Modellierung

Entscheidungsbäume, Boosting, Metalerner, Random Forest

Wolfgang Konen
Fachhochschule Köln
Oktober 2007

Inhalt

- Typen der Modellierung
- Wissensrepräsentation
  - Entscheidungstabellen
  - Entscheidungsbäume
  - Regressionsbäume
- Metalerner
  - Das Trainings-Testmengen-Problem
  - Bootstrap
  - Bagging
  - Random Forests

Inhalt

- Typen der Modellierung
- Wissensrepräsentation
  - Entscheidungstabellen
  - Entscheidungsbäume
  - Regressionsbäume
- Metalerner
  - Das Trainings-Testmengen-Problem
  - Bootstrap
  - Bagging
  - Random Forests

Aktivierung: Aufgaben im Data Mining

- Welche Aufgaben im Data Mining kennen Sie?
- Also: Für welche Zielsetzungen baut DM Modelle?

Taxonomie von Aufgaben im Data Mining

- Klassifikation
  - Lernen, die Datensätze aufgrund ihrer Attribute vorgegebenen Klassen zuzuordnen
  - Beispiel: Klassifikation von Bankkunden ob kreditwürdig oder nicht
- Regression
  - Modellierung einer kontinuierlichen Outputgröße, z.B. zeitliche Veränderungen
  - Beispiele: Zeitreihenanalyse, Aktienkurse, Vorhersage von Systemausfällen auf Basis von Messdaten
- Clustering
  - Aufteilung einer Datenmenge in disjunkte Gruppen ähnlicher Objekte
  - Beispiel: Welche Pflanzengruppen haben ähnliche Standortbewertung?

Inhalt

- Typen der Modellierung
- Wissensrepräsentation
  - Entscheidungstabellen
  - Entscheidungsbäume
  - Regressionsbäume
- Metalerner
  - Das Trainings-Testmengen-Problem
  - Bootstrap
  - Bagging
  - Random Forests

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 mehrere Attribute
- Blätter weisen den Instanzen Klassen, Mengen von Klassen oder Wahrscheinlichkeitsverteilungen zu
- Unbekannte Instanz durchläuft den Baum von der Wurzel bis zu einem Blatt

Nominale und numerische Attribute

- Nominales Attribut:

  normalerweise: Anzahl der Kinder = Anzahl der möglichen Werte ⇒ Jedes Attribut wird höchstens einmal getestet

  - Andere Möglichkeit: Aufteilung in zwei Teilmengen

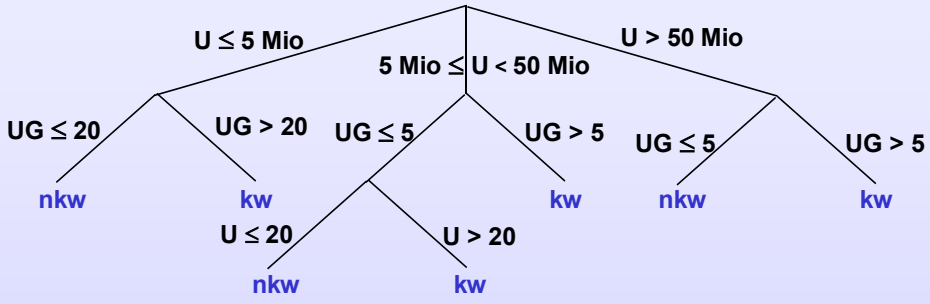
- Numerisches Attribut:

  Test, ob Attributwert größer gleich einer Konstanten ist, Attribut kann mehrfach getestet werden

  - Andere Möglichkeit: Drei-Wege-Split (oder Mehr-Wege-Split)
    - Integer: *kleiner, gleich, größer*
    - Reell: *unterhalb, innerhalb, oberhalb*

