

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Institut für Informatik  
Lehrstuhl Wissensmanagement



---

# Assoziationsregeln, Suche nach Zusammenhängen in Datenbanken

Tobias Scheffer  
Steffen Bickel

# Assoziationsregeln

- Suche nach Regeln, Mustern, Zusammenhängen in Datenbanken.
- Explorative Datenanalyse, ungerichtete Suche nach Auffälligkeiten.
- Muster werden von Menschen analysiert, die neues Wissen interpretieren.
- neue Zusammenhänge können z.B. helfen, Geschäftsprozesse zu optimieren.

# Assoziationsregeln

- Werden z.B. verwendet für
  - ◆ Analyse von „Basket Data“ in Supermärkten.
  - ◆ Cross-marketing,
  - ◆ Layout-Optimierung in Supermärkten.
  - ◆ Qualitätsmanagement, Analyse von Qualitäts-, Garantie-, Werkstattdaten..
  - ◆ Analyse von Produktions- und Logistikdaten mit dem Ziel der Optimierung von Prozessen.

# Attribute und Items

- Ausgangsbasis:
  - ◆ Datenbank mit Attributen  $\{A_1, \dots, A_m\}$ .
  - ◆ z.B. ein Attribut pro Produkt für Basket Data.
- Item: Belegung eines Attributs mit einem Wert.
  - ◆ Häufig der Einfachheit halber binäre Attribute.
  - ◆ z.B. ein Produkt, das bei einer Transaktion gekauft wurde.
- Transaktion: Menge von Items.
  - ◆ Attribute, die in einer Zeile (für eine Transaktion) den Wert 1 annehmen.
  - ◆ z.B. Alle Produkte, die bei Transaktion gekauft wurden.

# Itemsets und Assoziationsregeln

- Frequent Itemsets:
  - ◆ Mengen von Items, die häufig gemeinsam auftreten.
  - ◆ z.B. {Caçasa, Limetten, brauner\_Zucker}.
  - ◆ Für sich genommen interessant, zusätzlich Vorstufe von Assoziationsregeln.
- Assoziationsregeln:
  - ◆ Implikationen zwischen Itemsets, die häufig gelten,
  - ◆ z.B. Caçasa, Limetten  $\Rightarrow$  brauner\_Zucker.

# Itemsets: Beispiel

- I = Menge von Items (z.B. {Chips, Bier, Windeln, Caçasa, Limetten, brauner\_Zucker, Milch, ...})
- T = Menge von Transaktionen  
(z.B. { {Chips, Bier}, {Caçasa, Limetten, brauner\_Zucker}, {Milch, Bier, Windeln}, ...})
- Beispiel-Itemsets
  - ◆ „Caçasa, Limetten, brauner\_Zucker“
  - ◆ „Bier, Chips, Fernsehzeitschrift“
  - ◆ „Cornflakes, Babybrei, Milch, Windeln“

# Assoziationsregeln: Beispiel

- I = Menge von Items (z.B. {Chips, Bier, Windeln, Caçasa, Limetten, brauner\_Zucker, Milch, ...})
- T = Menge von Transaktionen (z.B. { {Chips, Bier}, {Caçasa, Limetten, brauner\_Zucker}, {Milch, Bier, Windeln}, ...})
- Beispiel-Assoziationsregeln
  - ◆ „Caçasa, Limetten  $\Rightarrow$  brauner\_Zucker“
  - ◆ „Bier, Chips  $\Rightarrow$  Fernsehzeitschrift“
  - ◆ „Cornflakes, Babybrei  $\Rightarrow$  Milch, Windeln“

# Support, Confidence, Lift

- Gegeben Itemsets  $X$ ,  $Y$ , Transaktionen  $T$ .
- Support von Itemset  $X$ :
  - ◆ Anzahl der Transaktionen, in denen  $X$  auftritt.
  - ◆  $s(X) = |\{t \in T : X \subseteq t\}|$
- Support der Assoziationsregel  $X \Rightarrow Y$ :
  - ◆ Anteil der Transaktionen, in denen  $X$  und  $Y$  auftreten.
  - ◆  $s(X \Rightarrow Y) = \frac{|\{t \in T : X \subseteq t, Y \subseteq t\}|}{|T|}$

