Maschinelles Lernen und Datenanalyse

In der Werkstoff- und Prüftechnik

Prof. Dr. Stefan Bosse

Universität Koblenz - FB Informatik - Praktische Informatik

Universität Siegen - FB Maschinenbau / LMW

Klassifikation und Regression mit Entscheidungsbäumen

Zielvariablen: Primär Kategorische Variablen; Sekundär Numerische Variablen

Eigenschaftsvariablen: Kategorische und Numerische Variablen

Modell: Gerichteter azyklischer Graph (Baumstruktur)

Training und Algorithmen: C4.5, ID3, C5.0, ICE, CART, RF

Klasse: Überwachtes Lernen

Entscheidungsbäume

Einsatzbereiche von Entscheidungsbäumen:

- Große Datensätze, die komplexe Zusammenhänge beschreiben.
- Die Beziehung zwischen den Beobachtungen innerhalb der Datensätze müssen nicht linear sein!
- Die Modellfunktion und deren Parameter sind nicht bekannt.
  - Um den Modellzusammenhang zu beschreiben, wird das Modell trainiert (maschinelles Lernen)
  - Das erfordert, dass die Daten in mindestens einem Trainings- und einem Modelltest-Datensatz geteilt werden. Ab und an wird der Datensatz nicht nur in die zwei genanten, sondern noch in einem weiteren Datensatz, dem Validierungsdatensatz, aufgeteilt.
  - Diese Vorgehensweise ist notwendig, um Modellüberanpassungen zu erkennen!

Entscheidungsbäume

- Ein Entscheidungsbaum ist ein gerichteter azyklischer Graph bestehend aus einer Menge von Knoten $\boldsymbol{N}$ die mit den Eingabevariablen x verknüpft sind und Kanten $\boldsymbol{E}$ die die Knoten verbinden
- Die Endknoten sind Blätter und enthalten Werte der Zielvariablen y (daher kann y nur eine kategorische Variable sein, oder eine intervallkategorisierte)
- Die Kanten bestimmen die Evaluierung des Entscheidungsbaum beginnend von dem Wurzelknoten bis zu einem Blattknoten
  - Jede Kante hat eine Evaluierungsbedingung $\varepsilon(x)$ der Variable des ausgehenden Knotens x

Ein Entscheidungsbaum besteht aus Regeln. Jeder Knoten kann als eine Evaluierungsregel aufgefasst werden.

- Zusammengefasst ausgedrückt:

$$M(X) : X \rightarrow Y, X = \{x_i\}, Y = \{y_j\}$$
$$DT = \langle N_x, N_y, E \rangle$$
$$N_x = \{n_i : n_i \leftrightarrow x_j\}, N_y = \{n_i : n_i \leftrightarrow val(y_j)\}$$
$$E = \{e_{ij} : n_i \mapsto n_j | \epsilon_{ij}\}$$

- Entscheidungsbäume können neben einem Graphen auch funktional dargestellt werden:

$$M(X) = \begin{cases} x_i = v_1, \begin{cases} x_j = v_1, val(y) \\ x_j = v_2, val(y) \\ x_j = v_3, \{.. \end{cases} \\ x_i = v_2, \begin{cases} x_k = v_1, \{.. \\ x_k = v_2, \{.. \\ x_k = v_3, \{.. \end{cases} \\ x_i = v_3, \begin{cases} x_l = v_1, \{.. \\ x_l = v_2, \{.. \\ x_l = v_3, \{.. \end{cases} \end{cases}$$

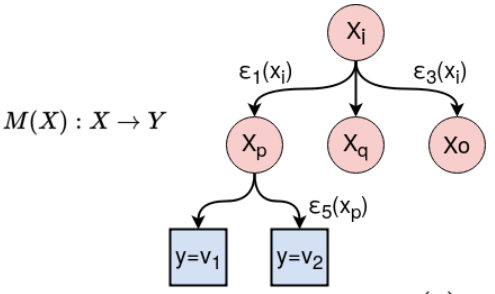
Baumklassen

Man unterscheidet:

- **Binäre Bäume**. Jeder Knoten hat genau (oder maximal) zwei ausgehende Kanten (Verzweigungen). Der Test der Variable x kann daher nur $x < v$, $x > v$, $x \geq v$, oder $x \leq v$ sein! Wird vor allem bei numerischen Variablen eingesetzt.

- **Bereichs- und Mehrfachbäume**. Jeder Knoten hat $1..k$ ausgehende Kanten (Knotengrad k). Der Test der Variable x kann auf einen bestimmten Wert $x \in V$ oder auf ein Intervall $[a,b]$ erfolgen! Wird vor allem bei kategorischen Variablen eingesetzt.

Baumstruktur



$$\varepsilon(x) = \begin{cases} x = v \\ x < v, x \leqslant v \\ x > v, x \geqslant v \\ x \in V, x \in [a, b] \end{cases}$$

Abb. 1. Grundlegende Struktur eines Entscheidungbaumes

Vorteile

Entscheidungsbäume sind einfach aufgebaut und können mit einfachen Algorithmen erzeugt werden.

- Entscheidungsbäume als inferiertes Modell erlauben eine **Erklärbarkeit** des Modells, also die Antwort auf die Frage wie sich ein y aus einem x ergibt.
- Weiterhin ist eine Ableitung eines **inversen Problems** möglich, d.h. welche Werte x für gegebenes y sind möglich?

Nachteile

Entscheidungsbäume können schnell **spezialisieren**, d.h. es fehlt an **Generalisierung**.

- Theoretisch kann mit einem Entscheidungsbaum jede Trainingsdatentabelle mit einer Trefferquote von 100% abgebildet werden. Der Test mit nicht trainierten Daten ergibt aber Prädiktion in der Größenordnung der Ratewahrscheinlichkeit!

Bevor man das Training startet, insbesondere bei mehrschrittigen Verfahren, kann es hilfreich sein den Fehler für die "Ratewahrscheinlichkeit" gemäß der im Training benutzten Fehlerfunktion (loss) zu berechnen.

Beispiel Regression:

```
use math
x=[1,2,3,4,5,6,7,8]
y=[1,2,3,4,5,6,7,8]
y.median = fivenum(y)$median
loss2 = sqrt(mean((y-y.median)^2))
>> 2.29
```

Solange beim oder nach dem Training der fehler/Verlust nicht nennenswert kleiner (mindestens 1/2, besser 1/10) ist kann ist das Modell nicht brauchbar (bei Regressionsmodellen spricht man auch von der Todeslinie wenn das Modell konstant ungefähr den Median ausgibt).

Training

- Das Training mit Trainingsdaten $\boldsymbol{D}_{\text{train}}$ erzeugt den Baum *schrittweise*:
  - Es werden geeignete Variablen $x \in \boldsymbol{X}$ ausgewählt die einen *Knoten* im Baum erzeugen
  - Jeder hinzugefügte Knoten erzeugt neue Teilbäume (durch Verzweigungen)
  - Die *Verzweigungsbedingungen* ε (Kanten) werden ebenfalls vom Trainer anhand der Werte der Variable x in Abhängigkeit von der Zielvariablen y gewählt/berechnet.
- Die Auswahl der Variablen und die Verzweigungsbedingungen können je nach Algorithmus und Baumklasse variieren!

