Recommender System

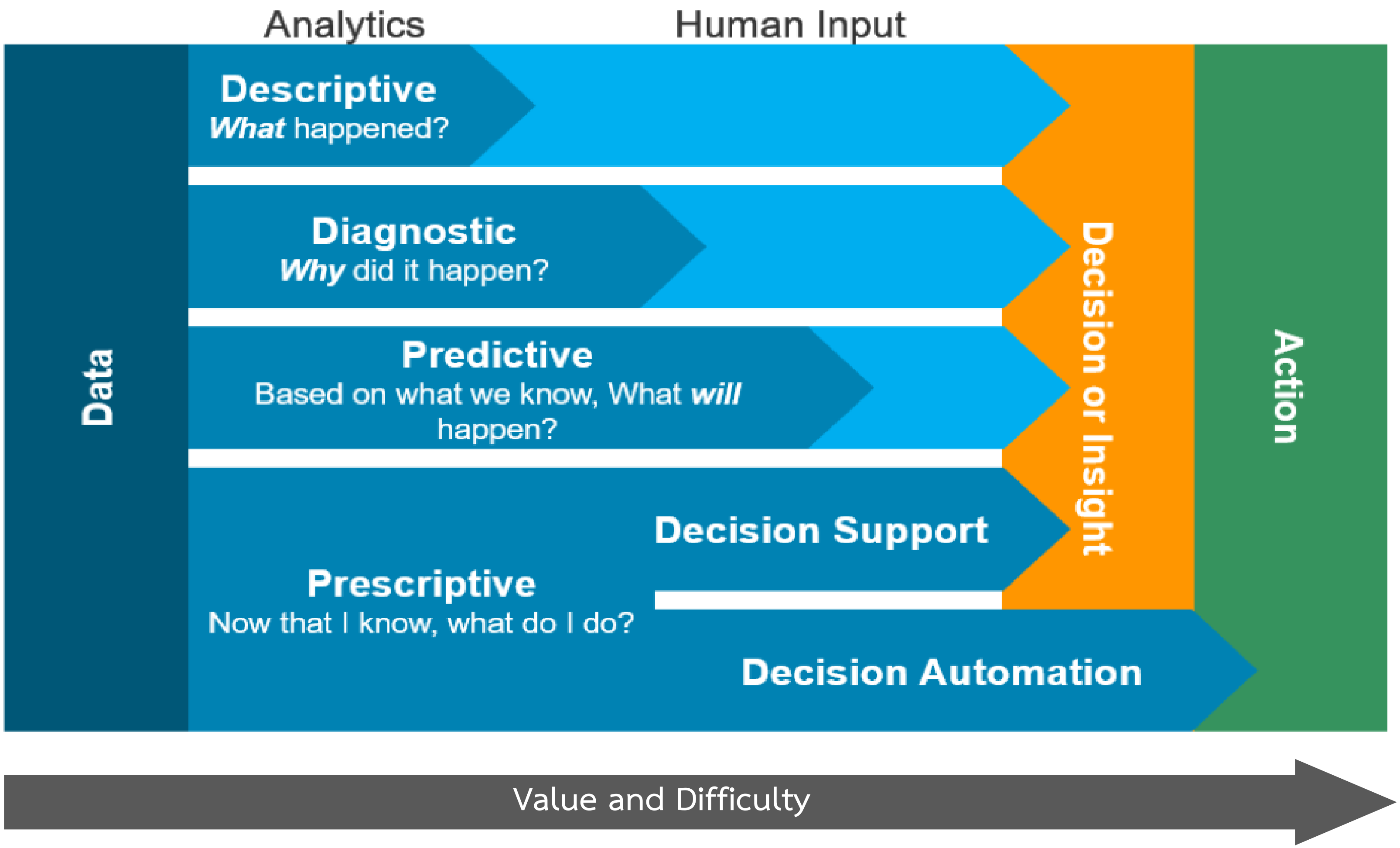
Dr. Rathachai Chawuthai

Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang

Agenda

- Overview
- Evaluating Recommender Systems
- Collaborative Recommendation
- Content-based Recommendation
- Knowledge-based Recommendation
- Hybrid Recommender Approach

Data Analytics



Ref:
- Four types of analytics capability (Gartner, 2014)
- (image) https://www.healthcatalyst.com/closed-loop-analytics-method-healthcare-data-insights

Overview

Recommender Systems

Recommender Systems

- **Application areas**

You may also like

Jack & Jones
JAMIE - Polo shirt - orange
£21.00
Free delivery & returns

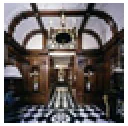
ALTERNATIVE PRODUCTS

Beko Washing Machine
Code: WMB81431LW
£269.99

Zanussi Washing Machine
Code: ZWH6130P
£269.99

Blomberg Washing Machine
Code: WNF6221
£299.99

Related hotels...



Hotel 41 — Show Prices
1,170 Reviews
London, England

Read | Commented | Recommended

Germany Just Rejected The Idea That The European Bailout Fund Would Buy Spanish Debt ×

There Is Almost No Gold In The Olympic Gold Medal ×

You may also like

(109)

(53)

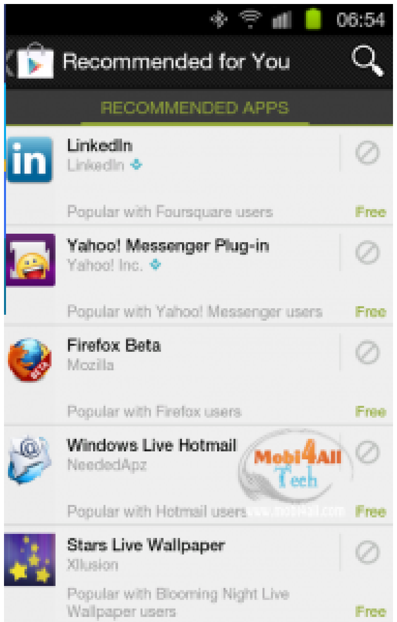
(33)

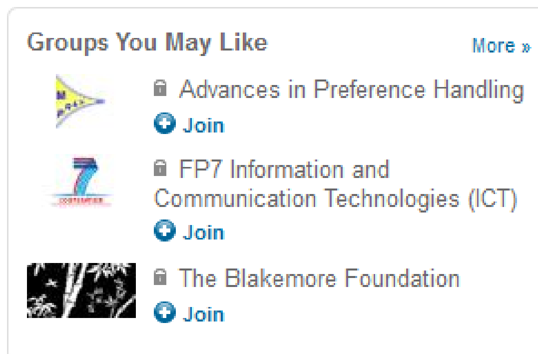
MOST POPULAR | RECOMMENDED

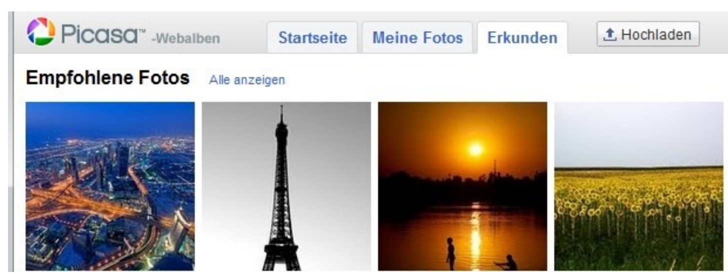
How to Break NRA's Grip on Politics: Michael R. Bloomberg

Growth in U.S. Slows as Consumers Restrain Spending

In the Social Web

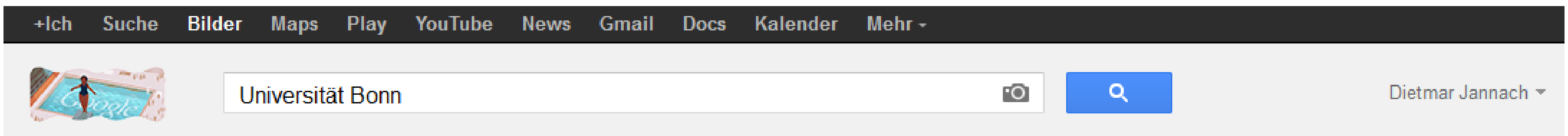






Even more ...

- **Personalized search**



- **"Computational advertising"**

Why using Recommender Systems?

- **Value for the customer**
  - Find things that are interesting
  - Narrow down the set of choices
  - Help me explore the space of options
  - Discover new things
  - Entertainment
  - …
- **Value for the provider**
  - Additional and probably unique personalized service for the customer
  - Increase trust and customer loyalty
  - Increase sales, click trough rates, conversion etc.
  - Opportunities for promotion, persuasion
  - Obtain more knowledge about customers
  - …

Real-world check

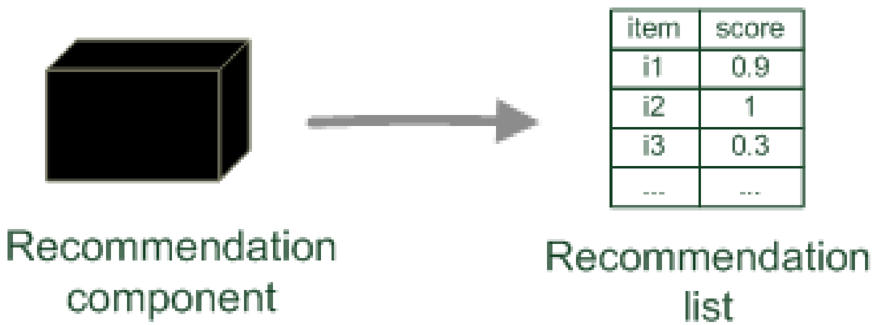
- **Myths from industry**
  - Amazon.com generates X percent of their sales through the recommendation lists ($30 < X < 70$)
  - Netflix (DVD rental and movie streaming) generates X percent of their sales through the recommendation lists ($30 < X < 70$)
- **There must be some value in it**
  - See recommendation of groups, jobs or people on LinkedIn
  - Friend recommendation and ad personalization on Facebook
  - Song recommendation at last.fm
  - News recommendation at Forbes.com (plus 37% CTR)
- **Academia**
  - A few studies exist that show the effect
    - increased sales, changes in sales behavior

Recommender systems

- **RS seen as a function**
- **Given:**
  - User model (e.g. ratings, preferences, demographics, situational context)
  - Items (with or without description of item characteristics)
- **Find:**
  - Relevance score. Used for ranking.
- **Finally:**
  - Recommend items that are assumed to be relevant
- **But:**
  - Remember that relevance might be context-dependent
  - Characteristics of the list itself might be important (diversity)

