Big Data
Session 8: Recommender Systems

Frank Hopfgartner
Institute for Web Science and Technologies

Intended Learning Outcomes

At the end of this lecture, you will be able to:

- Explain what **a recommender system** is and provide examples where they are used
- Explain different **recommendation types** and understand their advantages and disadvantages
- Explain how recommender systems are **evaluated**

Outline

universität koblenz
weiter:denken

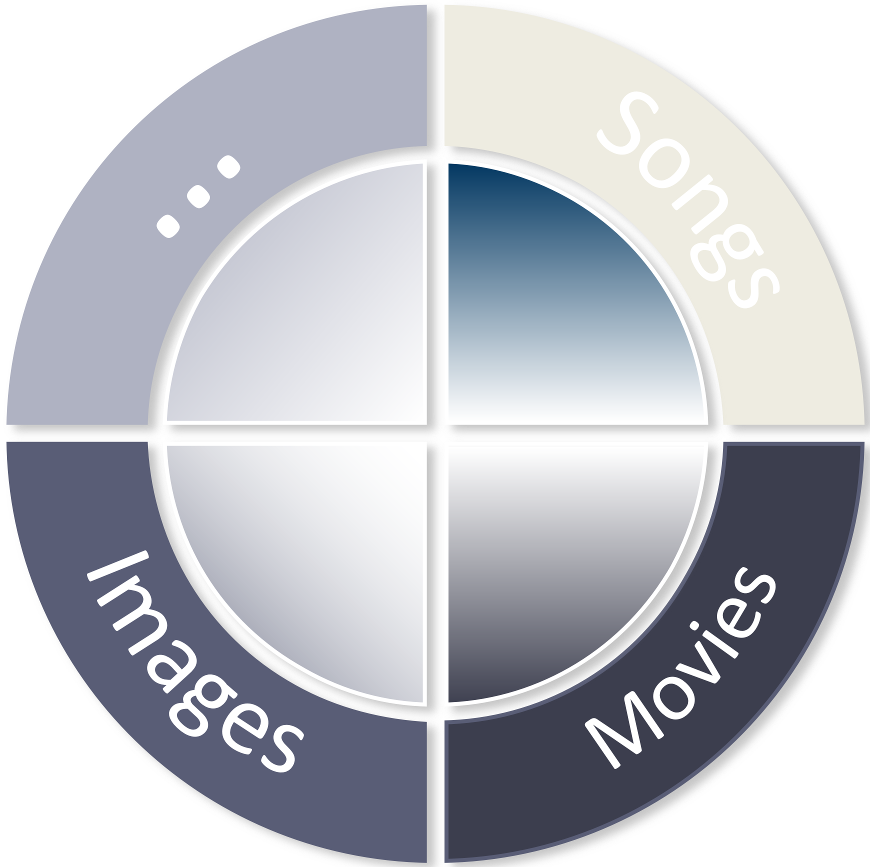
- Introduction
- Basics
- Collaborative Filtering
- Content-based Recommendations
- Distributed Recommender Systems

What are recommender systems?

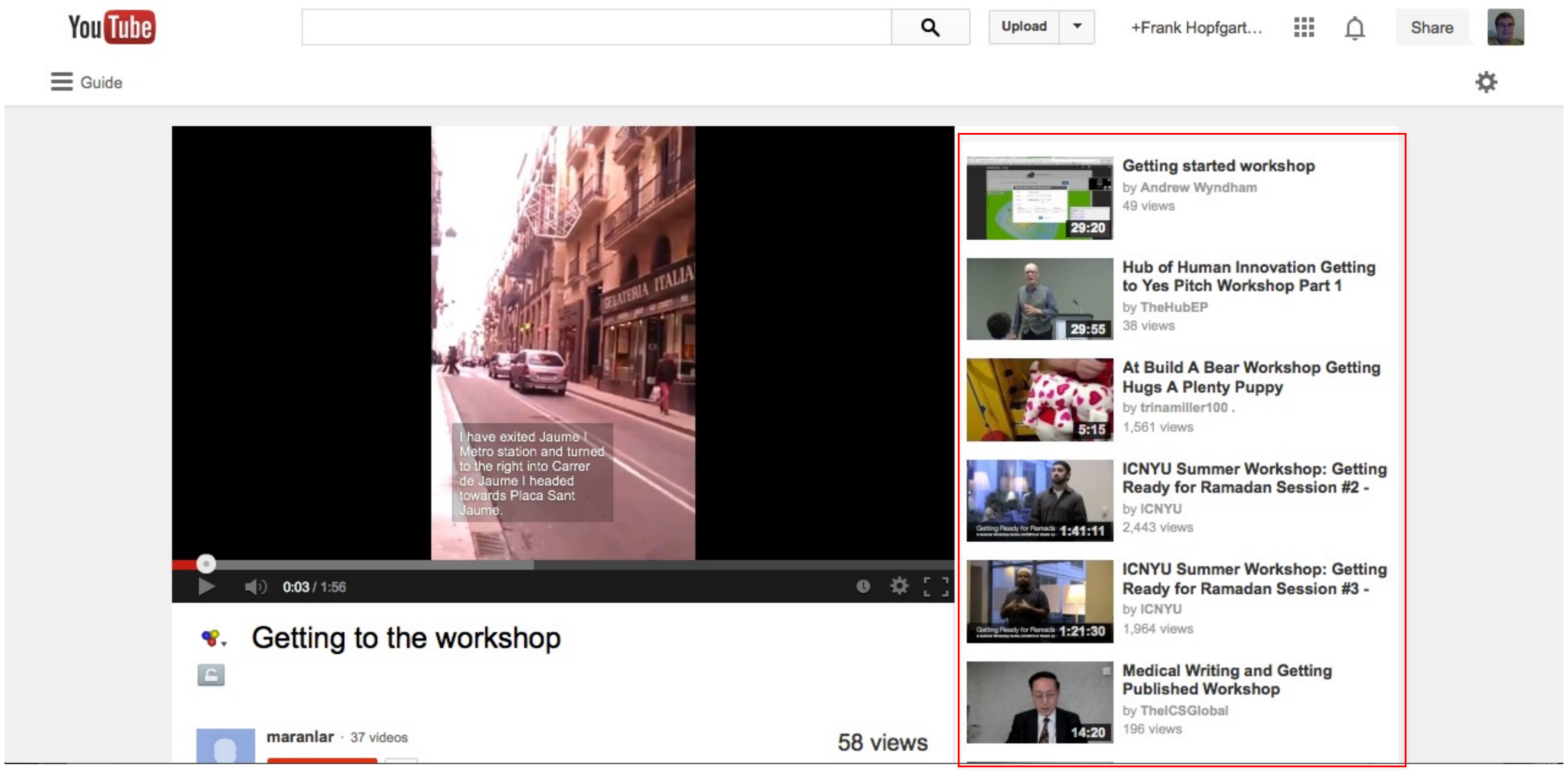
Recommender systems help users to find **items** they were not searching for.

Items?

