

# DATA SCIENCE 2

## VORLESUNG 5 - WARENKORBANALYSE

PROF. DR. CHRISTIAN BOCKERMANN

HOCHSCHULE BOCHUM

SOMMERSEMESTER 2023

1 Frequent Itemsets / Patterns

2 Assoziationsregeln

# Frequent Itemsets / Patterns

## Frequent Itemset Mining sucht häufige Muster

- Gegeben ist Menge  $\mathbf{S}$  von Symbolen (z.B. Artikel)
- Eingabe ist Menge  $\mathbf{X}$  von Transaktionen (Einkäufe) über  $\mathbf{S}$

$$\mathbf{X} = \{x \mid x \subseteq \mathbf{S}\}$$

### Ziel:

- Frage: Welche Symbole tauchen häufig zusammen auf?
- Finde die Muster  $p \in \mathcal{P}(\mathbf{S})$  die in  $\mathbf{X}$  am häufigsten vorkommen

## Beispiel: Frequent Itemsets auf Einkäufen

ID	Artikel
1	{ A, B, F }
2	{ B, D, E, F }
3	{ C, E }
4	{ B, E, F }
5	{ A, B, E }

## Beispiel: Frequent Itemsets auf Einkäufen

ID	Artikel
1	{ A, B, F }
2	{ B, D, E, F }
3	{ C, E }
4	{ B, E, F }
5	{ A, B, E }

- Artikel B = Muster { B } taucht in 4/5 der Einkäufe auf

## Beispiel: Frequent Itemsets auf Einkäufen

ID	Artikel
1	{ A, B, F }
2	{ B, D, E, F }
3	{ C, E }
4	{ B, E, F }
5	{ A, B, E }

- Artikel B = Muster { B } taucht in 4/5 der Einkäufe auf
- Muster { B, F } taucht in 3/5 aller Einkäufe auf

## Beispiel: Frequent Itemsets auf Einkäufen

ID	Artikel
1	{ A, B, F }
2	{ B, D, E, F }
3	{ C, E }
4	{ B, E, F }
5	{ A, B, E }

- Artikel B = Muster { B } taucht in 4/5 der Einkäufe auf
- Muster { B, F } taucht in 3/5 aller Einkäufe auf

**Welche Artikel werden häufig zusammen gekauft?**



## Transaktionsdatenbank

Transaktionen für Frequent Pattern Mining werden in **Transaktionsdatenbank** gespeichert:

ID	A	B	C	D	E	F
1	1	1	0	0	0	1
2	0	1	0	1	1	1
3	0	0	1	0	1	0
4	0	1	0	0	1	1
5	1	1	0	0	1	0

## Transaktionsdatenbank

Transaktionen für Frequent Pattern Mining werden in  
Transaktionsdatenbank gespeichert:

ID	A	B	C	D	E	F
1	1	1	0	0	0	1
2	0	1	0	1	1	1
3	0	0	1	0	1	0
4	0	1	0	0	1	1
5	1	1	0	0	1	0

Transaktion  $t_1 = \{ A B F \}$

## Transaktionsdatenbank

Transaktionen für Frequent Pattern Mining werden in  
Transaktionsdatenbank gespeichert:

ID	A	B	C	D	E	F
1	1	1	0	0	0	1
2	0	1	0	1	1	1
3	0	0	1	0	1	0
4	0	1	0	0	1	1
5	1	1	0	0	1	0

Transaktion  $\mathbf{t}_1 = \{ A B F \}$   
 $\mathbf{t}_1$  enthält Muster  $\mathbf{p} = \{ A B \}$

## Allgemein: Wann ist ein Muster häufig?

Sei  $\mathcal{D}$  eine Datenbank mit Transaktionen.

Der *support* eines Musters  $p$  in einer Datenbank  $\mathcal{D}$  ist

$$\text{support}(p) = \frac{|\{t \in \mathcal{D} \mid p \subseteq t\}|}{|\mathcal{D}|}.$$

## Allgemein: Wann ist ein Muster häufig?

Sei  $\mathcal{D}$  eine Datenbank mit Transaktionen.

Der *support* eines Musters  $p$  in einer Datenbank  $\mathcal{D}$  ist

$$\text{support}(p) = \frac{|\{t \in \mathcal{D} \mid p \subseteq t\}|}{|\mathcal{D}|}.$$

Muster  $p$  ist häufig, wenn  $\text{support}(p) > \text{min}_s$ , d.h.

