

# ID3 und C4.5

---

Carsten Gips (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

# Wie Attribute wählen?

Erinnerung: CAL2/CAL3

- Zyklische Iteration durch die Trainingsmenge
- Ausschließlich aktuelles Objekt betrachtet
- Reihenfolge der “richtigen” Attributwahl bei Verzweigung unklar

=> Betrachte stattdessen die **komplette** Trainingsmenge!

## Erinnerung Entropie: Maß für die Unsicherheit

- Entropie  $H(S)$  der Trainingsmenge  $S$ : Häufigkeit der Klassen zählen
- Mittlere Entropie nach Betrachtung von Attribut  $A$

$$R(S, A) = \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

- Informationsgewinn durch Betrachtung von Attribut  $A$

$$\begin{aligned} \text{Gain}(S, A) &= H(S) - R(S, A) \\ &= H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \end{aligned}$$

=> Je kleiner  $R(S, A)$ , um so größer der Informationsgewinn

# Informationsgewinn: Kriterium zur Auswahl von Attributen

- 1) Informationsgewinn für alle Attribute berechnen
- 2) Nehme Attribut mit größtem Informationsgewinn als nächsten Test

Nr.	$x_1$	$x_2$	$x_3$	$k$
1	0	0	0	A
2	1	0	2	A
3	0	1	1	A
4	1	1	0	B
5	0	1	1	B
6	0	1	0	A

$$H(S) = 0.92 \text{ Bit}$$

$$\text{Gain}(S, x_1) = 0.92 - 0.87 = 0.05 \text{ Bit}$$

$$\begin{aligned}\text{Gain}(S, x_2) &= 0.92 - 2/6 \cdot 0 - 4/6 \cdot 1 \\ &= 0.25 \text{ Bit}\end{aligned}$$

$$\begin{aligned}\text{Gain}(S, x_3) &= 0.92 - 3/6 \cdot 0.92 - 2/6 \cdot 1 - 1/6 \cdot 0 \\ &= 0.13 \text{ Bit}\end{aligned}$$

Informationsgewinn für  $x_2$  am höchsten  $\Rightarrow$  wähle  $x_2$  als nächsten Test

# Entscheidungsbaumlerner ID3 (Quinlan, 1986)

```
def ID3(examples, attr, default):  
    # Abbruchbedingungen  
    if examples.isEmpty(): return default  
    if examples.each(class == A): return A # all examples have same class  
    if attr.isEmpty(): return examples.MajorityValue()  
  
    # Baum mit neuem Test erweitern  
    test = MaxInformationGain(examples, attr)  
    tree = new DecisionTree(test)  
    m = examples.MajorityValue()  
    for v_i in test:  
        ex_i = examples.select(test == v_i)  
        st = ID3(ex_i, attr - test, m)  
        tree.addBranch(label=v_i, subtree=st)  
    return tree
```

## Beobachtung: Gain ist bei mehrwertigen Attributen höher

- Faire Münze:
  - Entropie =  $H(\text{Fair}) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \text{ Bit}$
- 4-seitiger Würfel:
  - Entropie =  $H(\text{Dice}) = -4 \cdot (0.25 \log_2 0.25) = 2 \text{ Bit}$

=> Gain ist bei mehrwertigen Attributen höher

Normierter Informationsgewinn:  $\text{Gain}(S, A) \cdot \text{Normalisation}(A)$

$$\text{Normalisation}(A) = \frac{1}{\sum_{v \in \text{Values}(A)} p_v \log_2 \frac{1}{p_v}}$$

# Beispiele zur Normierung bei C4.5

- Faire Münze:
  - Entropie =  $H(\text{Fair}) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \text{ Bit}$
  - Normierung:  $1 / (0.5 \log_2(1/0.5) + 0.5 \log_2(1/0.5)) = 1 / (0.5 \cdot 1 + 0.5 \cdot 1) = 1$
  - Normierter Informationsgewinn:  $\text{Gain}(S, A) \cdot \text{Normalisation}(A) = 1 \text{ Bit} \cdot 1 = 1 \text{ Bit}$
- 4-seitiger Würfel:
  - Entropie =  $H(\text{Dice}) = -4 \cdot (0.25 \log_2 0.25) = 2 \text{ Bit}$
  - Normierung:  $1 / (4 \cdot 0.25 \log_2(1/0.25)) = 1 / (4 \cdot 0.25 \cdot 2) = 0.5$
  - Normierter Informationsgewinn:  $\text{Gain}(S, A) \cdot \text{Normalisation}(A) = 2 \text{ Bit} \cdot 0.5 = 1 \text{ Bit}$

⇒ Normierung sorgt für fairen Vergleich der Attribute



- Entscheidungsbaumlerner **ID3**
  - Nutze *Information Gain* zur Auswahl des nächsten Attributs
  - Teile die Trainingsmenge entsprechend auf (“nach unten hin”)
- Verbesserung durch Normierung des *Information Gain*: **C4.5**

# LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.