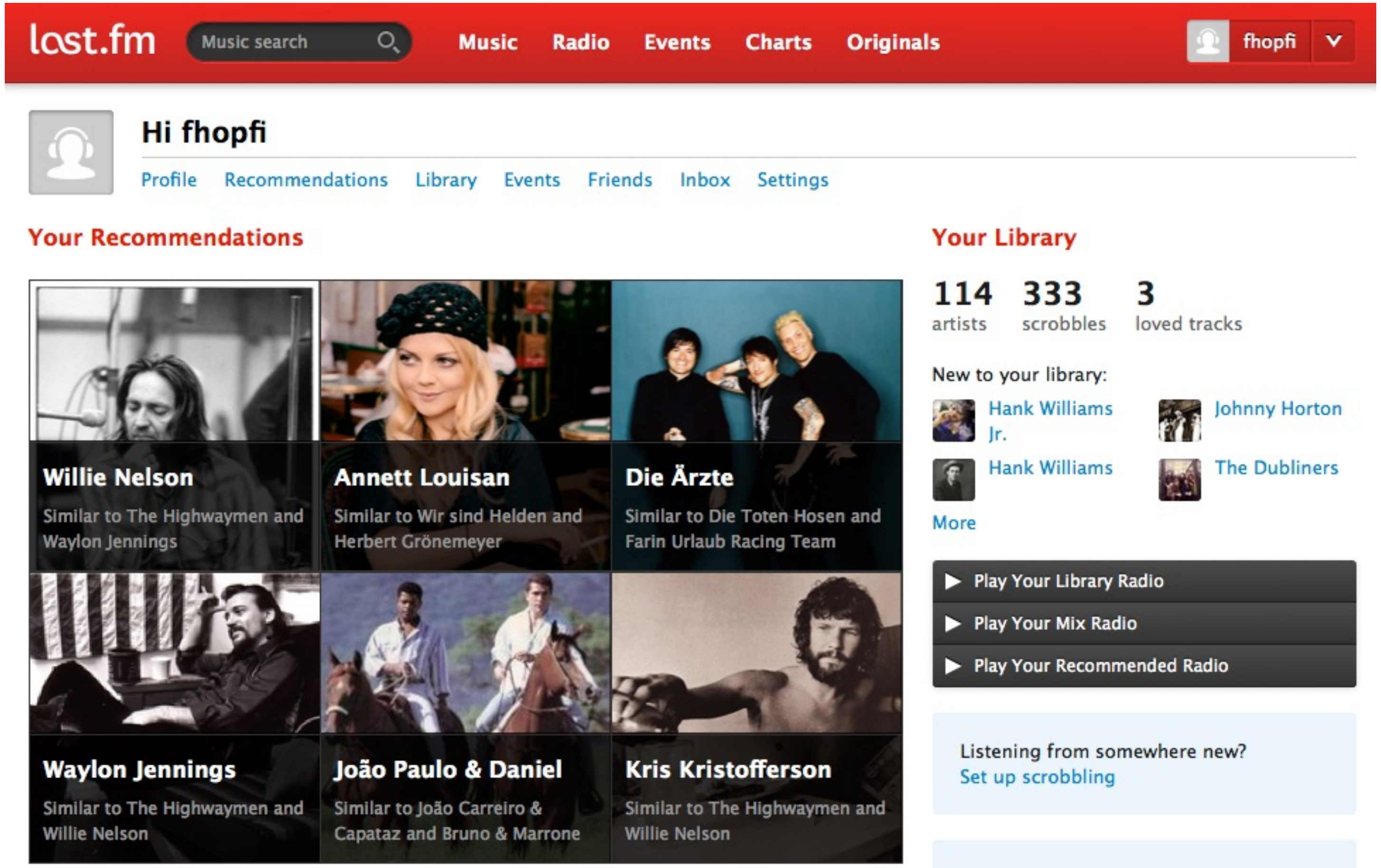
universität koblenz
weiter:denken



Example: YouTube



Example: Music



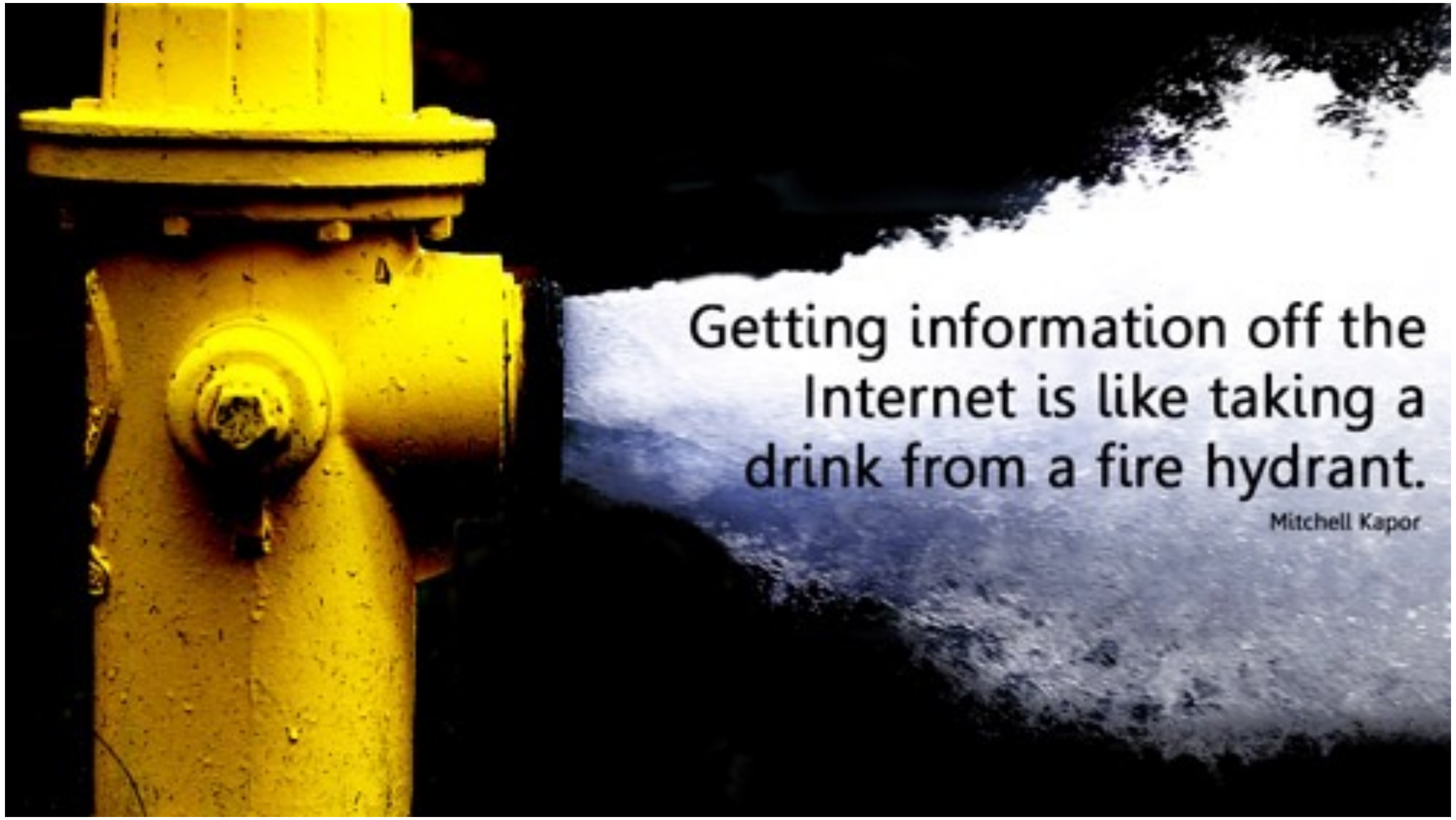
Example: News articles

Source (Image): T. Brodt of plista.com

And many others…

- Apartments
- Routes
- Clothes
- Shoes
- Restaurants
- Cafés
- Travels
- Scientific publications
- Programmers
- Craftsmen
- Grocery products
- Partners on online dating sites
- Friends in social networks
- Courses (E-Learning)
- …

Why do we need recommender systems?



