# Syntaktische und Statistische Mustererkennung

VO 1.0 840.040
(UE 1.0 840.041)

Bernhard Jung

bernhard@jung.name
http://bernhard.jung.name/VUSSME/

# Rückblick

- Distanzmaße, Metriken, Pattern Matching
- Entscheidungstabellen, Entscheidungsbäume
- Entropie, Information Gain

# Durchschnittliche bedingte spezifische Entropie H(Y|X)

| X | Y |
|---|---|
| Math | Yes |
| History | No |
| CS | Yes |
| Math | No |
| Math | No |
| CS | Yes |
| History | No |
| Math | Yes |

| $v_j$ | Prob(X=$v_j$) | H(Y\|X=$v_j$) |
|---|---|---|
| Math | 0,50 | 1,00 |
| History | 0,25 | 0,00 |
| CS | 0,25 | 0,00 |

$$H(Y|X) = 0.5 * 1.0 + 0.25 * 0 + 0.25 * 0 = 0.5$$

# Informationsgewinn

IG(Y|X) = Ich muss Y übertragen. Wieviele bits erspare ich mir, wenn beide Seiten X kennen?

IG(Y|X) = H(Y) - H(Y | X)

wealth values: poor rich

gender Female 14423 1769  H( wealth | gender = Female ) = 0.497654

Male 22732 9918  H( wealth | gender = Male ) = 0.885847

H(wealth) = 0.793844 H(wealth|gender) = 0.757154

IG(wealth|gender) = 0.0366896

# Relativer Informationsgewinn

RIG(Y|X) = Ich muss Y übertragen. Welchen Anteil an bits erspare ich mir, wenn beide Seiten X kennen?

RIG(Y|X) = IG(Y|X) / H(Y) = (H(Y) - H(Y | X)) / H(Y)

# Wozu Informationsgewinn?

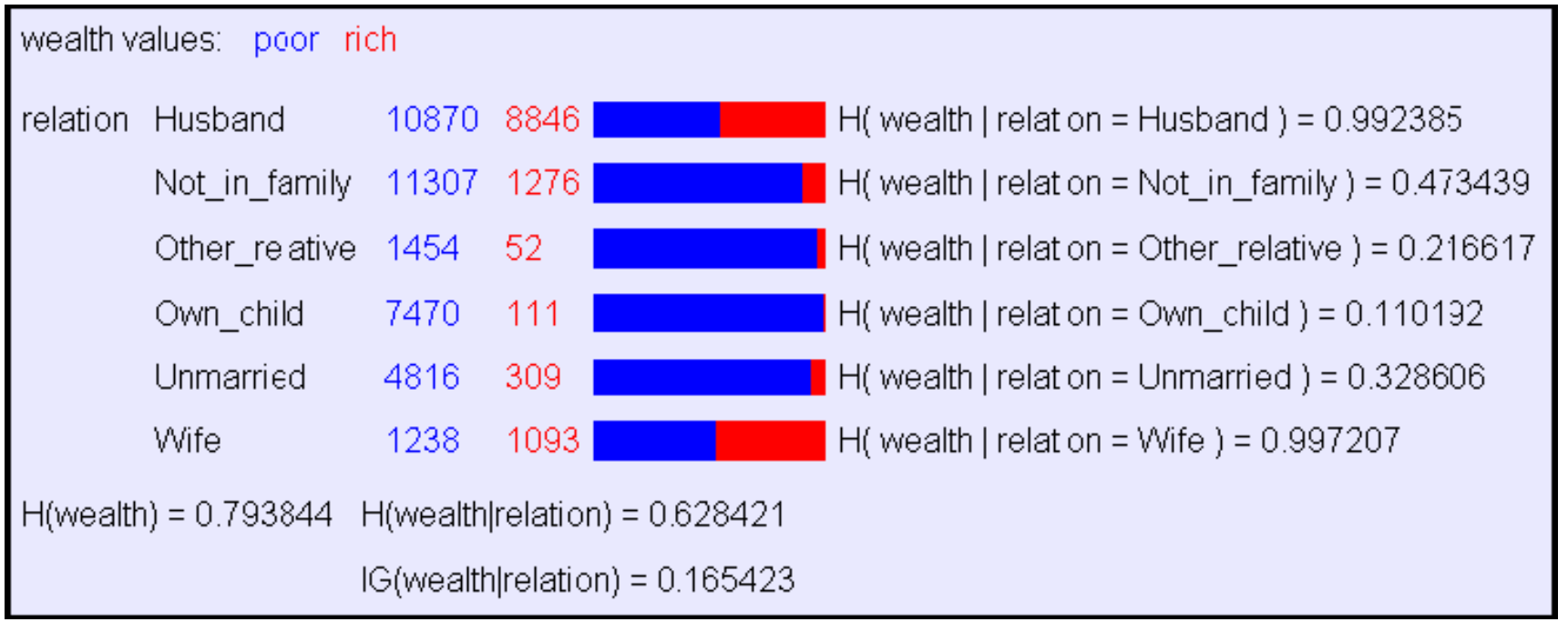
Angenommen, man möchte vorhersagen, ob jemand älter als 80 Jahre wird ...

- IG(LongLife | HairColor) = 0.01
- IG(LongLife | Smoker) = 0.2
- IG(LongLife | Gender) = 0.25
- IG(LongLife | LastDigitOfSSN) = 0.00001

Der Informationsgewinn gibt an, wie „interessant“ eine Kontingenztabelle ist.

# Suche nach hohem Informationsgewinn

Gegeben dass eine bestimmte Variable (Y = z.b. wealth) vorhergesagt werden soll, ist es einfach jenes Attribut zu finden, das den höchsten Informationsgewinn ermöglicht.



# Information Gain

**Entropie:**

$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m = -\sum_{j=1}^{m} p_j \log_2 p_j$$

**Bedingte spezifische Entropie:**

$$H(Y|X) = \sum_j P(X = v_j) H(Y|X = v_j)$$

**Information Gain:**

IG(Y|X) = H(Y) - H(Y | X)

# Entscheidungsbaumlernen

Ein Entscheidungsbaum ist ein in baumform **strukturierter Plan von Tests** von Attributen, um eine **Zielvariable** vorherzusagen.

Algorithmus:

1. Wähle jenes Attribut als Test, welches den höchsten Informationsgewinn ermöglicht.
2. Wiederhole diesen Schritt rekursiv ...

