Karlsruhe Institute of Technology

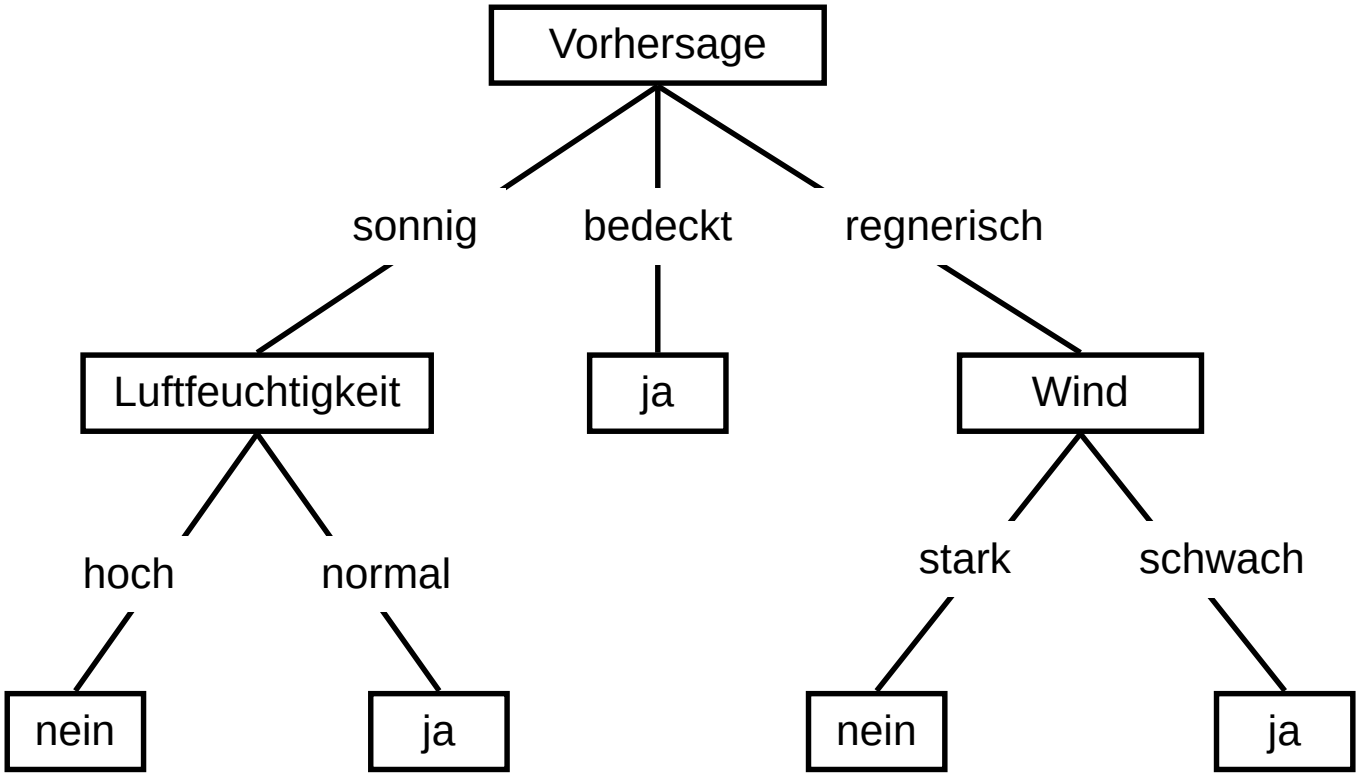
Entscheidungsbäume

Prof. Dr.-Ing. Rüdiger Dillmann
Prof. Dr.-Ing. J. Marius Zöllner

Übersicht

- Motivation
- ID3
- Overfitting
- Erweiterungen
- C4.5
- ID5R
- Random Forests
- Zusammenfassung

Entscheidungsbaum für Beispiel „Tennis spielen“



Entscheidungsbäume

Repräsentation:

- Jeder Knoten repräsentiert einen Attributtest
- Jeder Zweig entspricht einem bestimmten Attributwert
- Jedes Blatt entspricht einer Aussage i.A. Klassifikation

Allgemein – Beschreiben von:

- Disjunktion von Konjunktionen von Bedingungen an die Attributwerte einer Instanz:

$$(\text{Vorhersage} = \text{sonnig} \wedge \text{Luftfeuchtigkeit} = \text{normal}) \vee (\text{Vorhersage} = \text{bedeckt}) \vee (\text{Vorhersage} = \text{regnerisch} \wedge \text{Wind} = \text{schwach})$$

Für welche Problemstellungen eignen sich EB?

- Instanzen lassen sich als Attribut-Wert Paare beschreiben
- Zielfunktion besitzt diskrete Ausgabewerte
- Disjunkte Hypothesen erforderlich
- Beispieldaten sind möglicherweise verrauscht
- Beispieldaten enthalten evtl. fehlende Attributwerte

Lernen von EB: Verfahren

ID3 (Quinlan, 1986):

- nicht-inkrementelles Verfahren

C4.5 (Quinlan, 1993):

- Verbesserung von ID3 durch generalisierte Regeln (Pruning)
- kommerzielles System

ID5R (Utgoff, 1989):

- inkrementelles Verfahren

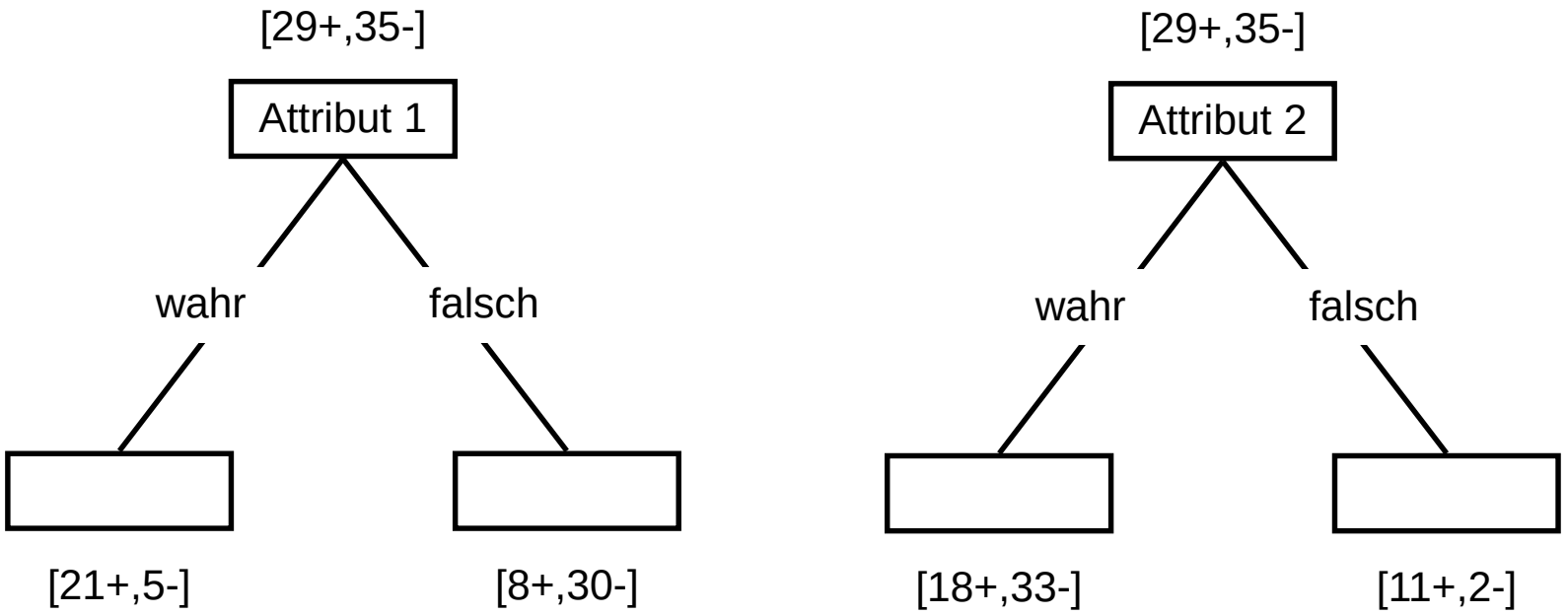
ID3: Top down Aufbau von EB

1. A ← Das beste Entscheidungsattribut für den nächsten Knoten.
2. Weise A als Entscheidungsattribut für den nächsten Knoten zu.
3. Füge für jeden möglichen Wert von A einen Nachfolgeknoten ein.
4. Verteile die Trainingsdaten gemäß ihrer Attributwerte auf die Nachfolgeknoten.
5. Terminiere wenn die Trainingsdaten perfekt abgebildet (klassifiziert) sind, sonst iteriere über die Nachfolgeknoten.

ID3: Top down Aufbau von EB

1. A ← Das beste Entscheidungsattribut für den nächsten Knoten.
2. Weise A als Entscheidungsattribut für den nächsten Knoten zu.
3. Füge für jeden möglichen Wert von A einen Nachfolgeknoten ein.
4. Verteile die Trainingsdaten gemäß ihrer Attributwerte auf die Nachfolgeknoten.
5. Terminiere wenn die Trainingsdaten perfekt abgebildet (klassifiziert) sind, sonst iteriere über die Nachfolgeknoten.

ID3: Auswahl des besten Testattributs?

