Lecture 11 – Unsupervised Learning

Association Rules and Collaborative Filtering

Information Systems for Sustainable Society (is3) | WiSo Faculty | Univ.-Prof. Dr. Wolfgang Ketter | 12/15/21

Agenda

Association Rules

Collaborative Filtering

Association Rules: Motivation

Imagine you are a leading retailer having a database with all transactions (e.g., market basket data) – what questions could you answer with that?

For my next promotion, what product bundles should I offer?

Are there any products that are usually co-purchased with beer?

Which products should I place in a specific aisle?

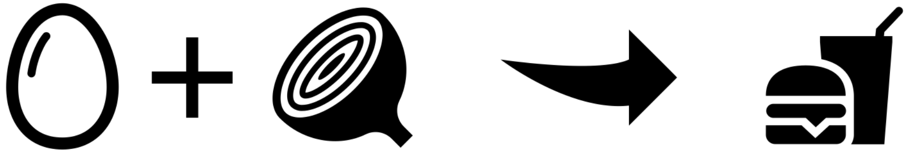
and many more…

Association Rules: Definition

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

Based on the concept of strong rules, association rules were introduced for discovering regularities between products in large-scale transaction data recorded by point of sales (POS) systems in supermarkets.

- **Example**: a supermarket found in the sales data indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placement.



Idea: Identify If-Then rules between products that are most likely to be indicators of true dependence

Antecedent → Both are sets of items that are *disjoint*. ← Consequent

Transactional Data (e.g., from receipts)

Transaction						
1	Water		Bread	Butter		
2			Bread	Butter		Chips
3	Water	Beer				Chips
4	Water	Beer			Toothbrush	
5	Water	Beer				Chips
6	Water		Bread	Butter		Chips
7	Water		Bread			Chips
8	Water					Chips
9	Water		Bread	Butter	Toothbrush	Chips
10			Bread		Toothbrush	Chips

Source: https://commons.wikimedia.org/wiki/File:18-02-16-Kassenbons-RalfR-1.jpg
Author: Ralf Roletschek/roletschek.at [GFDL 1.2 (http://www.gnu.org/licenses/old-licenses/fdl-1.2.html)]

Transforming Transactional Data into Binary Matrix Format

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

Each column represents an item

If item ”water” was purchased in transaction k, then set the entry to 1, otherwise to 0.

Universität zu Köln

Association Rules

Measure Association Rules

Generate Association Rules

Measures for Association Rules: Support

- **Support:** Measure the degree to which the data support the validity of the association rule.
- **Computation:** The support is defined as the ratio of the total number of occurrences of an itemset to the total number of records in the database:

$$\text{Support} = \frac{\textit{Number of transactions with both antecedent and consequent itemsets}}{\textit{Number of all transactions}}$$

- **Example** In the case that antecedent = {Water} and consequent = {Beer}, we obtain P({Water} AND {Beer}) = $\frac{3}{10}$

Measures for Association Rules: Confidence

- **Confidence:** Measure the degree of uncertainty of the if-then rule.
- **Computation** Confidence is defined as the ratio of the number of transactions that include all antecedents and consequent itemsets (i. e., support) to the number of transactions that include all the antecedent itemsets.

$$\text{Confidence} = \frac{\textit{Number of transactions with both antecedent and consequent itemsets}}{\textit{Number of transactions with antecedent itemset}}$$

- **Example** In the case that antecedent = {Water} and consequent = {Beer}, we obtain P({Water} AND {Beer}) = $\frac{3}{10}$, P({Water}) = $\frac{8}{10}$.
  Thus, Confidence = P({Beer}|{Water}) = $\frac{3}{8}$.

Support vs. Confidence

- One way to think of support is that it is the (estimated) probability that a transaction selected randomly from the database will contain all items in the antecedent and the consequent:

$$Support = \hat{P}(antecedent\ AND consequent)$$

- The confidence is the (estimated) conditional probability that a transaction selected randomly will include all the items in the consequent given that the transaction includes all the items in the antecedent:

$$Confidence = \frac{P(antecedent\ AND\ consequent)}{P(antecedent)} = P(consequent \mid antecedent)$$

Example: Confidence & Support

Antecedent = {Water, Beer}

Consequent = {Chips}

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

Example: Confidence & Support

Antecedent = {Water, Beer}

Consequent = {Chips}

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

Support = P({Water, Beer} AND {Chips}) = P({Water, Beer, Chips}) = $\frac{2}{10}$

Example: Confidence & Support

Antecendent = {Water, Beer}

Consequent = {Chips}

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

$$\text{Confidence} = \frac{P(\{\text{Water, Beer}\} \text{ AND } \{\text{Chips}\})}{P(\{\text{Water, Beer}\})} = \frac{\left(\frac{2}{10}\right)}{\left(\frac{3}{10}\right)} = \frac{2}{3}$$

Problem: If the support for antecedent and/or consequent is high, we can have a high value for confidence even when the antecedent and consequent are independent.

