Gradient-Boosting für Regressionsprobleme

Machine Learning Seminar

Kevin Scislak

geleitet von Dr. Zoran Nikolić

28. Juni 2019

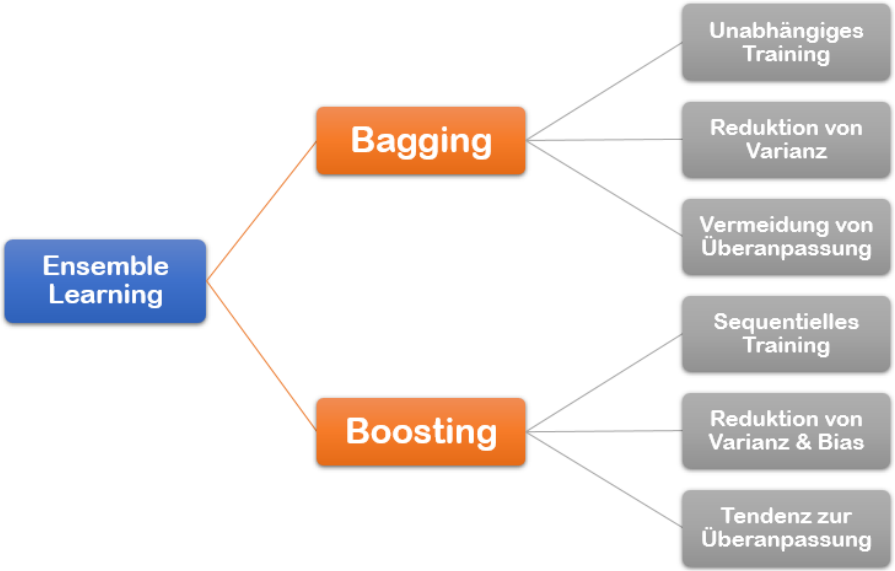
Agenda

1. Motivation
2. Ensemble Learning
3. Gradient-Boosting
4. Programmierung

Motivation

- Mächtiger Lernalgorithmus für Klassifikations- und Regressionsprobleme
  - → **Klassifikation:** Identifikation betrügerischer Kreditkartentransaktionen
  - → **Regression:** Vorhersage des Körpergewichts von Neugeborenen
- Gewinner von Kaggle-Wettbewerben nutzen meistens Gradient-Boosting
- Datenvorverarbeitung nicht notwendig und hohe Flexibilität des Modells

Überblick



Gradient-Boosting

Was ist die grundlegende Idee?

- Kombination vieler schwacher Modelle zu einem starken Modell
  → Typischerweise viele einzelne Entscheidungsbäume
- Sequentielle Anpassung bis gewählte Abbruchbedingung eintritt
  → Optimierung durch differenzierbare Fehlerfunktion

Wie funktioniert das Verfahren?

1. Initialisiere ein einfaches Vorhersagemodell
2. Trainiere neues Modell, das aus Fehlern des alten lernt
3. Kombiniere die schwachen Modelle zu einem stärkeren Modell
4. Wiederhole Schritt 2 und 3 bis gewählte Abbruchbedingung eintritt

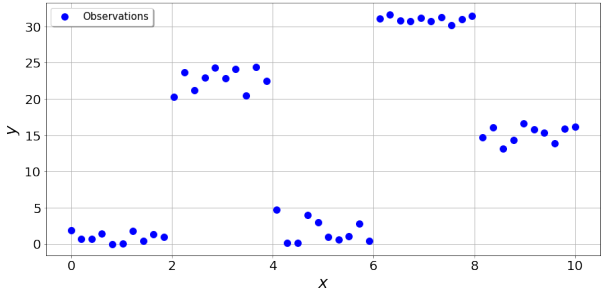
Einführungsbeispiel

Regressionsproblem

Gegeben: Input $X = \{x_i\}_{i=1}^N$ mit $x_i \in \mathbb{R}$ und Output $Y = \{y_i\}_{i=1}^N$ mit $y_i \in \mathbb{R}$ sowie differenzierbare Fehlerfunktion $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$.

Gesucht: Modell $\hat{f}(x)$, welches den Fehler $\sum_{i=1}^N \frac{1}{2}\left(y_i - \hat{f}(x_i)\right)^2$ minimiert.

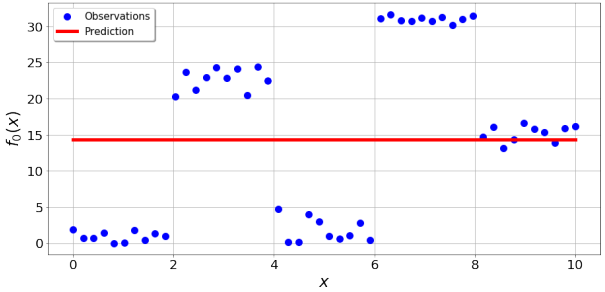
i	x_i	y_i
1	0.0	1.88
2	0.2	0.64
3	0.4	0.69
⋮	⋮	⋮
50	10	1.61



Gradient-Boosting: Schritt 1

- Initialisiere schwaches Vorhersagemodell $f_0(x) = \frac{1}{N} \sum_{i=1}^{N} y_i$.