- es taucht in  $\text{min}_s$  % aller Transaktionen auf
- Parameter  $\text{min}_s$  vom Benutzer gewählt
- $\text{min}_s$  ist der minimale Support

## Wie finden wir häufige Muster?

Sei  $\mathcal{D}$  die Datenbank mit Transaktionen  $\mathbf{t}_i$  und jedes  $\mathbf{t}_i \subseteq \mathbf{S}$  eine Menge von Symbolen aus  $\mathbf{S} = \{s_1, \dots, s_k\}$ .

## Wie finden wir häufige Muster?

Sei  $\mathcal{D}$  die Datenbank mit Transaktionen  $\mathbf{t}_i$  und jedes  $\mathbf{t}_i \subseteq \mathbf{S}$  eine Menge von Symbolen aus  $\mathbf{S} = \{s_1, \dots, s_k\}$ .

Menge aller möglichen Teilmengen von  $\mathbf{S}$  ist die Potenzmenge

$$\mathcal{P}(\mathbf{S}) = \{\{\}, \{s_1\}, \{s_2\}, \dots, \{s_1, s_2\}, \dots, \{s_1, \dots, s_k\}\}$$

## Wie finden wir häufige Muster?

Sei  $\mathcal{D}$  die Datenbank mit Transaktionen  $\mathbf{t}_i$  und jedes  $\mathbf{t}_i \subseteq \mathbf{S}$  eine Menge von Symbolen aus  $\mathbf{S} = \{s_1, \dots, s_k\}$ .

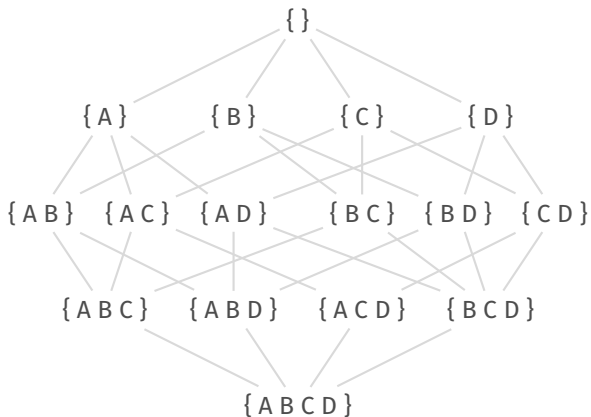
Menge aller möglichen Teilmengen von  $\mathbf{S}$  ist die Potenzmenge

$$\mathcal{P}(\mathbf{S}) = \{\{\}, \{s_1\}, \{s_2\}, \dots, \{s_1, s_2\}, \dots, \{s_1, \dots, s_k\}\}$$

Wir müssen die Häufigkeit aller Teilmengen aus  $\mathcal{P}(\mathbf{S})$  zählen!



## Potenzmenge als **Teilmengen-Verbund**



**Problem: Größe der Potenzmenge ist exponentiell**

$$|\mathcal{P}(\mathbf{S})| = 2^{|\mathbf{S}|}$$

## Problem: Größe der Potenzmenge ist exponentiell

$$|\mathcal{P}(\mathbf{S})| = 2^{|\mathbf{S}|}$$

### Beispiel:

- Ein Online-Shop hat 10 Produkte: 1024 Teilmengen

## Problem: Größe der Potenzmenge ist exponentiell

$$|\mathcal{P}(\mathbf{S})| = 2^{|\mathbf{S}|}$$

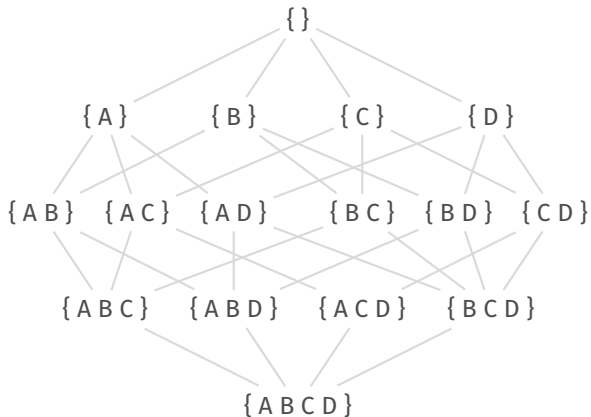
### Beispiel:

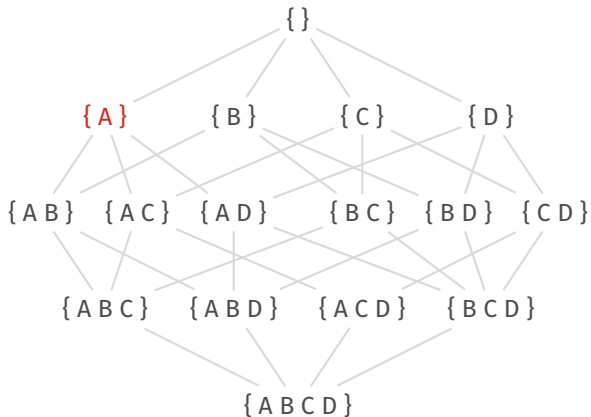
- Ein Online-Shop hat 10 Produkte: 1024 Teilmengen
- Ein Online-Shop hat 100 Produkte:  
1267650600228229401496703205376 Teilmengen

**Welcher Shop hat nur 100 Produkte?**

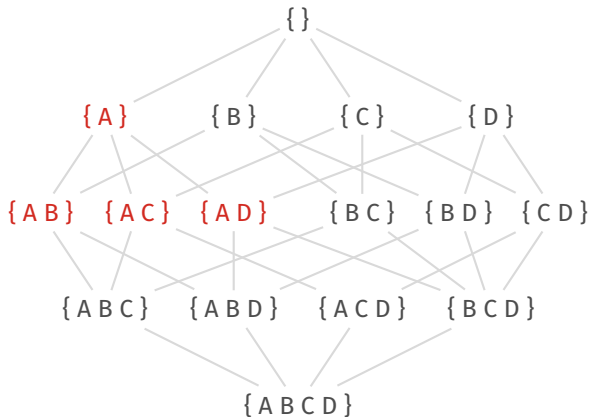
## Idee: Häufigkeit ist monoton fallend

Wenn eine Menge  $t$  nicht häufig ist, ist jede Menge  $p$ , die  $t$  enthält, ebenfalls nicht häufig.



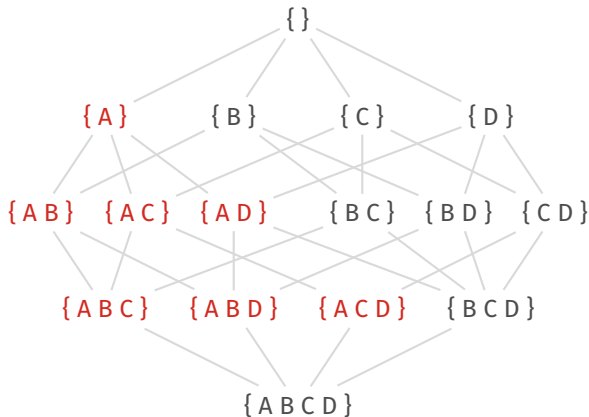


Wenn  $\{A\}$  nicht häufig ist, brauchen wir alle Mengen, die A enthalten nicht mehr zu zählen!

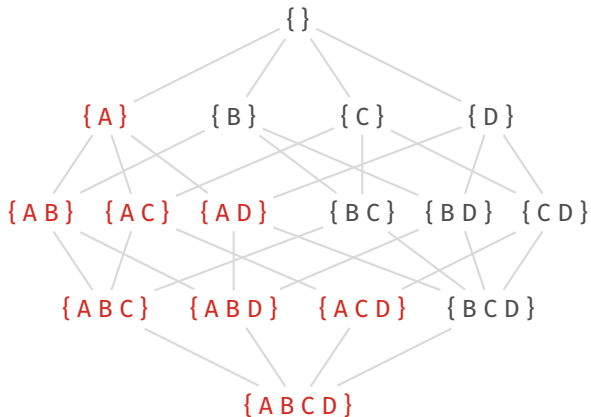


Wenn  $\{A\}$  nicht häufig ist, brauchen wir alle Mengen, die A enthalten nicht mehr zu zählen!





Wenn  $\{A\}$  nicht häufig ist, brauchen wir alle Mengen, die A enthalten nicht mehr zu zählen!



Wenn { A } nicht häufig ist, brauchen wir alle Mengen, die A enthalten nicht mehr zu zählen!

## Der **Apriori** Algorithmus

1. Sei  $k = 1$ .  
Bilde die  $k$ -Elementigen Teilmengen  $L_k$  über  $\mathbf{S}$  und zähle ihre Häufigkeit.
2. Bilde aus den häufigen  $k$ -elementigen Mengen die möglichen  $(k + 1)$ -Mengen  $L_{k+1}$
3. Zähle die Häufigkeit der  $L_{k+1}$ -Mengen in  $\mathcal{D}$
4. Wenn  $L_{k+1}$  noch häufige Mengen enthält, setze  $k := k + 1$  und wiederhole ab Schritt 2.  
andernfalls: Stopp.

