

# **Techniken des Maschinellen Lernens für Data Mining**

**Ian Witten, Eibe Frank**

**(übersetzt von Norbert Fuhr)**

# 3 Ausgabe: Wissensrepräsentation

- ❖ Entscheidungstabellen
- ❖ Entscheidungsbäume
- ❖ Entscheidungsregeln
- ❖ Assoziationsregeln
- ❖ Regeln mit Ausnahmen
- ❖ Regeln mit Relationen
- ❖ Lineare Regression
- ❖ Bäume zur numerischen Vorhersage
- ❖ Instanz-basierte Repräsentation
- ❖ Cluster



# Repräsentation struktureller Muster

- ❖ Viele Möglichkeiten, um Muster darzustellen
  - Entscheidungsbäume, Regeln, Instanz-basiert
- ❖ Auch “Wissensrepräsentation” genannt
- ❖ Repräsentation bestimmt die Inferenzmethode
- ❖ Verständnis der Ausgabe ist der Schlüssel zum Verständnis der zugrundeliegenden Lernmethoden
- ❖ Verschiedene Arten der Ausgabe für verschiedene Lernprobleme (z.B. Klassifikation, Regression)

# Entscheidungstabellen

❖ Rudimentärste Form der Ausgabe-Repräsentation:

❑ Benutzt dasselbe Format wie die Eingabe!

❖ Entscheidungstabelle für das Wetterproblem:

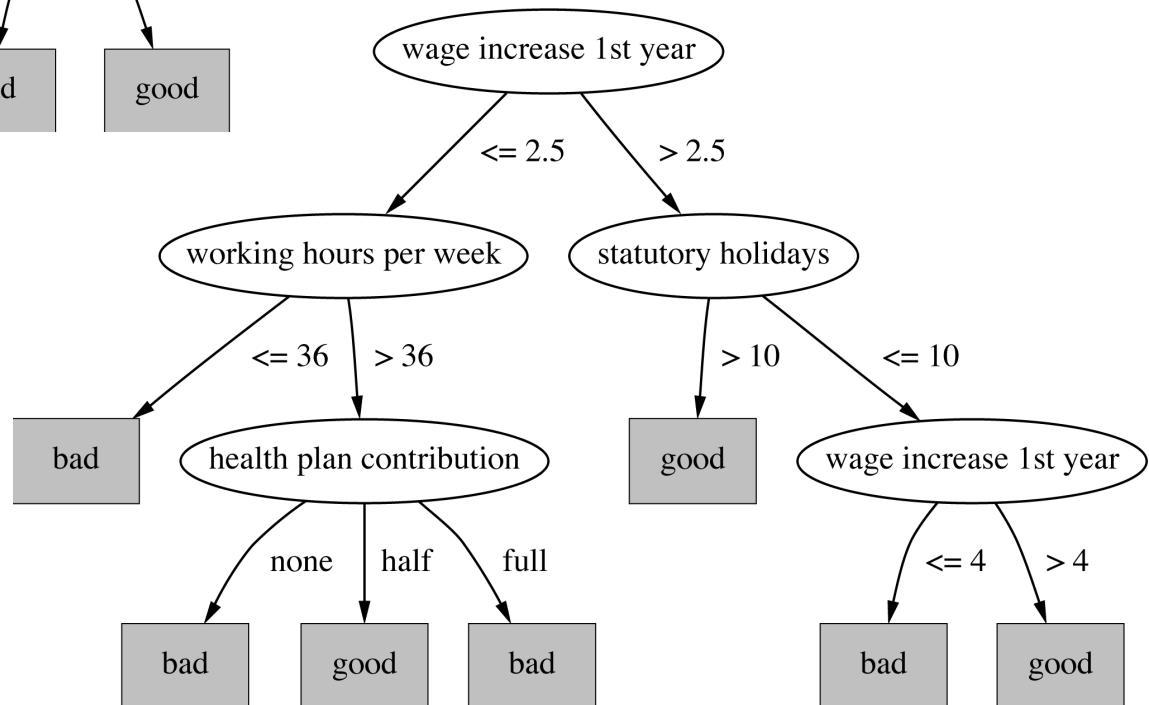
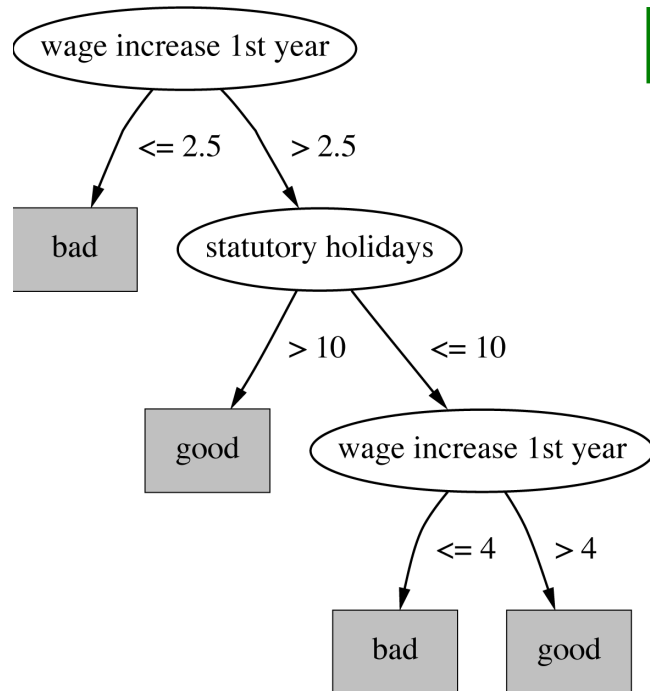
Outlook	Humidity	Play
Sunny	High	No
Sunny	Normal	Yes
Overcast	High	Yes
Overcast	Normal	Yes
Rainy	High	No
Rainy	Normal	No

