



Vorlesung Maschinelles Lernen

Additive Modelle

Katharina Morik

LS 8 Künstliche Intelligenz Fakultät für Informatik
Technische Universität Dortmund

4.11.2008



Gliederung

- 1 Baumlerner
 - Merkmalsauswahl
 - Implementierung
 - Gütemaße und Fehlerabschätzung
- 2 Basisexpansionen und Strafterm
 - Stückweise Funktionen
 - Glätten
- 3 Generelle Additive Modelle



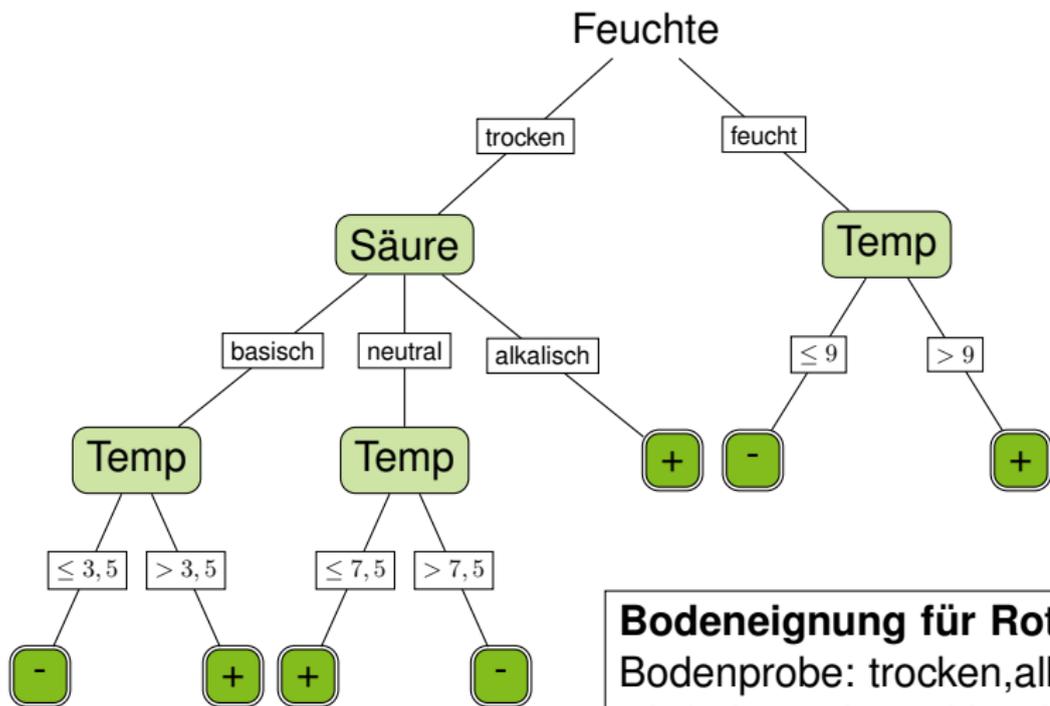
Aufteilen der Beispiele und Modellierung jeder Region

Von globalen zu lokalen Modellen:

- Lineare Modelle können als Vorverarbeitung Basisfunktionen für einzelne Merkmale verwenden.
- Generelle additive Modelle passen die Merkmale einzeln an die Daten an.
- **Baumlerner** teilen den Merkmalsraum in Rechtecke auf und passen in jedem ein Modell an. Dabei wird die Wahl des Merkmals in der rekursiven Aufteilung automatisch bestimmt.
- kNN teilt den Raum der Beispiele bei einer Anfrage x in die Nachbarschaft von x und den Rest auf.



Klassifizieren mit Entscheidungsbäumen



Bodeneignung für Rotbuchen:
 Bodenprobe: trocken,alkalisch,7
 wird als geeignet klassifiziert (+)



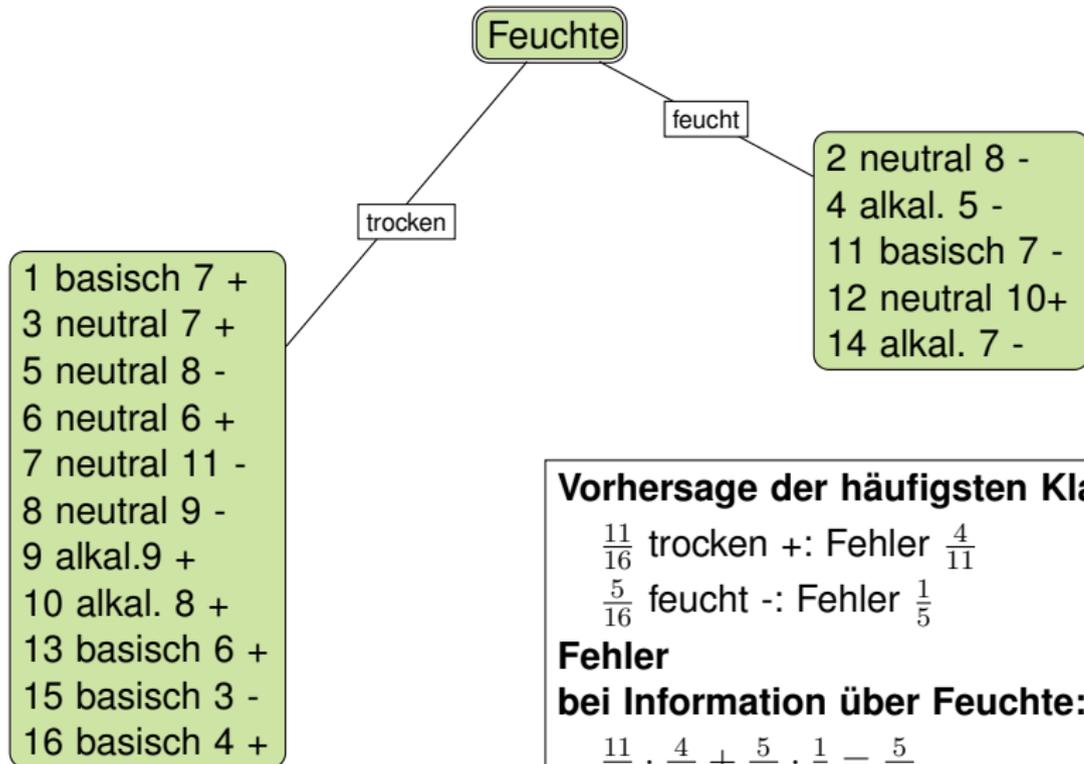
Lernen aus Beispielen

+				-			
ID	Feuchte	Säure	Temp	ID	Feuchte	Säure	Temp
1	trocken	basisch	7	2	feucht	neutral	8
3	trocken	neutral	7	4	feucht	alkal.	5
6	trocken	neutral	6	5	trocken	neutral	8
9	trocken	alkal.	9	7	trocken	neutral	11
10	trocken	alkal.	8	8	trocken	neutral	9
12	feucht	neutral	10	11	feucht	basisch	7
13	trocken	basisch	6	14	feucht	alkal.	7
16	trocken	basisch	4	15	trocken	basisch	3

Ohne weiteres Wissen können wir als Vorhersage immer - sagen. Der Fehler ist dann 8/16.