Klassifikation: Entscheidungsbaum

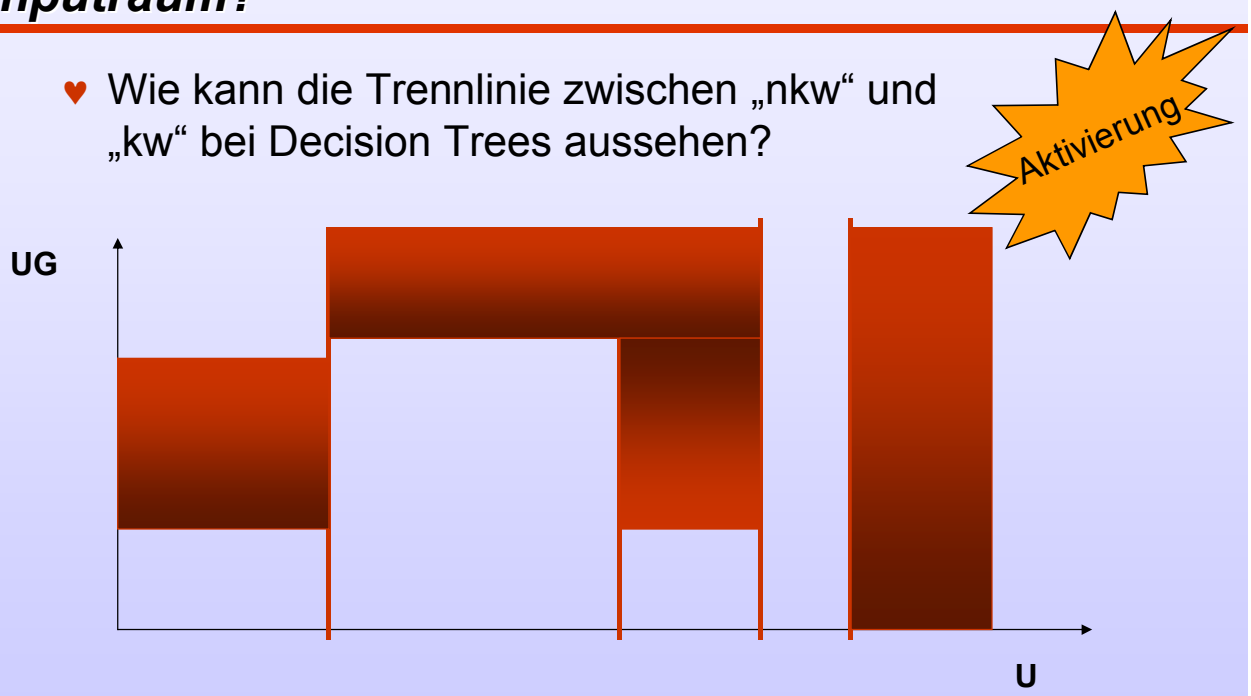
Beispiel: Kreditbeurteilung



- Ein Entscheidungsbaum ist eine graphische Darstellung einer Menge von Regeln für ein Klassifikationsproblem:
  - die Knoten im Baum entsprechen Entscheidungen
  - an den Wurzeln sind Klassen
- Beispiel: **U ≤ 5 Mio und UG ≤ 20 ⇒ nkw**
  - "Wenn der Umsatz kleiner-gleich 5 Mio DM ist und der Gewinn höchstens 20% des Umsatzes beträgt, dann ist die Firma nicht kreditwürdig"

Was macht der Entscheidungsbaum im Inputraum?

- Wie kann die Trennlinie zwischen „nkw“ und „kw“ bei Decision Trees aussehen?



⇒ Parzellierung in senkrechte Rechtecke

Aufbau von Entscheidungsbäumen

- Die Kunst beim Aufbau von Decision Trees: Wie kann man mit möglichst wenig Schnitten möglichst gute Klassifikation erreichen?
- Es werden verschiedene Split-Kriterien benutzt
  - Gain-Ratio,
  - Informationsentropie
- Darauf gehen wir in dieser Einführung nicht weiter ein. Wir benutzen Decision Trees eher als Black Box.

Regression: Bäume für die numerische Vorhersage

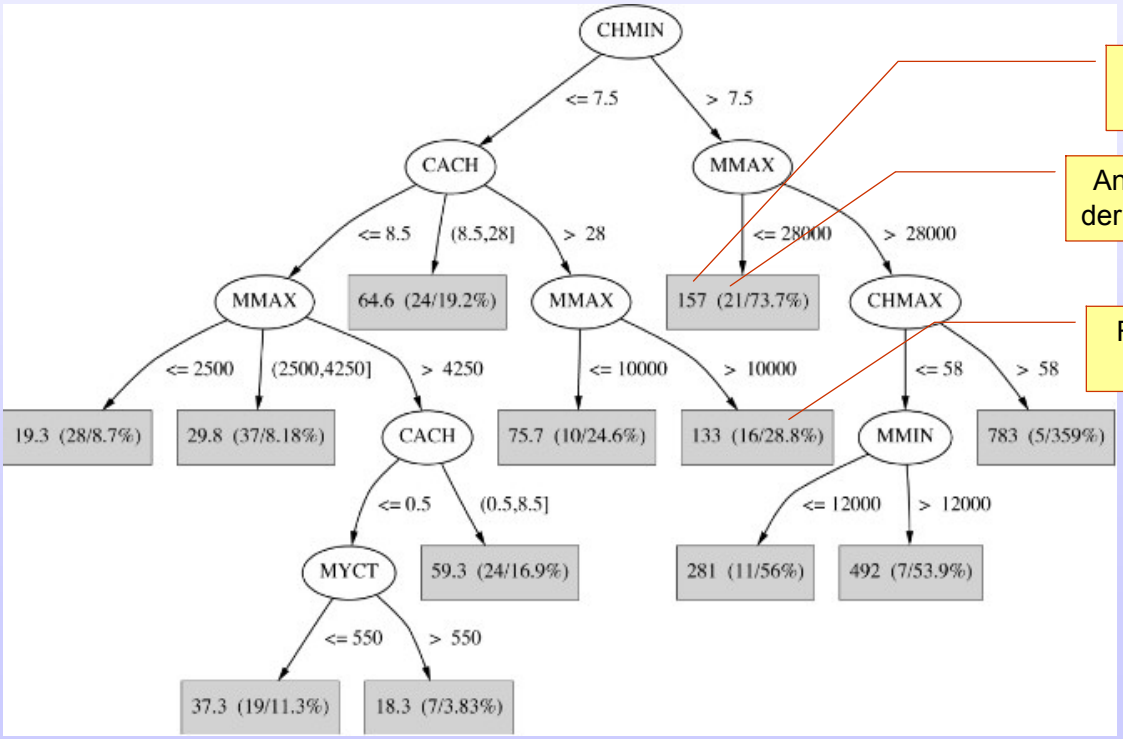
- *Regression*: Der Prozess zur Berechnung eines Ausdrucks, der eine numerische Größe vorhersagt
- *Regressionsbaum*: „Entscheidungsbaum“, bei dem jedes Blatt eine numerische Größe vorhersagt
  - Vorhersagewert ist der Mittelwert aller Trainingsinstanzen, die dieses Blatt erreicht haben
- *Modellbaum*: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

