

Entscheidungsbäume

Vorlesung 186.844

09.01.2017

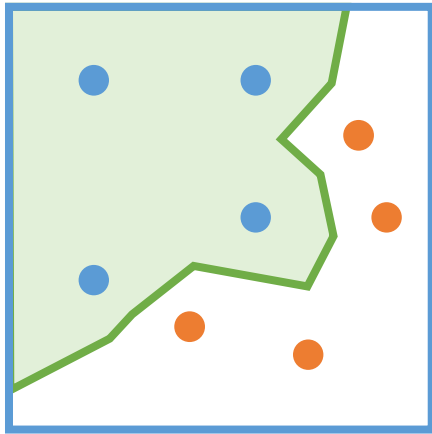
Überblick

- I. Was sind Entscheidungsbäume?
- II. Erstellen von Entscheidungsbäumen
- III. CART im Detail
- IV. Möglichkeiten und Einschränkungen

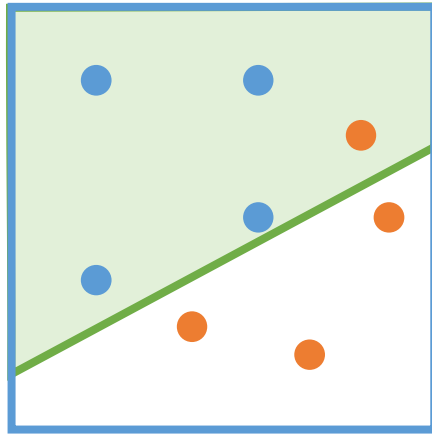
I. Was sind Entscheidungsbäume?

Entscheidungsgrenzen

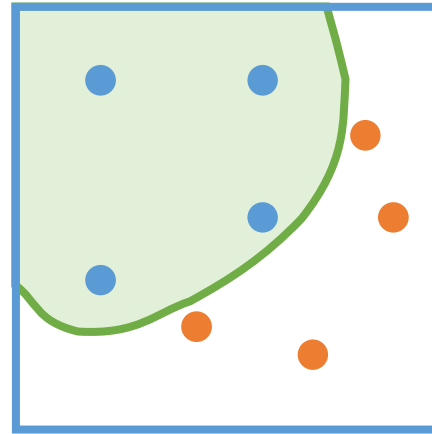
... gefunden oder definiert durch



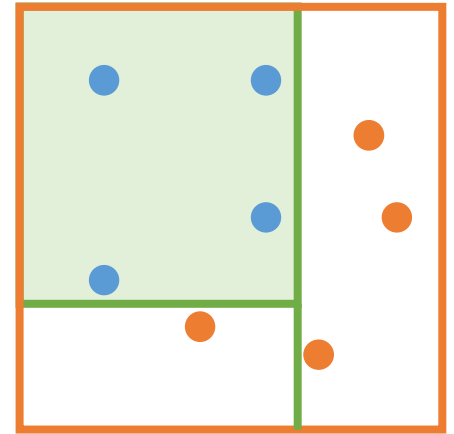
Nächster
Nachbar
(Nearest
Neighbor)



lineare
Diskriminan-
tenfunktion



nicht-lineare
Diskriminan-
tenfunktion



Entscheid-
ungsbäume

Merkmale

bisher:

- Merkmalsvektoren mit Elementen aus \mathbb{R}
- Distanz zwischen Merkmalsvektoren \rightarrow Metriken, Ähnlichkeitsmaße (similarity measures), etc.

$$\mathbf{x} \in \mathbb{R}^{n \cdot m \cdot f}$$



Listen von
Eigenschaften
z.B.: Fell = orange

in dieser Vorlesung:

- nominelle Merkmale
- Beschreibungen ohne Ordnung
- keine Metriken zur Distanzmessung

Nominelle Merkmale

- Beschreibung eines Musters durch Eigenschaften und ihre Ausprägungen
- Katze
 - Eigenschaften: Fell, Augen, Konstitution
 - Ausprägungen (Merkmalsvektor): orange, grün, fett
- Obst
 - Eigenschaften: Geschmack, Farbe, Form, Größe
 - Ausprägungen (Merkmalsvektor): sauer, rot, rund, klein



Entscheidungsbäume

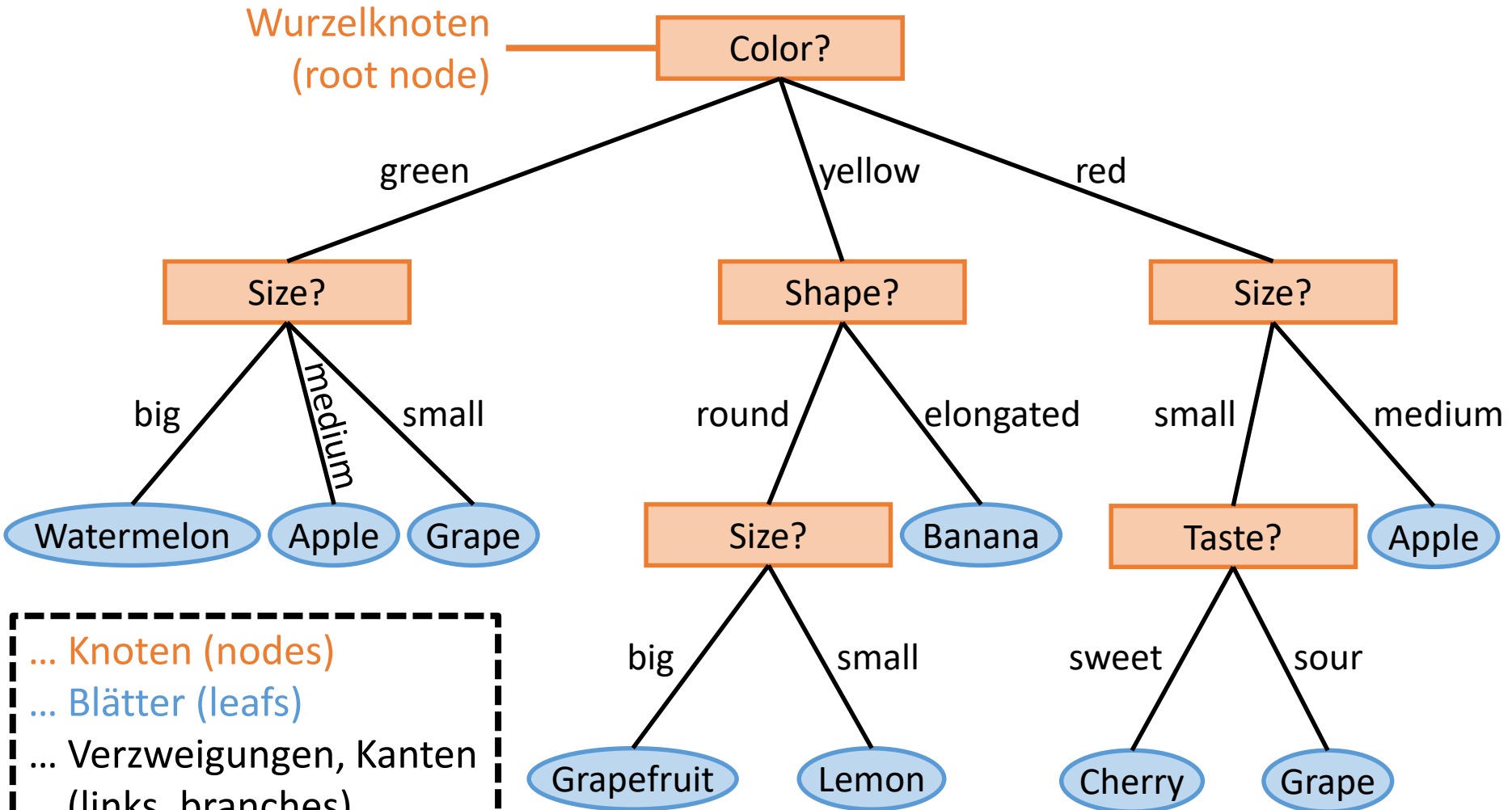
... Möglichkeit zur Klassifikation von Mustern mit nominellen Merkmalsvektoren

- intuitive **Klassifikation** durch eine Abfolge von **Fragen**
- nächste Frage ist **abhängig** von der vorherigen Antwort
- Antwort: ja/nein, richtig/falsch oder eine konkrete Ausprägung



Entscheidungsbäume

Wurzelknoten
(root node)



... Knoten (nodes)

... Blätter (leafs)

... Verzweigungen, Kanten
(links, branches)

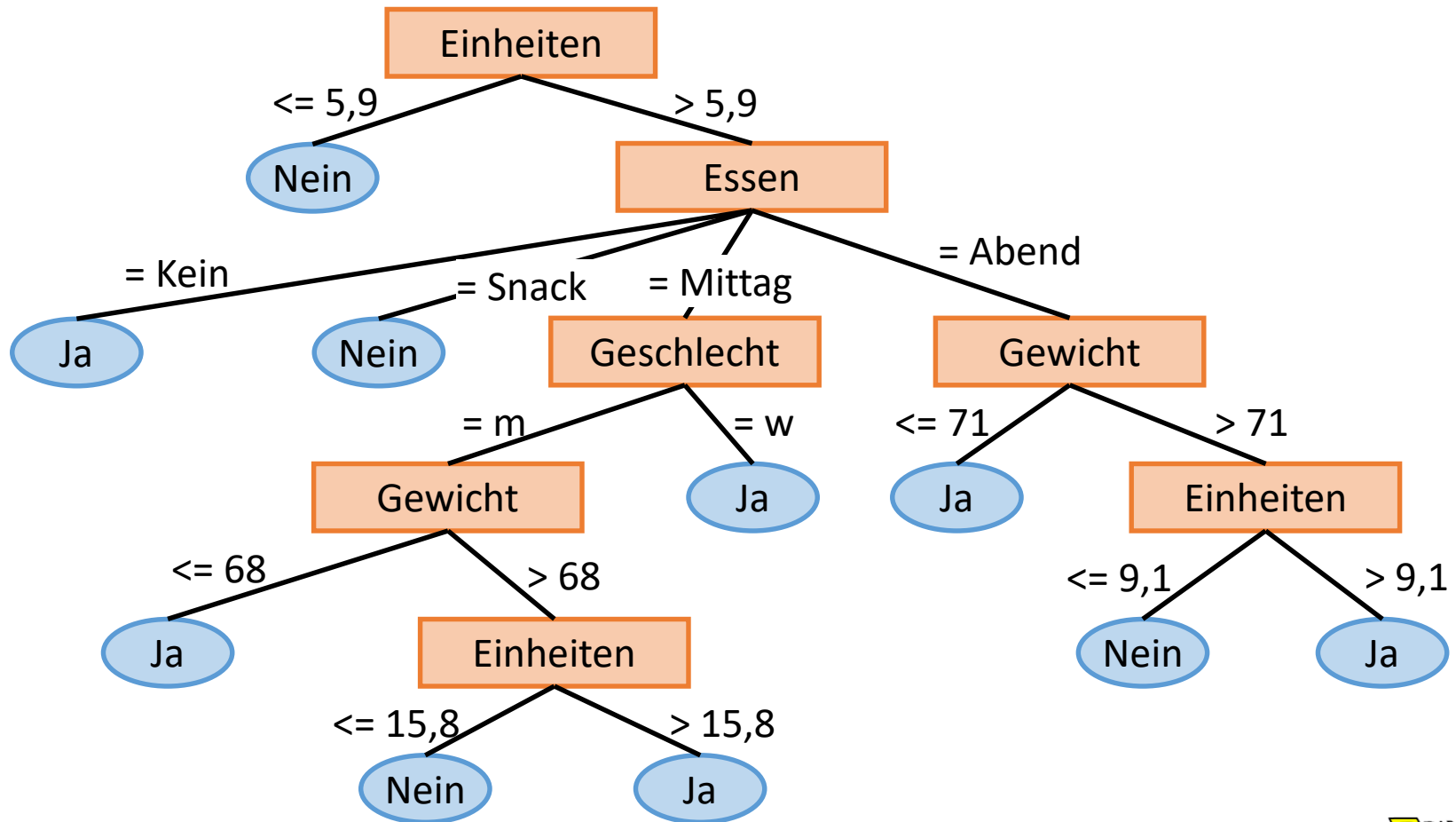
Entscheidungsbäume

... Besoffenheitstest nach Padraig Cunningham

Eigenschaft	Ausprägung
Alter	in Jahren
Geschlecht	männlich / weiblich
Pause	in Minuten
Dauer	in Stunden
Gewicht	in kg
Größe	in cm
Essen	letztes Essen: Kein, Snack, Mittag, Abend
Einheiten	wieviel getrunken wurde