❖ Hauptproblem: Auswahl der richtigen Attribute

# Entscheidungsbäume

- ❖ "Teile und herrsche"-Ansatz produziert Baum
- ❖ Jeder Knoten testet ein bestimmtes Attribut
- ❖ Normalerweise wird der Attributwert mit einer Konstanten verglichen
- ❖ Andere Möglichkeiten:
  - Vergleich der Werte zweier Attribute
  - Betrachte Funktionswert eines oder mehrerer Attribute
- ❖ Blätter weisen den Instanzen Klassen, Mengen von Klassen oder Wahrscheinlichkeitsverteilungen zu
- ❖ Unbekannte Instanz durchläuft den Baum von der Wurzel bis zu einem Blatt

# Entscheidungs- bäume für die Tarifdaten



# Die Kontaktlinsen-Daten

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

# Wetterdaten mit gemischten Attributen

## ❖ Zwei Attribute mit numerischen Werten

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
...	...	...	...	...

If outlook = sunny and humidity > 83 then play = no

If outlook = rainy and windy = true then play = no

If outlook = overcast then play = yes

If humidity < 85 then play = yes

If none of the above then play = yes



# Klassifikation von Irisblumen



	Sepal length	Sepal width	Petal length	Petal width	Type
1	5.1	3.5	1.4	0.2	Iris setosa
2	4.9	3.0	1.4	0.2	Iris setosa
...					
51	7.0	3.2	4.7	1.4	Iris versicolor
52	6.4	3.2	4.5	1.5	Iris versicolor
...					
101	6.3	3.3	6.0	2.5	Iris virginica
102	5.8	2.7	5.1	1.9	Iris virginica
...					

```
If petal length < 2.45 then Iris setosa
If sepal width < 2.10 then Iris versicolor
...
```

# Nominale und numerische Attribute

- ❖ Nominales Attribut:
  - normalerweise: Anzahl der Kinder = Anzahl der möglichen Werte  $\Rightarrow$  Jedes Attribut wird höchstens einmal getestet
    - ❑ Andere Möglichkeit: Aufteilung in zwei Teilmengen
- ❖ numerisches Attribut:
  - Test, ob Attributwert größer gleich einer Konstanten ist  $\Rightarrow$  Attribut kann mehrfach getestet werden
    - ❑ Andere Möglichkeit: Drei-Wege-Split (oder Mehr-Wege-Split)
      - Integer: *kleiner, gleich, größer*
      - Real: *unterhalb, innerhalb, oberhalb*

# Fehlende Werte

- ❖ Hat das Fehlen eines Wertes eine Bedeutung?
- ❖ Ja  $\Rightarrow$  “fehlt” ist ein spezieller Wert
- ❖ Nein  $\Rightarrow$  “fehlt” muss gesondert behandelt werden
  - ❑ Lösung A: Weise Instanz dem Zweig mit dem häufigsten Attributwert zu
  - ❑ Lösung B: Instanz aufsplitten
    - Jeder Teil erhält Gewichtung entsprechend der relativen Häufigkeit der Trainingsinstanzen in diesem Zweig
    - Klassen aus den Blättern werden unter Berücksichtigung dieser Gewichtungen kombiniert

# Klassifikationsregeln

- ❖ Populäre Alternative zu Entscheidungsbäumen
- ❖ *Antezedent* (Vorbedingung): Reihe von Tests (vergleichbar den Tests in den Knoten eines Entscheidungsbaums)
- ❖ Tests werden normalerweise mit UND verknüpft (können aber auch allgemeine logische Ausdrücke sein)
- ❖ *Konsequenz* (Folgerung): Regel weist Klasse, Menge von Klassen oder Wahrscheinlichkeits-verteilung zu
- ❖ Einzelne Regeln werden oft mit ODER verknüpft
  - ❑ Konflikte, falls unterschiedliche Folgerungen für eine Instanz möglich sind

# Von Bäumen zu Regeln

- ❖ Einfach: Konvertieren eines Baums in eine Regelmenge
  - ❑ Eine Regel pro Blatt:
    - Antezedent enthält eine Bedingung für jeden Knoten auf dem Weg von der Wurzel zum Blatt
    - Konsequenz ist die durch das Blatt zugewiesene Klasse
- ❖ Generiert eindeutige Regeln
  - ❑ Reihenfolge der Abarbeitung egal
- ❖ Aber: erzeugte Regeln sind unnötig komplex
  - ❑ Pruning (Beschneiden), um redundante Tests/Regeln zu eliminieren