Consequent = {Water}

Antecedent = {Toothbrush}

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

$$\text{Confidence} = \frac{P(\{\text{Water}\} \text{ AND } \{\text{Toothbrush}\})}{P(\{\text{Toothbrush}\})} = \frac{(\frac{2}{10})}{(\frac{3}{10})} = \frac{2}{3}$$

Our intuition suggests that the items *water* and *toothbrush* are **independent**. However, we obtain a high confidence score (2/3). The confidence for an association rule having a very frequent consequent will be high.

To overcome the problems that the confidence score entails, consider the **Lift Ratio** measure.

- **Idea:** A better way to judge the strength of an association rule is to compare the confidence of the rule with a **benchmark** value, where we assume that the occurrence of the consequent itemset is independent of the occurrence of the antecedent.
- **Computation:** Compute the confidence score under the assumption that the consequent itemset and the antecedent itemset are independent:
  - $P(antecedent\ AND\ consequent) = P(antecedent) * P(consequent)$
  - Thus,

$$\text{Benchmark Confidence} = \frac{P(antecedent) * P(consequent)}{P(antecedent)} = P(consequent)\text{, or}$$

$$\text{Benchmark Confidence} = \frac{Number\ of\ transactions\ with\ consequent\ itemset}{Number\ of\ transactions}$$

  - Finally,

$$\text{Lift ratio} = \frac{Confidence}{Benchmark\ confidence}$$

  - Note: A lift greater than 1 suggest that there is some usefulness to the rule.

Example: Lift Ratio Measure

Consequent = {Water}

Antecedent = {Toothbrush}

Transaction	Water	Beer	Bread	Butter	Toothbrush	Chips
1	1	0	1	1	0	0
2	0	0	1	1	0	1
3	1	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	0	0	1
6	1	0	1	1	0	1
7	1	0	1	0	0	1
8	1	0	0	0	0	1
9	1	0	1	1	1	1
10	0	0	1	0	1	1

$$\text{Confidence} = \frac{\text{P(\{Water\} AND \{Toothbrush\})}}{\text{P(\{Toothbrush\})}} = \frac{(\frac{2}{10})}{(\frac{3}{10})} = \frac{2}{3} = 0.666$$

$$\text{Benchmark Confidence} = P(\{Water\}) = \frac{8}{10} = 0.8$$

$$\text{Lift Ratio} = \frac{0.666}{0.8} = 0.8333$$ **< 1 => Association Rule is not appropriate**.

Universität zu Köln

Association Rules

Measure Association Rules

Generate Association Rules

Generating Association Rules

- So far, we have just learnt how to quantify the importance of association rules.
- However, the question of which association rules should be considered still remains unanswered. For a retailer offering thousands of products, the lift ratio computation for each product combination becomes computationally intractable.
- To exemplify, the number of rules that one can generate for N items is $3^N - 2^{N+1} + 1$.
- Thus, we need to find a way to efficiently extract association rules from existing data (e.g., transaction in the database).
- Association Rule Generation can be split into two tasks:
  - Generating itemsets from a list of items
  - Generating all possible rules from the **frequent** itemsets

Generating itemsets from a list of items

Available Items

Items = {Water, Beer, Bread, Butter, Toothbrush, Chips}
# Combinations = 2^(6) – 1 = 63

(Some) combinations	Support (see table)
{Water}	8/10
{Beer}	3/10
{Bread}	6/10
{Water, Beer}	3/10
{Water, Bread}	3/10
{Beer, Bread}	0/10
{Water, Beer, Bread}	0/10

- **Candidate Identification:** The first step in association rules is to generate all the rules that would be candidates for indicating associations between items.
- **Problem:** Finding all possible combinations of items requires a long computation time that.
  - For N items, there exist 2^N-1 combinations.
- **Solution** Consider only *frequent itemsets* – itemsets that occur with higher frequency in the database.
  - Set p, and select only itemsets with support > p.
  - However, still many combinations to consider.