Entscheidungsbäume

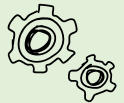
... Besoffenheitstest nach Padraig Cunningham



Klassifikation

... mit Entscheidungsbäumen

- Klassifikation eines Musters beginnt immer bei **Frage des Wurzelknotens**
- abhängig von Antwort (Ausprägung) folgt man entsprechender **Verbindung zu einem Folgeknoten**
- es muss **eine Entscheidung** getroffen werden → nur einer Verbindung wird gefolgt
- dann wird Entscheidung für „**Frage**“ **des Folgeknotens** gefällt, usw...
- bis man bei einem **Blattknoten** ankommt → keine weiteren Fragen
- Blattknoten ist einer **Klasse** zugeordnet → Muster wird zugewiesen



Vorteile

... von Entscheidungsbäumen

- einfach zu **interpretieren**
- logische **Strukturen** → Pfad vom Wurzelknoten zum Blatt → Klassifikation eines Musters
- Interpretation der **Klassen** selbst
 - z.B.: Apfel=(grün UND mittel) oder (rot UND mittel)
- in der Regel **schnelle** Klassifikation
- einfache Integration von **Expertenwissen**



II. Erstellen von Entscheidungsbäumen

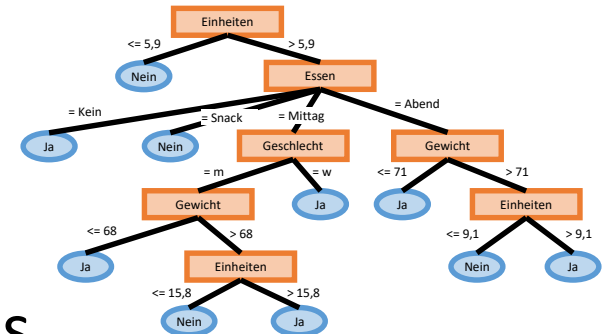
Wie wachsen Entscheidungsbäume?

Training

- Trainingsdatensatz \mathcal{D} mit Labels \rightarrow überwachtes Lernen
- Satz von Eigenschaften/Fragen

Ziel des Trainings

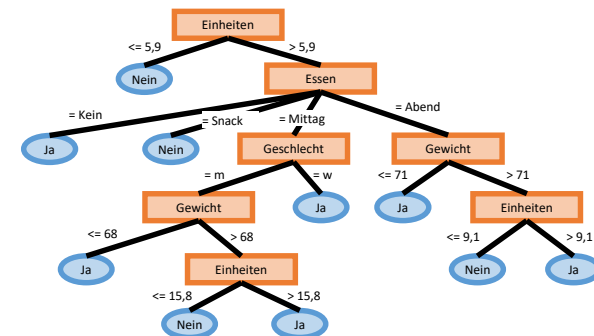
- **Automatische** Erzeugung eines Baums
- Unterteilung der Trainingsdaten in immer kleiner werdende **Untermengen**
- **Idealfall**: jede Untermenge ist rein \rightarrow beinhaltet Trainingsdaten von nur einer Klasse



Wie wachsen Entscheidungsbäume

Herausforderung

- normalerweise sind **Untermengen nicht rein**
- Untermengen bestehen aus einer **Mischung von Klassen**
- in jedem Zweig:
 - Soll **Verzweigung beendet** werden? Ist Fehler klein genug? → Teilung/Wachstum wird gestoppt
 - Soll eine **weitere Eigenschaft** gewählt werden? → Baum wächst weiter



Beispiel: Restaurantauswahl

9 Eigenschaften für eine Restaurantauswahl:

Eigenschaft	Erklärung
Alternative	Gibt es eine geeignete Restaurantalternative in der Nähe?
Bar	Gibt es eine bequeme Bar im Restaurant?
Fri/Sa	Ist es Freitag oder Samstag?
Hungrig	Ist man hungrig oder nicht?
Kunden	Wie viele andere Leute sind im Restaurant?
Preis	Preisspanne
Regen	Regnet es draußen?
Reservierung	Hat man eine Reservierung?
Typ	Art des Restaurants (französisch, thailändisch, Fast Food, italienisch)

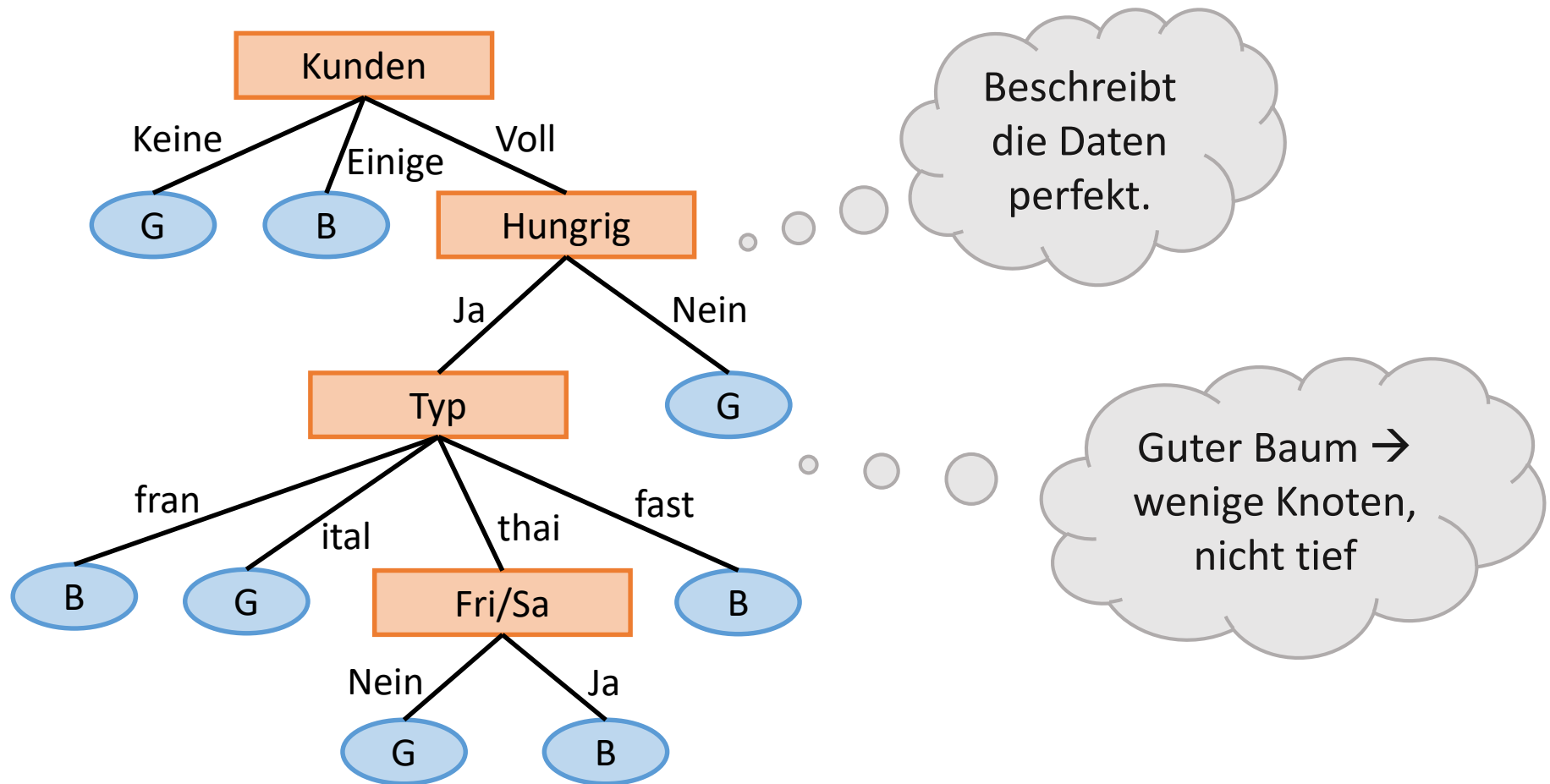
Beispiel: Restaurantauswahl

Trainingsdaten für 2 Klassen: Bleiben (B), Gehen (G)

Fall	Alt.	Bar	Fri/Sat	Hun.	Kun.	Preis	Reg.	Res.	Typ	Klasse
1	Ja	Nein	Nein	Ja	Einige	\$\$\$	Nein	Ja	fran	B
2	Ja	Nein	Nein	Ja	Voll	\$	Nein	Nein	thai	G
3	Nein	Ja	Nein	Nein	Einige	\$	Nein	Nein	fast	B
4	Ja	Nein	Ja	Ja	Voll	\$	Nein	Nein	thai	B
5	Ja	Nein	Ja	Nein	Voll	\$\$\$	Nein	Ja	fran	G
6	Nein	Ja	Nein	Ja	Einige	\$\$	Ja	Ja	ital	B
7	Nein	Ja	Nein	Nein	Keine	\$	Ja	Nein	fast	G
8	Nein	Nein	Nein	Ja	Einige	\$\$	Ja	Ja	thai	B
9	Nein	Ja	Ja	Nein	Voll	\$	Ja	Nein	fast	G
10	Ja	Ja	Ja	Ja	Voll	\$\$\$	Nein	Ja	ital	G
11	Nein	Nein	Nein	Nein	Keine	\$	Nein	Nein	thai	G
12	Ja	Ja	Ja	Ja	Voll	\$	Nein	Nein	fast	B

Beispiel: Restaurantauswahl

Automatisch erzeugter Entscheidungsbaum:



CART

CART (= **C**lassification **A**nd **R**egression **T**rees) ist eine **iterative Methode** um Entscheidungsbäume zu erstellen/wachsen zu lassen. Abhängig von den Daten in einem Knoten wird: (1) der **Knoten als Blatt deklariert** und eine Klasse gewählt oder (2) ein Merkmal (Eigenschaft) gefunden, um die Daten in **kleinere Untermengen** zu teilen.