Aufteilen nach Bodenfeuchte



Vorhersage der häufigsten Klasse:

$\frac{11}{16}$ trocken +: Fehler $\frac{4}{11}$

$\frac{5}{16}$ feucht -: Fehler $\frac{1}{5}$

Fehler bei Information über Feuchte:

$\frac{11}{16} \cdot \frac{4}{11} + \frac{5}{16} \cdot \frac{1}{5} = \frac{5}{16}$



Bedingte Wahrscheinlichkeit

- Wahrscheinlichkeit, dass ein Beispiel zu einer Klasse gehört, gegeben der Merkmalswert

$$P(Y|X_j) = P(Y \cap X_j)/P(X_j)$$

- Annäherung der Wahrscheinlichkeit über die Häufigkeit
- Gewichtung bezüglich der Oberklasse
- Beispiel: $Y = \{+, -\}$, $X_j = \{feucht, trocken\}$

$P(+|feucht) = 1/5$, $P(-|feucht) = 4/5$ gewichtet mit $5/16$

$P(+|trocken) = 7/11$, $P(-|trocken) = 4/11$ gewichtet mit $11/16$

Wahl des Merkmals mit dem höchsten Wert (kleinsten Fehler)



Information eines Merkmals

- Wir betrachten ein Merkmal als Information.
- Wahrscheinlichkeit p_+ , dass das Beispiel der Klasse + entstammt. $I(p_+, p_-) = (-p_+ \log p_+) + (-p_- \log p_-)$
Entropie
- Ein Merkmal X_j mit k Werten teilt eine Menge von Beispielen \mathbf{X} in k Untermengen $\mathbf{X}_1, \dots, \mathbf{X}_k$ auf. Für jede dieser Mengen berechnen wir die Entropie.

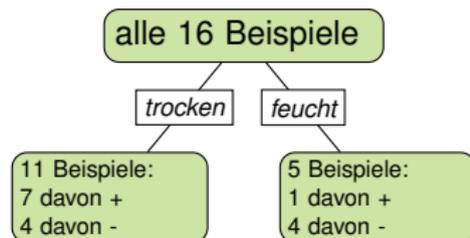
$$\text{Information}(X_j, \mathbf{X}) := - \sum_{i=1}^k \frac{|\mathbf{X}_i|}{|\mathbf{X}|} I(p_+, p_-)$$

- Der **Informationsgewinn** ist die Differenz zwischen der Entropie der Beispiele mit und ohne die Aufteilung durch X_j .

Feuchte

Güte des Attributs Feuchte mit den 2 Werten *trocken* und *feucht*:

$$\begin{aligned}
 & - \left[\underbrace{\frac{11}{16} \cdot I(+, -)}_{\text{trocken}} + \underbrace{\frac{5}{16} \cdot I(+, -)}_{\text{feucht}} \right] \\
 = & - \left[\underbrace{\frac{11}{16} \cdot \left(-\frac{7}{11} \cdot \log \left(\frac{7}{11} \right) - \frac{4}{11} \cdot \log \left(\frac{4}{11} \right) \right)}_{\text{trocken}} \right. \\
 & \left. + \underbrace{\frac{5}{16} \cdot \left(-\frac{1}{5} \cdot \log \left(\frac{1}{5} \right) - \frac{4}{5} \cdot \log \left(\frac{4}{5} \right) \right)}_{\text{feucht}} \right] = -0,27
 \end{aligned}$$





Säure

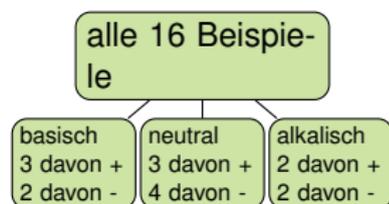
Güte des Attributs Säure mit den 3
Werten basisch, neutral und alkalisch:

$$- \left(\underbrace{\frac{5}{16} \cdot I(+, -)}_{\text{basisch}} + \underbrace{\frac{7}{16} \cdot I(+, -)}_{\text{neutral}} + \underbrace{\frac{4}{16} \cdot I(+, -)}_{\text{alkalisch}} \right) = -0,3$$

$$\text{basisch} \quad -\frac{3}{5} \cdot \log\left(\frac{3}{5}\right) + -\frac{2}{5} \cdot \log\left(\frac{2}{5}\right)$$

$$\text{neutral} \quad -\frac{3}{7} \cdot \log\left(\frac{3}{7}\right) + -\frac{4}{7} \cdot \log\left(\frac{4}{7}\right)$$

$$\text{alkalisch} \quad -\frac{2}{4} \cdot \log\left(\frac{2}{4}\right) + -\frac{2}{4} \cdot \log\left(\frac{2}{4}\right)$$





Temperatur

- Numerische Merkmalswerte werden nach Schwellwerten eingeteilt.
 - 9 verschiedene Werte in der Beispielmenge, also 8 Möglichkeiten zu trennen.
 - Wert mit der kleinsten Fehlerrate bei Vorhersage der Mehrheitsklasse liegt bei 7.
 - 5 Beispiele mit $\text{Temp} < 7$, davon 3 in +, 11 Beispiele $\text{Temp} \geq 7$, davon 6 in -.
- Die Güte der Temperatur als Merkmal ist $-0,29$.