# Miles per Gallon Dataset

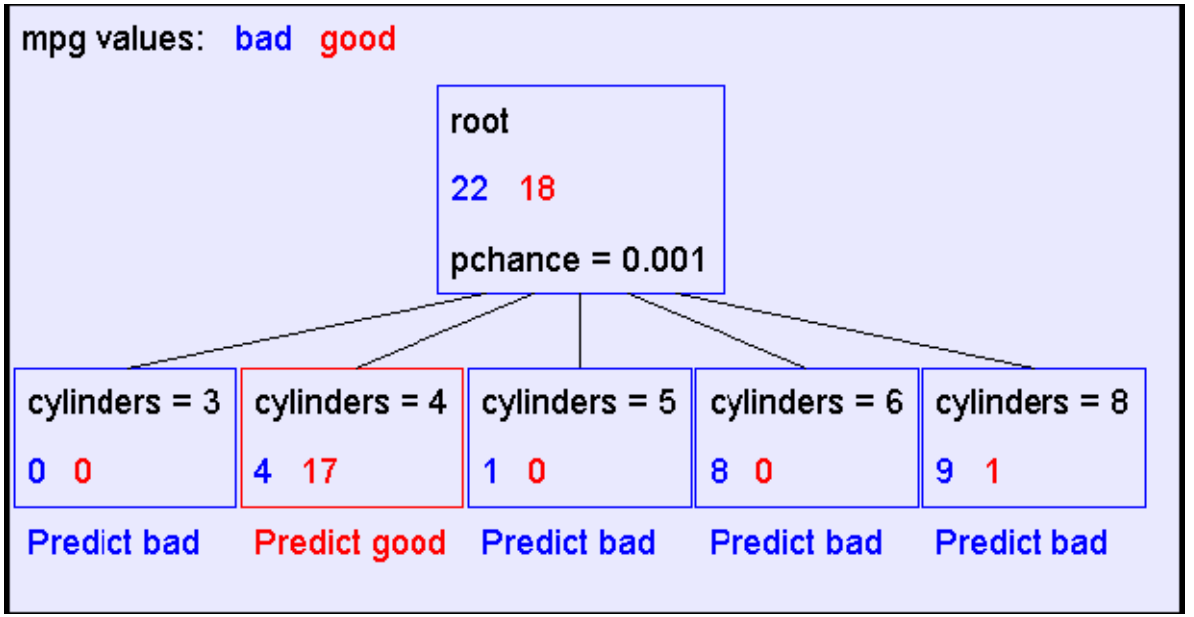
| mpg | cylinders | displacement | horsepower | weight | acceleration | modelyear | maker |
|---|---|---|---|---|---|---|---|
| good | 4 | low | low | low | high | 75to78 | asia |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | medium | medium | medium | low | 75to78 | europe |
| bad | 8 | high | high | high | low | 70to74 | america |
| bad | 6 | medium | medium | medium | medium | 70to74 | america |
| bad | 4 | low | medium | low | medium | 70to74 | asia |
| bad | 4 | low | medium | low | low | 70to74 | asia |
| bad | 8 | high | high | high | low | 75to78 | america |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| : | : | : | : | : | : | : | : |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 8 | high | medium | high | high | 79to83 | america |
| bad | 8 | high | high | high | low | 75to78 | america |
| good | 4 | low | low | low | low | 79to83 | america |
| bad | 6 | medium | medium | medium | high | 75to78 | america |
| good | 4 | medium | low | low | low | 79to83 | america |
| good | 4 | low | low | medium | high | 79to83 | america |
| bad | 8 | high | high | high | low | 70to74 | america |
| good | 4 | low | medium | low | medium | 75to78 | europe |
| bad | 5 | medium | medium | medium | medium | 75to78 | europe |

Information gains using the training set (40 records)

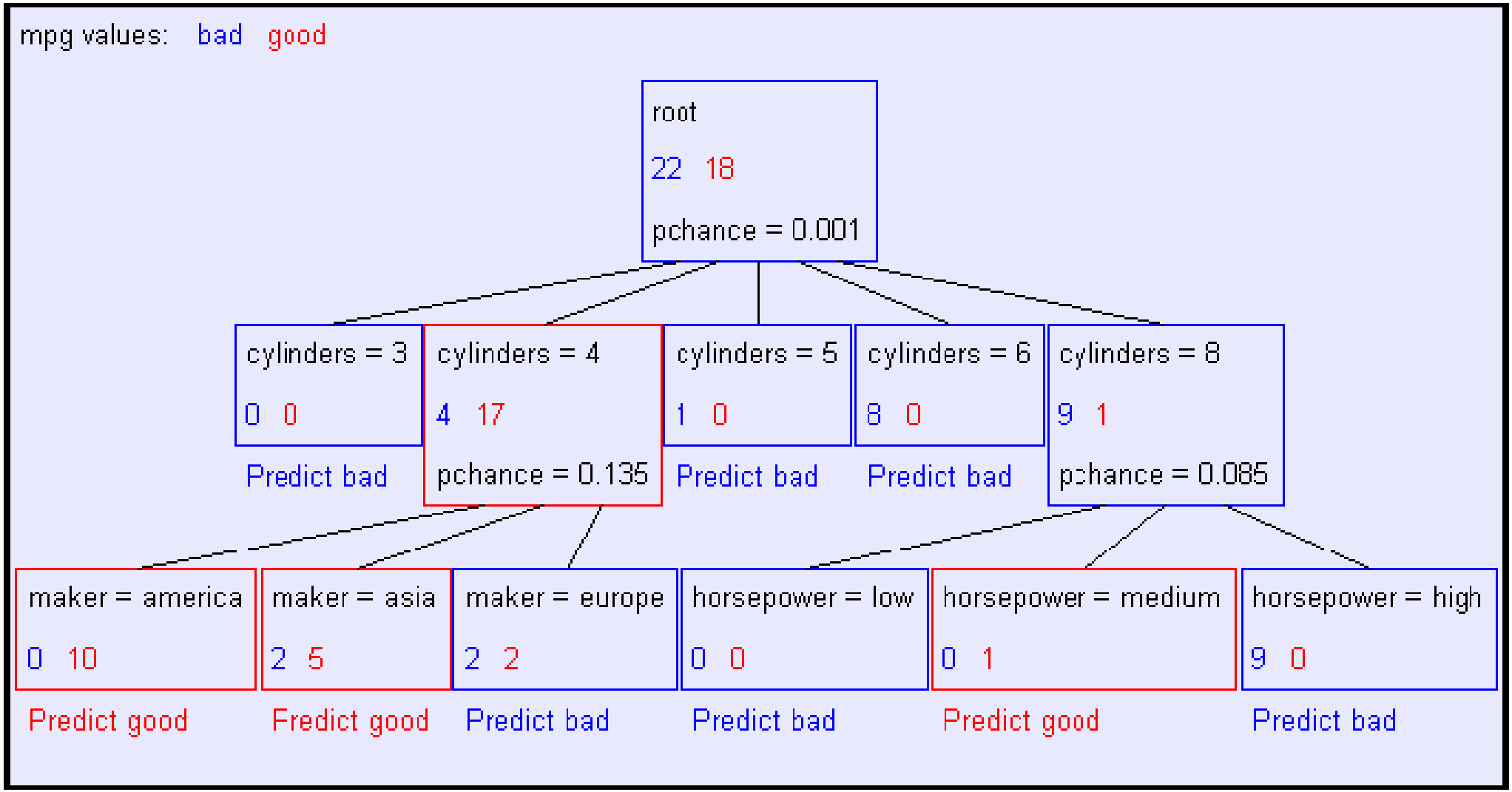
mpg values: bad good

| Input | Value | Distribution | Info Gain |
|---|---|---|---|
| cylinders | 3 | | 0.506731 |
| | 4 | | |
| | 5 | | |
| | 6 | | |
| | 8 | | |
| displacement | low | | 0.223144 |
| | medium | | |
| | high | | |
| horsepower | low | | 0.387605 |
| | medium | | |
| | high | | |
| weight | low | | 0.304018 |
| | medium | | |
| | high | | |
| acceleration | low | | 0.0642088 |
| | medium | | |
| | high | | |
| modelyear | 70to74 | | 0.267964 |
| | 75to78 | | |
| | 79to83 | | |
| maker | america | | 0.0437265 |
| | asia | | |