# Support, Confidence, Lift

- Confidence der Regel  $X \Rightarrow Y$ :
  - ◆ Anteil der Regeln für die  $Y$  gilt unter denen, für die schon  $X$  gilt.
  - ◆ 
$$c(X \Rightarrow Y) = \frac{|\{t \in T : X \subseteq t, Y \subseteq t\}|}{|\{t \in T : X \subseteq t\}|} = \frac{s(X, Y)}{s(X)}$$
- Lift der Regel  $X \Rightarrow Y$ :
  - ◆ Support von  $X, Y$  geteilt durch Support bei Unabhängigkeit von  $X$  und  $Y$ .
  - ◆ 
$$l(X \Rightarrow Y) = \frac{s(X, Y)}{s(X)s(Y)} = \frac{c(X \Rightarrow Y)}{s(Y)}$$

# Assoziationsregeln: Problemstellung

- Gegeben
  - ◆ Menge  $I$  von Items,
  - ◆ Menge  $T$  von Transaktionen;  $t \in T \Rightarrow t \subseteq I$ .
- Finde alle Regeln  $r = (X \Rightarrow Y)$  mit  $X \subseteq I, Y \subseteq I$ , so dass
  - ◆ Support  $\geq s_{\min}$ .
  - ◆ Confidence  $\geq c_{\min}$ .
- Finde alle Regeln, die allgemeingültig und genau sind.

# Assoziationsregeln: Apriori-Algorithmus

1. Finde häufige Itemsets; Itemsets mit Support  $\geq S_{\min}$ .
  1. Starte von Itemsets der Größe 1
  2. Erzeuge aus den Itemsets der Größe  $i$  die der Größe  $i+1$ .
2. Für alle gefundenen häufigen Itemsets, bilde alle Regeln über diesen Items und teste die Confidence.
3. Liefere alle Regeln mit ausreichend hohem Support und Confidence.

# Suche nach häufigen Itemsets

- Häufige-Itemsets ( $s_{\min}$ )
  1.  $C_1 = \{ \{ i \} \mid i \in I \}$  (Itemsets der Größe 1)
  2.  $L_1 = \text{Prune}(C_1, s_{\min})$ .
  3. Solange  $L_k \neq \{ \}$ 
    1.  $C_{k+1} = \text{Generate}(L_k)$
    2.  $L_{k+1} = \text{Prune}(C_{k+1}, s_{\min})$
    3. Increment  $k$ .
  
- Generate ( $L_k$ ) [Version 1]
  1. Return  $\{ \{L_k\} \cup i : i \in I, |\{L_k\} \cup i| = k+1 \}$

## $L_k = \text{Prune}(C_k, s_{\min})$

1. Für alle  $c \in C_k : \text{count}(c) = 0$ .
  2. Für alle  $t \in T$ : (Datenbank-Scan)
    1. Für alle  $c \subseteq t$ : increment  $\text{count}(c)$ .
  3. Return  $\{c \in C_k \mid \text{count}(c) \geq s_{\min}\}$
- 
- Finden wir so wirklich alle häufigen Itemsets? Das Pruning beschränkt den Suchraum.

# Suche nach häufigen Itemsets

- Wie hoch kann der Support von  $(X \Rightarrow Y)$  sein, wenn  $s(X \cup Y) = \text{sup}$  ist?
- Kann der Support  $s(X \cup Y)$  größer sein als der Support  $s(X)$ ?
- Für alle  $c \in \text{Generate}(X)$ :  $s(c) \leq s(X)$ .
- Suchraum nach Itemsets ist baumförmig; der Support sinkt in jedem Ast monoton.

# Bilden von Regeln aus häufigen Itemsets

- Für alle gefundenen häufigen Itemsets, bilde alle Regeln über diesen Items und teste die Confidence.

