Techniken des Maschinellen Lernens für Data Mining

Ian Witten, Eibe Frank

(übersetzt von Norbert Fuhr)

2 Eingabe: Konzepte, Instanzen, Attribute

- Terminologie
- Was ist ein Konzept?
  - Klassifikation, Assoziation, Clustering, nummerische Vorhersage
- Woraus besteht ein Beispiel?
  - Relationen, flache Dateien, Rekursion
- Was steckt in einem Attribut?
  - Nominal-, Ordinal, Intervall-, Verhältnis-Skalen
- Vorbereitung der Eingabe
  - ARFF, Attribute, fehlende Werte, die Daten kennenlernen

Vorbereitung zum Lernen

- Komponenten der Eingabe:
  - Konzepte: Arten von Dingen, die gelernt werden können
    - Ziel: verständliche und operationale Konzeptbeschreibung
  - Instanzen: individuelle, unabhängige Beispiele eines Konzepts
    - Anmerkung: komplexere Formen der Eingabe sind möglich
  - Attribute: Messwerte für Aspekte einer Instanz
    - Hier: Beschränkung auf nominale und ordinale Werte
- Praktisches Problem: Dateiformat für die Eingabe

Was ist ein Konzept?

- Arten von Lernen:
  - Lernen von Klassifikationen:
    Vorhersage einer diskreten Klasse
  - Lernen von Assoziationen:
    Entdecken von Assoziationen zwischen Merkmalen
  - Clustering:
    Gruppieren ähnlicher Instanzen in Clustern
  - Nummerische Vorhersage:
    Vorhersage einer nummerischen Größe
- Konzept: Das zu Lernende
- Konzeptbeschreibung:
  Ausgabe des Lernverfahrens

Lernen von Klassifikationen

- Beispielprobleme: Wetterdaten, Kontaktlinsen, Irisblumen, Tarifverhandlungen
- Klassifikationslernen ist *überwachtes* Lernen
  - Klassifikationsschema vorgegeben durch die Trainingsbeispiele
- Ergebnis ist jeweils die *Klasse* einer Instanz
- Qualität kann gemessen werden an neuen Daten, für die die Klassenzuordnungen bekannt sind (*Testdaten*)
- In der Praxis wird die Qualität oft subjektiv bewertet

Lernen von Assoziationen

- Anwendung, wenn keine Klasse spezifiziert wurde oder jegliche Struktur als "interessant" angesehen wird
- Unterschiede zum Klassifikationslernen:
  - Kann die Werte beliebiger Attribute vorhersagen, und mehr als einen Attributwert auf einmal
  - Daraus folgt: weit mehr Assoziationsregeln als Klassifikationsregeln
  - Ergo: Einschränkungen notwendig:
    - minimale Abdeckung und minimale Präzision

Clustering

- Finde Gruppen von ähnlichen Instanzen
- Clustering ist *unüberwachtes* Lernen
  - Klasse eines Beispiels ist nicht bekannt
- Qualität des Clustering wird oft subjektiv beurteilt
- Beispiel: Iris-Daten ohne Klassen:

