

Data Warehousing

Kapitel 6: Data-Mining-Verfahren

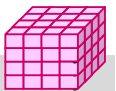
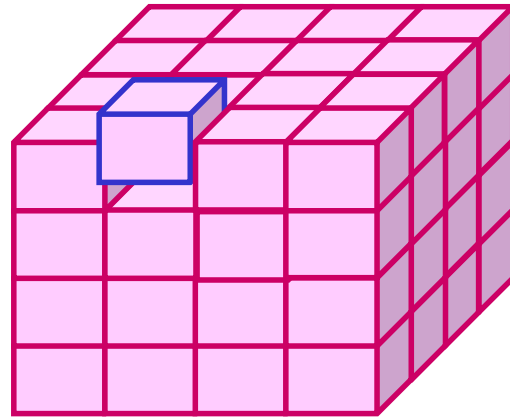
Dr. Andreas Thor

Wintersemester 2009/10

Universität Leipzig

Institut für Informatik

<http://dbs.uni-leipzig.de>



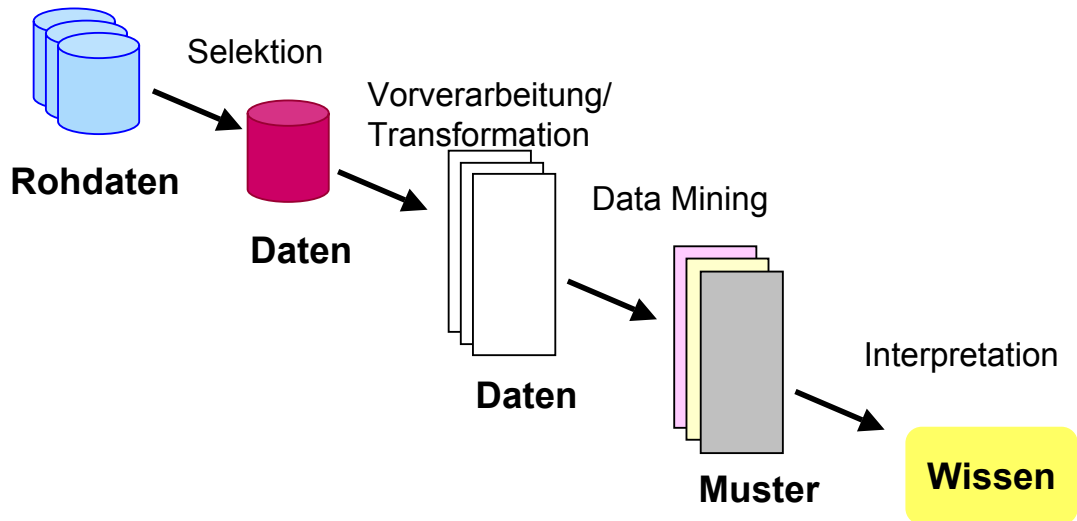
6. Überblick zu Data Mining-Verfahren

- Einführung
- Clusteranalyse
 - k-Means-Algorithmus
 - Canopy Clustering
- Klassifikation
 - Klassifikationsprozess
 - Konstruktion eines Entscheidungsbaums
- Assoziationsregeln / Warenkorbanalyse
 - Support und Konfidenz
 - A Priori-Algorithmus



Knowledge Discovery in Databases (KDD)

- (semi-)automatische Extraktion von Wissen aus Datenbanken, das
 - gültig (im statistischen Sinn)
 - bisher unbekannt
 - und potentiell nützlich ist
- Kombination von Verfahren zu Datenbanken, Statistik und KI (maschinelles Lernen)



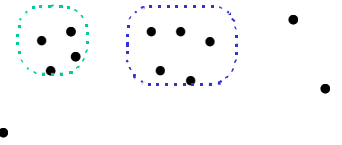
Data Mining

- Data Mining: Anwendung effizienter Algorithmen zur Erkennung von Mustern in großen Datenmengen
- bisher meist Mining auf speziell aufgebauten Dateien
- notwendig: Data Mining auf Datenbanken bzw. Data Warehouses
 - Skalierbarkeit auf große Datenmengen
 - Nutzung der DBS-Performance-Techniken (Indexstrukturen, materialisierte Sichten, Parallelverarbeitung)
 - Vermeidung von Redundanz und Inkonsistenzen
 - Integration mehrerer Datenquellen
 - Portabilität
- Datenaufbereitung für Data Mining
 - Datenintegration und Datenbereinigung (data cleaning)
 - Diskretisierung numerischer Attribute (Aufteilung von Wertebereichen in Intervalle, z.B. Altersgruppen)
 - Erzeugen abgeleiteter Attribute (z.B. Aggregationen für bestimmte Dimensionen, Umsatzänderungen)
 - Einschränkung der auszuwertenden Attribute