- $$c(X \Rightarrow Y) = \frac{|\{t \in T : X \cup Y \subseteq t\}|}{|\{t \in T \mid X \subseteq t\}|} = \frac{s(X \cup Y)}{s(X)}$$

# Bilden von Regeln über Itemset

- Finde alle Regeln über die häufigen Itemsets  $I_k$  mit  $k \geq 2$  Elementen
- 1. Für alle  $X \in I_k$ 
  1. Für alle  $Y \subseteq X$ ,  $Y$  nicht leer.
    1. Wenn  $c(X \setminus Y \Rightarrow Y) = \frac{s(X)}{s(X \setminus Y)} \geq c_{\min}$
    2. Dann gib Regel  $(X \setminus \{Y\} \Rightarrow Y)$  aus.

# Apriori: Beispiel

	Gin	Tonic	Kaluha	Chips	Cacasa	Limetten	Zucker
1		1	1	1			
2	1	1		1			
3					1	1	1
4					1	1	1
5	1	1	1				
6				1	1	1	1
7	1	1			1	1	1

- $S_{\min} = 3$
- $C_{\min} = 1$

# Bilden von Regeln aus häufigen Itemsets (Optimierung)

- Pruning-Strategie für die Bildung von Regeln

- Beobachtung

$$c(XY \Rightarrow Z) = \frac{|\{t \in T : X \cup Y \cup Z \in t\}|}{|\{t \in T \mid X \cup Y \in t\}|}$$
$$\geq \frac{|\{t \in T : X \cup Y \cup Z \in t\}|}{|\{t \in T \mid X \in t\}|} = c(X \Rightarrow YZ)$$

- Wenn also schon  $(XY \Rightarrow Z)$  unter  $c_{\min}$  liegt, dann müssen wir  $(X \Rightarrow YZ)$  und  $(Y \Rightarrow XZ)$  nicht mehr testen.

# Bilden von Regeln über Itemset

- Finde alle Regeln über die häufigen Itemsets  $I_k$  mit  $k \geq 2$  Elementen
  1.  $m=1$ .
  2.  $H_m =$  alle einelementigen Teilmengen von  $I_k$ .
  3. Für  $m = 1 \dots k-1$ 
    1. Für alle  $h \in H_m$ ,  $r = (I_k \setminus \{h\} \Rightarrow h)$ 
      - Wenn  $c(I_k \setminus \{h\} \Rightarrow h) = \frac{s(I_k)}{s(I_k \setminus \{h\})} \geq c_{\min}$
      - Dann gib Regel  $r$  aus.
      - Sonst  $H_m = H_m \setminus \{h\}$
    2.  $H_{m+1} = \text{Generate}(H_m)$

# Generate ( $L_k$ ) (Optimierung)

- Alle Teilmengen eines häufigen Itemsets sind auch wieder häufige Itemsets.
- Idee: Generierung häufiger Itemsets  $m+1$  durch Kombination häufiger Itemsets der Größe  $m$ .
- Generate ( $L_k$ )
  1.  $C_{k+1} = \{ \}$
  2. Für alle  $X \in L_k$ , Für alle  $Y \in L_k$ 
    1. Wenn sich  $X$  und  $Y$  nur im letzten Element unterscheiden und das letzte Element von  $X$  kleiner als das von  $Y$  ist, dann  $C_{k+1} = C_{k+1} + X \cup Y$

# Generate ( $L_k$ ) (Optimierung)

- Korrektheit? Vollständigkeit?
- Supermarkt-Beispiel mit neuem Generate-Algorithmus und neuem Regel-Generator-Algorithmus