RecSys types

Recommender Engine
Create a list of recommended products (items) from a larger set of products (items).

Recommender Engine

items	score
i1	0.95
i2	0.94
i3	0.89
…	…

Recommendations

RecSys types

User Profile & Context

Personalised Recommendation
System requires information about user to create personalised recommendations.

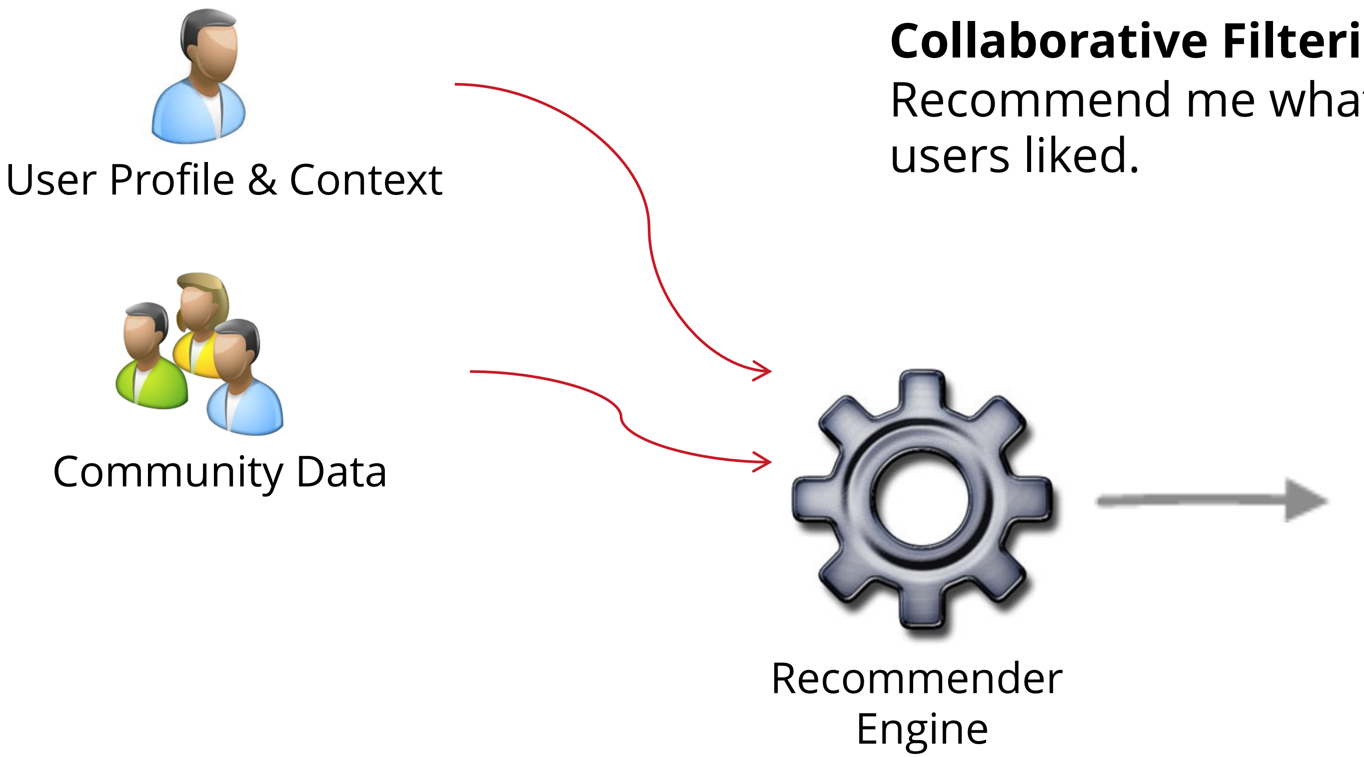
Recommender Engine

items	score
i1	0.95
i2	0.94
i3	0.89
…	…

Recommendations

RecSys types

Collaborative Filtering Systems
Recommend me what similar users liked.



items	score
i1	0.95
i2	0.94
i3	0.89
...	...

Recommendations

Y. Koren, S. Rendle, and R. Bell. Advances in Collaborative Filtering, Recommendation Systems Handbook, pp. 91-142, 2021.

Collaborative Filtering

universität koblenz
weiter:denken

- “word-of-mouth“

  recommendation
- Personalised

  recommendations based on

  user preferences
  - Own preferences
  - Others with similar taste

Collaborative Filtering

- **Advantages**
  - No information about products required
  - Expensive data enrichment not necessary
- **Disadvantages**
  - No strategy to recommend similar items of a highly rated item
  - Cold start problem

RecSys types

Content-based systems
Recommend me products that are similar to those that I like.

User Profile & Context

Title	Genre	Actors	...
...	...	...	...

Product descriptions

Recommender Engine

items	score
i1	0.95
i2	0.94
i3	0.89
...	...

Recommendations

C. Musto, M. de Gemmis, P. Lops, F. Narducci, G. Semeraro. Semantics and Content-based Recommendations. Recommender Systems Handbook, pp. 251-298, 2021.

Content-based recommendations

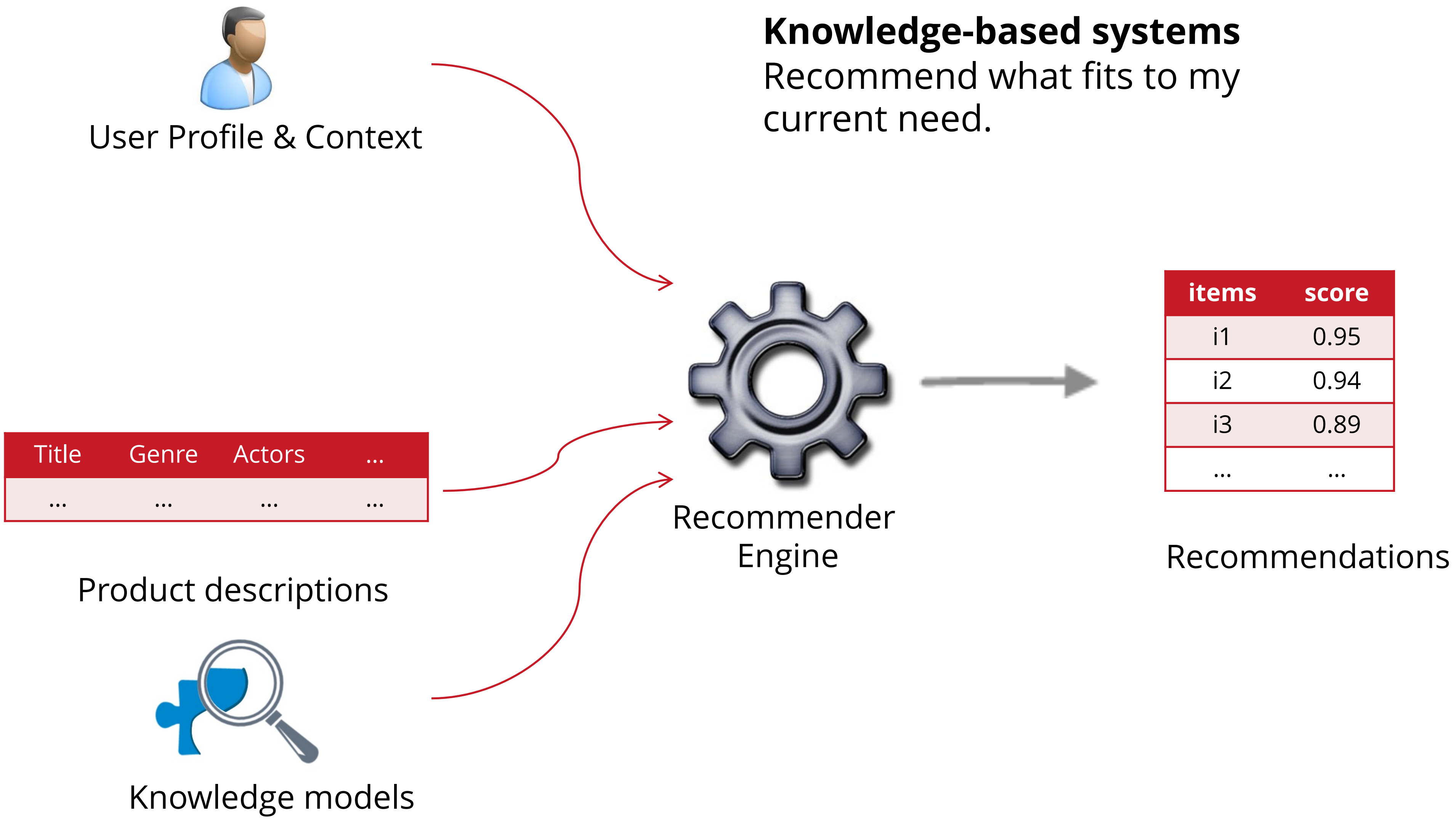
- Recommend most relevant items based on **content features**
- **Item profile** is a collection of all metadata fields describing this item
- **User profile** is an aggregation of all item profiles of a user