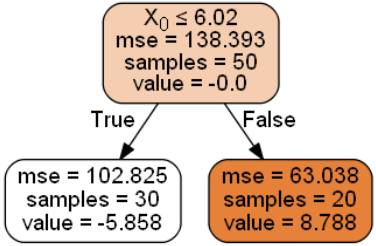
i	x_i	y_i	$f_0(x_i)$
1	0.0	1.88	14.3
2	0.2	0.64	14.3
3	0.4	0.69	14.3
⋮	⋮	⋮	⋮
50	10	16.1	14.3



Gradient-Boosting: Schritt 2

- Berechne Residuen $r_{i1} = y_i - f_0(x_i)$. Trainiere neues Modell auf $\{(x_i, r_{i1})\}_{i=1}^{N}$.

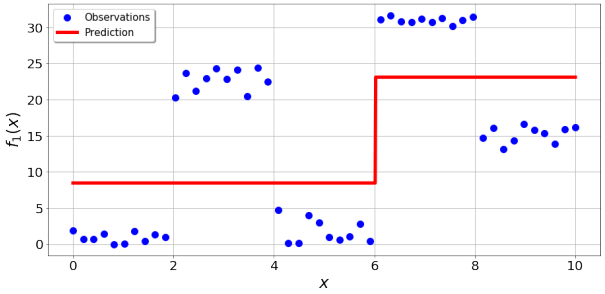
i	x_i	y_i	$f_0(x_i)$	r_{i1}	$h_1(x_i)$
1	0.0	1.88	14.3	−12.42	−5.858
2	0.2	0.64	14.3	−13.66	−5.858
3	0.4	0.69	14.3	−13.61	−5.858
⋮	⋮	⋮	⋮	⋮	⋮
50	10	16.1	14.3	1.8	8.788



Gradient-Boosting: Schritt 3

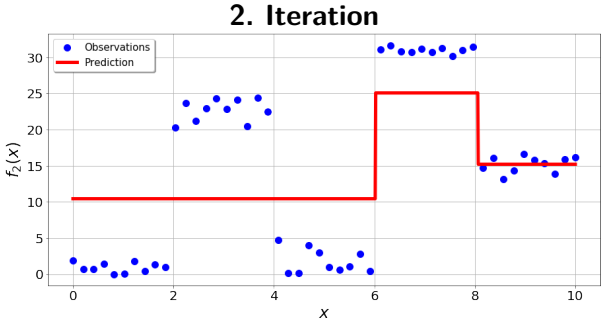
- Anpassung des neuen Modells durch $f_1(x_i) = f_0(x_i) + h_1(x_i)$.

i	x_i	y_i	$f_1(x_i)$
1	0.0	1.88	8.442
2	0.2	0.64	8.442
3	0.4	0.69	8.442
⋮	⋮	⋮	⋮
50	10	16.1	23.088



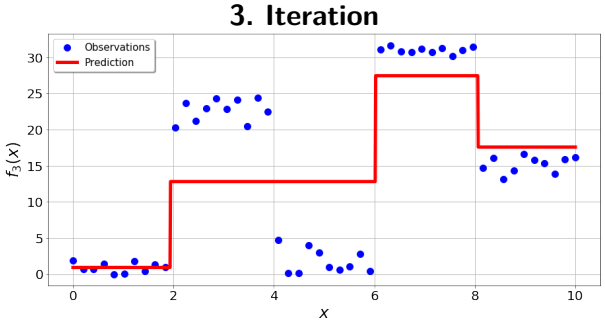
Gradient-Boosting: Optimierung

- Wiederhole Schritt 2 und 3 bis Abbruchbedingung eintritt.



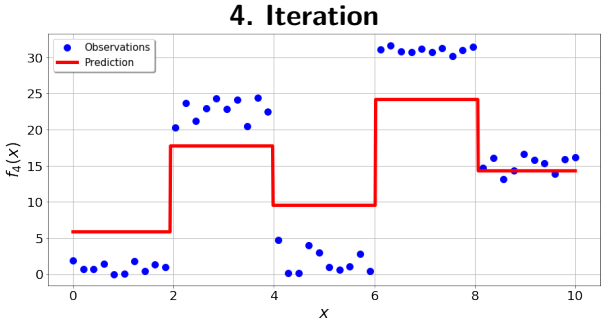
Gradient-Boosting: Optimierung

- Wiederhole Schritt 2 und 3 bis Abbruchbedingung eintritt.



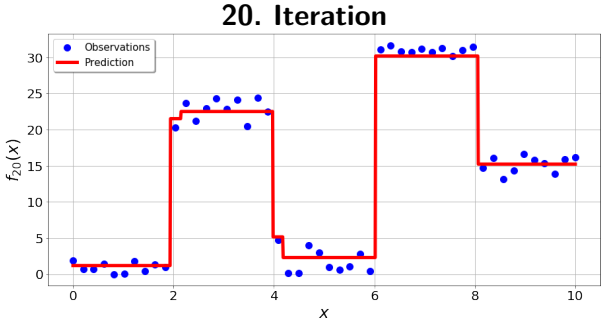
Gradient-Boosting: Optimierung

- Wiederhole Schritt 2 und 3 bis Abbruchbedingung eintritt.