CART



... wirft sechs grundlegende Fragen auf:

1. Wie viele **Verzweigungen** sollen von einem Knoten ausgehen?
Wie viele Entscheidungsausgänge/Teilungen gibt es?
2. Welches **Kriterium** soll an einem Knoten getestet werden?
3. Wann soll ein Knoten **in ein Blatt umgewandelt** werden?
4. Wie kann man einen **Baum verkleinern** oder vereinfachen, wenn er zu „groß“ ist?
5. Welcher **Klasse** soll ein Blatt zugeordnet werden, wenn es nicht rein ist?
6. Wie geht man mit **fehlenden Daten** um?

III. CART im Detail

1. Verzweigungen

Jeder **Entscheidungsausgang** an einem Knoten wird als **Verzweigung** (split) bezeichnet, weil die Trainingsdaten aufgespalten/aufgeteilt werden.



Die Anzahl von Verzweigungen (Kanten) ausgehend von einem Knoten wird häufig als **Verteilungsfaktor** (branching factor) oder **Verteilungsverhältnis** (branching ratio) bezeichnet.

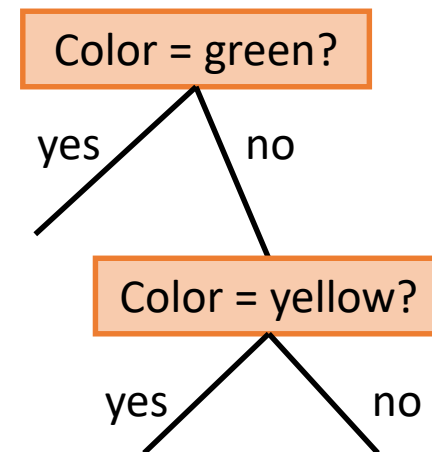
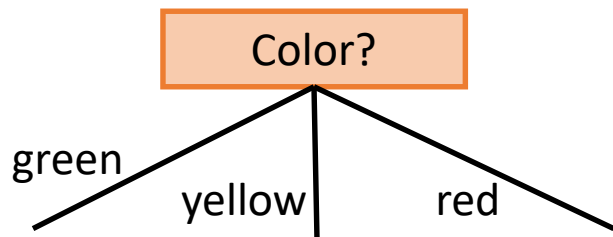


Verteilungsfaktor:

- abhängig vom Design des Baumes
- nicht notwendigerweise konstant in einem Baum

1. Verzweigungen

Jeder Baum kann als **binärer Entscheidungsbaum** dargestellt werden \rightarrow Verteilungsfaktor $B = 2$. Solch ein binärer Baum führt zum **gleichen Klassifikationsergebnis** wie der ursprüngliche Baum mit beliebigem Verteilungsfaktor B an den Knoten.

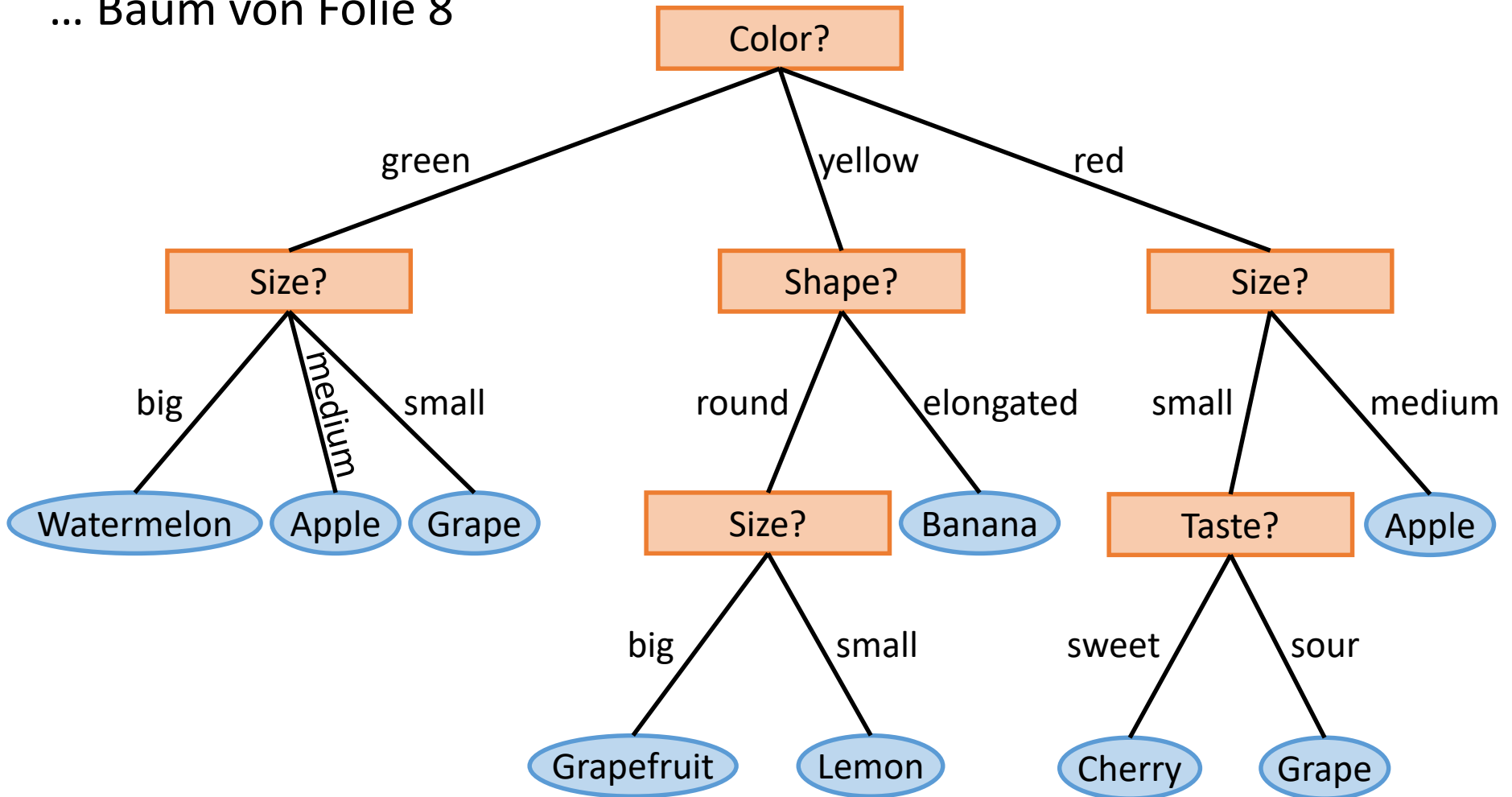


Binäre Entscheidungsbäume sind **einfacher zu trainieren** und werden deshalb häufig verwendet. CART erstellt binäre Bäume.



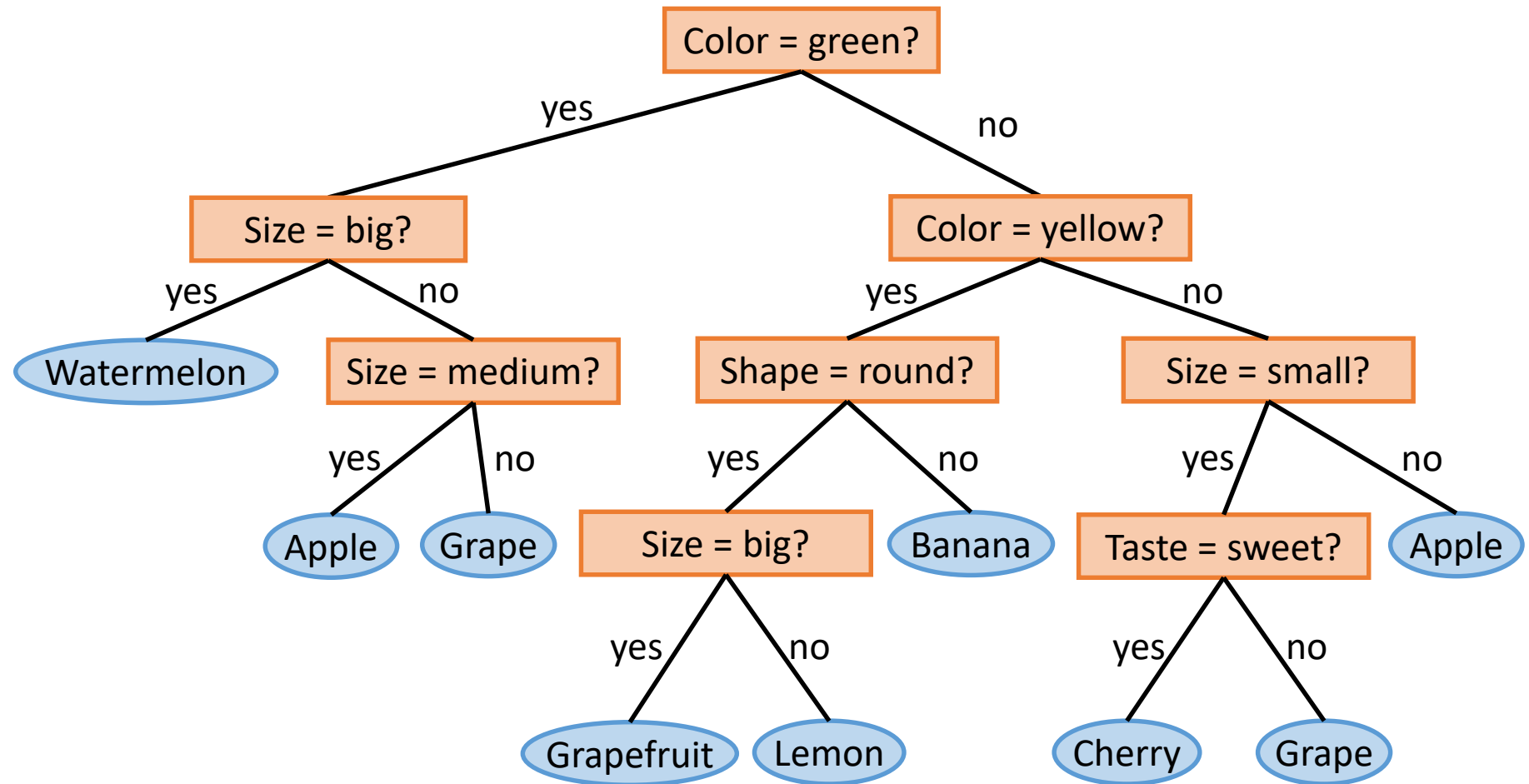
Beispiel

... Baum von Folie 8



Beispiel

... Baum von Folie 8 als Binärbaum



2. Kriterium

Ein Kriterium besteht aus einer oder mehreren **Eigenschaften** (Merkmale) und einer **Regel** (Heuristik) wie die Daten dadurch aufgeteilt werden.