Beispiel

[10]

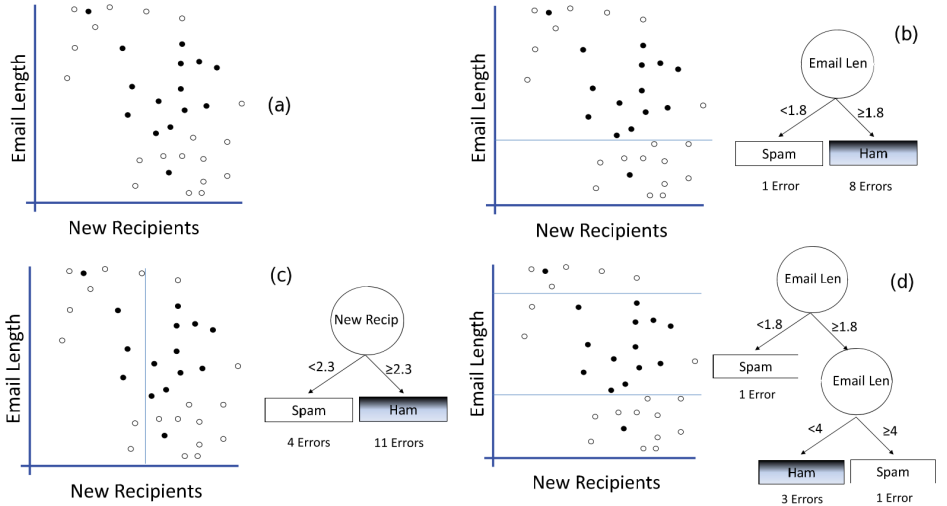
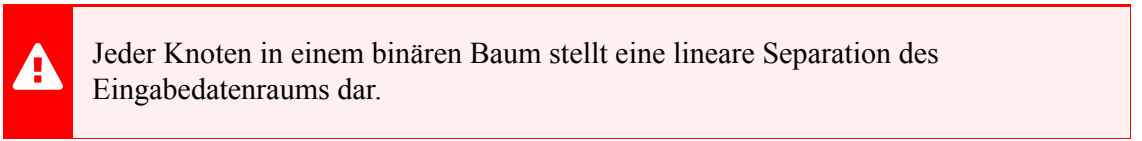


Abb. 2. Schrittweise Erzeugung des Entscheidungsbaums aus den Eingabedaten (a) erst mit einer Variable (b,c), dann mit zwei (d) unter Beachtung des Klassifikationsfehlers



Jeder Knoten in einem binären Baum stellt eine lineare Separation des Eingabedatenraums dar.

Probleme bei Mehrbereichsbäumen

- Wenn die Wertemenge $val(x)$ groß ist gibt es entsprechend auch viele Verzweigungen im Baum!
  - Die Größe des Baums wächst an (Speicher)
  - Die Rechenzeit für das Training (Induktion) aber auch die Anwendung (Inferenz, Deduktion) wächst
  - Die Entropie kann als Maß der Varianz der Wertemenge gesehen werden und kann die Komplexität des Baumes bestimmen.

Das "NP" Problembeispiel

[14]

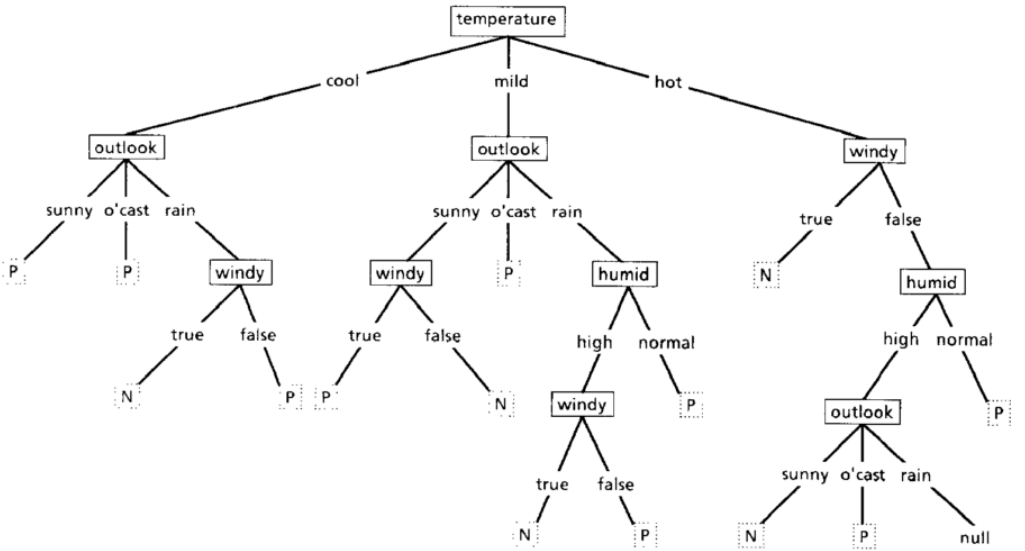


Abb. 3. k-stelliger Entscheidungsbaum für kategorische Variablen

Das Titanic Überlebensbeispiel

[www.statistik-dresden.de]

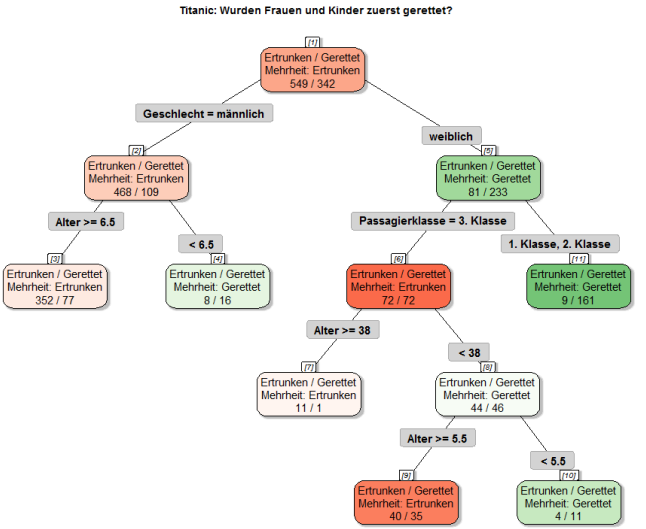


Abb. 4. Binärer Entscheidungsbaum (Relation und Auswahl) für numerische und kategorische Variablen: Beantwortung "soziologischen Fragen", und nicht Prädiktion

Beispiel Materialeigenschaften

[100]

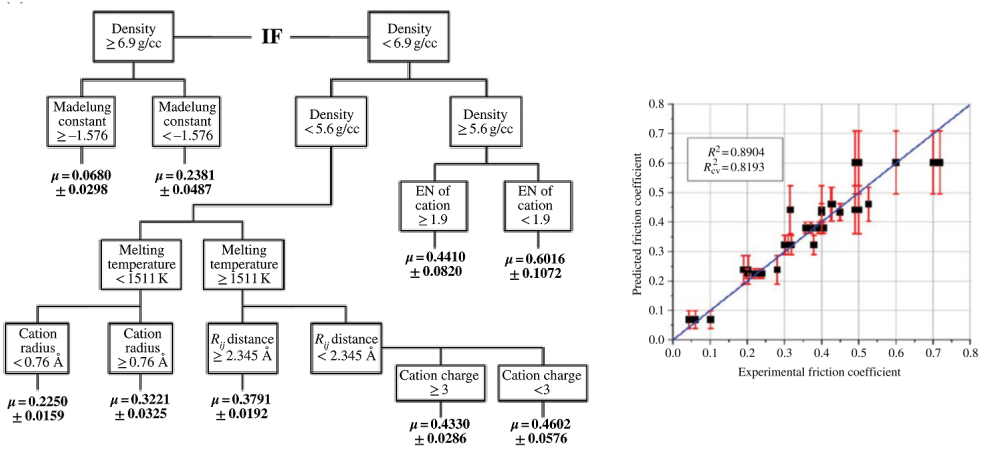


Abb. 5. (Links) Entscheidungsbaum für die Vorhersage von Reibungskoeffizienten von Materialien auf der Grundlage von sechs grundlegenden Materialmerkmalen (Rechts) Vergleich der vorhergesagten und experimentellen Reibungskoeffizienten

