

Data Mining und Maschinelles Lernen

Baumbasierte Verfahren



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Entscheidungsbäume sind auf den ersten Blick ein grundlegend anderer Ansatz zur Konstruktion von Klassifikationsregeln. Ihre Regeln sind meist sehr verständlich für den Anwender, können sich aber nicht sehr flexibel an die Daten anpassen. In R kann z.B. `rpart()` aus dem Paket `rpart` benutzt werden.

Basierend auf Folien von Katharina Morik, Uwe Ligges, Christoph Sawade, Niels Landwehr, Paul Prasse, Silvia Makowski, Tobias Scheffer und Johannes Fürnkranz.
Danke fürs Offenlegen der Folien.



Was wollen wir hier kennenlernen?

- ▶ Was sind Entscheidungsbäume?
- ▶ Was ist der Informationsgewinn von Tests?
- ▶ Wie lernen wir Entscheidungsbäume?
- ▶ Wie vermeiden wir “Wildwuchs” bei Entscheidungsbäumen?
- ▶ Gütekosten

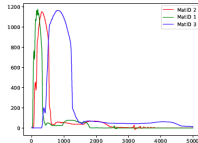
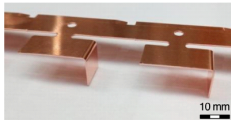
Aufteilen der Beispiele und Modellierung jeder Region



TECHNISCHE
UNIVERSITÄT
DARMSTADT

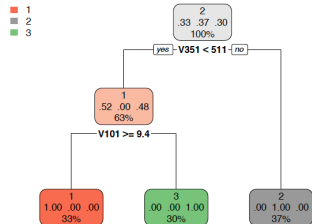
- ▶ Globale Modelle: **Lineare Modelle** passen die Parameter **eines Modells für den gesamten Merkmalsraum** der Beispiele an, der evtl. durch eine implizite Transformation (Kernfunktionen, dazu später mehr) oder explizite Transformationen (Vorverarbeitung) in einen Merkmalsraum überführt werden.
- ▶ Lokale Modelle: **k-nächste Nachbarn** (kNN) Verfahren, die wir auch schon kennengelernt haben, teilen den Raum der Beispiele **bei einer Anfrage x** in die Nachbarschaft von x und den Rest auf.
- ▶ Partitionierte Modelle: **Baumlerner** teilen den **gesamten Merkmalsraum** in Rechtecke auf und **passen in jedem ein Modell an**. Dabei wird die Wahl des Merkmals in der rekursiven Aufteilung automatisch bestimmt.

Ein Beispiel aus der Produktion



Stanzen von Löchern in Abhängigkeit des
Typs und der Dicke des Materials. (Links)
Produktionsteil. (Rechts) Sensorwerte.

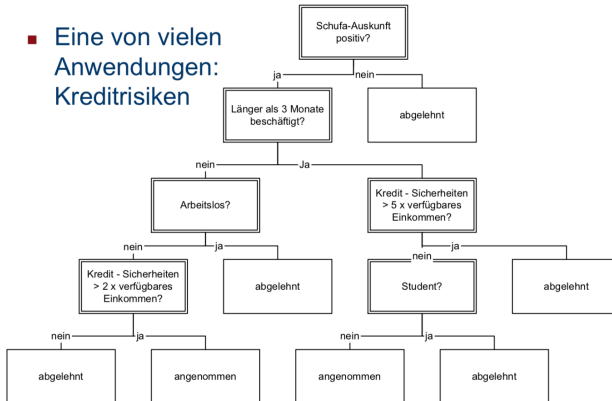
Danke an das Institut für Produktionstechnik und Umformmaschinen der TU Darmstadt für die Daten!



Vorhersage des Materials

Ein weiteres Beispiel aus der Kreditwirtschaft

- Eine von vielen Anwendungen: Kreditrisiken

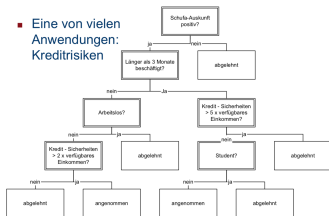


Warum Entscheidungsbäume?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Eine von vielen Anwendungen: Kreditrisiken



- ▶ Bäume sind (recht) einfach zu interpretieren
- ▶ Sie liefern direkt **Vorhersagen** und **Begründungen**: “Abgelehnt, weil weniger als 3 Monate beschäftigt und Kredit-Sicherheiten $< 2 \times$ verfügbares Einkommen”.