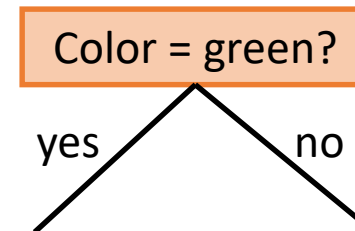
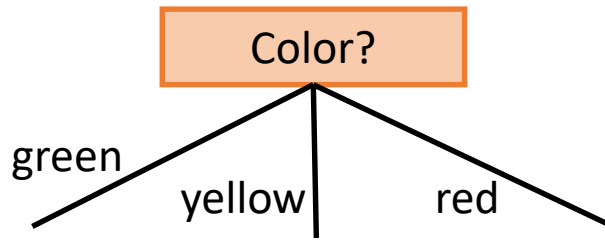


Ziel: Entscheidungsbaum soll so einfach wie möglich sein
→ wenige Knoten, flach

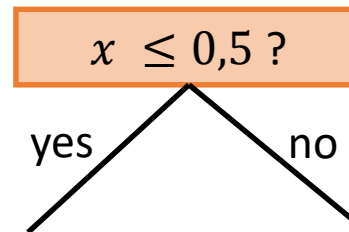
Welche Kriterien sollen in den Knoten verwendet werden?
In welcher Reihenfolge?

2. Beispiele für Kriterien

■ nominell

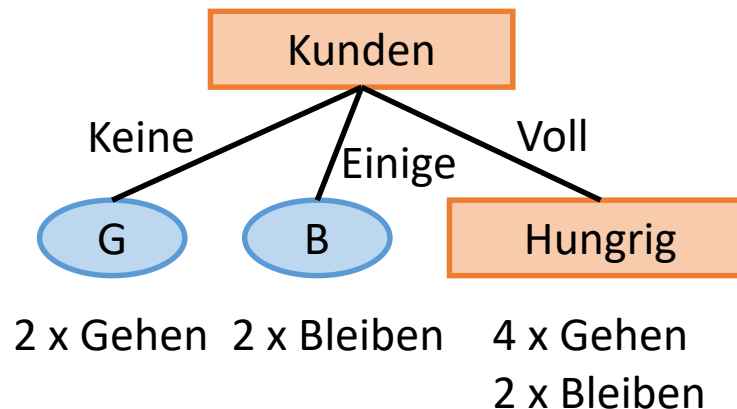


■ numerisch



2. Das perfekte Kriterium

Ein **perfektes Kriterium** führt zu Knoten die rein sind. Das bedeutet, dass die resultierenden Teilmengen nur Mitglieder einer Klasse beinhalten.



Wie kann man die Reinheit eines Knotens messen?

2. Unreinheit

In der Praxis wird die **Unreinheit** (impurity) eines Knoten definiert, weil es einfacher ist.

Sei $i(N)$ die Unreinheit eines Knotens N . Dann ist $i(N) = 0$, wenn alle Muster (Daten) die in N ankommen in die gleiche Klasse fallen. $i(N)$ ist am größten, wenn alle Klassen gleich häufig vorkommen.



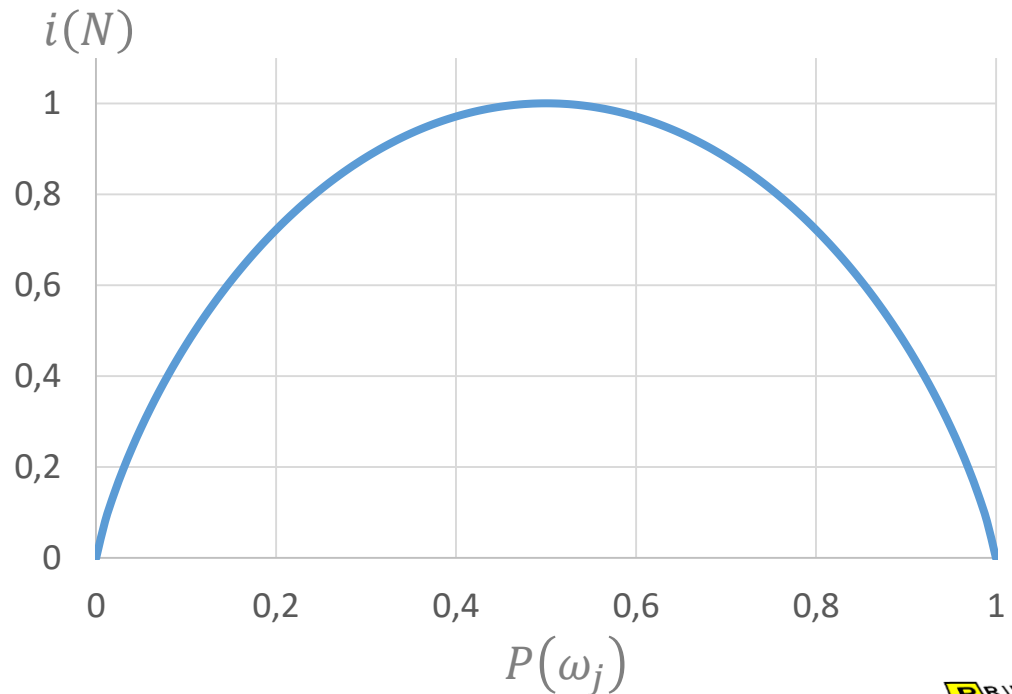
2. Entropie

Die Entropie (entropy) ist ein Maß mit dem man die Unreinheit wie folgt bestimmen kann: $i(N) = - \sum_j P(\omega_j) \log_2 P(\omega_j)$



wobei $P(\omega_j)$ die Häufigkeit der Klasse ω_j im Knoten N angibt.

z.B.: für zwei Klassen



2. Kriteriumsauswahl

Ziel: Wähle ein Kriterium, sodass die Unreinheit $i(N)$ eines Knotens N möglichst klein wird. Das bedeutet, dass die **Änderung der Unreinheit $\Delta i(N)$ maximiert** werden soll.

Für eine **binäre Verzweigung** ist $\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$, wobei N_L und N_R die resultierenden linken und rechten Knoten (Teilmengen) sind. P_L ist der Anteil der Muster der dem linken Knoten zugeordnet wird.



2. Beispiel

Ziel: Erstellen eines binären Entscheidungsbaumes für numerische Merkmale

Trainingsdaten:

$n = 16$ Muster

Merkmale x_1 und x_2

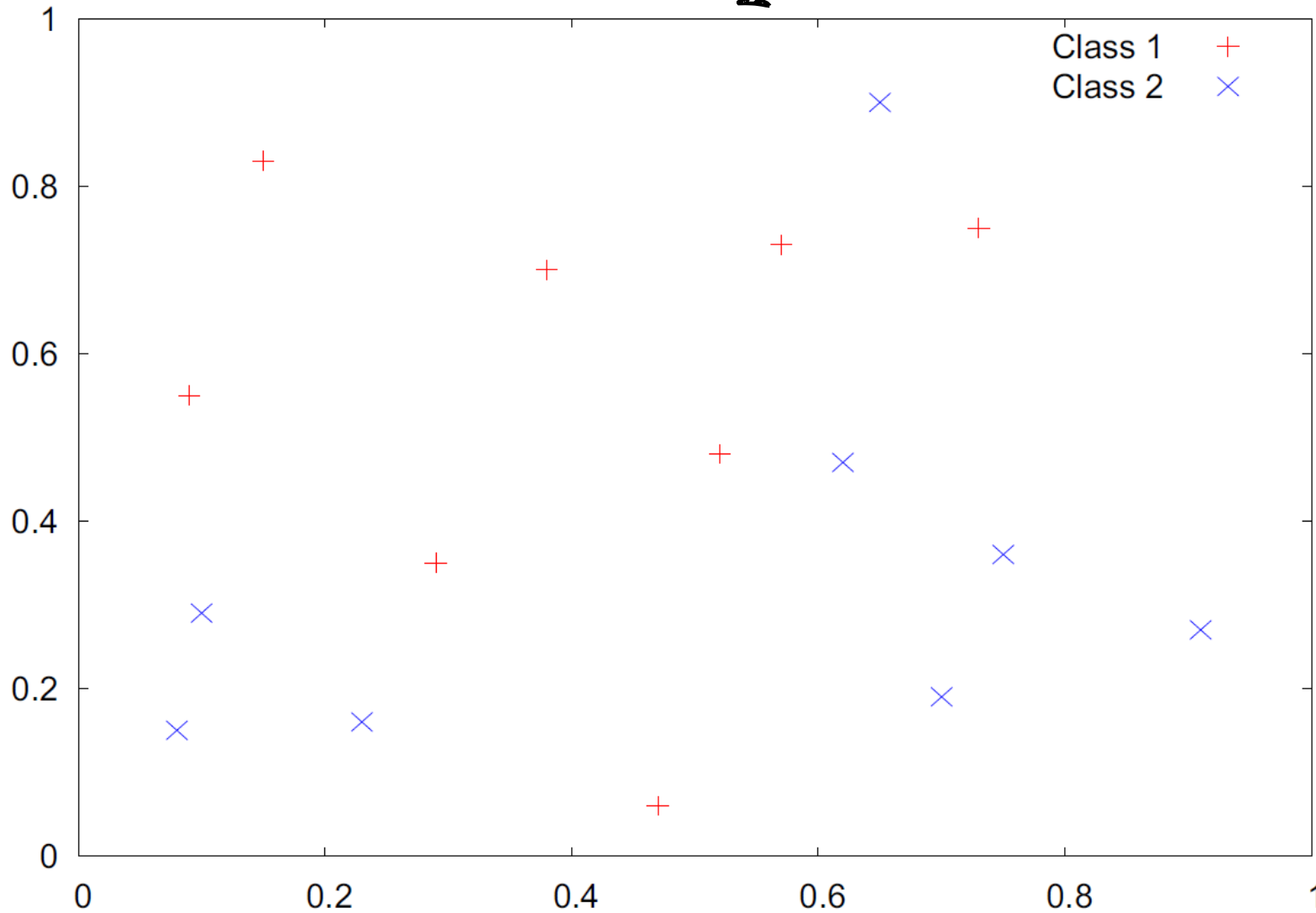
Klassen w_1 und w_2

Kriterien:

$$x_i < x_s$$

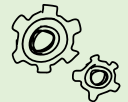
w_1		w_2	
x_1	x_2	x_1	x_2
0,15	0,83	0,10	0,29
0,09	0,55	0,08	0,15
0,29	0,35	0,23	0,16
0,38	0,70	0,70	0,19
0,52	0,48	0,62	0,47
0,57	0,73	0,91	0,27
0,73	0,75	0,65	0,90
0,47	0,06	0,75	0,36

2. Beispiel



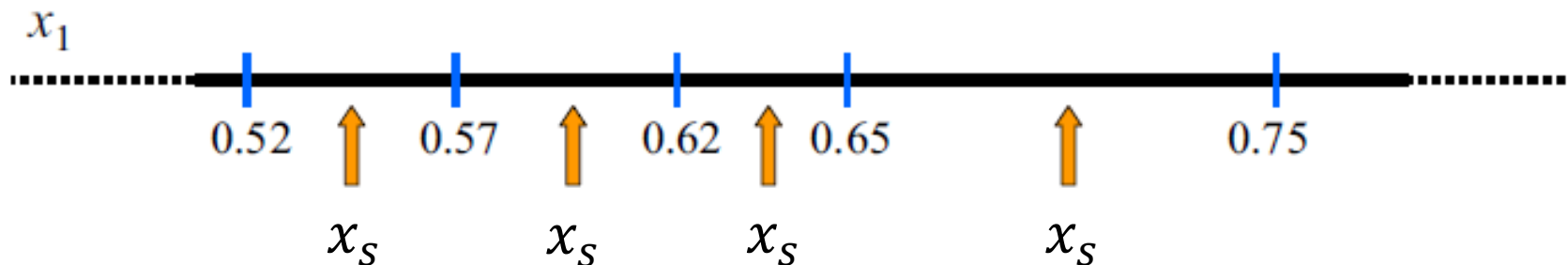
2. Beispiel

Berechnung der **Unreinheit des Wurzelknotens**:




$$i(N_0) = - \sum_{i=1}^2 P(\omega_i) \log_2 P(\omega_i) = - [0,5 \log_2 0,5 + 0,5 \log_2 0,5] = 1,0$$

Es gibt jeweils $n - 1 = 15$ Möglichkeiten, um x_1 und x_2 zu teilen.