	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
…					

Nummerische Vorhersage

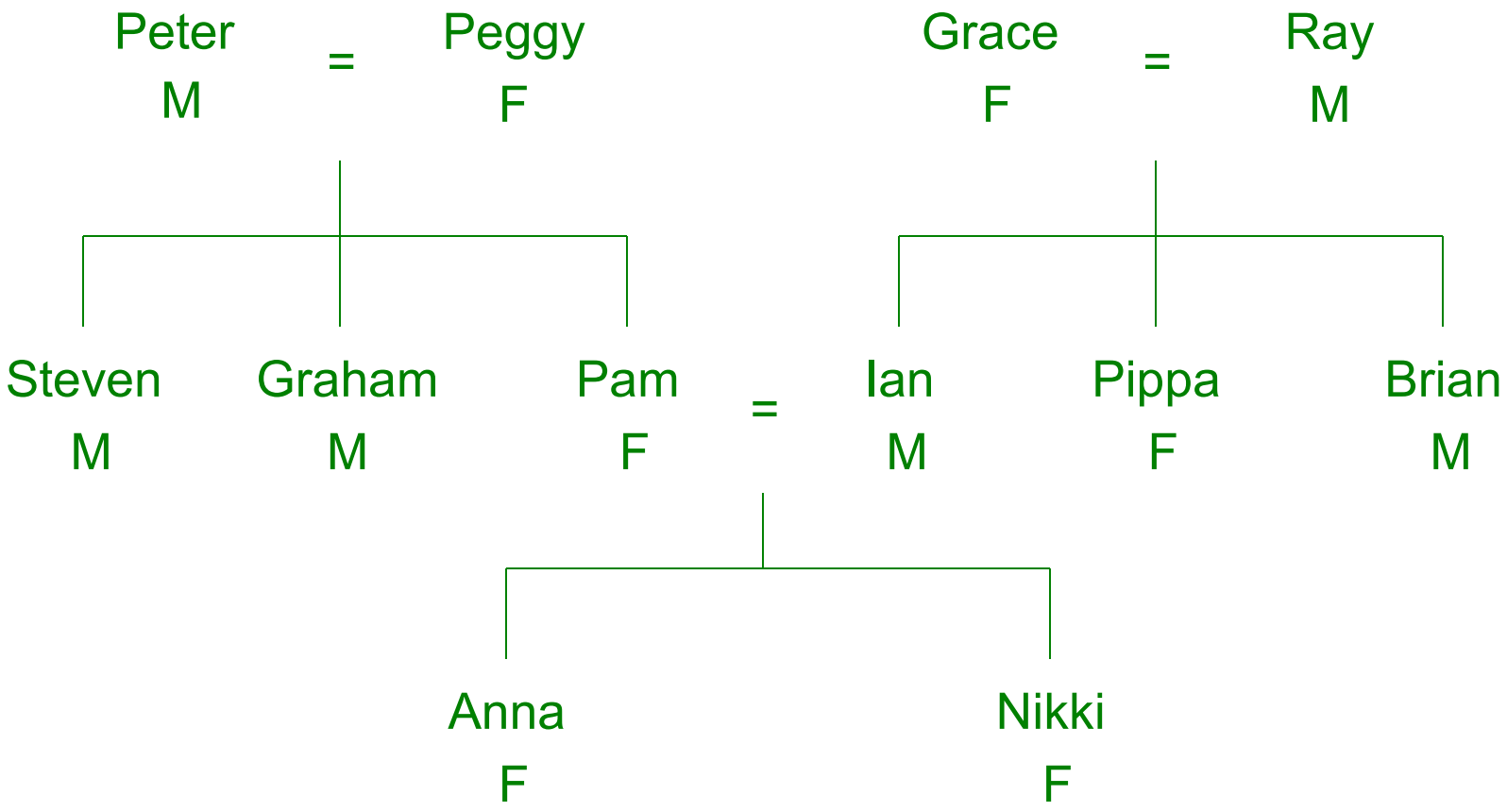
- Wie Klassifikationslernen, aber mit nummerischer "Klasse"
- Lernen ist *überwacht*
  - Schema ist durch die nummerischen Zielwerte gegeben
- Qualität wird auf den Testdaten gemessen (oder subjektiv, falls die Konzeptbeschreibung verständlich ist)
- Beispiel: modifizierte Version der Wetterdaten:

Outlook	Temperature	Humidity	Windy	**Play-time**
Sunny	Hot	High	False	**5**
Sunny	Hot	High	True	**0**
Overcast	Hot	High	False	**55**
Rainy	Mild	Normal	False	**40**
...	...	...	...	...

Woraus besteht ein Beispiel?