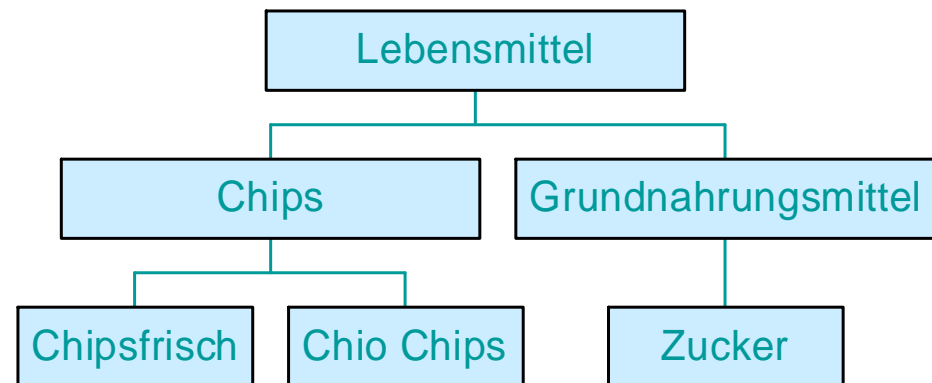
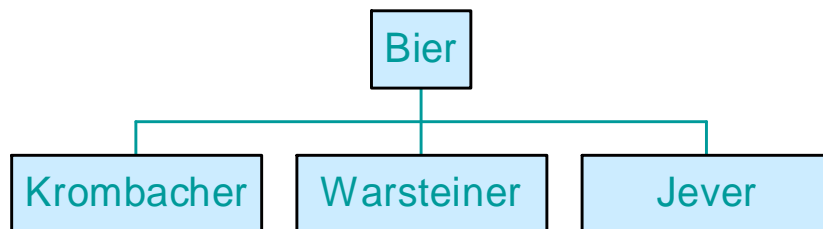
# Apriori TID

- Vor allem Datenbankzugriffe verursachen Kosten bei Apriori.
- Für großes  $k$  gibt es viele Datenbankeinträge, die zu keinem häufigen Itemset gehören, trotzdem wird auf diese zugegriffen.
- Apriori TID: Für jeden Eintrag wird die Liste aller häufigen Itemsets gespeichert, zu denen der Eintrag gehört. Wenn die Liste leer ist, wird der Eintrag gelöscht.

# Taxonomien: Generalisierten Assoziationsregeln

- Häufig sind Assoziationsregeln recht redundant.
- Assoziationen zwischen Produktgruppen können schlecht ausgedrückt werden.
- Wenn z.B. jeder Kunde, der Chips kauft, auch Bier kauft, dann finden sich Regeln
  - ◆ Chio Chips  $\Rightarrow$  Krombacher
  - ◆ Chio Chips  $\Rightarrow$  Warsteiner
  - ◆ Chipsfrisch Chips  $\Rightarrow$  Jever
- Idee: Abstraktion durch Taxonomien

# Taxonomien



- „Chips  $\Rightarrow$  Bier“
- „Jever  $\Rightarrow$  Chipsfrisch“

# Generalisierte Assoziationsregeln: Problemstellung

- Gegeben
  - ◆ Menge  $I$  von Items,
  - ◆ Menge  $T$  von Transaktionen;  $t \in T \Rightarrow t \subseteq I$ .
- Azyklischer Graph  $TAX$  über Knoten  $I$ .
  - ◆  $(p,c) \in TAX$ :  $p$  ist allgemeiner als  $c$ .
  - ◆  $TAX^*$ : Transitive Erweiterung von  $TAX$ .
- Finde alle Regeln  $r = (X \Rightarrow Y)$  mit  $X \subseteq I$ ,  $Y \subseteq I$ , so dass
  - ◆ Support  $s(r) \geq s_{\min}$
  - ◆ Confidence  $c(r) \geq c_{\min}$

- Wobei

$$s(X) = \frac{|\{t \in T \mid \forall x \in X : x \in t \vee (\exists y : (x, y) \in TAX^* \wedge y \in t)\}|}{|T|}$$