Content-based recommendations

- Advantages
  - Independent from user base
  - New items can directly be recommended
  - High transparency
- Disadvantages
  - Maintenance and data cleaning costly
  - “over specialisation“
  - Cold start problem for new users

RecSys types

Knowledge-based systems
Recommend what fits to my current need.



Title	Genre	Actors	…
…	…	…	…

items	score
i1	0.95
i2	0.94
i3	0.89
…	…

Knowledge-based recommendations

universität koblenz
weiter:denken

- Time span plays an important role
  - five-year-old ratings for computers
  - News articles about the World Cup
  - user lifestyle or family situation changes

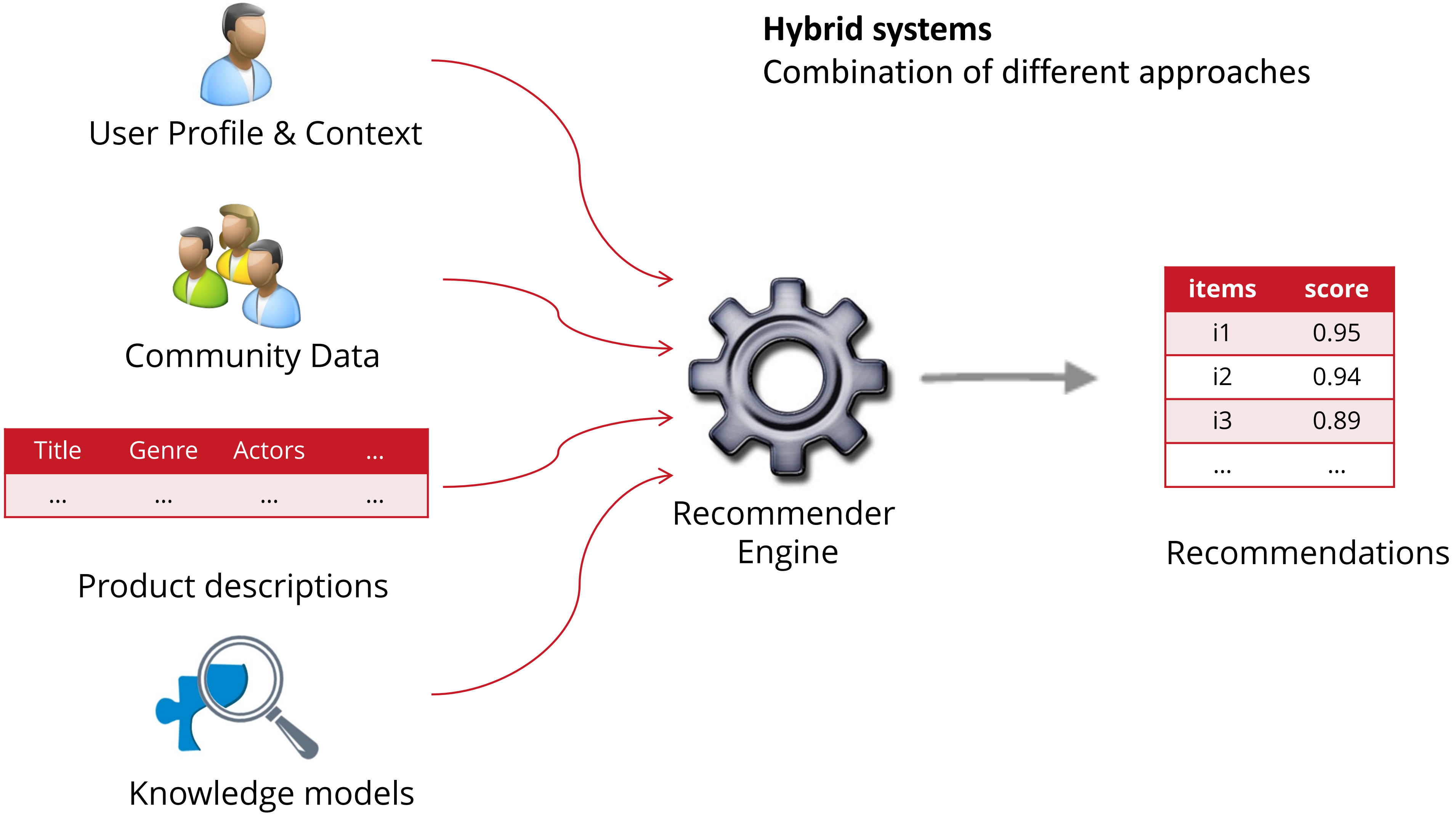
- Customers want to explicitly define their requirements
  - "the color of the jersey should be orange"

Knowledge-based recommendations

- **Advantage**
  - Allow to narrow down recommendations
- **Disadvantages**
  - cost of knowledge acquisition
    - from domain experts
    - from users
    - from web resources
  - accuracy of preference models
    - very fine granular preference models require many interaction cycles