## Der Apriori Algorithmus

- In SciKit Learn leider **nicht** enthalten
- Modul `mlxtend` enthält Apriori Implementierung

## Zusätzlich

- One-Hot-Encoder um Transaktionstabelle (0, 1 pro Symbol) zu erzeugen
- Funktioniert auch mit CountVectorizer (vgl. Text-Clustering: DataScience 2, 5. Vorlesung)

## Beispiel:

```
import pandas as pd
from mlxtend.frequent_patterns import apriori

# Transaktionen laden (pro Artikel 1 Spalte mit 0/1)
txs = pd.read_csv('Kurse/DataScience2/data/
                  transactions.csv')

# Apriori-Algorithmus anwenden:
result = apriori(txs, min_support=0.02,
                 use_colnames=True)

# Ergebnis ist wieder ein DataFrame Objekt!
```

## Beispiel: (Ergebnis)

	<b>support</b>	<b>itemsets</b>
<b>32</b>	0.05	(antioxydant, juice)
<b>33</b>	0.05	(cottage, cheese)
<b>34</b>	0.05	(drink, energy)
<b>35</b>	0.05	(fat, low)
<b>36</b>	0.05	(yogurt, fat)
<b>37</b>	0.05	(flour, weat)

## Alternative zu Apriori-Algorithmus

- Apriori-Algorithmus basiert auf *Kandidatengenerierung*
- u.U. recht langsam, viele DB-Iterationen
- **FP-Growth** ist alternativer Algorithmus
- Nutzt Kompakte Repräsentation der Datenbank
- nur 2 Iterationen auf DB, danach in-Memory

## Häufige Mengen - was nun?

Angenommen, wir haben unsere häufigen Mengen gefunden:

Support	Itemsets
0.53	{ A D }
0.42	{ D F }
0.17	{ A D F }
0.13	{ A C G }



# Assoziationsregeln

## Assoziationsregeln – Wenn-Dann

Regeln der Art

$$\mathbf{x} \rightarrow \mathbf{y}$$

wobei  $\mathbf{x}$  und  $\mathbf{y}$  jeweils Mengen von Symbolen aus  $\mathbf{S}$  sind.

## Assoziationsregeln – Wenn-Dann

Regeln der Art

$$\mathbf{x} \rightarrow \mathbf{y}$$

wobei  $\mathbf{x}$  und  $\mathbf{y}$  jeweils Mengen von Symbolen aus  $\mathbf{S}$  sind.

### Wie häufig kommt die Regel in der Datenbank vor?

Für Assoziationsregeln aus einer Datenbank  $\mathcal{D}$  ist der **Support**:

$$\text{support}(\mathbf{x} \rightarrow \mathbf{y}) = \frac{\text{support}(\mathbf{x} \cup \mathbf{y})}{|\mathcal{D}|}$$

## Beispiel:

Kunden, die Brot und Eier gekauft haben,  
haben auch Milch gekauft.

Dies läßt sich als Regel formulieren:

$$\{ \text{Brot, Eier} \} \rightarrow \{ \text{Milch} \}$$

## Beispiel:

Kunden, die Brot und Eier gekauft haben,  
haben auch Milch gekauft.

Dies lässt sich als Regel formulieren:

$$\{ \text{Brot, Eier} \} \rightarrow \{ \text{Milch} \}$$

Der Support der Regel ist:

$$\frac{\text{support}(\{ \text{Brot, Eier, Milch} \})}{|\mathcal{D}|}$$

## Regeln aus häufigen Mengen erzeugen

Regeln werden aus den häufigen Mengen erzeugt. Die Menge

$$\{ A B C \}$$

führt zu den Regeln

$$\{ A B \} \rightarrow \{ C \}$$

$$\{ A C \} \rightarrow \{ B \}$$

$$\{ B C \} \rightarrow \{ A \}$$

## Regeln aus häufigen Mengen erzeugen

Regeln werden aus den häufigen Mengen erzeugt. Die Menge

$$\{ A B C \}$$

führt zu den Regeln

$$\{ A B \} \rightarrow \{ C \}$$

$$\{ A C \} \rightarrow \{ B \}$$

$$\{ B C \} \rightarrow \{ A \}$$

**Das führt zu sehr vielen Regeln!**

**Welche sind davon relevant?**

## Bewerten von Assoziationsregeln – **Konfidenz**

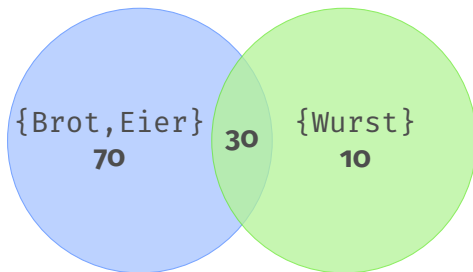
Wenn **x** gilt – wie häufig gilt dann auch **y**?

