

Marcel Gehrke

---

# Data Understanding vs. Machine Training

## Affinity Analysis (Unsupervised Learning)

# Recap: Supervised Learning

Given:

- Tabular data,
- Classification attribute exists  
(Supervision through classified data)

Wanted: Classifier for new data

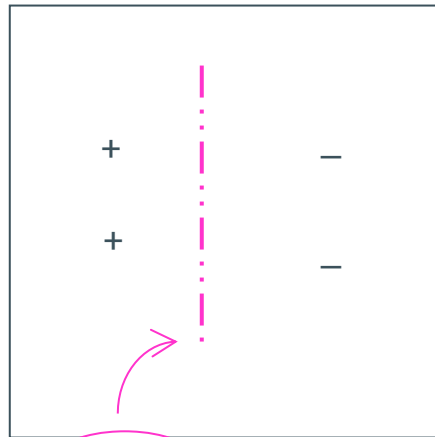
Today: Unsupervised learning  
(no classification attribute available)

Unsupervised Learning

---

# Frequency Analysis

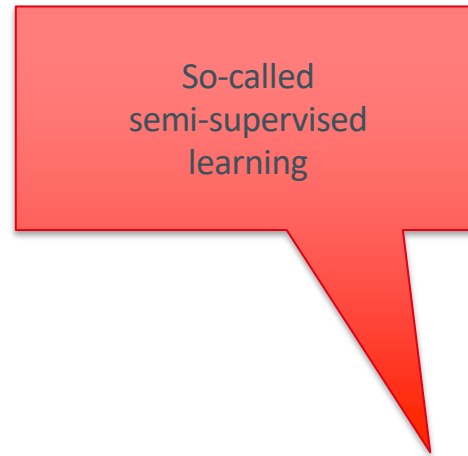
# Adapting to New Data: Clustering



seperator supervised

Classified data only

Clustering, e.g., by performing k-nearest neighbour classification  
(i.e. instance-based, not model-based)



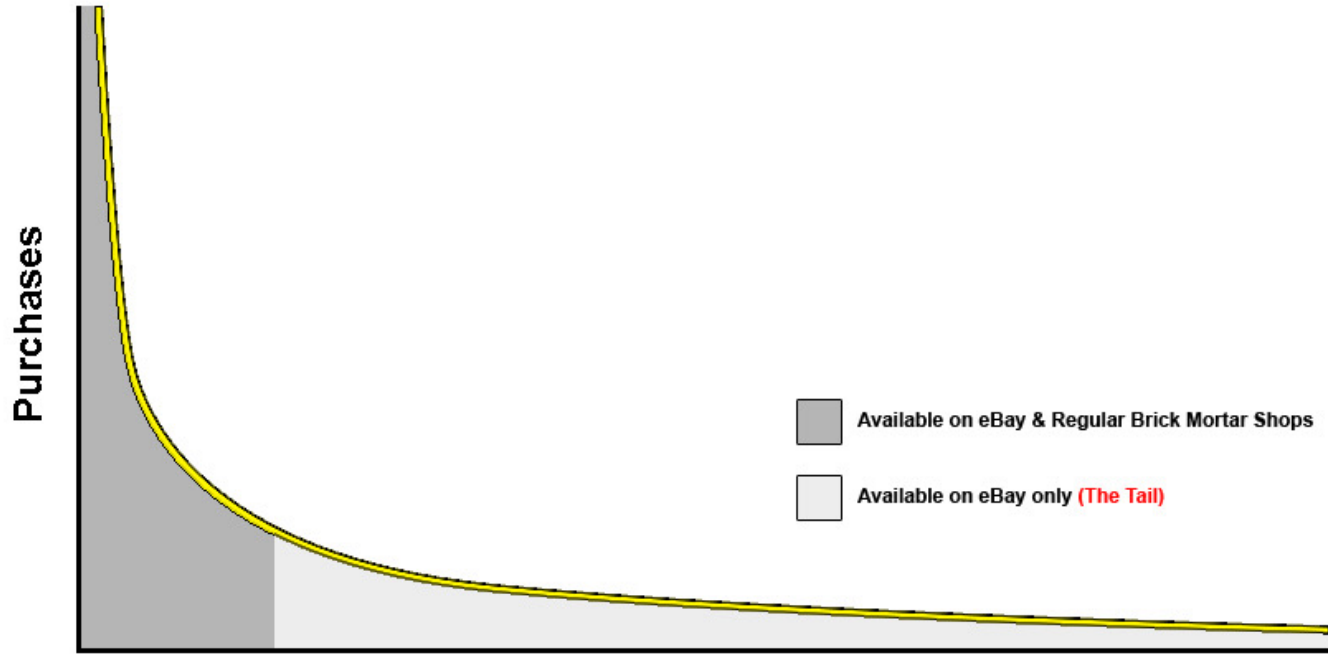
# About Diapers, Beer, and Bicycles ...