RecSys types

Hybrid systems
Combination of different approaches



Title	Genre	Actors	…
…	…	…	…

items	score
i1	0.95
i2	0.94
i3	0.89
…	…

Hybrid systems

universität koblenz
weiter:denken

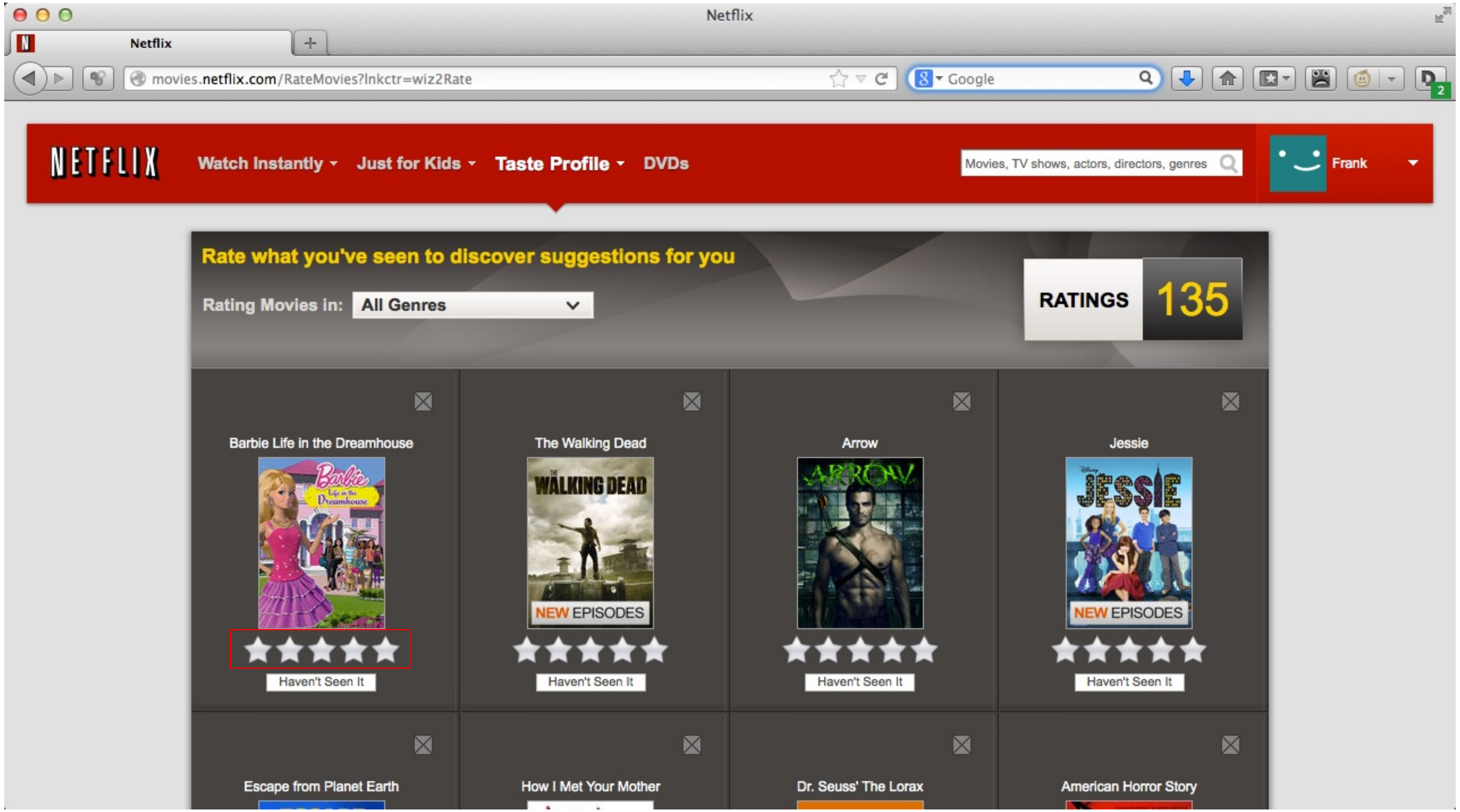
- Typical questions:
  - Which techniques should be combined?
  - How can we process recommendations of different systems?

- Advantages and Disadvantages
  - Derive from individual techniques

Outline

- Introduction
- **Basics**
  - **Example**
  - **Terminology**
  - **Recommendation task**
  - **Rating function**
  - **Evaluation**
- Collaborative Filtering
- Content-based Recommendations
- Distributed Recommender Systems

Example



Netflix
movies.netflix.com/RateMovies?lnkctr=wiz2Rate
Google
NETFLIX
Watch Instantly
Just for Kids
Taste Profile
DVDs
Movies, TV shows, actors, directors, genres
Frank
Rate what you've seen to discover suggestions for you
Rating Movies in:
All Genres
RATINGS
135
Barbie Life in the Dreamhouse
The Walking Dead
Arrow
Jessie
NEW EPISODES
NEW EPISODES
Haven't Seen It
Haven't Seen It
Haven't Seen It
Haven't Seen It
Escape from Planet Earth
How I Met Your Mother
Dr. Seuss' The Lorax
American Horror Story

Example

universität koblenz
weiter:denken

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	★★★★★	☆☆☆☆☆	★☆☆☆☆	★★★★☆	★★☆☆☆
Lea	☆☆☆☆☆	★★☆☆☆	★★★★★	☆☆☆☆☆	★★★★★
Max	★☆☆☆☆	☆☆☆☆☆	★★★★☆	★★★★★	★★★★☆
Sara	☆☆☆☆☆	★★★★☆	★★☆☆☆	★★★★★	☆☆☆☆☆

Ratings:

- Users can rate movies

Rating scale:

- 1 – 5 stars; from Flop (1 star) to Top (5 stars)
- 0 stars: Movie has not been rated yet

Note:

- other interpretations possible such as “useless” to “very helpful”
- other scales possible, such as 1-10 stars, like/dislike, etc.

Example

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	★★★★★	☆☆☆☆☆	★☆☆☆☆	★★★★☆	★★☆☆☆
Lea	☆☆☆☆☆	★★☆☆☆	★★★★★	☆☆☆☆☆	★★★★★
Max	★☆☆☆☆	☆☆☆☆☆	★★★★☆	★★★★★	★★★★☆
Sara	☆☆☆☆☆	★★★★☆	★★☆☆☆	★★★★★	☆☆☆☆☆

Question:

- Which movies might be interesting for Lea?

Possible answer: “The Matrix“, because ...

- Max seems to have a similar taste as Lea
- Max likes “The Matrix“

Example

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	★★★★★	☆☆☆☆☆	★☆☆☆☆	★★★★☆	★★☆☆☆
Lea	☆☆☆☆☆	★★☆☆☆	★★★★★	☆☆☆☆☆	★★★★★
Max	★☆☆☆☆	☆☆☆☆☆	★★★★☆	★★★★★	★★★★☆
Sara	☆☆☆☆☆	★★★★☆	★★☆☆☆	★★★★★	☆☆☆☆☆

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	5		1	4	2
Lea		2	5		5
Max	1		4	5	4
Sara		4	2	5	

Example

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	5		1	4	2
Lea		2	5		5
Max	1		4	5	4
Sara		4	2	5	

	i_1	i_2	i_3	i_4	i_5
u_1	5		1	4	2
u_2		2	5		5
u_3	1		4	5	4
u_4		4	2	5	

Terminology

universität koblenz
weiter:denken

	i_1	i_2	i_3	i_4	i_5	…	i_m
u_1	r_{11}		r_{13}	r_{14}	r_{15}		
u_2		r_{22}	r_{23}		r_{25}		
u_3	r_{31}		r_{33}	r_{34}	r_{35}		r_{3m}
u_4		r_{42}	r_{43}	r_{44}			
…							
u_n	r_{n1}				r_{n5}		r_{nm}

users $U = \{u_1, u_2, \ldots, u_n\}$

items $I = \{i_1, i_2, \ldots, i_m\}$

scores $S = \{s_1, \ldots, s_k\}$ Example.: $S = \{1, .., 5\}$; $S = \{\text{like, dislike}\}$

ratings $R = (r_{ij})$ Incomplete matrix with elements r_{ij} of S

Recommendation task

- **Given:**

  Users U = $\{u_1, u_2, \ldots, u_n\}$

  Items I = $\{i_1, i_2, \ldots, i_m\}$

  Ratings R = (r_{ij})

- **Problem:**
  - Which items are interesting for the user?