Trainingsalgorithmen

- Es gibt verschiedene Trainingsverfahren (für verschiedene Baumklassen):
  - **ID3**. Der Klassiker (Iterative DiChaudomiser 3, Ross Quinlan, 1975-1986) für kategorische Variablen (k-stelliger Baum)
  - **C4.5**. Der Klassiker (Ross Quinlan 1988-1993) für numerische (und kategorische) Variablen (Binär- und k-stelliger Baum) als Erweiterung des ID3 Verfahrens.
  - **C5.0** Nachfolger von **C4.5** mit verbesserten Split und Baumreduzierung
  - **INN**. Die Eigenkreation (Interval Nearest Neighbor oder auch *ICE*, Stefan Bosse, 2016) für numerische unsichere und verrauschte Sensorwerte mit Intervallarithmetik (also im Prinzip Intervallkategorisierung und Kantenbedingungen sind $x \in [a,b]$), basierend auf C4.5 und ID3
  - **CART** Classification and Regression Tree (kat. und kont. numerische Zielvariablen)

Bewertung der Qualität eines Modells

- Binäre Kreuzentropie bei **kategorischen Ausgabevariable(n)**

Die Kreuzentropie ist in der Informationstheorie und der mathematischen Statistik ein Maß für die Qualität eines Modells für eine Wahrscheinlichkeitsverteilung. Eine Minimierung der Kreuzentropie in Bezug auf die Modellparameter kommt einer Maximierung der Log-Likelihood-Funktion gleich. Es gilt mit p als Zielwertverteilung von y und q als Verteilung der Prädktion y_p:

$$H(p,q) = -\sum_c p(c) \log\left(q(c)\right)$$

$$p(c) = \frac{\mathrm{count}(y|y=c)}{N}$$

$$q(c) = \frac{\mathrm{count}\left(y_p|y_p=c\right)}{N}$$

$$c \in C = \{U, V, W, ..\}$$

Bewertung der Qualität eines Modells

- *Logloss*, die Abkürzung für logarithmischer Verlust, misst die Genauigkeit eines Klassifikators mit **kontinuierlichen Ausgabevariablen**, d.h. im normierten Werteintervall [0,1], indem es falsche Klassifizierungen bestraft. Für eine binäre Klassifizierung mit echter Bezeichnung y={0,1} und vorhergesagter Wahrscheinlichkeit *p*=[0,1] ist der Logloss gegeben durch:

$$L(y,p) = \frac{1}{N}\sum_{i=1}^{N} - y_i \log\left(p_i\right) - \left(1-y_i\right)\log\left(1-p_i\right)$$

Bewertung der Qualität eines Modells

- Mean-Squared Error (MSE) und Root-Mean-Squared-Error (RMSE) bei numerischen Ausgabevariablen (sowohl diskrete als auch regressive Bäume), hier ist y_p der vom Modell vorhergesagte Wert, und y_0 der bekannte Ground Truth Wert:

$$\mathrm{rmse}\left(y_p, y_0\right) = \sqrt{\frac{1}{N}\sum\left(y_p - y_0\right)^2}$$

Weitere Informationen und Vertiefung:

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

Vergleich ID3 - C4.5

- Der ID3-Algorithmus wählt das beste Attribut basierend auf dem Konzept der Entropie und dem Informationsgewinn für die Entwicklung des Baumes.
- Der C4.5-Algorithmus verhält sich ähnlich wie ID3, verbessert jedoch einige ID3-Verhaltensweisen:
  - Möglichkeit, numerische (kont.) Daten zu verarbeiten.
  - Verarbeitung unbekannter (fehlender) Werte
  - Möglichkeit, Attribute mit unterschiedlichen Gewichten zu verwenden.
  - Beschneiden des Baumes nach der Erstellung (**Modellkompaktierung**).
  - Vorhersage der Fehler
  - Hervorhebung und Extraktion von Teilbäumen

ID3 Verfahren

[1] J. R. Quinlan, “Induction of Decision Trees,” in Machine Learning, Kluwer Academic Publishers, Boston, 1986.

Entropie

- Ausgangspunkt für die Konstruktion des Entscheidungsbaums ist die (Shannon) Entropie einer Spalte *X* der Datentabelle (mit der Variable *x*):

$$E(X) = -\sum_{i=1,k} p_i \log_2(p_i), p_i = \frac{count(X = c_i)}{|X|}, X = \{c|c \in C\}$$

Alle Werte gleich ⇒ Entropie=0; Alle Werte gleichverteilt ⇒ Entropie=-$\log_2|c_i|$

[https://towardsdatascience.com/understanding-entropy-the-golden-measurement-of-machine-learning-4ea97c663dc3]

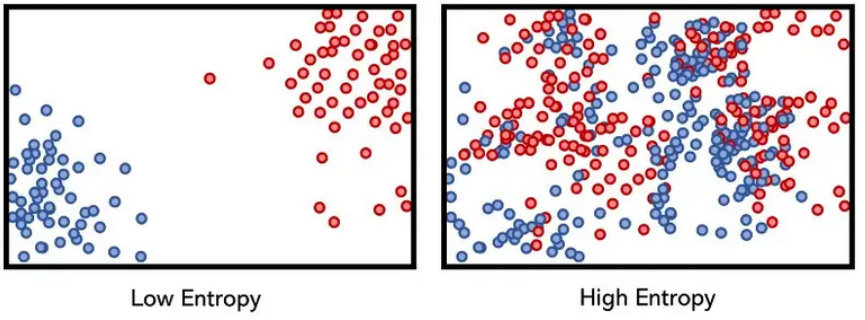


Abb. 6. Illustration der Bedeutung der Entropie: Entropie hat ihre Wurzeln in der Physik- sie ist ein Maß für Unordnung oder Unvorhersehbarkeit in einem System.

Entscheidungsbäume verwenden Entropie für ihre Konstruktion: Um Eingaben so effektiv wie möglich nach einer Reihe von Bedingungen zu einem korrekten Ergebnis (Zielvariable) zu lenken, werden Merkmalteilungen (mit Bedingungen) mit niedrigerer Entropie (höherer Informationsgewinn) höher im Baum platziert.

Um die Idee von Bedingungen mit niedriger und hoher Entropie zu veranschaulichen, betrachten wir hypothetische Merkmale mit einer durch Farbe (rot oder blau) markierten Klasse und der durch eine vertikale gestrichelte Linie markierten Teilung.

[https://towardsdatascience.com/understanding-entropy-the-golden-measurement-of-machine-learning-4ea97c663dc3]

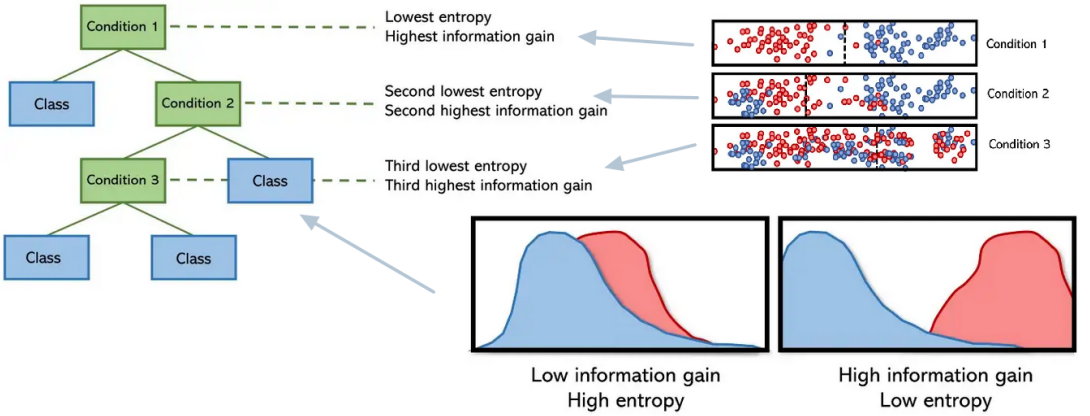


Abb. 7. Betrachtet man die Verteilung der Zielvariablen *y*, dann ergibt sich ein hoher Informationsgewinn bei niedriger Entropie (von *y*) für eine gute Teilung

- Entscheidungsbäume berechnen die Entropie von Merkmalen und ordnen sie so an, dass die Gesamtentropie des Modells minimiert (und der Informationsgewinn maximiert) wird.

