misst die komplette Varianz in den Ausgabedaten y^i
- $TSS - RSS(w)$ misst die durch das Modell mit Parametern w erklärte Varianz
- R^2 misst die komplette Varianz des Modells und ist $\in [0, 1]$

- R^2 nahe 1 zeugt von einem passenden Model das die Daten gut erklärt

- R^2 nahe 0 bedeutet, dass das Modell die Daten schlecht erklärt

- R^2 ist unabhängig von Anzahl an Trainingsdaten *UND* dem Wertebereich
- Allgemeine Aussage ab welchem R^2-Wert das Modell *gut* ist, ist nicht möglich. Hängt vom Anwendungsfall (Medizin / Physik) ab

2.4 Interpretierbarkeit

- Die Parameter w von Linearen Regressionsmodellen sind interpretierbar:

- $w_i > 0$: positiver Zusammenhang, steigt x_i um m so steigt y um $m \cdot |w_i|$
- $w_i nahe 0$: kein linearer Zusammenhang zwischen x_i und y
- $w_i < 0$ negativer Zusammenhang, steigt x_i um m so sinkt y um $m \cdot |w_i|$

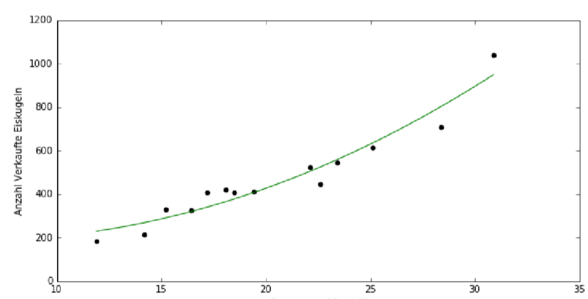
2.5 Nichtlineare Zusammenhänge

- Mit der mehrdimensionalen linearen Regressions lassen sich auch *nichtlineare* Zusammehänge lernen
- Mit Funktion $\Phi : \mathbb{R} \to \mathbb{R}^d$ wird ein *Basiswechsel* vollzogen

- Die Konkatenation von $\Phi : \mathbb{R} \to \mathbb{R}^d$, $\Phi(x) = (x, x^2)^T$ und $f : \mathbb{R}^2 \to \mathbb{R}, f(x) = w_2 x_w + w_1 x_1 + w_0$ durch $f \circ \Phi$ erlaubt Darstellung der quadratischen Funktion $(f \circ \Phi)(x) = f(\Phi(x)) = w_2 x^2 + w_1 x + w_0$
- $\Phi : \mathbb{R}^2 \to \mathbb{R}^5, \Phi(x) = (x_2, x_1, x_1 x_2, x_2^2, x_1^2)^T$
und $f : \mathbb{R}^5 \to \mathbb{R}, f(x) = \sum_{i=1}^{5} w_i x_i + w_0$ ergibt $(f \circ \Phi)(x) = f(\Phi(x)) = w_5 x_1^2 + w_4 x_2^2 + w_3 x_1 x_2 + w_2 x_1 + w_1 x_2 + w_0$

2.5.1 Beispiel

- Annahme eines quadratischen Zusammenhangs $f(x) = w_2 \cdot x^2 + w_1 \cdot x + w_0$



2.5.2 Richtiger Grad

- Mit der mehrdimensionalen Regression, dem Basiswechsel und Gradientenabstiegsverfahren ist es möglich, ein Polyon n-ten Grades an n Datenpunkte zu fitten

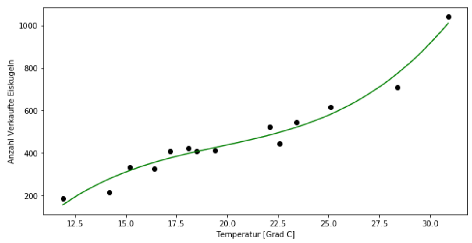


Abbildung 12: Lineare Regression eines Polynoms 3-ten Grades an die Eisverkaufdaten durch Basiserweiterung. Gewichte $\mathbf{w} \approx (-1853, 307, -14.6, 0.247)^T$

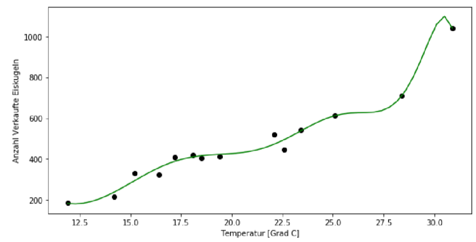


Abbildung 13: Lineare Regression eines Polynoms 12-ten Grades an die Eisverkaufdaten durch Basiserweiterung. Gewichte $\mathbf{w} = (0, -0.0000457, -0.00000496, -0.0000570, -0.000489, -0.00297, -0.00977, 0.00256, -0.000271, 0.0000152, -0.000000471, 0.00000000777, -0.0000000000530)^T$