Efficiently searching for itemsets: Apriori algorithm

Available Items

Items = {Water, Beer, Bread, Butter, Toothbrush, Chips}
# Combinations = 2^(6) – 1 = 63
Specify Minimum Support p: 3/10

(Some) combinations	Support (see table)
{Water}	8/10
{Beer}	3/10
{Bread}	6/10
{Water, Beer}	3/10
{Water, Bread}	3/10
{Beer, Bread}	0/10
{Water, Beer, Bread}	0/10

- **Algorithm:** The key idea of the algorithm is to begin by generating frequent itemsets with just one item and recursively generate frequent itemsets with two items, three items, and so on, until we have generated frequent itemsets of all sizes.
  - Leverage the fact, that it is easy to generate itemsets with one element.
  - To generate frequent itemsets with 2 items, we use the frequent itemsets with 1 item. If an itemset with 1 element has a support value **less** then the minimum support **p**, then we do not further consider this itemset (**and all supersets that do contain this itemset**).
  - In general, generating itemsets with k items uses the frequent itemsets with k-1elements that were generated in the preceding step.

Efficiently searching for itemsets: Apriori algorithm

Available Items

Items = {Water, Beer, Bread, Butter, Toothbrush, Chips}
\# Combinations = 2^(6) – 1 = 63
Minimum Support **p**: 3/10

(Some) combinations	Support (see table)	
{Water}	8/10	
{Beer}	3/10	>= 3/10
{Bread}	6/10	
{Water, Beer}	3/10	
{Water, Bread}	3/10	
{Beer, Bread}	0/10	
{Water, Beer, Bread}	0/10	

- **Algorithm** The key idea of the algorithm is to begin by generating frequent itemsets with just one item and recursively generate frequent itemsets with two items, three items, and so on, until we have generated frequent itemsets of all sizes.
  - Leverage the fact, that it is easy to generate itemsets with one element.
  - To generate frequent itemsets with 2 items, we use the frequent itemsets with 1 item. If an itemset with 1 element has a support value **less** then the minimum support **p**, then we do not further consider this itemset (**and all supersets that do contain this itemset**).
  - In general, generating itemsets with k items uses the frequent itemsets with k-1elements that were generated in the preceding step.

Efficiently searching for itemsets: Apriori algorithm

Available Items

Items = {Water, Beer, Bread, Butter, Toothbrush, Chips}
# Combinations = 2^(6) – 1 = 63
Minimum Support **p**: 3/10

(Some) combinations	Support (see table)	
{Water}	8/10	
{Beer}	3/10	
{Bread}	6/10	
{Water, Beer}	3/10	
{Water, Bread}	3/10	>= 3/10
~~{Beer, Bread}~~	~~0/10~~	
{Water, Beer, Bread}	0/10	

- **Algorithm** The key idea of the algorithm is to begin by generating frequent itemsets with just one item and recursively generate frequent itemsets with two items, three items, and so on, until we have generated frequent itemsets of all sizes.
  - Leverage the fact, that it is easy to generate itemsets with one element.
  - To generate frequent itemsets with 2 items, we use the frequent itemsets with 1 item. If an itemset with 1 element has a support value **less** then the minimum support **p**, then we do not further consider this itemset (**and all supersets that do contain this itemset**).
  - In general, generating itemsets with k items uses the frequent itemsets with k-1elements that were generated in the preceding step.

Efficiently searching for itemsets: Apriori algorithm

Available Items

Items = {Water, Beer, Bread, Butter, Toothbrush, Chips}
# Combinations = 2^(6) – 1 = 63
Minimum Support **p**: 3/10

(Some) combinations	Support (see table)
{Water}	8/10
{Beer}	3/10
{Bread}	6/10
{Water, Beer}	3/10
{Water, Bread}	3/10
{Beer, Bread}	0/10
{Water, Beer, Bread}	0/10

NOTE: The algorithm do not consider {Water, Beer, Bread} since {Beer, Bread} was removed in the preceding step.