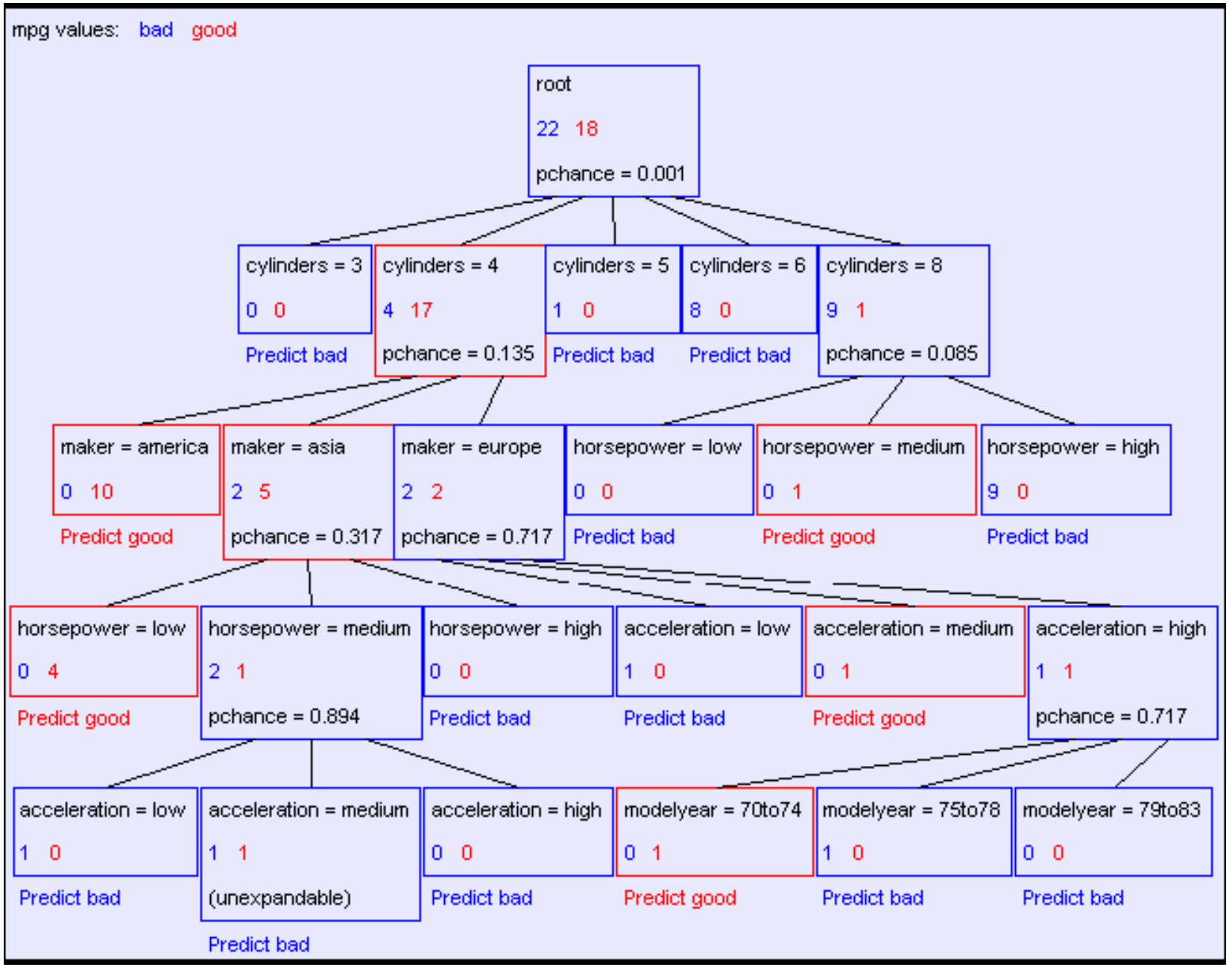
# Entscheidungsbaumstumpf



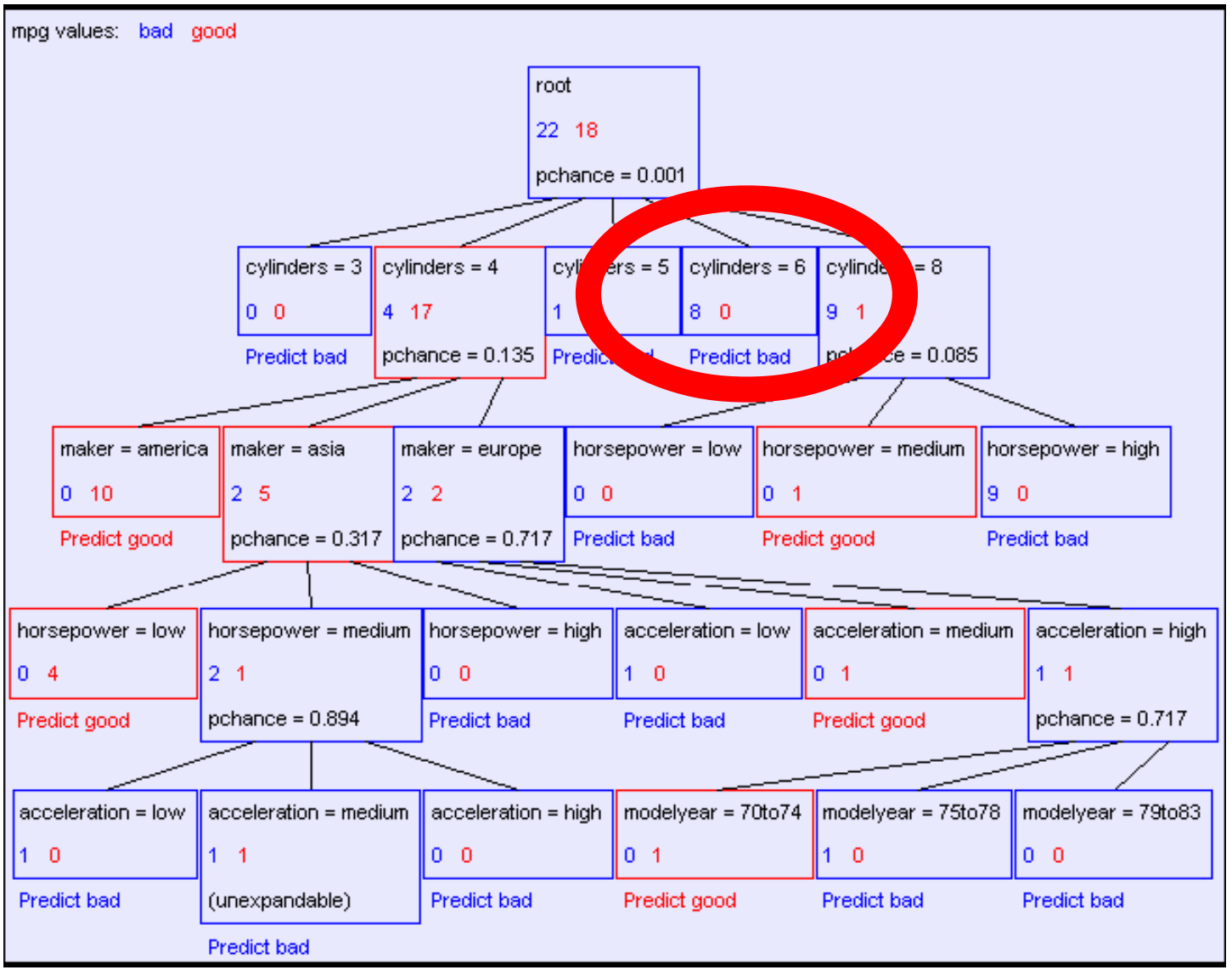
# Nächster Rekursionschritt



# Vollständiger Entscheidungsbaum

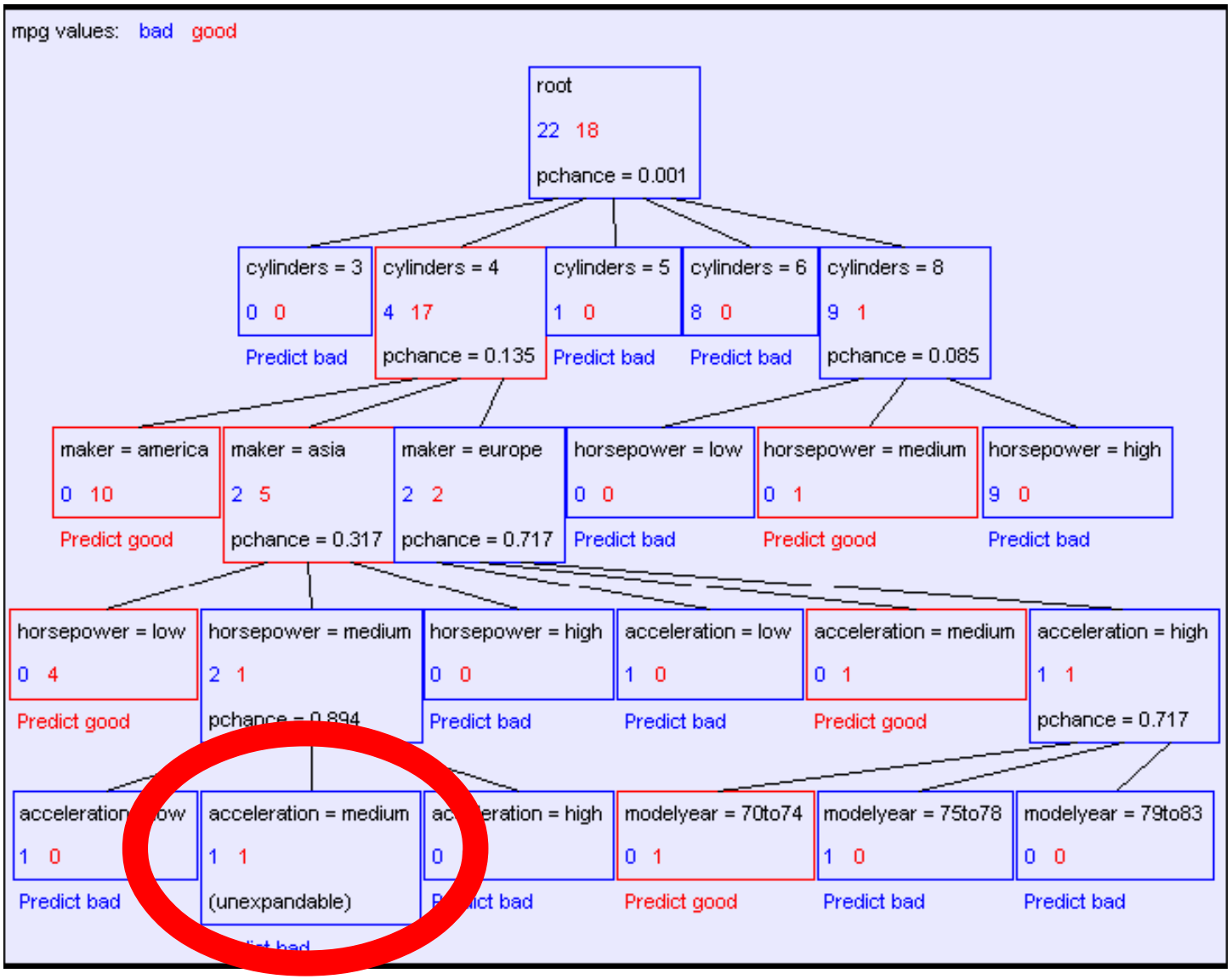


# Rekursionsabbruch: 1. Fall




Alle verbleibenen Records haben denselben Zielwert

# Rekursionsabbruch: 2. Fall




Information gains using the training set (2 records)

mpg values: bad good

| Input | Value | Distribution | Info Gain |
|---|---|---|---|
| cylinders | 3 | | 0 |
| | 4 | | |
| | 5 | | |
| | 6 | | |
| | 8 | | |
| displacement | low | | 0 |
| | medium | | |
| | high | | |
| horsepower | low | | 0 |
| | medium | | |
| | high | | |
| weight | low | | 0 |
| | medium | | |