Data Mining: Techniken

■ Clusteranalyse

- Objekte werden aufgrund von Ähnlichkeiten in Klassen eingeteilt (Segmentierung)



■ Assoziationsregeln

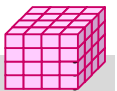
- Warenkorbanalyse (z.B. Kunde kauft A und B => Kunde kauft C)
- Sonderformen zur Berücksichtigung von Dimensionshierarchien (z.B. Produktgruppen), quantitativen Attributen, zeitlichen Beziehungen (sequence mining)

■ Klassifikation

- Zuordnung von Objekten zu Gruppen/Klassen mit gemeinsamen Eigenschaften bzw. Vorhersage von Attributwerten
- explizite Erstellung von Klassifikationsregeln (z.B. “guter Kunde” wenn Alter > 25 und ...)
- Verwendung von Stichproben (Trainingsdaten)
- Ansätze: Entscheidungsbaum-Verfahren, statistische Auswertungen (z.B. Maximum Likelihood-Schätzung / Bayes-Schätzer), neuronale Netze

■ Weitere Ansätze:

- Genetische Algorithmen (multivariate Optimierungsprobleme, z.B. Identifikation der besten Bankkunden)
- Regressionsanalyse zur Vorhersage numerischer Attribute . . .



Data Mining: Anwendungsbeispiele

■ Kundensegmentierung für Marketing

- Gruppierung von Kunden mit ähnlichem Kaufverhalten / ähnlichen Interessen
- Nutzung für gruppenspezifische Empfehlungen, Product Bundling, ...

■ Warenkorbanalyse: Produkt-Platzierung im Supermarkt, Preisoptimierung, ...

■ Bestimmung der Kreditwürdigkeit von Kunden (elektronische Vergabe von Kreditkarten, schnelle Entscheidung über Versicherungsanträge, ...)

- schnelle Entscheidung erlaubt neue Kunden zu gewinnen
- Technik: Entscheidungsbaum-Klassifikator

■ Entdeckung wechselbereiter Kunden

■ Entdeckung von Kreditkarten-Missbrauch

■ Unterstützung im Data Cleaning

■ Web Usage Mining

■ Text Mining: inhaltliche Gruppierung von Dokumenten, E-Mails, ...



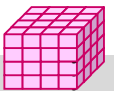
Evaluation/Interpretation

■ Ablauf

- Präsentation der gefundenen Muster, z.B. über Visualisierungen
- Bewertung der Muster durch den Benutzer
- falls schlechte Bewertung: erneutes Data Mining mit anderen Parametern, anderem Verfahren oder anderen Daten
- falls gute Bewertung: Integration des gefundenen Wissens in die Wissensbasis / Metadaten und Nutzung für zukünftige KDD-Prozesse

■ Bewertung der gefundenen Muster: Interessantheit, Vorhersagekraft

- sind Muster schon bekannt oder überraschend?
- wie gut lassen sich mit „Trainingsdaten“ (Stichproben) gefundene Muster auf zukünftige Daten verallgemeinern?
- Vorhersagekraft wächst mit Größe und Repräsentativität der Stichprobe



Clusteranalyse

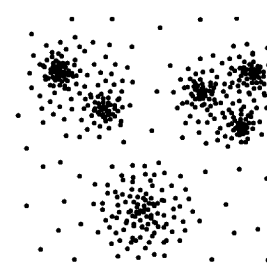
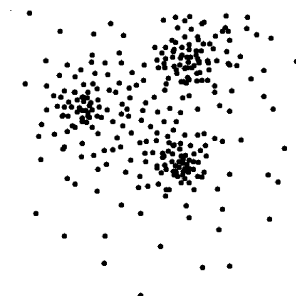
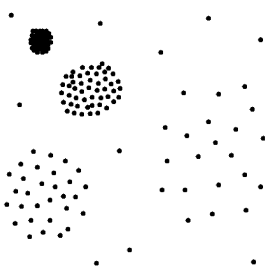
■ Ziele

- automatische Identifikation einer endlichen Menge von Kategorien, Klassen oder Gruppen (Cluster) in den Daten
- Objekte im gleichen Cluster sollen möglichst ähnlich sein
- Objekte aus verschiedenen Clustern sollen möglichst unähnlich zueinander sein

■ Ähnlichkeitsbestimmung

- meist: Distanzfunktion $dist(o_1, o_2)$ für Paare von Objekten o_1 und o_2
 - z.B. Euklidische Distanz für numerische Attribute: $dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
- spezielle Funktionen für kategoriale Attribute oder Textdokumente

■ Clustering-Ansätze: partitionierend, hierarchisch, dichtebasiert, ...