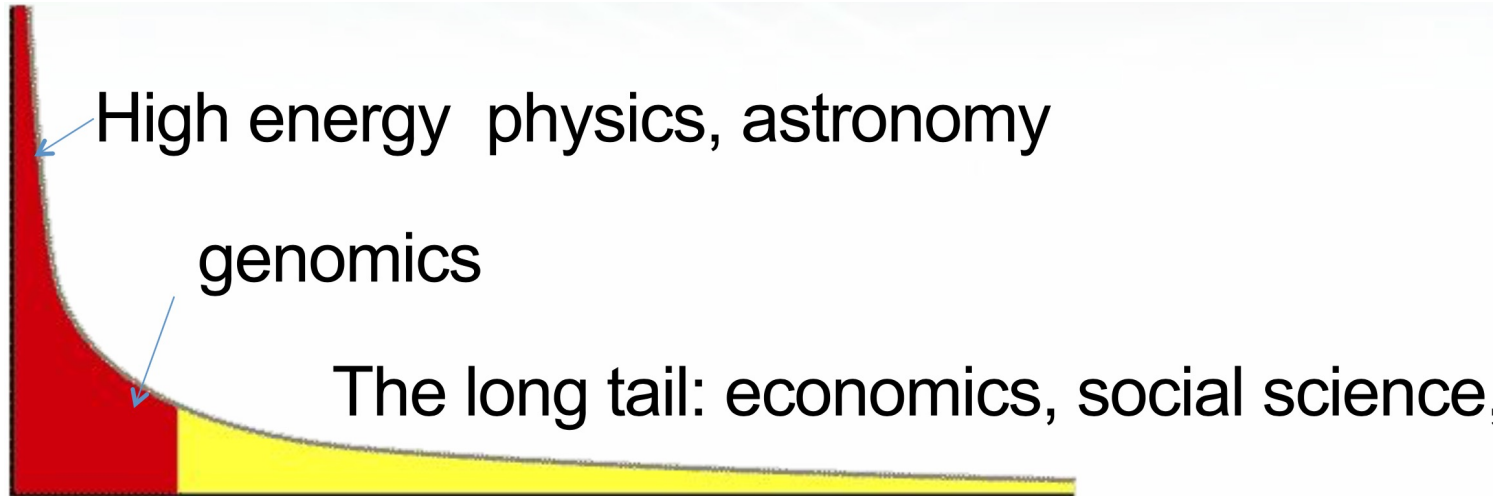
A legend... But shows the idea!  
Marcel Gehrke

# Long Tail

## The Long Tail of **ebay**.



## Useful Insights Only with Large Amounts of Data?



Wal-Mart example: Non-personalized recording of "shopping carts" via receipt.



# Question

How can user data be collected in a personalized way?

- Do you have a Payback card?

The (digital) card is presented at the checkout during the payment process while shopping. Payback customer number, date, branch, turnover and, from some Payback partners, product group codes are transmitted to Payback. The customer receives a bonus in the form of points credited to his points account on the purchase price, which varies according to the company.

As an advantage, retailers expect customers with a Payback card to prefer to shop with them in order to collect points instead of from competitors. Merchants can also create Payback coupons to support promotions and draw attention to specific product groups or dates. In addition, the partner companies will have access to the data mining information on customer behaviour from their own customers as well as from all customers of all Payback partners. In 2004, the drugstore operator dm found that Payback customers spent on average around 50 percent more money at dm than non-Payback customers. [10]

What are the central techniques and problems?

[Wikipedia]

Unsupervised Learning

---

# Association Rules

# Association Rules

- Given a set of **shopping carts**, find **rules** that **predict** the occurrence of an item (or multiple items)
- Shopping cart entry called transaction in data base jargon (Data from Online-Transaction-Processing, OLTP)

## Shopping cart transactions

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

## Examples of association rules

$\{\text{Diapers}\} \rightarrow \{\text{Beer}\},$   
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Diapers, Coke}\},$   
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

Rakesh Agrawal, Tomasz Imieliński, Arun Swami: Mining Association Rules between Sets of Items in Large Databases. In: Proc. 1993 ACM SIGMOD International Conference on Management of data, SIGMOD Record. Bd. 22, Nr. 2, Juni 1993

# Sets of Frequent Items

Given a data set **D** in the form of shopping carts, find a combination of items that often occur together

Shopping cart transactions

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

Examples of sets of frequent items:  
{Diapers, Beer},  
{Milk, Bread}  
{Beer, Bread, Milk},

# Definition: Frequent Item Sets

- Set of items  $\subseteq$  Complete set of items  $I$ 
  - E.g.: {Milk, Bread, Diapers}
- Support counter (scount)
  - Number  $scount(w)$  of the occurrence of a set of items  $w$  in the data  
(Number of the shopping carts in which a set of items occurs)
  - For example:  $scount(\{\text{Milk, Bread, Diapers}\}) = 2$
- Support
  - Ratio of the shopping carts in which the set of items occurs:  $sup(w) = scount(w) / |D|$
  - E.g.:  $sup(\{\text{Milk, Bread, Diapers}\}) = 2/5$
- Frequent item set
  - Item set with  
Support  $\geq$  minsup (threshold)

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

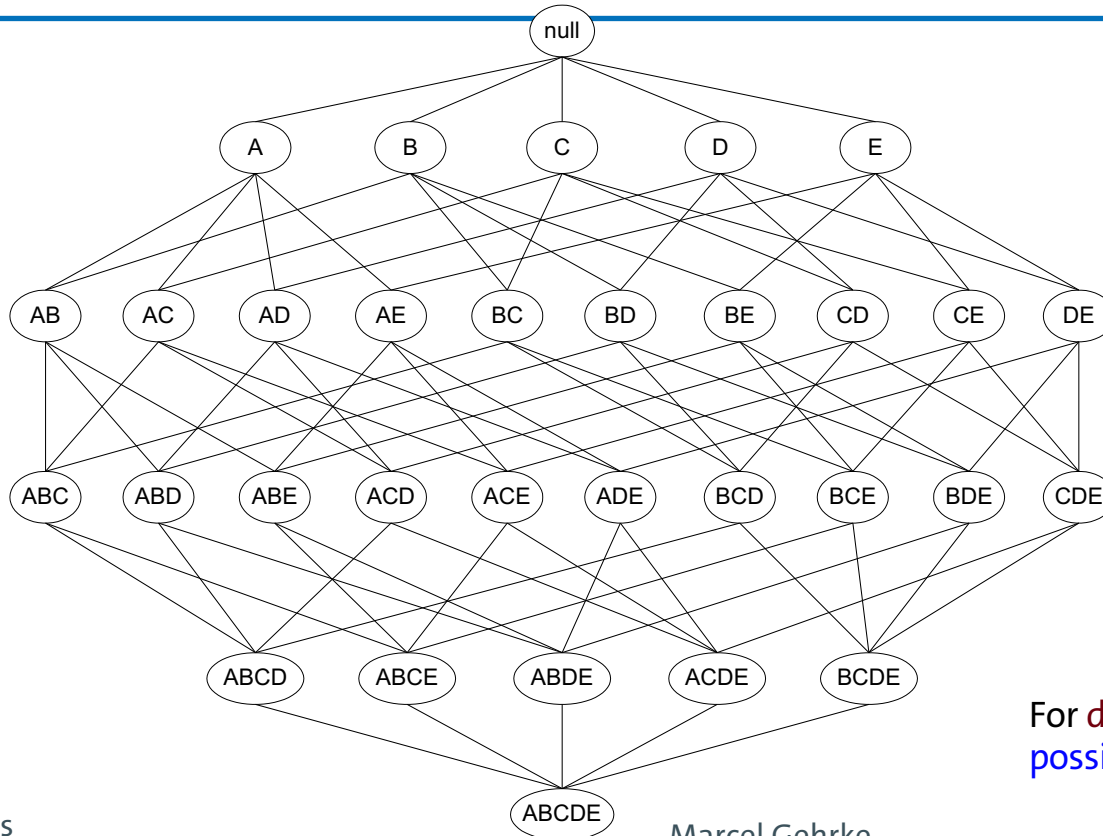
## Why find Frequent Item Sets?