| | high | | |
| acceleration | low | | 0 |
| | medium | | |
| | high | | |
| modelyear | 70to74 | | 0 |
| | 75to78 | | |
| | 79to83 | | |
| maker | america | | 0 |
| | asia | | |
| | europe | | |

Alle verbleibenen Records haben dieselben Attribute

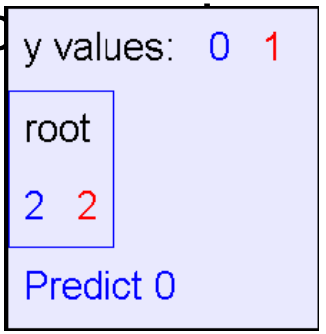
# Rekursionsabbruch: 3. Fall

Alle verbleibenden Attribute haben null Informatic

Ist das eine gute Idee?

XOR-Problem a, b -> y:

Mit Fall 3

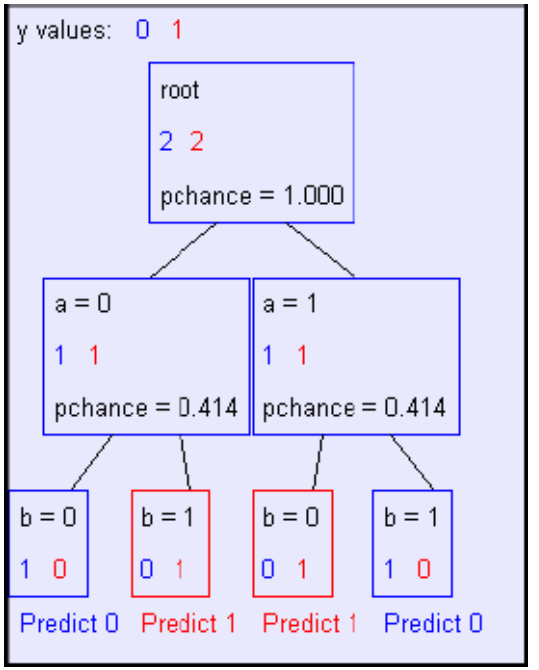



Information gains using the training set (4 records)

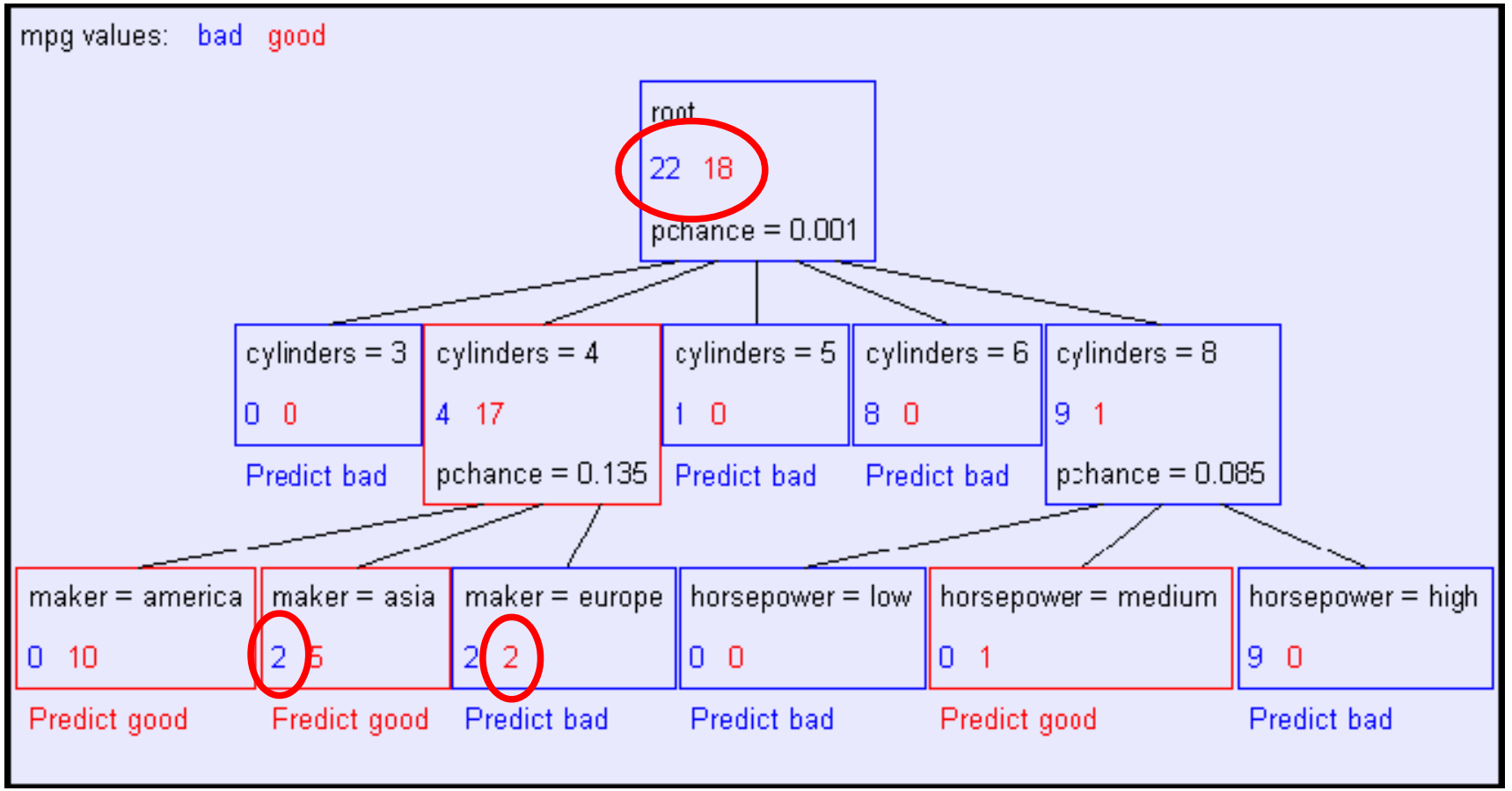
y values: 0 1

| Input | Value | Distribution | Info Gain |
|---|---|---|---|
| a | 0 | | 0 |
| | 1 | | |
| b | 0 | | 0 |
| | 1 | | |