- Instanz: spezifischer Typ von Beispiel
  - Objekt, das klassifiziert, assoziiert oder geclustert werden soll
  - Individuelles, unabhängiges Beispiel für das Zielkonzept
  - Charakterisiert durch eine vorgegebene Menge von Merkmalen
- Eingabe für das Lernverfahren: Menge von Instanzen/Datensatz
  - Repräsentiert als einzelne Relation/flache Datei
- Stark eingeschränkte Form der Eingabe
  - Keine Beziehungen zwischen den Objekten
- Am meisten verbreitete Form des Data Mining

Ein Stammbaum



Stammbaum repräsentiert als Tabelle

Name	Gender	Parent1	parent2
Peter	Male	?	?
Peggy	Female	?	?
Steven	Male	Peter	Peggy
Graham	Male	Peter	Peggy
Pam	Female	Peter	Peggy
Ian	Male	Grace	Ray
Pippa	Female	Grace	Ray
Brian	Male	Grace	Ray
Anna	Female	Pam	Ian
Nikki	Female	Pam	Ian

Die “Schwester-von”-Relation

First person	Second person	Sister of?
Peter	Peggy	No
Peter	Steven	No
…	…	…
Steven	Peter	No
Steven	Graham	No
Steven	Pam	Yes
…	…	…
Ian	Pippa	Yes
…	…	…
Anna	Nikki	Yes
…	…	…
Nikki	Anna	yes

First person	Second person	Sister of?
Steven	Pam	Yes
Graham	Pam	Yes
Ian	Pippa	Yes
Brian	Pippa	Yes
Anna	Nikki	Yes
Nikki	Anna	Yes
All the rest		No

geschlossene-Welt Annahme

Darstellung in einer Tabelle

First person				Second person				Sister of?
Name	Gender	Parent1	Parent2	Name	Gender	Parent1	Parent2	
Steven	Male	Peter	Peggy	Pam	Female	Peter	Peggy	Yes
Graham	Male	Peter	Peggy	Pam	Female	Peter	Peggy	Yes
Ian	Male	Grace	Ray	Pippa	Female	Grace	Ray	Yes
Brian	Male	Grace	Ray	Pippa	Female	Grace	Ray	Yes
Anna	Female	Pam	Ian	Nikki	Female	Pam	Ian	Yes
Nikki	Female	Pam	Ian	Anna	Female	Pam	Ian	Yes
All the rest								No

```
If second person's gender = female
   and first person's parent = second person's parent
   then sister-of = yes
```

Generieren einer flachen Datei

- Abbildung auf eine flache Datei wird auch "Denormalisierung" genannt

  ⇨ Vorlesung Datenbanken

  - Join über mehrere Relationen, um eine einzige Relation zu generieren
- Für jede endliche Menge von endlichen Relationen möglich
- Problematisch: Beziehungen ohne vorspezifizierte Anzahl von Objekten
  - Konzept der *Kernfamilie*
- Denormalisierung kann merkwürdige Regularitäten produzieren, die die Struktur der Datenbasis wiedergeben
  - Beispiel: "supplier" bestimmt "supplier address"

Die “Vorfahr”-Relation

First person				Second person				Ancestor of?
Name	Gender	Parent1	Parent2	Name	Gender	Parent1	Parent2	
Peter	Male	?	?	Steven	Male	Peter	Peggy	Yes
Peter	Male	?	?	Pam	Female	Peter	Peggy	Yes
Peter	Male	?	?	Anna	Female	Pam	Ian	Yes
Peter	Male	?	?	Nikki	Female	Pam	Ian	Yes
Pam	Female	Peter	Peggy	Nikki	Female	Pam	Ian	Yes
Grace	Female	?	?	Ian	Male	Grace	Ray	Yes
Grace	Female	?	?	Nikki	Female	Pam	Ian	Yes
Other positive examples here								Yes
All the rest								No

Rekursion

- Unendliche Relationen erfordern Rekursion

```
If person1 is a parent of person2
   then person1 is an ancestor of person2

If person1 is a parent of person2
   and person2 is an ancestor of person3
   then person1 is an ancestor of person3
```

- Entsprechende Techniken werden als "induktive logische Programmierung" bezeichnet (z.B. Quinlan's FOIL)
  - Probleme:
    - verrauschte Daten
    - Berechnungsaufwand