- ▶ Bäume können aus Beispielen gelernt werden: **Einfache Lernalgorithmen, die effizient und skalierbar ist.**
- ▶ Es gibt Bäume sowohl für Klassifikation und Regression.
- ▶ Und sie lassen sich mit Modellen wie z.B. linearen Modellen kombinieren.

Was sind denn nun Entscheidungsbäume?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Eine von vielen Anwendungen: Kreditrisiken



Entscheidungsbaum

(n-äre) Entscheidungsäume bestehen aus einer Abfolge von n Entscheidungen (Test), die als Baum darstellbar sind.

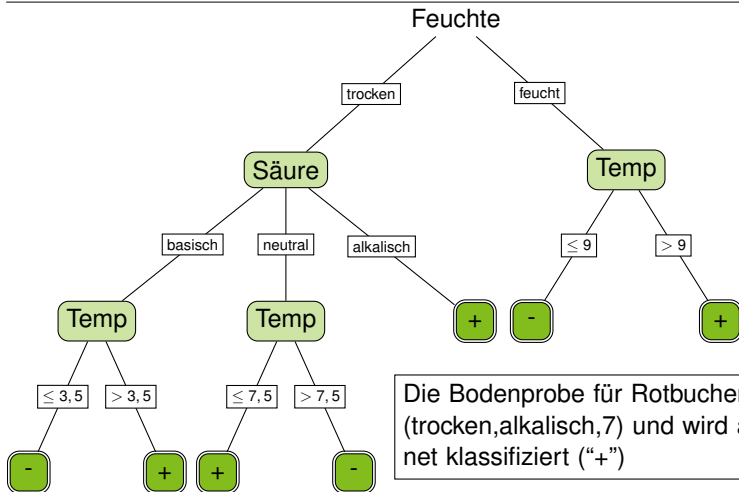
- ▶ Bei binären Bäumen findet in jedem Innerenknoden eine “Ja-Nein” (binäre) Entscheidung statt. Bei “Ja” folgt die Entscheidung im nächsten Knoten auf der linken Seite, bei “Nein” auf der rechten Seite.
- ▶ Die Entscheidungen werden so getroffen, dass die Knoten möglichst eine “reine” Klasse darstellen.
- ▶ In jedem Terminalknoten (Blatt) wird eine Zuordnung zu der Klasse getroffen, die dort am häufigsten vorkommt.

Klassifizieren mit Entscheidungsbäumen

Ein weiteres Beispiel aus der landwirtschaftlichen Produktion



TECHNISCHE
UNIVERSITÄT
DARMSTADT



+				-			
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

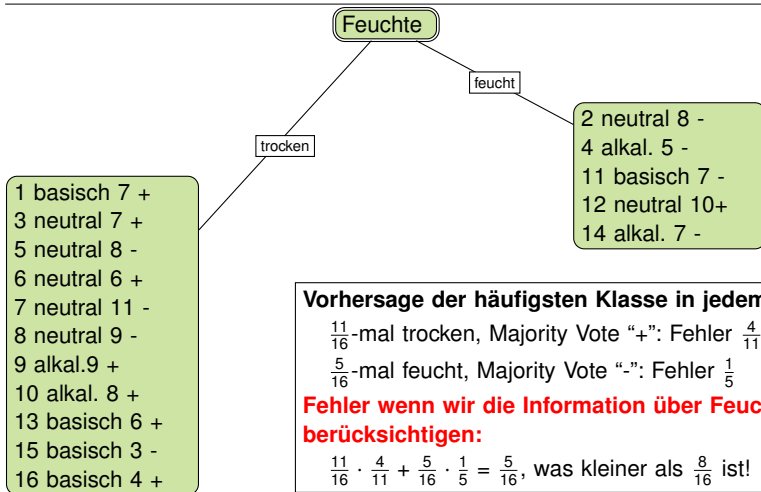
Ohne einen Test zu benutzen, können wir als Vorhersage immer “-” sagen
(Majority Vote, hier mit Tie-Break für “-”). Der Fehler ist dann 8/16.