# „Cumulate“-Algorithmus

- Häufige-Itemsets ( $s_{\min}$ , TAX)
- 1. Bestimme TAX\*
- 2.  $C_1$  =häufige Itemsets Größe 1;  $k=2$
- 3. Solange  $L_k \neq \{ \}$ 
  - 1.  $C_{k+1}$  = Generiere Kandidaten
  - 2. Wenn  $k=2$ , dann eliminiere Kandidaten, die aus einem Item und seinem Vorgänger bestehen
  - 3. Lösche alle  $c \in TAX^*$  für die kein  $(c,p)$ ,  $p \in c$ ,  $c \in C_{k+1}$  existiert.
  - 4. Für alle  $t \in T$ , für alle  $p \in t$ , füge alle  $c$  mit  $(c,p) \in TAX^*$  zu  $t$  hinzu, entferne Duplikate.
  - 5. Erhöhe Zähler für alle Kandidaten  $c$ , die in  $t$  und in  $C_{k+1}$  vorkommen.
  - 6.  $L_{k+1}$  = Prune ( $C_{k+1}$ ); Erhöhe  $k$ .
- 4. Rückgabe ist Vereinigung aller  $L_k$ .

# Redundanz

- Assoziationsregeln enthalten
  - ◆ viele redundante Regeln, die sich gegenseitig implizieren.
  - ◆ viele schon bekannte Zusammenhänge.
- Redundanz beseitigen:
  - ◆ Finde möglichst allgemeine Regeln.
  - ◆ Eliminiere Regeln, die von anderen Regeln impliziert werden.
  - ◆  $[X \Rightarrow Y] \models [X' \Rightarrow Y']$  wenn  $X \subseteq X'$  und  $Y' \subseteq Y$
  - ◆ Closed Itemsets.
  - ◆ Verschiedene Ansätze, Problem aber nicht gelöst.

# Redundanz

- Assoziationsregeln enthalten
  - ◆ viele redundante Regeln, die sich gegenseitig implizieren.
  - ◆ viele schon bekannte Zusammenhänge.
  
- Bekannte Zusammenhänge beseitigen:
  - ◆ Erfordert Modell davon, was benutzer schon weiß.
  - ◆ Verschiedene Ansätze, Problem aber nicht gelöst.
  - ◆ Z.B. interaktive Wissensentdeckung,
  - ◆ Entdeckte, vom Benutzer bestätigte Zusammenhänge werden in Modell eingefügt.
  - ◆ Algorithmus sucht nach neuen Zusammenhängen, die noch nicht vom Modell erklärt werden.

# Subgruppen

- Variante der Problemstellung.
- Zusätzlich gegeben: Zielattribut  $x$ .
  - ◆ Z.B. Indikator, ob Person bestimmtes Produkt gekauft hat.
- A-Priori-Wahrscheinlichkeit über Datenbank:
  - ◆  $p_0 = P(x = 1 | T)$
- Subgruppe:
  - ◆ Teilmenge von  $T$ , durch Itemset beschrieben.
  - ◆ z.B. [weiblich, alter\_25\_30].

# Subgruppen

- Generality, Allgemeinheit einer Subgruppe:
  - ◆  $g(X) = \frac{s(X)}{|T|}$
- Usefulness, Ungewöhnlichkeit einer Subgruppe:
  - ◆  $u(X) = |P(x=1|X) - p_0|$
- Finde n Subgruppen  $X$ , die
  - ◆  $g(X)u(X)$
- maximieren.
- Kombination von Ungewöhnlichkeit und Generalität.
  - ◆ Z.B. Gruppen von Personen, die das Produkt besonders häufig oder besonders selten kaufen.

# Muster in Datenbanken

- Itemsets, Assoziationsregeln.
  - ◆ Drücken Häufungen in Datenbanken aus.
  - ◆ Explorative, ungerichtete Datenanalyse.
  - ◆ Apriori-Algorithmus, Optimierungen, Varianten.
- Meist werden sehr viele Regeln gefunden:
  - ◆ Eliminieren redundanter, bekannter Regeln.
- Variante der Problemstellung: Subgruppen.
  - ◆ Ungewöhnliche Gruppen bezogen auf Zielattribut.