Multi-Instanz-Probleme

- Jedes Beispiel besteht aus mehreren Instanzen
- Beispiel: Vorhersage der Wirksamkeit von pharmazeutischen Wirkstoffen
  - Beispiele sind Moleküle, die aktiv/inaktiv sind
  - Jedes Molekül besteht aus mehreren Gruppen (Instanzen)
  - Molekül aktiv → zumindest eine seiner Gruppen ist aktiv (positiv)
  - Molekül inaktiv → alle seiner Gruppen sind inaktiv (negativ)
- Problem: Identifikation der wirklich positiven Instanzen (Gruppen)

Attribute

- Jede Instanz wird durch eine feste, vordefinierte Menge von Merkmalen beschrieben, seinen "Attributen"
- Aber: Anzahl der Attribute kann in der Praxis schwanken
  - Mögliche Lösung: "irrelevanter Wert"-Flag (wie Nullwerte in Datenbanken)
- Verwandtes Problem: Existenz eines Attributs kann vom Wert eines anderen Attributs abhängen
- Mögliche Attributtypen ("Skalenniveau"):
  - *Nominal-, Ordinal-, Intervall-, Verhältnis-Skala*

Nominal skalierte Werte

- Werte sind unterschiedliche Symbole
  - Werte selbst dienen nur als Labels oder Namen
  - *Nominal* kommt von lateinischen Wort für Name
- Beispiel: Attribut "outlook" bei den Wetterdaten
  - Werte: "sunny","overcast", und "rainy"
- Es werden keine Beziehungen zwischen den einzelnen Werten angenommen (keine Ordnung oder Distanzen)
- Nur Tests auf Gleichheit möglich

Ordinal skalierte Werte

- Es existiert eine (lineare) Ordnung auf den Werten
- Aber: keine Distanzen zwischen den Werten definiert
- Beispiel: Attribut "temperature" bei den Wetterdaten
  - Werte: "hot" > "mild" > "cool"
- Anmerkung: Addition und Subtraktion nicht anwendbar
- Beispielregel:
  temperature < hot ⇨ play = yes
- Unterscheidung zwischen nominalen und ordinalen Werten nicht immer klar (z.B. Attribut "outlook")

Intervall-skalierte Werte

- Skalenwerte sind nicht nur geordnet, sondern die Skala ist auch in feste Einheiten gleicher Größe unterteilt
- Beispiel 1: Attribut “temperature”, gemessen in Grad Fahrenheit
- Beispiel 2: Attribut “year”
- Differenz zwischen zwei Werten stellt sinnvolle Größe dar
- Summe oder Produkt nicht sinnvoll
  - Nullpunkt nicht definiert!

Verhältnisskalen

- Bei Verhältnisskalen ist ein Nullpunkt definiert
- Beispiel: Attribut "Distanz"
  - Distanz eines Objekts zu sich selbst ist 0
- Werte einer Verhältnisskala werden als reelle Zahlen behandelt
  - Alle mathematischen Operationen sind möglich
- Aber: gibt es einen inhärenten Nullpunkt?
  - Antwort hängt von der wissenschaftlichen Erkenntnis ab (z.B. Fahrenheit kannte keine untere Schranke für die Temperatur)

Attributtypen in der Praxis

- Die meisten Verfahren berücksichtigen nur zwei Skalenniveaus: nominal und ordinal
- Nominale Attribute werden auch als "kategorisch", "diskret" oder "Aufzählungstyp" bezeichnet
  - Aber: "diskret" und "Aufzählungstyp" implizieren eine Ordnung
- Spezialfall: Dichotomie ("boolesches" Attribut)
- Ordinale Attribute werden als "nummerisch" oder "stetig" bezeichnet
  - Aber: "stetig" impliziert mathematische Stetigkeit

Transformation ordinal → boolesch

- Einfache Transformation erlaubt die Darstellung eines ordinalen Attributs mit

  n Werten durch *n-1* boolesche Attribute

- Beispiel: Attribut "temperature"

Original data

Temperature
Cold
Medium
Hot

⇨

Transformed data

Temperature > cold	Temperature > medium
False	False
True	False
True	True

- Besser als Codierung durch nominales Attribut

Metadaten

- Information über die Daten, die Hintergrundwissen darstellt
- Kann ausgenutzt werden, um den Suchraum einzuschränken
- Beispiele:
  - Berücksichtigung der Dimension