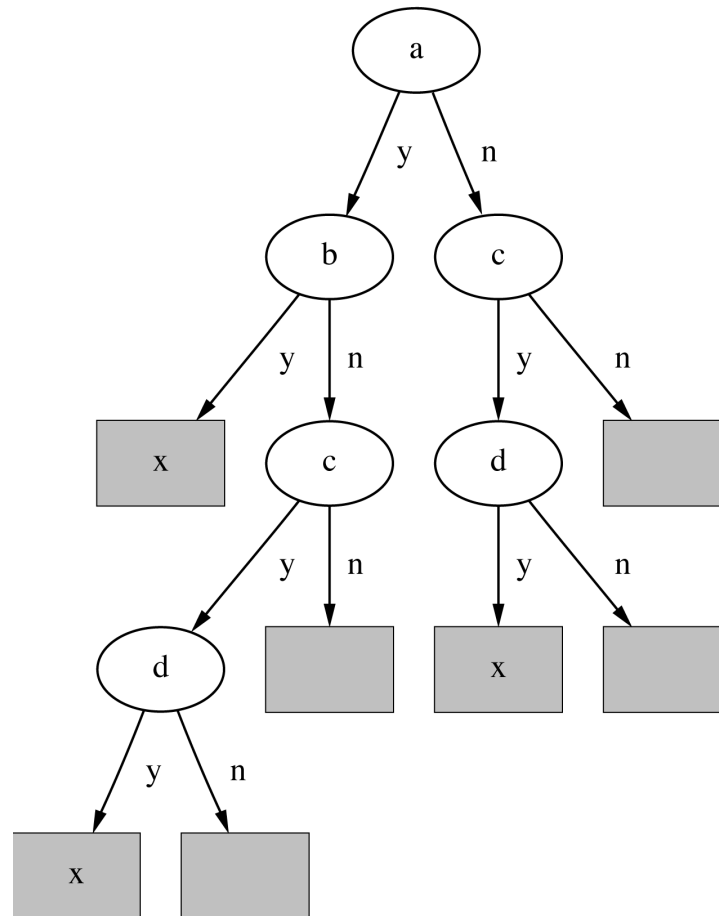
# Von Regeln zu Bäumen

- ❖ Schwieriger: Transformation einer Regelmenge in einen Baum
  - Baum kann disjunktive Verknüpfung der Regeln nur schwer darstellen
- ❖ Beispiel: Regeln, die verschiedene Attribute testen

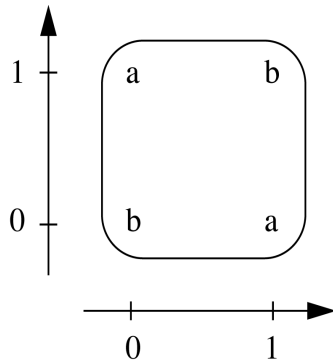
```
If a and b then x  
If c and d then x
```

- ❖ Symmetrie muss aufgebrochen werden
- ❖ Korrespondierender Baum enthält identische Teilbäume ( $\Rightarrow$  "Problem der replizierten Teilbäume")

# Ein Baum für eine einfache Disjunktion



# Das EXOR-Problem

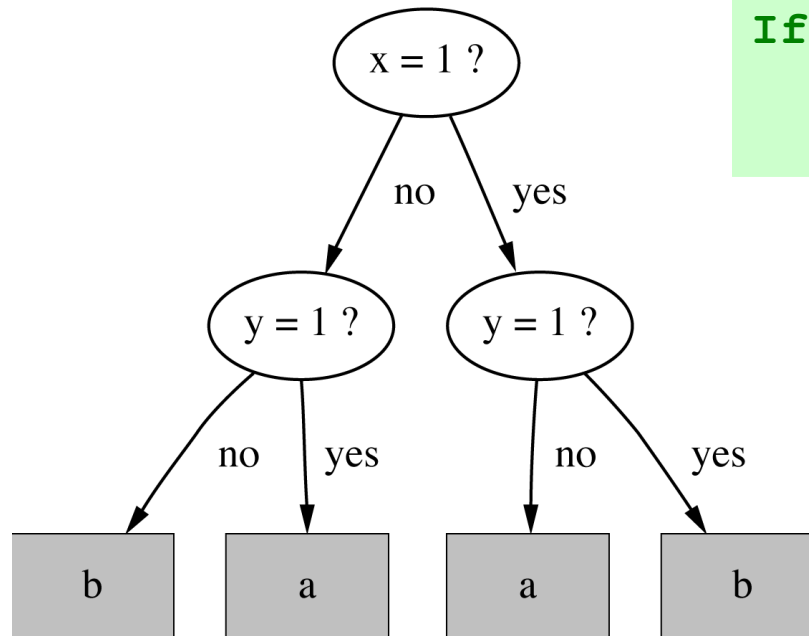


If  $x = 1$  and  $y = 0$   
then class = a

If  $x = 0$  and  $y = 1$   
then class = a

If  $x = 0$  and  $y = 0$   
then class = b

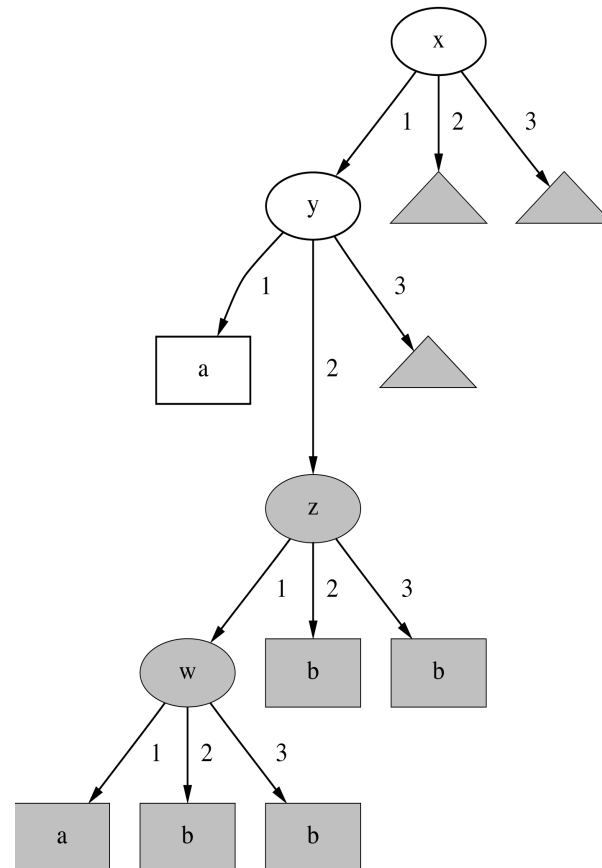
If  $x = 1$  and  $y = 1$   
then class = b