K-Means Algorithmus

■ Ausgangssituation

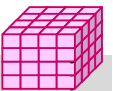
- Objekte besitzen Distanzfunktion
- für jedes Cluster kann ein Clusterzentrum bestimmt werden („Mittelwert“)
- Anzahl k der Cluster wird vorgegeben

■ Basis-Algorithmus

- Schritt 1 (Initialisierung): k Clusterzentren werden (zufällig) gewählt
- Schritt 2 (Zuordnung): Jedes Objekt wird dem nächstgelegenen Clusterzentrum zugeordnet
- Schritt 3 (Clusterzentren): Für jedes Cluster wird Clusterzentrum neu berechnet
- Schritt 4 (Wiederholung): Abbruch, wenn sich Zuordnung nicht mehr ändert, sonst zu Schritt 2

■ Probleme

- Konvergenz zu lokalem Minimum, d.h. Clustering muss nicht optimal sein
 - Work-around: Algorithmus mehrfach starten
- relativ hoher Aufwand für Abstandsberechnungen, Neuberechnung der Clusterzentren



K-Means Algorithmus: Beispiel

■ Clustering der Zahlen 1, 3, 6, 14, 17, 24, 26, 31 in drei Cluster

- (1) Zentren: 10, 21, 29 (zufällig gewählt)
- (2) Cluster:
- (3) Zentren (arithmetisches Mittel):
- (2) Cluster:
- (3) Zentren:
- (2) Cluster:
- (3) Zentren:
- (2) Cluster:
- Abbruch, da sich das Clustering nicht mehr geändert hat.



Canopy Clustering Algorithmus*

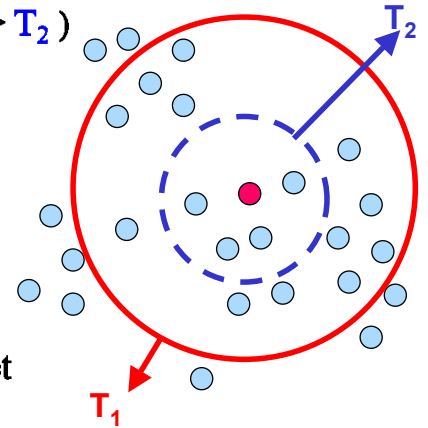
■ Bildung von überlappenden Clustern (Canopies)

- oft Nutzung als erster Schritt in mehrstufigem Analyseverfahren
- skalierbar auf sehr große Datenmengen
- auf Stringdaten anwendbar (Nutzung indexbasierter Ähnlichkeitsfunktionen zur Distanzberechnung, z.B. TF/IDF)

■ Gegeben: Distanzfunktion, zwei Schwellwerte T_1 und T_2 ($T_1 > T_2$)

■ Algorithmus

- Schritt 1 (Initialisierung): Kandidatenliste für Wahl der Canopyzentren wird mit allen Objekten initialisiert
- Schritt 2 (Canopyzentrum): Canopyzentrum Z wird (zufällig) aus der Kandidatenliste gewählt
- Schritt 3 (Zuordnung): Alle Objekte, deren Abstand zu Z geringer ist als Schwellwert T_1 , werden (Canopy) Z zugeordnet
- Schritt 4 (Kandidatenliste): Alle Objekte, die innerhalb des Schwellwertes T_2 zu Z liegen, werden aus Kandidatenliste gelöscht
- Schritt 5 (Ende/Wiederholung): Abbruch, wenn Kandidatenliste leer ist, sonst zu Schritt 2



* McCallum, A. et al.: *Efficient clustering of high-dimensional data sets with application to reference matching*
Proc. ACM KDD, 2000