Aufteilen nach Bodenfeuchte

Ein einzelner Test verbessert die Vorhersage!



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Bedingte Wahrscheinlichkeit

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

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

- Wir kennen die echte Verteilung nicht! Daher Annäherung der Wahrscheinlichkeit über die Häufigkeit in den Lernbeispielen mit Gewichtung bezüglich der Oberklasse. Beispiel: $Y = \{+, -\}$, $X_j = \{feucht, trocken\}$

$$P(+|feucht) = 1/5, \quad P(-|feucht) = 4/5 \text{ gewichtet mit } 5/16$$

$$P(+|trocken) = 7/11, \quad P(-|trocken) = 4/11 \text{ gewichtet mit } 11/16$$

- Wahl des Merkmals mit dem höchsten Wert (kleinsten Fehler)

Eine erste Idee: Betrachte den Informationsgewinn eines Merkmals

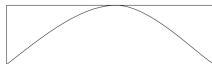


TECHNISCHE
UNIVERSITÄT
DARMSTADT

Information für eine Menge X von Beispielen

Sei p_+ die Wahrscheinlichkeit, dass ein Beispiel der Klasse “+” entstammt. Damit können wir die **Entropy** berechnen:

$$I(p_+, p_-) = (-p_+ \log p_+) + (-p_- \log p_-)$$



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.

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

Information für die Untermenge

Informationsgewinn

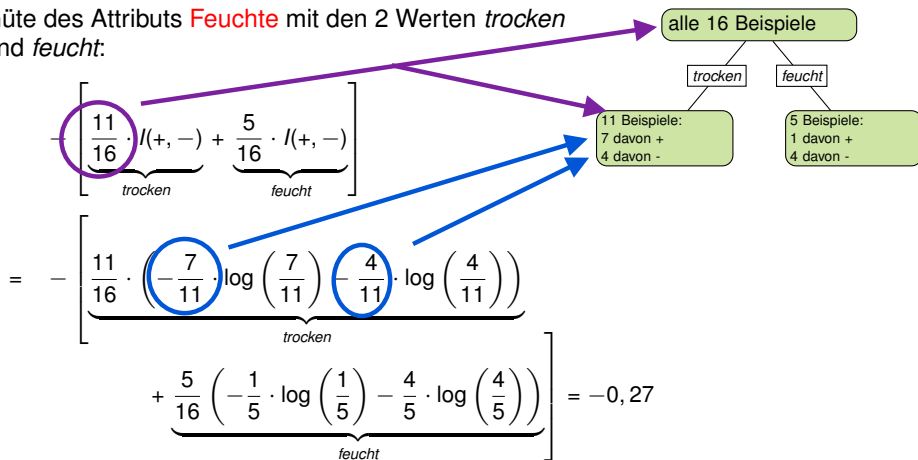
Differenz zwischen den Informationen der Beispiele mit und ohne die Aufteilung durch X_j .

Informationsgewinn des Merkmals Feuchte



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



gezeigt ist nur die Information/Entropie von FEUCHTE.

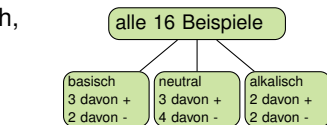
Informationsgewinn des Merkmals Säure



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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

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



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

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

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



- ▶ Die **Temperatur** ist ein numerischer Wert! Es gibt also unendlich viele Temperaturwerte
- ▶ **Einfachste Lösung**: 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 $\text{Temp} \leq 7$, davon 3 in +,
11 Beispiele $\text{Temp} > 7$, davon 6 in -.
- ▶ Die Güte (also hier der Informationsgewinn) der Temperatur als Merkmal ist dann $-0,29$.