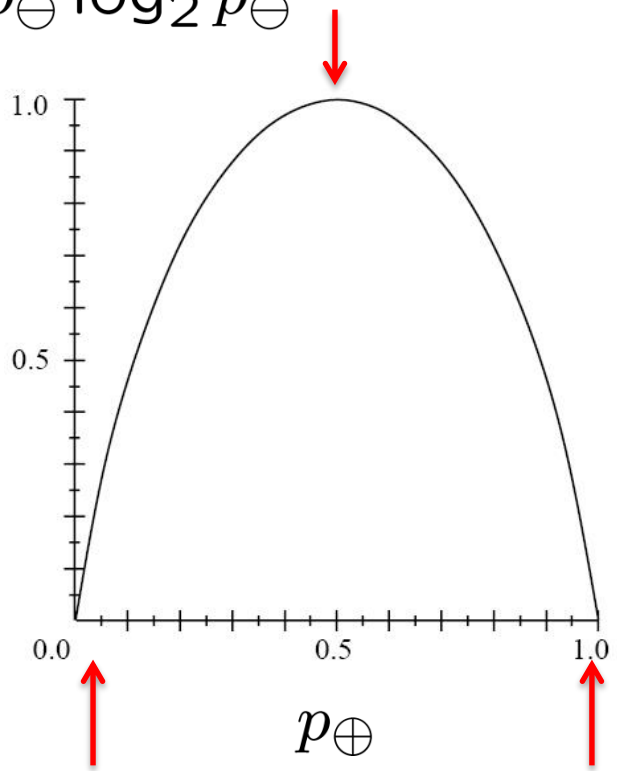


Schreibweise: [Anzahl Positive Bsp (+), Anzahl Negative Bsp. (-)]

Entropie I

Die Entropie als Maß der Homogenität der Trainingsdaten:

$$\text{Entropie}(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

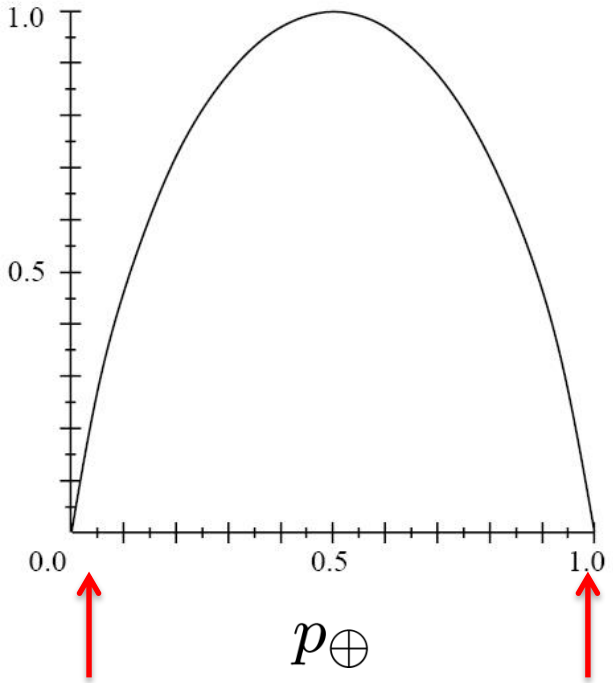


- S Menge der Trainingsbeispiele
- $p_{\oplus}$ Anteil der positiven Beispiele in S
- $p_{\ominus}$ Anteil der negativen Beispiele in S

Entropie II

Ziel ist es die Daten durch Festhalten eines Attributwertes v möglichst die Klasse $\oplus$ oder $\ominus$ einzuteilen, d.h. sukzessive die Entropie maximal zu reduzieren

$\text{Entropie}(S_v)$



S_v Menge der durch v eingeschränkter Trainingsbeispiele

Informationsgewinn

$\text{Gewinn}(S, A)$ = Erwartete Reduzierung der Entropie durch die Einsortierung über Attribut A

$$\text{Gewinn}(S, A) \equiv \text{Entropie}(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \text{Entropie}(S_v)$$

Wobei

$V(A)$ Menge aller möglichen Attributwerte von A

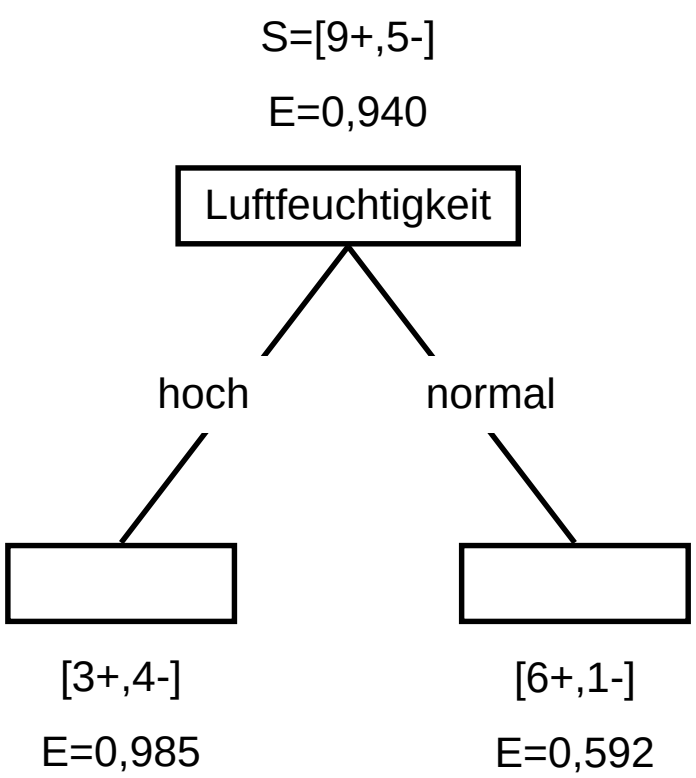
S_v Untermenge von S, für die A den Wert v annimmt

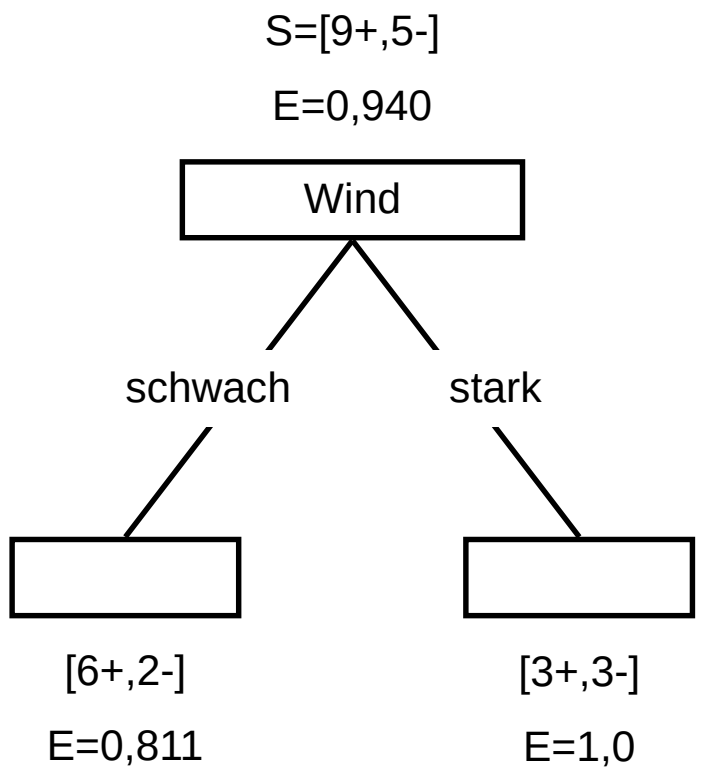
Ziel: Maximierung → Baum mit niedriger Tiefe

Beispiel I

Nr.	Vorhersage	Temperatur	Luftfeuchtigkeit	Wind	Tennis?
1	sonnig	heiß	hoch	schwach	nein
2	sonnig	heiß	hoch	stark	nein
3	bedeckt	heiß	hoch	schwach	ja
4	regnerisch	warm	hoch	schwach	ja
5	regnerisch	Kalt	normal	schwach	ja
6	regnerisch	Kalt	normal	stark	nein
7	bedeckt	Kalt	normal	stark	ja
8	sonnig	Warm	hoch	schwach	nein
9	sonnig	Kalt	normal	schwach	ja
10	regnerisch	Warm	normal	schwach	ja
11	sonnig	Warm	normal	stark	ja
12	bedeckt	Warm	hoch	stark	ja
13	bedeckt	Heiß	normal	schwach	ja
14	regnerisch	Warm	hoch	stark	nein

