„Verallgemeinerte additive Modelle“ und „Baumbasierte Methoden“

Helena Schmitz

Mathematisches Institut Köln

21. Mai 2019

Inhaltsverzeichnis

Verallgemeinerte Additive Modelle

Lineares Regressionsmodell:

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i X_i$$

Additives Regressionsmodell:

$$\hat{Y} = \alpha + \sum_{i=1}^{p} \hat{f}_i(X_i)$$

Im Normalfall ist $\hat{Y}$ normalverteilt und stetig.

Verallgemeinertes Additives Modell:

$$\hat{Y} = \alpha + \sum_{i=1}^{p} \hat{f}_i(X_i)$$

- $X_1, \ldots, X_p$: unabhängige Variablen = Prädiktoren/Features
- Y: abhängige Variable = Zielgröße/Antwort
- f_j: unspezifische Glättungsfunktion
- α: Parameter

$\hat{Y}$ muss nicht normalverteilt und stetig sein.

Link-Funktion

Link-Funktion: Additive Funktion g, die in Beziehung zu dem bedingten Erwartungswert $\mu(X)$ von Y gesetzt wird:

$$g[\mu(X)] = \alpha + f_1(X_1) + \cdots + f_p(X_p).$$

Beispiele von Link-Funktionen:

- $g(\mu) = \mu$
- $g(\mu) = \text{logit}(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$
- $g(\mu) = \log(\mu)$

Nicht alle $f_j's$ müssen nicht-linear sein.
$\rightarrow$ Mischen von linearen und nicht-linearen Formen möglich.

Beispiel:

- $g(\mu) = X^{\mathrm{T}}\beta + \alpha_k + f(Z)$:
  Semiparametrisches Modell: X, wird linear und Z nichtparametrisch modelliert.

Betrachte:

$$Y = \alpha + \sum_{j=1}^{p} f_j(X_j) + \epsilon,$$

mit Fehlerterm ϵ, der den Erwartungswert 0 hat.

Wir nutzen einen Streudiagrammglätter: Einen *(cubic) smoothing spline*.
Dieser minimiert die bestrafte Summe der Quadrate:

$$PRSS(\alpha, f_1, \ldots, f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 \, \mathrm{d}t_j.$$

Backfitting Algorithmus

1. Initialisierung: $\hat{\alpha} = \frac{1}{N} \sum_1^N y_i$, $\hat{f}_j \equiv 0 \; \forall \; i, .j$;

2. Schleife:

for $j = 1, \ldots, p$ **do**

$$\hat{f}_j \leftarrow S_j \left[\{ y_i - \hat{\alpha} - \textstyle\sum_{k \neq j} \hat{f}_k(x_{ik}) \}_1^N \right] ;$$

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N} \sum_{i=1}^{N} \hat{f}_j(x_{ij}) ;$$

end

Wiederhole Schritt 2 so lange, bis sich die Funktionen $\hat{f}_j$ nur noch marginal ändert.

Standardkonvention : $\sum_1^N f_j(x_j) = 0 \; \forall \; j \rightarrow \hat{\alpha} = ave(y_i)$.

Beispiel

Betrachte: 256 Testdaten von Company1.
Implementierung des Additiven Modells mit Hilfe des Packages `pyGAM` in Python.