    (z.B. Ausdrücke müssen korrekt bzgl. der Dimensionen sein)
    - (z.B. Vergleich von Länge und Temperatur sinnlos)
  - Zirkulare Ordnungen

    (z.B. Gradeinteilung beim Kompass)
  - Partielle Ordnungen

    (z.B. Generalisierungen/Spezialisierungen)

Aufbereitung der Eingabe

- Denormalisierung ist nicht das einzige Problem
- Problem: verschiedene Datenquellen (z.B. Vertriebsabteilung, Rechnungsabteilung, ...)
  - Unterschiede: Arten der Datenverwaltung. Konventionen, Zeitabstände, Datenaggregation, Primärschlüssel, Fehler
  - Daten müssen gesammelt, integriert und bereinigt werden
  - "Data warehouse": konsistenter, integrierter Datenbestand
- Zusätzlich können externe Daten benötigt werden ("overlay data")
- Kritisch: Typ und Ebene der Datenaggregation

Das ARFF-Format

```
%
% ARFF file for weather data with some numeric features
%
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {true, false}
@attribute play? {yes, no}

@data
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
...
```

Attributtypen

- ARFF unterstützt nummerische und ordinale Werte
- Interpretation hängt vom Lernverfahren ab
  - nummerische Attribute werden interpretiert als
    - ordinale Werte, falls Vergleiche < und > benutzt werden
    - verhältnisskalierte Werte, falls Distanzberechnungen benutzt werden (evtl. müssen die Werte vorher normalisiert/standardisiert werden)
  - Instanz-basierte Verfahren definieren Distanzen zwischen nominalen Werten

    (0 bei Gleichheit, 1 bei Ungleichheit)
- Integer: nominal, ordinal, oder Verhältnisskala?

Nominal vs. ordinal

- Attribut "age" nominal

```
If age = young and astigmatic = no
    and tear production rate = normal
    then recommendation = soft
If age = pre-presbyopic and astigmatic = no
    and tear production rate = normal
    then recommendation = soft
```

- Attribut "age" ordinal (z.B. "young" < "pre-presbyopic" < "presbyopic")

```
If age ≤ pre-presbyopic and astigmatic = no
    and tear production rate = normal
    then recommendation = soft
```

Fehlende Werte

- Häufig dargestellt als Wert außerhalb des Wertebereichs
  - Arten von fehlenden Werten: unbekannt, nicht erfasst, irrelevant
  - Gründe:
    - Erfassungsfehler,
    - Änderungen in der Versuchsanordnung,
    - Vereinigung von Datenmengen,
    - Messung nicht möglich
- fehlender Wert an sich kann spezielle Bedeutung haben (z.B. fehlende Erhebung bei medizinischer Untersuchung)
  - Die meisten Verfahren berücksichtigen dies nicht
    ⇨ "fehlt" sollte als spezieller Wert codiert werden

Ungenaue Werte

- Grund: Daten wurden nicht für Data Mining gesammelt
- Ergebnis: Fehler und fehlende Werte, die den ursprünglichen Zweck nicht beeinflussen (z.B. Alter des Kunden)
- Tippfehler bei nominalen Attributen
  → Werte müssen auf Konsistenz geprüft werden
- Tipp- und Messfehler bei nummerischen Attributen
  → Ausreißer müssen identifiziert werden
- Fehler können willkürlich sein (z.B. falsche Postleitzahl)
- andere Probleme: Duplikate, veraltete Daten

Die Daten kennen lernen

- Einfache Visualisierungswerkzeuge sehr nützlich, um Probleme zu identifizieren
  - Nominale Attribute: Histogramme (Verteilung konsistent mit dem Hintergrundwissen?)
  - Nummerische Attribute: Verteilungskurven (Offensichtliche Ausreißer?)
- 2-D und 3-D-Visualisierungen zeigen Abhängigkeiten
- Anwendungsexperten sollten hinzugezogen werden
- Datenbestand zu umfangreich? Betrachte Stichprobe!