Ohne Fall 3



# Trainingsetfehler



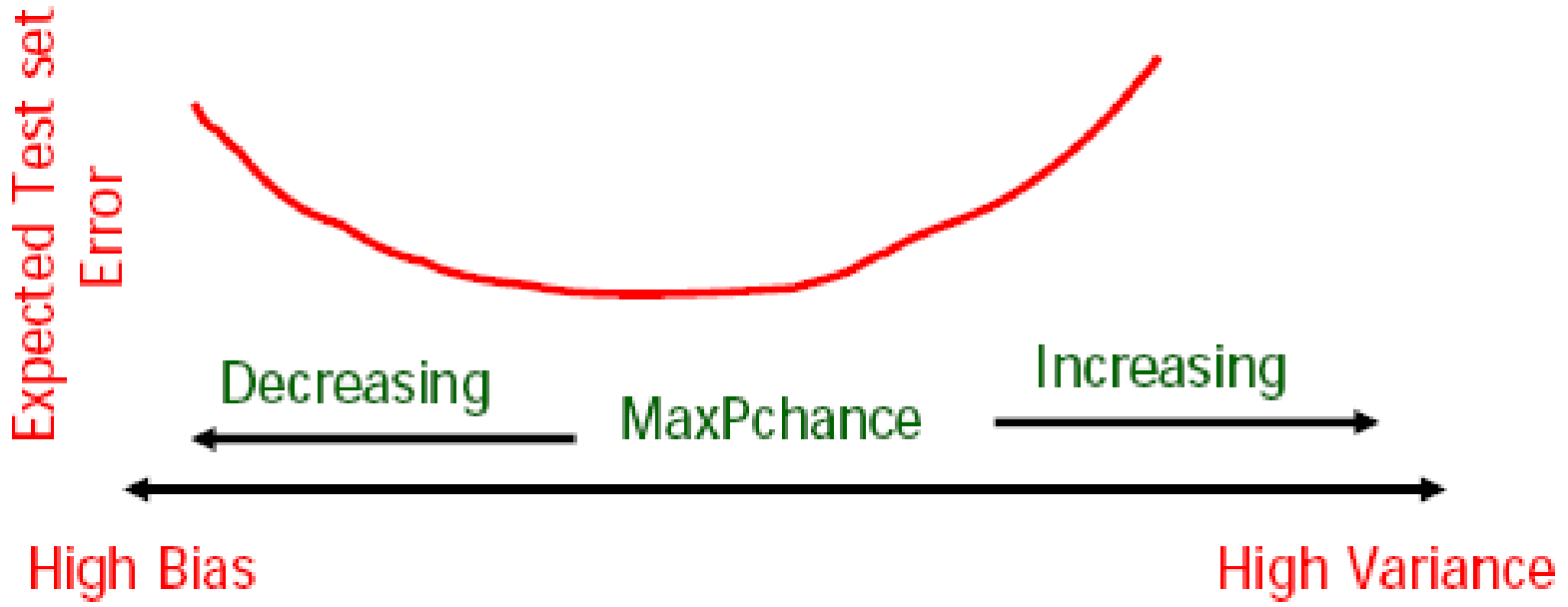

Für wieviele Records ist die Vorhersage falsch?
4 von 40 = 10%

# Overfitting

- Trainingssetfehler vs. Testsetfehler
- Modellierung von Rauschen
- Vermeiden von Overfitting
  - Unter der Annahme, dass der Label nicht mit dem Attribut korreliert ist, wie wahrscheinlich ist die Beobachtung (pchance).
  - Abschneiden des Teilbaums, wenn diese Wahrscheinlichkeit größer einem Schwellwert ist
    (pchance > MaxPchance)

# Bias-Varianz-Dilemma

- Modellparameter MaxPchance




- Schätzung von MaxPChance mittels Kreuzvalidierung

# Numerische Attribute

Thresholded Splits:

Definition: IG(Y|X:t) = H(Y) – H(Y|X:t)

wobei H(Y|X:t) =
H(Y|X < t) P(X < t) + H(Y|X >= t) P(X >= t)

IG(Y|X:t) ist der Informationsgewinn für die Vorhersage von Y, gegeben dass bekannt ist, ob X größer gleich bzw. kleiner als t ist.

IG*(Y|X) ist definiert als $\max_t$ IG(Y|X:t)

# Entscheidungstheorie

# Entscheidungstheorie

- Bewertet die Kosten einer Klassifikationsentscheidung mit Hilfe von Wahrscheinlichkeitsaussagen
- Annahme: alle Wahrscheinlichkeiten sind im Vorhinein bekannt

# Begriffe

- A-priori Wahrscheinlichkeit $p(w_i)$
  - Vorwissen
  - Summe aller $p(w_i) = 1$
- Klassenbedingte Wahrscheinlichkeit $p(x \mid w_i)$
  - Messung hängt von der tatsächlichen Klasse ab
- Gemeinsame Verteilung $p(x, w_i) = p(x \mid w_i)\, p(w_i)$
  - [ Kettenregel: $P(A,B,C) = P(A|B,C)\, P(B|C)\, P(C)$ ]