Canopy Clustering Algorithmus: Beispiel

- Clustering der Zahlen 1, 3, 6, 11, 13
 - Abstandsfunktion absolute Differenz, $T_1=8$, $T_2=4$
 - (1) Kandidatenliste = {1, 3, 6, 11, 13}
 - (2) Canopyzentrum
 - (3) Canopy bilden
 - (4) Entferne aus Kandidatenliste

 - (5) Kandidatenliste =
 - (2) Canopyzentrum:
 - (3) Canopy bilden
 - (4) Entferne aus Kandidatenliste

 - (5) Kandidatenliste =
 - (2) Canopyzentrum:
 - (3) Canopy bilden
 - (4) Entferne aus Kandidatenliste

 - (5) Abbruch, da Kandidatenliste leer



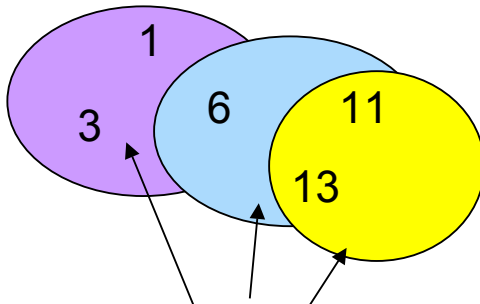
Kombination mit K-Means

■ 1. Schritt: Canopy Clustering

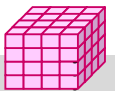
■ 2. Schritt: K-Means

- Für jedes Canopy wird mindestens ein Clusterzentrum bestimmt
- Statt für jedes Objekt die Distanz zu jedem Clusterzentrum zu bestimmen, werden nur die Distanzen zu den Clusterzentren bestimmt, die sich mit dem Objekt in denselben Canopies befinden.

■ Beispiel:



- Drei Clusterzentren: 3, 3 10 12
- Für 1 und 3 keine Abstandsbestimmung notwendig, da sich nur ein Clusterzentrum im selben Canopy befindet
- 6, 11 und 13 liegen jeweils in zwei Canopies, deshalb sind zwei Abstandsbestimmungen zu zwei Clusterzentren notwendig.



Klassifikation

■ Klassifikationsproblem

- Gegeben sei Stichprobe (Trainingsmenge) O von Objekten des Formats (a_1, \dots, a_d) mit *Attributen* A_i , $1 \leq i \leq d$, und Klassenzugehörigkeit c_i , $c_i \in C = \{c_1, \dots, c_k\}$
- Gesucht: die Klassenzugehörigkeit für Objekte aus $D \setminus O$
d.h. ein *Klassifikator* $K : D \rightarrow C$

■ weiteres Ziel:

- Generierung (Lernen) des expliziten Klassifikationswissens (**Klassifikationsmodell**, z.B. Klassifikationsregeln oder Entscheidungsbaum)

■ Abgrenzung zum Clustering

- Klassifikation: Klassen vorab bekannt
- Clustering: Klassen werden erst gesucht