- Mathematisch bedeutet dies, dass die Bedingung mit der niedrigsten Entropie oben platziert wird, so dass sie dazu beitragen kann, Knoten darunter zu spalten, um die Entropie zu verringern.

- Informationsgewinn und relative Entropie, die beim Training von Entscheidungsbäumen verwendet werden, sind definiert als der Abstand zwischen zwei Wahrscheinlichkeitsverteilungen $p(x)$ und $q(x)$ (siehe vorherige Abb.).

Bedingte Entropie

- Interessant ist die Werteverteilung einer Eingabevariablen *X* in Bezug auf die Werte (Partitionen) der Zielvariable *Y* ⇒ Bedingte Entropie

$$H(X|Y=y) = -\sum_{i=1,k} p_i \log_2(p_i),$$

$$p_i = \frac{count(X|X=c_i \wedge Y=y)}{N_y},$$

$$X_y = \{c|c \in C \wedge Y=y\},$$

$$C = \{c_i|i=1,2,..,k\}$$

- *C* ist die Menge aller unterscheidbaren Werte von *X*!

Beispiel

a	b	y
u	u	A
v	v	A
w	u	B
w	v	B

$$E(a) = -\frac{1}{3}\log\left(\frac{1}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right) - \frac{1}{3}\log\left(\frac{2}{3}\right) = 1.5$$

$$E(b) = -\frac{2}{4}\log\left(\frac{2}{3}\right) - \frac{2}{3}\log\left(\frac{2}{3}\right) = 1$$

$$H(a|y=B) = -\frac{2}{2}\log\left(\frac{1}{2}\right) - \frac{2}{2}\log\left(\frac{1}{2}\right) = 0$$

$$H(b|y=B) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$$

Informationsgewinn

- Ausgehend von der bedingten Entropie kann der Informationsgewinn einer Spalte X hinsichtlich der Zielvariablenspalte Y berechnet werden:

$$G(Y|X) = E(Y) - \sum_{v \in Val(X)} \frac{|(Y|X = v)|}{|Y|} E(Y|X = v)$$

- Der Informationsgewinn, der durch Auswahl des Attributs x und der Spalte X erzielt wird, errechnet sich dann als Differenz der Entropie von Y und der erwarteten/durchschnittlichen Entropie von Y bei Fixierung von x.

Der Informationsgewinn ist auf Y Verteilung bezogen, nicht wie vorher auf X!

Algorithmus

0. Starte mit leeren Baum, allen Eingansattributen X, der Zielvariablen Y, und der vollständigen Datentabelle D(X,Y).
1. Berechne den Informationsgewinn für jede Attributevariable $x \in X$.
2. Wenn nicht alle Zeilen zum selben Zielvariablenwert gehören, wird der Datensatz D in Teilmengen $D'_{xbest,v1}$, $D'_{xbest,v2}$, usw. aufgeteilt für das Attribut $x_{best} \in X$ mit dem größten Informationsgewinn.
3. Es wird ein Knoten mit der Attributvariable x_{best} erstellt.
4. Wenn alle Zeilen zur selben Klasse gehören, wird ein Blattknoten mit dem Wert der Zielvariable erstellt.
5. Wiederholung von 1-4 für die verbleibenden Attribute $X'=X / x_{best}$, allen Teilbäumen (Verzweigungen von aktuellen Knoten) mit jeweiligen D', bis alle Attribute verwendet wurden, oder der Entscheidungsbaum alle Blattknoten enthält.

C4.5 Verfahren

[1] J. R. Quinlan, "C4.5: Programs For Machine Learning". Morgan Kaufmann, 1988.

- Wie ID3 werden die Daten und Attribute an jedem Knoten des Baums bewertet um das beste Teilungsattribut zu bestimmen.
- Aber C4.5 verwendet die Methode der "gain ratio impurity", um das Teilungsattribut zu bewerten (Quinlan, 1993).
- Entscheidungsbäume werden in C4.5 mithilfe eines Satzes von Trainingsdaten oder Datensätzen wie in ID3 erstellt.
- An jedem Knoten des Baums wählt C4.5 ein Attribut der Daten aus, das seinen Satz von Samples am effektivsten in Teilmengen aufteilt, die in der einen oder anderen Klasse verteilt sind.

- Das Kriterium ist der **normalisierte Informationsgewinn**:
  - Verhältnis des Informationsgewinns *G* (Gain) zu einer sog. Teilungsqualität (Split Info *SI*), die sich aus der Zielvariable *Y* zum Aufteilen nach den *Y* Werten der Daten ergibt.
  - Das Attribut mit dem höchsten Verhältnis *GR* (Gain Ratio) wird ausgewählt, um die Entscheidung für die Teilung zu treffen.

$$G(Y|X) = E(Y) - \sum_{v \in Val(X)} \frac{|Y_v|}{|Y|} E(Y_v)$$

$$SI(Y) = \sum_{c \in Val(Y)} -\frac{|Y_c|}{|Y|} \log_2 \frac{|Y_c|}{|Y|}$$

$$GR = \frac{G(Y|X)}{SI(Y)}$$

Teilung von kategorischen und numerischen Variablen

- Bei kategorischen Variablen bestimmen die Werte $Val(X)$ einer Spalte der Datentabelle einer Variablen x die Aufteilung eines Entscheidungbaums (**Partitionierung**).
- Bei numerischen Variablen muss ein Wert als Teilungspunkt aus der Werteverteilung bestimmt!
  - Nicht trivial; Welches Kriterium?
  - Intervallkategorisierung und Wertepartitionierung kann helfen!
  - D.h. mit intervallkategorisierten diskrete Werter wird die Spalte X entsprechend der Zielvariable Y partitioniert.
  - Und diese Partitionen werden bewertet und der Teilungspunkt $x_{\text{split}} \in X$ bestimmt (z.B. über Mittelwerte der Intervalle)

Teilung von numerischen Variablen mit Intervallanalyse

Naiver Ansatz:

1. Paritionierung der x Variable in Paritionen $x|y$ (also nach y Werten)
2. Berechnung der statistischen *fivenum* Verteilunge: $\{min,q1,meadian,q3,max\}$
3. Die Intervalle $[min,max]$ der Paritionen aufsteigen sortieren
4. Fall A: Zwei Parititionen P_1 und P_2 besitzen nicht überlappende Intervalle $\Rightarrow$ Teilungspunkt ist Mittelpunkt $(max_2\text{-}min_1)/2$
5. Fall B: Zwei Parititionen P_1 und P_2 besitzen überlappende Intervalle,aber $q3_1 < q1_2 \Rightarrow$ Teilungspunkt ist Mittelpunkt $(q1_2\text{-}q3_1)/2$
6. Fall C: Zwei Parititionen P_1 und P_2 besitzen überlappende Intervalle,aber $median_1 < median_2 \Rightarrow$ Teilungspunkt ist Mittelpunkt $(meadian_2\text{-}median_1)/2$

Vertiefung

[1] L. Rokach and O. Maimon, Data Mining with Decision Trees - Theory and Applications. World Scientific Publishing, 2015.