# Ein Baum mit einem replizierten Teilbaum

If  $x = 1$  and  $y = 1$   
then class = a  
If  $z = 1$  and  $w = 1$   
then class = a  
Otherwise class = b



# Kern des repräsentierten Wissens

- ❖ Regeln als unabhängige Wissensfragmente? (Regeln können einfach zu einer Regelbasis hinzugefügt werden)
- ❖ Problem: ignoriert, wie Regeln ausgeführt werden
- ❖ Zwei Arten der Regelinterpretation:
  - ❑ Geordnete Liste von Regeln (Entscheidungsliste)
    - Ordnung ist wichtig für die Interpretation
  - ❑ Ungeordnete Regelmenge
    - Regeln(-Antezedenten) können sich überlappen und zu verschiedenen Folgerungen für dieselbe Instanz führen

# Regelinterpretation

- ❖ Was geschieht, falls zwei Regeln im Konflikt zueinander stehen?
  - Keine Konklusion geben?
  - Die populärste Regel (gemäß den Trainingsdaten) auswählen
  - ...
- ❖ Was geschieht, falls keine Regel für eine Instanz anwendbar ist?
  - Keine Konklusion geben?
  - Die populärste Klasse (gemäß den Trainingsdaten) zuweisen
  - ...

# Spezialfall: Boolesche Fallunterscheidungen

- ❖ Annahme: Wenn eine Instanz nicht zum positiven Fall gehört, wird sie dem negativen Fall zugeordnet
- ❖ Nur Regeln für den positiven Fall lernen, Default-Regel gilt für den negativen Fall

```
If x = 1 and y = 1 then class = a  
If z = 1 and w = 1 then class = a  
Otherwise class = b
```

- ❖ Ordnung der Regeln unwichtig.  
Keine Konflikte!
- ❖ Regel kann in *disjunktiver Normalform* geschrieben werden

# Assoziationsregeln

- ❖ Assoziationsregeln
  - ❑ ... können jedes Attribut und jede Attributkombination vorhersagen
  - ❑ ... sind nicht dazu gedacht, gemeinsam als Menge genutzt zu werden
- ❖ Problem: große Anzahl von möglichen Assoziationen
  - ❑ Ausgabe muss beschränkt werden, um nur die stärksten Assoziationen auszugeben ⇒ nur die Regeln mit der größten Unterstützung und der höchsten Konfidenz

# Unterstützung und Konfidenz einer Regel

- ❖ Unterstützung: Anzahl der Instanzen, die korrekt vorhergesagt werden
- ❖ Konfidenz: Anzahl korrekter Vorhersagen im Verhältnis zur Gesamtzahl aller Instanzen, auf die die Regel anwendbar ist
- ❖ Beispiel: 4 kalte Tage mit normaler Luftfeuchtigkeit

```
If temperature = cool  
then humidity = normal
```

⇒ Unterstützung = 4, Konfidenz = 100%

- ❖ Meist werden Minima für Unterstützung und Konfidenz vorgegeben (z.B. 58 Regeln mit Unterstützung  $\geq 2$  und Konfidenz  $\geq 100\%$  für die Wetterdaten)

# Interpretation von Assoziationsregeln

❖ Interpretation ist nicht offensichtlich:

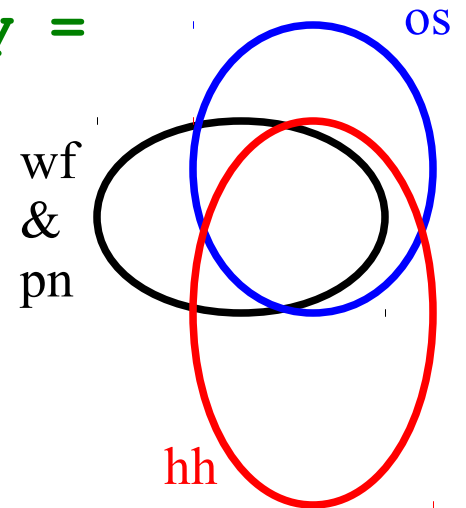
□ If windy = false and play = no  
then outlook = sunny and humidity = high

ist nicht das Gleiche wie

□ If windy = false and play = no  
then outlook = sunny  
If windy = false and play = no  
then humidity = high

❖ Allerdings wird die folgende Regel impliziert:

□ If windy = false and play = no and  
humidity = high  
then outlook = sunny



# Regeln mit Ausnahmen

❖ Idee: Ausnahmen bei Regeln erlauben

❖ Beispiel: Regel für die Iris-Daten:

□ If petal-length  $\geq 2.45$  and petal-length  $< 4.45$   
then Iris-versicolor

❖ Neue Instanz:

Septal length	Septal width	Petal length	Petal width	Type
5.1	3.5	2.6	0.2	Iris-setos

a

❖ Modifizierte Regel

□ If petal-length  $\geq 2.45$  and petal-length  $< 4.45$   
then Iris-versicolor

EXCEPT if petal-width  $< 1.0$  then Iris-setosa



# Ausnahmen zu Ausnahmen zu Ausnahmen...

default: Iris-setosa

except if petal-length  $\geq 2.45$  and petal-length  $< 5.355$   
and petal-width  $< 1.75$

then Iris-versicolor

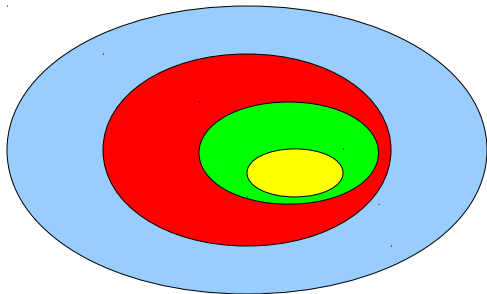
except if petal-length  $\geq 4.95$  and petal-width  $< 1.55$   
then Iris-virginica

else if sepal-length  $< 4.95$  and sepal-width  $\geq 2.45$   
then Iris-virginica

else if petal-length  $\geq 3.35$   
then Iris-virginica

except if petal-length  $< 4.85$  and sepal-length  $<$

then Iris-versicolor



# Vorteile bei der Benutzung von Ausnahmen

- ❖ Regeln können inkrementell aktualisiert werden
  - ❑ Einfache Berücksichtigung neuer Daten
  - ❑ Einfache Berücksichtigung von Hintergrundwissen
- ❖ Menschen denken oft in Ausnahmen
- ❖ Jede Konklusion kann gesehen werden als Kontext von Regeln und Ausnahmen, in der sie gilt
  - ❑ Lokalitätseigenschaft ist wichtig zum Verständnis großer Regelmengen
  - ❑ „Normale“ Regelmengen bieten diese Vorteile nicht

# Mehr über Ausnahmen

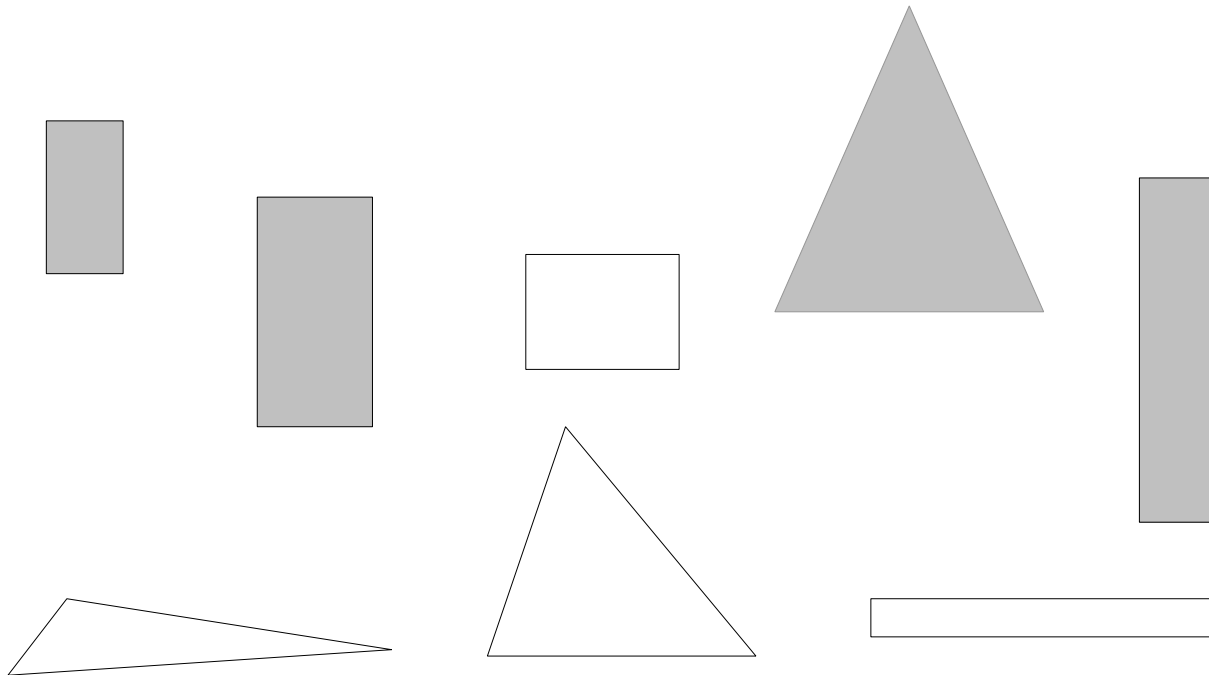
- ❖ `Default...except if...then...`  
ist logisch äquivalent zu  
`if...then...else`  
(wobei **else** für den Default-Fall steht)
- ❖ Aber: Ausnahmen bieten einen psychologischen Vorteil
  - ❑ Annahme: Defaults und frühe Tests gelten häufiger als Ausnahmen, die weiter unten formuliert werden
  - ❑ Ausnahmen behandeln Spezialfälle

# Relationale Regeln

- ❖ Bisher: Regeln vergleichen Attributwerte mit einer Konstanten (z.B. Temperatur < 25)
- ❖ Regeln werden “propositional” genannt, da sie dieselbe Ausdruckskraft wie Aussagenlogik besitzen
- ❖ Was passiert, wenn das Problem die Berücksichtigung von Beziehungen zwischen den Beispielen erfordert (wie z.B. im Familien-Stammbaum)
  - ❑ Kann nicht mit propositionalen Regeln formuliert werden
  - ❑ Ausdrucksstärkere Repräsentation erforderlich