Lineare Regression für die CPU-Daten

PRP =
- 56.1
+ 0.049 MYCT
+ 0.015 MMIN
+ 0.006 MMAX
+ 0.630 CACH
- 0.270 CHMIN
+ 1.46 CHMAX

(kein Baum, sondern nur eine lineare Funktion)

Regressionsbaum für die CPU-Daten

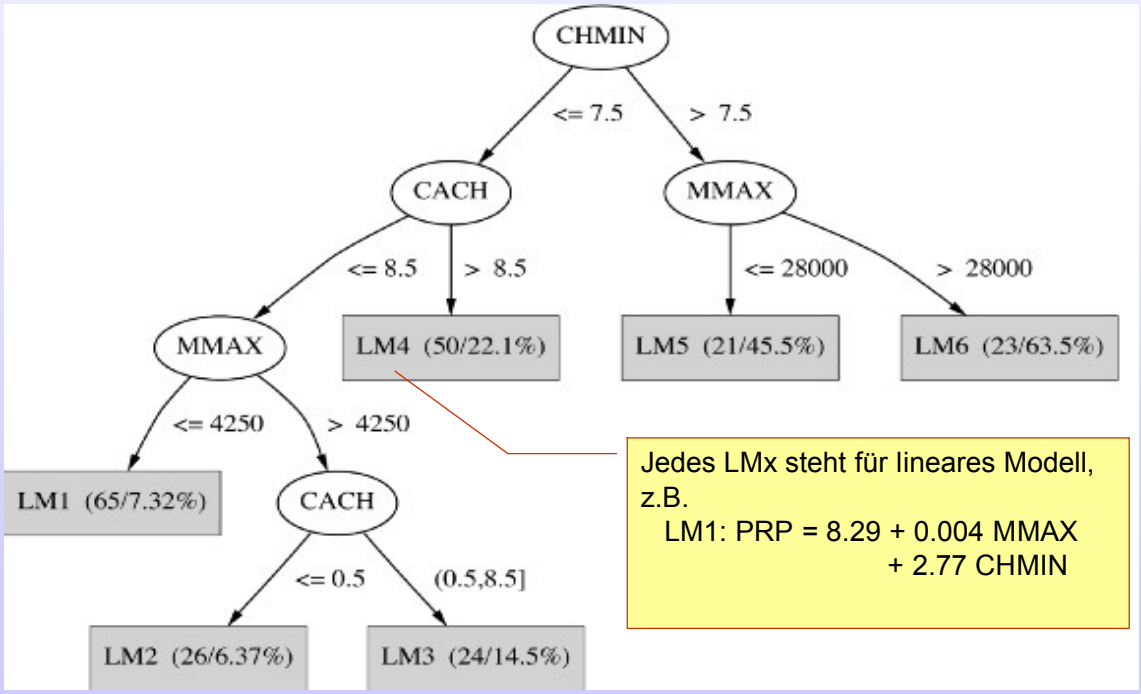


Mittel-wert

Anzahl der Fälle

Fehler in %

Modellbaum für die CPU-Daten



Jedes LMx steht für lineares Modell, z.B.