Gradient-Boosting: Optimierung

- Wiederhole Schritt 2 und 3 bis Abbruchbedingung eintritt.



$\Rightarrow$ Output: $\hat{f}(x) = f_{20}(x)$.

Optimierungsproblem

Wo taucht der Gradient auf?

- Betrachten eine differenzierbare Fehlerfunktion

$$L\big(y, f(x)\big) \;=\; \frac{1}{2}\big(y - f(x)\big)^2.$$

- Suchen ein optimales Modell, sodass gilt

$$\hat{f}(x) \;=\; \underset{f(x)}{\operatorname{argmin}} \sum_{i=1}^{N} L\big(y_i, f(x_i)\big) \;=\; \underset{f(x)}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2}\big(y_i - f(x_i)\big)^2.$$

- Komponente des Gradienten ergibt

$$\frac{\partial}{\partial f(x_i)} \sum_{k=1}^{N} \frac{1}{2}\big(y_k - f(x_k)\big)^2 \;=\; \frac{\partial}{\partial f(x_i)} \frac{\big(y_i - f(x_i)\big)^2}{2} \;=\; f(x_i) - y_i.$$

Folgerung: Residuen entsprechen hier negativen Gradienten der Fehlerfunktion.

Fehlerfunktionen für Regressionsprobleme

- **Quadratischer Fehler:**
  - $L(y_i, f(x_i)) = \frac{1}{2}(y_i - f(x_i))^2 \quad \Rightarrow \quad -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = y_i - f(x_i)$
- **Absoluter Fehler:**
  - $L(y_i, f(x_i)) = |y_i - f(x_i)| \quad \Rightarrow \quad -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = \operatorname{sgn}(y_i - f(x_i))$
- **Huber Fehler:**
  - $L(y_i, f(x_i)) = \begin{cases} \frac{1}{2}(y_i - f(x_i))^2, & |y_i - f(x_i)| \leq \delta \\ \delta |y_i - f(x_i)| - \delta^2/2, & |y_i - f(x_i)| > \delta \end{cases}$

$$\Rightarrow -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = \begin{cases} y_i - f(x_i), & |y_i - f(x_i)| \leq \delta \\ \delta \operatorname{sgn}(y_i - f(x_i)), & |y_i - f(x_i)| > \delta \end{cases}$$

Gradient-Tree-Boosting Algorithmus

Input: Trainingsdaten $\{(x_i, y_i)\}_{i=1}^N$ und differenzierbare Fehlerfunktion $L(y, f(x))$.

1. Schritt: Initialisiere konstantes Modell $f_0(x) = \arg\min_\gamma \sum_{i=1}^N L(y_i, \gamma)$.

2. Schritt: Für $m = 1, 2, \ldots, M$:

a) Berechne *Pseudo-Residuen*: $r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$, $i = 1, 2, \ldots, N$.

b) Trainiere einen Entscheidungsbaum auf den Trainingsdaten $\{(x_i, r_{im})\}_{i=1}^N$.
→ Feature-Raum wird in die Regionen R_{jm} mit $j = 1, 2, \ldots, J_m$ unterteilt.

c) Berechne Werte in Blättern:

$$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma), \quad j = 1, 2, \ldots, J_m.$$

d) Passe das Regressionsmodell an: $f_m(x) = f_{m-1}(x) + \alpha \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Schritt: Output $\hat{f}(x) = f_M(x)$.

Stärken und Schwächen

- **Stärken:**
  - Flexibel einsetzbar für Klassifikations- und Regressionsprobleme
  - Unterstützt unterschiedliche differenzierbare Fehlerfunktionen
  - Komplexe Zusammenhänge in Trainingsdaten werden erlernt

- **Schwächen:**
  - Optimierung der Hyperparameter in der Regel sehr rechenintensiv
  - Sequentielles Training des Modells erlaubt keine Parallelisierung
  - Überanpassung bei verrauschten Trainingsdatensätzen möglich

Programmierung

1. Einführungsbeispiel
2. Anwendungsbeispiel
   - 2.1 Hyperparameteroptimierung
     - 2.1.1 Decision Tree
     - 2.1.2 Random Forest
     - 2.1.3 Gradient Boosting
     - 2.1.4 Histogram-based Gradient Boosting
   - 2.2 Fehleranalyse
   - 2.3 Voting & Stacking
   - 2.4 Interpretation
3. Tipps für die praktische Anwendung

Literatur

- T. Hastie, R. Tibshirani, and J.H. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Verlag, 2009.
- A. Mayr, H. Binder, O. Gefeller and M. Schmid. (2014). The Evolution of Boosting Algorithms From Machine Learning to Statistical Modelling. Methods of information in medicine. 53. 10.3414/ME13-01-0122.