# Das Gestalt-Problem

- ❖ Zielkonzept: *stehend*
- ❖ Schattiert: *stehend*  
Unschattiert: *liegend*



# Eine propositionale Lösung

Width	Height	Side	Class
2	4	4	Standing
3	6	4	Standing
4	3	4	Lying
7	8	3	Standing
7	6	3	Lying
2	9	4	Standing
9	1	4	Lying
10	2	3	Lying

```
If width  $\geq$  3.5 and height  $<$  7.0  
  then lying
```

```
If height  $\geq$  3.5 then standing
```

# Eine relationale Lösung

- ❖ Vergleich zwischen Attributwerten einer Instanz

```
If width > height then lying
```

```
If height > width then standing
```

- ❖ Bessere Generalisierung für neue Daten
- ❖ Standard-Relationen: =, <, >
- ❖ Aber: Lernen relationaler Regeln ist aufwändiger
- ❖ Einfache Lösung: Zusätzliche Attribute (z.B. binäres Attribut: *width < height?*)

# Regeln mit Variablen

- ❖ Benutzung von Variablen und mehreren Relationen:

```
If height_and_width_of(x,h,w) and h > w  
then standing(x)
```

- ❖ Die Spitze eines Turms steht:

```
If height_and_width_of(x,h,w) and h > w  
and is_top_of(x,y)  
then standing(x)
```

- ❖ Der ganze Turm steht:

```
If is_top_of(x,z) and  
height_and_width_of(z,h,w) and h > w  
and is_rest_of(x,y) and standing(y)  
then standing(x)
```

```
If empty(x) then standing(x)
```

- ❖ Rekursive Definition!



# Induktive logische Programmierung

- ❖ Rekursive Definition kann als logisches Programm aufgefasst werden
- ❖ Techniken zum Lernen von logischen Programmen stammen aus dem Bereich der “induktiven logischen Programmierung” (ILP)
- ❖ Aber: rekursive Definitionen sind in der Praxis sehr schwer zu lernen
  - ❑ Außerdem: wenige praktische Probleme erfordern Rekursion
  - ❑ Ergo: viele ILP-Techniken beschränken sich auf nichtrekursive Definitionen, um das Lernen zu erleichtern.