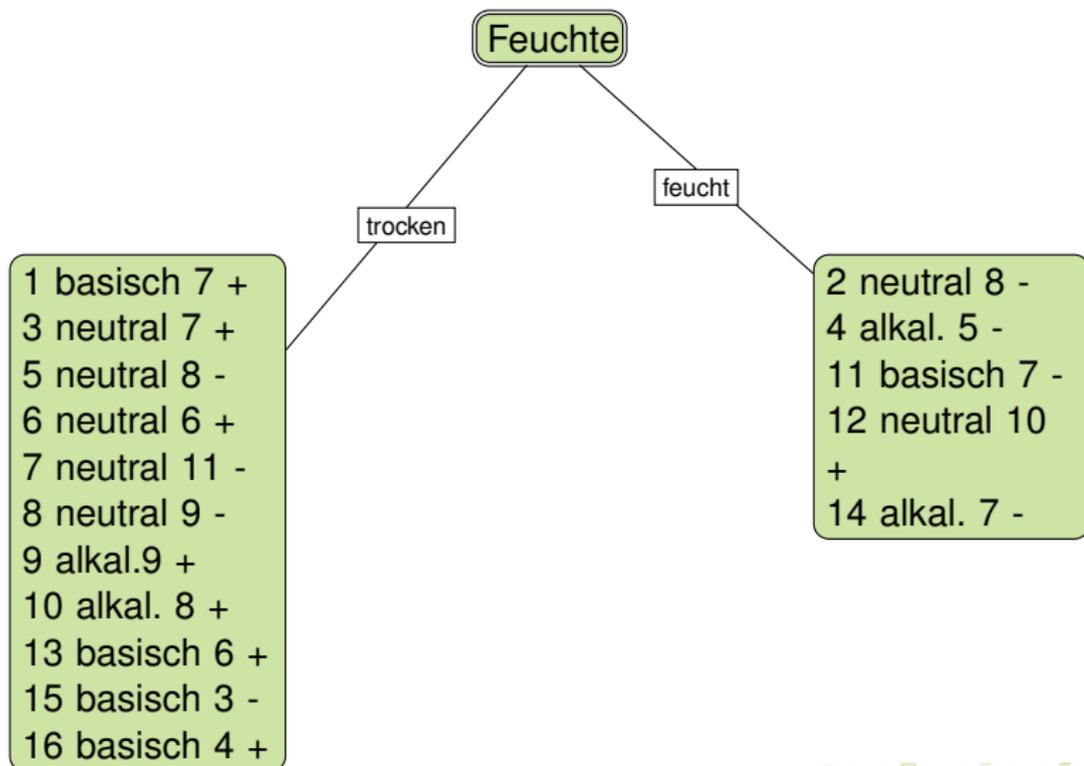


Merkmalsauswahl

- Gewählt wird das Merkmal X_j , dessen Werte am besten in (Unter-)mengen X_i aufteilen, die geordnet sind.
- Das Gütekriterium **Information** (Entropie) bestimmt die Ordnung der Mengen.
- Im Beispiel hat *Feuchte* den höchsten Gütewert.