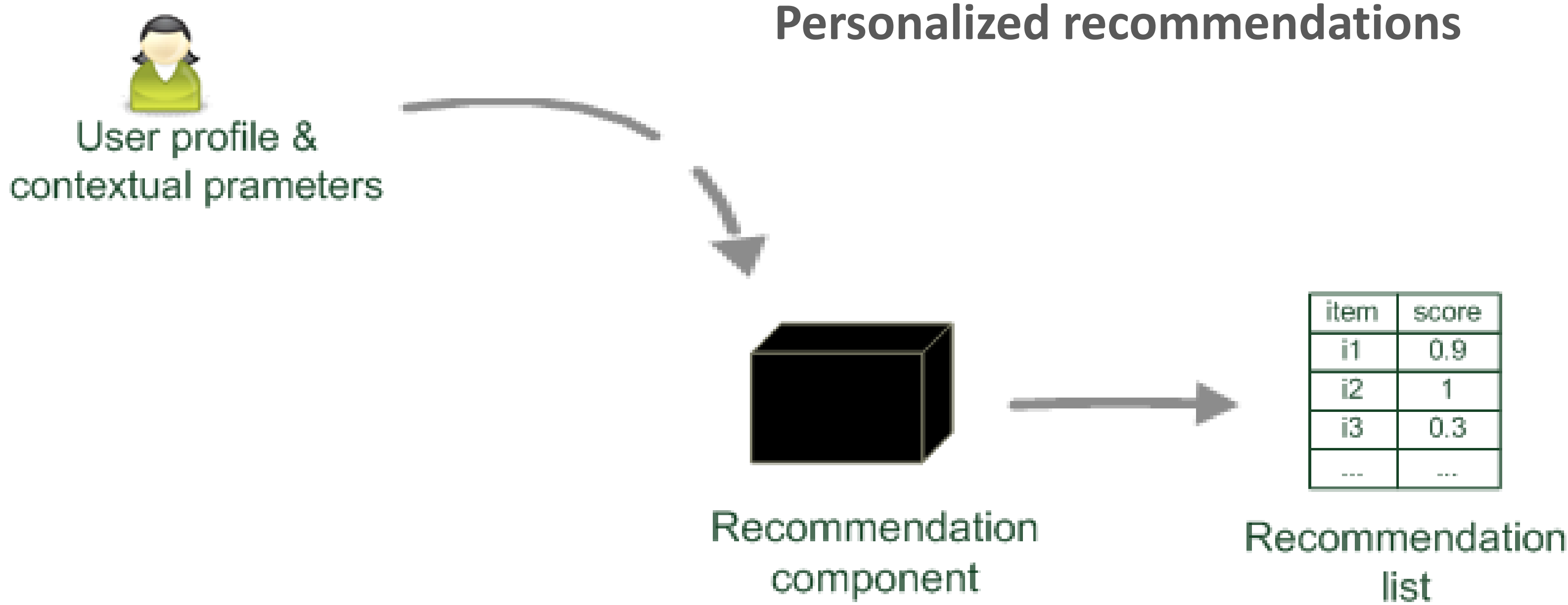
Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance



Paradigms of recommender systems

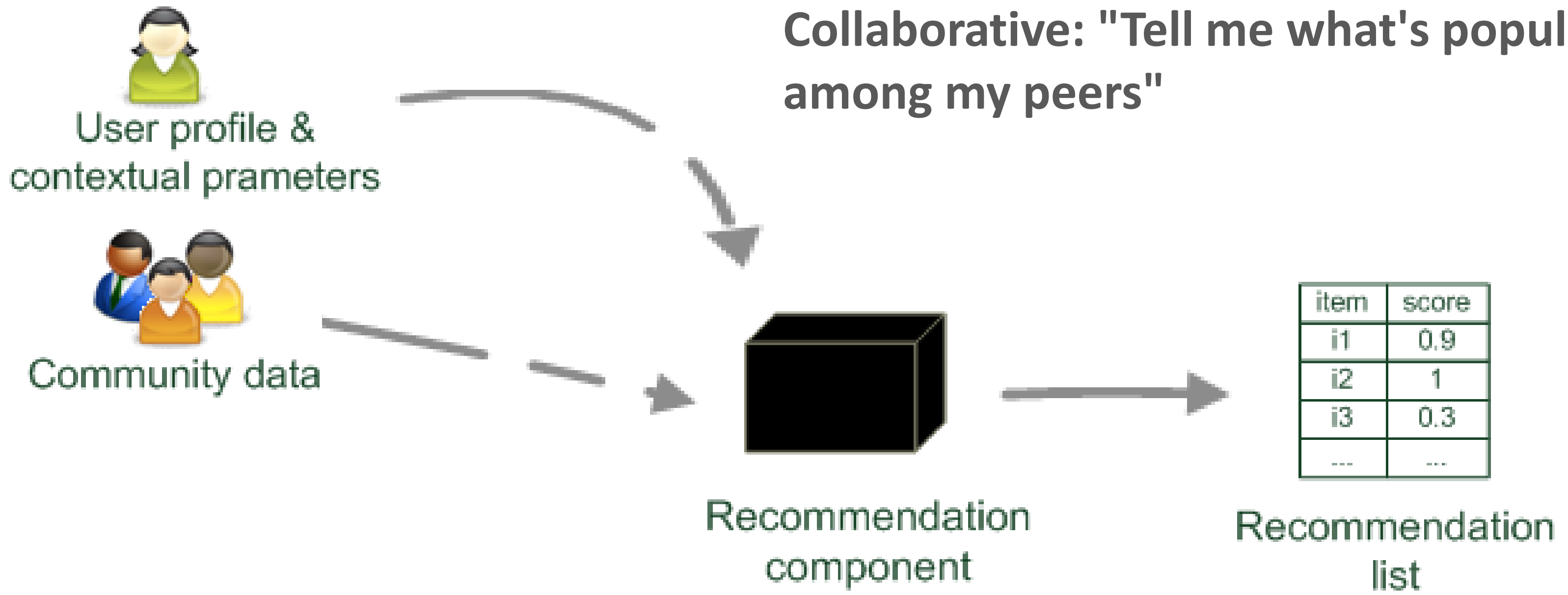
Personalized recommendations



item	score
i1	0.9
i2	1
i3	0.3
...	...

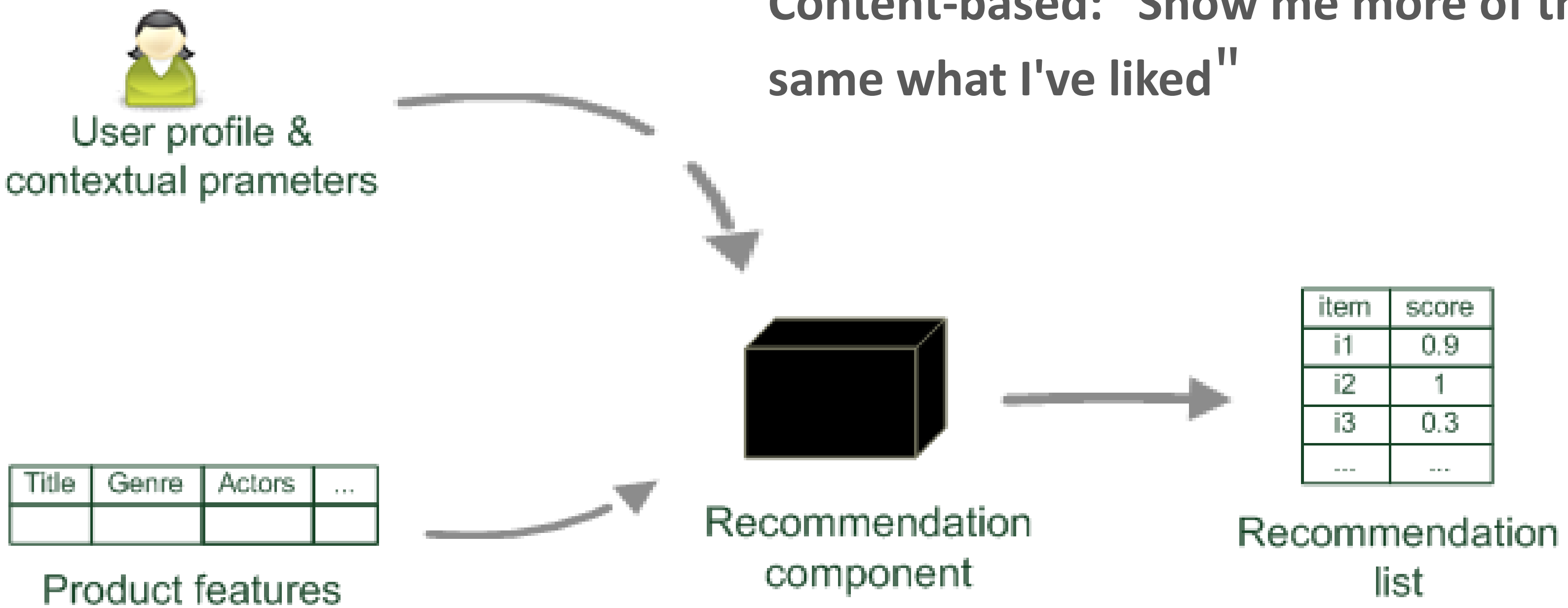
Paradigms of recommender systems

Collaborative: "Tell me what's popular among my peers"



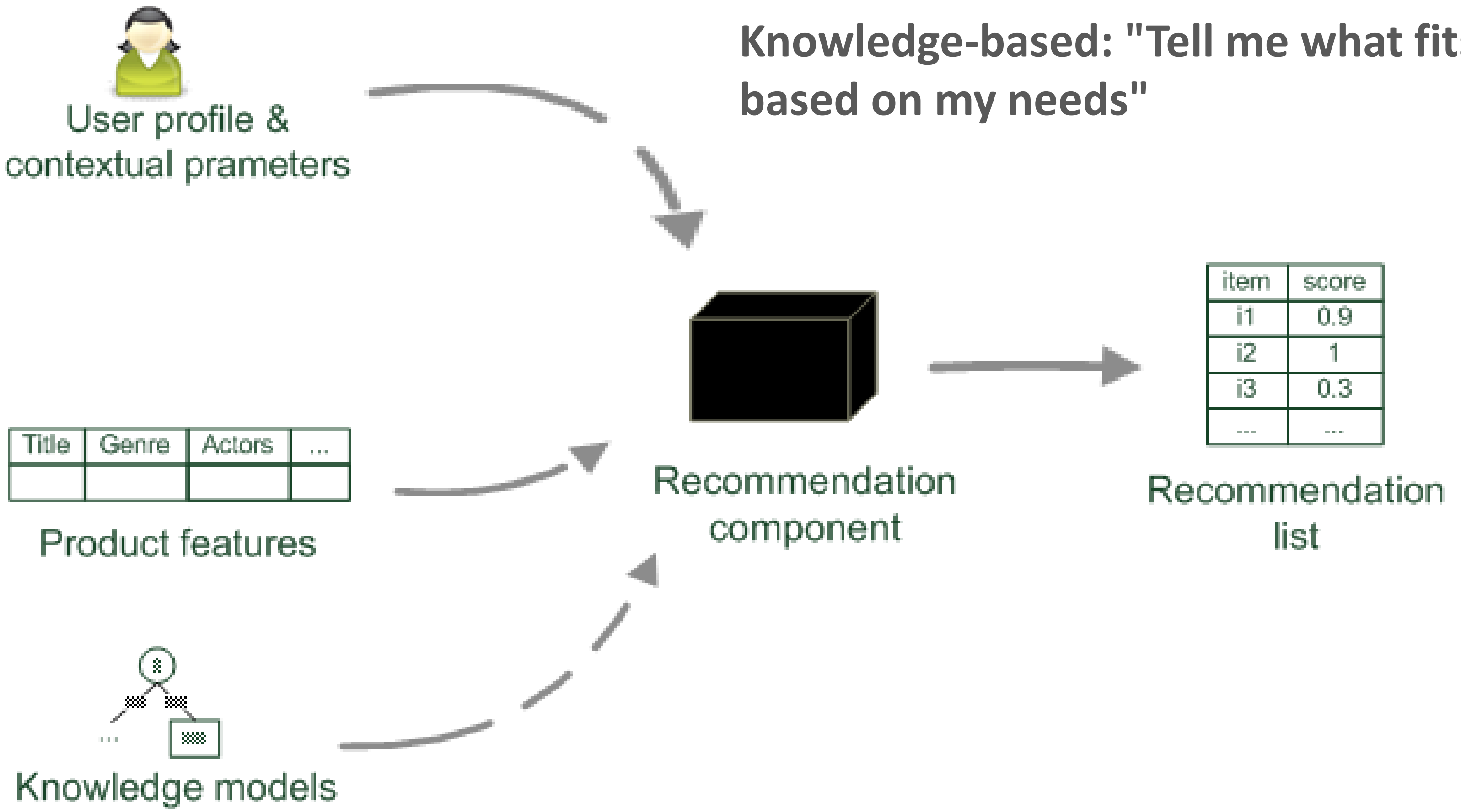
Paradigms of recommender systems

Content-based: "Show me more of the same what I've liked"