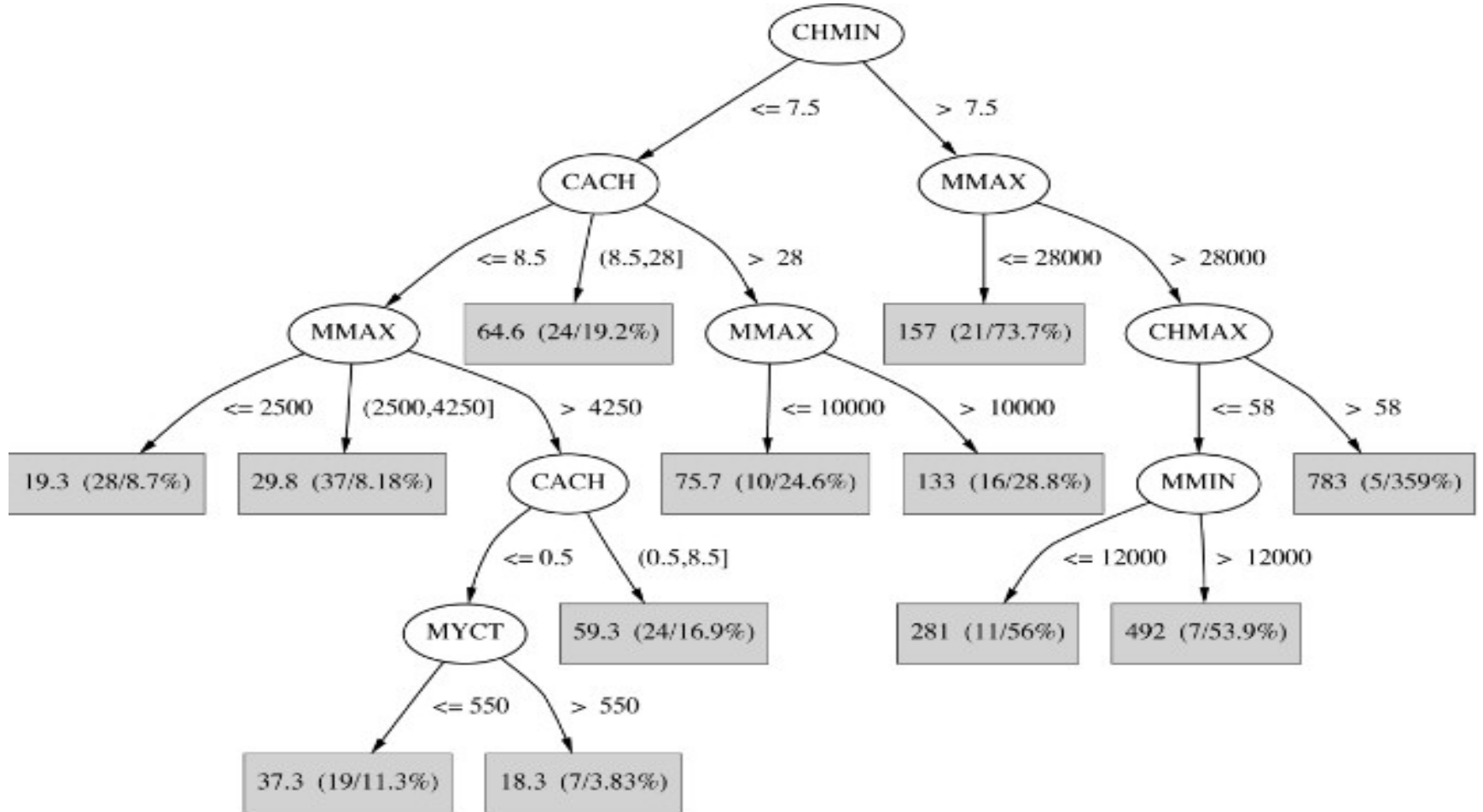
# Bäume für die numerische Vorhersage

- ❖ *Regression*: Der Prozess zur Berechnung eines Ausdrucks, der eine numerische Größe vorhersagt
- ❖ *Regressionsbaum*: „Entscheidungsbaum“, bei dem jedes Blatt eine numerische Größe vorhersagt
  - ❑ Vorhersagewert ist der Mittelwert aller Trainingsinstanzen, die dieses Blatt erreicht haben
- ❖ *Modellbaum*: „Regressionsbaum“ mit linearen Regressionsmodellen in den Blattknoten
  - ❑ lineare Stücke approximieren stetige Funktion

# Lineare Regression für die CPU-Daten

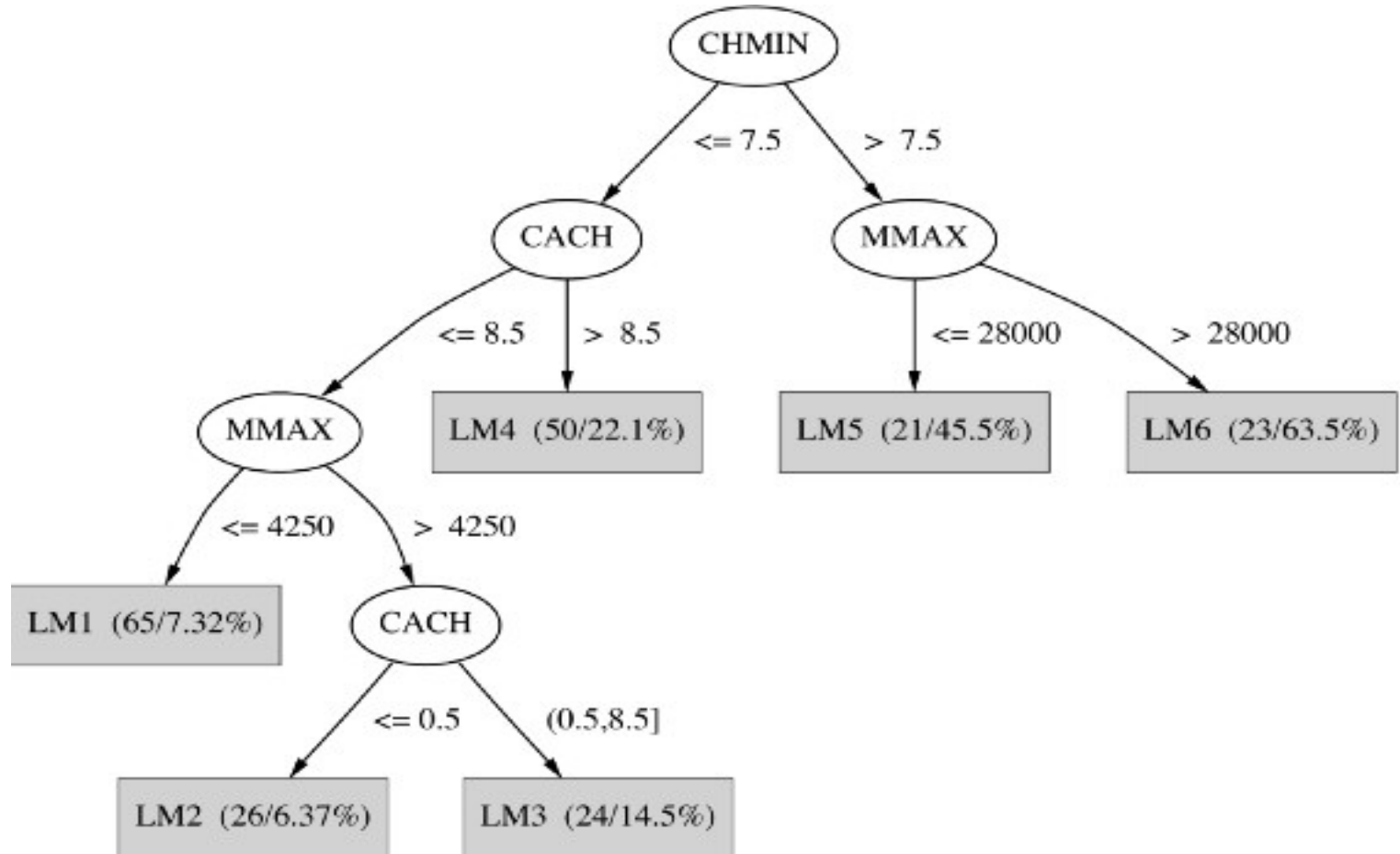
$$\begin{aligned} \text{PRP} = & \\ & - 56.1 \\ & + 0.049 \text{ MYCT} \\ & + 0.015 \text{ MMIN} \\ & + 0.006 \text{ MMAX} \\ & + 0.630 \text{ CACH} \\ & - 0.270 \text{ CHMIN} \\ & + 1.46 \text{ CHMAX} \end{aligned}$$