Beispiel II

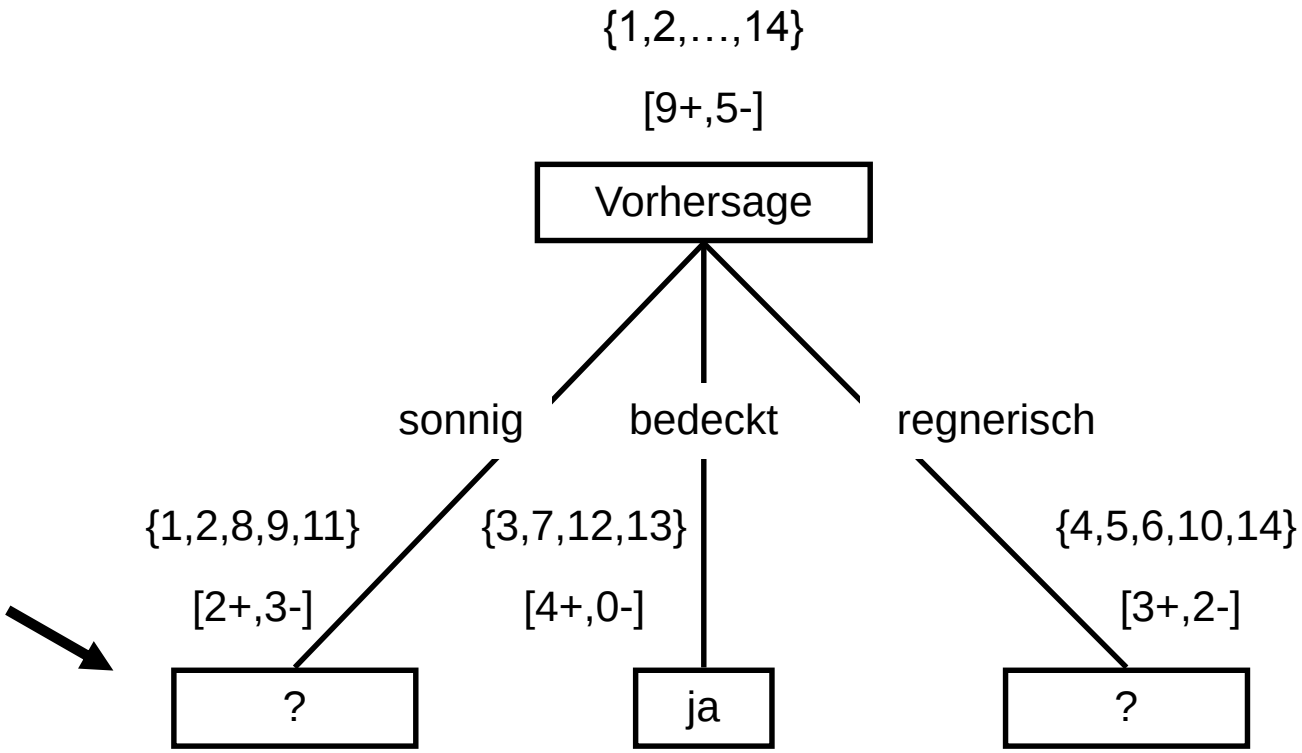




$$\text{Gewinn}(S, \text{Luftfeuchtigkeit}) = 0,940 - (7/14)0,985 - (7/14)0,592 = 0,151$$

$$\text{Gewinn}(S, \text{Wind}) = 0,940 - (8/14)0,811 - (6/14)1,0 = 0,048$$

Beispiel III

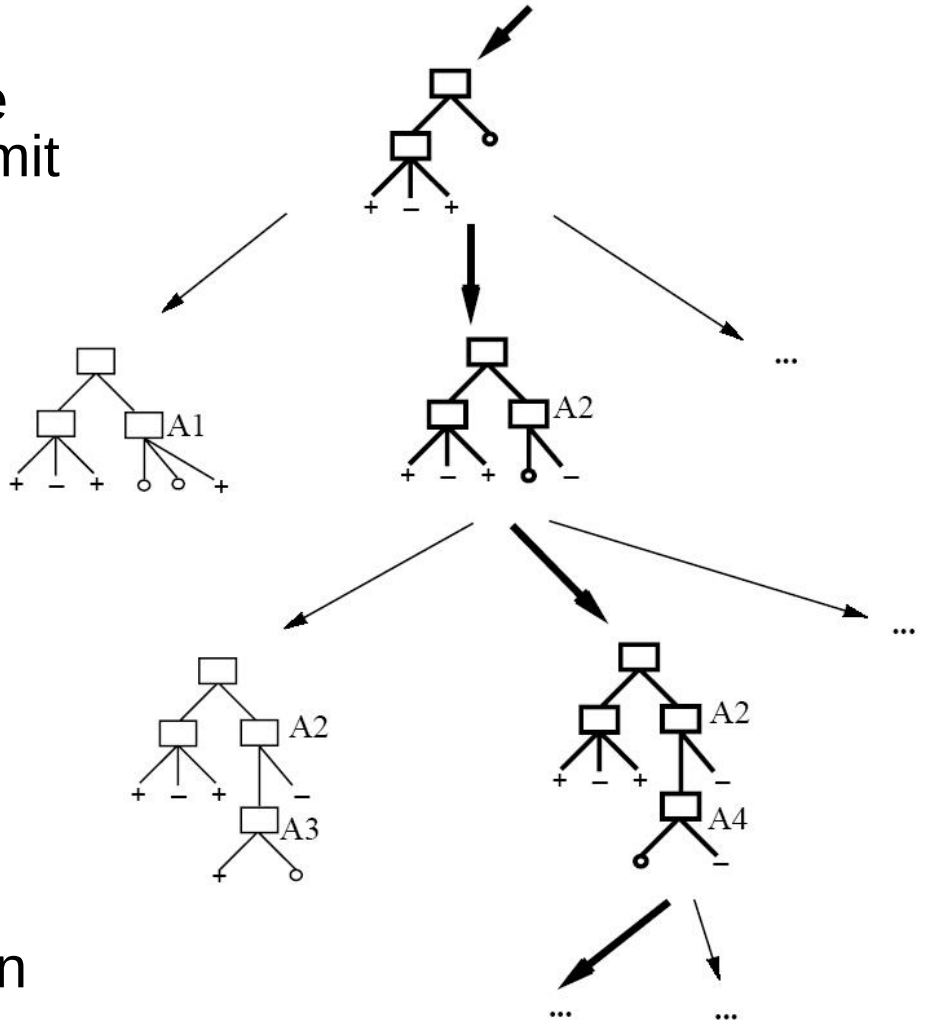


$$\text{Gewinn}(S_{\text{sonnig}}, \text{Luftfeucht.}) = 0,970 - (3/5)0,0 - (2/5)0,0 = 0,970$$
$$\text{Gewinn}(S_{\text{sonnig}}, \text{Temp.}) = 0,970 - (2/5)0,0 - (2/5)1,0 - (1/5)0,0 = 0,570$$
$$\text{Gewinn}(S_{\text{sonnig}}, \text{Wind}) = 0,970 - (2/5)1,0 - (3/5)0,918 = 0,19$$

alle in einer Klasse oder gleichverteilt

ID3: Suche im Hypothesenraum I

- Es gibt typischerweise viele Entscheidungsbäume, die mit den Trainingsbeispielen konsistent sind
- Hypothesenraum ist bei Bäumen vollständig, d.h. Zielfunktion ist enthalten
- Suche der Hypothese: „Simple-to-complex“ oder „hill climbing“ nach Informationsgewinn
- Lokale Minima (wie bei allen hill climbing Algorithmen) möglich



Allgemein: Präferenz- / Restriktions-Bias

Präferenzbias:

- Ordnung auf dem Raum der Hypothesen.
- Wähle Hypothese h mit der höchsten Präferenz.

Restriktionsbias:

- Einschränkung des Hypothesenraums, z.B. auf
  - lineare Schwellwertfunktionen
  - ....

ID3: Induktiver Bias

H ist bei ID3 die Potenzmenge der möglichen Instanzen X

- Kein Bias?

Nicht ganz:

- Präferenz für kleine Bäume und für Bäume, deren Attribute nahe der Wurzel einen hohen Informationsgewinn besitzen.
- ID3-Bias ist eine Präferenz für bestimmte Hypothesen, aber keine Restriktion des Hypothesenraums H.
- Occam‘s Razor: Bevorzuge die einfachste Hypothese, die mit den Trainingsdaten übereinstimmt.

Occam‘s Razor

Warum sollten kurze Hypothesen bevorzugt werden?

Argumente dafür:

- Es gibt weniger kurze als lange Hypothesen:
  Eine kurze Hypothese, welche die Daten korrekt beschreibt, ist wahrscheinlich kein Zufall.
  Eine lange Hypothese, welche die Daten korrekt beschreibt, könnte hingegen Zufall sein.

- Kurze Bäume sind effizienter bezüglich interner Repräsentation und Auswertung

ID3: Overfitting I

ID3 Basisverfahren:

- Jeder Zweig wächst solange, bis die Trainingsbeispiele perfekt klassifiziert werden.
- Dies basiert auf dem statistisch approximierten Informationsgewinn (Entropie, etc...)

➔

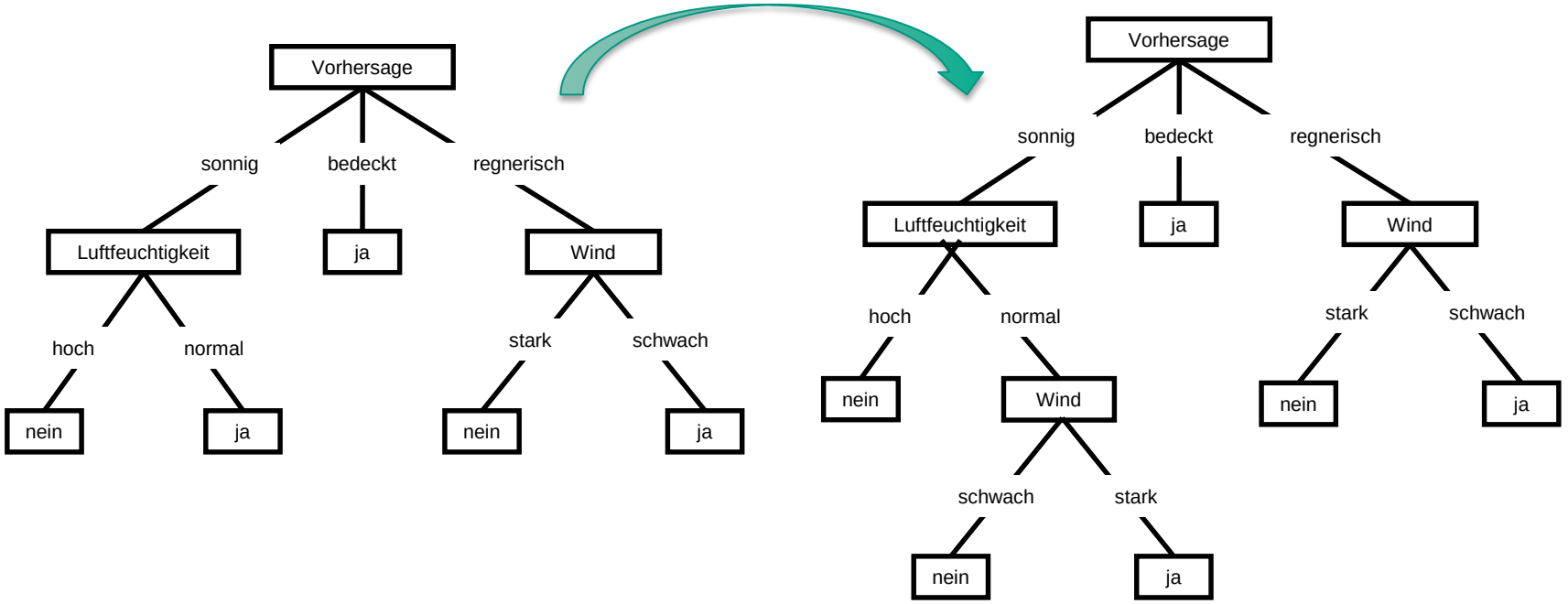
Dies kann zu Problemen führen, wenn

- die Daten verrauscht sind (z.B. Klasseninformation)
- die Beispiele nicht repräsentativ sind (z.B. zu wenige)

Beispiel

Was passiert bei Hinzufügen eines verrauschten Beispiels?