- Input: i_1 und i_6
- Output: o_1

Beispiel: Ergebnis

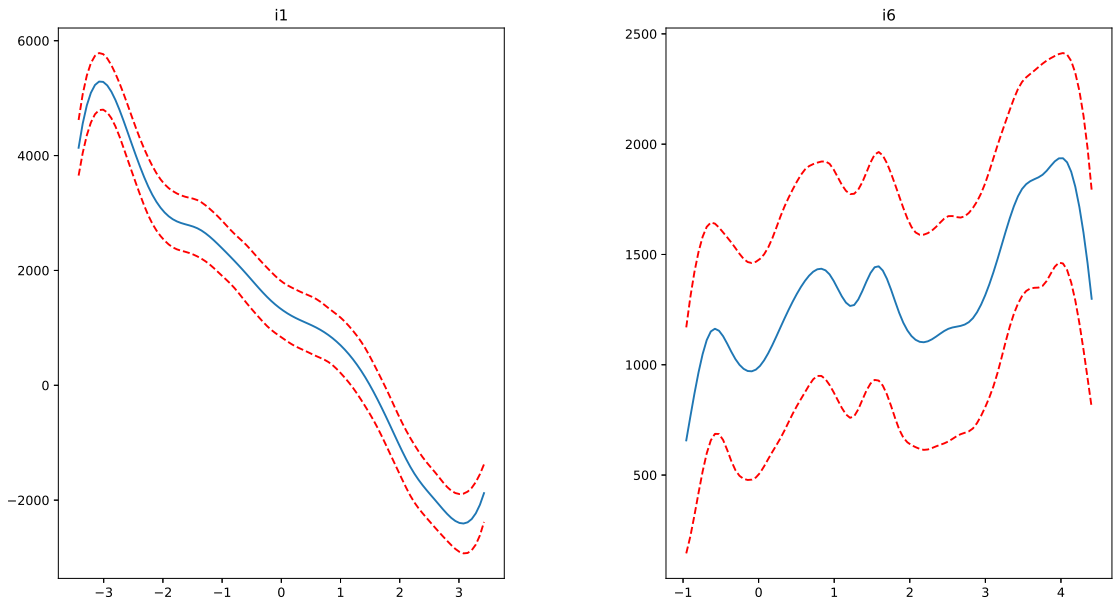


Abbildung: Funktionen $\hat{f}_1$ und f_6 zu den Inputs i_1 und i_6 und zum Output o_1 (blau) mit 95%-Konfidenzintervall (rot).

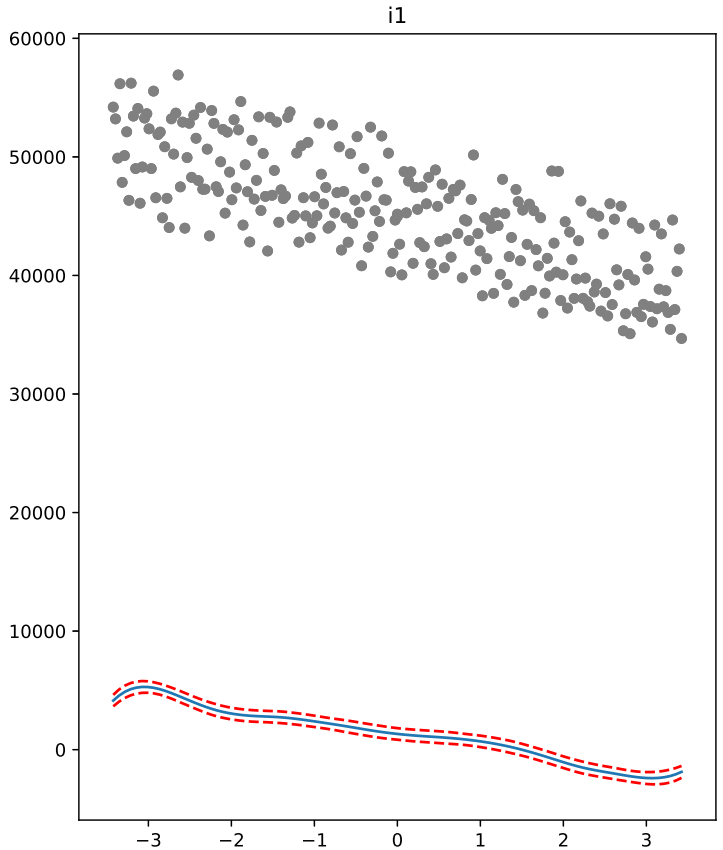


Abbildung: Streudiagramm der einzelnen Werte (graue Punkte) von i_1 und Funktion $\hat{f}_1$ (blau) mit 95%-Konfidenzintervall (rot).

Baumbasierte Methoden

Durchführung:

1. (Rekursives, binäres) Aufteilen des Featureraumes in Vierecke (Regionen).
2. Fitten eines einfachen Modells.