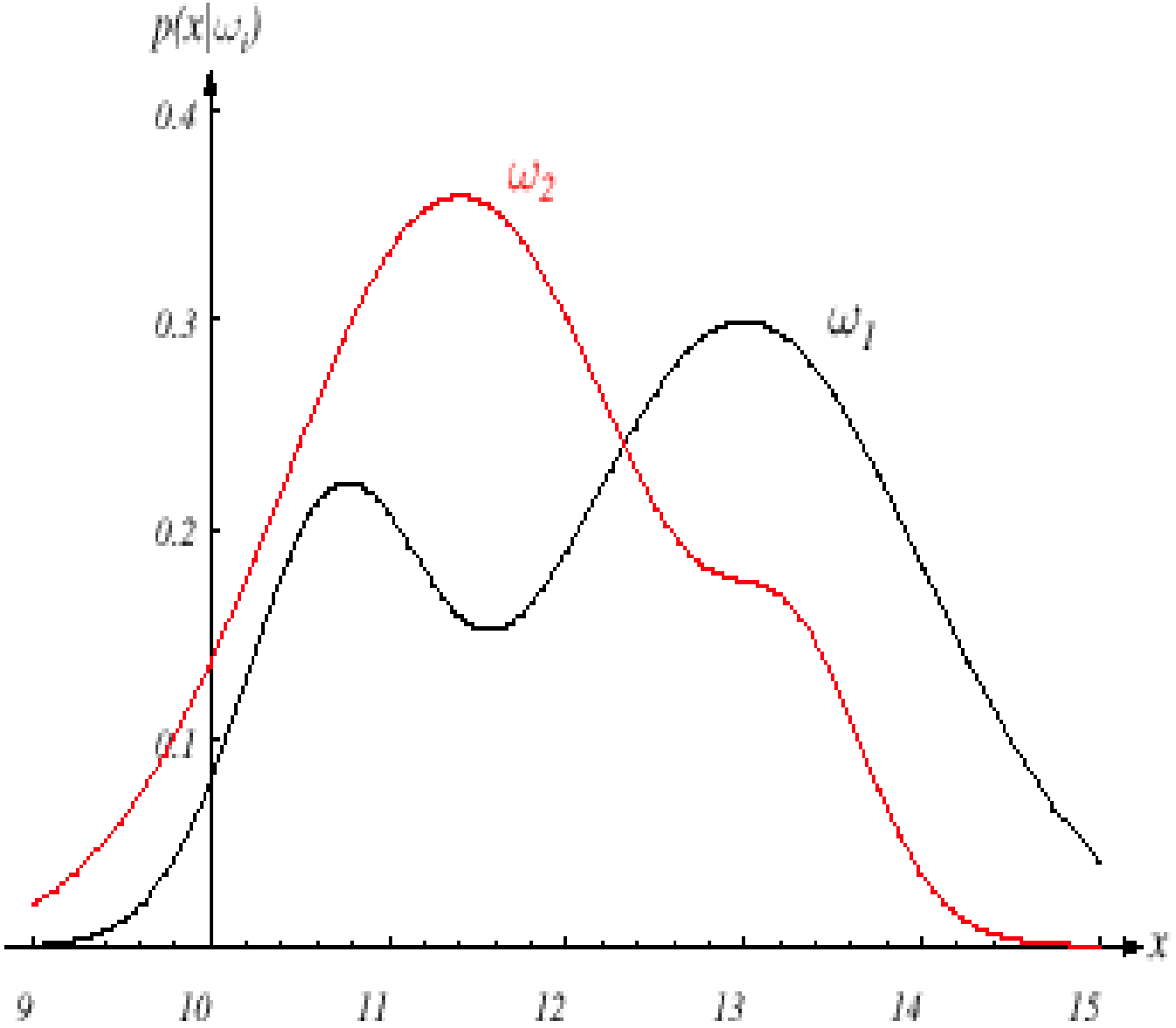
# Satz von Bayes

$$p(w_i|x) = \frac{p(x|w_i)\,p(w_i)}{p(x)}$$

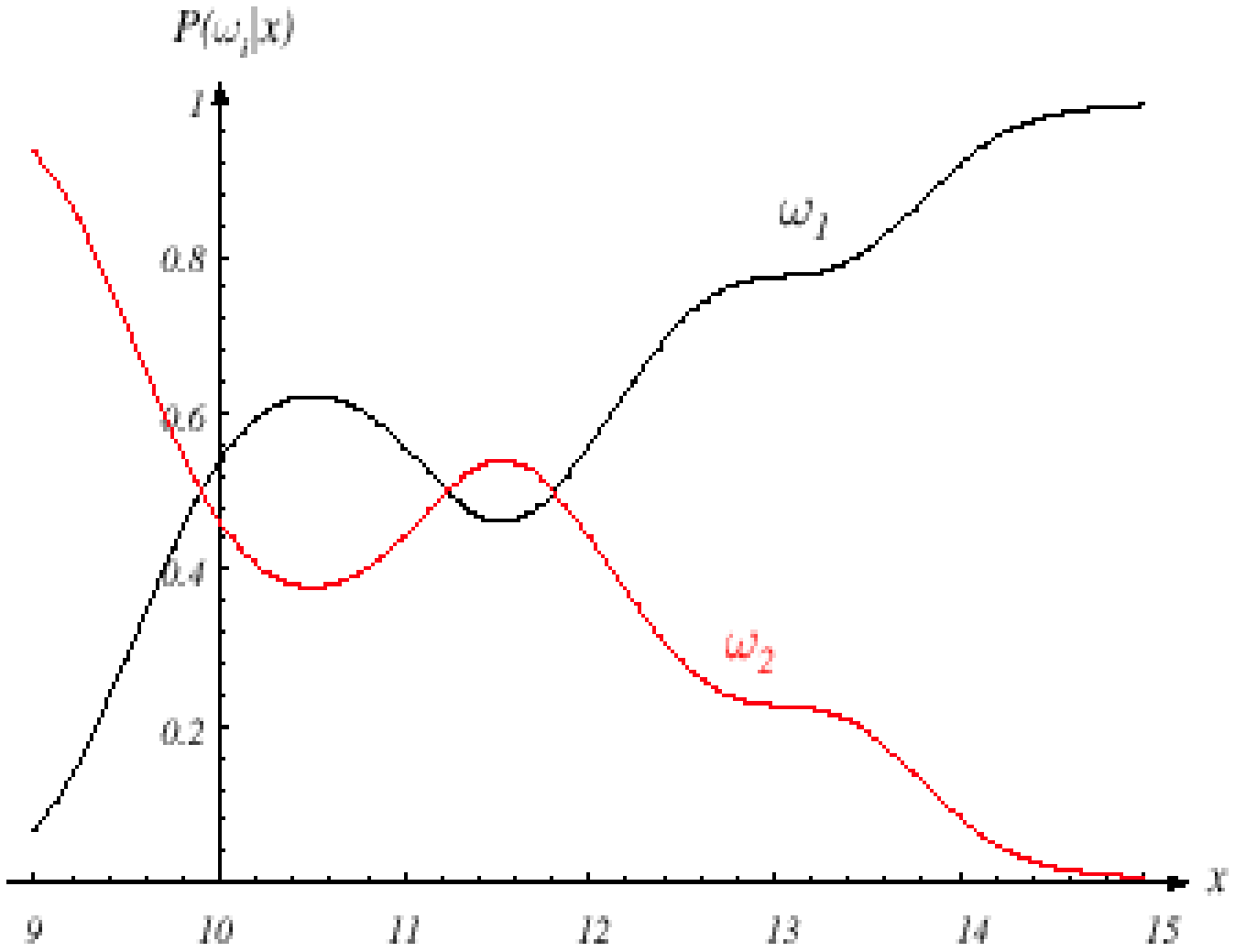
$$\text{mit} \quad p(x) = \sum_{i=1}^{n} p(x|w_i)\,p(w_i)$$

$$\text{A-posteriori-Wahrscheinlichkeit} = \frac{\text{Likelihood x A-priori-Wahrscheinlichkeit}}{\text{Evidenz}}$$

# Likelihood



# A-posteriori



# Bayessche Entscheidungsregel

Entscheide für $w_1$ wenn

$p(w_1 \mid x) > p(w_2 \mid x)$ sonst für $w_2$

bzw.

$p(x \mid w_1)\, p(w_1) > p(x \mid w_2)\, p(w_2)$

# Verallgemeinerung

- Mehr als 1 Merkmal
  - Variable x wird durch d-dimensionalen Merkmalsvektor X ersetzt
- Mehr als 2 Optionen (Ja/Nein)
  - a Aktionen $\{a_1, ..., a_a\}$
- Mehr als 2 Klassen
  - Mehrklassenproblem, c Klassen $\{w_1, ..., w_c\}$
- Kostenfunktion
  - $\lambda(a_i \mid w_j)$ Kosten der Aktion $a_i$ für Klasse $w_j$

# Verallgemeinerte Bayessche Entscheidungsregel

$$p(w_i|x) = \frac{p(x|w_i)\,p(w_i)}{p(x)}$$

$$\text{mit} \quad p(x) = \sum_{i=1}^{n} p(x|w_i)\,p(w_i)$$

$$R(a_i|x) = \sum_{j=1}^{c} \lambda(a_i|w_j)\,p(w_j|x)$$

Wähle immer die Aktion $a_i$, die das bedingte Risiko für die gegebene Beobachtung x minimiert

# 2 Klassen, 2 Aktionen

- Vereinfachte Kostenfunktion $\lambda_{ij} = \lambda(a_i \mid w_j)$
- Bedingtes Risiko
  - $R(a_1 \mid x) = \lambda_{11}\, p(w_1 \mid x) + \lambda_{12}\, p(w_2 \mid x)$
  - $R(a_2 \mid x) = \lambda_{21}\, p(w_1 \mid x) + \lambda_{22}\, p(w_2 \mid x)$
- Entscheide für $a_1$, wenn