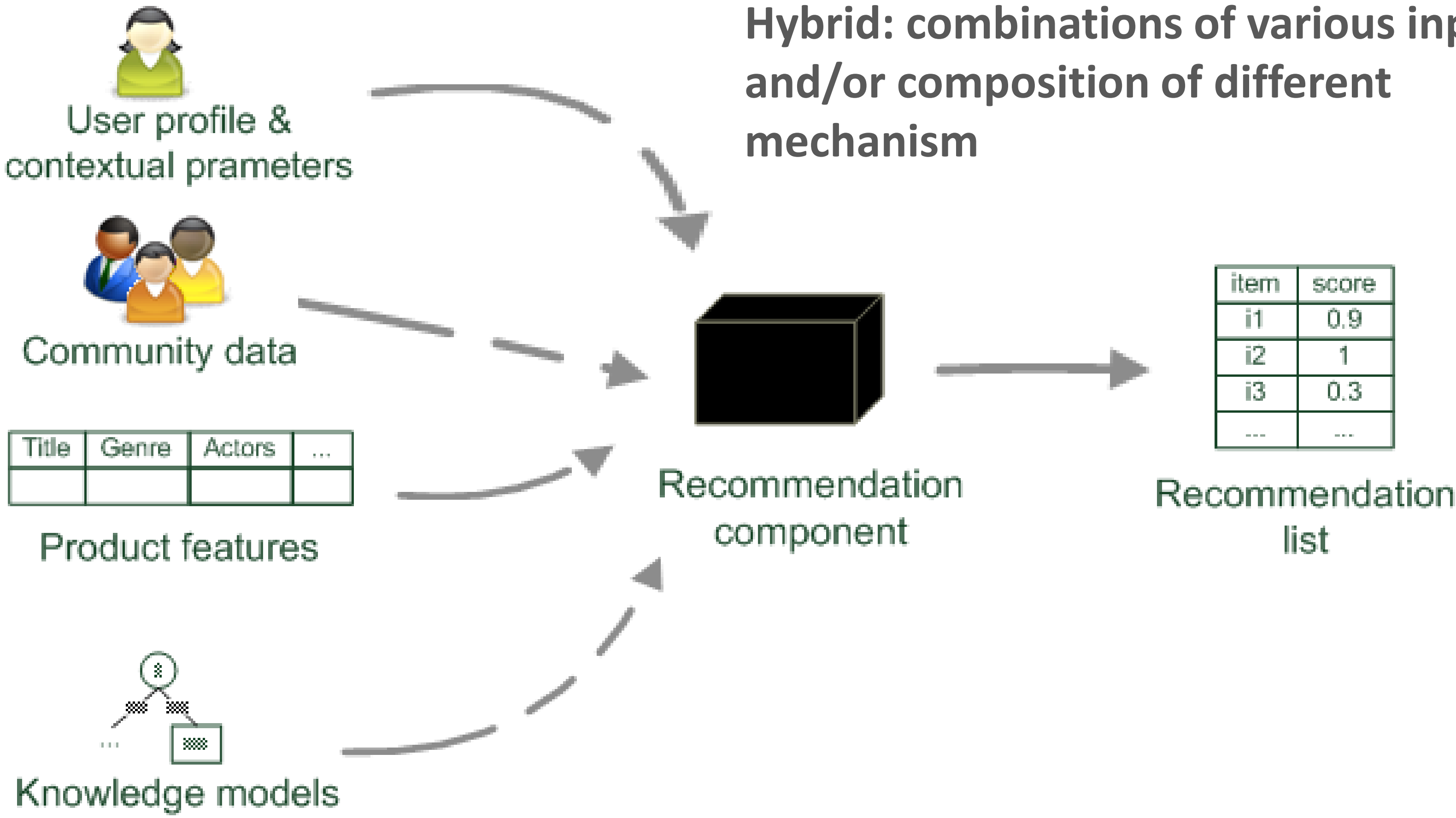


Paradigms of recommender systems

Knowledge-based: "Tell me what fits based on my needs"



Paradigms of recommender systems



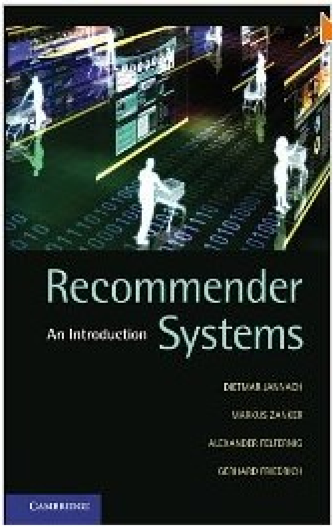
Hybrid: combinations of various inputs and/or composition of different mechanism

Recommender systems: basic techniques

	Pros	Cons
Collaborative	No knowledge-engineering effort, serendipity of results, learns market segments	Requires some form of rating feedback, cold start for new users and new items
Content-based	No community required, comparison between items possible	Content descriptions necessary, cold start for new users, no surprises
Knowledge-based	Deterministic recommendations, assured quality, no cold-start, can resemble sales dialogue	Knowledge engineering effort to bootstrap, basically static, does not react to short-term trends

Evaluating Recommender Systems

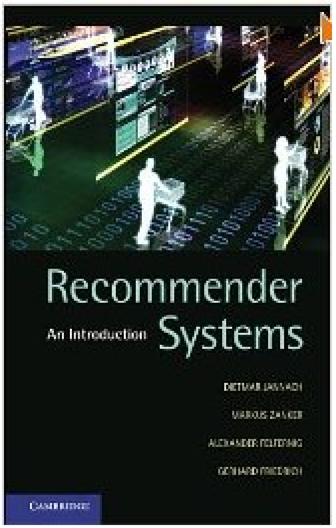
Recommender Systems in e-Commerce



- One Recommender Systems research question?
  - What should be in that list?

Ref: • Deitmar Jannach, Makus Zanker and Gerheard Friedrich

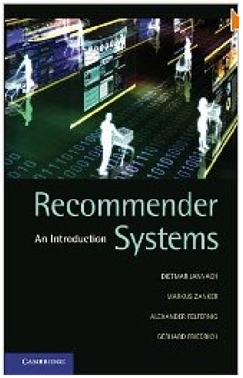
Recommender Systems in e-Commerce



- Another question both in research and practice
  - How do we know that these are good recommendations?

Ref: • Deitmar Jannach, Makus Zanker and Gerheard Friedrich

Recommender Systems in e-Commerce



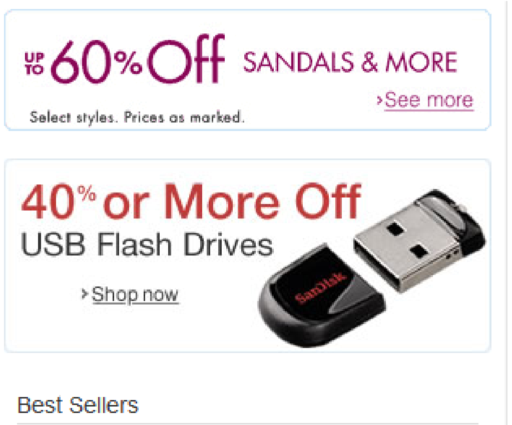
- This might lead to …
  - What is a good recommendation?
  - What is a good recommendation strategy?
  - What is a good recommendation strategy for my business?

Ref: • Deitmar Jannach, Makus Zanker and Gerheard Friedrich

What is a good recommendation?

What are the measures in practice?

- Total sales numbers
- Promotion of certain items
- …
- Click-through-rates
- Interactivity on platform
- …
- Customer return rates
- Customer satisfaction and loyalty



MAE

- Mean Absolute Error

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}$$

MAE

(for rating)

Movie	Actual Rating (y_i)
A	4
B	3
C	
D	5
E	2
F	

MAE

(for rating)

Movie	Actual Rating (y_i)	Predicted Rating ($\hat{y}_i$)
A	4	4
B	3	4
C		3
D	5	5
E	2	3
F		2

MAE

(for rating)

Movie	Actual Rating (y_i)	Predicted Rating ($\hat{y}_i$)	$\lvert y_i - \hat{y}_i \rvert$
A	4	4	0
B	3	4	1
C		3	
D	5	5	0
E	2	3	1
F		2	

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$$

MAE

(for rating)

Movie	Actual Rating (y_i)	Predicted Rating ($\hat{y}_i$)	$\lvert y_i - \hat{y}_i \rvert$
A	4	4	0
B	3	4	1
C		3	
D	5	5	0
E	2	3	1
F		2	

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$$

$\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$ 1+1 = **2**

$\frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$ 2/4 = **0.5**

MAE

(for buying)

Product	is Bought (y_i)
A	1
B	1
C	0
D	1
E	1
F	0

MAE

(for buying)

Product	is Bought (y_i)	predicting Score ($\hat{y}_i$)	$\lvert y_i - \hat{y}_i \rvert$
A	1	0.8	0.2
B	1	0.6	0.4
C	0	0.3	0.3
D	1	0.5	0.5
E	1	0.8	0.2
F	0	0.2	0.2

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$$

$$\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert \quad 1.8$$

$$\frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert \quad 1.8/6 = \mathbf{0.3}$$

Top-K Precision

- Pickup ***K*** items from the recommended list
- Count how many actual items that users interact
  - If there are ***N*** items,

$$\boldsymbol{Top - K\ Precision = \frac{N}{K}}$$

Top-K Precision

rec(u,i)

Item	Score
A	0.59
B	0.66
C	0.41
D	0.51
E	0.67
F	0.09
G	0.13
H	0.49
I	0.66
J	0.64
K	0.13
L	0.81
M	0.40
N	0.06

Top-K Precision

rec(u,i)

Item	Score
A	0.59
B	0.66
C	0.41
D	0.51
E	0.67
F	0.09
G	0.13
H	0.49
I	0.66
J	0.64
K	0.13
L	0.81
M	0.40
N	0.06

Bought Item

Bought Item

Bought Item

Bought Item

Top-K Precision

rec(u,i)

Item	Score ▼	
L	0.81	Bought Item
E	0.67	Bought Item
I	0.66	
B	0.66	Bought Item
J	0.64	
A	0.59	
D	0.51	
H	0.49	
C	0.41	
M	0.40	
K	0.13	
G	0.13	Bought Item
F	0.09	
N	0.06	

Top-K Precision

rec(u,i)

Item	Score ▼
L	0.81
E	0.67
I	0.66
B	0.66
J	0.64
A	0.59
D	0.51
H	0.49
C	0.41
M	0.40
K	0.13
G	0.13
F	0.09
N	0.06

TOP 5

$$Top - 5\ Precision = \frac{N}{K}$$

$$= \frac{3}{5}$$

Top-K Precision

What happened if

Item	Score ▼

?