- Interesting for **placement** of items  
(in the supermarket, on websites, ...)
- Frequent item sets indicate positive combinations  
(infrequent item sets hardly relevant),  
so they offer **a good glimpse** into a data set
- ...

## Identifying Frequent Item Sets

- **Task:**
  - Given a transaction database  $D$  (Shopping carts) and a threshold  $minsup$
  - Identify all frequent item sets (and their respective number in the data)
- **In other words:** Count the respective occurrences of combinations of items in the data above a threshold  $minsup$
- **Assumption:** Complete item set  $I$  known

# How many Item Sets are there?



For  $d$  items there are  $2^d$   
possible item sets

# Monotonicity of the Support

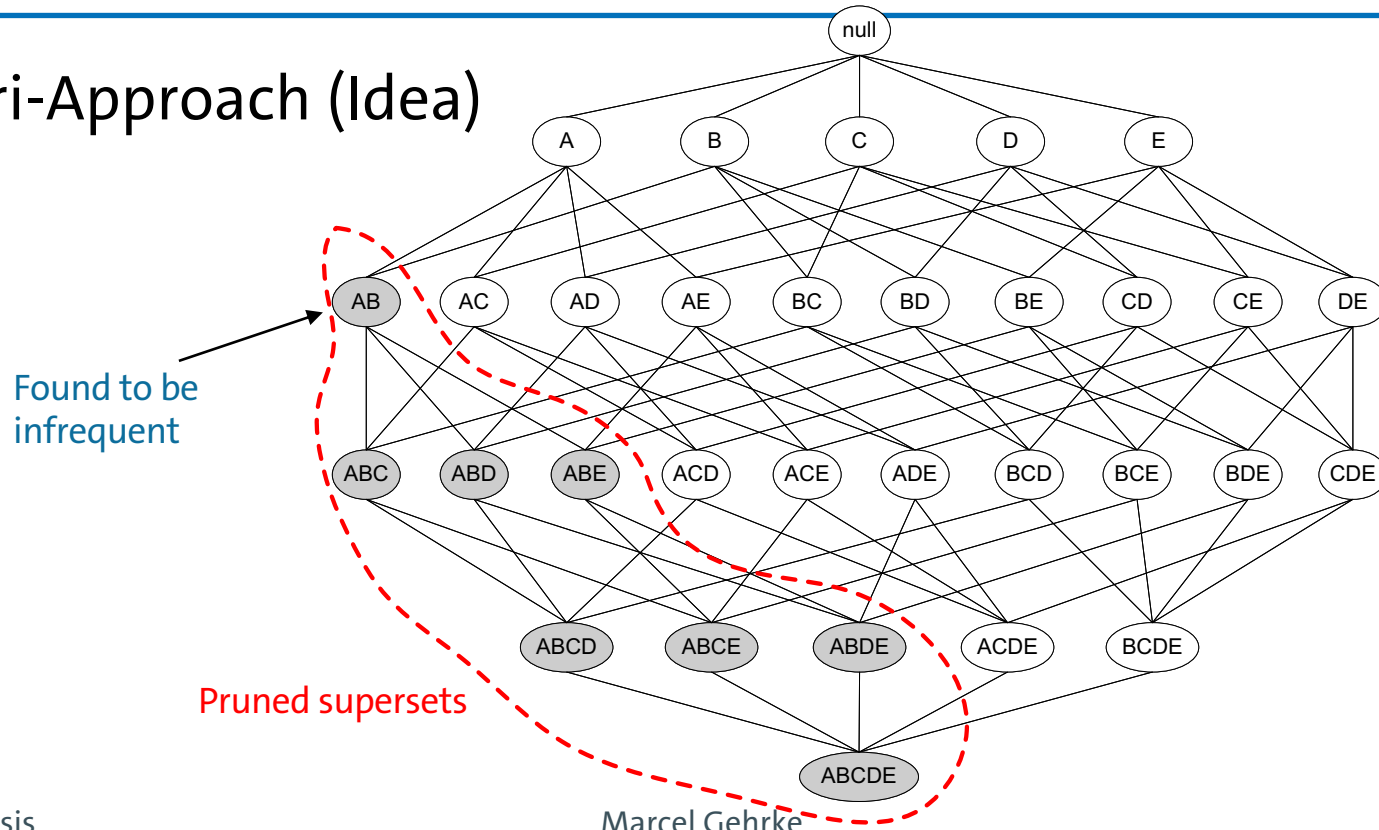
TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