Intervallkodierung

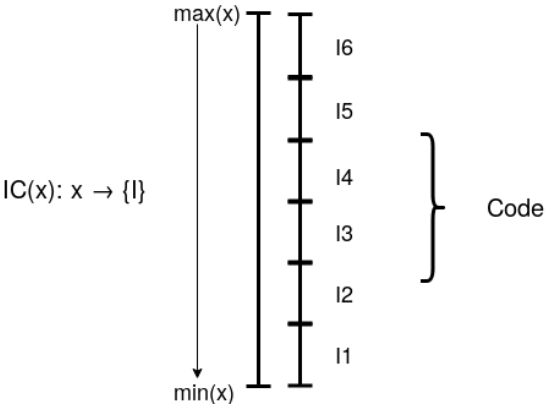


Abb. 8. Einteilung von kontinuierlichen Werteverteilungen in Intervalle und Abbildung auf kategorische (diskrete) Werte

Intervallkodierung

Man kann die R *cut* Funktion benutzen im eine kontinuierliche numerische Variable in eine kategorische zu überführen.

```
# Äquidistante Intervalle
x=runif(10)
x.disc = cut(x,breaks=3,labels=['A','B','C'])
# ['A','B','A','C',...]
# Unregelmäßige Intervalle
x=runif(10)
x.disc = cut(x,breaks=[0,0.2,0.8,1],labels=['A','B','C'])
# ['A','B','A','C',...]
```

Unvollständige Trainingsdaten

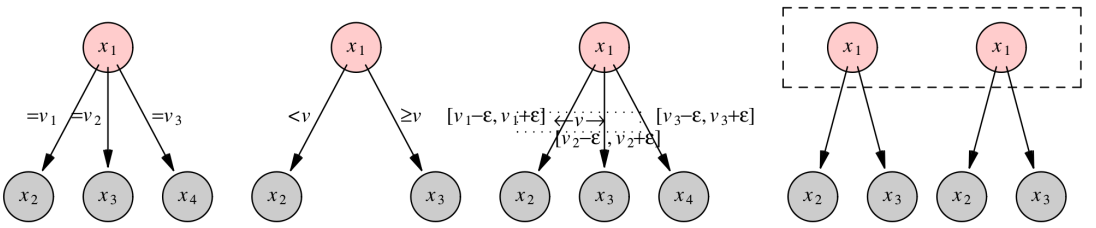
- Es kommt vor allem in der Soziologie aber auch in der Mess- und Prüftechnik vor, dass nicht alle Werte der Attributvariablen $\boldsymbol{X}$ für alle Trainingssätze bekannt sind.
  - Die Behandlung fehlender Attributwerte in den Zeilen der Datentabellen ist schwierig
- Es gibt keine Universallösung für den Umgang mit ? Werten. Möglichkeiten:
  - Ersetzen des fehlenden Wertes mit einem Standardwert
  - Ersetzen des fehlenden Wertes mit einem probabalistisch über Verteilungshäufigkeiten bestimmten Wert (auch unter Einbeziehung des gesamten Datensamples)
  - Attributvariablen mit fehlenden Werten nicht verwenden

ICE/INN Intervallkategorisierte Entscheidungsbäume

- Bisherige Entscheidungsbäume (C4.5/ID3) wurden entweder mit einer diskreten Anzahl von kategorischen Werten verzweigt oder mittels binärer Relationen!
- Aber Sensoren (sowohl in der Mess-und Prüftechnik als auch in der Soziologie) sind fehlerbehaftet, d.h. es gibt bei jedem x-Wert ein Unsicherheitsintervall $[x-\delta,x+\delta]$ → **Rauschen**, ebenso Teilungsintervalle $[x_1-\delta,x_2+\delta]$
- Damit können Entscheidungsbäume (anders als Neuronale Netze oder Regressionslerner) nicht umgehen.
  - Wenn die Teilung mit $x<50$ und $x\geq 50$ an einem Knoten mit x erfolgt würde bei Werten um 50 und überlagerten Rauschen ein Entscheidungsproblem entstehen!

- Lösung: k-stellige Knoten mit Intervallverzweigungen, also:

$$M(X) = \begin{cases} x_i \in [v_{1,1} - \varepsilon_i, v_{1,2} + \varepsilon_i], \{\cdots \\ x_i \in [v_{2,1} - \varepsilon_i, v_{2,2} + \varepsilon_i], \{\cdots \\ \cdots \\ x_i \in [v_{n,1} - \varepsilon_, v_{n,2} + \varepsilon_i], \{\cdots \end{cases}$$



(a) ID3 (b) C45 (c) ICE (d) RF

Abb. 9. Vergleich der verschiedenen Baumarten und Knotenverzweigungen

- Bei der Konstruktion des Entscheidungsbaums werden wieder nach Informationsgewinn bzw. Gewinnverhältnis Attributvariablen und Spalten der Datentabelle ausgewählt.
- Die numerischen Werte werden sowohl beim Training als auch bei der Inferenz durch Intervalle ersetzt → Ersetzung von diskreter mit **Intervallarithmetik**
- Entropie usw. werden durch kategorisierte Intervalle bestimmt
- Das große Problem: Für jede Variable muss ein ε abgeschätzt werden → Statistisches Modell erforderlich.
- Und was bedeuten jetzt überschneidende Intervalle?
    - Überschneidungen bedeuten Ununterscheidbarkeit!

Inferenz mit NN Suche

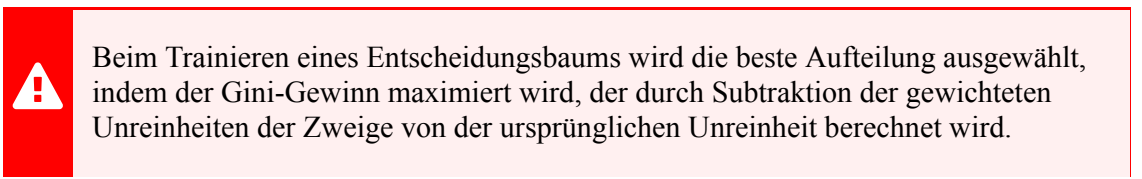
- Jeder Knoten x_i hat ausgehende Kanten mit annotierten Intervallen $[v_j-\varepsilon,v_j+\varepsilon]$
- Bei einem neuen zu testenden Variablenwert v wird einerseits auch ein Intervall $[v-\varepsilon,v+\varepsilon]$ gebildet und mit den Kantenintervallen verglichen, andererseits wird das nächstliegende Intervall gesucht

Gini-Index/Gini-Unreinheit

Die Gini-Unreinheit (Ginit Impurity GI) ist die Wahrscheinlichkeit, ein zufällig ausgewähltes Element im Datensatz falsch zu klassifizieren, wenn es gemäß der Klassenverteilung im Datensatz zufällig annotiert wurde. GI wird berechnet als:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

wobei C die Anzahl der Klassen und $p(i)$ die Wahrscheinlichkeit ist, zufällig ein Element der Klasse i auszuwählen.



Beim Trainieren eines Entscheidungsbaums wird die beste Aufteilung ausgewählt, indem der Gini-Gewinn maximiert wird, der durch Subtraktion der gewichteten Unreinheiten der Zweige von der ursprünglichen Unreinheit berechnet wird.

Gini-Index/Gini-Unreinheit

Wir bestimmen die Qualität der Aufteilung, indem wir die Ungleichverteilung (Unreinheit) jedes Zweigs mit der Anzahl seiner Elemente gewichten.