Area Under ROC Curve (AUC)

- Area Under ROC Curve for Recommender System

$$AUC = \frac{n' + 0.5n''}{n}$$

For n comparisons,

- n' is number of times when the tested items have higher score than the recommended items.
- n'' is number of times when the tested items have same score as the recommended items.

AUC

rec(u,i)

Item	Score ▼
E	0.8
C	0.7
B	0.6
A	0.6
D	0.5
F	0.3
G	0.2

- There are 5 recommended items
- Comparisons are
  - C compares to E, A, D, F and G (5 times)
  - B compares to E, A, D, F and G (5 times)
- There are 5+5 = 10 comparisons
- So, *n = 10*

AUC

rec(u,i)

Item	Score ▼
E	0.8
C	0.7
B	0.6
A	0.6
D	0.5
F	0.3
G	0.2

Focus on tested items

- C compares to E, A, and D
  - C > A, D, F, G (4 times)
- B compares to E, A, and D
  - B = A (1 time)
  - B > D, F, G (3 time)
- There are 4+3 =7 times that tested items have higher score than recommended items, so $n' = 7$
- There are 1 times that tested items have the same score as recommended items, so $n'' = 1$

AUC

rec(u,i)

Item	Score ▼
E	0.8
C	0.7
B	0.6
A	0.6
D	0.5
F	0.3
G	0.2

For n comparisons,

- n' is number of times when the tested items have higher score than the recommended items.
- n'' is number of times when the tested items have same score as the recommended items.

$$AUC = \frac{n' + 0.5n''}{n}$$

$$= \frac{7 + 0.5 \times 1}{10} = \frac{7.5}{10} = 0.75$$

0.5 is a random recommendation

AUC ?

Item	Score ▼

AUC = ?

Item	Score ▼

AUC = ?

Item	Score ▼

AUC = ?

Decide to Evaluate?

Firstly,

- Use Top-K Precision
  - check the accuracy of top recommended items
  - close to the real situation when the number of recommend items are limited

Secondly,

- Use AUC
  - check the overall performance of an algorithm

Collaborative Recommendation

Collaborative Recommendation

- The most prominent approach to generate recommendations
  - used by large, commercial e-commerce sites
  - well-understood, various algorithms and variations exist
  - applicable in many domains (book, movies, DVDs, ..)
- Approach
  - use the "wisdom of the crowd" to recommend items
- Basic assumption and idea
  - Users give ratings to catalog items (implicitly or explicitly)
  - Customers who had similar tastes in the past, will have similar tastes in the future

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

- Given an "active user" (Alice) and an item I not yet seen by Alice
- The *goal is to estimate Alice's rating for this item*, e.g., by
  - find a set of users (peers) who liked the same items as Alice in the past **and** who have rated item I
  - use, e.g. the average of their ratings to predict, if Alice will like item I
  - do this for all items Alice has not seen and recommend the best-rated

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

- Some first questions
  - How do we measure similarity?
  - How many neighbors should we consider?
  - How do we generate a prediction from the neighbors' ratings?

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

- A popular similarity measure in user-based CF: **Pearson correlation**

a, b : users

$r_{a,p}$: rating of user a for item p

P : set of items, rated both by a and b

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

Possible similarity values between -1 and 1; $\bar{r}_a, \bar{r}_b$ = user's average ratings

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim(alice,user1) = 0.85

sim(alice,user2) = 0.70

sim(alice,user0) = 0.00

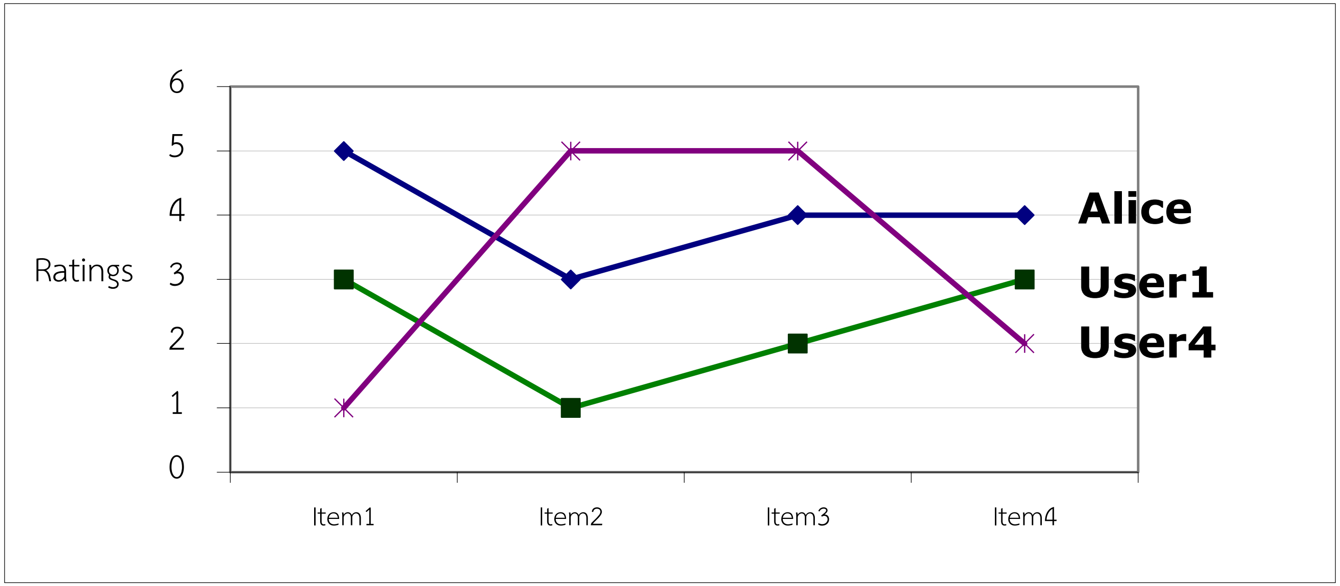
sim(alice,user4) = -0.79

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

Pearson Correlation

- Takes differences in rating behavior into account



- Works well in usual domains, compared with alternative measures
  - such as cosine similarity

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

User-based Collaborative Filtering

Making Prediction

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim(alice,user1) = 0.85

sim(alice,user2) = 0.70

sim(alice,user3) = 0.00

sim(alice,user4) = -0.79

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

$$\frac{3+1+2+3+3}{5}$$

$$\frac{4+3+4+3+5}{5}$$

$$\text{pred(alice, item5)} = \frac{5+3+4+4}{4} + \frac{0.85 * (3 - 2.4) + 0.7 * (5 - 3.8)}{0.85 + 0.7} = 4.87$$

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Improving the metrics / prediction function

- Not all neighbor ratings might be equally "valuable"
  - Agreement on commonly liked items is not so informative as agreement on controversial items
  - **Possible solution**: Give more weight to items that have a higher variance
- Value of number of co-rated items
  - Use "significance weighting", by e.g., linearly reducing the weight when the number of co-rated items is low
- Case amplification
  - Intuition: Give more weight to "very similar" neighbors, i.e., where the similarity value is close to 1.
- Neighborhood selection
  - Use similarity threshold or fixed number of neighbors

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Memory-based and model-based approaches

- User-based CF is said to be "memory-based"
  - the rating matrix is directly used to find neighbors / make predictions
  - does not scale for most real-world scenarios
  - large e-commerce sites have tens of millions of customers and millions of items
- Model-based approaches
  - based on an offline pre-processing or "model-learning" phase
  - at run-time, only the learned model is used to make predictions
  - models are updated / re-trained periodically
  - large variety of techniques used
  - model-building and updating can be computationally expensive

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

- Basic idea:
  - Use the similarity between items (and not users) to make predictions
- Example:
  - Look for items that are similar to Item5
  - Take Alice's ratings for these items to predict the rating for Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1		1	2		
User2		3	4		
User3		3	1		
User4		5	5		

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

- Produces better results in item-to-item filtering
  - for some datasets, no consistent picture in literature
- Ratings are seen as vector in n-dimensional space
- Similarity is calculated based on the angle between the vectors

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

- Adjusted cosine similarity
  - take average user ratings into account, transform the original ratings
  - U: set of users who have rated both items a and b

$$sim(a, b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

	Item1	Item2	Item3	Item4	Item5	avg of user, or $\overline{r_u}$
Alice	5	3	4	4	?	**4.0**
User1	3	1	2	3	3	**2.4**
User2	4	3	4	3	5	**3.8**
User3	3	3	1	5	4	**3.2**
User4	1	5	5	2	1	**2.8**

	Item1	Item2	Item3	Item4	Item5
Alice	5 – *4.0*	3 – *4.0*	4 – *4.0*	4 – *4.0*	?
User1	3 – *2.4*	1 – *2.4*	2 – *2.4*	3 – *2.4*	3 – *2.4*
User2	4 – *3.8*	3 – *3.8*	4 – *3.8*	3 – *3.8*	5 – *3.8*
User3	3 – *3.2*	3 – *3.2*	1 – *3.2*	5 – *3.2*	4 – *3.2*
User4	1 – *2.8*	5 – *2.8*	5 – *2.8*	2 – *2.8*	1 – *2.8*

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

avg of user, or $\bar{r}_u$
4.0
2.4
3.8
3.2
2.8

	Item1	Item2	Item3	Item4	Item5
Alice	1.0	-1.0	0.0	0.0	?
User1	0.6	-1.4	-0.4	0.6	0.6
User2	0.2	-0.8	0.2	-0.8	1.2
User3	-0.2	-0.2	-2.2	1.8	0.8
User4	-1.8	2.2	2.2	-0.8	-1.8

Mean-Adjusted Rating Database

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

Mean-Adjusted Rating Database

	Item1	Item2	Item3	Item4	Item5
Alice	1.0	-1.0	0.0	0.0	?
User1	0.6	-1.4	-0.4	0.6	0.6
User2	0.2	-0.8	0.2	-0.8	1.2
User3	-0.2	-0.2	-2.2	1.8	0.8
User4	-1.8	2.2	2.2	-0.8	-1.8