$\text{sup}(\text{Bread}) > \text{sup}(\text{Bread, Beer})$

$\text{sup}(\text{Milk}) > \text{sup}(\text{Bread, Milk})$

$\text{sup}(\text{Diapers, Beer}) > \text{sup}(\text{Diapers, Beer, Coke})$

# Apriori-Approach (Idea)



# Apriori-Approach (Principle)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diapers	4
Eggs	1

minsup = 3/5

1-item sets

Item set	Count
{Bread, Milk}	3
{Bread, Beer}	2
{Bread, Diapers}	3
{Milk, Beer}	2
{Milk, Diapers}	3
{Beer, Diapers}	3

2-item sets

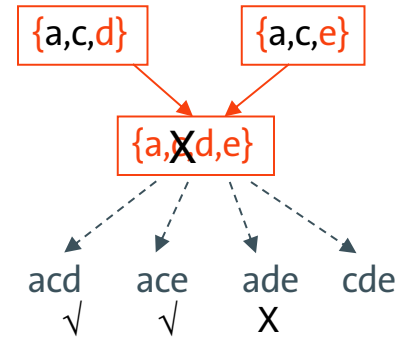
(Coke and eggs are not taken into account anymore)

3-item sets

Item set	Count
{Bread, Milk, Diapers}	2
...	...

# Apriori-Approach (Principle)

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Self-Join:  $L_3 \bowtie L_3$ 
  - $abcd$  from  $abc$  and  $abd$
  - $acde$  from  $acd$  and  $ace$
- Truncation:
  - Remove  $acde$ , as  $ade$  not in  $L_3$
- $C_4 = \{abcd\}$



## Definition: Association Rules

Let  $D$  be a database of transactions

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

- Let  $I$  be the item set of items in the DB, e.g.:  $I = \{A, B, C, D, E, F\}$
- A rule is defined by  $X \rightarrow Y$ ,  
where  $X \subset I$ ,  $Y \subset I$ ,  $X \neq \emptyset$ ,  $Y \neq \emptyset$ , and  $X \cap Y = \emptyset$ 
  - For example:  $\{B, C\} \rightarrow \{A\}$  is a rule

## Measurements for Rules $X \rightarrow Y$

- **Support**  $\text{sup}(\cdot)$ 
  - Ration of transactions, which contain  $X$  and  $Y$
- **Confidence**  $\text{conf}(\cdot)$ 
  - Measure how often items  $Y$  occur in transactions, which also contain  $X$

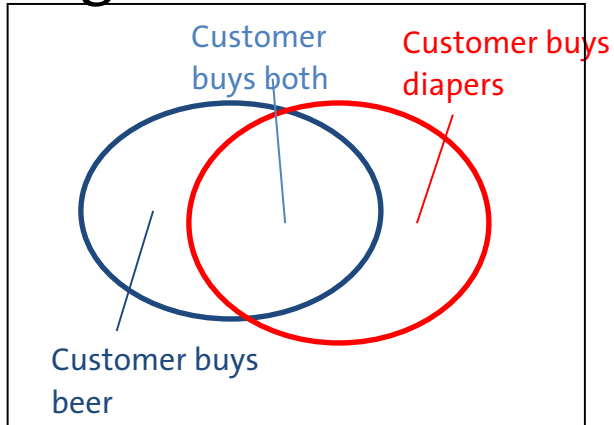
**Example:**  $\{\text{Milk, Diapers}\} \rightarrow \text{Beer}$

$$\text{sup} = \frac{\text{scout}(\text{Milk, Diapers, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$\text{conf} = \frac{\text{scout}(\text{Milk, Diapers, Beer})}{\text{scout}(\text{Milk, Diapers})} = \frac{2}{3} = 0.67$$

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

# Mining Association Rules



TID	Items
100	A,B,C
200	A,C
300	A,D
400	B,E,F

Find all rules  $r = X \rightarrow Y$  with

- $\text{sup}(r) \geq \text{minsup}$  and
- $\text{conf}(r) \geq \text{minconf}$
- **Support:** relative frequency (in %), of transactions, which contain  $X \cup Y$
- **Confidence:** Conditional relative frequency (in %) of transactions, which contain  $Y$ , if they also contain  $X$

Let the minimal support be 50% and the minimal confidence 50%:

- $A \rightarrow C$  (50%, 66.6%)
- $C \rightarrow A$  (50%, 100%)

## Brute-Force-Approach

- Consider all possible association rules
- Calculate support and confidence for each rule
- Eliminate rules whose support or confidence is less than **minsup** and **minconf** thresholds.
  
- ⇒ **Too expensive!** Combinatorial explosion

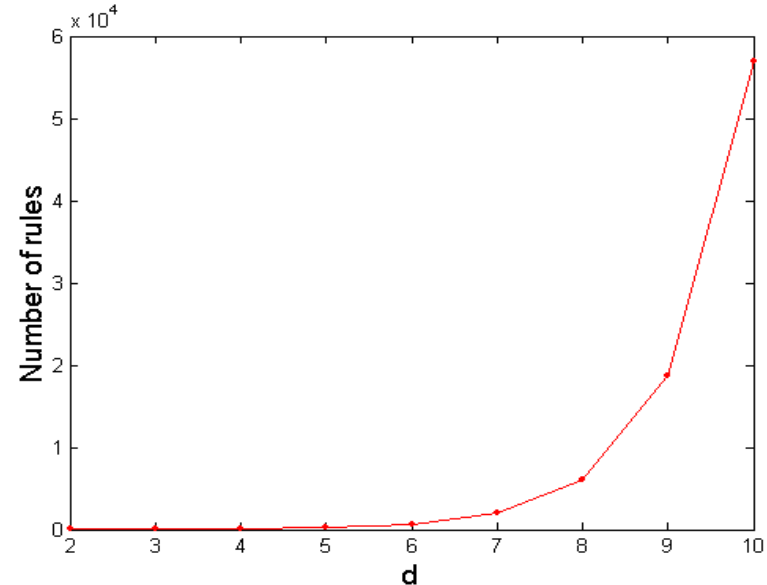
# Computational Effort

Given  $d$  items in  $I$ :

- Number of item sets:  $2^d$
- Number of association rules:

$$\sum_{k=1}^{d-1} \left[ \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$
$$= 3^d - 2^{d+1} + 1$$