- Recommendation task:
  - Learn rating function $f: U \times I \rightarrow S$
    which accurately predicts ratings $f(u, i) = r_{ui}$ of items *i* for User *u*

  - Recommend User *u* new items *i*, for which $f(u, i)$ is large.

Recommendation task

- **Given:**

Users	U	=	$\{u_1, u_2, \ldots, u_n\}$
Items	I	=	$\{i_1, i_2, \ldots, i_m\}$
Ratings	R	=	(r_{ij})

- **Problem:**
  - Which items are interesting for the user?

- Recommendation task:
  - Learn rating function $f: U \times I \rightarrow S$
    which accurately predicts ratings *f(u, i)* = r_{ui} of items *i* for User *u*
  - Recommend User *u* new items *i*, for which *f(u, i)* is large.

What does learning mean?

Recommendation task

universität koblenz
weiter:denken

- **Given:**

  Users $U = \{u_1, u_2, \ldots, u_n\}$

  Items $I = \{i_1, i_2, \ldots, i_m\}$

  Ratings $R = (r_{ij})$

- **Problem:**
  - Which items are interesting for the user?

- Recommendation task:
  - Learn rating function $f: U \times I \rightarrow S$
    which accurately predicts ratings $f(u, i) = r_{ui}$ of items i for User u
  - Recommend User u new items i, for which $f(u, i)$ is large.

What does learning mean?

What does accurate mean?

Rating function

Note: We will see later how to get a rating function.

- Given:
  - Rating function $f: U \times I \to S$

- Problem:
  - How can *f* be used to recommend “interesting“ items to User *u*?

- Main principles:
  - Notation: Let I_u be the set of all items rated by User *u*
  - Compute *f(u,i)* for all items $i \in I \setminus I_u$ that User *u* did not rate yet
  - Sort items $i \in I \setminus I_u$ in descending order respectively *f(u,i)*
  - Recommend User *u* the first *N* items

Rating function

- Example for User u = Lea
  - Random-based rating function $f: U \times I \to S$
  - Recommend User u an item $i \in I \setminus I_u$ with maximum value *f(u,i)*

Note: Function f is not very useful but easy enough to explain the principle.

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea	?	2	5	?	5
Max			4	5	4
Sara		4	2	5	

Rating function

- Example for User u = Lea
  - Random-based rating function $f: U \times I \rightarrow S$
  - Recommend User *u* an item $i \in I \setminus I_u$ with maximum value *f(u,i)*

Compute f(u,i)

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea	**5**	2	5	**1**	5
Max			4	5	4
Sara		4	2	5	

Rating function

- Example for User u = Lea
  - Random-based rating function $f: U \times I \rightarrow S$
  - Recommend User *u* an item $i \in I \setminus I_u$ with maximum value *f(u,i)*

Sort:
1. Pulp Fiction (5)
2. The Matrix (1)

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea	**5**	2	5	**1**	5
Max			4	5	4
Sara		4	2	5	

Rating function

- Example for User u = Lea
  - Random-based rating function $f: U \times I \rightarrow S$
  - Recommend User *u* an item $i \in I \setminus I_u$ with maximum value *f(u,i)*

Recommend Pulp Fiction

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea	**5**	2	5	**1**	5
Max			4	5	4
Sara		4	2	5	

Evaluation

- **Given:**
  - Rating function $f: U \times I \rightarrow S$
- **Problem:**
  - How can we measure the quality of rating function *f*?
- **Root-Mean-Squared-Error (RMSE):**
  - Quantified discrepancy between prediction *f(u,i)* and rating r_{ui}
  - Commonly used measurement since Netflix-Challenge
  - Definition:

$$RMSE(f) = \sqrt{\frac{1}{|R_{test}|} \sum_{r_{ui} \in R_{test}} \left(f(u,i) - r_{ui} \right)^2}$$

Evaluation Example

x – Prediction f(u, i)
(x) – Rating of the users

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4	**3** (5)	1	4	2
Lea	**5** (1)	2	5	**1** (3)	5
Max	**4** (2)	**2** (2)	4	5	4
Sara	**3** (5)	4	2	5	**2** (2)

$$RMSE(f) = \sqrt{\frac{1}{7}\Big(\underbrace{(3-5)^2}_{Ben} + \underbrace{(5-1)^2 + (1-3)^2}_{Lea} + \underbrace{(4-2)^2 + (2-2)^2}_{Max} + \underbrace{(3-5)^2 + (2-2)^2}_{Sara}\Big)}$$

$$= \sqrt{\frac{1}{7}\Big(4+16+4+4+0+4+0\Big)} = \sqrt{\frac{32}{7}}$$

$$\approx 2.14$$

Question

How can we measure the quality of rating function f if we don't know the actual rating?

Evaluation

- Given:
  - Incomplete rating matrix $R = (r_{ij})$
- Problem:
  - How can we measure the quality of rating function f, if we don't know the actual rating?
- Main principle
  - Split ratings $R = (r_{ij})$ in two subsets
    - Training set R_{train}, to construct rating function f (learning)
    - Test set R_{test}, to evaluate rating function f (e.g., with RMSE)
  - Compute $f(u,i)$ for all pairs (u,i), that have one r_{ui} in R_{test}
  - Compute $RMSE(f)$ over all ratings r_{ui} of test set R_{test}

Evaluation

- Example
  - Chose time point t_0
  - Classifiy ratings before t_0 as training set
  - Classifiy ratings after t_0 as test set

Training set R_{train}	Test set R_{test}

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea		2	5		5
Max			4	5	4
Sara		4	2	5	

Evaluation

Training set R_{train}

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben					
Lea					
Max				5	4
Sara		4	2	5	

Test set R_{test}

	Pulp Fiction	Inception	Star Trek	The Matrix	Star Wars
Ben	4		1	4	2
Lea		2	5		5
Max			4		
Sara					

Outline

- Introduction
- Basics
- **Collaborative Filtering**
  - **Neighbourhood-based recommendations**
    - **User-based**
    - **Item-based**
  - Matrix factorisation
  - Graph-based algorithms
- Content-based Recommendations
- Distributed Recommender Systems

User-based recommendations

- **Question:**
  - How will I rate the movie “Pulp Fiction”?
- **Idea:**
  - I like movies that my “friends“ like
- **Prediction:**
  - Chose *k* users with a similar taste (k-nearest neighbours)
  - Rating for the movie is predicted based on the ratings of my k nearest neighbours

User-based recommendations
Example 1

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	4		1	4	2
Lea		2	5		5
Max			4	5	4
Sara		4	2	5	

User-based recommendations
Example 1

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	4		1	4	2
Lea		2	5		5
Max			4	5	4
Sara		4	2	5	

Question: How would Ben rate the movie “Inception“?