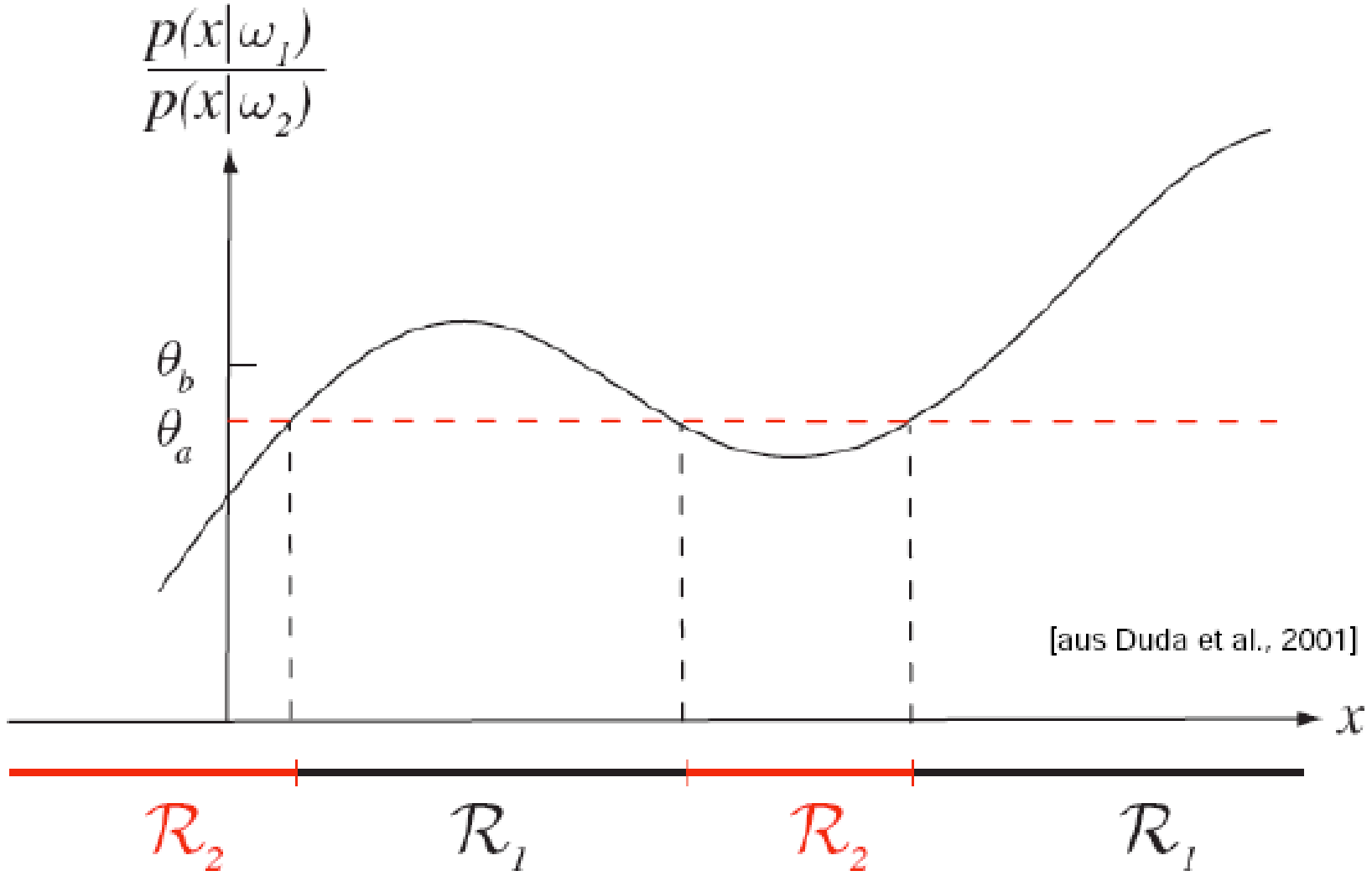
$$R(a_1|x) < R(a_2|x) \Leftrightarrow \lambda_{11}\, p(w_1 \mid x) + \lambda_{12}\, p(w_2 \mid x) < \lambda_{21}\, p(w_1 \mid x) + \lambda_{22}\, p(w_2 \mid x)$$

$$(\lambda_{11} - \lambda_{21})\, p(w_1 \mid x) < (\lambda_{22} - \lambda_{12})\, p(w_2 \mid x)$$

$$(\lambda_{11} - \lambda_{21})\, p(x \mid w_1)\; p(w_1) < (\lambda_{22} - \lambda_{12})\, p(x \mid w_2)\, p(w_2)$$

$$\frac{p(x \mid w_1)}{p(x \mid w_2)} > \frac{(\lambda_{12} - \lambda_{22})\, p(w_2)}{(\lambda_{21} - \lambda_{11})\, p(w_1)}$$ Likelihood ratio

# Likelihood Ratio



# Klassifikation mit minimaler Fehlerrate

- Aktionen: $a_i$ ... Klassifiziere als $w_i$
- Kostenfunktion:
  $\lambda_{ij} = 0$ für i=j  Keine Kosten für korrekte Klassifikation
  $\lambda_{ij} = 1$ für i≠j  „Bestrafung“ für falsche Klassifikation

| | **w1** | **w2** |
|---|---|---|
| **a1** | Hit (TP)<br>True Positive | False Alarm (FP)<br>False Positive |
| **a2** | Miss (FN)<br>False Negative | Reject (TN)<br>True Negative |

(Confusion Matrix)

Accuracy = (tp+tn) / (tp+fp+fn+tn)

Precision = tp / (tp+fp)

Sensitivity = tp / (tp+fn)

Recall = tp / (tp+fn)

Specificity = tn / (tn+fp)

# Klassifikation mit minimaler Fehlerrate

- Bedingtes Risiko $R(a_i \mid x) = p(w_j \mid x) = 1 - p(w_i \mid x)$ (für $i \neq j$)
- Entscheidungsfunktion: entscheide für $w_1$

  $R(a_1 \mid x) < R(a_2 \mid x)$

  $1 - p(w_1 \mid x) < 1 - p(w_2 \mid x)$

  $p(w_1 \mid x) > p(w_2 \mid x)$
- Bayesfehler: Minimal erreichbarer Fehler

  $p(\text{Fehler}) = p(x \in R_1 \mid w_2)\, p(w_2) + p(x \in R_2 \mid w_1)\, p(w_1)$

# Bayes'sche Klassifikation

# Gemeinsame Verteilung
# Joint Distribution

| A | B | C | Prob |
|---|---|---|---|
| 0 | 0 | 0 | 0,10 |
| 0 | 0 | 1 | 0,25 |
| 0 | 1 | 0 | 0,02 |
| 0 | 1 | 1 | 0,05 |
| 1 | 0 | 0 | 0,55 |
| 1 | 0 | 1 | 0,01 |
| 1 | 1 | 0 | 0,01 |
| 1 | 1 | 1 | 0,01 |
| | | | **1,00** |

Prob = Anzahl passender Records / Gesamtanzahl Records

# Bedingte Wahrscheinlichkeit
# Conditional Probabilities

| A | B | C | Prob |
|---|---|---|---|
| 0 | 0 | 0 | 0,1 |
| 0 | 0 | 1 | 0,25 |
| 0 | 1 | 0 | 0,02 |
| 0 | 1 | 1 | 0,05 |
| 1 | 0 | 0 | 0,55 |
| 1 | 0 | 1 | 0,01 |
| 1 | 1 | 0 | 0,01 |
| 1 | 1 | 1 | 0,01 |
| | | | **1,00** |

| A | B | C | P(A^B^C=0) | P(A^B\|C=0) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0,1 | 0,15 |
| 0 | 1 | 0 | 0,02 | 0,03 |
| 1 | 0 | 0 | 0,55 | 0,81 |
| 1 | 1 | 0 | 0,01 | 0,01 |
| | | | **0,68** | **1,00** |
| | | | P(C=0) | |

| A | B | C | P(A^B^C=1) | P(A^B\|C=1) |
|---|---|---|---|---|
| 0 | 0 | 1 | 0,25 | 0,78 |
| 0 | 1 | 1 | 0,05 | 0,16 |
| 1 | 0 | 1 | 0,01 | 0,03 |
| 1 | 1 | 1 | 0,01 | 0,03 |
| | | | **0,32** | **1,00** |
| | | | P(C=1) | |

# Bayesscher Klassifizierer

- Wie sieht der Bayessche Klassifizierer für gemeinsame Verteilungen aus?
  - Häufigste Klasse für zu klassifizierende Attribute
  - Wenn Input unbekannt muss Klasse geraten werden

# Bayesscher Klassifizierer

Wie baue ich mir einen Bayesschen Klassifizierer?