If  $d=6$ ,  $R = 602$  rules



# Mining Association Rules

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

## Examples of rules:

$\{\text{Milk, Diapers}\} \rightarrow \{\text{Beer}\}$  (sup=0.4, conf=0.67)

$\{\text{Milk, Beer}\} \rightarrow \{\text{Diapers}\}$  (sup=0.4, conf=1.0)

$\{\text{Diapers, Beer}\} \rightarrow \{\text{Milk}\}$  (sup=0.4, conf=0.67)

$\{\text{Beer}\} \rightarrow \{\text{Milk, Diapers}\}$  (sup=0.4, conf=0.67)

$\{\text{Diapers}\} \rightarrow \{\text{Milk, Beer}\}$  (sup=0.4, conf=0.5)

$\{\text{Milk}\} \rightarrow \{\text{Diapers, Beer}\}$  (sup=0.4, conf=0.5)

## Observations:

Rules are binary partitions of the same item set:  $\{\text{Milk, Diapers, Beer}\}$

Rules from the same item set have the same support but different confidence

Decoupling support and confidence

## Two-step Approach

- Generate frequent item sets with  
 $\text{support} \geq \text{minsup}$
- Generate association rules  
by **binary partitioning**  
of frequent item sets, so that  
 $\text{confidence} \geq \text{minconf}$

## Rule Generation – Simple Approach

- Given the frequent item set  $X$ , find all non-empty subsets  $y \subset X$  such that  $y \rightarrow X - y$  satisfies the minimum confidence requirement

Example:  $\{A, B, C, D\}$  be a frequent item set:

$ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB$		

- If  $|X| = k$ , then there are  $2^k - 2$  candidate association rules (ignoring  $L \rightarrow \emptyset$  und  $\emptyset \rightarrow L$ )

Unsupervised Learning

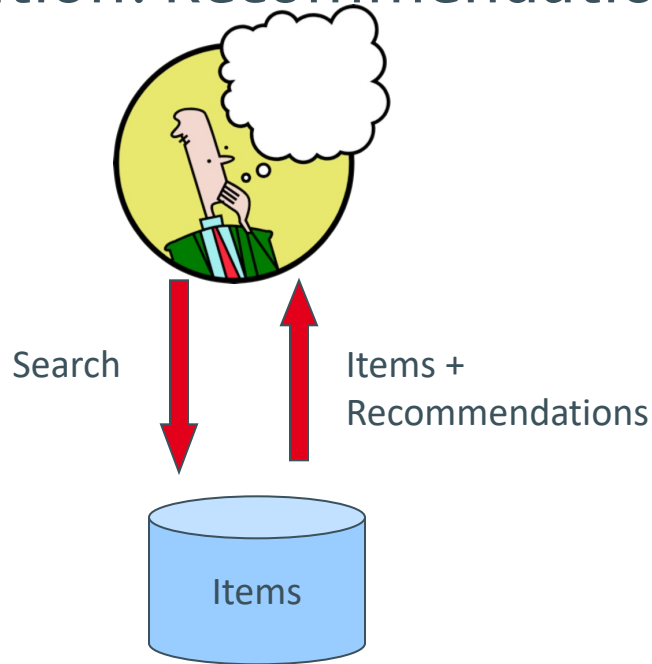
---

# Affinity Analysis

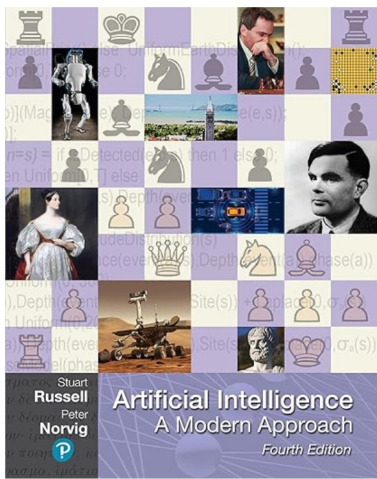
# Applications of Affinity Analysis

- "Understanding" of transaction data
- Data Mining (offline)
  - Systematic application of statistical methods on large data sets, with the aim to ...
  - ... identify new cross-connections and trends
- Generating recommendations (online)
  - Make a prediction that quantifies how much interest a user has in an object, ...
  - ... to recommend to the user exactly those objects from the set of all existing objects in which he is most likely to be interested

# Application: Recommendation Generation



- Limitation of the resource "space"
- What are "good" recommendations
  - Increase customer satisfaction
  - Increase in revenue of vendor
- Techniques
  - Use of frequent item sets
  - Use of association rules



Read sample

Follow the author



Stuart Russell

Follow

# Artificial Intelligence: A Modern Approach (Pearson Series in

## Artificial Intelligence) 4th Edition

by [Stuart Russell](#) (Author), [Peter Norvig](#) (Author)

4.5 ★★★★★ 444 ratings

Part of: [Pearson Series in Artificial Intelligence \(1 books\)](#)

[See all formats and e](#)

The most comprehensive, up-to-date introduction to the theory and practice of artificial intelligence

The long-anticipated revision of *Artificial Intelligence: A Modern Approach* explores the full breadth and depth of the field of artificial intelligence (AI). The 4th Edition brings readers up to date on the latest technologies, presents content in a more unified manner, and offers new or expanded coverage of machine learning, deep learning, transfer learning, multiagent systems, robotics, natural language processing, causality, probabilistic programming, privacy, fairness, and a safe AI.