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

Mean-Adjusted Rating Database

	Item1	Item2	Item3	Item4	Item5
Alice	1.0	-1.0	0.0	0.0	?
User1	0.6	-1.4	-0.4	0.6	0.6
User2	0.2	-0.8	0.2	-0.8	1.2
User3	-0.2	-0.2	-2.2	1.8	0.8
User4	-1.8	2.2	2.2	-0.8	-1.8

$$sim(a,b) = \frac{\sum_{u \in U} (r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U} (r_{u,a} - \overline{r_u})^2} \sqrt{\sum_{u \in U} (r_{u,b} - \overline{r_u})^2}}$$

$$\boldsymbol{sim(i1, i5) = \frac{0.6 \times 0.6 + 0.2 \times 1.2 + (-0.2) \times 0.8 + (-1.8) \times (-1.8)}{\sqrt{0.6^2 + 0.2^2 + (-0.2)^2 + (-1.8)^2} \times \sqrt{0.6^2 + 1.2^2 + 0.8^2 + (-1.8)^2}} = 0.8}$$

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

- after calculation
  - *sim(i5, i1)* = +0.8 *(very high similarity)*
  - *sim(i5, i2)* = -0.9 *(very low similarity)*
  - *sim(i5, i3)* = -0.8 *(very low similarity)*
  - *sim(i5, i4)* = +0.4 *(high similarity)*

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

- choose similar items
  - *sim(i5, i1) = +0.8*
  - *sim(i5, i2) = -0.9*
  - *sim(i5, i3) = -0.8*
  - *sim(i5, i4) = +0.4*

	Item1	Item2	Item3	Item4	Item5
Alice	$\boxed{5}$	3	4	$\boxed{4}$	$\boxed{?}$
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

$$\boldsymbol{pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) \times r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}}$$

$$\boldsymbol{pred(alice, i5) = \frac{0.8 \times 5 + 0.4 \times 4}{0.8 + 0.4} = \frac{5.6}{1.2} = 4.7}$$

	Item5
Alice	4.7

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Item-based Collaborative Filtering

- choose similar items
  - *sim(i5, i1) = +0.8*
  - *sim(i5, i2) = -0.9*
  - *sim(i5, i3) = -0.8*
  - *sim(i5, i4) = +0.4*

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) \times r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$

$$pred(alice, i5) = \frac{0.8 \times 5}{0.8} = 5$$

	Item5
Alice	5

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Pre-processing for item-based filtering

- Item-based filtering does not solve the scalability problem itself
- Pre-processing approach by Amazon.com (in 2003)
  - Calculate all pair-wise item similarities in advance
  - The neighborhood to be used at run-time is typically rather small, because only items are taken into account which the user has rated
  - Item similarities are supposed to be more stable than user similarities
- Memory requirements
  - Up to N^2 pair-wise similarities to be memorized (N = number of items) in theory
  - In practice, this is significantly lower (items with no co-ratings)
  - Further reductions possible
    - Minimum threshold for co-ratings (items, which are rated at least by *n* users)
    - Limit the size of the neighborhood (might affect recommendation accuracy)

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

More on ratings

- Pure CF-based systems only rely on the rating matrix
- Explicit ratings
  - Most commonly used (1 to 5, 1 to 7 Likert response scales)
  - Research topics
    - "Optimal" granularity of scale; indication that 10-point scale is better accepted in movie domain
    - Multidimensional ratings (multiple ratings per movie)
  - Challenge
    - Users not always willing to rate many items; sparse rating matrices
    - How to stimulate users to rate more items?
- Implicit ratings
  - clicks, page views, time spent on some page, demo downloads …
  - Can be used in addition to explicit ones; question of correctness of interpretation

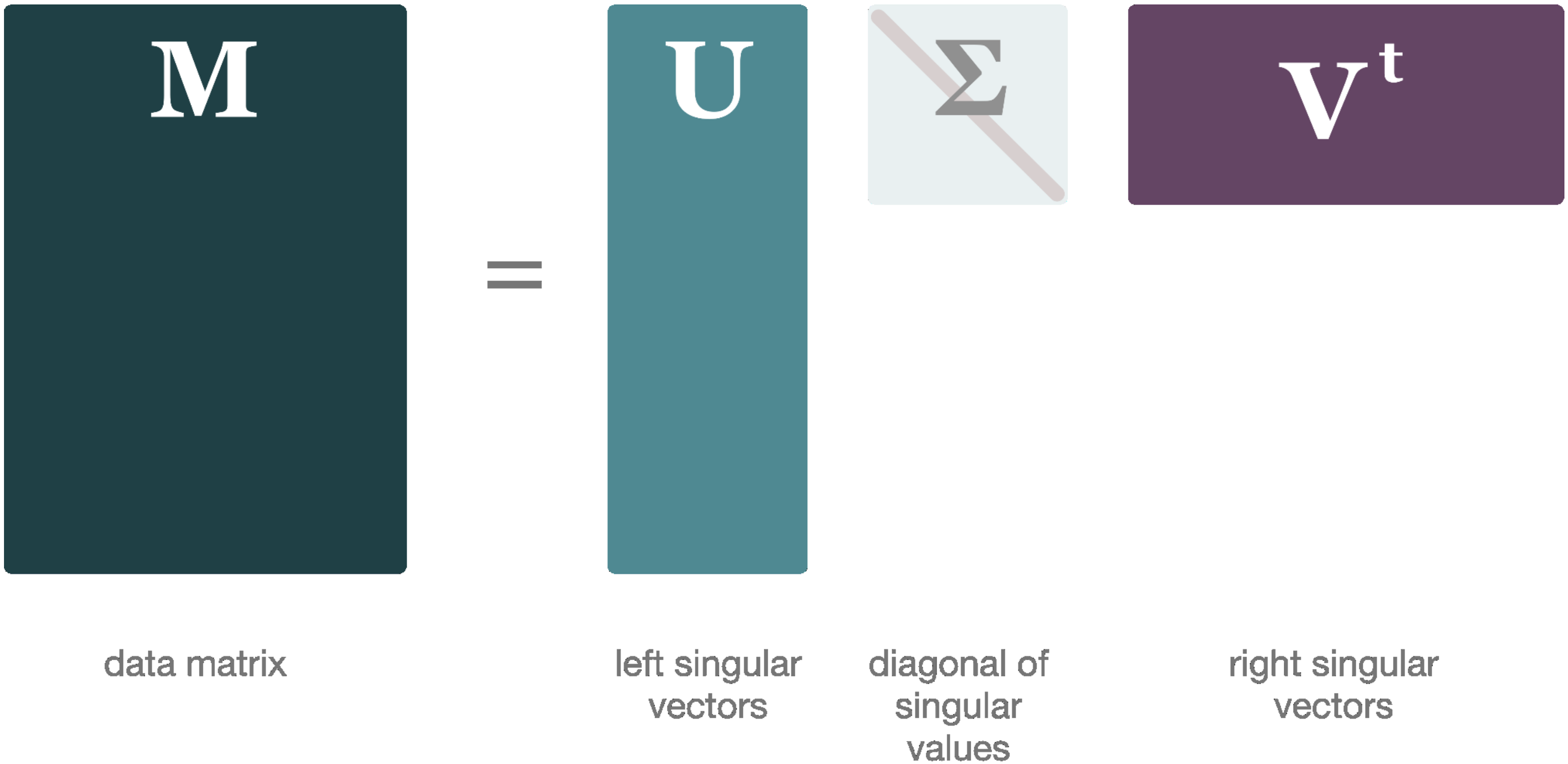
Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Data sparsity problems

- Cold start problem
  - How to recommend new items? What to recommend to new users?
- Straightforward approaches
  - Ask/force users to rate a set of items
  - Use another method (e.g., content-based, demographic or simply non-personalized) in the initial phase
- Alternatives
  - Use better algorithms (beyond nearest-neighbor approaches), for example:
    - In nearest-neighbor approaches, the set of sufficiently similar neighbors might be to small to make good predictions
    - Assume "transitivity" of neighborhoods

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

SVD (Singular Value Decomposition)



Ref: • (image) https://dsc-spidal.github.io/harp/docs/harpdaal/svd/

PCA

Principal Component Analysis (PCA) ภาษาไทย Update Version

https://www.youtube.com/watch?v=1qZyWTTw0mI

Ref: • https://www.youtube.com/watch?v=1qZyWTTw0mI

SVD

- Example

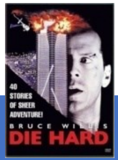
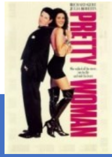
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Ref: • (image) http://people.seas.harvard.edu/~babis/kdd09.html

Matrix Factorization

SVD $$M_k = U_k \times \Sigma_k \times {V_k}^T$$

U_k	Dim1	Dim2
Alice	0.47	-0.30
Bob	-0.44	0.23
Mary	0.70	-0.06
Sue	0.31	0.93

${V_k}^T$					
Dim1	-0.44	-0.57	0.06	0.38	0.57
Dim2	0.58	-0.66	0.26	0.18	-0.36

Σ_k	Dim1	Dim2
Dim1	5.63	0
Dim2	0	3.23

- **Prediction:** $\hat{r}_{ui} = \bar{r}_u + U_k(Alice) \times \Sigma_k \times V_k^T(EPL)$

  = 3 + 0.84 = 3.84

2008: Factorization meets the neighborhood: a multifaceted collaborative filtering model, Y. Koren, ACM SIGKDD

- Merges neighborhood models with latent factor models
- Latent factor models
  - good to capture weak signals in the overall data
- Neighborhood models
  - good at detecting strong relationships between close items
- Combination in one prediction single function
  - Local search method such as stochastic gradient descent to determine parameters
  - Add penalty for high values to avoid over-fitting

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

$$\min_{p_*, q_*, b_*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Collaborative Filtering: Pros & Cons

- Pros:
  - well-understood, works well in some domains, no knowledge engineering required

- Cons:
  - requires user community, sparsity problems, no integration of other knowledge sources, no explanation of results

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

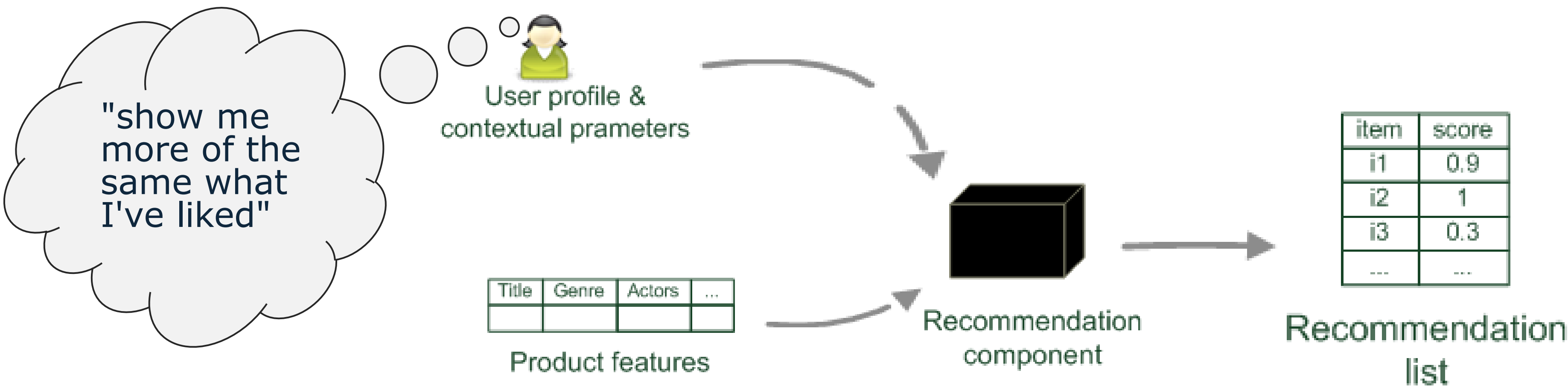
Content-based Recommendation

Content-based recommendation

- While CF – methods do not require any information about the items,
  - it might be reasonable to exploit such information; and
  - recommend fantasy novels to people who liked fantasy novels in the past
- What do we need:
  - some information about the available items such as the genre ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- The task:
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Content-based recommendation



Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

What is the "content"?