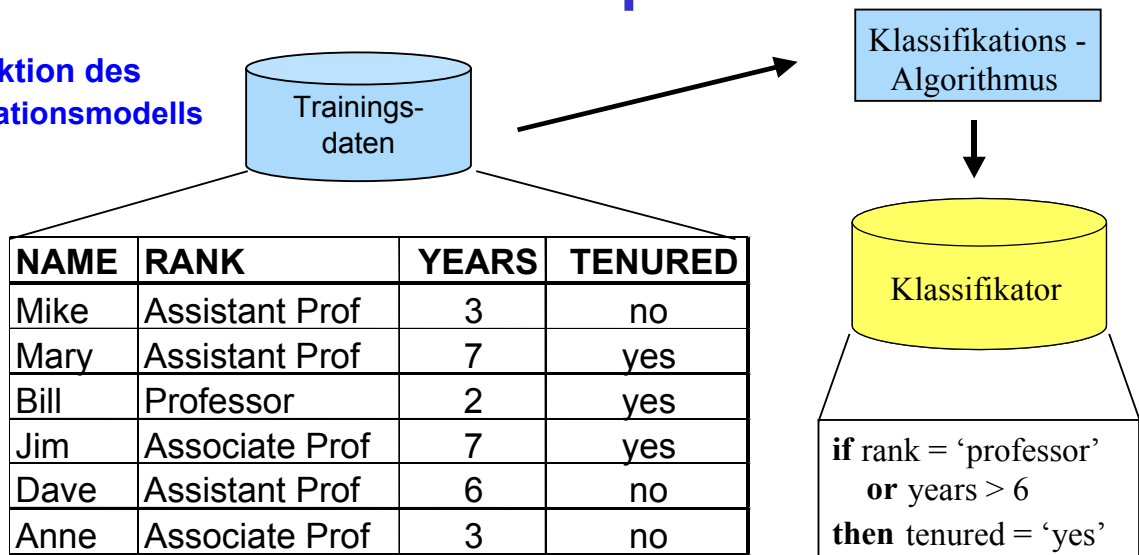
■ Klassifikationsansätze

- Entscheidungsbaum-Klassifikatoren
- Bayes-Klassifikatoren (Auswertung der bedingten Wahrscheinlichkeiten von Attributwerten)
- Neuronale Netze

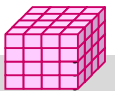
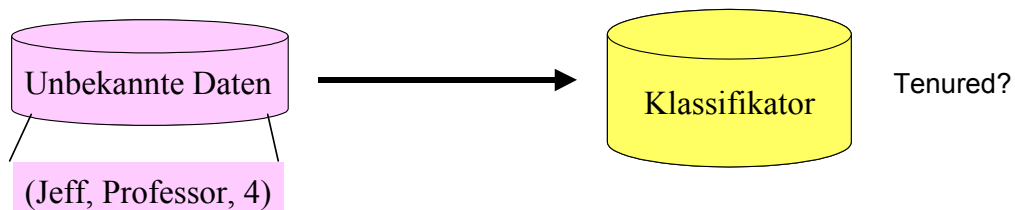


Klassifikationsprozess

1. Konstruktion des Klassifikationsmodells



2. Anwendung des Modells zur Vorhersage (Prediction)



Bewertung von Klassifikatoren

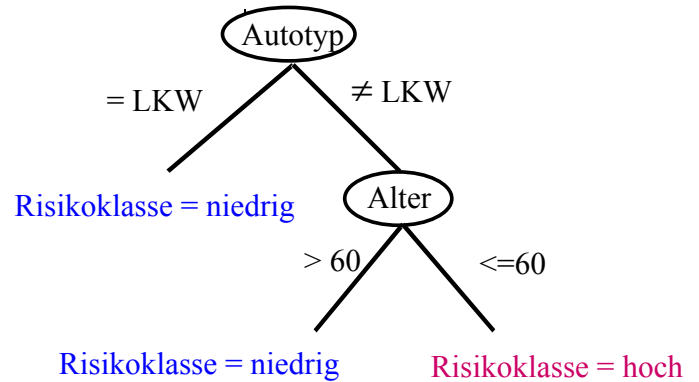
- Klassifikator ist für die Trainingsdaten optimiert
 - liefert auf der Grundgesamtheit der Daten evtl. schlechtere Ergebnisse (-> Overfitting-Problem)
- Bewertung mit von Trainingsmengen unabhängigen Testmengen
- Gütemasse für Klassifikatoren
 - Klassifikationsgenauigkeit
 - Kompaktheit des Modells, z.B. Größe eines Entscheidungsbaums
 - Interpretierbarkeit des Modells (wie viel Einsichten vermittelt das Modell dem Benutzer?)
 - Effizienz der Konstruktion des Modells sowie der Anwendung des Modells
 - Skalierbarkeit für große Datenmengen
 - Robustheit gegenüber Rauschen und fehlenden Werten
- Klassifikationsgenauigkeit: Anteil der korrekten Klassenzuordnungen in Testmenge
- Klassifikationsfehler: Anteil der falschen Klassenzuordnungen



Entscheidungsbäume

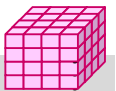
- explizite, leicht verständliche Repräsentation des Klassifikationswissens

ID	Alter	Autotyp	Risiko
1	23	Familie	hoch
2	17	Sport	hoch
3	43	Sport	hoch
4	68	Familie	niedrig
5	32	LKW	niedrig



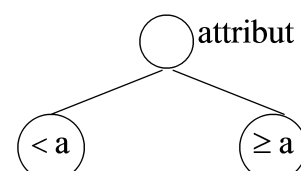
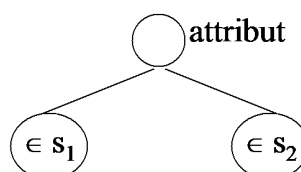
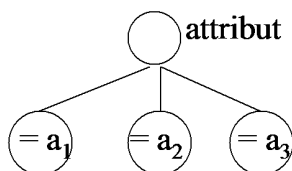
Regeldarstellung:

- Entscheidungsbaum ist Baum mit folgenden Eigenschaften:
 - ein innerer Knoten repräsentiert ein Attribut
 - eine Kante repräsentiert einen Test auf dem Attribut des Vaterknotens
 - ein Blatt repräsentiert eine der Klassen
- Anwendung zur Vorhersage:
 - Top-Down-Durchlauf des Entscheidungsbaums von der Wurzel zu einem der Blätter
 - eindeutige Zuordnung des Objekts zur Klasse des erreichten Blattes



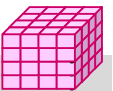
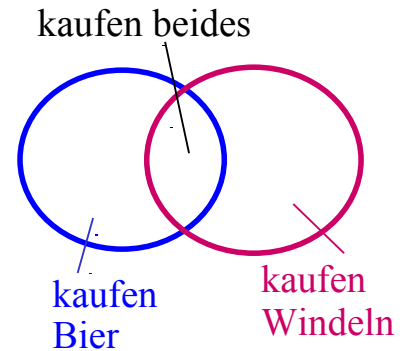
Konstruktion eines Entscheidungsbaums

- Basis-Algorithmus (divide and conquer)**
 - Anfangs gehören alle Trainingsdatensätze zur Wurzel
 - Auswahl des nächsten Attributs (Split-Strategie): Maximierung des Informationsgewinns (meßbar über Entropie o.ä.)
 - Partitionierung der Trainingsdatensätze mit Split-Attribut
 - Verfahren wird rekursiv für die Partitionen fortgesetzt
- Abbruchbedingungen**
 - keine weiteren Split-Attribute
 - alle Trainingsdatensätze eines Knotens gehören zur selben Klasse
- Typen von Splits**
 - *Kategorische Attribute*: Split-Bedingungen der Form „attribut = a“ oder „attribut ∈ set“ (viele mögliche Teilmengen)
 - *Numerische Attribute*: Split-Bedingungen der Form „attribut < a“ (viele mögliche Split-Punkte)



Assoziationsregeln

- Warenkorbanalyse auf Transaktions-Datenbank
 - Transaktion umfasst alle gemeinsam getätigten Einkäufe, innerhalb eines Dokuments vorkommenden Worte, innerhalb einer Web-Sitzung referenzierten Seiten, ...
- Regeln der Form “Rumpf → Kopf [support, confidence]”
- Beispiele
 - kauft(X, “Windeln”) → kauft(X, “Bier”) [0.5%, 60%]
 - 80% aller Kunden, die Reifen und Autozubehör kaufen, bringen ihr Auto auch zum Service
- Relevante Größen
 - **Support** einer Regel $r \rightarrow k$: Anteil der Transaktionen, in denen alle Objekte r und k vorkommen
 - **Konfidenz** einer Regel $r \rightarrow k$: Anteil der Transaktionen mit Rumpf-Objekten r , für die Regel erfüllt ist (d.h. für die auch Objekte k vorliegen)
 - **Interessantheit**: hoher Wahrscheinlichkeitsunterschied für k gegenüber zufälliger Verteilung



Assoziationsregeln (2)

- Aufgabe: Bestimmung aller Assoziationsregeln, deren Support und Konfidenz über bestimmten Grenzwerten liegen

- Beispiel 1: Transaktionen

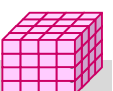
- Support
- (A):
- (B), (C):
- (D), (E), (F):
- (A, C): 50%
- (A, B), (A, D), (B, C), (B, E), (B, F), (E, F): 25%
- (A, B, C), (B, E, F): 25%
- Assoziationsregeln:

TransaktionsID	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

$minsup = 50\%$,
 $minconf = 50\%$

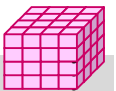
- Beispiel 2: Warenkörbe

- Drucker, Papier, PC, Toner
- PC, Scanner
- Drucker, Papier, Toner
- Drucker, PC
- Drucker, Papier, PC, Scanner, Toner