Regressionsmodell:

$$f(x) = \sum_{i=1}^{M} c_m \mathrm{I}(x \in R_m),$$

mit Regionen $R_1, \ldots, R_M$ und Konstante c_m.

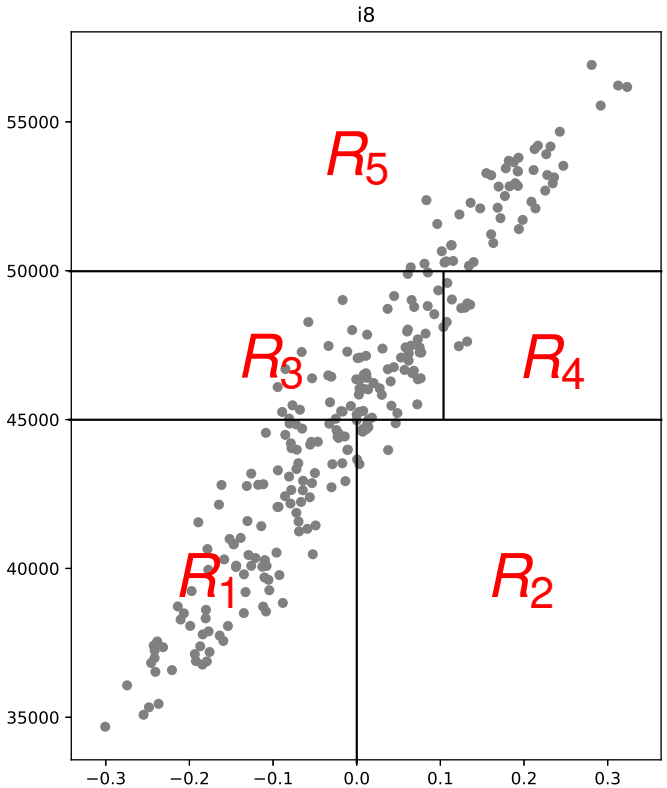
Beispiel

Betrachte: 256 Testdaten von Company1.

- Input: i_8
- Output: o_1

Ziel von Company1:

1. $o_1 \geq 45.000$
2. Minimieren von i_8 ($i_8 \leq 0$)
3. Wenn $o_1 \geq 45.000$, dann sind Werte bis $i_8 = 0,1$ akzeptabel
4. Wenn sogar $o_1 \geq 50.000$, dann sind auch höhere Werte für i_8 möglich