LM1: PRP = 8.29 + 0.004 MMAX + 2.77 CHMIN

Inhalt

- Typen der Modellierung
- Wissensrepräsentation
  - Entscheidungstabellen
  - Entscheidungsbäume
  - Regressionsbäume
- Metalerner
  - Das Trainings-Testmengen-Problem
  - Bootstrap
  - Bagging
  - Random Forests

Das Training-Testmengen-Problem

- Beispiel: Sie nehmen am Data-Mining-Cup (DMC) teil
  - DMC-Trainingsdaten frei verfügbar
  - DMC-Testdaten unbekannt! Ziel: Ein Modell bauen, das möglichst gut auf den DMC-Testdaten ist.
- Welche Probleme haben Sie beim Modellbau?
  -

Das Training-Testmengen-Problem

- Problem 1: Trainingsfehler zu optimistisch
  - Grund: Auswendiglernen, Überanpassung (Overtraining)
  - Effekt: zu „langes“ Lernen auf den Trainingsdaten, eher „Aufhören“ (Stopping, Pruning) wäre besser gewesen
- Abhilfe: Daten aufteilen in Trainings- u. Testmenge. Die Testmenge simuliert die DMC-Testdaten.
- Sobald der Fehler auf der Testmenge steigt, hören wir auf
- Problem 2: Welche und wieviele Daten in Test-Set?
  - zu wenig: Testmengen-Fehler nicht signifikant, streut stark
  - zu viele: das Modell hat zuwenig Daten zum Trainineren, es fehlen evtl. wichtige Datensätze
- Verschiedene Lösungen (Leave-One-Out, Kreuzvalidierung u.a.m) werden wir noch in Kap. „Evaluierung“ kennenlernen, hier eine Methode vorweg:

Die Bootstrap-Methode

- *Bootstrap = Stiefelschlaufe*
  - sinngemäß: an eigener Stiefelschlaufe aus dem Sumpf ziehen (Münchhausen)
  - Modellierung: Wie kann ich den Fehler auf unbekannter Testmenge schätzen, ohne einen Trainingsdatensatz auszulassen?

- *Bootstrap* zieht Stichproben *mit Ersetzen,* um die Trainingsmenge(n) zu bilden
  - Ziehe *n*-mal *mit Ersetzung* aus einer Datenmenge mit *n* Instanzen, um eine Stichprobe mit *n* Instanzen zu bilden
  - Benutze diese Daten als Trainingsmenge
  - Die Instanzen aus der ursprünglichen Datenmenge, die nicht in der Trainingsmenge vorkommen, werden als Testmenge verwendet

Der 0.632-Bootstrap

- Verfahren wird auch *0.632-Bootstrap* genannt
  - Die Wahrscheinlichkeit, dass eine bestimmte Instanz beim einmaligen Ziehen *nicht* ausgewählt wird, ist 1–1/*n*
  - Daraus ergibt sich die Wahrscheinlichkeit, dass die Instanz in den Testdaten landet:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0.368 = 1-0.632$$

  - Somit wird die Trainingsmenge ungefähr 63.2% aller Instanzen enthalten

Bagging, Random Forests

- **Bagg**ing = **B**ootstrap **Agg**regation
- Metalerner: kombiniere N Modelle zu einem besseren Gesamtmodell:
  - Ziehe aus Trainingsdaten N Bootstrap-Samples
  - Trainiere mit jedem Sample ein Modell
  - Vorhersage auf neuen Daten: Jedes Modell sagt getrennt vorher, Gesamtergebnis = Mehrheitsentscheid (Klassifikation) bzw. Mittelwert (Regression)
- **Random Forests** (**RF**): Bagging mit Entscheidungsbäumen als Modellen.
- Wie bekommt man mit RFs eine gute Schätzung des Testset-Fehlers? – Ihre Aufgabe, das herauszufinden!

Leo Breiman – der Mann hinter Random Forest

- Breiman, der Erfinder von CART, starb 2005. Aus seinem Nachruf:
  - *„Some of Breiman's best work was done after retirement. In particular, he developed one of the most successful state-of-the-art classification programs, 'Random Forest.' This method was based on a series of new ideas that he developed in papers during the last seven years, and it is extensively used in government and industry.„*

Prof. P. Bickel at www.berkeley.edu/news/media/releases/2005/07/07_breiman.shtml

Fazit: Wieso Metalerner?

- Kein Verfahren für sich alleine ist perfekt.
- Erst die Kombination mehrerer Lerner (auf verschiedenen Datensets) bringt die beste Leistung.
- „Committee of Experts“ ist ein anderer Name.

- Es ist meine feste Überzeugung, dass man nur mit solchen oder anderen Metalernern in die vorderen Ränge beim DM-Cup vorstoßen kann.
- Bagging ist eine generelle Meta-Lernmethode, die das Trainings-Testmengen-Problem löst.