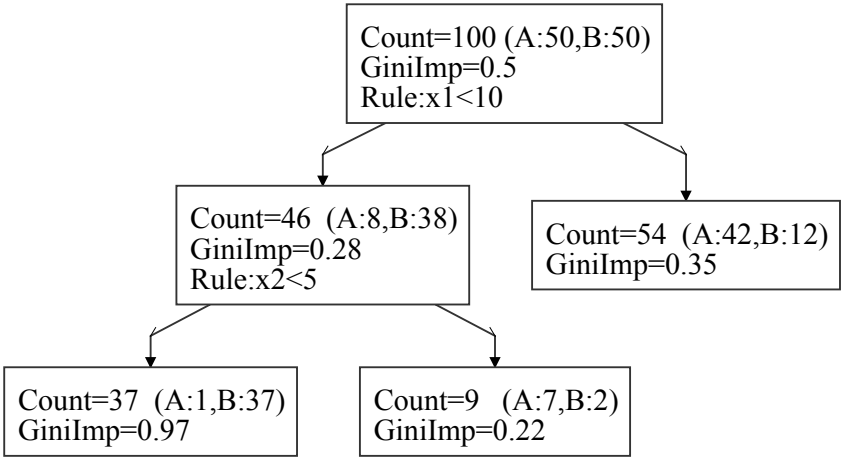
$$\text{WeightedGini}(split_i) = \left(\frac{n_{i,l}}{n_j} GI_l + \frac{n_{i,r}}{n_j} GI_r \right)$$

$$\text{GiniGain}(split_i) = WGI_i - WGI_j > 0$$

Beispiele: https://victorzhou.com/blog/gini-impurity/

Gini-Index/Gini-Unreinheit

Beispiel



Weighted Gini-Impurity (erster Split) = 46/100•0.28 + 54/100•0.35 = 0.32, Split Gain = 0.5 - 0.32

Random Forest Trees

Konzept und Idee: Mehrere schwache Modelle zu einem starken kombinieren.

- Multiinstanzmodell
  - Es werden *m* Entscheidungsbäume $DT=\{dt_1,..,dt_m\}$ getrennt gelernt und erzeugt
  - "Random": Die Aufteilung der Daten in Teilungsvariablen erfolgt randomisiert!
  - Eingabedaten werden zur Inferenz an alle Teilbäume $dt_i \in DT$ gegeben
  - Alle Ausgabevariablen der Teilbäume werden fusioniert

- Fusion:
  - Mittelwert (bei intervallkodierten oder intervallskalierbaren kat. Zielvariablen durch Dekodierung in numerische Werte)
  - Mehrheitsentscheid
  - Konsensfindung (Verhandlung)

- Parametersatz:
  - Stelligkeit eines Knotens (Anzahl der ausgehenden Kanten)
  - **Anzahl der Teilbäume**
  - **Partitionierung** des Eingaberaums (d.h. ein bestimmter Baum verwendet nur eine Teilmenge der Spalten aus $\boldsymbol{D}$)
  - Maximale **Höhe** eines Teilsbaumes
  - Fusionsmodell und Algorithmus

[Abhishek Sharma, 2020, www.analyticsvidhya.com]

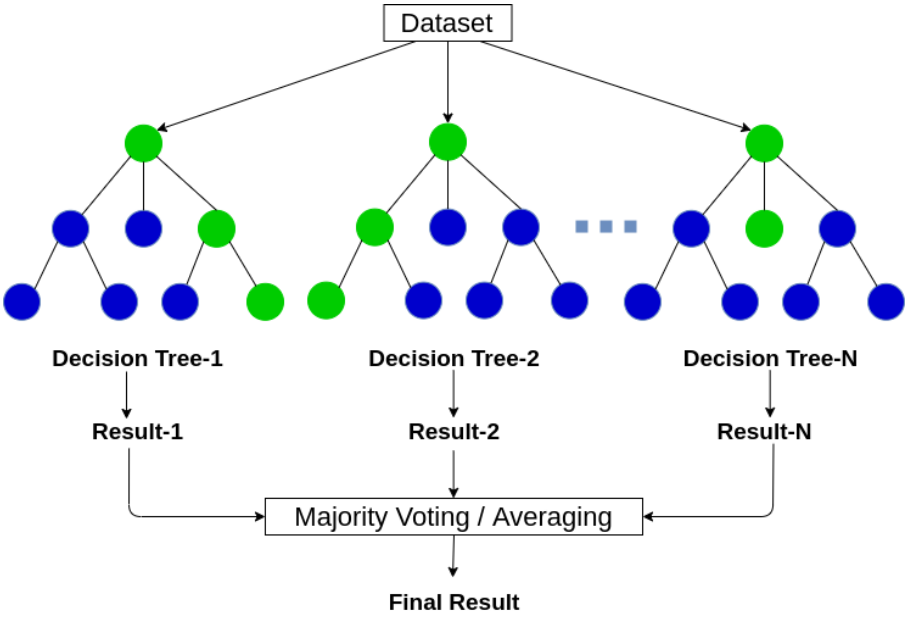
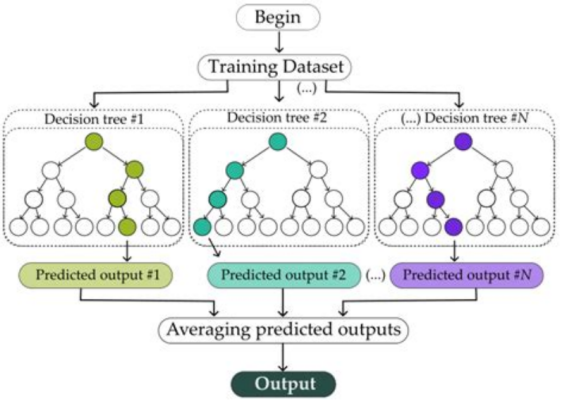


Abb. 10. Grundprinzip von Multibaumklassifikatoren

Random Forest Trees



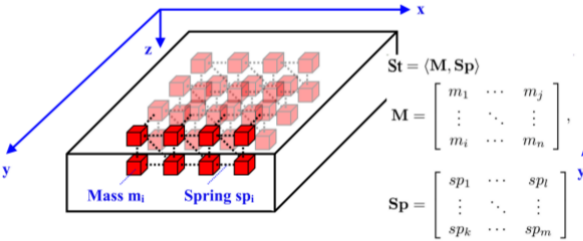
[https://www.mdpi.com/1424-8220/23/13/6153/html]

Abb. 11. Verschiedene Inferenzpfade in multiplen Bäumen

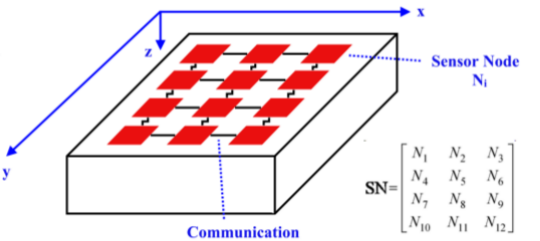
Beispiel

Experiment

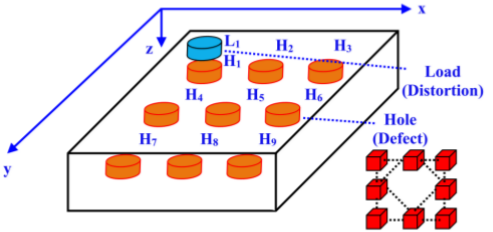
- Sensornetzwerk von (3 × 4) Dehnungssensoren
- Stimulus: Bauteilschwingung
- Varianz: Bauteilschäden (Defekte)
- Zielvariable: Schadensklassifikation (9 Positionen)
- Merkmalsvektor: Downgesampletes zeitaufeglöstes Sensorsignal einer


x
z
y
Mass m_i
Spring sp_i
$St = \langle M, Sp \rangle$
$M = \begin{bmatrix} m_1 & \cdots & m_j \\ \vdots & \ddots & \vdots \\ m_i & \cdots & m_n \end{bmatrix}$,
$Sp = \begin{bmatrix} sp_1 & \cdots & sp_l \\ \vdots & \ddots & \vdots \\ sp_k & \cdots & sp_m \end{bmatrix}$