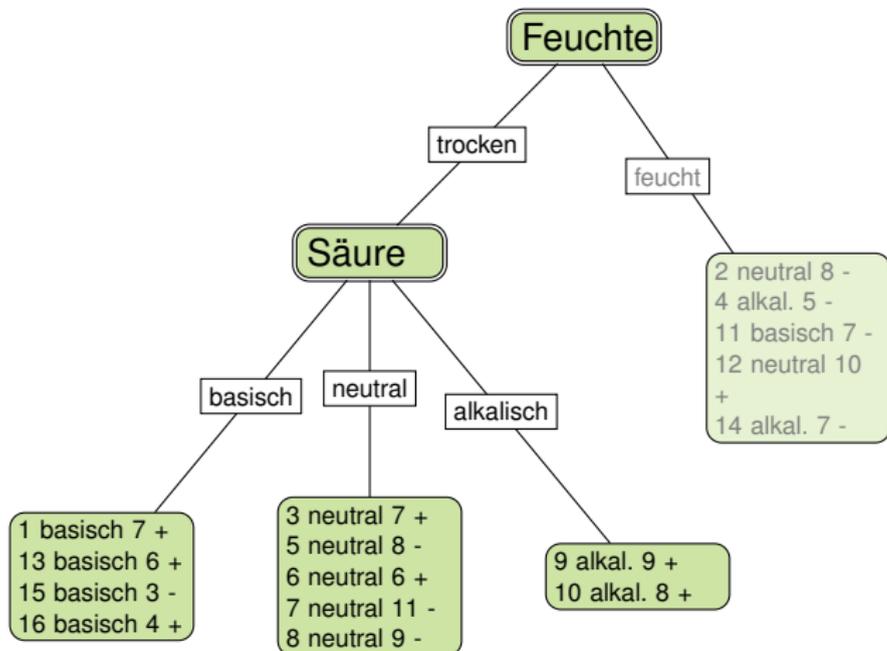


Algorithmus Top Down Induction of Decision Trees (TDIDT, hier: ID3) am Beispiel



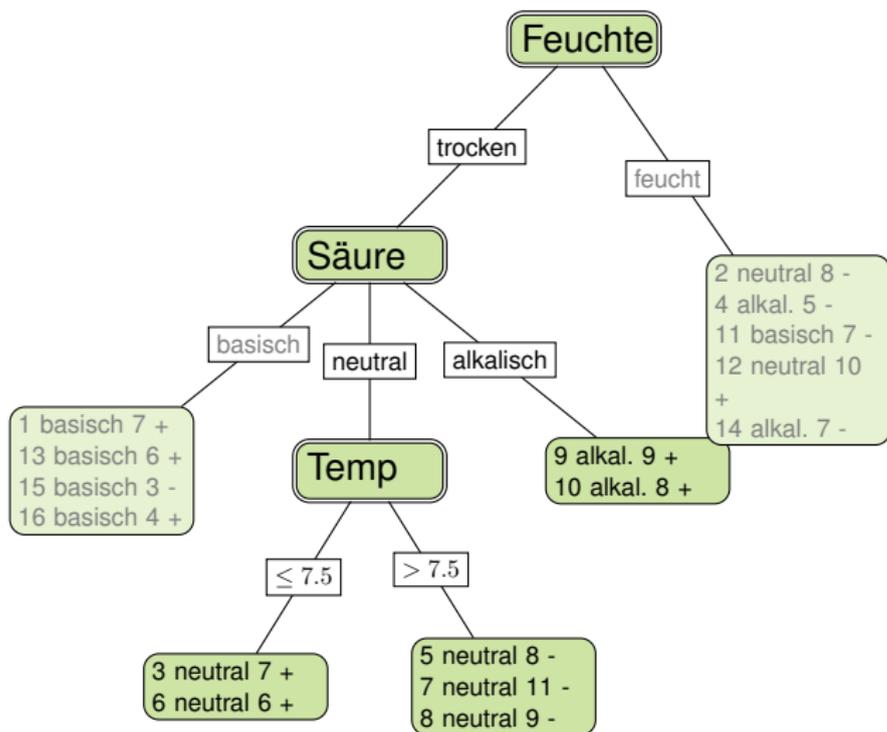


Algorithmus TDIDT (ID3) am Beispiel



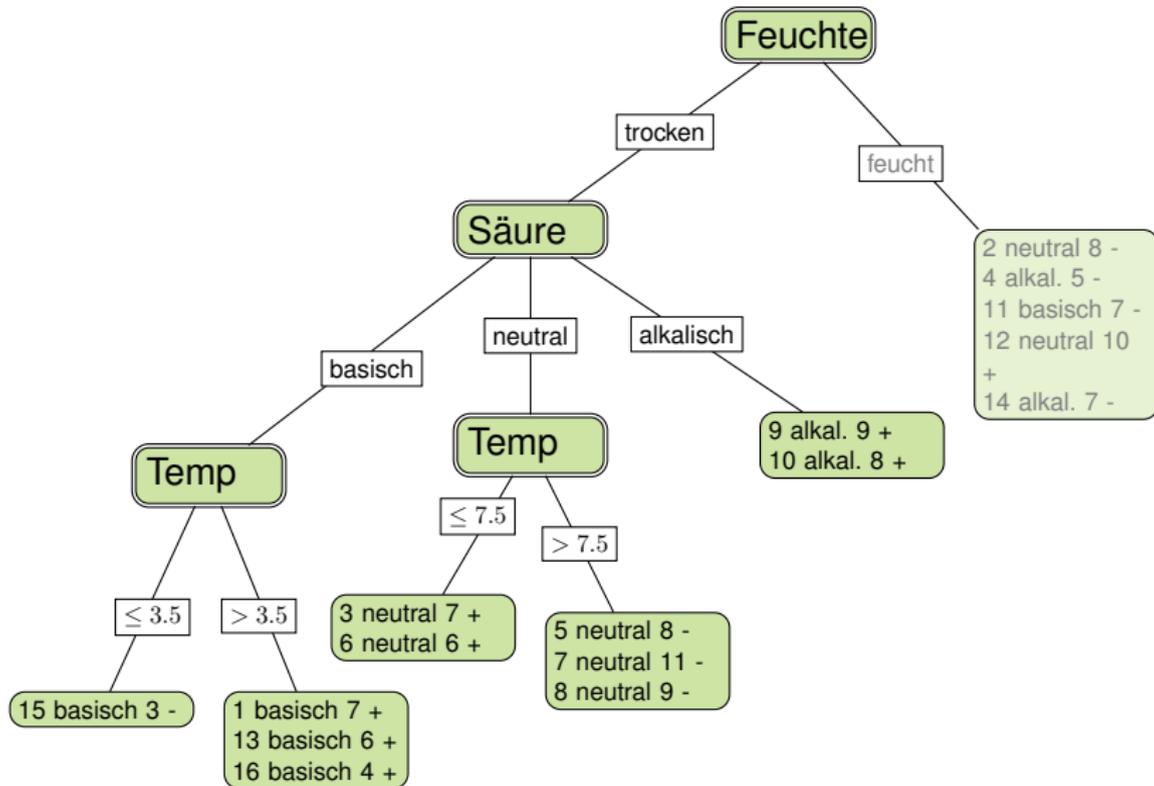


Algorithmus TDIDT (ID3) am Beispiel





Algorithmus TDIDT (ID3) am Beispiel





Algorithmus ID3 (TDIDT)

Rekursive Aufteilung der Beispielmenge nach Merkmalsauswahl:

- 1 $TDIDT(\mathbf{X}, \{X_1, \dots, X_p\})$
- 2 \mathbf{X} enthält nur Beispiele einer Klasse \rightarrow fertig
- 3 \mathbf{X} enthält Beispiele verschiedener Klassen:
 - $Güte(X_1, \dots, X_p, \mathbf{X})$
 - Wahl des besten Merkmals X_j mit k Werten
 - Aufteilung von \mathbf{X} in $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k$
 - für $i = 1, \dots, k$:
 $TDIDT(\mathbf{X}_i, \{X_1, \dots, X_p\} \setminus X_j)$
 - Resultat ist aktueller Knoten mit den Teilbäumen T_1, \dots, T_k



Komplexität TDIDT ohne Pruning

Rekursive Aufteilung der Beispielmenge nach Merkmalsauswahl:

- Bei p (nicht-numerischen) Merkmalen und N Beispielen ist die Komplexität $\mathcal{O}(pN \log N)$
 - Die Tiefe des Baums sei in $\mathcal{O}(\log N)$.
 - $\mathcal{O}(N \log N)$ alle Beispiele müssen “in die Tiefe verteilt” werden, also: $\mathcal{O}(N \log N)$ für ein Merkmal.
 - p mal bei p Merkmalen!



Was muss man implementieren?

```
import com.rapidminer.example.Attribute;  
import com.rapidminer.example.ExampleSet;  
split(ExampleSet exampleSet, Attribute attribute)
```

- - Die Beispielmenge gemäß der Attributwerte aufteilen.
 - Das Attribut auswählen, das zur Partitionierung einer Beispielmenge genutzt wird.
 - Information (Entropie) für alle Attribute berechnen.
 - Bei numerischen Attributen den numerischen Wert bestimmen, der die Beispiele am besten aufteilt.



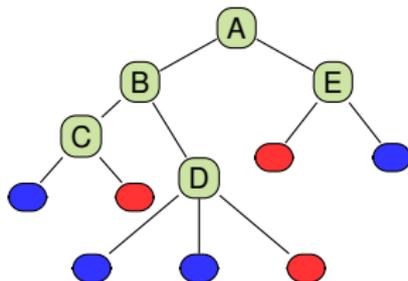
Implementieren in RapidMiner

- **X**: ExampleSet mit den Methoden u.a.
 - `size()` – gibt die Anzahl der Beispiele zurück
 - `getAttributes()` – liefert die Attribute zurück, über `getAttributes().size()` läßt sich die Anzahl ermitteln
 - `iterator()` – liefert einen Iterator über die Beispiele
- \vec{x}_i : ein Beispiel (Example) mit den Methoden u.a.
 - `getValue(a)` – gibt den Wert des Attributs a
 - Mit `getAttributes().iterator()` läßt sich über die Attribute eines Examples iterieren
- X_j : Methoden für Werte nominaler Merkmale :
 - Nominale Merkmale werden durch ein Mapping von double-Werten auf Strings realisiert. Für ein nominales Attribut liefert `getMapping()` das Mapping für dieses Attribut.
 - `getMapping().size()` liefert die Anzahl der unterschiedlichen Werte des Attributs
 - `getLabel()` – liefert den Wert des Zielmerkmals als double

