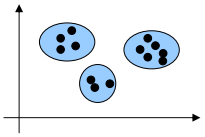
Datenbanken und Informationssysteme

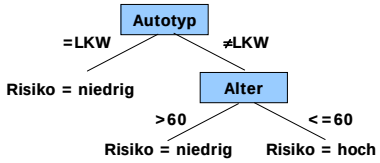
Data Mining

Data Mining

- Anwendung effizienter Algorithmen zur Erkennung von Mustern in großen Datenmengen
- Clustering
  - automatische Identifikation einer endlichen Menge von Kategorien, Klassen oder Gruppen (Cluster)
  - Vergleich mittels Distanzfunktion



- Klassifikation
  - Ziel: Erlernen eines Klassifikators (z.B. Entscheidungsbaum)
  - Klassen vorab bekannt, Trainingsdaten vorhanden



- Assoziationsregeln
  - Warenkorbanalyse auf Transaktions-Datenbank
  - Bsp.: Kauft(PC)=>Kauft(Drucker)

Assoziationsregeln

- Regeln der Form: $r \rightarrow k$ *[support, confidence]*
- Support:
  Anteil der Transaktionen, in denen alle Objekte r und k vorkommen
- Confidence:
  Anteil der Transaktionen mit Rumpf-Objekten r, für die Regel erfüllt ist

TAID	Items
001	Beer, Coke, Peanuts
002	Beer, Chips, Cigarettes
003	Beer, Chips, Cigarettes, Coke
004	Beer, Cigarettes

Beer $\rightarrow$ Coke: Confidence = 50%
Coke $\rightarrow$ Beer: Confidence = 100%
...

Support(Beer) = 100%
Support(Chips) = 50%
...
Support(Beer, Coke) = 50%
Support(Beer,Peanuts) = 25%
...
Support(Beer, Coke, Peanuts) = 25%
Support(Beer, Chips, Cigarettes) = 50%
...

A-Priori-Algorithmus (1)

- Frequent Item Set:
  Item-Menge, deren Support gewisse Schranke s übersteigt
- Bestimmung der Frequent Item Sets 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 >=1),
  - Ausklammern von Kombinationen, welche Teilmengen haben, die Support s nicht erreichen
  - wird „a priori“ getestet, bevor Support besimmt wird

A-Priori-Algorithmus (2)

```
L_1 = find_frequent_1_itemsets();
for(k=2; L_{k-1}≠∅;k++){
  C_k = generateCandidates(L_{k-1});
  for each transaction t{
      for each candidate c ∈ C_k {
          if (t contains c)
            c.count++;
      }
  }
  L_k = {c ∈ C_k | c.count >= MIN_SUP}
}
```


A-Priori-Algorithmus (3)

```
procedure generateCandidates(L_{k-1}){
  for each itemset l_1 ∈ L_{k-1} {
      for each itemset l_2 ∈ L_{k-1} {
         if(l_1[1]=l_2[1] && … && l_1[k-2]=l_2[k-2] &&
                                      l_1[k-1]<l_2[k-1]){
            c = l_1[1],l_1[2],… l_1[k-1],l_2[k-1];
           C_k.add(c) unless prune(c, L_{k-1});
}}}}

procedure prune(c, L_{k-1}){
  for each (k-1)-subset s of c {
      if(s ∉ L_{k-1}) return true;
  }
  return false;
}
```

A-Priori-Algorithmus: Beispiel (1)

TAID	Items
001	I1, I2, I5
002	I2, I4
003	I2, I3
004	I1, I2, I4
005	I1, I3
006	I2, I3
007	I1, I3
008	I1, I2, I3, I5
009	I1, I2, I3

MIN_SUP = 2

k=1: C_1

Itemset	Sup #
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare with MIN_SUP →

L_1

Itemset	Sup #
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

k=2: C_2

Itemset	Sup #
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

kein pruning für k=2 da alle subsets frequent

Compare with MIN_SUP →

L_2

Itemset	Sup #
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

A-Priori-Algorithmus: Beispiel (1)

L_2

Itemset	Sup #
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

k=3: C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}
~~{I1, I3, I5}~~
~~{I2, I3, I4}~~
~~{I2, I3, I5}~~
~~{I2, I4, I5}~~

{I3, I5} ∉ L_2
{I3, I4} ∉ L_2
{I3, I5} ∉ L_2
{I4, I5} ∉ L_2

Count Support →

C_3

Itemset	Sup #
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare with MIN_SUP ↓

L_3

Itemset	Sup #
{I1, I2, I3}	2
{I1, I2, I5}	2

k=4: C_4

Itemset
~~{I1, I2, I3, I5}~~

{I2, I3, I5} ∉ L_3

Erzeugen der Assoziationsregeln

- Basis: Frequent Item Sets

1. Für jedes Frequent Item Set l erzeuge alle Teilmengen
2. Für jede Teilmenge s von l erzeuge die Regel
   $s \rightarrow (l - s)$ falls confidence$(s \rightarrow (l - s))$ > MIN_CONF

- confidence$(A \rightarrow B) = P(B|A) = \text{support}(A \cup B)/\text{support}(A)$