Mit dem Informationsgehalt können wir Merkmale als Tests auswählen



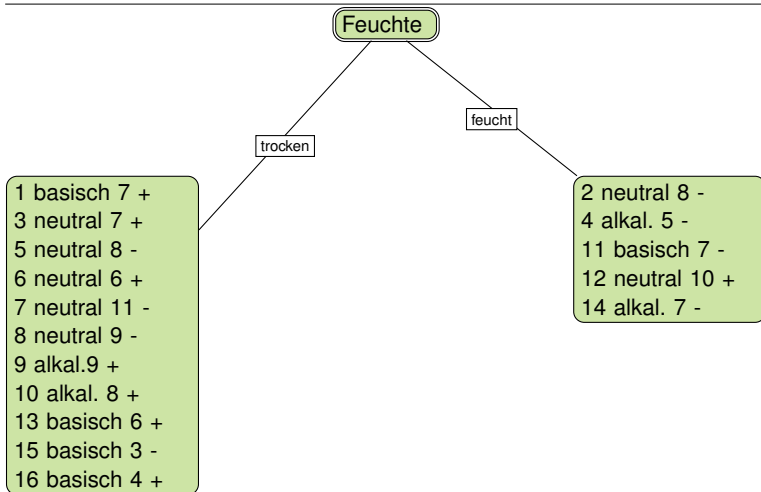
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Wir wählen das Merkmal X_j , dessen Werte am besten in (Unter-)mengen X_i aufteilen, die geordnet sind.
- ▶ Das Gütekriterium **Informationsgewinn** (mittels Entropie) bestimmt die Ordnung der Mengen.
- ▶ In unserem Beispiel hat *Feuchte* den höchsten Gütewert.

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



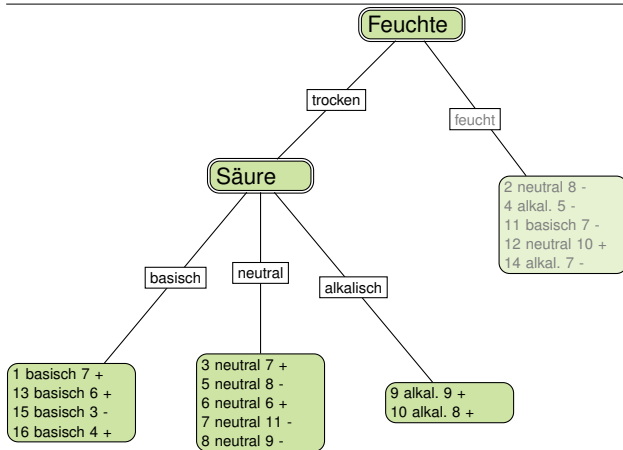
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Algorithmus TDIDT (ID3) am Beispiel



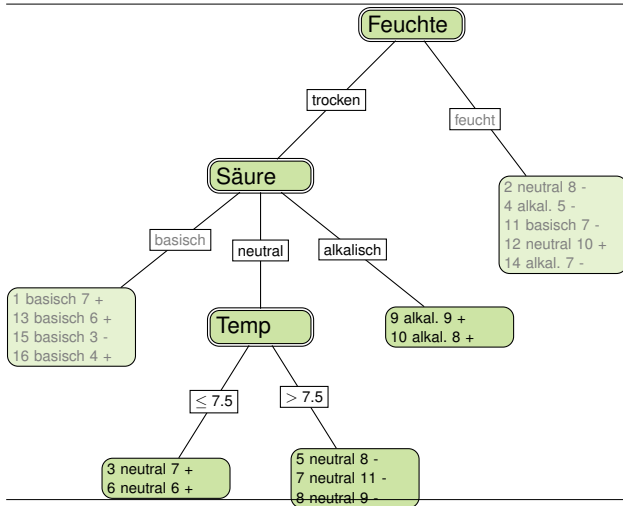
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Algorithmus TDIDT (ID3) am Beispiel