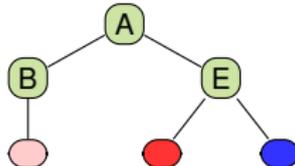


Stutzen

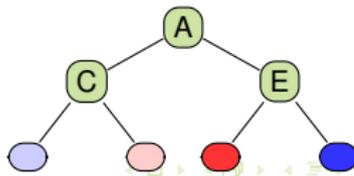
- Überanpassung des Baums an die Trainingsdaten verringern!
- Verständlichkeit erhöhen!
- Stutzen (Pruning):
 - a) Knoten an Stelle eines Teilbaums setzen
 - b) Einen Teilbaum eine Ebene höher ziehen
- Schätzen, wie sich der wahre Fehler beim Stutzen entwickelt.



- a) Knoten an Stelle eines Teilbaums setzen



- b) Einen Teilbaum eine Ebene höher ziehen





Stützen durch Fehlerschätzen

- Wenn der Fehler eines Knotens kleiner ist als die Summe der Fehler seiner Unterknoten, können die Unterknoten weggestutzt werden.
- Dazu müssen wir (bottom-up) die Fehler an allen Knoten schätzen.
- Obendrein sollten wir berücksichtigen, wie genau unsere Schätzung ist. Dazu bestimmen wir ein Konfidenzintervall.
- Wenn die obere Schranke der Konfidenz in den Fehler beim oberen Knoten kleiner ist als bei allen Unterknoten zusammen, werden die Unterknoten gestutzt.



Was ist ein Konfidenzintervall?

Konfidenzintervall

Vorgegeben eine tolerierte Irrtumswahrscheinlichkeit α , gibt das Konfidenzintervall

$$P(u \leq X \leq o) = 1 - \alpha$$

an, dass X mit der Wahrscheinlichkeit $1 - \alpha$ im Intervall $[u, o]$ liegt und mit der Wahrscheinlichkeit α nicht in $[u, o]$ liegt.

Meist wird das Konfidenzintervall für den Erwartungswert gebildet. Beispiel $\alpha = 0,1$: Mit 90% iger Wahrscheinlichkeit liegt der Mittelwert \bar{X} im Intervall $[u, o]$, nur 10% der Beobachtungen liefern einen Wert außerhalb des Intervalls.



z-Transformation in eine standard-normalverteilte Zufallsvariable

Die Zufallsvariable X wird bezüglich ihres Mittelwerts \bar{X} standardisiert unter der Annahme einer Normalverteilung:

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{N}}} \sim \mathcal{N}(0; 1)$$

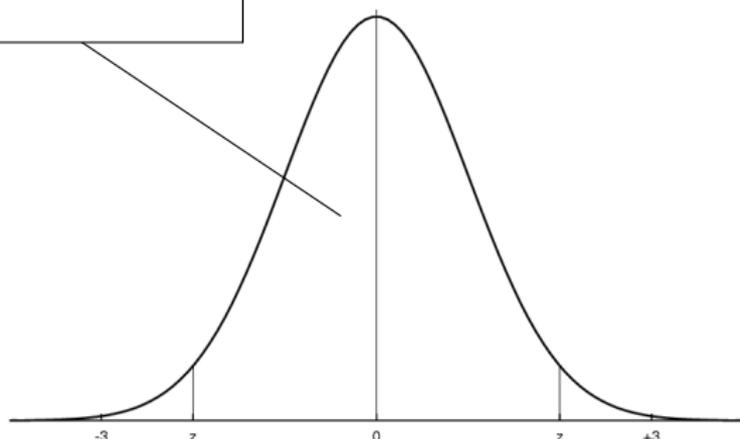
Die Wahrscheinlichkeit dafür, dass der Mittelwert im Intervall liegt, ist nun:

$$P\left(-z\left(1 - \frac{\alpha}{2}\right) \leq \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{N}}} \leq z\left(1 - \frac{\alpha}{2}\right)\right) = 1 - \alpha$$



Verteilung mit z-Werten

Fläche unter der Glocke in
 $[-z, z] = c$



- $P(-z \leq X \leq z) = 1 - \alpha$ Konfidenzniveau
Wahrscheinlichkeit, dass X mit Mittelwert 0 im Intervall der Breite $2z$ liegt ist $1 - \alpha$.
- z kann nachgeschlagen werden (z.B. Bronstein), wobei wegen Symmetrie nur angegeben ist: $P(X \geq z)$



Rechnung für reellwertige Beobachtungen und Mittelwert

Wir wollen ein bestimmtes Konfidenzniveau erreichen, z.B. 0,8.

- $P(X \geq -z) P(X \leq z)$ ist dann $(1 - 0,8)/2 = 0,1$.
- Der z -Wert, für den die Fläche der Glockenkurve zwischen $-z$ und z genau $1 - \alpha = 0,8$ beträgt, ist das $(1 - \frac{\alpha}{2})$ -Quantil der Standardnormalverteilung, hier: 1,28 (nachschielen).
- Das standardisierte Stichprobenmittel liegt mit der Wahrscheinlichkeit 0,8 zwischen -1,28 und +1,28.

$$\begin{aligned} 0,8 &= P(-1,28 \leq \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{N}}} \leq 1,28) \\ &= P(-1,28 \frac{\sigma}{\sqrt{N}} \leq \bar{X} - \mu \leq 1,28 \frac{\sigma}{\sqrt{N}}) \\ &= P(\bar{X} - 1,28 \frac{\sigma}{\sqrt{N}} \leq \mu \leq \bar{X} + 1,28 \frac{\sigma}{\sqrt{N}}) \end{aligned}$$

Das Intervall ist $[\bar{X} - 1,28 \frac{\sigma}{\sqrt{N}}; \bar{X} + 1,28 \frac{\sigma}{\sqrt{N}}]$.



Fehler oder Erfolg schätzen

- Bei den Entscheidungsbäumen beobachten wir nur zwei Werte $Y \in \{+, -\}$.
- Wir haben eine Binomialverteilung mit wahrer Wahrscheinlichkeit p_+ für $y = +$ (Erfolg).
- Beobachtung der Häufigkeit f_+ bei N Versuchen.