# Regressionsbaum für die CPU-Daten



Vorhersage (#Instanzen / Reduktion des quadr. Fehlers)

# Modellbaum für die CPU-Daten



Vorhersage (#Instanzen / Reduktion des quadr. Fehlers)

# Instanzbasierte Repräsentation

- ❖ Einfachste Form des Lernens:  
Auswendiglernen
  - ❑ In den Trainingsinstanzen wird nach demjenigen Beispiel gesucht, das am ähnlichsten zur neuen Instanz ist
  - ❑ Die Instanzen selbst repräsentieren das Wissen
  - ❑ Auch als *Instanz-basiertes Lernen* bezeichnet
- ❖ Die Ähnlichkeitsfunktion bestimmt das „Gelernte“
- ❖ Instanzbasiertes Lernen ist „faules“ (lazy) Lernen
- ❖ Methoden: nächster Nachbar, k-nächste Nachbarn

# Die Abstandsmetrik

- ❖ Einfachster Fall: nur ein numerisches Attribut
  - Distanz = Differenz der beiden beteiligten Attributwerte (oder eine Funktion hiervon)
- ❖ Mehrere numerische Attribute: meist Euklidische Distanz (wobei die Attribute normalisiert werden)
- ❖ Nominale Attribute: Distanz=0, falls Attributwerte gleich, sonst 1
- ❖ Alle Attribute gleich wichtig?
  - Evtl. Gewichtung der Attribute notwendig

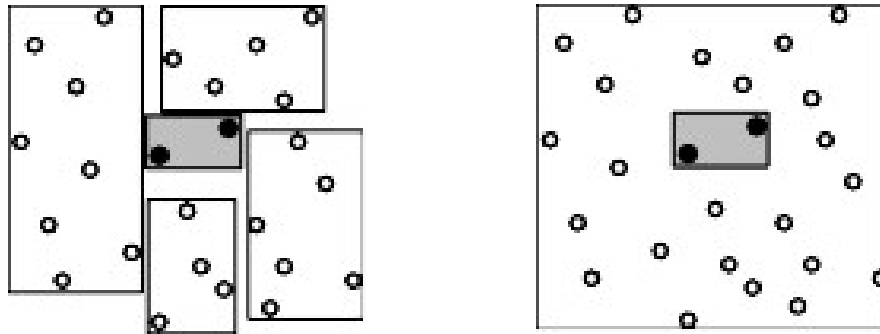
# Lernen von Prototypen



- ❖ Nur die an der Entscheidung beteiligten Instanzen müssen gespeichert werden
- ❖ Verrauschte Instanzen sollten ausgefiltert werden
- ❖ Idee: Benutze nur *prototypische* Beispiele



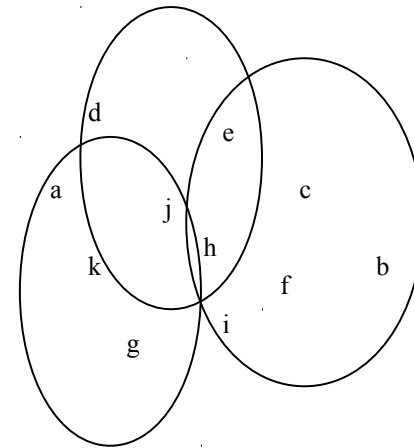
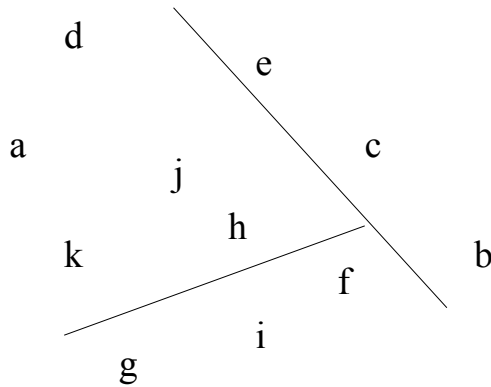
# Rechteckige Generalisierungen



- ❖ Nächster-Nachbar-Regel wird außerhalb der Rechtecke benutzt
- ❖ Rechtecke sind Regeln! (Aber sie können konservativer sein als „normale“ Regeln)
- ❖ Geschachtelte Rechtecke sind Regeln mit Ausnahmen

# Repräsentation von Clustern

Einfache 2-D Repräsentation  
Venn-Diagramm

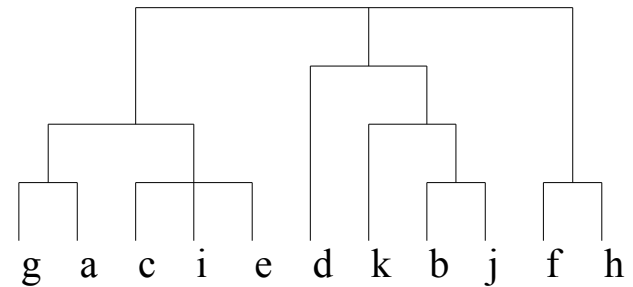


Überlappende  
Cluster

# Repräsentation von Clustern II

Probabilistische Zuordnung  
Dendrogramm

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
...			



Dendron ist das  
griechische  
Wort für Baum