[Report an issue with this product or seller](#)

ISBN-10



0134610997

ISBN-13



978-0134610993

Edition



4th

Publisher



Pearson

Publication date

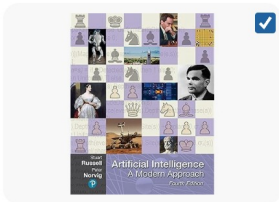


May 8, 2020

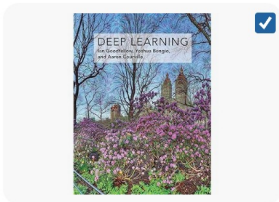
[See all details](#)

# Use of frequent item sets

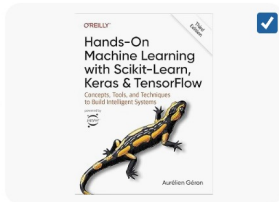
## Frequently bought together



+



+



Total price: \$280.53

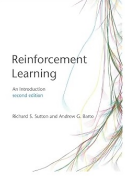
Add all 3 to Cart

[Some of these items ship sooner than the others. Show details](#)

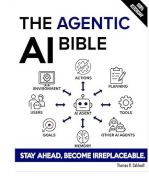
# Using Association Rules



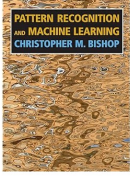
More items to explore



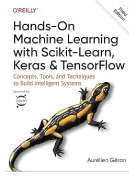
**Reinforcement Learning**  
Introduction, second edition:  
An Introduction (Adaptive Computation and...  
> Richard S. Sutton  
★★★★☆ 606  
Hardcover  
\$75.99  
\$13.49 shipping



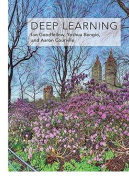
**The Agentic AI Bible: The Complete and Up-to-Date Guide to Design,...**  
> Thomas R. Caldwell  
★★★★☆ 330  
Paperback  
#1 Best Seller  
\$52.92  
\$12.72 shipping




**Pattern Recognition and Machine Learning** (Information Science and Statistics)  
> Christopher M. Bishop  
★★★★☆ 775  
Hardcover  
\$48.91  
\$13.49 shipping  
Only 3 left in stock (more o...



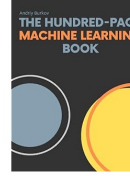
**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**  
> Aurélien Géron  
★★★★☆ 760  
Paperback  
\$46.95  
\$13.38 shipping



**Deep Learning** (Adaptive Computation and Machine Learning series)  
> Ian Goodfellow  
★★★★☆ 2,321  
Hardcover  
\$59.90  
Get it as soon as **Monday, Nov 3**



**Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python**  
> Sebastian Raschka  
★★★★☆ 476  
Paperback  
\$39.95  
\$13.38 shipping



**The Hundred-Page Machine Learning Book** (The Hundred-Page Books)  
> Andriy Burkov  
★★★★☆ 1,287  
Paperback  
\$79.94  
\$9.88 shipping

## Customers also bought or read

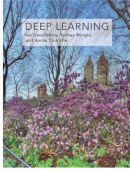
Similar books

In this series

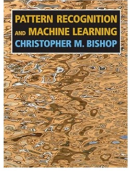
by Stuart Russell

Deep Learning

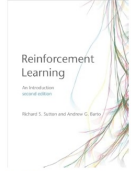
Computer



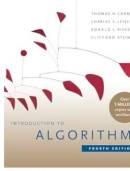
**Deep Learning** (Adaptive Computation and Machine Learning series)  
> Ian Goodfellow  
★★★★☆ 2,321  
Hardcover  
\$59.90  
Delivery **Fri, Nov 7**



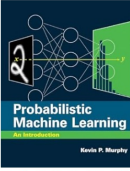
**Pattern Recognition and Machine Learning** (Information Science...  
> Christopher M. Bishop  
★★★★☆ 775  
Hardcover  
\$48.91  
Delivery **Mon, Nov 3**



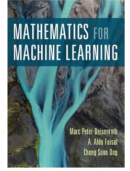
**Reinforcement Learning, second edition: An Introduction...**  
> Richard S. Sutton and Andrew G. Barto  
★★★★☆ 606  
Hardcover  
\$75.99  
Delivery **Fri, Oct 31**



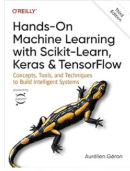
**Introduction to Algorithms, fourth edition**  
> Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford A. Stein  
★★★★☆ 731  
#1 Best Seller ...  
Hardcover  
\$85.99  
Delivery **Fri, Oct 31**



**Probabilistic Machine Learning: An Introduction**  
> Kevin P. Murphy  
★★★★☆ 228  
Hardcover  
\$83.34  
Delivery **Mon, Nov 3**



**Mathematics for Machine Learning**  
> Peter Deisenroth, A. A. Faisal, C. S. Ong  
★★★★☆ 952  
Paperback  
\$47.77  
Delivery **Fri, Oct 31**



**Hands-On Machine Learning with Scikit-Learn, Keras, and Ten...**  
> Aurélien Géron  
★★★★☆ 760  
Paperback  
\$46.95  
Delivery **Fri, Oct 31**

Unsupervised Learning

---

Recommendations

# Refinement of Recommendation Generation

**Personalisation:** Customer-specific recommendation

Estimating customer satisfaction via "utility".

- $C$  : Set of customers
- $S$ : Set of items
- **Utility function:**  $u : C \times S \rightarrow R$ 
  - $R$ : Set of reviews (total ordered set)
  - Examples: 0-5 Stars, real numbers from  $[0,1]$

# Maximizing the Utility Estimation

- **Utility function:**  $u : C \times S \rightarrow R$
- For each user  $c \in C$  determine those **items**  $s'$  from the range of goods  $S$ , which maximise the utility for user  $c$ :
- **What** is the recommendation generation problem?

$$\forall c \in C : s_c' = \arg \max_{s \in S} u(c, s)$$

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

# Central Problem

- Utility is only partially defined, i.e. not known for all elements from the  $C \times S$  space
- Utility function  $u$  has to be extrapolated

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4



# Obtaining Utility Measures

- Explicit
  - Users rate items
  - Does not work in practice, users get annoyed
- Implicit
  - Obtain measurements from user actions
    - Buying an item gives a good rating
    - Viewing details of an item shows interest in the type of item
    - What about bad reviews?