$< \text{sonnig}, \text{heiß}, \text{normal}, \text{stark} >$ Tennis = nein



Baum wird komplexer ➔ potentiell mehr Fehler

ID3: Overfitting II

Fehler der Hypothese h auf

- den Trainingsdaten: $\text{Fehler}_{\text{Training}}(h)$
- der gesamten Verteilung D der Daten: $\text{Fehler}_{\text{D}}(h)$

Definition:

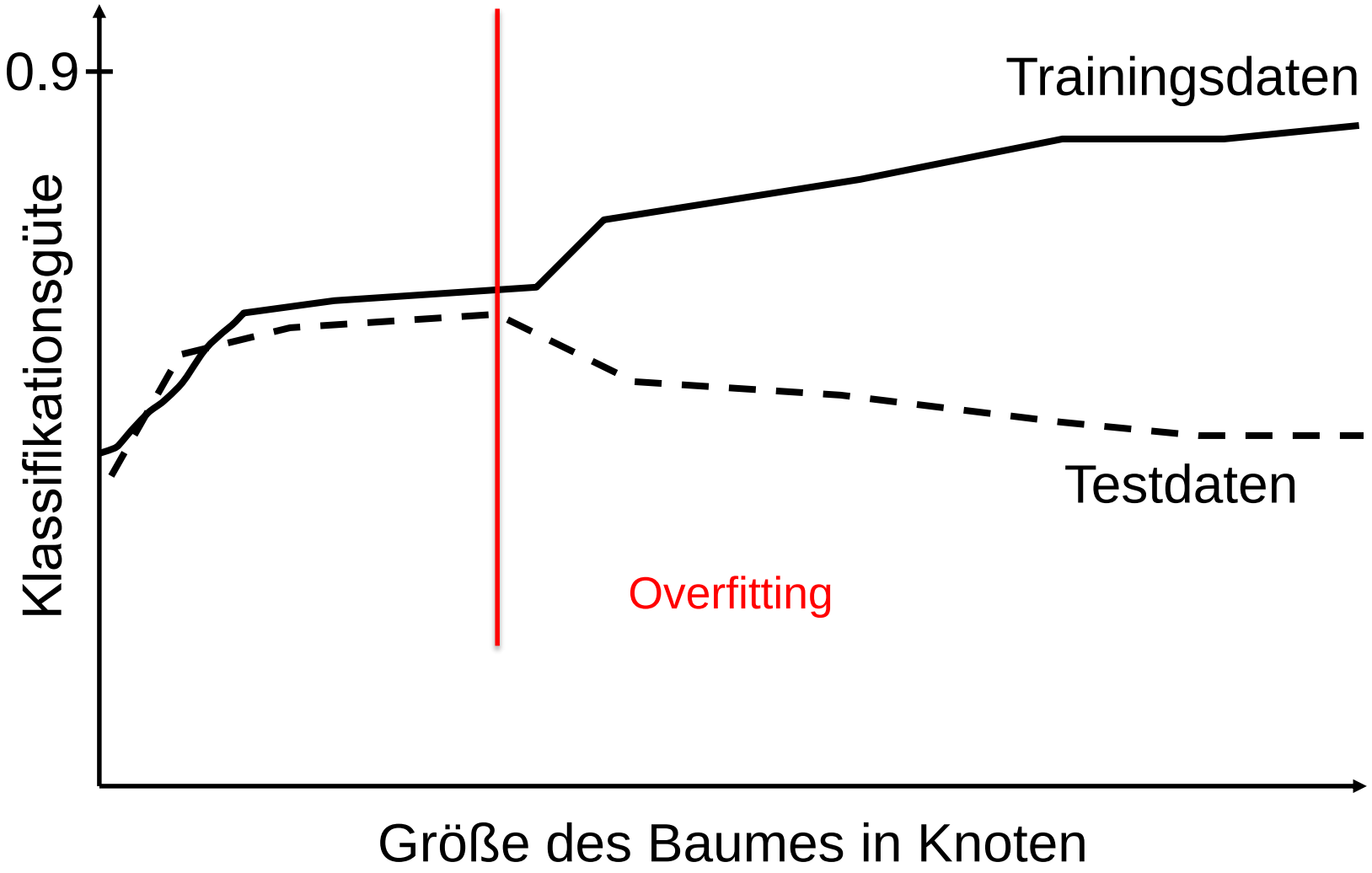
- Eine Hypothese h overfittet die Daten D, wenn es eine alternative Hypothese h‘ gibt, so dass

$$\text{Fehler}_{\text{Training}}(h) < \text{Fehler}_{\text{Training}}(h')$$

und

$$\text{Fehler}_{\text{D}}(h) > \text{Fehler}_{\text{D}}(h')$$

ID3: Overfitting III



Vermeidung von Overfitting

Lösungen:

- Frühzeitiges Stoppen des Baumwachstums
- Nachträgliches „Prunen“ des Baumes
  (in der Praxis erfolgreicher)

Kriterium für die Bestimmung der optimalen Baumgröße?

- Separate Testdatenmenge
- Statistischer Test auf den Trainingsdaten
  (z.B. χ^2 -Test für Unabhängigkeit von Attributen)
- Maß für die Kodierungskomplexität der Trainingsbeispiele und des Entscheidungsbaums
  (Prinzip der minimalen Beschreibungslänge)

Reduced Error Pruning I

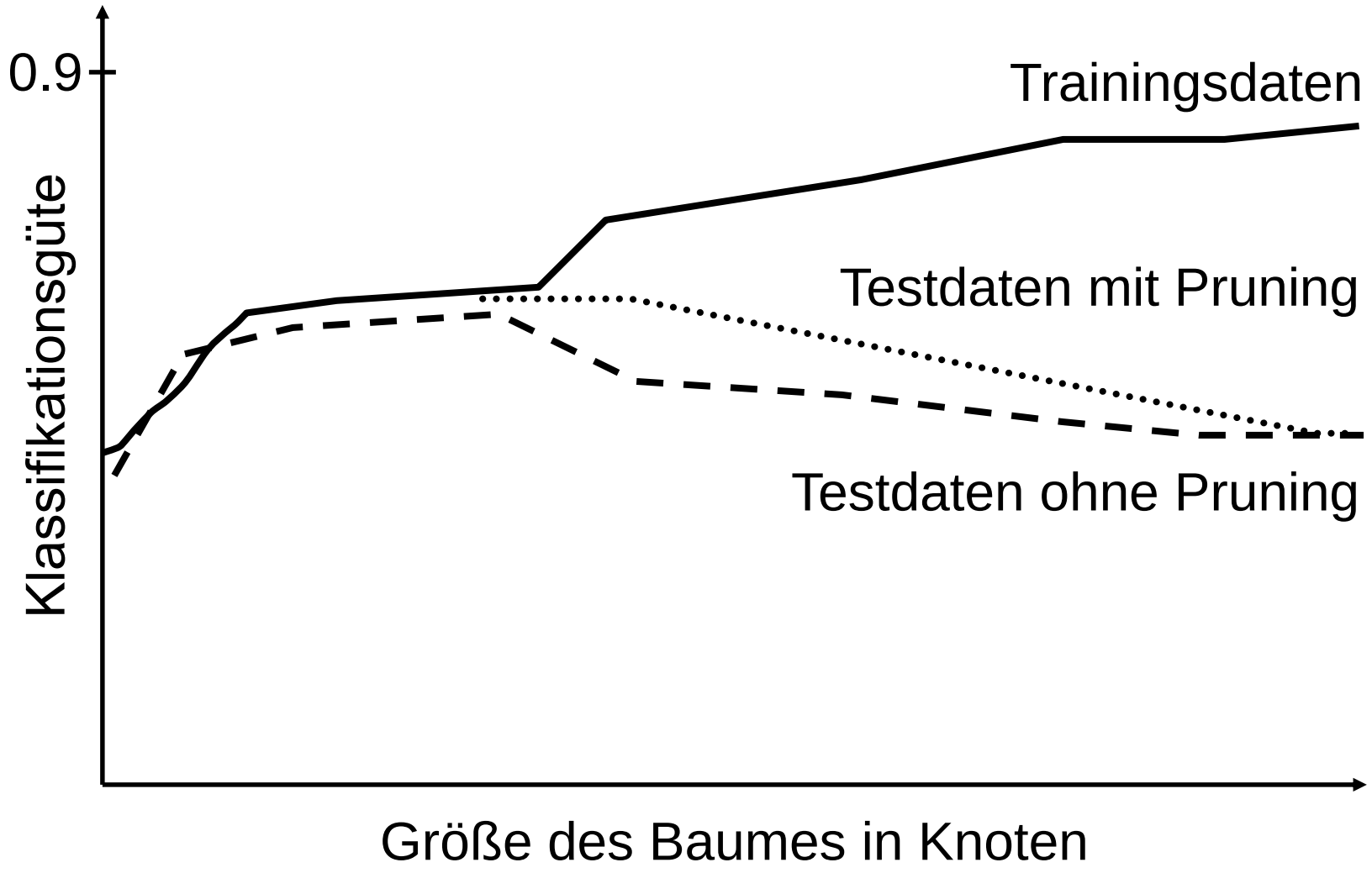
- Teile die Daten in Trainings- und Testdaten auf.
- Solange sich das Pruning nicht negativ auswirkt, verfahre wie folgt:
  - Evaluiere die Auswirkungen des Entfernens jedes Knotens (und seiner Nachfolgeknoten) auf die Klassifikationsgüte bzgl. der Testdaten.
  - Entferne den Knoten, dessen Entfernen die Klassifikationsrate bzgl. der Testdaten am meisten erhöht.