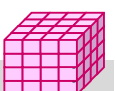
Frequent Itemsets

- Frequent Item-Set:
Item-Menge, deren Support gewisse Schranke s übersteigt
- Bestimmung der Frequent-Itemsets wesentlicher Schritt zur Bestimmung von Assoziationsregeln
- effiziente Realisierung über A-Priori-Algorithmus
- Nutzung der sog. **A-Priori-Eigenschaft**:
 - Jede Teilmenge eines Frequent Itemsets muss auch ein Frequent Itemset sein
 - Support jeder Teilmenge und damit jedes einzelnen Items muss auch über Schranke s liegen
- Effiziente, iterative Realisierung beginnend mit 1-elementigen Itemsets
 - schrittweise Auswertung für k -Itemsets Teilmengen von k Elementen ($k \geq 1$),
 - Ausklammern von Kombinationen, welche Teilmengen haben, die Support s nicht erreichen
 - wird „a priori“ getestet, bevor Support bestimmt wird



A-Priori-Algorithmus

- Bestimmung aller Frequent Itemsets mit k Elementen
 - $k := 1$
 - prüfe für alle Items, ob sie Frequent Item sind, d.h. minimalen Support s einhalten (Kandidatenmenge C_1)
 - iteriere solange bis keine neuen Frequent Itemsets hinzukommen
 - generiere aus jedem Frequent Itemset I_k mit k Items aus C_k Itemsets I_{k+1} mit $k+1$ Items, davon k aus I_k
 - prüfe jedes I_{k+1} , ob es Support s erfüllt und wenn ja, nehme es in Kandidatenmenge C_{k+1} auf
 - $k := k + 1$
- Überprüfung der I_{k+1} kann durch sequentiellen Durchgang auf der reduzierten Faktentabelle (Kandidatentabellen) realisiert werden



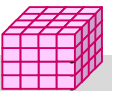
A-Priori-Algorithmus: Beispiel

- 5 PC-Warenkörbe (Forderung: minimaler Support: 60%)
 - Drucker, Papier, PC, Toner
 - PC, Scanner
 - Drucker, Papier, Toner
 - Drucker, PC
 - Drucker, Papier, PC, Scanner, Toner
- Algorithmus-Ausführung
 - k=1:

 - k=2:

 - k=3:

 - k=4:



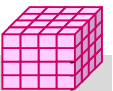
Bestimmung der Assoziationsregeln

- einfache Bestimmung der Assoziationsregeln aus Frequent Itemsets
- Berechnung der Konfidenz
 - für alle Frequent Itemsets F sei bekannt: $support(F) = \#Vorkommen \text{ von } F / \#Transaktionen$
 - Zerlegen jedes Frequent Itemsets F mit $k (>1)$ Items in 2 disjunkte Itemmengen L und R , mit $F = L \cup R$
 - dann gilt: $confidence(L \rightarrow R) = support(F) / support(L)$
 - Elimination aller Kombinationen, deren Konfidenz Minimalwert unterschreitet
- Beispiel
 - confidence ($\{Drucker\} \rightarrow \{Papier, Toner\}$) =
- Konfidenz kann erhöht werden, in dem man Items von rechts auf die linke Seite bringt
 - Beispiel: confidence ($\{Drucker, Papier\} \rightarrow \{Toner\}$) =



Assoziationsregeln: weitere Aspekte

- Nutzbarkeit u.a. für Cross-Selling, Produkt-Platzierung ...
 - Amazon: Kunden die dieses Buch gekauft haben, kauften auch ...
- Sonderfall: Sequenzanalyse (Erkennung sequentieller Muster)
 - Berücksichtigung der Zeit-Dimension
 - Bsp. 1: In 50% der Fälle, wenn Produkt A gekauft wurde, wird bei einem späteren Besuch Produkt B gekauft
 - Bsp. 2: In 40% aller Fälle, in denen ein Nutzer über einen Werbebanner auf die Web-Site gelangt und die Site vorher schon einmal besucht hat, kauft er einen Artikel; dies kommt in insgesamt 10% aller Sessions vor
- Probleme
 - sehr viele Produkte / Web-Seiten / Werbebanner / Besucher etc. erfordern Bildung größerer Bezugseinheiten
 - es können sinnlose Korrelationen ermittelt werden
 - fehlender kausaler Zusammenhang
 - z.B. Schmutzeffekte aufgrund transitiver Abhängigkeiten (Bsp.: Haarlänge korreliert negativ mit Körpergröße)