User-based recommendations
Example 1

universität koblenz
weiter:denken

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	4		1	4	2
Lea		2	5		5
Max			4	5	4
Sara		4	2	5	

Question: How would Ben rate the movie “Inception“?
Prediction:

1. Determine nearest neighbour of Ben (k = 1)

User-based recommendations
Example 1

universität koblenz
weiter:denken

	Pulp Fiction (1994)	Inception (2010)	Star Trek (2009)	The Matrix (1999)	Star Wars (1977)
Ben	4	?	1	4	2
Lea		2	5		5
Max			4	5	4
Sara		4	2	5	

Question: How would Ben rate the movie “Inception“?
Prediction:

1. Determine nearest neighbour of Ben (k = 1)
2. Predict Ben‘s rating for “Inception“ based on Sara‘s rating

User-based recommendations
Example 2

Prediction $f(u_2, i_4)$?

	i_1	i_2	i_3	i_4	i_5	…	i_m
u_1	3		2	4	4		
u_2		3	4	?	1		
u_3			1	2	5		2
u_4		3	5	3			
u_5			4	5	1		1
…							
u_n	3		4	4	2		2

A user is similar to u_2, when there are at least two ratings with difference ≤ 1

	i_1	i_2	i_3	i_4	i_5	…	i_m	#
u_1	3		2	4	4			0
u_2		3	4	?	1		2	4
u_3			1	2	5		2	1
u_4		3	5	3				**2**
u_5			4	5	1		1	**3**
…								
u_n	3		4	4	2		2	**3**

User-based recommendations
Example 2

- 3 NN of User u_2:
  - u_4
  - u_5
  - u_n
- All 3 NN rated Item i_4
- $f(u_2, i_4) = 4$

$$\begin{aligned} f(u_2, i_4) &= \frac{1}{3}(r_{44} + r_{54} + r_{n4}) \\ &= \frac{1}{3}(3 + 5 + 4) = \frac{12}{3} \\ &= 4 \end{aligned}$$

A user is similar to u_2, when there are at least two ratings with difference ≤ 1

	i_1	i_2	i_3	i_4	i_5	…	i_m	#
u_1	3		2	4	4			0
u_2		3	4	4	1		2	4
u_3			1	2	5		2	1
u_4		3	5	3				**2**
u_5			4	5	1		1	**3**
…								
u_n	3		4	4	2		2	**3**

User-based recommendations

$$f(u,i) = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

$$= \frac{1}{k}\left(r_{v_1 i} + r_{v_2 i} + \cdots + r_{v_k i}\right)$$

User-based recommendations

User u did not rate Item i yet

$$f(u,i) = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

$$= \frac{1}{k}\left(r_{v_1 i} + r_{v_2 i} + \cdots + r_{v_k i}\right)$$

User-based recommendations

User u did not rate Item i yet

$$f(u,i) = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

$$= \frac{1}{k}\left(r_{v_1 i} + r_{v_2 i} + \cdots + r_{v_k i}\right)$$

k-nearest neighbours of User u, who already rated Item i

User-based recommendations

universität koblenz
weiter:denken

User u did not rate Item i yet

$$f(u,i) = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi}$$

$$= \frac{1}{k}\left(r_{v_1 i} + r_{v_2 i} + \cdots + r_{v_k i}\right)$$

k-nearest neighbours of User u, who already rated Item i

Rating of neighbour v for Item i

Item-based recommendations

- **Question:**
  - How will I rate the movie “Pulp Fiction”?
- **Idea:**
  - Exploit similarities between movies
- **Prediction:**
  - Chose *k* movies that received a similar rating as “Pulp Fiction“ (k-nearest neighbours)
  - Most likely rating for the movie is predicted based on ratings for *k* nearest items

Item-based recommendations
Example

Prediction $f(u_2, i_4)$?

	i_1	i_2	i_3	i_4	i_5	…	i_m
u_1	3		2	4	4		
u_2		3	4	?	1		
u_3			1	2	5		2
u_4		3	5	3			
u_5			4	5	1		1
…							
u_n	3		4	4	2		2

An item is similar to i_4, when there are at least two ratings with a difference ≤ 1

	i_1	i_2	i_3	i_4	i_5	…	i_m
u_1	3		2	4	4		3
u_2	2	3	4	?	1		
u_3			1	2	5		2
u_4	4	3	5	3			
u_5			4	5	1		1
…							
u_n	3		4	4	2		5
#	**3**	1	**3**	5	1		**3**

Item-based recommendations
Example

- 3 NN of Item i_4:
  - i_1
  - i_3
  - i_m
- 2 NN received rating:
  - i_1
  - i_3
- $f(u_2, i_4) = 3$ (compare user-based)

$$
\begin{aligned}
f(u_2, i_4) &= \frac{1}{2}(r_{21} + r_{23}) \\
&= \frac{1}{2}(2+4) = \frac{6}{2} \\
&= 3
\end{aligned}
$$

An item is similar to i_4, when there are at least two ratings with a difference ≤ 1

	i_1	i_2	i_3	i_4	i_5	…	i_m
u_1	3		2	4	4		3
u_2	2	3	4	**3**	1		
u_3			1	2	5		2
u_4	4	3	5	3			
u_5			4	5	1		1
…							
u_n	3		4	4	2		5
#	**3**	1	**3**	5	1		**3**

Item-based recommendations

$$f(u,i) = \frac{1}{|N_u(i)|} \sum_{j \in N_u(i)} r_{uj}$$
$$= \frac{1}{k}\left(r_{uj_1} + r_{uj_2} + \cdots + r_{uj_k}\right)$$

Item-based recommendations

universität koblenz
weiter:denken

User u did not rate Item i yet

$$f(u,i) = \frac{1}{|N_u(i)|} \sum_{j \in N_u(i)} r_{uj}$$

$$= \frac{1}{k}\left(r_{uj_1} + r_{uj_2} + \cdots + r_{uj_k}\right)$$

k-nearest neighbours of Item i that User u rated

Rating for neighbour j of User u

K-NN methods

- **Advantages**
  - Easy to understand and to implement
  - No content analysis necessary
  - Easy to generate explanations for recommendations
- **Disadvantages**
  - Sparse Data Problem
  - Cold Start Problem
  - Popularity Bias
  - Hacking / Spam

Outline

- Introduction
- Basics
- Collaborative Filtering
- **Content-based Recommendations**
- Distributed Recommender Systems

Content-based recommendations

- **Question:**
  - How will I rate the movie “Pulp Fiction”?
- **Idea:**
  - Recommend items that match user preferences
- **Approach:**
  - Content: Information about items
  - User profile: Preferences of the user
  - Learn user preferences
  - Match content with user profile and recommend movie

Content-based recommendations
Example

Title	Genre	Director	Actor	Storyplot	…
Pulp Fiction	Crime Thriller	Tarantino	Travolta Thurman …	The lives of two mob hit men, a boxer, a gangster's wife, and a pair of diner bandits...	

Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman	**Term**
0	0	1	1	1	1	1	**Vector**
0	0	0.8	1.2	0.7	1.2	0.5	**TF-IDF**

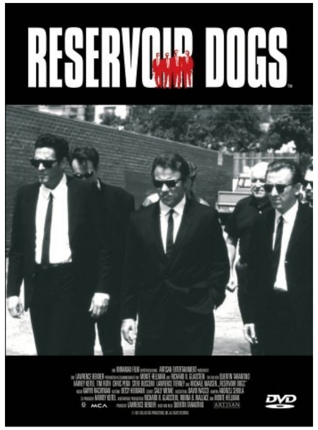
Transformation from text to Term-Vector Model and TF-IDF

Content-based recommendations Example

Title	Genre	Director	Actor	Storyplot	…
Reservoir Dogs	Crime Thriller	Tarantino	Buscemi Keitel …	After a simple jewelry heist goes terribly wrong, the surviving criminals begin to…	

Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman	**Term**
1	1	0	1	1	1	0	**Vector**
0.2	0.1	0	1.2	0.7	1.2	0	**TF-IDF**

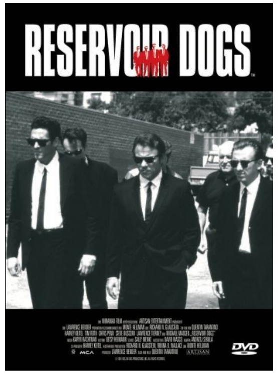
Transformation from text to Term-Vector Model and TF-IDF



Content-based recommendations

What are the preferences of user “Black“ (User Profile)?

Mr. Black



Pulp Fiction	Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman
	0	0	0.8	1.2	0.7	1.2	0.5
x 5	*0*	*0*	*4*	*6*	*3.5*	*6*	*2.5*

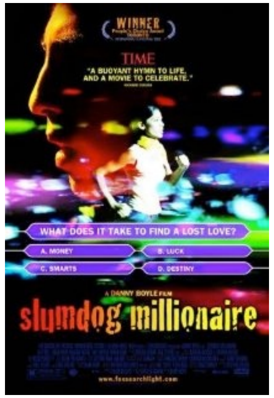
Reservoir Dogs	Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman
	0.2	0.1	0	1.2	0.7	1.2	0
x 4	*0.8*	*0.4*	*0*	*4.8*	*2.8*	*4.8*	*0*

Profile	Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman
∅	***0.4***	***0.2***	***2***	***5.4***	***3.15***	***5.4***	***1.25***

Content-based recommendations

Title	Genre	Director	Artist	Storyplot	...
Slumdog Millionaire	Drama Thriller Romanze	Boyle	Khan Patel ...	A Mumbai teen becomes a contestant on "Who Wants To Be A Millionaire?" ...	

Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman	**Term**
0	0	0	0	0	1	0	**Vector**
0	0	0	0	0	1.2	0	**TF-IDF**



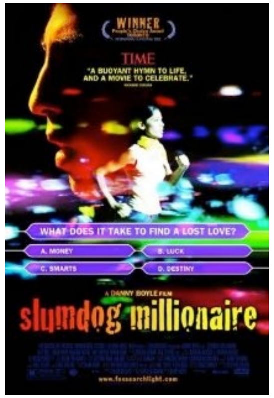
Content-based recommendations

Profile	Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman
x_u	*0.4*	*0.2*	*2*	*5.4*	*3.15*	*5.4*	*1.25*

Slumdog	Buscemi	Nickname	Gangster	Crime	Tarantino	Thriller	Thurman
x_i	0	0	0	0	0	1.2	0

Mr. Black

$$sim(u,i) = \frac{x_u^T x_i}{||x_u|| \, ||x_i||}$$



sim(Mr. Black, Slumdog) ≈ 0.63

Content-based recommendations

universität koblenz
weiter:denken

- Given:

Users	$U = \{u_1, u_2, \ldots, u_n\}$	
Items	$I = \{i_1, i_2, \ldots, i_m\}$	
Content	$X_I = \{x_{i1}, \ldots, x_{im}\}$	(Feature vectors of content of items)
Ratings	$R = (r_{ij})$	

- Task:
  - Construction / Learning of rating function $f: U \times I \rightarrow S$

- Approach
  - Learn user profile
  - Determine and recommend items that are close to the user profile

Learning of user profiles x_u

$$x_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui} x_i$$

Learning of user profiles x_u

$$x_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui} x_i$$

Set of all items that were rated by User u

Learning of user profiles x_u

$$x_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui} x_i$$

Feature vector of Item i

Set of all items that were rated by User u

Learning of user profiles x_u

$$x_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui} x_i$$

Weighted mean of
feature vectors of all
items rated by User u

Learning of user profiles x_u

Note: Update user profile after rating an item

$$x_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{ui} x_i$$

Weighted mean of feature vectors of all items rated by User u

Advantanges and Disadvantages

- **Advantage**
  - Does not require user community (as opposite to collaborative filtering)
- **Disadvantages**
  - Limited Content Analysis:
    - Insufficient information about Items/User
    - No information about the quality of the items
  - Over-Specialisation:
    - Algorithms tend to recommend items that are very similar
    - Example: Movies with same director, artist, genre
- Hardly any content-based recommender systems in use
  - Hybrid approaches: Collaborative filtering + content-based recommendations

Outline

universität koblenz
weiter:denken

- Introduction
- Basics
- Collaborative Filtering
- Content-based Recommendations
- **Distributed Recommender Systems**

Case study: Bloomberg Media

Real-time recommendation of news articles based on user activities

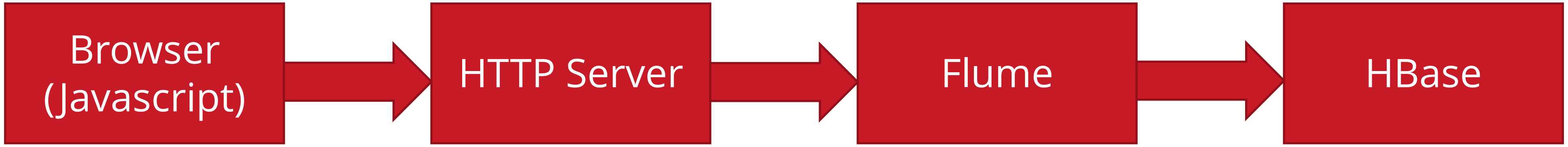
- Social media shares
- Page views

Shah et al. 2016. News Recommendations at scale at Bloomberg Media: Challenges and Approaches. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 369.

Storing user interaction data (HBase)

universität koblenz
weiter:denken

- 100s of millions of users
- Millions of stories/videos
- TBs of data
- Wide tables – 1 row per user
- High load
- Sub-second response times
- Multiple MR jobs every few mins



Shah et al. 2016. News Recommendations at scale at Bloomberg Media: Challenges and Approaches. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 369.

Generating user models using MR framework

- Content-based recommendation
  - Parallelize recommendations over users
  - Recommendations should be based on latest news/interaction
  - Train only when user has new interactions (every 5 minutes)
- Collaborative filtering
  - User model dependent of other users
  - Train all user models frequently

Shah et al. 2016. News Recommendations at scale at Bloomberg Media: Challenges and Approaches. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 369.

Computing recommendations (in HBase)

- Query HBase
- Evaluate articles against user models
- In-memory cache
- 1000s of requests per minute
- less than 50ms response time

Shah et al. 2016. News Recommendations at scale at Bloomberg Media: Challenges and Approaches. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 369.

Case study: bol.com

Recommender system handling billions of interactions on an e-commerce platform

Relies on Spark MLLib

B. Kersbergen and S. Schelter, "Learnings from a Retail Recommendation System on Billions of Interactions at bol.com," *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Chania, Greece, 2021, pp. 2447-2452.

Summary

universität koblenz
weiter:denken

- Recommender Systems
  - Recommend items that a user does not know, but might be interesting for him
  - Reduces information overload by predicting relevance
  - Information need is unspecific
  - Popular approach: Learning prediction models
- Approaches
  - Collaborative filtering
  - Content-based Recommendations
- Distributed Recommender Systems