→ Liefert die kleinste Variante des akkuratesten Unterbaums.

Problem:

- Bei wenigen Daten erhöht das Aufteilen der Daten die Fehleranfälligkeit noch weiter.

Reduced Error Pruning II



Attribute mit vielen Werten

Problem:

- Attribute mit vielen Werten werden durch $\text{Gewinn}(S, A)$ gegenüber solchen mit wenigen Werten bevorzugt.
- Beispiel: Attribut Datum

Lösung: Bestrafen von Attributen

Verwende

$$\text{GewinnAnteil}(S, A) \equiv \frac{\text{Gewinn}(S, A)}{\text{SplittInformation}(S, A)}$$

$$\text{SplittInformation}(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

wobei S_i die Untermenge von S mit Wert v_i für A ist.

Kontinuierliche Attributwerte I

Gegeben:

- Attribut A mit kontinuierlichen Werten

Vorgehen:

- Dynamische Definition eines neuen diskret-wertigen Attributs A_c : A_c ist wahr, wenn $A > c$

Problem:

- Wahl des Schwellwertes c?

→ Auswahl über den Informationsgewinn:

- Sortierung der Beispiele gemäß ihrer Werte
- Optimaler Schwellwert liegt in der Mitte zwischen zwei benachbarten Beispielen mit unterschiedlichen Klassenzugehörigkeiten

Kontinuierliche Attributwerte II

Beispiel:

Temperatur	4°	9°	16°	22°	27°	32°
Tennis	nein	nein	ja	ja	ja	nein

Potentielle Schwellwerte:

- (9° + 16°) / 2 = 12,5° → G= 1 - 2/6*0,00 - 4/6*0,81 = 0,34
- (27° + 32°) / 2 = 29,5° → G= 1 - 5/6*0,97 - 1/6*0,00 = 0,08

Höchster Informationsgewinn im ersten Fall →

- c = 12,5°

Unbekannte Attributwerte

Problem:

- Fehlende Attributwerte → wie solche Daten verwenden?

Lösung:

- Sortiere alle Beispiele wie gewohnt in den EB ein wobei fehlende Attribute bekommen:
  - den häufigsten Attributwert der Beispiele
  - den häufigsten Attributwert der Beispiele der gleichen Klasse
  - jedem Wert v_i mit Wahrscheinlichkeit p_i → Verteile das Beispiel gemäß p_i anteilig auf die Nachfolger (Umsetzung komplexer)
- Verfahre bei der Klassifikation entsprechend

Attribute mit Kosten

Problem:

- Bestimmung der Attributwerte mit unterschiedlichen Kosten verbunden (z.B. in der medizinischen Diagnose).

Lösung:

- Finden eines korrekten Entscheidungsbaums mit niedrigen erwarteten Kosten durch Verwendung von

$$\frac{\text{Gewinn}^2(S, A)}{\text{Kosten}(A)}$$

oder

$$\frac{2^{\text{Gewinn}(S,A)} - 1}{(\text{Kosten}(A) + 1)^w}$$

wobei $w \in [0, 1]$ die Gewichtung (Bedeutung) der Kosten angibt.

ID3: Window

→ Lernmethode für großen Datenmengen

Vorgehen:

- Wähle zufällige Untermenge der Trainingsdaten aus („Window“)
- Bestimme EB mit diesen Beispielen
- Klassifiziere alle übrigen Beispiele mit gelerntem EB
- Falls nicht alle Daten korrekt klassifiziert, füge einen Teil der falsch klassifizierten zum Fenster hinzu (zufällig ausgewählt) und erstelle neuen EB

ID3: Zusammenfassung

- Top down Wachstum der EB
- Vollständiger Hypothesenraum
- Induktiver Bias: Präferenz für kleine EB und solche, die Attribute mit hohem Informationsgewinn nahe der Wurzel besitzen.
- Wichtiges Problem: Overfitting
  → Nachträgliches Prunen notwendig
- Erweiterungen:
  - Kontinuierliche / fehlende Attributwerte
  - Einbeziehung von Kosten für Attribute

Einordnung

Typ der Inferenz	*induktiv*	⟷	*deduktiv*
Ebenen des Lernens	*symbolisch*	⟷	*subsymbolisch*
Lernvorgang	*überwacht*	⟷	*unüberwacht*
Beispielgebung	*inkrementell*	⟷	*nicht inkrementell*
Umfang der Beispiele	*umfangreich*	⟷	*gering*
Hintergrundwissen	*empirisch*	⟷	*axiomatisch*

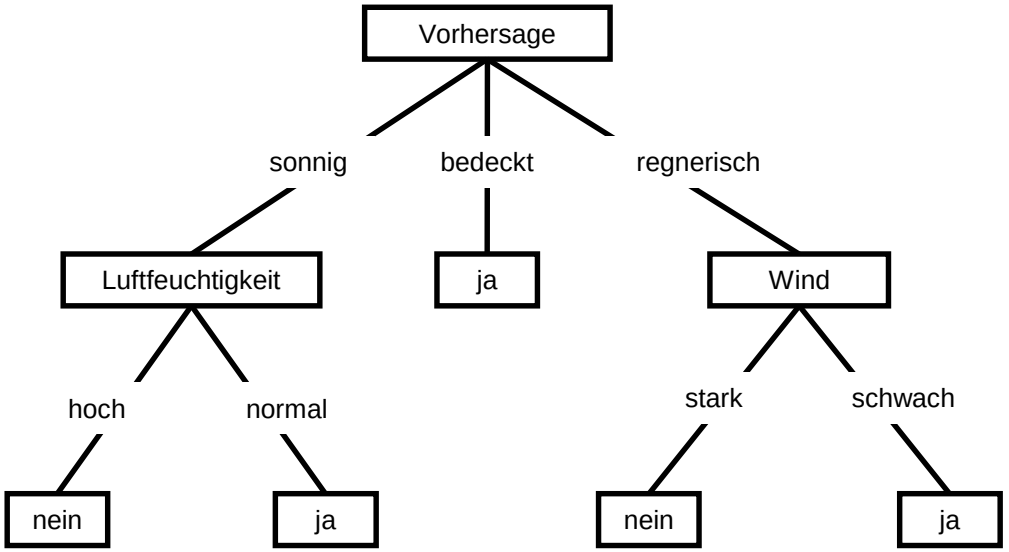
C4.5

- Quinlan, 1993
- Weiterentwicklung des ursprünglichen ID3-Algorithmus
- Unterstützt kontinuierliche Attribut-Werte
- Kann mit fehlenden Attributwerten umgehen
- Verwendet Rule Post-Pruning

C4.5: Rule Post-Pruning

1. Erstelle Baum wie gewohnt aus den Trainingsdaten (Overfitting erlaubt).
2. Konvertiere den Baum in äquivalente Menge von IF-THEN Regeln.
   IF-Teil enthält alle Attributtests eines Pfads, THEN die Ausgabe/Klasse
3. „Prune“ (Generalisiere) die Regeln so lange sich ihre Vorhersagegenauigkeit nicht verschlechtert (durch Entfernen von Vorbedingungen).
4. Sortiere alle Regeln nach ihrer Klassifikationsgüte und verwende sie in dieser Reihenfolge.

Beispiel: Regeln



IF (Vorhersage = sonnig) $\wedge$ (Luftfeuchtigkeit = hoch)
THEN (Tennis = nein)

IF (Vorhersage = sonnig) $\wedge$ (Luftfeuchtigkeit = normal)
THEN (Tennis = ja)

....

C4.5: Abschätzung der Güte der Regeln

„Pessimistische“ Abschätzung:

- Bestimmung der Regelgüte auf den Trainingsdaten
- Verwenden v. statistische Verfahren z.B. s.g. Urnenmodell
  → Binomialverteilung
  - Berechnung der Standardabweichung
  - Verwendung der unteren Grenze des gewählten Konfidenzintervalls als Regelgüte

→ Diese Heuristik ist statistisch nicht ganz einwandfrei, führt in der Praxis jedoch dennoch zu guten Ergebnissen

Umwandlung in Regeln

Vorteile:

- Unterscheidung zwischen verschiedenen Kontexten, in denen ein Entscheidungsknoten benutzt wird (eine Regel für jeden Pfad durch den Baum)
- Keine Unterscheidung zwischen Attributen näher an der Wurzel und solchen näher an den Blättern
  → Vereinfacht das Prunen
- Lesbarkeit für den Menschen

Einordnung

Typ der Inferenz	*induktiv*	⟷	*deduktiv*
Ebenen des Lernens	*symbolisch*	⟷	*subsymbolisch*
Lernvorgang	*überwacht*	⟷	*unüberwacht*
Beispielgebung	*inkrementell*	⟷	*nicht inkrementell*
Umfang der Beispiele	*umfangreich*	⟷	*gering*
Hintergrundwissen	*empirisch*	⟷	*axiomatisch*