$$\text{conf}(\mathbf{x} \rightarrow \mathbf{y}) = \frac{\text{support}(\mathbf{x} \cup \mathbf{y})}{\text{support}(\mathbf{x})}$$



## Beispiel: Konfidenz

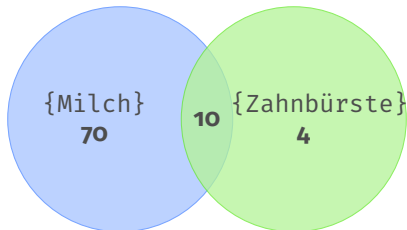
Sei  $|\mathcal{D}| = 100$ .

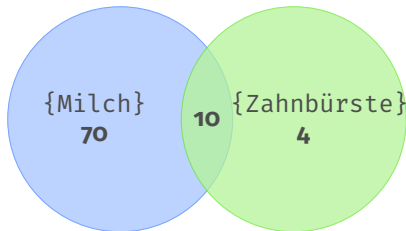


$$\frac{\text{support}(\{ \text{Brot, Eier, Wurst} \})}{\text{support}(\{ \text{Brot, Eier} \})} = \frac{0.3}{0.7} \approx 0.428$$

## Problem: Was ist mit sehr populären Artikeln?

{ Zahnbürste } → { Milch }

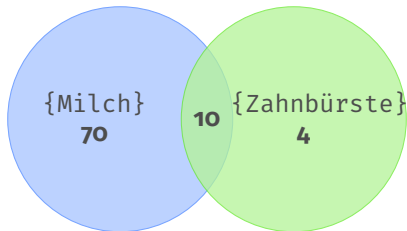


**Problem:** Was ist mit sehr populären Artikeln? $\{ \text{Zahnbürste} \} \rightarrow \{ \text{Milch} \}$ 

$$\frac{\text{support}(\{ \text{Zahnbürste}, \text{Milch} \})}{\text{support}(\{ \text{Zahnbürste} \})} = \frac{10}{10+4} \approx \mathbf{0.7}$$

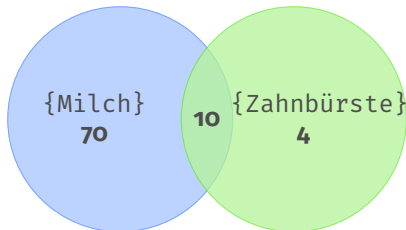
## Problem: Was ist mit sehr populären Artikeln?

$$\text{conf}(\{ \text{Zahnbürste} \} \rightarrow \{ \text{Milch} \}) = \mathbf{0.7}$$



## Problem: Was ist mit sehr populären Artikeln?

$$\text{conf}(\{ \text{Zahnbürste} \} \rightarrow \{ \text{Milch} \}) = \mathbf{0.7}$$



**Wie aussagekräftig ist die hohe Konfidenz für  
 $\{ \text{Zahnbürste} \} \rightarrow \{ \text{Milch} \}$ ?**

## Lift Kriterium

“Das Lift Kriterium gibt die Steigerung der Wahrscheinlichkeit für **y** an, wenn **x** gilt.”

$$\text{lift}(\mathbf{x} \rightarrow \mathbf{y}) = \frac{\text{support}(\mathbf{x} \rightarrow \mathbf{y})}{\text{support}(\mathbf{x}) \cdot \text{support}(\mathbf{y})}$$

### Interpretation:

$\text{lift}(\mathbf{x} \rightarrow \mathbf{y}) > 1 \Rightarrow \mathbf{x}, \mathbf{y}$  sind positiv korreliert

$\text{lift}(\mathbf{x} \rightarrow \mathbf{y}) < 1 \Rightarrow \mathbf{x}, \mathbf{y}$  sind negativ korreliert

$\text{lift}(\mathbf{x} \rightarrow \mathbf{y}) = 1 \Rightarrow \mathbf{x}, \mathbf{y}$  sind unabhängig

## Assoziationsregeln in Python

- Modul `mlxtend` ermöglicht Generieren von Regeln
- Häufige Mengen vorab mit z.B. Apriori berechnen

```
from mlxtend.frequent_patterns import
                                association_rules

# haeufige Mengen berechnen...
freq_patterns = apriori(...)

# Regeln generieren:
rules = association_rules(freq_patterns,
                          metric="confidence",
                          min_threshold=0.7)
```