- **Algorithm** The key idea of the algorithm is to begin by generating frequent itemsets with just one item and recursively generate frequent itemsets with two items, three items, and so on, until we have generated frequent itemsets of all sizes.
  - Leverage the fact, that it is easy to generate itemsets with one element.
  - To generate frequent itemsets with 2 items, we use the frequent itemsets with 1 item. If an itemset with 1 element has a support value **less** than the minimum support **p**, then we do not further consider this itemset (**and all supersets that do contain this itemset**).
  - In general, generating itemsets with k items uses the frequent itemsets with k-1elements that were generated in the preceding step.

Generating Association Rules from frequent itemsets

Example itemset with 3 elements

itemset = {Water, Bread, Chips}
P({Water, Bread, Chips}) = 3/10

Resulting Association Rules

{Water} -> {Bread, Chips}
{Bread} -> {Water, Chips}
{Chips} -> {Bread, Water}
{Water, Bread} -> {Chips}
{Water, Chips} -> {Bread}
{Bread, Chips} -> {Water}
…
{Water} -> {Bread}
{Water} -> {Chips}

Objective

Compute Confidence Value for each Association Rule

- Once the frequent itemsets are generated, computing the association rules are less taxing.
- From the resulting itemsets, generate association rules. Since we significantly reduced the number of itemsets, computing the association rules becomes computationally tractable.
- Again, set a minimum confidence value c.
- Drop all association rules whose confidence values is less than the minimum confidence value c.

Agenda

Association Rules

Collaborative Filtering

Recommender Systems – Motivation

Amazon.com

Netflix

Collaborative Filtering

- A recommender system provides personalized recommendations to a user based on the user's information as well as on similar users' information (e.g., rating, clicking, watched movies, purchased products).
- Collaborative Filtering is a popular technique used by recommendation systems.
- The term collaborative filtering is based on the notions of identifying relevant items for a specific user from the very large set of items ("filtering") by considering preferences of many users ("collaboration").
- Recommendation systems help companies to sell more products (cross-selling), convert browsers to buyers and increase loyalty.

Data Format for Collaborative Filtering

	Item 1	Item 2	Item 3	…	Item P
User 1	r_1,1	r_1,2	r_1,3		r_1,P
User 2	r_2,1	r_2,2	r_2,3		r_2,P
User 3	r_3,1	r_3,2	r_3,3		r_3,P
…					
User K	r_k,1	r_k,2	r_k,3		r_k,P

- Users are denoted by
  $$U_i, \qquad i \in \{1, \dots, K\}$$
- Items are denoted by
  $$I_l, \qquad l \in \{1, \dots, P\}$$
- Ratings are denoted by r
  $$r_{i,l}, i \in \{1, \dots, K\}, l \in \{1, \dots, P\}$$
- Ratings can be either binary or numerical (e.g., ratings ranging from 1 to 5)

User-Based Collaborative Filtering

"People like you"

Watched
Bachelor
GMNT
Shark Tank

Watched
Harry Potter
?

Watched
Star Wars
Frozen
Harry Potter

- **Idea** Find users with similar preferences and recommend items that they liked but they have not purchased/watched yet.
- **Approach**
  - Find users who are most similar to the the user of interest. *This step requires choosing a distance metric to measure the similarity*.
  - Consider only the items that the user has not yet purchased, recommend the ones that are most preferred by the user's *neighbors*.