DBS-Unterstützung für Data Mining

- SQL-Standardisierung: SQL/MM
 - Standardisierte Schnittstelle (UDTs, UDFs) zur Definition von Data Mining-Modellen, Bereitstellung für Trainingsdaten und Data Mining-Anfragen
- Ähnliche Unterstützung in DB2 und MS SQL Server
- Beispiel: Klassifikation bei MS SQL-Server
 - **Repräsentation eines Mining-Modells als geschachtelte Tabellen**

```
CREATE MINING MODEL AgePrediction
  ( CustomerID    LONG KEY,
    Gender TEXT   DISCRETE ATTRIBUTE,
    HairColor    TEXT   DISCRETE ATTRIBUTE,
    Age          DOUBLE CONTINUOUS ATTRIBUTE PREDICT )
  USING [Microsoft Decision Tree]
```
 - **Bereitstellung von Trainingsdaten (bekannte „Fälle“)**

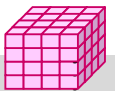
```
INSERT INTO AgePrediction (CustID, Gender, HairColor, Age)
  (SELECT CustID, Gender, HairColor, Age FROM Customers )
```
 - **Anwendung des Modells zur Analyse über Prediction Joins**

```
SELECT Customers.ID, MyDMM.Age, PredictProbability(MyDMM.Age)
  FROM AgePrediction MyDMM PREDICTION JOIN Customers ON
  MyDMM.Gender = Customers.Gender AND ...
```



Zusammenfassung

- wichtige Verfahrensklassen zum Data Mining: Clusteranalyse, Klassifikation, Assoziationsregeln
- Clusteranalyse: Segmentierung über Ähnlichkeits- bzw. Distanzmaß
- Klassifikation z.B. über Entscheidungsbäume
 - setzt Trainingsphase mit Testdaten und bekannter Klassenzuordnung voraus
 - Konstruktion des Entscheidungsbaumes mit Bestimmung der Split-Attribute
- Assoziationsregeln
 - effiziente Berechenbarkeit von Frequent Itemsets über A-Priori-Algorithmus
 - Bestimmung von Konfidenz
- zahlreiche Nutzungsmöglichkeiten: Kundensegmentierung, Vorhersage des Kundenverhaltens, Warenkorbanalyse etc.
 - keine „out of the box“-Lösungen
 - Interpretation der Ergebnisse nicht immer einfach
- zunehmende Unterstützung durch kommerzielle DBS, z.B. MS SQL Server und DB2



Übungsaufgaben

- Welche der vorgestellten Data Mining Verfahren sind für die folgenden Anwendungsgebiete einsetzbar?
 - Herausfiltern von Spam-Mails
 - Auffinden häufiger Wortfolgen in Texten
 - Bestimmung von Sterngruppen in Himmelsbildern
 - Automatische Handschriftenerkennung
 - Was ist der wesentliche Unterschied zwischen Clustering und Klassifikation?
 - k-Means Algorithmus
 - Für ein Clustering in zwei Cluster sind die Zahlen 1, 2, 6, 10, 15, 16, 17 gegeben. Bestimmen Sie mit Hilfe des k-Means Algorithmus (Abstandsfunktion ist absolute Differenz; Clusterzentren sind arithmetisches Mittel) die beiden Cluster
 - a) mit den Anfangszentren 2 und 9
 - b) mit den Anfangszentren 8 und 14
- Was stellen Sie fest?



Übungsaufgaben (2)

■ Assoziationsregeln

Eine statistische Untersuchung hat ergeben:

- 60% der Schüler spielen Fußball, 75% der Schüler essen Schokoriegel
- 40% der Schüler spielen Fußball und essen Schokoriegel

Bestimmen sie Support und Konfidenz der Assoziationsregeln

- „Spielt Fußball“ → „Isst Schokoriegel“
- True → „Isst Schokoriegel“

Beurteilen Sie die Relevanz der ersten Regel!

■ Warenkorbanalyse : Gegeben seien folgende acht Warenkörbe

Milch, Limonade, Bier
Milch, Apfelsaft, Orangensaft
Milch, Bier
Limonade, Orangensaft

Milch, Apfelsaft, Bier
Milch, Bier, Orangensaft, Apfelsaft
Limonade, Bier, Orangensaft
Bier, Apfelsaft

- Wie hoch ist der Support von Itemset {Bier, Orangensaft}?
- Wie hoch ist die Konfidenz von {Bier} → {Milch} ?
- Welche Produktpaare haben einen Support von mehr als 35% ?
- Welche Produkt-Tripel haben einen Support von mehr als 35% ?
- Welche Assoziationsregeln haben Support und Konfidenz von mind. 50%?