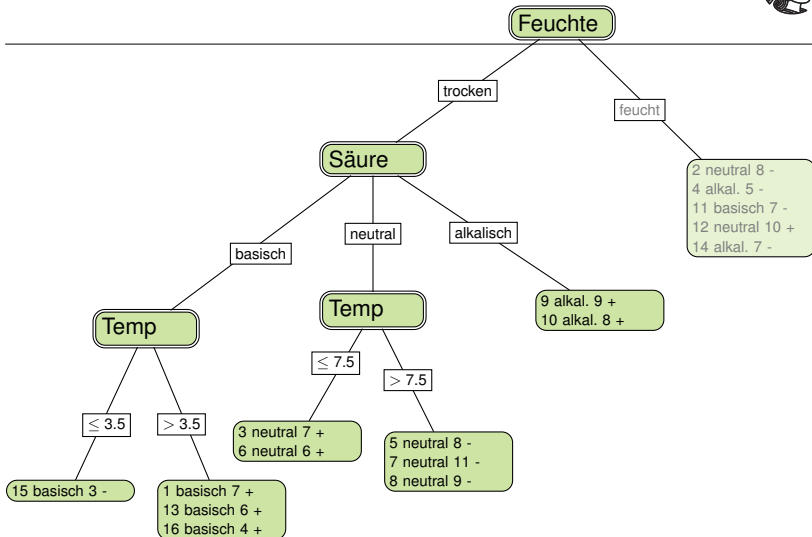
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Algorithmus TDIDT (ID3) am Beispiel



TECHNISCHE
UNIVERSITÄT
DARMSTADT





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 (Stutzen des Baumes)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Das Lernen von Entscheidungsbäumen ist effizient und skaliert gut.

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!



Aber Achtung! Es können große Bäume entstehen¹. Grosse Bäume können zwei Probleme haben:

- ▶ Obgleich sie sehr genau sind, erzeugen sie grosse Fehlerraten auf neuen Datensätzen, und
- ▶ das Verstehen und Interpretieren von Bäumen mit vielen Terminalknoten ist kompliziert. Grosse Bäume sind also komplexe Bäume.

Zwei Extreme: Nur ein Terminalknoten vs. grösster Baum, der möglich ist.

Die “richtigen Grösse” sollte zwischen den Extremen liegen.

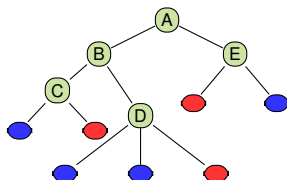
¹Die Komplexität eines Baumes wird gemessen durch die Anzahl seiner Terminalknoten.

Stutzen (pruning) des Baumes

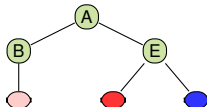


TECHNISCHE
UNIVERSITÄT
DARMSTADT

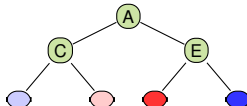
- ▶ Die Ziele des Stutzens:
 - ▶ Überanpassung (Overfitting) des Baums an die Trainingsdaten verringern!
 - ▶ Verständlichkeit erhöhen!
- ▶ Operationen des Stutzens (Pruning):
 - a) Knoten an Stelle eines Teilbaums setzen
 - b) Einen Teilbaum eine Ebene höher ziehen
- ▶ Schätzen, wie sich der wahre Fehler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



a) Knoten an Stelle eines Teilbaums setzen



b) Einen Teilbaum eine Ebene höher ziehen



Die Suche nach dem Baum der “richtigen Grösse” beginnt mit dem Stutzen (pruning) der Äste des grössten Baumes von den Endknoten her (“bottom up”).

- ▶ 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?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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

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

“grosses Z” = Z = Zufallsvariable
“kleines z” = z = ein Zustand von Z

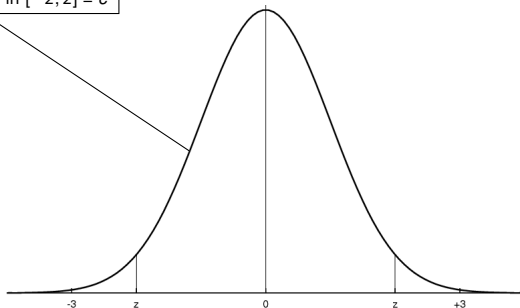
Die Wahrscheinlichkeit dafür, dass der Mittelwert im Intervall liegt, ist nun:

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

Bei einer z-Transformation werden die Abweichungen der ursprünglichen Werte von ihrem Mittelwert durch ihre Standardabweichung geteilt. Z-transformierte Variablen haben einen Mittelwert von 0 und eine Streuung von 1. Eine z-Transformation überführt z.B. jede Normalverteilung in eine Standardnormalverteilung

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



TECHNISCHE
UNIVERSITÄT
DARMSTADT

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

- ▶ $P(X \geq -z)$ bzw. $P(X \leq z)$ 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 (nachschiessen).
- ▶ 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 $[\bar{X} - 1,28 \frac{\sigma}{\sqrt{N}}; \bar{X} + 1,28 \frac{\sigma}{\sqrt{N}}]$.

Werte im Originalbereich.



- ▶ 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 $[\bar{X} - z(1 - \alpha/2) \frac{\sigma}{\sqrt{N}}; \bar{X} + 1,28 \frac{\sigma}{\sqrt{N}}]$ setzen wir diese Varianz ein und erhalten:

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



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

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

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.



- 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.



Wir wollen messen, wie “rein” die Klasse ist, die ein Knoten darstellt. Neben dem Informationsgewinn gibt es viele andere Maße.

Gini-Index

Das Gini-Index eines Knoten t ist definiert als

kleiner Gini-Index \Leftrightarrow geringe Unreinheit,
großer Gini-Index \Leftrightarrow hohe Unreinheit

$$I(t) := 1 - S \text{ mit } S = \sum_{j=1}^k p^2(j|t)$$

wobei S die Reinheitsfunktion ist.

- ▶ CART, ein anderer Baumlearner, nimmt im Wesentlichen den Gini-Index.
- ▶ Der Gini-Index nimmt sein Maximum an, wenn jede Klasse in dem Knoten mit gleicher Wahrscheinlichkeit angenommen wird.



Dazu passen wir dann unser Gütemaß an:

Allgemeines Gütemaß

Sei s eine Teilung im Knoten t . Das Gütemaß dieser Teilung mißt die Verringerung der “Unreinheit”:

$$\Delta I(s, t) = I(t) - p_+ \cdot I(t_+) - p_- \cdot I(t_-)$$

wobei $I(t_+)$ das Gütemaß des linken und $I(t_-)$ das des rechten Teilbaums ist.

- ▶ CART, ein anderer Baumlearner, nimmt im Wesentlichen den Gini-Index.
- ▶ Der Gini-Index nimmt sein Maximum an, wenn jede Klasse in dem Knoten mit gleicher Wahrscheinlichkeit angenommen wird.



Unser Ziel sind jetzt niedrige Fehlerquadrate (SSE). Sei t ein Knoten.

$$SSE(t) = \sum_{(\mathbf{x}, y) \in t} (y - \bar{y})^2 \quad \text{mit} \quad \bar{y} = \frac{1}{|t|} \sum_{(\mathbf{x}, y) \in t} y$$

SSE-Reduktion für $[x \leq z]$ Tests

Seien t_+ und t_- die Auteilung durch $[x \leq z]$. Analog zum allgemeinen Gütemaß, berechnen wir

$$\Delta SSE(s, t) = SSE(t) - SSE(t_+) - SSE(t_-)$$

- Führe keinen neuen Split ein, wenn SSE nicht um Mindestbetrag reduziert wird. Erzeuge dann Terminalknoten mit Mittelwert aus den aktuellen Beispielen in dem Knoten.



Doch zurück zur Klassifikation. Die Konfusionsmatrix lautet:

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

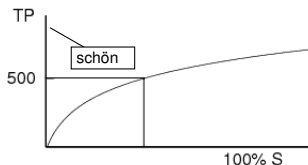
Accuracy: $P(\hat{f}(x) = y)$ geschätzt als $(TP + TN) / total$

Balance von FP und FN

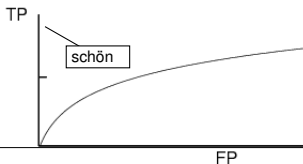
► F-measure: $\frac{\beta \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} = \frac{\beta TP}{\beta TP + FP + FN}$

► Verlaufsformen:

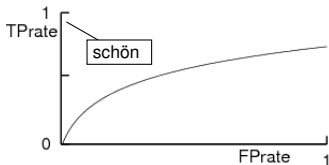
- Lift: TP für verschiedene Stichprobengrößen S



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



- ▶ 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

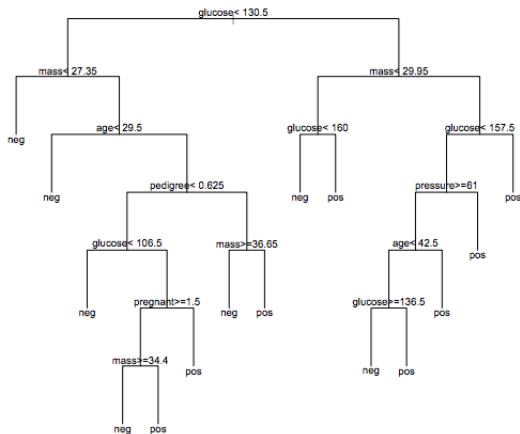


- ▶ Nicht immer sind FP so schlimm wie FN
 - ▶ medizinische Anwendungen: lieber ein Alarm zu viel als einen zu wenig!
- ▶ Gewichtung der Beispiele:
 - ▶ Wenn FN 3x so schlimm ist wie FP, dann gewichte negative Beispiele 3x höher als positive.
 - ▶ Wenn FP 10x so schlimm ist wie FN, dann gewichte positive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



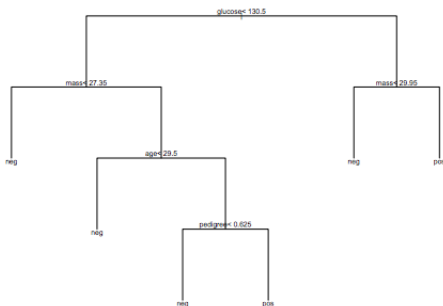
Zum Lernen: Funktion `rpart()` aus Paket `rpart`.

```
> library("rpart")  
> set.seed(20060911)  
> pid_rpart <- rpart(diabetes ~., data = pid_learn)  
> plot(pid_rpart, uniform = TRUE)  
> text(pid_rpart)
```



```
> pid_rpart <- prune(pid_rpart, cp = 0.018)
> plot(pid_rpart, uniform = TRUE)
> text(pid_rpart)
```





Zum Prunen:

```
> pid_rpart$scptable
```

	CP	nsplit	rel error	xerror	xstd
1	0.24117647	0	1.0000000	1.0000000	0.06230853
2	0.09411765	1	0.7588235	0.9117647	0.06083331
3	0.02745098	2	0.6647059	0.7705882	0.05783822
4	0.01764706	5	0.5823529	0.7647059	0.05769506
5	0.01372549	10	0.4941176	0.8000000	0.05853104
6	0.01000000	14	0.4294118	0.8411765	0.05943855



Zum Vorhersagen:

```
> pred_rpart <- predict(pid_rpart, newdata = pid_test,  
+                        type = "class")  
> mc(pred_rpart)
```

```
$table  
      pred  
true neg pos  
neg 133  25  
pos  36  58
```

```
$rate  
[1] 0.2420635
```

Was haben sie kennengelernt?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Sie kennen ID3 und CART als Beispiel für TDIDT.
- ▶ Für das Lernen verwendet ID3 das Gütemaß des Informationsgewinns auf Basis der Entropie. CART verwendet den Gini-Index
- ▶ Man kann etwas über die Performanz aussagen:
 - ▶ Man kann abschätzen, wie nah das Lernergebnis der unbekannten Wahrheit kommt → Konfidenz
 - ▶ Man kann abschätzen, wie groß der Fehler sein wird und dies zum Stutzen des gelernten Baums nutzen.
- ▶ Lernergebnisse werden also evaluiert:
 - ▶ Einzelwerte: accuracy, precision, recall, F-measure
 - ▶ Verläufe: ROC

Diese Evaluationsmethoden gelten nicht nur für Entscheidungsbäume!

- ▶ Sie kennen das R package `rpart`