Teilungsposition?

2. Beispiel

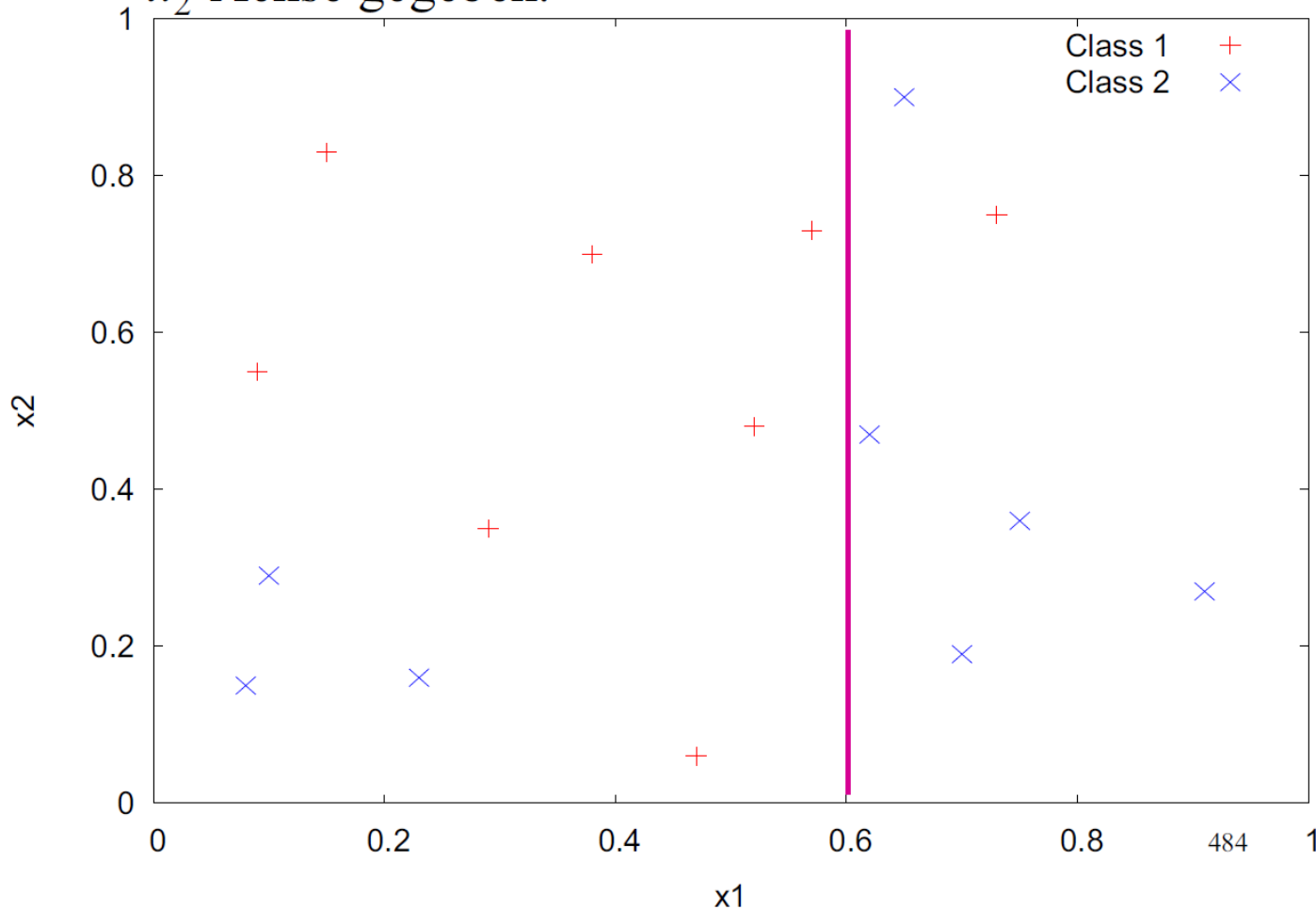
- berechne die **Änderung der Unreinheit** $\Delta i(N_0)$ für alle möglichen „Teilungspositionen“ (Kriterien) für x_1 und x_2 
- Teilungsposition x_s überall zwischen x_l und x_r möglich \rightarrow oft wird **Mittelpunkt** verwendet: $x_s = \frac{x_l + x_r}{2}$
- wähle Kriterium, dass Änderung der Unreinheit maximiert

Ergebnis:

- $\Delta i(N_0)$ ist am größten für x_1 zwischen 0,57 und 0,62
- daraus folgt $\rightarrow x_s \approx 0,60$

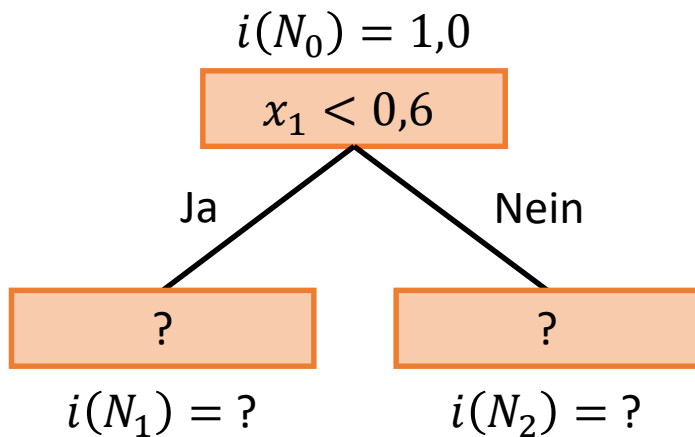
2. Beispiel

- Diese Trennung ist durch eine Linie parallel zur x_2 -Achse gegeben.



2. Beispiel

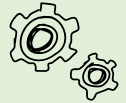
... Ergebnis nach 1. Verzweigung



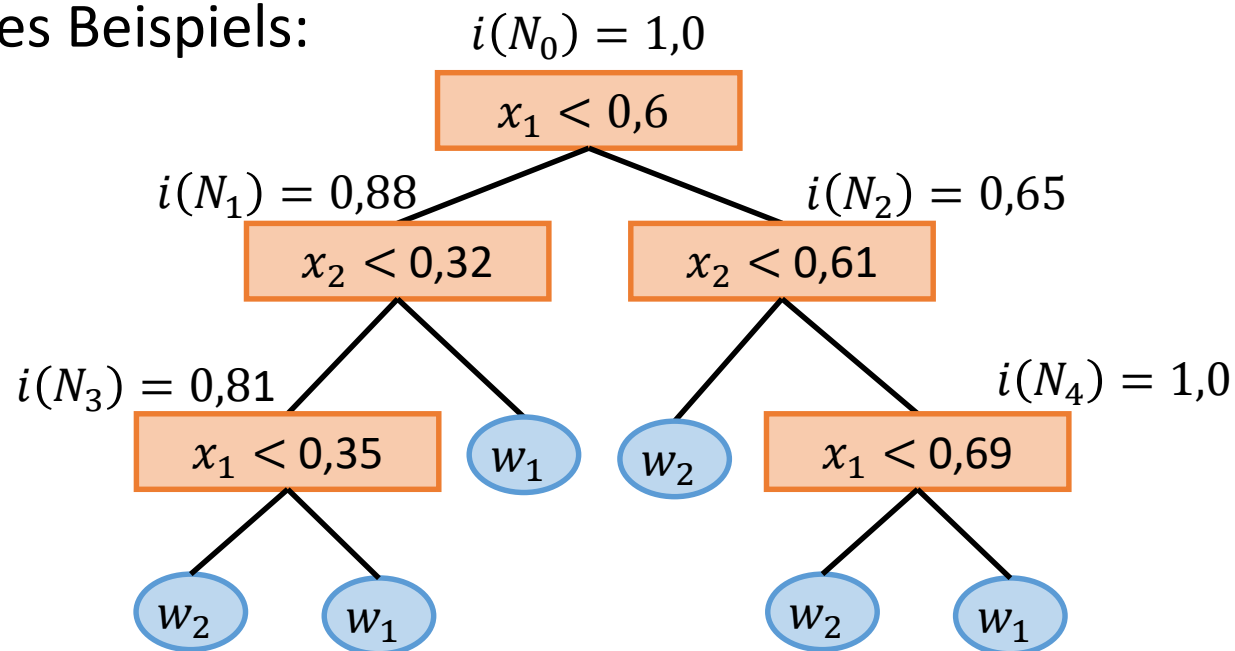
w_1		w_2	
x_1	x_2	x_1	x_2
0,15	0,83	0,10	0,29
0,09	0,55	0,08	0,15
0,29	0,35	0,23	0,16
0,38	0,70	0,70	0,19
0,52	0,48	0,62	0,47
0,57	0,73	0,91	0,27
0,73	0,75	0,65	0,90
0,47	0,06	0,75	0,36