Varianz:

$$\sigma^2 = \frac{f_+(1 - f_+)}{N}$$

Erwartungswert:

$$E(p_+) = f_+/N$$

- In das allgemeine Konfidenzintervall $[\bar{X} - z(1 - \alpha/2)\frac{\sigma}{\sqrt{N}}; \bar{X} + z(1 - \alpha/2)\frac{\sigma}{\sqrt{N}}]$ setzen wir diese Varianz ein und erhalten:

$$\left[f_+ - z(1 - \alpha/2)\frac{\sqrt{f_+(1 - f_+)}}{N}; f_+ + z(1 - \alpha/2)\frac{\sqrt{f_+(1 - f_+)}}{N} \right]$$



Konfidenz bei Binomialverteilung

Allgemein berechnet man die obere und untere Schranke der Konfidenz bei einer Binomialverteilung für ein Bernoulli-Experiment:

$$p_+ = \frac{f_+ + \frac{z^2}{2N} \pm z \sqrt{\frac{f_+}{N} - \frac{f_+^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Hierzu muss lediglich die Häufigkeit f_+ gezählt werden, N , z bekannt sein.

Diese Abschätzung für den Erfolg können wir symmetrisch für den Fehler (p_-) durchführen.



Anwendung zum Stutzen

- Für jeden Knoten nehmen wir die obere Schranke (pessimistisch):

$$p_- = \frac{f_- + \frac{z^2}{2N} + z \sqrt{\frac{f_-}{N} - \frac{f_-^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

- Wenn der Schätzfehler eines Knotens kleiner ist als die Kombination der Schätzfehler seiner Unterknoten, werden die Unterknoten weggestutzt. Die Kombination wird gewichtet mit der Anzahl der subsumierten Beispiele.



Gütemaße

- Konfusionsmatrix:

tatsächlich	Vorhergesagt +	Vorhergesagt −	
+	True positives TP	False negatives FN	Recall: $TP/(TP + FN)$
−	False positives FP	True negatives TN	
	Precision: $TP/(TP + FP)$		

- Accuracy: $P(\hat{f}(x) = y)$ geschätzt als $(TP + TN)/total$

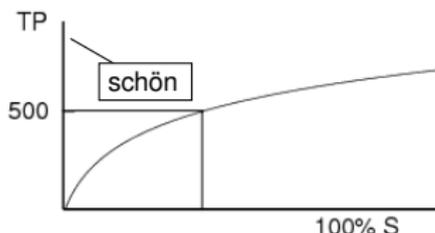


Balance von FP und FN

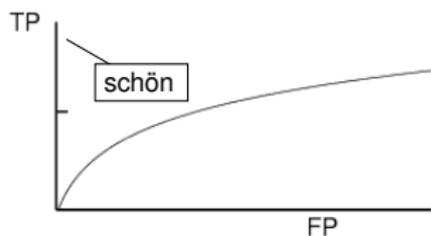
- F-measure: $\frac{\beta \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = \frac{\beta TP}{\beta TP + FP + FN}$

- Verlaufsformen:

- Lift: TP für verschiedene Stichprobengrößen S



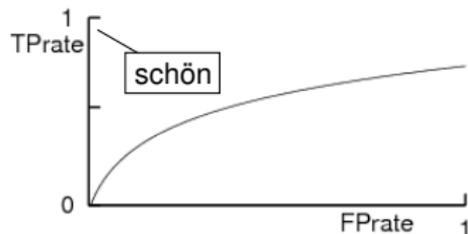
- Receiver Operating Characteristic (ROC): für verschiedene TP jeweils die FP anzeigen





ROC genauer

- Statt der absoluten Anzahl TP nimm die Raten von true oder false positives – ergibt eine glatte Kurve.
 - Für jeden Prozentsatz von falschen Positiven nimm eine Hypothese h , deren Extension diese Anzahl von FP hat und zähle die TP .
 - $TP_{rate} := TP/P \sim recall$ bezogen auf eine Untermenge
 - $FP_{rate} := FP/N \sim FP/FP + TN$ bezogen auf Untermenge





Kosten von Fehlern

- Nicht immer sind FP so schlimm wie FN
 - medizinische Anwendungen: lieber ein Alarm zu viel als einen zu wenig!
- Gewichtung der Beispiele:
 - Wenn FN 3x so schlimm ist wie FP, dann gewichte negative Beispiele 3x höher als positive.
 - Wenn FP 10x so schlimm ist wie FN, dann gewichte positive Beispiele 10x höher als negative.
- Lerne den Klassifikator mit den gewichteten Beispielen wie üblich. So kann jeder Lerner Kosten berücksichtigen!



Was wissen Sie jetzt?

- Sie kennen den Algorithmus ID3 als Beispiel für TDIDT.
- Für das Lernen verwendet ID3 das Gütemaß des Informationsgewinns auf Basis der Entropie.
- Man kann abschätzen, wie nah das Lernergebnis der unbekanntenen Wahrheit kommt → Konfidenz
- Man kann abschätzen, wie groß der Fehler sein wird und dies zum Stutzen des gelernten Baums nutzen.
- Lernergebnisse werden evaluiert:
 - Einzelwerte: accuracy, precision, recall, F-measure
 - Verläufe: Lift, ROC

Diese Evaluationsmethoden gelten nicht nur für Entscheidungs bäume!



Ausgangspunkt: Funktionsapproximation

- Die bisher vorgestellten Lernverfahren, sind Instanzen der Funktionsapproximation.
- Gegeben sind die Trainingsbeispiele \mathcal{T} , gesucht ist eine Funktion

$$f_{\theta}(x) = \sum_{m=1}^M h_m(x)\theta_m$$

- Dabei gibt es Parameter θ , die abzuschätzen sind, bei den linearen Modellen ist dies $\hat{\beta}$.
- Darüber hinaus können die Daten durch Basisfunktionen in einen Raum transformiert werden, der für das Lernen besser geeignet ist: $h_m(x)$.
- Jetzt gehen wir auf $h_m(X) : \mathcal{R}^p \rightarrow \mathcal{R}$ ein.