# Extrapolation of Utilities

- Key problem: Matrix  $U$  is sparse
  - Most of the items are not rated by users
  - **Extrapolation** necessary (**called filtering**)
- Approach: **How** to solve the recommendation generation problem?
  - **Content-based** filtering
    - Recommendation of items that are "similar" to the already highly rated ones:  
Choose  $u(c,s)$  like  $u(c, s')$  with  $\text{sim}(s, s')$
  - **Collaborative** filtering
    - Recommendation of items that are highly rated by "similar" users:  
Choose  $u(c,s)$  like  $u(c', s)$  with  $\text{sim}(c, c')$

# Content-based Filtering: Item Features

For each item  $s$  generate item profile  $\text{content}(s)$

Profile is set of feature values

- Text: Set of (Word, Weight)-pairs
  - Can be interpreted as a vector
- For movies:
  - Extract text from information about title, actor, director, etc.

How do you obtain weight information?

- Standard approach: TF.IDF  
(Term Frequency times Inverse Doc Frequency)

## TF.IDF

$f_{ij}$  = Number of terms  $t_i$  in document  $d_j$

$$TF_{ij} = f_{ij} / \text{Number of terms } d_j$$

$n_i$  = Number of documents in which term  $i$  occurs

$N$  = Total number of documents

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF-Measure  $w_{ij} = TF_{ij} \cdot IDF_i$

# Content-based Utility Estimation(Filtering)

- For user  $c$  use assigned items  $\text{items}(c)$ ...
- ... and determine  $\text{content}(s)$  for all  $s \in \text{items}(c)$
- Define  $\text{profile}(c)$  (e.g.) as
  - Mean of  $\text{content}(s)$  for all  $s \in \text{items}(c)$
  - (other definitions are possible)
- We obtain: Set of (Term, Weight)-pairs
  - Can be interpreted as vector  $\mathbf{w}$
- Utility function  $u(c, s)$ :
- Profile also called rating

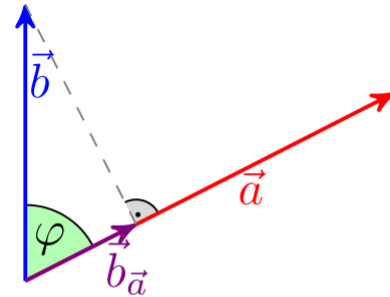
$$\begin{aligned} u(c, s) &= \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} \\ &= \frac{\sum_{i=1}^K w_{i,c} \cdot w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}} \end{aligned}$$

# Scalar Product or Dot Product

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \varphi$$

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \varphi$$

$$\cos(\varphi) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



Scalar projection of  $\vec{b}$   
on  $\vec{a}$

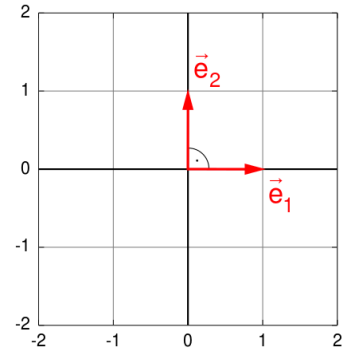
$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2$$

For the standard basis vectors  $\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $\vec{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  it holds that

$$\vec{e}_1 \cdot \vec{e}_1 = 1, \vec{e}_1 \cdot \vec{e}_2 = \vec{e}_2 \cdot \vec{e}_1 = 0 \text{ and } \vec{e}_2 \cdot \vec{e}_2 = 1$$

From this directly follows:

$$\begin{aligned} \vec{a} \cdot \vec{b} &= (a_1 \cdot \vec{e}_1 + a_2 \cdot \vec{e}_2) \cdot (b_1 \cdot \vec{e}_1 + b_2 \cdot \vec{e}_2) \\ &= a_1 b_1 \vec{e}_1 \cdot \vec{e}_1 + a_1 b_2 \vec{e}_1 \cdot \vec{e}_2 + a_2 b_1 \vec{e}_2 \cdot \vec{e}_1 + a_2 b_2 \vec{e}_2 \cdot \vec{e}_2 \\ &= a_1 b_1 + a_2 b_2 \end{aligned}$$



# Content-based Filtering Limitations

- How well is the chosen solution to the problem??
- Features not always easy to define
  - Pictures?, music?
  - Mostly surrounding or associated text is used
- Overspecialization
  - Items outside the profile are not recommended
  - People have different interests
  - Clustering?
- Recommendations for new users
  - How is the profile defined?
  - Recourse to:
    - Frequent item sets (not user-specific)
    - Association rules (not user-specific)

# User-user Collaborative Filtering

- Consider user  $c$
- Determine set  $D$  of users, whose ratings are "similar" to those of  $c$
- Estimate  $profile(c)$  given the information from  $profile(d)$  for  $d \in D$
  
- What are "similar" users?

# Similar Users: Distance Measure

Be a user review  $r_c = \text{profile}(c)$  given, then define similarity  $\text{sim}$  of users  $c_1$  and  $c_2$  as

1. Cosine Similarity

–  $\text{sim}(c_1, c_2) = \cos(rc_1, rc_2)$  or as

2. Function over reviews (Profile)

$x = rc_1$  and  $y = rc_2$ , such that

- if  $c_1$  and  $c_2$  give the same rating  $\rightarrow$  max
- Normalization of  $x$  and  $y$  required
- Commonly known as:
  - Pearson correlation coefficient or
  - Empirical correlation coefficient

Normalized values (z-Transform)

# Empirical Correlation Coefficient

$$r = \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})}{S_x} \frac{(y_i - \bar{y})}{S_y}$$

$n$ = Number of samples (items)		
$x_i$ = Sample from $x$		$y_i$ = Sample from $y$
$\bar{x}$ = Mean of $x$		$\bar{y}$ = Mean of $y$
$S_x$ = Standard deviation $x$		$S_y$ = Standard deviation $y$