2. Beispiel

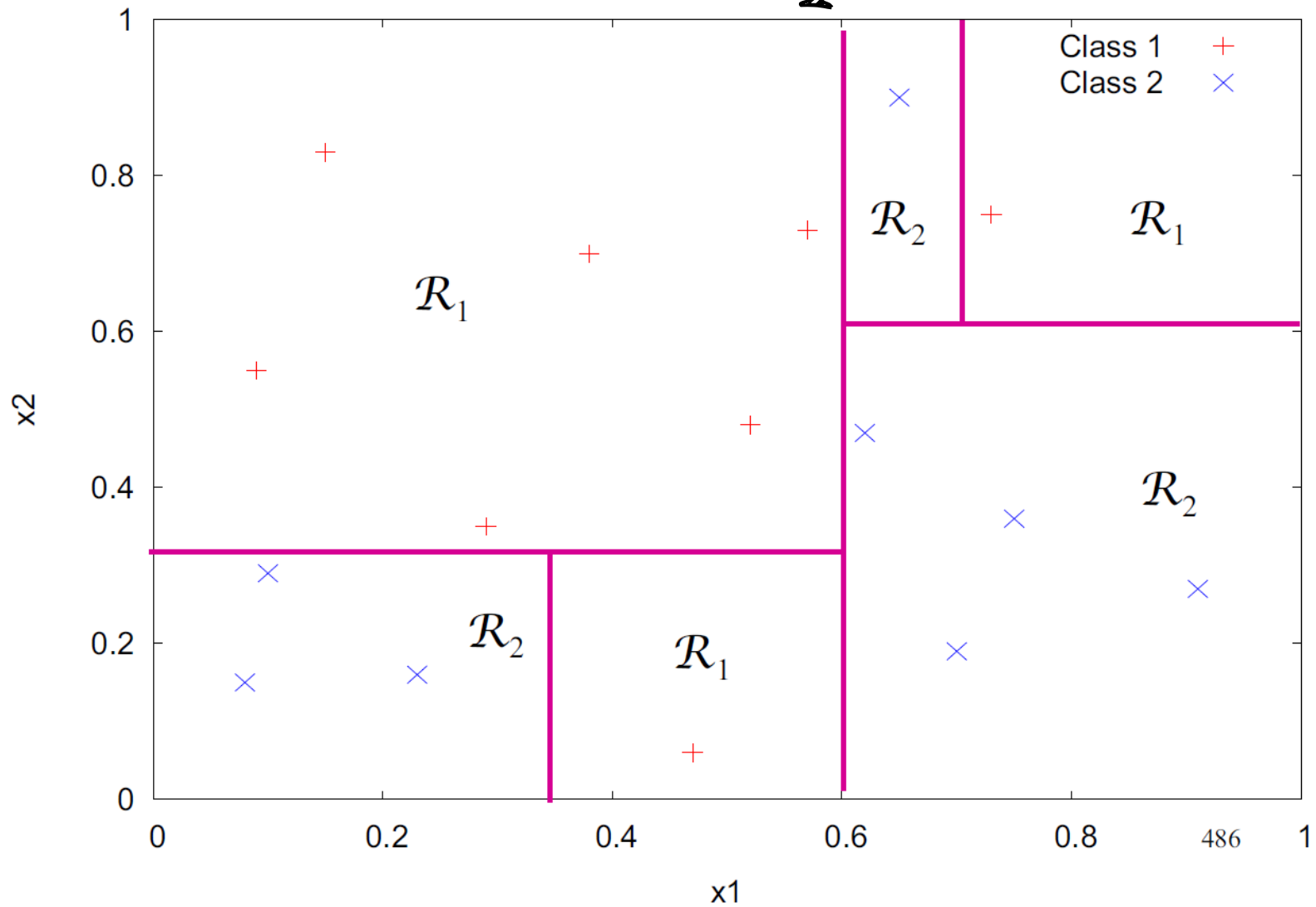
- die Knoten N_1 und N_2 sind nicht rein
- bestimme neue Kriterien für weitere Verzweigungen
- bis die Unreinheit aller Blätter 0 wird



Ergebnis des Beispiels:



2. Beispiel



3. Stoppen des Trainings

Fragen:

- Wann soll ein Knoten in ein Blatt umgewandelt werden?
- Wann soll die Trennung von Knoten gestoppt werden?

Probleme:

- **Overfitting** auf Trainingsdaten, wenn Training erst gestoppt wird, wenn alle Knoten rein sind
- Training sollte auch **nicht zu früh gestoppt** werden
- Entscheidungsbäume sind oft **sensibel auf kleine Unterschiede** in Trainingsdaten

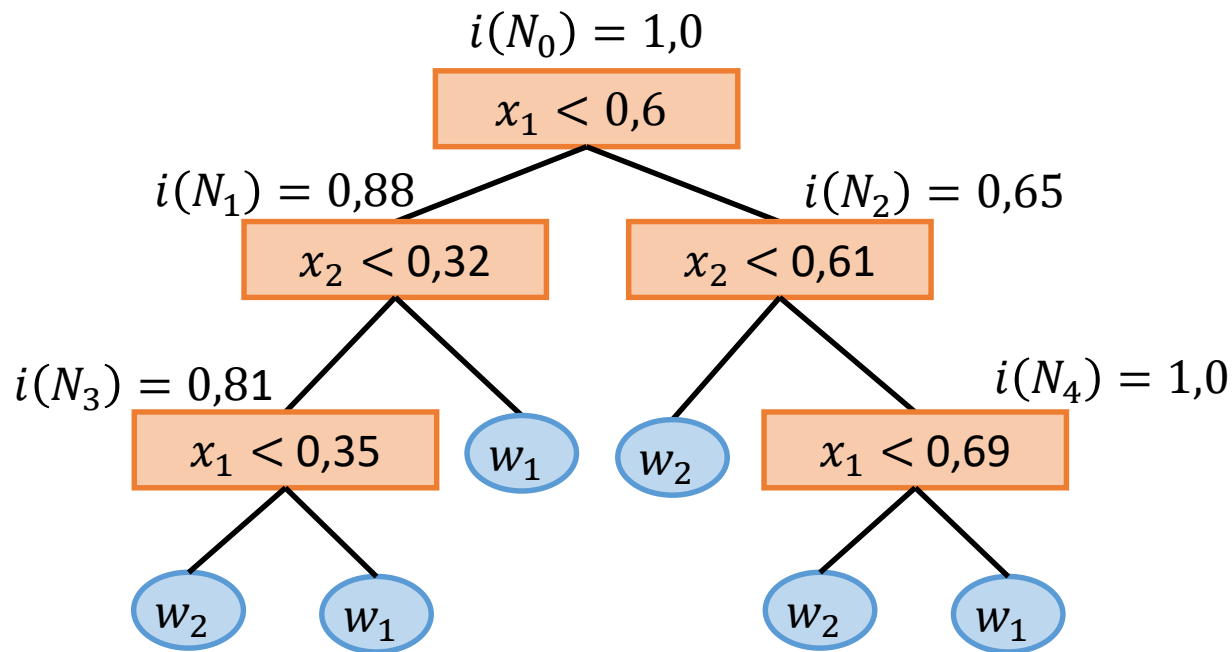
3. Beispiel

- numerisches Beispiel von vorher mit einem abgeänderten Wert

w_1		w_2	
x_1	x_2	x_1	x_2
0,15	0,83	0,10	0,29
0,09	0,55	0,08	0,15
0,29	0,35	0,23	0,16
0,38	0,70	0,70	0,19
0,52	0,48	0,62	0,47
0,57	0,73	0,91	0,27
0,73	0,75	0,65	0,90
0,47	0,06	0,75	0,32 0,36