- Mit höherer Modellkomplexität (Grad und Koeffizienten des Polynoms) kommt es zu

- Numerischen Problemen
- *Overfitting*: Das Modell passt sich zu sehr an die Daten an und ist nicht mehr in der Lage zu generalisieren → Schlechte Leistung in der Praxis

2.6 Trainings- und Testdaten

- Datensatz D wird in zwei disjunkte Teile T und V aufgeteilt
- Trainingsdatensatz T wird für das Lernen verwendet
- Testdatensatz V enthält ungesehene Daten zur Validierung der Praxistauglichkeit

- Ein hoher Fehler auf T lässt auf Unteranpassung schließen (zu geringe Modellkomplexität, zu wenig Daten)
- Ein geringer Fehler auf T aber hoher Fehler auf V bedeutet Überanpassung → Komplexität verringern

2.7 Optimierung von Hyperparametern

- Lineare Regression auf Polynomen mit Gradientenabstiegsverfahren besitzt

- Lernrate η: Einfluss auf Modellkomplexität
- Anzahl Lernschritte: Je geringer desto unwahrscheinlicher ist Überanpassung, allerdings Unteranpassung wiederum möglich
- Polynomgrad Zu Hoch → Überanpassung, zu niedrig → Unteranpassung

als Hpyerparameter

2.7.1 Rastersuche

- Durchsuchen des Hyperparameterraums entweder

- Entlang eines gleichmäßigen Rasters mit linearer oder logarithmischer Skala
- Entlang eines zufälligen Rasters mit uniformer oder logarithmischer Skala

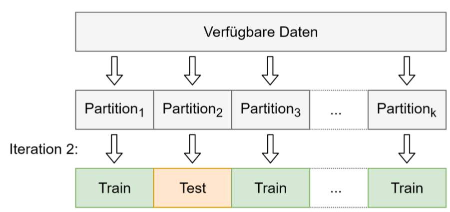
- Verfeinern der Suche durch Rekursion

2.7.2 Validierungsdaten

- Sollen die Hyperparameter des Modells optimiert werden, werden die verfügbaren Daten D in *Trainingsdaten*, *Validierungsdaten* und *Testdaten* aufgeteilt.
- Die Hyperparameter werden mit dem Validierungsdatensatz optimiert - Endgültige Performance des Modells wird auf den Testdaten bestimmt

2.7.3 Kreuzvalidierung

- Zerteilen des Datensatzes in k Partitionen, ws wird nun k-mal trainiert
- Mit jeder Iteration i wird eine andere Partition i getestet
- Die Restlichen Partitionen dienen als Trainingsdaten
- Final wird die ausgewählte Leistungsmetrik über k Iterationen gemittelt



2.7.4 Ridge Regression

- Verhindern von Überanpassung durch Bestrafung von w für exzessive Werte mit angepasster Fehlerfunktion $E(w) = \frac{1}{2} \cdot \sum_{i=1}^{n}(y^i - f_w(x^i))^2 + \alpha||w||^2$
- Hyperparamter $\alpha \in \mathbb{R} \geq 0$ ist ein weiterer Freiheitsgrad mit dem sich der Polynomgrad stufenlos einstellen lässt

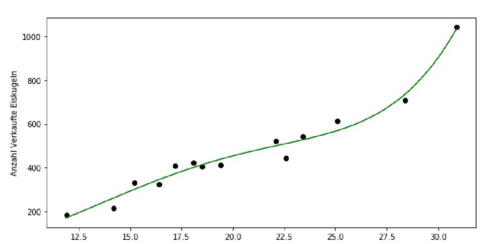


Abbildung 16: Ridge Regression eines Polynoms 5-ten Grades an die Eisverkaufdaten durch Basiserweiterung, $\alpha = 10$.

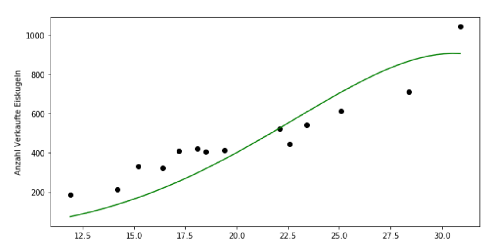


Abbildung 17: Ridge Regression eines Polynoms 5-ten Grades an die Eisverkaufdaten durch Basiserweiterung, $\alpha = 10^{10}$.

- $\alpha = 0$: klassische Regression
- $\alpha > 0$: Normaler Wirkungsbereich, mit wachsendem α werden w immer weiter eingeschränkt und der effektive Polynomgrad sinkt
- lim $\alpha \to \infty$: $f(x) = 0$ da Parameter lim $w \to 0$