- Most CB-recommendation techniques were applied to recommending text documents.
  - Like web pages or newsgroup messages for example.
- Content of items can also be represented as text documents.
  - With textual descriptions of their basic characteristics.
  - Structured: Each item is described by the same set of attributes

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Content representation and item similarities

- **Item representation**

Title	Genre	Author	Type	Price	Keywords
The Night of the Gun	Memoir	David Carr	Paperback	29.90	Press and journalism, drug addiction, personal memoirs, New York
The Lace Reader	Fiction, Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
Into the Fire	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism

- Simple approach
  - Compute the similarity of an unseen item with the user profile based on the keyword overlap (e.g. using the *Jaccard index*)

$$\frac{|\boldsymbol{keywords}(A) \cap \boldsymbol{keywords}(B)|}{|\boldsymbol{keywords}(A) \cup \boldsymbol{keywords}(B)|}$$

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Jaccard Index

$$\boldsymbol{Jaccard\ Index} = \frac{|\boldsymbol{keywords}(\boldsymbol{A}) \cap \boldsymbol{keywords}(\boldsymbol{B})|}{|\boldsymbol{keywords}(\boldsymbol{A}) \cup \boldsymbol{keywords}(\boldsymbol{B})|}$$

- e.g.
  - Book A contains "Thailand, Bangkok, Market, Tourist, Temple, Cuisine".
  - Book B contains "Thailand, Bangkok, Temple, Museum".

$$Jaccard\ Index = \frac{3}{7}$$

Limitations

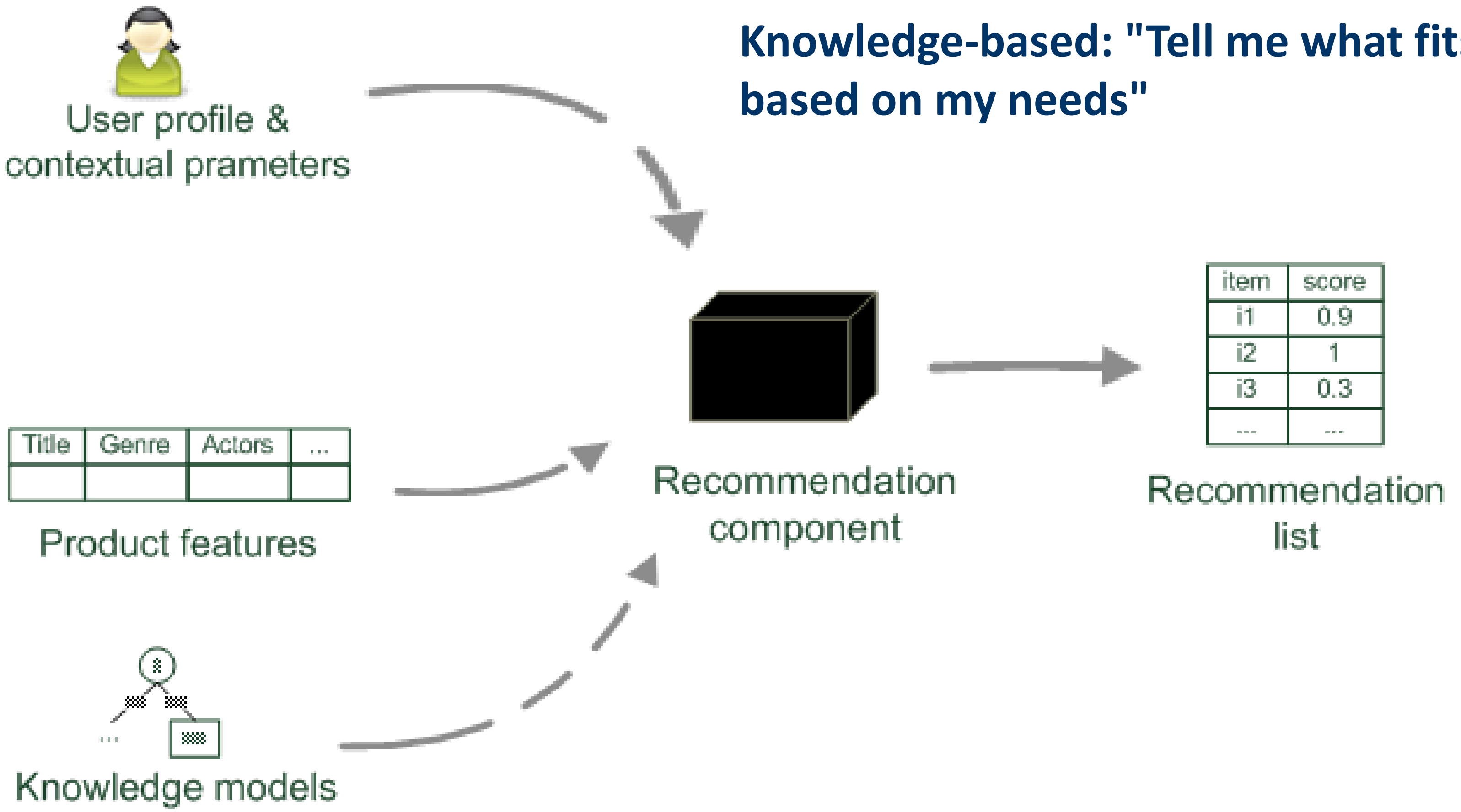
- Keywords alone may not be sufficient to judge quality/relevance of a document or web page
  - up-to-date-ness, usability, aesthetics, writing style
  - content may also be limited / too short
  - content may not be automatically extractable (multimedia)
- Ramp-up phase required
  - Web 2.0: Use other sources to learn the user preferences
- Overspecialization
  - Or: too similar news items
- Pure content-based systems are rarely found in commercial environments

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Knowledge-based Recommendation

Knowledge-Based

Knowledge-based: "Tell me what fits based on my needs"



Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Why do we need knowledge-based recommendation?

- Products with low number of available ratings

- Time span plays an important role
  - five-year-old ratings for computers
  - user lifestyle or family situation changes
- Customers want to define their requirements explicitly
  - "the color of the car should be black"

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Knowledge-based recommender systems

- Constraint-based
  - based on explicitly defined set of recommendation rules
  - fulfill recommendation rules
- Case-based
  - based on different types of similarity measures
  - retrieve items that are similar to specified requirements
- Both approaches → conversational recommendation process
  - users specify the requirements
  - systems try to identify solutions
  - if no solution can be found, users change requirements

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Constraint-based recommendation problem

- Select items from this catalog that match the user's requirements

id	price(€)	mpix	opt-zoom	LCD-size	movies	sound	waterproof
P_1	148	8.0	4×	2.5	no	no	yes
P_2	182	8.0	5×	2.7	yes	yes	no
P_3	189	8.0	10×	2.5	yes	yes	no
P_4	196	10.0	12×	2.7	yes	no	yes
P_5	151	7.1	3×	3.0	yes	yes	no
P_6	199	9.0	3×	3.0	yes	yes	no
P_7	259	10.0	3×	3.0	yes	yes	no
P_8	278	9.1	10×	3.0	yes	yes	yes

- User's requirements can, for example, be
  - "the price should be lower than 300 €"
  - "the camera should be suited for sports photography"

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Summary

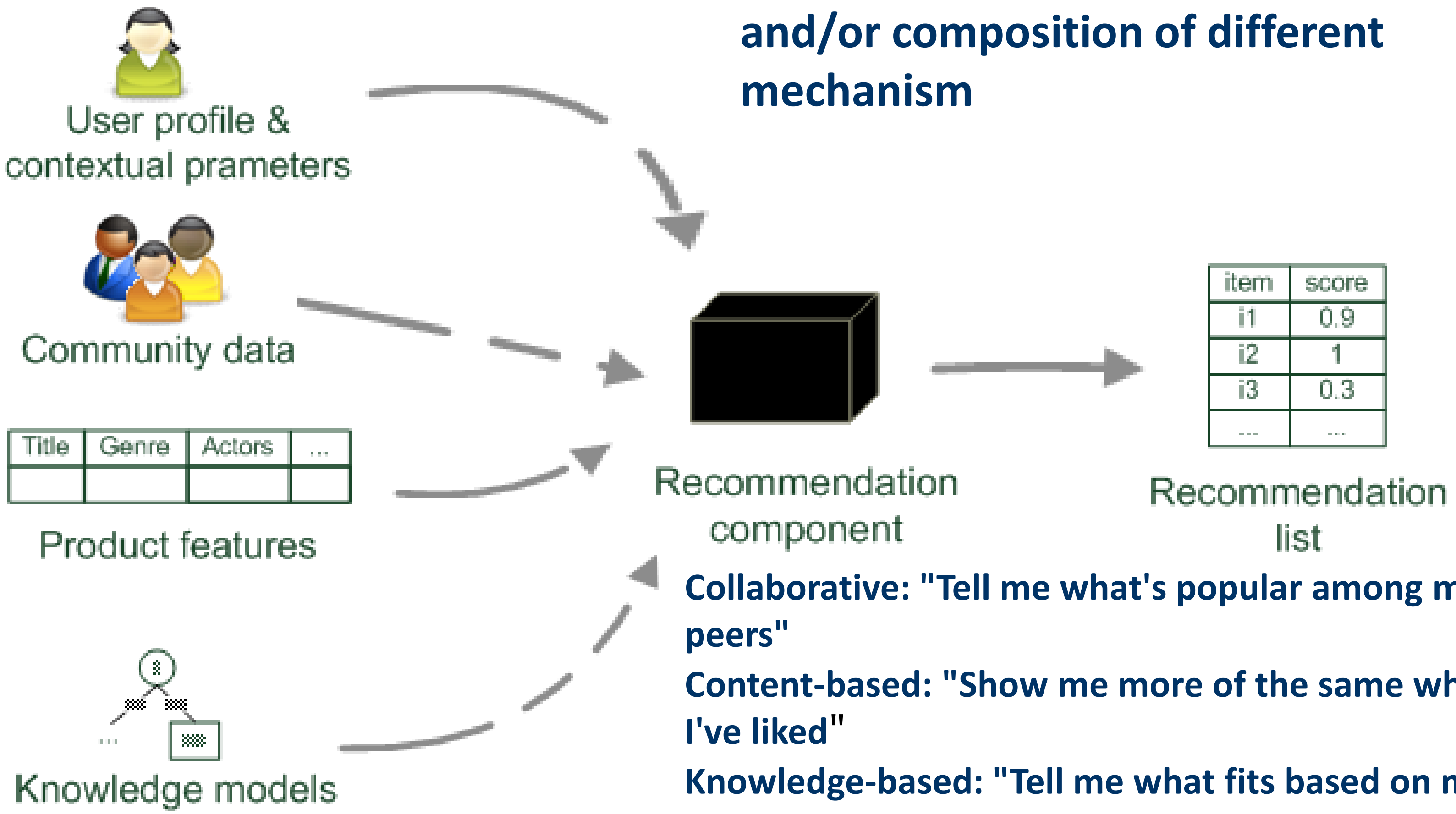
- Knowledge-based recommender systems
  - constraint-based
  - case-based
- Limitations
  - cost of knowledge acquisition
    - from domain experts
    - from users
    - from web resources
  - accuracy of preference models
    - very fine granular preference models require many interaction cycles
    - collaborative filtering models preference implicitly
  - independence assumption can be challenged
    - preferences are not always independent from each other