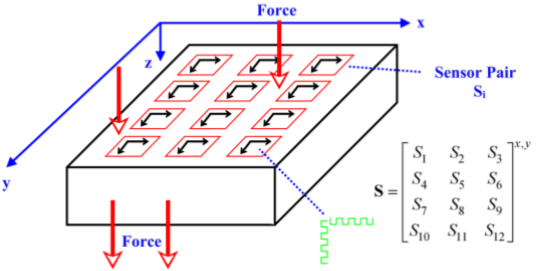
(a)


x
z
y
Sensor Node
N_i
Communication
$SN = \begin{bmatrix} N_1 & N_2 & N_3 \\ N_4 & N_5 & N_6 \\ N_7 & N_8 & N_9 \\ N_{10} & N_{11} & N_{12} \end{bmatrix}$

(b)


x
z
y
L_1
H_1
H_2
H_3
H_4
H_5
H_6
H_7
H_8
H_9
Load
(Distortion)
Hole
(Defect)

(d)

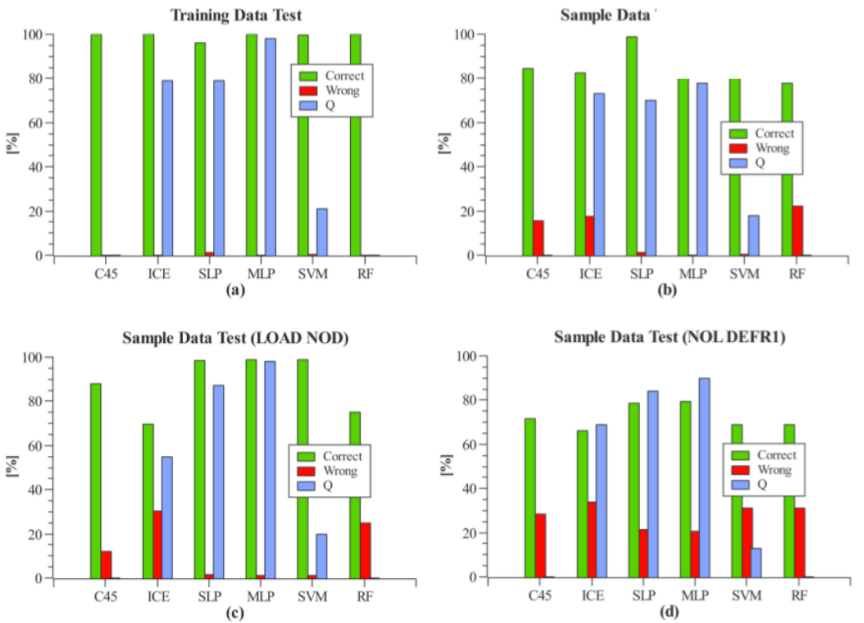

Force
x
z
y
Sensor Pair
S_i
Force
$S = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_4 & S_5 & S_6 \\ S_7 & S_8 & S_9 \\ S_{10} & S_{11} & S_{12} \end{bmatrix}^{x,y}$

(c)

Ressourcen

ML	Parameter	Learning Time	Modelsize (Bytes)
C45	-	8s	4k
ICE	ϵ=0.01	100ms	16k
SLP	$iter$=1000	1s	190k
MLP[1]	$iter$=1000, $layers_{\mathsf{hidden}} = [5]$	2s	210k
MLP[2]	$iter$=20000, $layers_{\mathsf{hidden}} = [5]$	22s	210k
SVM	$iter$=1000, $kernel$={$type$: rbf, C:0.5, σ:0.1}	90s	260k
RF	$depth_{\mathsf{max}} = 10$, $trees = 5$	150ms	1.2M

Genauigkeit



Regressionsbäume

Classification and Regression Tree CART

Bisher gaben Entscheidungsbäume diskrete kategorische Werte oder intervallkodierte numerische Werte aus. Ausgabewerte die nicht in den Trainingsdaten enthalten waren können auch nicht ausgegeben werden. Es gibt keine Inter- und Extrapolation!

- Regressionsbäume können zwar auch nur eine diskrete Menge von Werten ausgeben, die aber nicht unmittelbar in den Trainingsdaten enthalten sein müssen (nur numerische Zielvariablen) und aus einer statistischen Analyse stammen

Ein Regressionsbaum ist ein Hybrid aus Regressionsfunktion und Entscheidungsbaum.

Breiman, Friedman, Olshen, and Stone (1984)

Regressionsbäume

- Ein CART gruppiert einzelne Dateninstanzen und Trainingsbeispiele in Gruppen und berechnet statistische Größen der Zielvariable: Mittelwert, Standardabewichung usw.
- **Jeder Knoten ist hier auch ein Zielknoten der diese statistischen Informationen der Zielvariablen liefert** und kann für die Beantwortung einzelner Fragen verwendet werden d.h.,
  - Der Einfluss von Variablen und deren Wertebereiche auf die Zielvariable ist unmittelbar ablesbar,
  - Pfade entlang des Baumes ergeben Variablenkonditionale (also wenn A und dann B dann ...)

Jeder Knoten des Baumes ist die Wurzel eines Teilbaums mit einer statistisch gegebenen Verteilung der Zielvariablewerte

Regressionsbäume

- Je tiefer ein Knoten sich im Baum befindet, desto geringer sollte die Streuung und Breite der verteilung der Ergebnisvariable sein

Bei kategorischen Zielvariablen wird entropiebasiert der Trainingsdatensatz geteilt und auf Teilbäume abgebildet. Das geht bei kontinuierlichen variablen nicht!

- Bei Regressionsbäumen wird i.A. der mittlere Teilungsfehler für einen gegebenen Teilungspunkt berechnet und verwendet (i.a. Binärbaum bei numerischen Eingabevariablen)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

wobei Y der vorgebenene Trainingswert ist und $\hat{Y}$ der berechnete (vorhergesagte) Zielvariabenwert ist.

Regressionsbäume

[14]

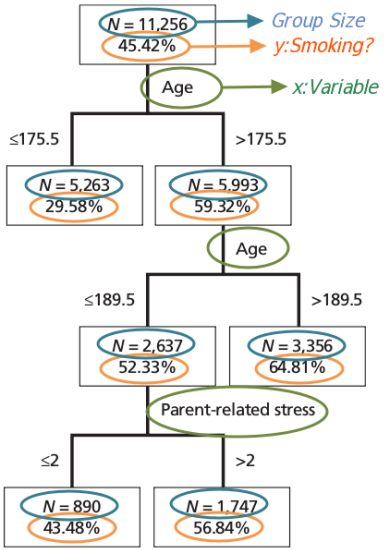


Abb. 12. CART mit der numerischen Zielvariable y:prob(Raucher) und verschiedenen Eingabevariablen (Attributen): Alter (Monate!), Elternstress (Score 0-5)

Regressionsbäume

Als eine statistische Methode gruppiert CART Individuen in eine Reihe von sich gegenseitig ausschließenden und repräsentativen Gruppen, die auf starke Zusammenhänge zwischen den unabhängigen Variablen basieren. CART ist eine effektive explorative statistische Technik.

Ohne auf einem speziellen statistisches Modell zu basieren, enthält CART keine komplexen mathematischen Gleichungen (die das statistische Modell beschreiben). Die Ergebnisse sind leicht zu interpretieren und zu verstehen.