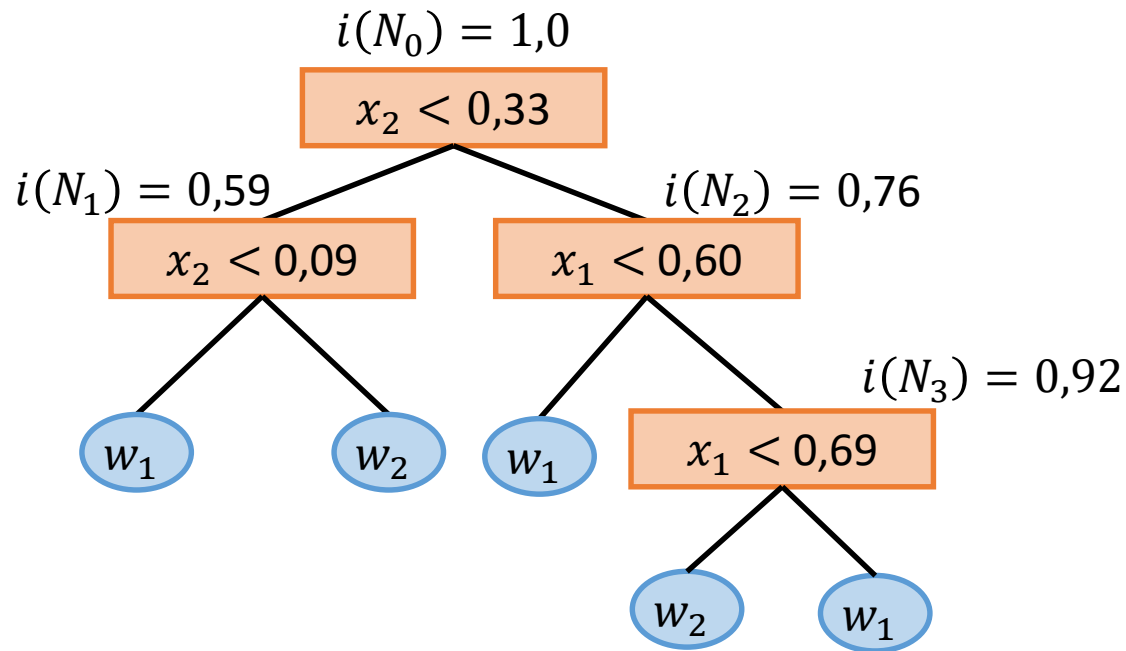
3. Beispiel

- ursprünglicher Entscheidungsbaum



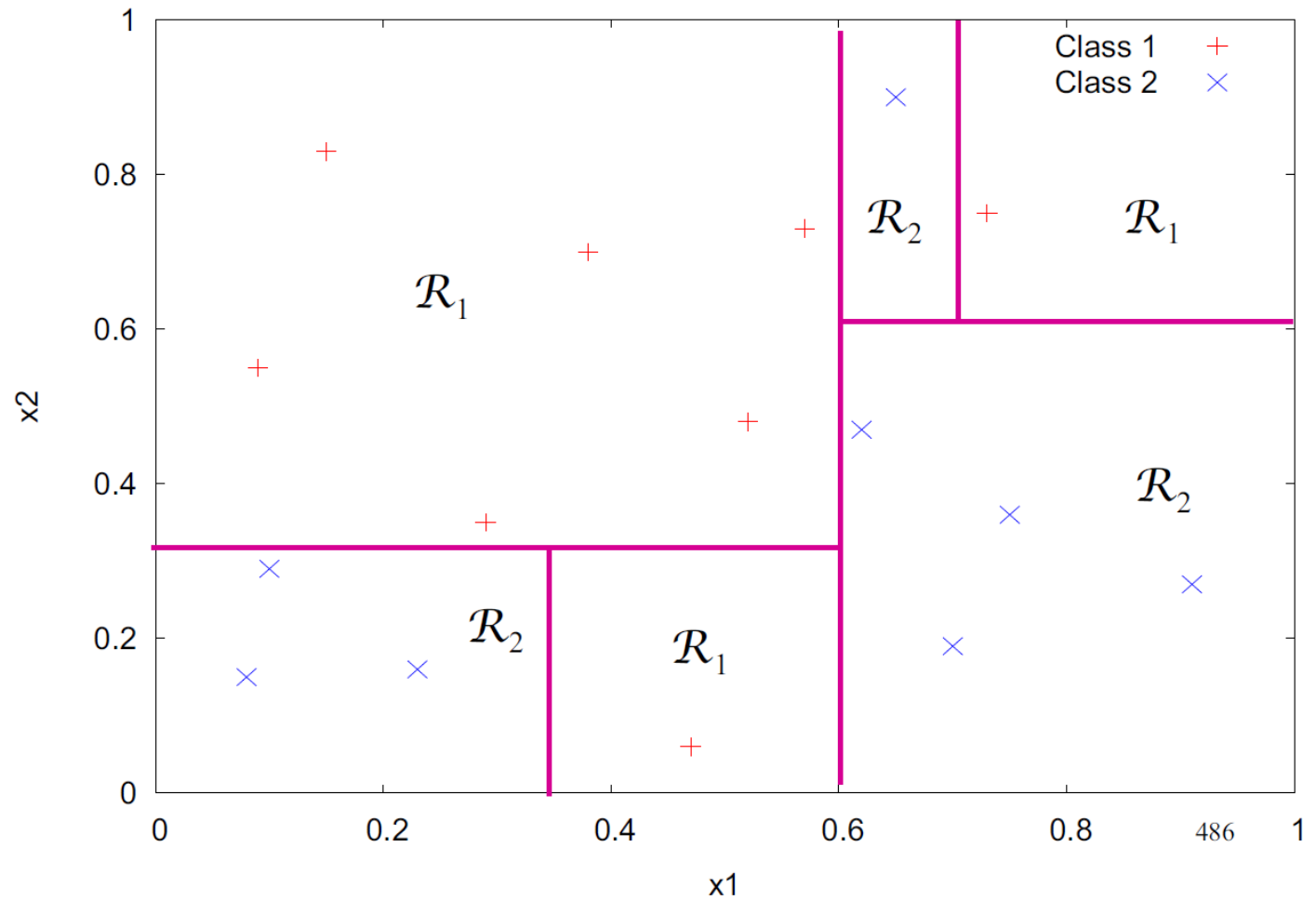
3. Beispiel

■ neuer Entscheidungsbaum



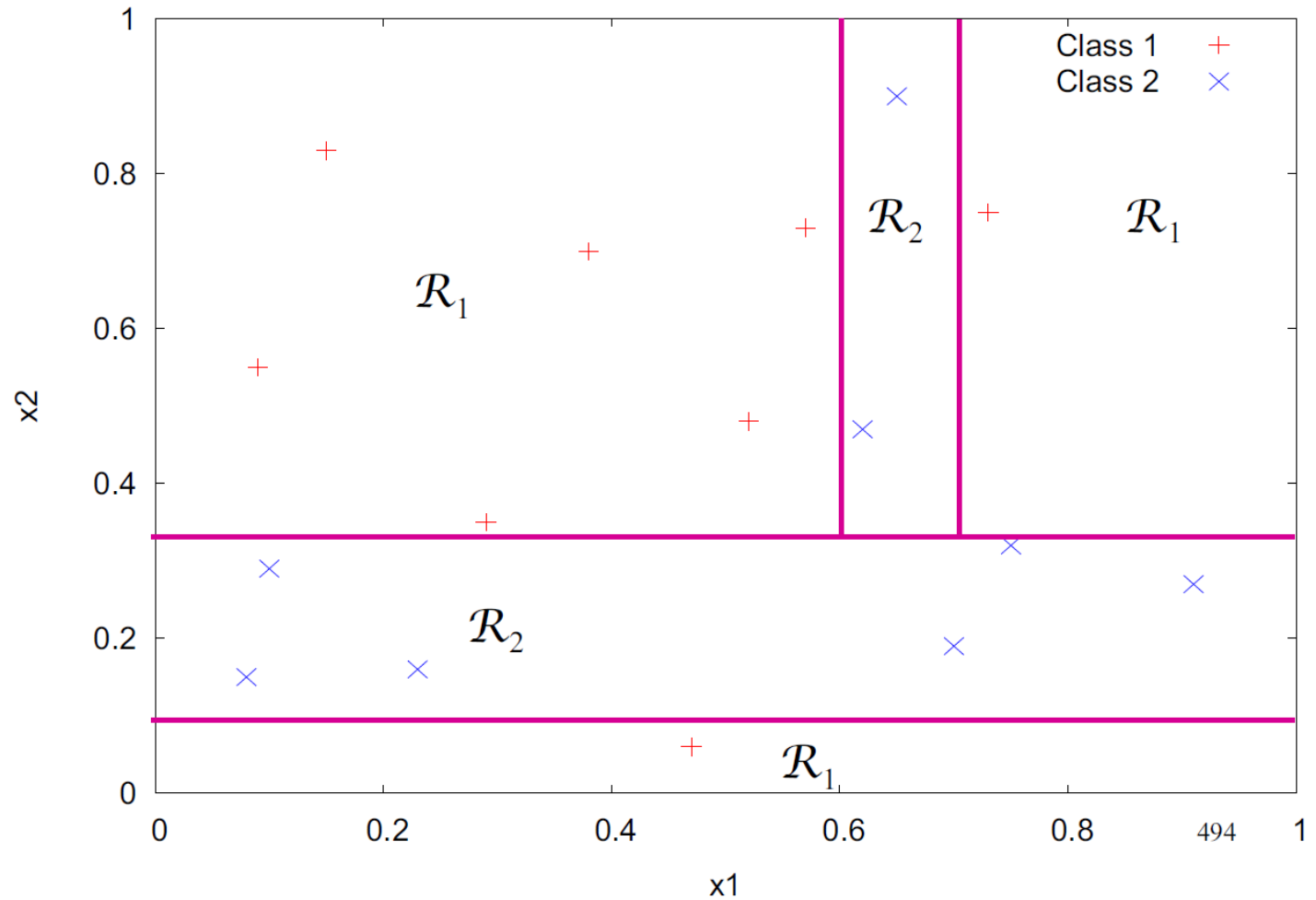
3. Beispiel

vorher ...



3. Beispiel

nachher ...



3. Stoppen des Trainings

... durch verschiedene Abbruchkriterien, z.B.:

Unabhängige Validierungsdaten (Testdaten)

- Training wird abgebrochen, wenn Fehler auf Validierungsdaten klein genug ist (Schwellwert)

Änderung der Unreinheit

- Knoten wird zu einem Blatt, wenn $\Delta i(N) < \beta$
- β ist schwierig festzulegen

Anzahl der Muster

- Knoten wird zu einem Blatt, wenn Anzahl der zugehörigen Muster unter einen Schwellwert fällt
- z.B.: 5% der Trainingsdaten

4. Pruning

Problem: Training wird auf Grund von **lokalen Abbruchkriterien** gestoppt. Es fehlt der sogenannte Weitblick. Dieses Phänomen wird als „**Horizon Effect**“ bezeichnet.

Lösung: Pruning

- Training wird gestoppt, wenn **Unreinheit aller Knoten null** ist
- Für **alle Paare** von Knoten (mit gemeinsamem Elternknoten):
 - überprüfe den **Einfluss der Knoten** auf die Unreinheit
 - falls **Einfluss klein** (Unreinheit wird nur leicht erhöht)
→ **entferne Knoten**, Elternknoten wird zu Blatt



4. Pruning

Informationen zum Pruning:

- Pruning **beginnt** meist mit den **Blatt-Knoten**
- **effiziente Algorithmen** können einen komplexen Teilbaum (subtree) direkt mit einem Blatt ersetzen
- Pruning verhindert den „**Horizon Effect**“
- komplexes Problem:
 - Abbruchkriterien sind vorzuziehen
 - es dauert zu lange bis alle Blätter eine Unreinheit von Null aufweisen
- einfaches Problem:
 - Pruning wird bevorzugt
 - Rechenaufwand vernachlässigbar

5. Klassenzugehörigkeit

... von Blättern

Problem: Für welche Klasse soll man sich entscheiden, wenn ein Blatt nicht rein ist?

Lösung: Das Blatt gehört zu jener Klasse die am häufigsten im Knoten vorkommt. D.h. die meisten Muster des Trainingsdatensatzes in dem Blatt waren dieser Klasse zugeordnet.

6. Fehlende Daten

... beim Training

Training

- **naive** Lösung: löschen der unvollständigen Muster
 - nur wenn ausreichend Trainingsdaten zur Verfügung stehen
- **„schlauere“** Lösung:
 - Berechnung von $i(N)$ und $\Delta i(N)$ auf den vorhandenen Informationen
 - z.B.: In Knoten N gibt es drei Muster mit zwei Merkmalen, wobei in einem Muster das zweite Merkmal fehlt. Um die beste Teilung zu finden, ermittelt man alle drei möglichen Teilungen basierend auf dem ersten Merkmal und nur zwei Teilungen für das zweite Merkmal.

IV. Möglichkeiten und Einschränkungen

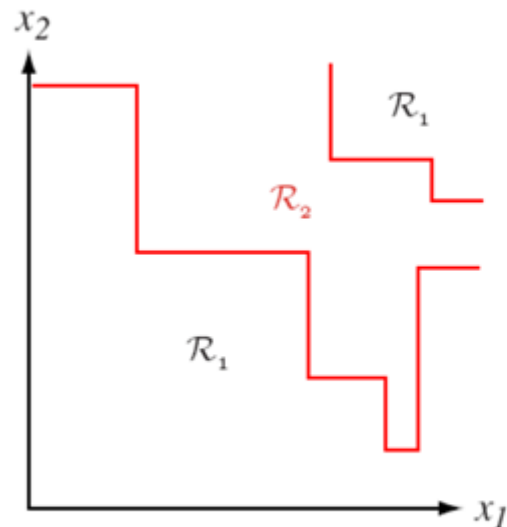
Entscheidungsgrenzen

Man kann **beliebige Entscheidungsgrenzen approximieren**, wenn die Größe des Entscheidungsbaumes ausreicht.

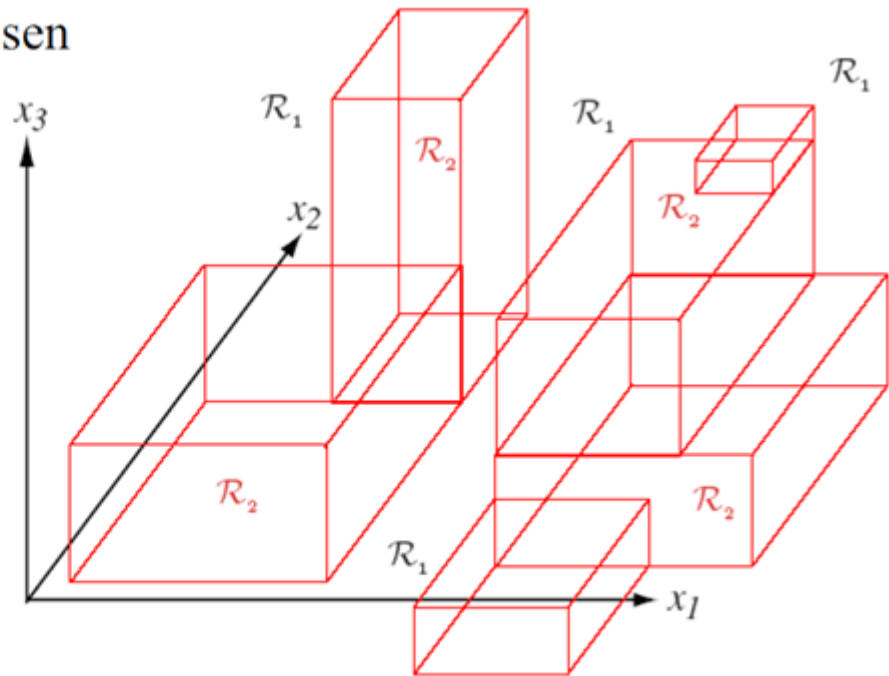


[Quelle: Duda et al., 2001]