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Hybrid Recommender Approach

Hybrid Recommender

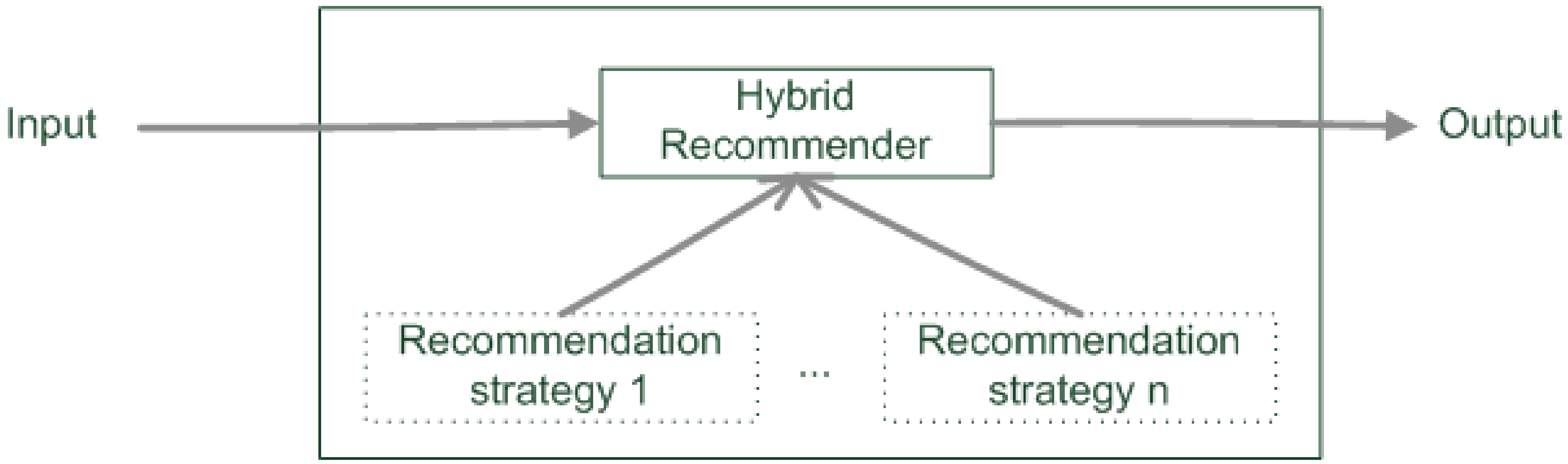
Hybrid: combinations of various inputs and/or composition of different mechanism



Collaborative: "Tell me what's popular among my peers"
Content-based: "Show me more of the same what I've liked"
Knowledge-based: "Tell me what fits based on my needs"

Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Abstract



Ref: • Dietmar J., et al.: An Introduction to Recommender Systems. Cambridge, 2010

Weighted

- The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.

Recommendation Score =

w1 x *Score_from_Algorithm_1* +

w2 x *Score_from_Algorithm_2* +

. . . .

wN x *Score_from_Algorithm_N*

Ref: • Burke, R.: "Hybrid Recommender Systems: Survey and Experiments", 2002

Switching

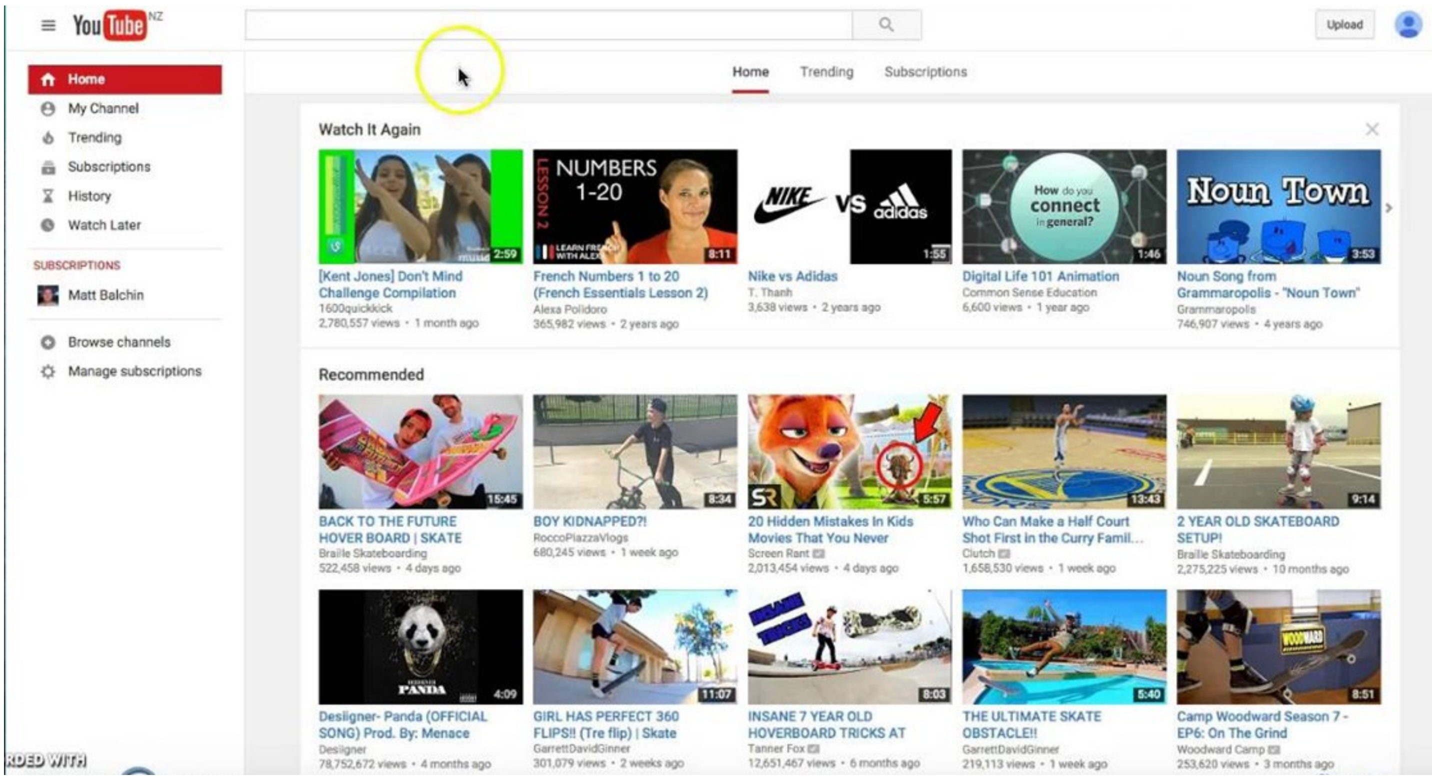
- The system switches between recommendation techniques depending on the current situation.

```
If (data is AAA ) {
        use Algorithm_A ;
}else if (data is BBB ){
        use Algorithm_B ;
else{ use Algorithm_C ;
```

Ref: • Burke, R.: “Hybrid Recommender Systems: Survey and Experiments”, 2002

Mixed

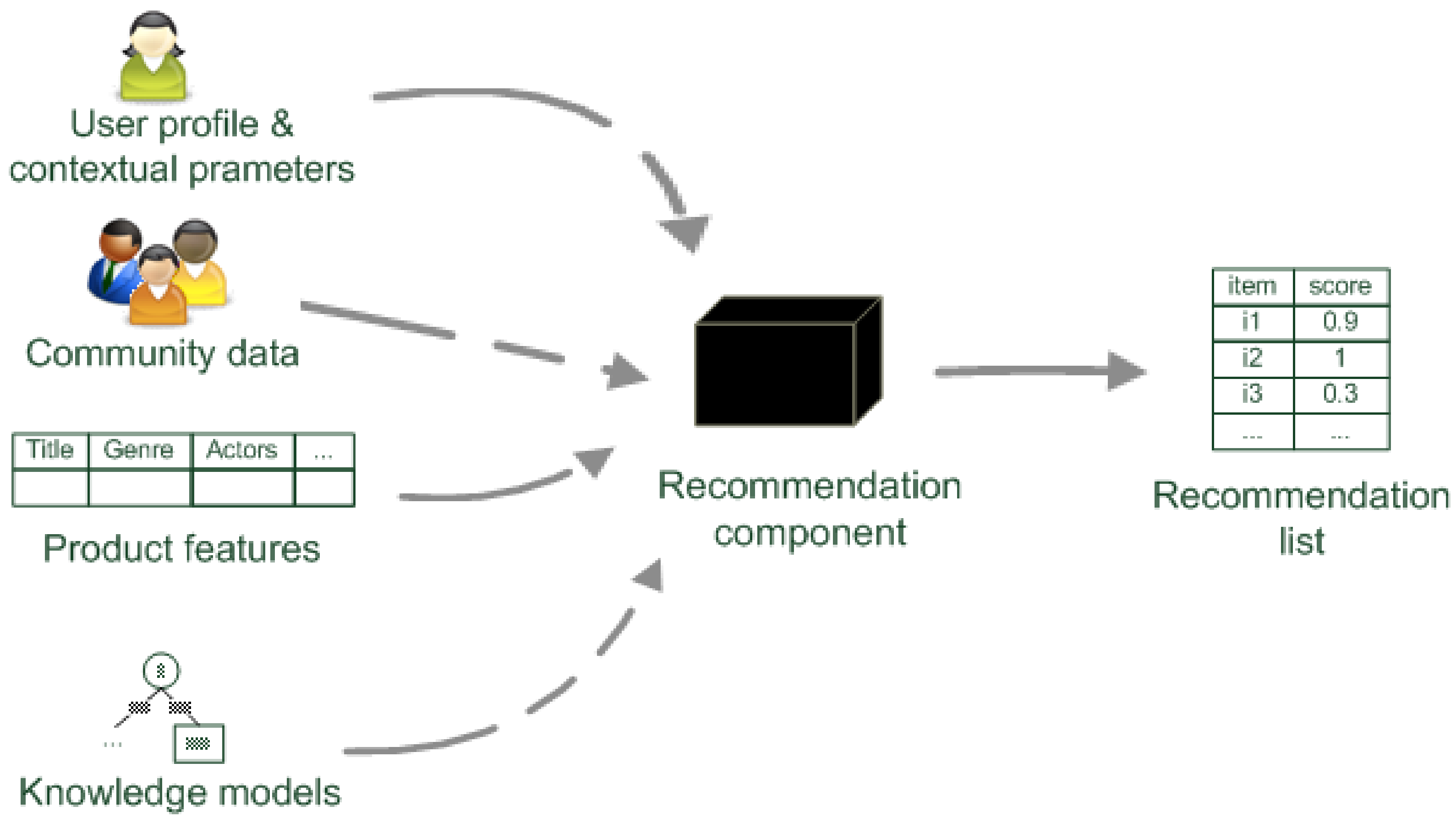
- Recommendations from several different recommenders are presented at the same time