Einfachste Basisfunktion: Stückweise Konstant

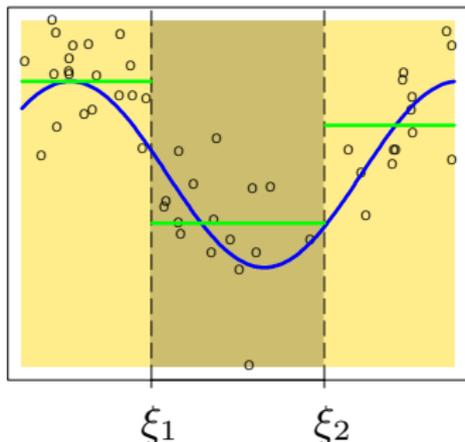
Einteilung von X in Intervalle durch

$$h_1(X) = I(X < \xi_1), h_2(X) = I(\xi_1 \leq X < \xi_2),$$

$$h_3(X) = I(\xi_2 \leq X).$$

Als lineares Modell ergibt sich der Durchschnitt von Y im jeweiligen Intervall: $f(X) = \sum_{m=1}^3 \hat{\beta}_m h_m(X)$

Piecewise Constant



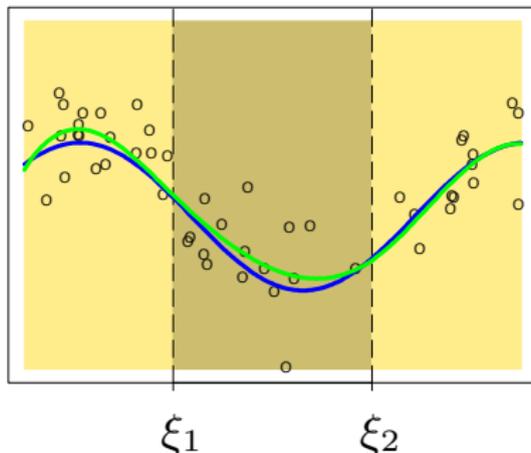


Stückweise kubisches Polynom

Kontinuierliche, differenzierbare Funktionen (1. und 2. Ableitung) ergeben glattere Annäherung:

$$h_1(X) = X^0, h_3(X) = X^2, h_5(X) = (X - \xi_1)_+^3$$
$$h_2(X) = X^1, h_4(X) = X^3, h_6(X) = (X - \xi_2)_+^3$$

Continuous Second Derivative





Kubische Splines und Verallgemeinerung

- Für ein Polynom 3. Grades (Ordnung $M = 4$) brauchen wir 4 Basisfunktionen h_i .
- Dazu kommen Basisfunktionen für die Stützstellen. Beim kubischen Polynom hatten wir $K = 2$ Stützstellen ξ mit jeweils einer kubischen Funktion $h_i(X)$.
- Allgemein haben die polynomielle Basisfunktionen die Form

$$\begin{aligned}h_j(X) &= X^{j-1}, j = 1, \dots, M \\h_{M+l}(X) &= (X - \xi_l)_+^{M-1}, l = 1, \dots, K\end{aligned}$$

- Polynomielle Basisfunktionen heißen **Splines**.



Regression Splines

- Funktionen, die sich an Werte in vorgegebenen Intervallen anpassen, heißen **Regression Splines**.
- Die Anzahl und Lage der Stützstellen ξ_i muss vorgegeben werden.
- Die Funktionen weichen jenseits der Stützstellen sehr vom wahren Wert ab.
- Verbesserung: **natürliche Splines**, bei denen jede Funktion jenseits der Intervallgrenzen als linear angenommen wird.



Natürliche kubische Splines

- Das Modell mit kubischem Spline:

$$f(X) = \sum_{j=0}^3 \beta_j X^j + \sum_{k=1}^K \theta_k (X - \xi_k)_+^3$$

- Die Bedingung der Linearität bedeutet: jenseits der Intervallgrenzen darf nur X^1 betrachtet werden. Dies impliziert Beschränkungen (constraints):

$$\begin{aligned} \beta_2 &= 0, & \beta_3 &= 0 \\ \sum_{k=1}^K \theta_k &= 0, & \sum_{k=1}^K \xi_k \theta_k &= 0 \end{aligned}$$

- Dadurch reduziert sich die Menge der Basisfunktionen.



Basisfunktionen der natürlichen kubischen Splines

Der natürliche kubische Spline mit K Stützstellen ist durch K Basisfunktionen gegeben.

$$N_1(X) = X^0,$$

$$N_2(X) = X^1,$$

$$N_{k+2}(X) = d_k(X) - d_{K-1}(X), \quad k = 1, \dots, K$$

$$d_k(X) = \frac{(X - \xi_k)_+^3 - (X - \xi_K)_+^3}{\xi_K - \xi_k}$$



Glätten erfordert keine Wahl und Platzierung der Trennungen

- Natürliche kubische Splines mit allen Beispielen $x_i, i = 1, \dots, N$ als Trennungen hätten zu viele Freiheitsgrade zu bestimmen.
- Mit einem Strafterm für die Krümmung wird aber die Komplexität begrenzt.
- Wir minimieren

$$RSS(f, \lambda) = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt \quad (1)$$

λ gewichtet den Strafterm: $\lambda = 0$ erlaubt alle Funktionen, $\lambda = \infty$ erlaubt nur noch das lineare Modell mit kleinstem RSS – also gar keine Basisfunktionen.



Optimierungsproblem mit Glättung

$$\hat{f}(x) = \sum_{j=1}^N N_j(x) \hat{\theta}_j$$

wobei $N_j(x)$ eine Menge von N Basisfunktionen für das Beispiel x ist. Es gibt ein eindeutiges Optimum für natürliche kubische Splines mit allen x_i als Trennstellen. Wir erhalten eine $N \times N$ -Matrix: eine Zeile je Beispiel; da jetzt $K = N$ ist, eine Spalte je Basisfunktion.

$$\mathbf{N} = \begin{pmatrix} N_1(x_1) & N_2(x_1) & \dots & N_N(x_1) \\ \dots & \dots & \dots & \dots \\ N_1(x_i) & \dots & \dots & N_N(x_i) \\ \dots & \dots & \dots & \dots \\ N_1(x_N) & \dots & \dots & N_N(x_N) \end{pmatrix}$$

$RSS(f, \lambda)$ soll minimiert werden.