- m (Eingabe-)Attribute $X_1$, ..., $X_m$
- (Ausgabe-)Attribut Y mit möglichen Werten $v_1$, ..., $v_n$
- Aufteilung der Daten in n Sets $DS_i$
  - $DS_i$ enthält nur Daten mit $Y = v_i$
- Für jedes $DS_i$ wird ein Schätzer $M_i$ gelernt
  - $M_i$ schätzt $P(X_1, ..., X_m \mid Y = v_i)$

# Bayesscher Klassifizierer

Wie baue ich mir einen Bayesschen Klassifizierer?

- Lerne Schätzer für P(Y = $v_i$)
  - Verteilung der Records auf $DS_i$
- Zur Vorhersage:
  - $Y^{predict}$ = argmax
    = argmax $P(Y=v|X_1=u_1,...,X_m=u_m)$
    $P(X_1=u_1,...,X_m=u_m|Y=v)\,P(Y=v)$

# MAP Classifier

Maximum A-posteriori

- $Y^{predict}$ = argmax $P(Y = v_i \mid X_1, ..., X_m)$

$$P(Y=v|X_1=u_1,...,X_m=u_m)$$
$$=\frac{P(X_1=u_1,...,X_m=u_m|Y=v)\,P(Y=v)}{P(X_1=u_1,...,X_m=u_m)}$$
$$=\frac{P(X_1=u_1,...,X_m=u_m|Y=v)\,P(Y=v)}{\sum_{i=1}^{n} P(X_1=u_1,...,X_m=u_m|Y=v_i)\,P(Y=v_i)}$$

# MLE Classifier

Maximum Likelihood Estimator

- Vorhersage der Klasse für neuen Input $(X_1 = u_1, ..., X_m = u_m)$
- $Y^{predict} = \text{argmax}\ P(X_1, ..., X_m \mid Y = v_i)$
  - MLE maximum likelihood estimator
  - Was passiert wenn manche Werte von Y sehr unwahrscheinlich sind?

# Dichteschätzung

# Dichteschätzung

- Bayessche Klassifikation benötigt Wahrscheinlichkeitsverteilung
  - Experten
  - Ein paar Fakten und Algebra (cf. Bayessche Netze)
  - Lernen aus Daten: Dichteschätzung
    - Attribute => Dichtefunktion
    - (cf. Decision Tree: Attribute => Klasse
      Regression: Attribute => reelle Zahl)

# Schätzen von Wahrscheinlichkeitsdichten

- Parametrisch, z.B. Normalverteilung:
  - $\mu = E\{x\}$ Mittelwert
  - $\sigma^2 = E\{(x-\mu)^2\}$ Varianz

  (Iterative Berechnung möglich)
  - Trennflächen max. 2. Ordnung
  - Bei gleichen Kovarianzmatrizen aller Klassen:
    - Trennebene 1. Ordnung

# Andere Methoden zur Dichteschätzung

- (Gaussian) Mixture Models
- Bayessche Netze
- Density Trees
- Kernel Densities (Parzen Window)

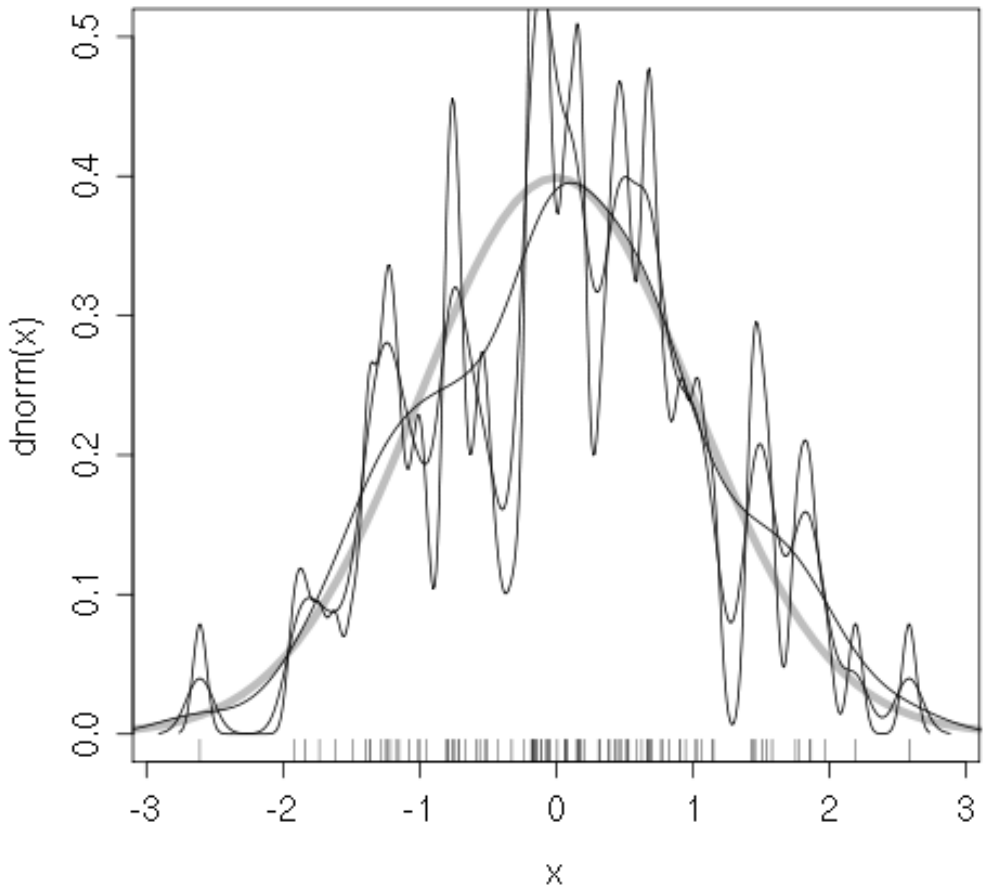
# Kernel Density Estimation

- Von Datenpunkten zu Probability Density
  Parzen window method

$$\widehat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

mit K z.B.:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$



# Evaluation des Schätzers

- Üblicherweise: Genauigkeit auf Testdaten
- Wie wahrscheinlich ist ein Record x für einen Schätzer M: $\hat{P}(x|M)$
- Wie wahrscheinlich ist eine Datenmenge mit R records

$$\hat{P}(Dataset|M) = \hat{P}(x_1 \wedge x_2 \wedge \cdots \wedge x_R|M) = \prod_{k=1}^{R} \hat{P}(x_k|M)$$

- Log P

$$\log \hat{P}(Dataset|M) = \log \prod_{k=1}^{R} \hat{P}(x_k|M) = \sum_{k=1}^{R} \log \hat{P}(x_k|M)$$

zur Vermeidung von numerischer Instabilität

- Wie wahrscheinlich ist für einen Schätzer ein neuer Record? Overfitting!

# Ausblick

- Nächste Termine:

**Donnerstag, 17.11.2016 13-15 (c.t.)**

Übung 1

**Donnerstag, 24.11.2016 13-15 (c.t.)**

Nicht-lineare Klassifikation, Clustering, Boosting, ...