Source: Icon made by *Freepik* from www.flaticon.com

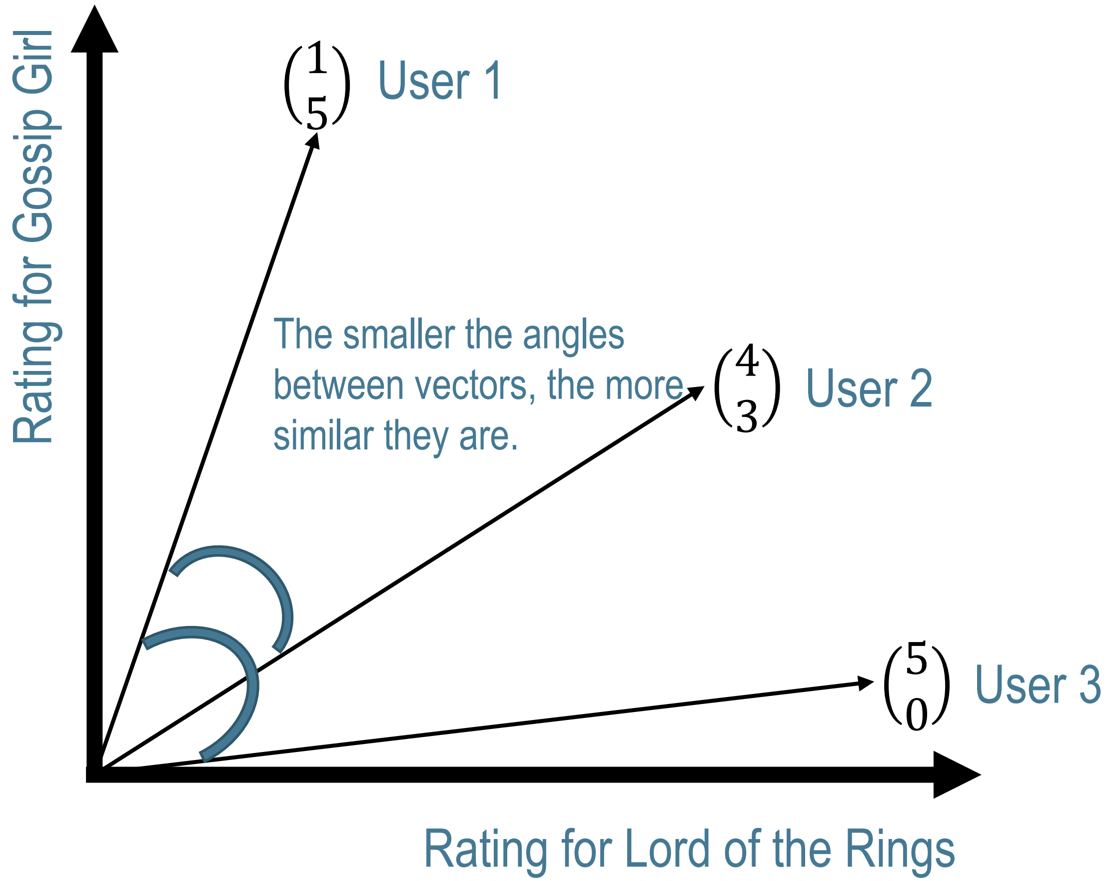
User-Based Collaborative Filtering: Pearson Correlation

- A popular approach to measure the proximity between two users is the Pearson correlation between their ratings.
- The Pearson correlation for the users U_m and U_n is defined as

$$Corr(U_m, U_n) = \frac{\sum_{i=1}^{P}(r_{m,i} - \overline{r_m})(r_{n,i} - \overline{r_n})}{\sqrt{\sum_{i=1}^{P}(r_{m,i} - \overline{r_m})^2} * \sqrt{\sum_{i=1}^{P}(r_{n,i} - \overline{r_n})^2}}$$

- We denote the ratings of items $I_1, \dots, I_P$ by user U_m as $r_{m,1}, \dots, r_{m,P}$ and their average by r_m
- The resulting correlation ranges from -1 (negative correlation) over 0 (no correlation) to 1 (positive correlation).

User-Based Collaborative Filtering: Cosine Similarity



- Another popular measure is a variant of the Pearson correlation called ***cosine similarity***.
- It differs from the correlation formula by not subtracting the means.
- The *Cosine Similarity* for the users U_m and U_n is defined as

$$Cosine\ Sim(U_m, U_n) = \frac{\sum_{i=1}^{P}(r_{m,i})(r_{n,i})}{\sqrt{\sum_{i=1}^{P}(r_{m,i})^2} * \sqrt{\sum_{i=1}^{P}(r_{n,i})^2}}$$

- The resulting cosine similarity ranges from -1 (opposite) to 1 (exactly the same).