Ref: • Burke, R.: "Hybrid Recommender Systems: Survey and Experiments", 2002

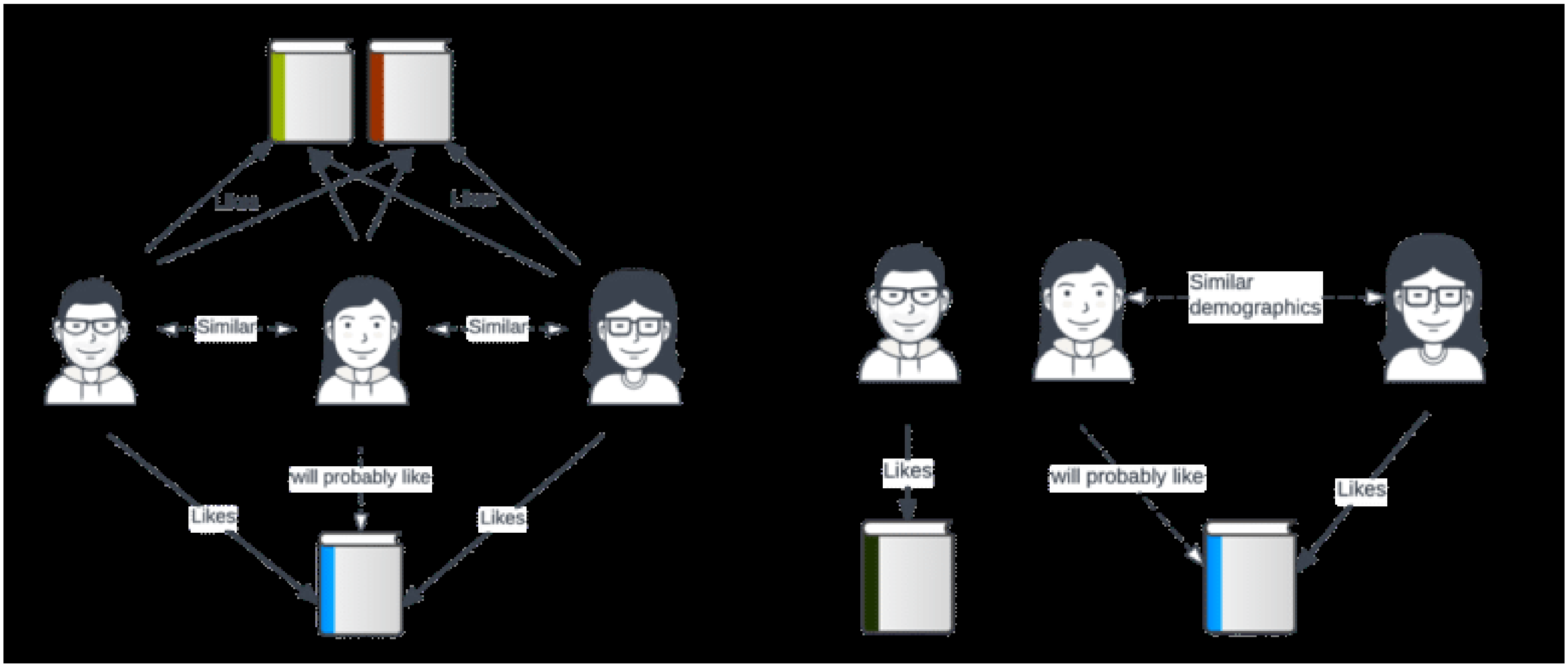
Summary

Summary



Ref: • Chulyadyo, R.: "A new horizon for the recommendation: Integration of spatial dimensions to aid decision making", 2016

Summary

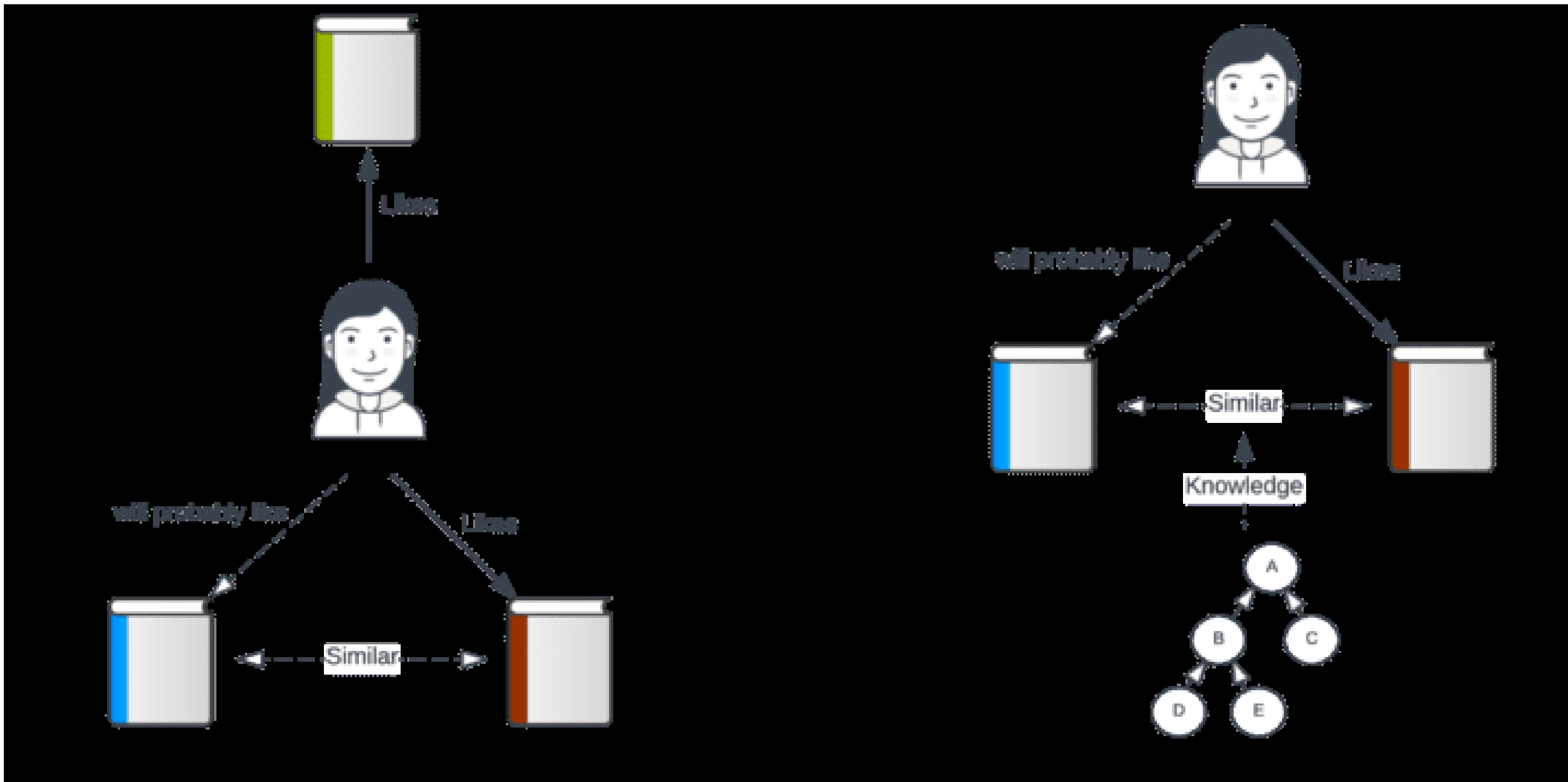


Collaborative

Demographics-Based

Ref: • Chulyadyo, R.: “A new horizon for the recommendation: Integration of spatial dimensions to aid decision making”, 2016

Summary

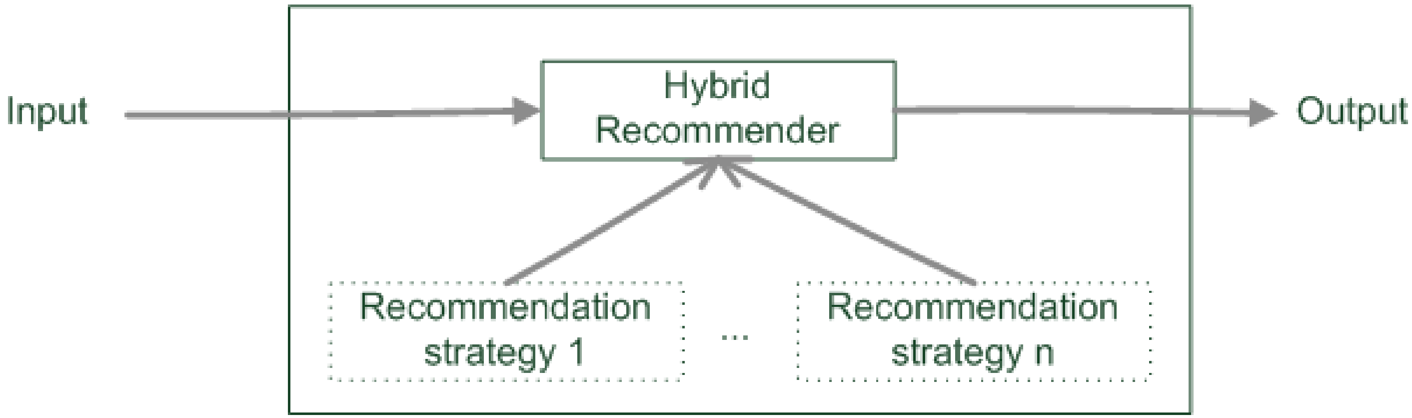


Content-Based　　　　Knowledge-Based

Ref: • Chulyadyo, R.: “A new horizon for the recommendation: Integration of spatial dimensions to aid decision making”, 2016

Summary

Hybrid Recommender Systems



- Weighted
- Switching
- Mixed

Machine learning allows us to build software solutions that exceed human understanding and shows us how AI can innervate every industry.

Steve Jurvetson

つづく