Vertiefung:

[https://towardsdatascience.com/cart-classification-and-regression-trees-for-clean-but-powerful-models-cc89e60b7a85]

[https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning-e4d7525d8047]

Algorithmus

Ziel: Mit jeder Ebene/Knoten die "Unordnung" (Ungleichvewrteilung, Impurity) der Datenverteilung mit Bezug zu der Zielvariable zu reduzieren

- Es wird wieder die Entropie ε als Maß für die Ungleichverteilung herangezogen (der Zielvariable)

Der Grad der Verringerung der Ungleichverteilung, der mit der Partitionierung eines übergeordneten Knotens in zwei untergeordnete Knoten verbunden ist, wird berechnet als [14]:

$$\Delta = \epsilon(\tau) - \epsilon(\tau_L)\frac{n_{l1}}{n_{l1}+n_{l2}} - \epsilon(\tau_R)\frac{n_{r1}}{n_{r1}+n_{r2}}$$

wobei ε(τ) die Entropie des Elternknoten ist und n_i die Verteilung der Variblenwerte.

Baumkompaktierung

- Viele Baumstrukturen können nach dem Training vereinfacht werden ⇒ **Tree Pruning**
  - Dabei können auf gleicher Ebene Teilungsknoten und Blätter zusammengafasst werden (**Redundanzen**)
  - Aber auch Reduktion ganzer Teilbäume ist möglich (**Komplexität**)
- Man unterscheidet Pre- und Postkompaktierung

[https://www.kdnuggets.com/2022/09/decision-tree-pruning-hows-whys.html]

Pre-Kompaktierung

Eigentlich Verhinderung von überkomplexen Bäumen!

Die Vorbeschneidungstechnik von Entscheidungsbäumen besteht darin, die Hyperparameter vor der Trainingspipeline zu optimieren. Es beinhaltet die Heuristik, die als 'frühes Stoppen' bekannt ist und das Wachstum des Entscheidungsbaums stoppt - und **verhindert, dass er seine volle Tiefe erreicht**.

Es stoppt den Baumbildungsprozess, um zu vermeiden, dass Blätter mit kleinen Probengrößen (wenige Trainingsbeispiele) produziert werden. Während jeder Phase der Aufteilung des Baums wird der Kreuzvalidierungsfehler überwacht. Wenn der Wert des Fehlers nicht mehr abnimmt (keine Verbesserung), wird das Wachstum des Entscheidungsbaums gestoppt.

Pre-Kompaktierung

Die Hyperparameter, die für ein frühzeitiges Stoppen und Verhindern einer Überanpassung eingestellt werden können, sind u.A.:

`max_depth, min_samples_leaf und min_samples_split`

Dieselben Parameter können auch zum Abstimmen verwendet werden, um ein robustes Modell zu erhalten.

Aber: Ein frühzeitiges Anhalten kann auch zu einer Unteranpassung führen kann!

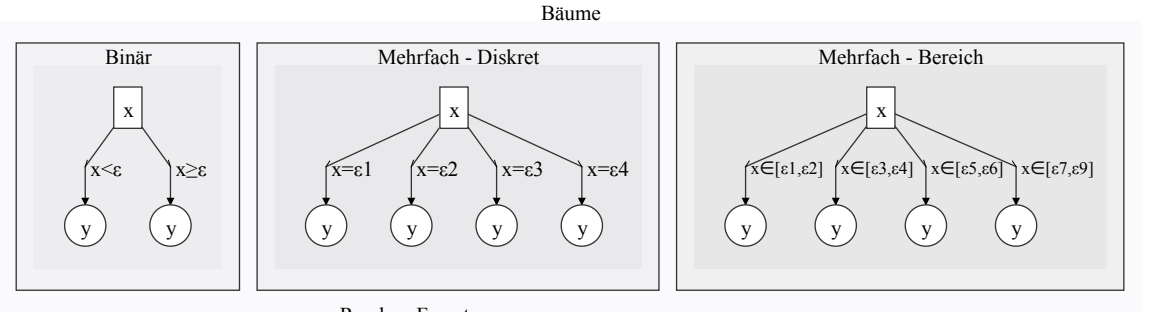
Post-Kompaktierung

Post-Pruning bewirkt das Gegenteil von Pre-Pruning und ermöglicht es dem Entscheidungsbaummodell, seine volle Tiefe zu erreichen. Sobald das Modell seine volle Tiefe erreicht hat, werden Äste entfernt, um eine Überanpassung des Modells zu verhindern.

Der Algorithmus partitioniert die Daten weiterhin in kleinere Teilmengen, bis die endgültigen erzeugten Teilmengen in Bezug auf die Ergebnisvariable ähnlich sind. Wenn jedoch ein neuer Datenpunkt eingeführt wird, der sich von den gelernten Daten unterscheidet, wird er möglicherweise nicht gut vorhergesagt.

- "Cost Complexity Pruning" ist eine gängige parametrisierbare Methode um die Komplexität von Bäumen iterativ zu reduzieren
- C4.5/ID3 bieten kaum eingebaute Möglichkeiten die Komplexität und Redundanzen zu reduzieren
- C5.0 bietet Kompaktierung zur Trainingszeit
- ICE kann Blattknoten zusammenfassen (mit fusionierten Intervallen)

Zusammenfassung



C5.0 Verfahren

- Erhältlich in *R* https://cran.r-project.org/web/packages/C50/vignettes/C5.0.html
- Einzelheiten: https://www.geeksforgeeks.org/c5-0-algorithm-of-decision-tree/

Schlüsselkonzepte des C5.0-Algorithmus

- Das MDL-Konzept (Minimum Description Length) beruht darauf, dass Modelle mit der kleinsten Kodierungslänge die Daten mit größerer Wahrscheinlichkeit effektiv erfassen.
- Konfidenzgrenzen: Um eine Überanpassung (Overfitting) zu vermeiden, werden Konfidenzgrenzen verwendet, um zu beurteilen, ob eine Knotenaufteilung statistisch signifikant ist.
- Beim Winnowing Verfahren werden weniger wichtige Regeln (Knoten) aus einem Entscheidungsbaum entfernt, um die Gesamtzahl der Regeln zu reduzieren.
- Die Baumteilung erfolgt wieder basierend auf Entropie und Informationsgewinn

C5.0 Verfahren

Pruning

Beim Beschneiden werden überflüssige oder redundante Zweige aus dem Entscheidungsbaum entfernt, um seine Genauigkeit und Generalisierungsfähigkeit zu erhöhen. Wenn ein Entscheidungsbaum genau mit den Trainingsdaten übereinstimmt, aber Schwierigkeiten hat, auf neue Fälle zu verallgemeinern, spricht man von **Überanpassung**. Durch das Beschneiden werden überflüssige Äste entfernt, die weniger für die Generalisierung als vielmehr für die Anpassung des Trainingssatzes wichtig sind.

Winnowing

Eine Methode namens Winnowing wird verwendet, um verrauschte oder unnötige Merkmale zu finden und zu entfernen, die die Qualität eines Entscheidungsbaums verschlechtern könnten. Dies beinhaltet die Bewertung des Informationsgewinns jedes Attributs und die Entfernung derjenigen Attribute, die wenig zur gesamten Entropiereduktion beitragen.

- Um festzustellen, ob der Informationsgewinn eines Attributs statistisch signifikant ist, verwendet der C5-Algorithmus einen Signifikanztest.
- Der Entscheidungsbaum verliert Attribute, die dieses Kriterium nicht erfüllen.

Zusammenfassung

- Entscheidungsbäume sind für primär die Klassifikation von kategorischen und sekundär für numerische Zielvariablen geeignet
- Mit Ausnahme von CART liefern EB nur Werte der Zielvariablen die im Training enthalten waren
- Numerische Zielvariablen müssen intervallkodiert werden (mit Ausnahme von CART).
- ID3/C4.5 Lerner können numerische und kategorische Eingabevariablen (Attribute) verwenden
  - Eine Attributvariable ist ein Teilungspunkt
- C5.0 ist die Quintessenz
- Rauschen auf Sensordaten muss durch "Unsicherheitsintervall" und Intervallarithmetik behandelt werden (und bei CART durch Standardabweichung)
- Vergleich mit anderen Lernverfahren zeigt gute Ergebnisse (je nach Problem)