ID5R

- Utgoff, 1989
- Inkrementelles Verfahren, d.h. sukzessives Einbringen von Beispielen in den Aufbau des EB
- Ergebnis ist äquivalent zu einem von ID3 erzeugten Baum

ID5R: Repräsentation der EB

Antwortknoten (Blätter):

- Klassenbezeichner
- Beschreibungen der Instanzen, die zu der Klasse gehören

Entscheidungsknoten:

- Attributtest mit Zweigen für jeden Attributwert
- Für jeden Attributwert Zähler für die positiven und negativen Beispiele
- Zusätzlich Zähler für die noch ausstehenden Attribut-Tests

→ Ermöglicht Berechnung des Informationsgewinns auf jeder Ebene ohne die bisherigen Beispiele erneut zu betrachten

ID5R: Beispiel

Gegeben:

- Attribute: Größe G, Haarfarbe H, Augenfarbe A
- Beispiele: $\ominus < G = \text{klein}, H = \text{blond}, A = \text{braun} >$
  $\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$
  $\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$

Michalski et al.: „Machine Learning - An Artificial Intelligence Approach“, Volume I-IV, Morgan Kaufmann, 1983-1994

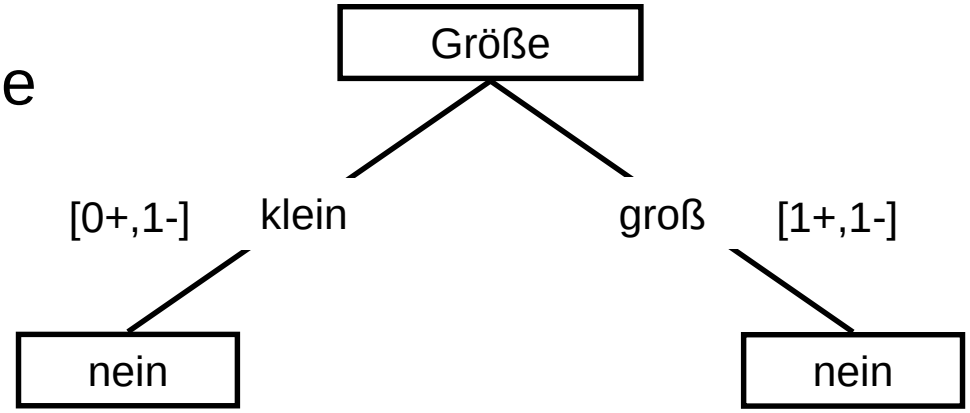
Paul E. Utgoff: „Incremental Induction of Decision Trees“, Vol. 4, S. 161-186, 1989

ID5R: Beispiel

Gegeben:

- Attribute: Größe G, Haarfarbe H, Augenfarbe A
- Beispiele: $\ominus < G = \text{klein}, H = \text{blond}, A = \text{braun} >$
  $\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$
  $\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$

- Teilbaum: siehe nächste Folien



$< H = \text{blond}, A = \text{braun} > < H = \text{dunkel}, A = \text{braun} >$

ID5R: Beispiel

Gegeben:

- Attribute: Größe G, Haarfarbe H, Augenfarbe A
- Beispiele:

$$\ominus < G = \text{klein}, H = \text{blond}, A = \text{braun} >$$
$$\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$$
$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$
$$\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{blau} >$$
$$\ominus < G = \text{klein}, H = \text{dunkel}, A = \text{blau} >$$
$$\oplus < G = \text{groß}, H = \text{rot}, A = \text{blau} >$$
$$\ominus < G = \text{groß}, H = \text{blond}, A = \text{braun} >$$
$$\oplus < G = \text{klein}, H = \text{blond}, A = \text{blau} >$$

ID5R: Baum Update Algorithmus I

Gegeben:

- Bestehender Entscheidungsbaum: EB
- Neue Instanz: I

Algorithmus:

1. Wenn EB leer, dann gibt einen Antwortknoten mit der Klasse von I zurück.
2. Wenn EB ein Antwortknoten mit der gleichen Klasse wie I, dann füge I zur Menge der Instanzen dieses Knotens hinzu.

ID5R: Beispiel I

Gegeben:

- Attribute: Größe G, Haarfarbe H, Augenfarbe A
- Beispiele: $\ominus < G = \text{klein}, H = \text{blond}, A = \text{braun} >$
  $\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$
  $\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$

Schritt 1+2: Einfügen eines Antwortknotens / Hinzufügen der Instanzen

nein

$< G = \text{klein}, H = \text{blond}, A = \text{braun} >$

$< G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$

ID5R: Baum Update Algorithmus II

3. Sonst:

a) Wenn EB Antwortknoten, dann Umwandlung in Entscheidungsknoten mit beliebigem Testattribut.

b) Aktualisiere die Zähler des Entscheidungsknotens (für Testattribut und alle anderen Attribute)

c) Ist das Testattribut nicht optimal, dann

   i. Restrukturiere Baum so, dass Attribut mit höchstem Informationsgewinn Testattribut wird (→ nächste Folie)

   ii. Wähle Testattribute mit höchstem Informationsgewinn rekursiv in den Unterbäumen (außer d))

d) Aktualisiere rekursiv den gemäß des Attributwerts von I (neue Instanz) gewählten Unterbaum.

ID5R: Restrukturierung

1. Wenn Attribut A_{neu} mit höchstem Informationsgewinn an der Wurzel, dann terminiere.
2. Sonst:
   a) Ziehe A_{neu} rekursiv an die Wurzel jedes direkten Unterbaums. Falls erforderlich wandle jeden Antwortknoten in Entscheidungsknoten mit Testattribut A_{neu} um.
   b) Transponiere den Baum so, dass A_{neu} an der Wurzel des neuen Baums und A_{alt} an der Wurzel jedes direkten Unterbaumes steht.

ID5R: Beispiel I

Gegeben:

- Attribute: Größe G, Haarfarbe H, Augenfarbe A
- Beispiele: $\ominus < G = \text{klein}, H = \text{blond}, A = \text{braun} >$
  $\ominus < G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$
  $\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$

Schritt 1+2: Einfügen eines Antwortknotens / Hinzufügen der Instanzen

nein

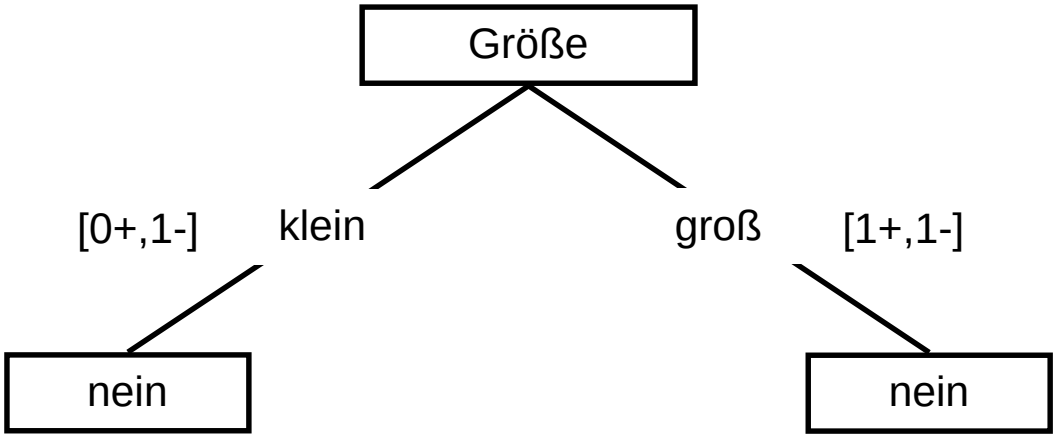
$< G = \text{klein}, H = \text{blond}, A = \text{braun} >$

$< G = \text{groß}, H = \text{dunkel}, A = \text{braun} >$

ID5R: Beispiel II

$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$

Schritt 3a,b): Umwandlung in Entscheidungsknoten mit beliebigem Testattribut (hier: Größe), Aktualisierung der Zähler

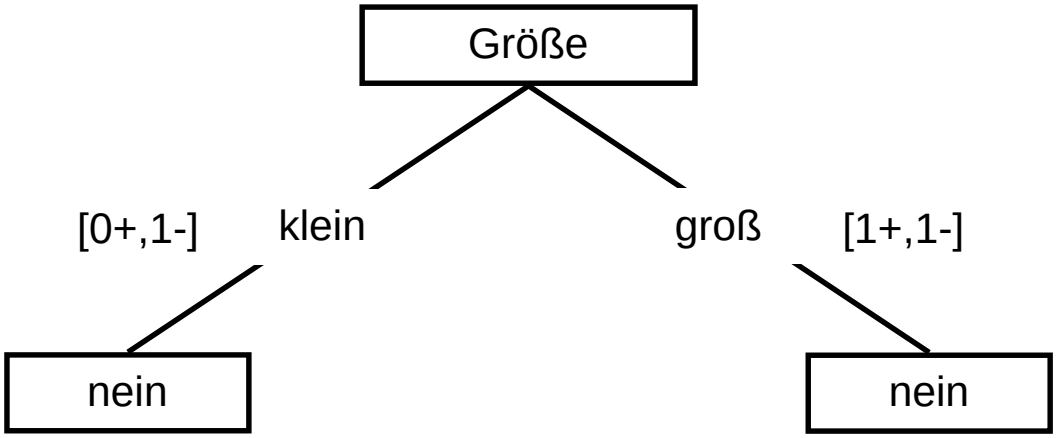


$$< H = \text{blond}, A = \text{braun} > \quad < H = \text{dunkel}, A = \text{braun} >$$

ID5R: Beispiel III

$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$

Schritt 3c: Attribut Augenfarbe hat größten Informationsgewinn (aus vorläufigen Zahlen im Teilbaum berechenbar)