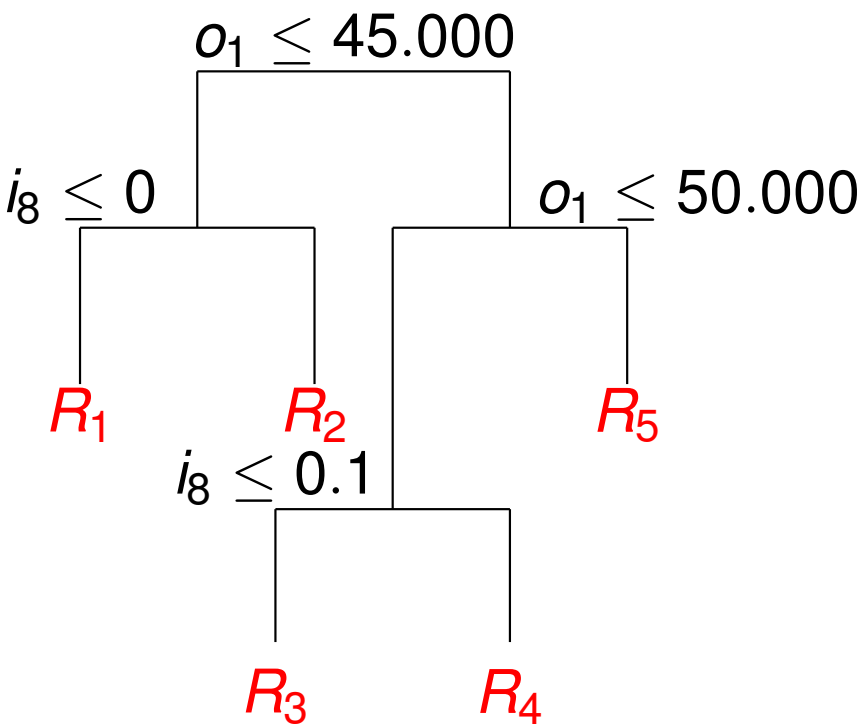


Abbildung: Aufteilung des Streudiagrammraumes in Regionen $R_1, \ldots R_5$ (links) und dazugehöriger Binärbaum (rechts)

Sortieren der Regionen: $R_3 \geq R_5 \geq R_1 \geq R_4 \geq R_2$.

$\rightarrow$ Wählen von Konstanten:
$c_1 = 3,\ c_2 = 1,\ c_3 = 5,\ c_4 = 2, c_5 = 4$.

$\rightarrow$ Aufstellen des Regressionsmodells:

$$\hat{f}(X) = 3\mathrm{I}\{(i_8, o_1) \in R_1\} + 1\mathrm{I}\{(i_8, o_1) \in R_2\} + 5\mathrm{I}\{(i_8, o_1) \in R_3\} + 2\mathrm{I}\{(i_8, o_1) \in R_4\} + 4\mathrm{I}\{(i_8, o_1) \in R_5\}.$$

Erstellen Regressionsbaum

Daten:

- p Inputs
- Anwort Y
- N Beobachtungen
  $\rightarrow$ Wert zur N-ten Beobachtung: (x_i, y_i),
  für $i = 1, \ldots, N$ mit $x_i = (x_{i1}, \ldots, x_{ip})$

Allgemeines Modell:

$$f(x) = \sum_{i=1}^{M} c_m \mathrm{I}(x \in R_m),$$

mit Regionen $R_m,\ m = 1, \ldots, M$.

Ziel: Minimieren der Summe der Quadrate $\sum(y_i - f(x_i))^2$.
$\rightarrow$ Beste Schätzung für c_m:

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m).$$

Unterteilung des Featureraumes durch Greedy-Algorithmus:

Wähle Splittingvariable j und Splitpunkt s.
Definiere Halbebenen:

$$R_1(j, s) = \{X | X_j \leq s\} \text{ und } R_2(j, s) = \{X | X_j > s\}.$$

Suche Splittingvarible j und Splitpunkt s, die

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \qquad (1)$$

minimieren.

$\rightarrow$ Innere Minima:

$$\hat{c}_1=\text{ave}(y_i|x_i\in R_1(j,s)) \text{ und } \hat{c}_2=\text{ave}(y_i|x_i\in R_2(j,s))$$

Lösung von (1) gefunden $\rightarrow$ Teilen des Raums in zwei Regionen.
Wiederholen des Prozesses bis Abbruchbedingung einsetzt.

Cost-Complexity Pruning

Abbruchbedingung bestimmen mit *Cost-Complexity Pruning*:
Definiere Unterbaum $T \subset T_0$, der durch das Prunen von T_0 entstehen kann.
Endknoten von T haben Indize m und repräsentieren die Regionen.

Setze:

$$N_m = \#\{x_i \in R_m\},$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i,$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2.$$

Cost complexitiy Kriterium:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$
$$= \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2 + \alpha|T|.$$

Ziel: Finde für jedes α einen Unterbaum $T_\alpha \subset T_0$, der $C_\alpha(T)$ minimiert.

Methode: *weakest link pruning*: Entferne den Knoten aus dem Baum, der das kleinste Wachstum pro Knoten in $\sum_m N_m Q_m(T)$ bewirkt.
Führe dies solange dich, bis nur noch ein Knoten übrig ist.

Ausgabe: Reihe von Unterbäumen, die T_α enthalten muss.

Vielen Dank für Ihre Aufmerksamkeit!

„Verallgemeinerte additive Modelle“ und „Baumbasierte Methoden“

Helena Schmitz

Mathematisches Institut Köln

21. Mai 2019