Universität zu Köln

User-Based Collaborative Filtering: Toy Example

User

Items (movies)

	Star Wars	The Avengers	Frozen	Ghost Busters	Gone Girl
Lisa	5	4	1	1	1
Matthias	1	3	5	5	1
Bob	1	1	1	1	5
Alice	5				

Ratings (from 1 to 5)

$$\text{Corr(Lisa, Matthias)} = \frac{\left(5-\frac{12}{5}\right)(1-3)+\left(4-\frac{12}{5}\right)(3-3)+\left(1-\frac{12}{5}\right)(5-3)+\left(1-\frac{12}{5}\right)(5-3)+\left(1-\frac{12}{5}\right)(1-3)}{\sqrt{\left(5-\frac{12}{5}\right)^2+\left(4-\frac{12}{5}\right)^2+\left(1-\frac{12}{5}\right)^2+\left(1-\frac{12}{5}\right)^2+\left(1-\frac{12}{5}\right)^2} * \sqrt{(1-3)^2+(3-3)^2+(5-3)+(5-3)^2+(1-3)^2}} = \frac{-8}{3,89*4,21} = -0,488$$

$$\text{Cos Sim(Lisa, Matthias)} = \frac{(5)(1)+(4)(3)+(1)(5)+(1)(5)+(1)(1)}{\sqrt{(5)^2+(4)^2+(1)^2+(1)^2+(1)^2} * \sqrt{(1)^2+(3)^2+(5)+(5)^2+(1)^2}} = \frac{28}{6,63*7,81} = 0,5404$$

Item-Based Collaborative Filtering

	Star Wars	The Avengers	Frozen	Ghost Busters	Gone Girl
Lisa	5	4	1	1	1
Matthias	1	3	5	5	1
Bob	1	1	1	1	5
Alice	5				

- **Idea** Instead of looking for other people similar to the user, look for similar items the user liked and recommend these items.
- There are two main reasons to use item-based Collaborative Filtering:
  - When the number of users is much larger than the number of items, it is computationally cheaper to find similar items.
  - When a user expresses interest in a particular item (e.g., Amazon).

Item-Based Collaborative Filtering: Algorithm

	Lisa	Matthias	Bob
Star Wars	5	1	1
The Avengers	4	3	1
Frozen	1	5	1
Ghost Busters	1	5	1
Gone Girl	1	1	5

- **Step 1** Transpose the user-item matrix
- Note that we ignored Alice's preferences since she did not rate all movies yet.

Item-Based Collaborative Filtering: Algorithm

Cos Sim(Star Wars, The Avengers) = 0,9058

Cos Sim(Star Wars, Frozen) = 0,4074

….

Cos Sim(Ghost Busters, Gone Girl) = 0,4074

- **Step 2** Compute the similarity between all items by using one of the following distance metrics
  - Pearson Correlation
  - Cosine Similarity

Universität zu Köln

Item-Based Collaborative Filtering: Algorithm

	Star Wars	The Avengers	Frozen	Ghost Busters	Gone Girl
Star Wars	1	0,905821627	0,407407	0,407407407	0,407407
The Avengers		1	0,754851	0,754851356	0,452911
Frozen			1	1	0,407407
Ghost Busters				1	0,407407
Gone Girl					1

- **Step 3** Based on the computed similarity measures among all items, set up the so-called item-item matrix.

Advantages and Weaknesses of Collaborative Filtering

Advantages

- Easy to implement and understand.
- It provides useful recommendations, even for “long tail” items, if our database contains sufficient similar users, so that each user can find other users with similar tastes.

Weaknesses

- Relies on the availability of subjective information regarding users’ preferences.
- It cannot generate recommendations for new users, nor for new items.
- User-based collaborative filtering looks for similarity in terms of highly rated items. Unwanted items will be nearly ignored.
- User-Based collaborative filtering becomes computationally intractable for real-time responses when the number of users is much higher than the number of items.

Collaborative Filtering vs. Association Rules

- **Frequent itemsets vs. personalized recommendation**: Association rules look for frequent item combinations and will provide recommendation only for those items. In contrast, collaborative filtering provides recommendations for every item (or user) and consequently is useful for “long tail” items.
- **Transactional data vs user data**: Association rules rely on items in many basket/transactions. Collaborative Filtering provides recommendations of items based on their co-rating/purchase by even a small number of other users.
- **Binary data and ratings data**: Association rules treat items as binary data, whereas collaborative filtering can operate on either binary or numerical data.
- **Two or more items** Association rules can recommend a bundle of items at one go (also handle multiple items as input). Collaborative Filtering only compares two items at a time.

Contact

For general questions and enquiries on **research**, **teaching**, **job openings** and new **projects** refer to our website at www.is3.uni-koeln.de

For specific enquiries regarding this course contact us by sending an email to the **IS3 teaching** address at is3-teaching@wiso.uni-koeln.de

Follow us on **Twitter** at **@IS3_UniCologne** to stay up to date with recent publication and presentations of our group