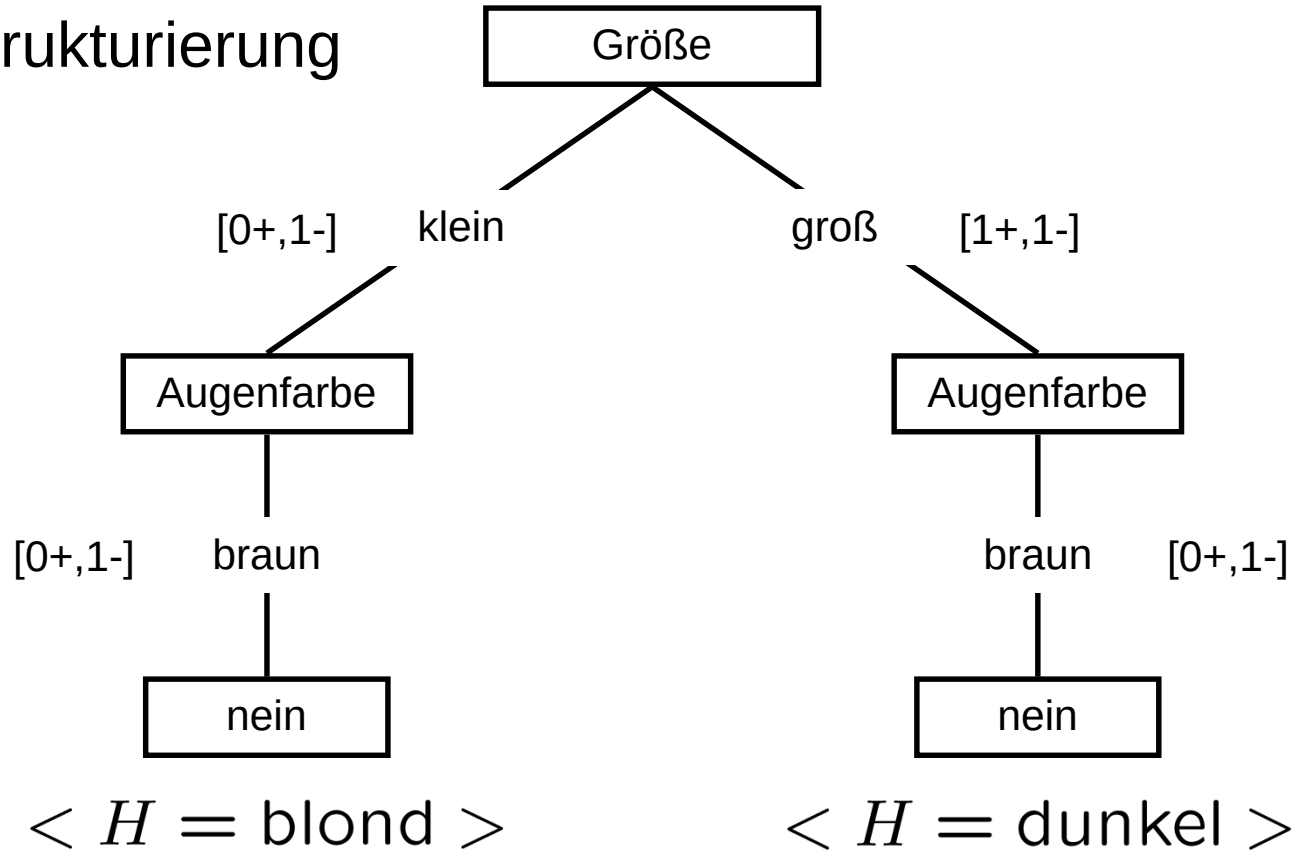
$$< H = \text{blond}, A = \text{braun} > \quad < H = \text{dunkel}, A = \text{braun} >$$

ID5R: Beispiel IV

$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$

Schritt 3c: Attribut Augenfarbe hat größten Informationsgewinn

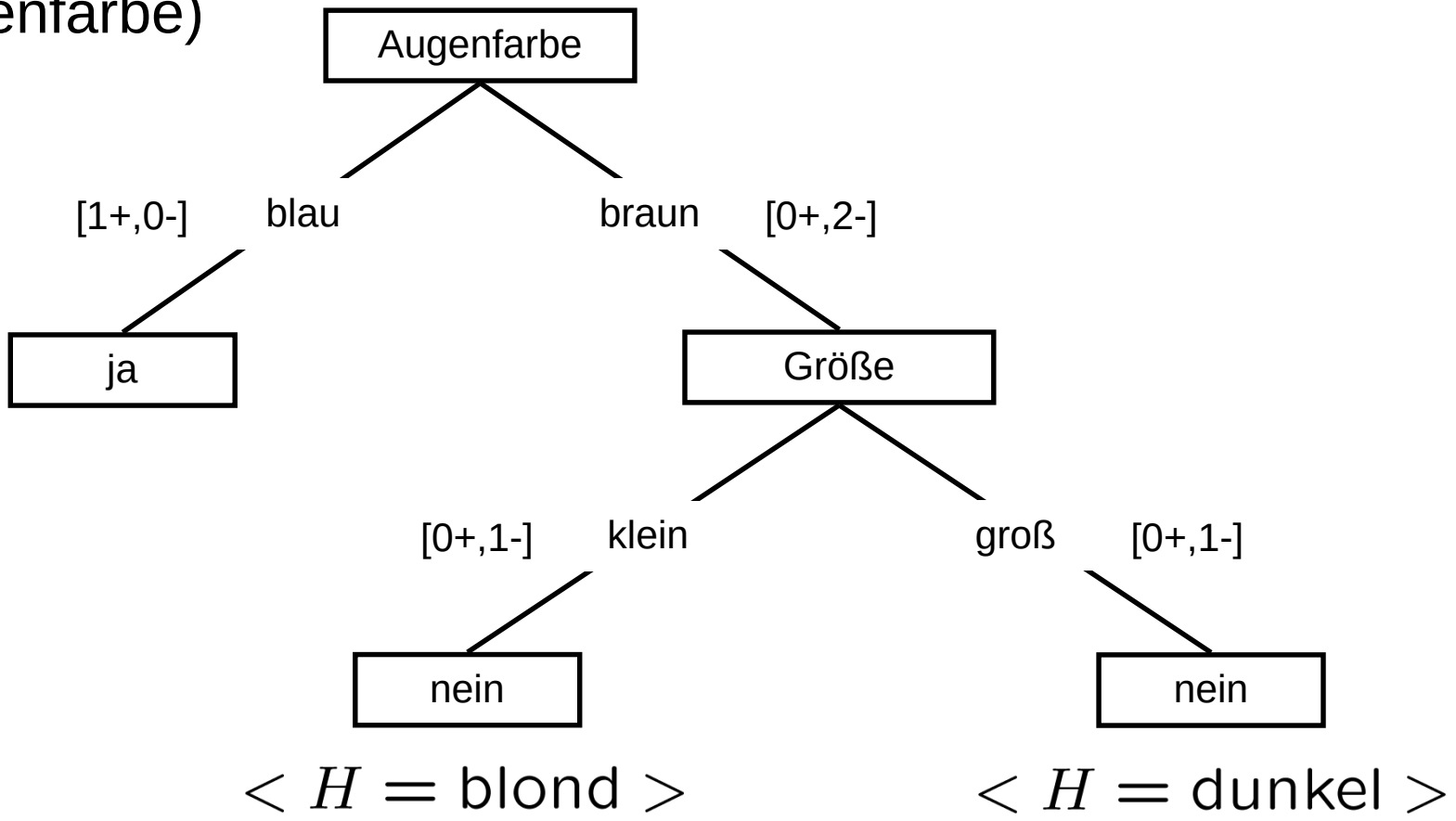
→ Restrukturierung



ID5R: Beispiel V

$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$

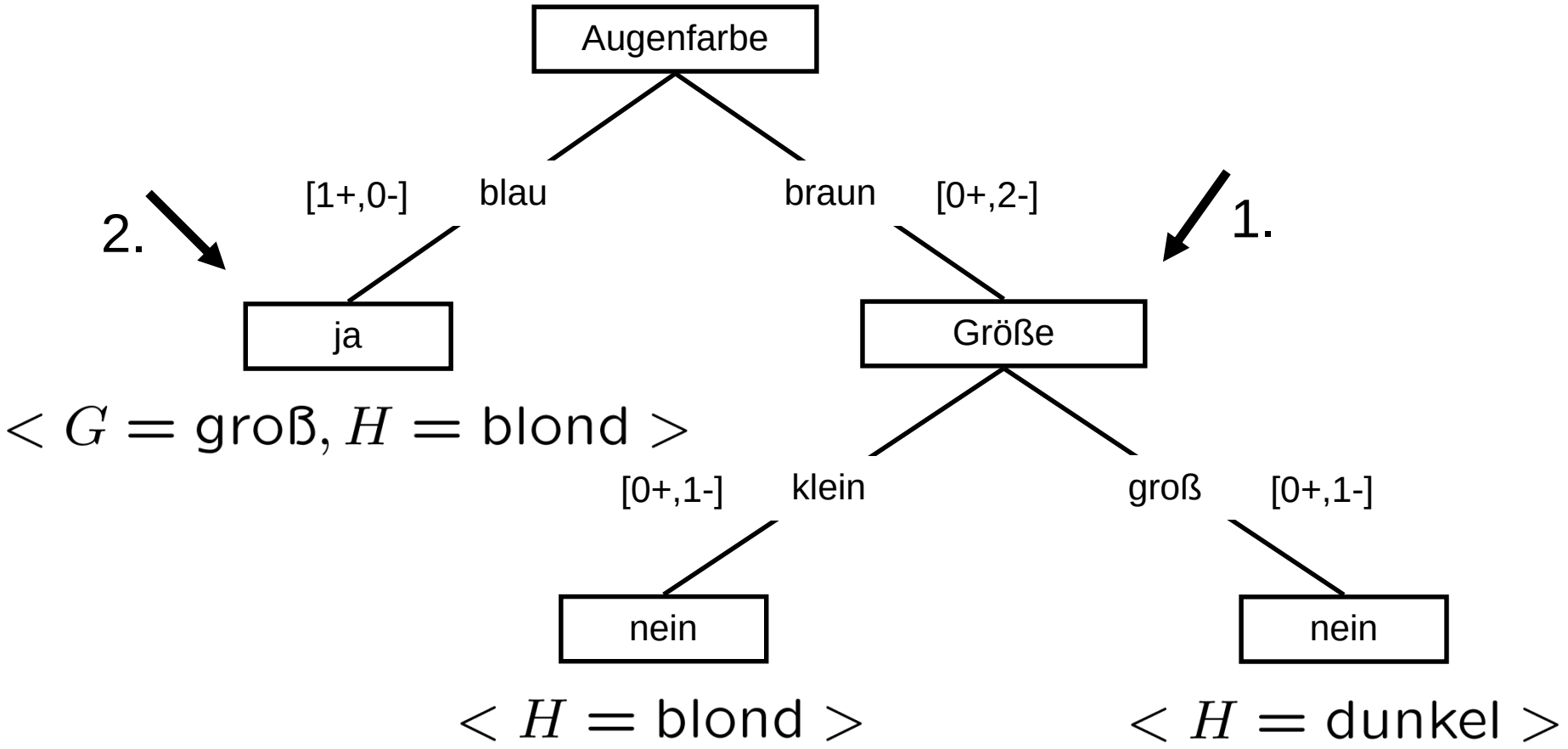
Schritt 3c: Transponieren des Baums (Hochziehen Augenfarbe)