## Excuse

- Actually

$$r = \frac{1}{n-1} \sum_{i=1}^n \frac{(x_i - \bar{x})}{S_x} \frac{(y_i - \bar{y})}{S_y}$$

- Why this is defined in this way is not immediately obvious, we will come back to this later.
- The same applies to variance (corrected variance):

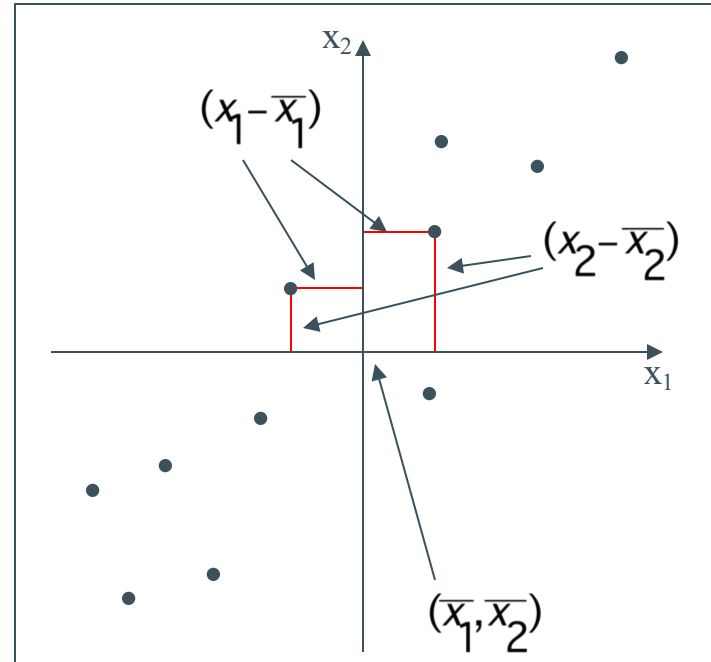
$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

- Therefore, the term "empirical correlation coefficient" is used for the uncorrected value (ditto empirical variance)

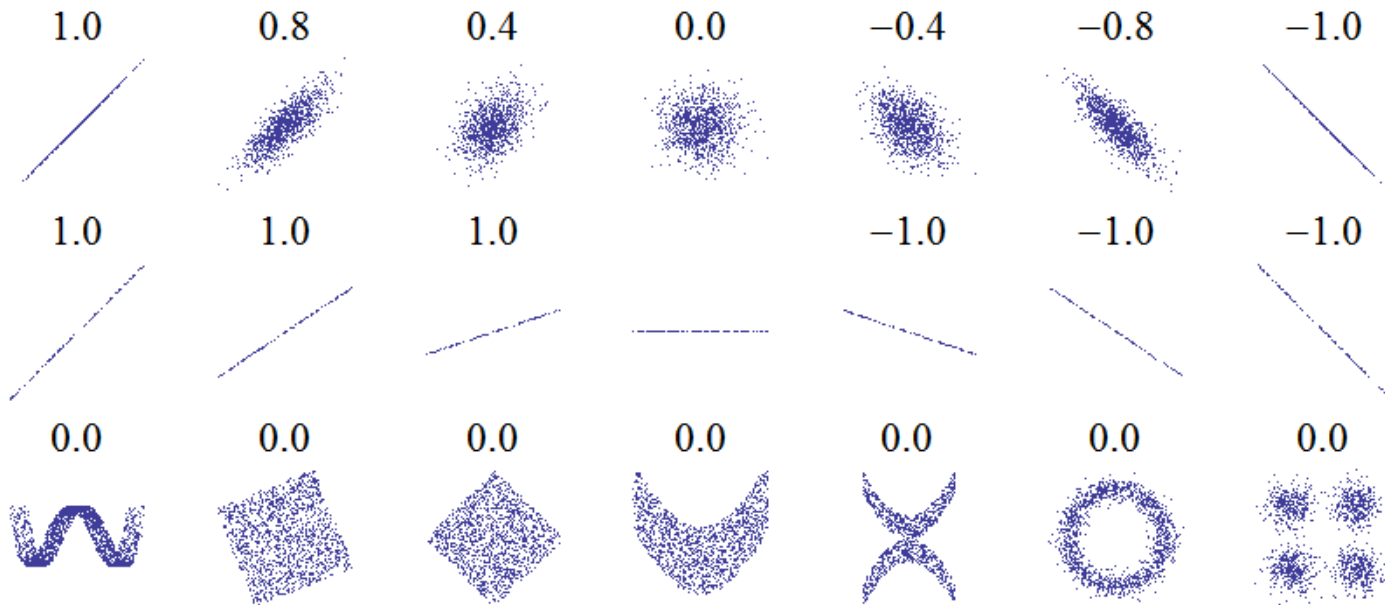
$$s_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

$$r = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}}$$

$$-1 \leq r \leq +1$$



# Illustration



Wikipedia

# Estimation of Ratings

- Let  $D$  be the set of the  $k$  most similar users w.r.t.  $c$ , which rated item  $s$
- Define estimation function for rating of  $s$ :
  - $r_{cs} = \frac{1}{k} \sum_{d \in D} r_{ds}$  or
  - $r_{cs} = (\sum_{d \in D} \text{sim}(c, d) \cdot r_{ds}) / (\sum_{d \in D} \text{sim}(c, d))$
  - ...

# Effort?

- Expensive search for most k-like users
  - Consideration of all users?
- Can hardly be done at "runtime"
  - Must be done offline
- Alternative for computing  $r_{cs}$  ?
  - Search for similar items
    - Item-Item-Collaborative Filtering
    - Otherwise, the same procedure
  - Search association rules with  $s$  as a prerequisite
  - Search frequent item set with  $s$  as an element

# Effort of Computations?

Large inventory: Many item sets to consider

- Support and confidence calculation time-consuming
- A priori algorithm helps, but effort is still high

Filtering for many users and items

- Effort is quite high

Considering only a **subset of shopping carts or users?**

- **How large** has a subset to be to be able to make statements?
- **Which** subset(s) to select?