Lösung des Optimierungsproblems mit Glättung

Das Qualitätskriterium (Gleichung 1)

$$RSS(f, \lambda) = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

lässt sich vereinfachen zu

$$RSS(\theta, \lambda) = (\mathbf{y} - \mathbf{N}\theta)^T (\mathbf{y} - \mathbf{N}\theta) + \lambda \theta^T \mathbf{\Omega}_N \theta \quad (2)$$

wobei $\{\mathbf{N}\}_{ij} = N_j(x_i)$ und $\{\mathbf{\Omega}_N\}_{jk} = \int N_j''(t) N_k''(t) dt$

Die Lösung ist dann

$$\hat{\theta} = (\mathbf{N}^T \mathbf{N} + \lambda \mathbf{\Omega}_N)^{-1} \mathbf{N}^T \mathbf{y} \quad (3)$$



Beispiel

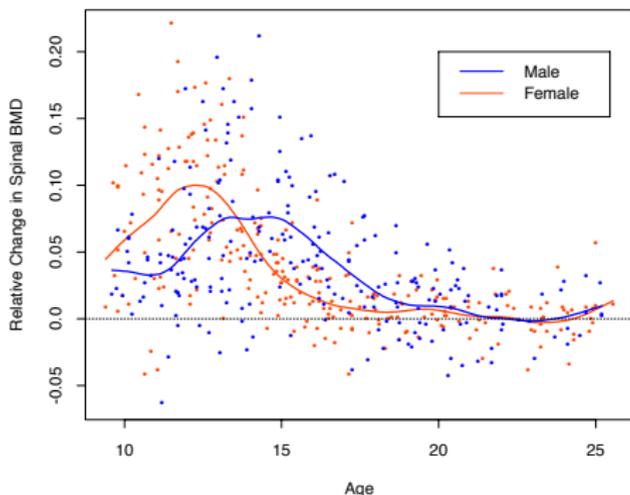


Figure 5.6: *The response is the relative change in bone mineral density measured at the spine in adolescents, as a function of age. A separate smoothing spline was fit to the males and females, with $\lambda \approx 0.00022$. This choice corresponds to about 12 degrees of freedom.*



Glättungsmatrix $S_{\lambda}y$

Eine Glättung mit vorher bestimmtem λ ist ein linearer Glättungsoperator.

$$\mathbf{S}_{\lambda}y = \hat{\mathbf{f}} = \mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\mathbf{\Omega}_N)^{-1}\mathbf{N}^T y \quad (4)$$

S_{λ} ist die Glättungsmatrix.

- S_{λ} ist eine symmetrische und semidefinite Matrix.
- S_{λ} hängt nur von x_i und λ ab.
- S_{λ} ist linear in y .
- Der Freiheitsgrad ist die Summe der Diagonalelemente von S_{λ} , bezeichnet $df_{\lambda} = \text{trace}(\mathbf{S}_{\lambda})$.



Was wissen Sie jetzt?

- Wir haben eine Methode gesehen, Nichtlinearität zu berücksichtigen. Die Daten werden durch Basisexpansionen umgeformt und erst danach linear modelliert.
- Insbesondere haben wir das kubische Polynom gesehen – noch höhere Exponenten ergeben für das menschliche Auge keine Verbesserung der Glättung.
- Die Fehlerminimierung mit Basisexpansion und Strafterm (Gleichungen (1) und (2)) ergibt bei fester Gewichtung λ des Strafterms eine **Glättungsmatrix S_λ** .



Generelle additive Modelle

- Lineare Modelle passen eine Hyperebene an alle Daten an. Die Hyperebene wird dann auf verschiedene Weisen zur Vorhersage genutzt.
- Basisfunktionen können Nichtlinearität ausdrücken: nach ihrer Anwendung wird dann mit einem linearen Modell vorhergesagt.
- Das Modell selbst kann aber auch nichtlinear sein! Die allgemeine Form genereller additiver Modelle für die Regression:

$$E(Y|X_1, X_2, \dots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p) \quad (5)$$

- Jedes f_i sei hier ein kubischer Spline.



Fehlerminimierung bei generellen additiven Modellen

Eben haben wir das Glätten jeweils für ein Merkmal bei der Funktionsapproximation gesehen mit der Fehlerminimierung beim Glätten einer Funktion (Gleichung 1):

$$RSS(f, \lambda) = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

Bei generellen additiven Modellen müssen wir parallel p Funktionen anpassen:

$$PRSS(\alpha, f_1, \dots, f_p) = \sum_{i=1}^N \left[y_i - \alpha - \sum_{j=1}^p f_j(x_{ij}) \right]^2 + \sum_{j=1}^p \lambda_j \int f_j''(t_j)^2 dt_j \quad (6)$$

Jede Funktion f_j ist ein natürlicher kubischer Spline für X_j mit Trennungen an den Werten $x_{ij}, i = 1, \dots, N$.



Annahmen für die Optimierung

Um eine eindeutige Lösung der Fehlerminimierung zu finden, nehmen wir an:

$$\forall j : \sum_{i=1}^N f_j(x_{ij}) = 0$$

Dann ist $\hat{\alpha} = \text{Mittelwert}(y_i)$.

Falls die $N \times N$ -Matrix der Beispiele nichtsingulär ist (invertierbar, die Determinante der Matrix ist $\det(\mathbf{N}) \neq 0$), hat Gleichung (6) eine eindeutige Lösung. Das Optimierungsproblem ist dann konvex.

Backfitting Verfahren ($\mathbf{X}, \mathbf{S}, \tau,$)

- 1 $\hat{\alpha} := \frac{1}{N} \sum_{i=1}^N y_i$; For $j=1$ until p do $stable_j := 0$;
- 2 Iterator j über allen Merkmalen $M \setminus Fertig$
 - If $stable_j > \tau$,
return \hat{f}_j ; $Fertig := Fertig \cup \hat{f}_j$; Goto 2;
 - For $i=1$ until N

$$\hat{f}_j := S_j \left[y_i - \hat{\alpha} - \sum_{k=1, k \neq j}^p \hat{f}_k(x_{ik}) \right]$$

% Bei Anpassung von \hat{f}_j alle anderen \hat{f}_k verwenden!

- If \hat{f}_j did not change, $stable_j++$;
- 3 If $M \neq \{\}$, Goto 2; else stop.



Was wissen Sie jetzt?

- Sie haben gesehen, dass auch das Modell selbst zusammengesetzt sein kann aus an die Beispiele angepassten Glättungsfunktionen.
- Solche Modelle heißen **additive Modelle**.
- Diese Modelle müssen die Glättungsfunktionen für alle Merkmale gleichzeitig anpassen.
- Sie haben den **Backfitting Algorithmus** dafür gesehen.
- Es gibt noch andere additive Modelle und deren Lernverfahren, z.B. additive logistische Regression.