ID5R: Beispiel VI

$$\oplus < G = \text{groß}, H = \text{blond}, A = \text{blau} >$$

Schritt 3c,d: Unterbäume rekursiv aktualisieren



Einordnung

Typ der Inferenz	*induktiv*	*deduktiv*
Ebenen des Lernens	*symbolisch*	*subsymbolisch*
Lernvorgang	*überwacht*	*unüberwacht*
Beispielgebung	*inkrementell*	*nicht inkrementell*
Umfang der Beispiele	*umfangreich*	*gering*
Hintergrundwissen	*empirisch*	*axiomatisch*

Random Forests

- Mehrere Entscheidungsbäume (=Wald/Forest) erstellen.
  - Einfach
  - Unkorreliert
  - Zufällige Wahl von Attributen (bzw. Trainingsdaten)
- Für eine Klassifikation darf jeder Baum in diesem Wald eine Entscheidung treffen und die Klasse mit den meisten Stimmen entscheidet die endgültige Klassifikation.
- Eigenschaften:
  - Schnelles Training
    - Da einzelne Entscheidungsbäume kleiner.
    - Trainingszeit steigt linear mit der Anzahl der Bäume.
  - Parallelisierbar (sowohl im Training als auch bei der Evaluation)
  - Effizient für große Datenmengen

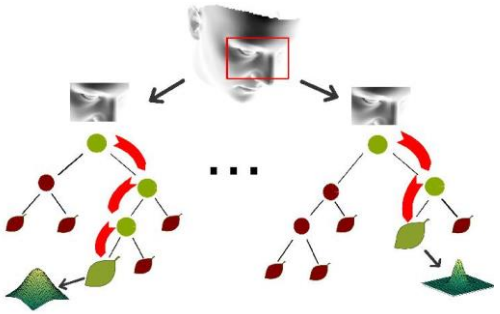
Random Forests II

- Vergleich zu Standard Entscheidungsbäumen
  - Kein Abschneiden der Bäume (kein Pruning) -> Overfitting erlaubt.
  - Attributwahl auf zufälliger Untermenge aller Attribute (Randomisierung).
- Eigenschaften der erstellten Bäume:
  - Jeder Baum sollte für sich ein guter Klassifikator sein.
  - Die Bäume sollten untereinander möglichst unkorreliert sein.
- Randomisierungsmöglichkeiten:
  - Bootstraping: Aus N Trainingsdaten werden N Trainingsbeispiele mit zurücklegen gezogen. Baum hat so ca ~63% der Größe verglichen mit allen Trainingsdaten.
  - Auswahl des Attributtests aus einer Teilmenge der vorhandenen Attributtests.
  - The main secret is to inject the „right kind of randomness“

Random Decision Trees

Real time head pose estimation from low-quality depth data

- Kopfstellung wird im Entscheidungsbaum eingelernt.
- Jeder Baum ‘votet’ für eine Kopfstellung, aus allen ‘votes’ wird die Stellung des Kopfes bestimmt.
- ca. 6 GB an anotierten Messdaten



http://www.vision.ee.ethz.ch/~gfanelli/pubs/cvpr11.pdf

Zusammenfassung

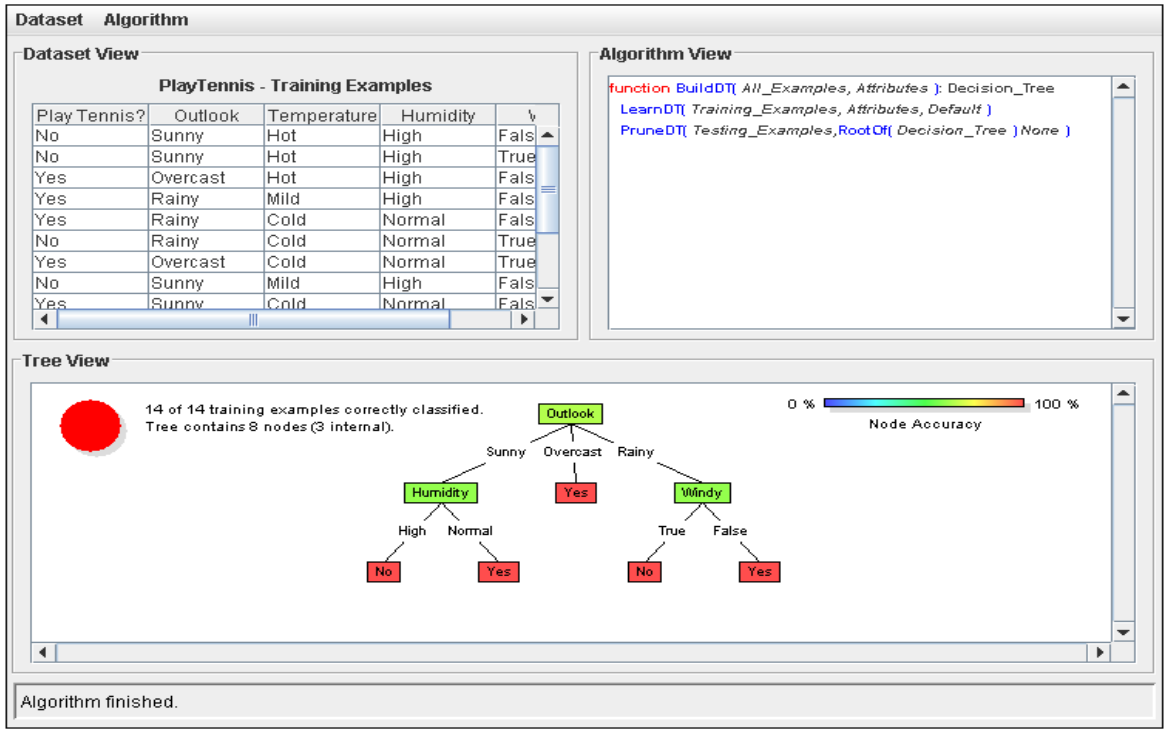
- Lernen von EB:
  - Praktische Methode für induktive Inferenz
- ID3
- Overfitting
- Erweiterungen
- C4.5:
  - Rule Post-Pruning
- ID5R:
  - Inkrementelle Beispielgebung
  - Ergebnis äquivalent zu ID3
  - Komplexere Repräsentation notwendig
- Random Forests
  - Mehrere (zufällige) Bäume. Ergebnis setzt sich aus den ‚votes' der Einzelbäume zusammen

Einordnung

Typ der Inferenz	*induktiv*	⟷	*deduktiv*
Ebenen des Lernens	*symbolisch*	⟷	*subsymbolisch*
Lernvorgang	*überwacht*	⟷	*unüberwacht*
Beispielgebung	*inkrementell*	⟷	*nicht inkrementell*
Umfang der Beispiele	*umfangreich*	⟷	*gering*
Hintergrundwissen	*empirisch*	⟷	*axiomatisch*

Demo-Applet

AIxploratorium (University of Alberta, Canada)



http://webdocs.cs.ualberta.ca/~aixplore/learning/DecisionTrees/index.html

Literatur I

- [1] *Tom Mitchell*: ***Machine Learning, Kapitel 3***. McGraw-Hill, New York, 1997.
- [2] *Michalski et al.*: **Machine Learning - An Artificial Intelligence Approach.** Volume I-IV, Morgan Kaufmann, 1983-1994.
- [3] *J.R. Quinlan*: **Induction of Decision Trees.** Vol. 1, S. 81-106, 1986.
- [4] *Paul E. Utgoff*: **Incremental Induction of Decision Trees.** Volume 4, S. 161-186, 1989.

Literatur II

- [5] *J. Ross Quinlan*: **C4.5: Programs for Machine Learning.** Morgan Kaufmann, 1993.
  - Detaillierte Beschreibung und Programm für C4.5
- [6] Homepage von *Tom Mitchell:* http://www-2.cs.cmu.edu/~tom/
  - Programm und Daten für ID3
- [7] Data Mining Tutorials von *Andrew W. Moore*: http://www-2.cs.cmu.edu/~awm/tutorials/

Literatur II

- [8] http://de.wikipedia.org/wiki/Random_Forest
- [9] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- [10] http://www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html