2 Klassen



2 Merkmale

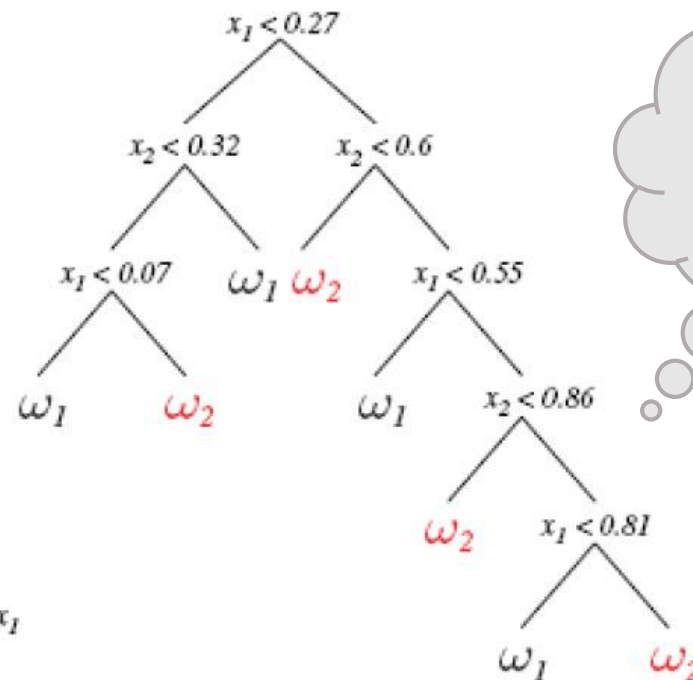
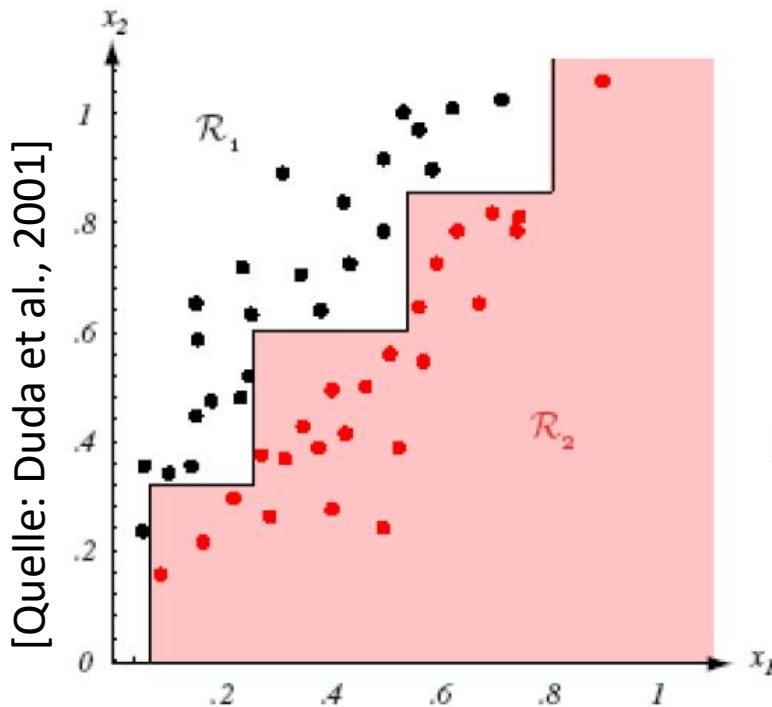


3 Merkmale

487

Merkmale und Kriterien

CART und andere Methoden für Entscheidungsbäume funktionieren am besten, wenn **sinnvolle Merkmale und Kriterien** gewählt werden

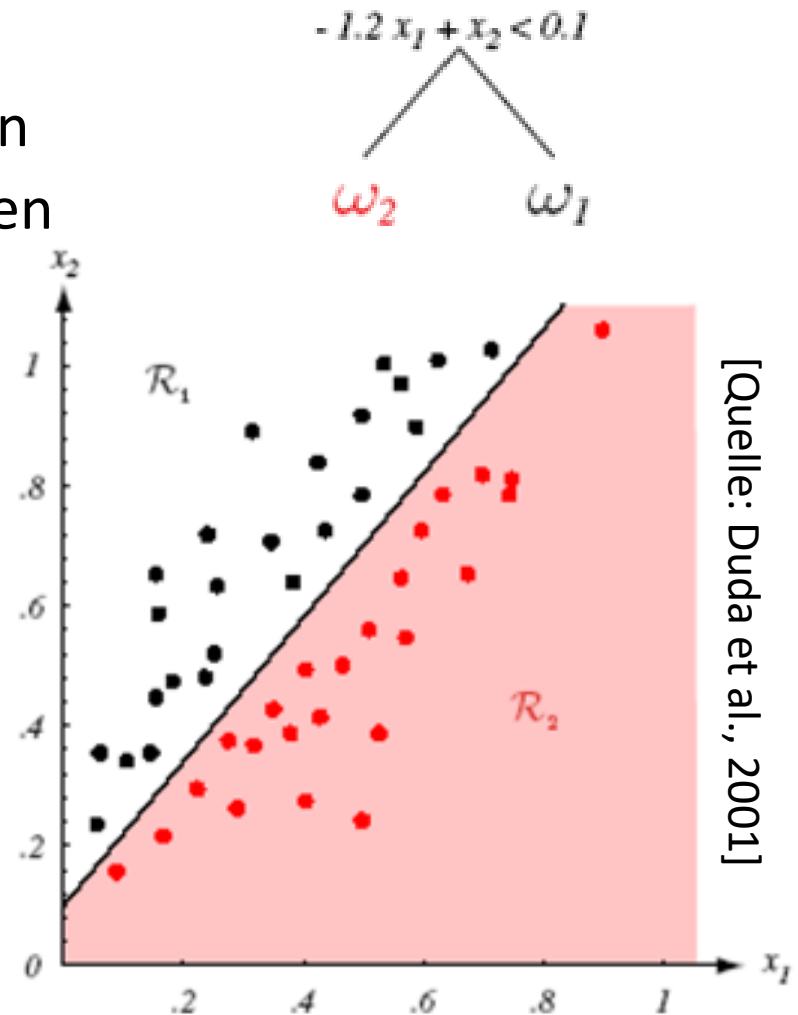


Schlecht
gewählte
Kriterien!

Merkmale und Kriterien

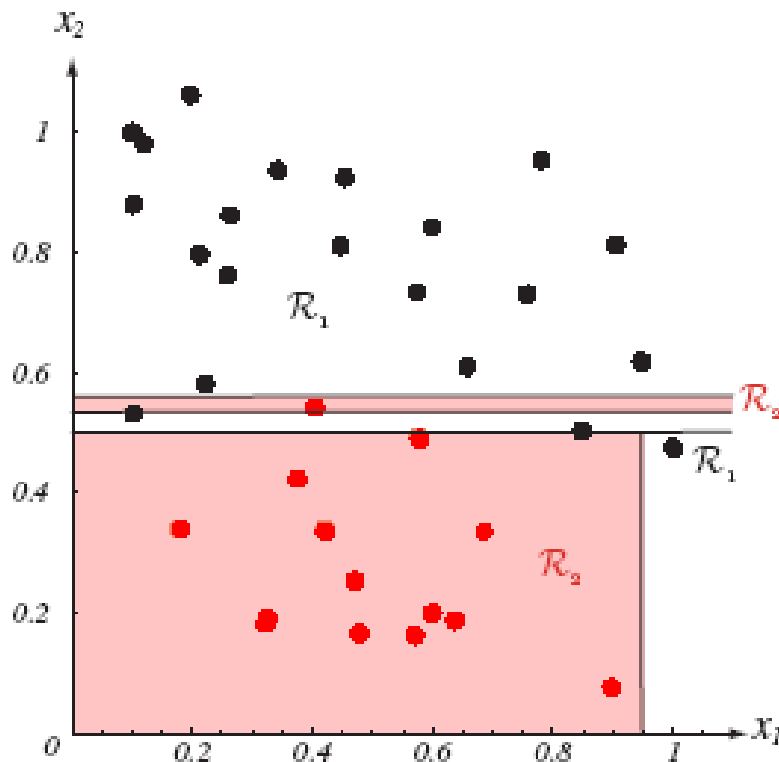
- Vorverarbeitung mit PCA
 - auffinden der wichtigsten Achsen
 - optimierte Kriterien in den Knoten

- Beispiel:
 - Entscheidungsgrenze:
linear Kombination
 - Entscheidungsbaum:
Wurzelknoten + zwei Blätter

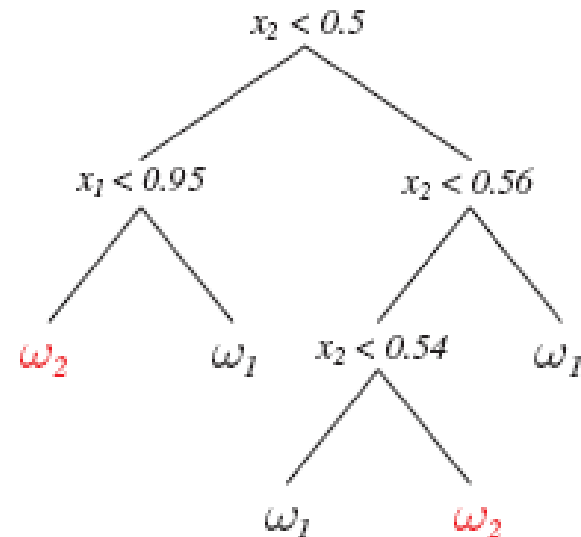


Multivariate Bäume

- optimale Entscheidungsgrenzen nicht parallel zu den Achsen
- CART und ähnliche Methoden sind nicht ausreichend

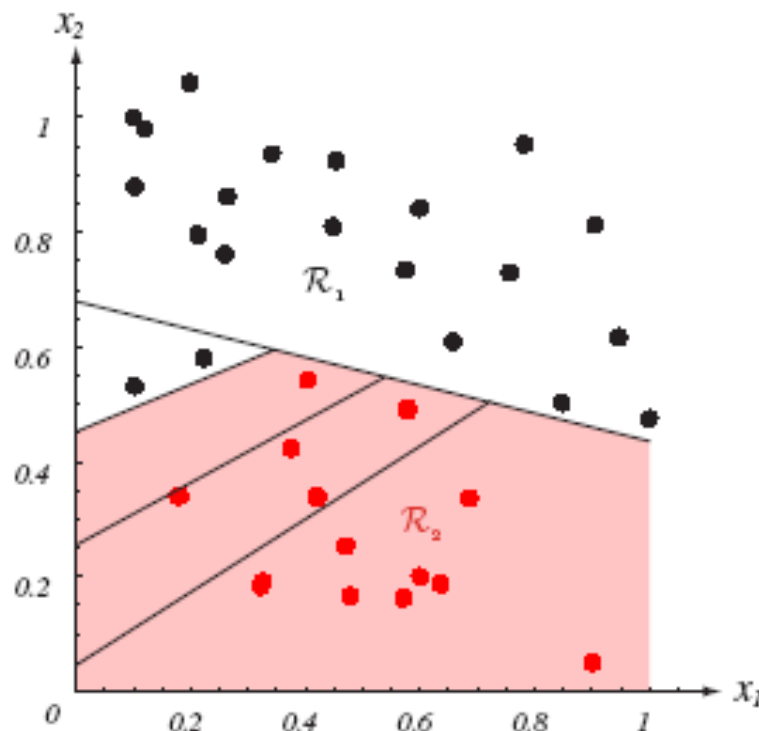


[Quelle: Duda et al., 2001]



Multivariate Bäume

- Lösung: Verwendung von Entscheidungsgrenzen die nicht parallel zu den Achsen liegen
- z.B. in jedem Knoten linearer Klassifikator, trainiert durch Perceptron o.ä.



[Quelle: Duda et al., 2001]

