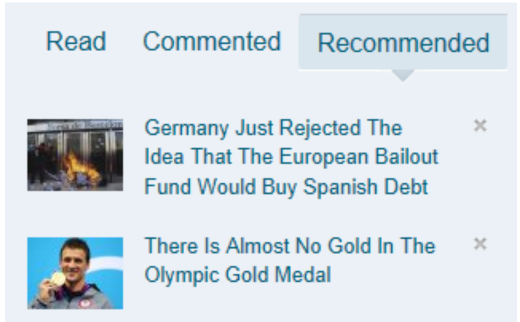
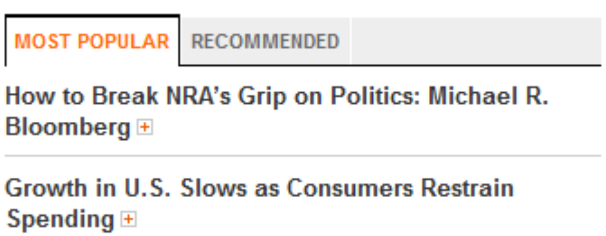
Recommender Systems

Elaborated from IJCAI tutorial by Dietmar Jannach, TU Dortmund and **Gerhard Friedrich**
Alpen-Adria Universität Klagenfurt

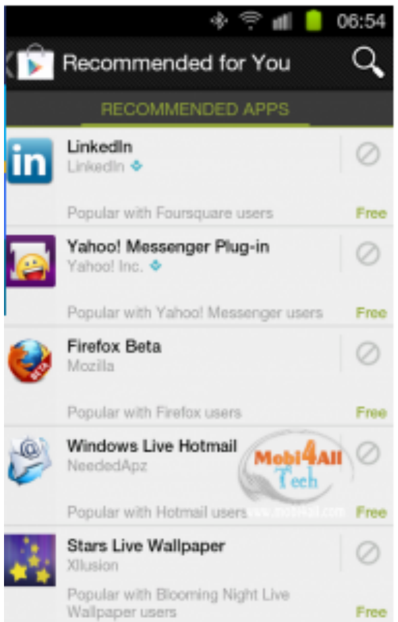
Recommender Systems

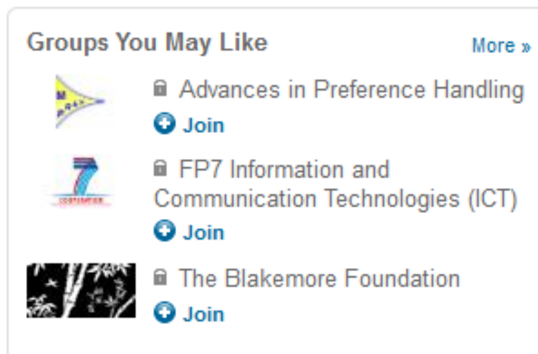
- ## Application areas

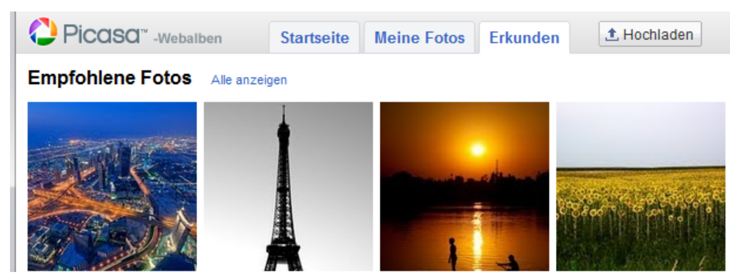




In the Social Web







Syllabus

- **What are recommender systems for?**
  - Introduction
- **How do they work (Part I) ?**
  - Collaborative Filtering
- **How do they work (Part II) ?**
  - Content-based Filtering
  - Knowledge-Based Recommendations
- **How to measure their success?**
  - Evaluation techniques

Introduction

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

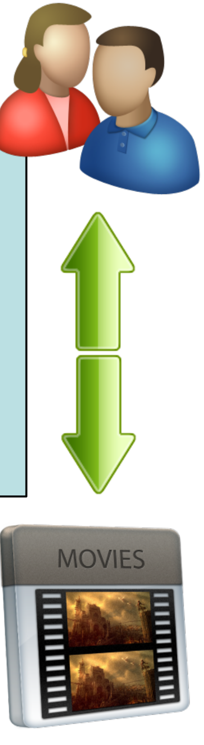
- **Myths from industry**
  - Amazon.com generates X percent of their sales through the recommendation lists (**30 < X < 70**)
  - Netflix (DVD rental and movie streaming) generates X percent of their sales through the recommendation lists (30 < X < 70)
- **There must be some value in it**
  - See recommendation 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 a very hot research topic!!

Problem domain

- **Recommendation systems (RS) help to match users with items**
  - reduce information overload
  - Sales assistance (guidance, advisory, persuasion,…)

RS are software agents that elicit the interests and preferences of individual consumers […] and make recommendations accordingly.
They have the potential to support and improve the quality of the decisions consumers make while searching for and selecting products online.

» [Xiao & Benbasat, MISQ, 2007]



- **Different system designs / paradigms**
  - Based on availability of exploitable data
  - Implicit and explicit user feedback
  - Domain characteristics

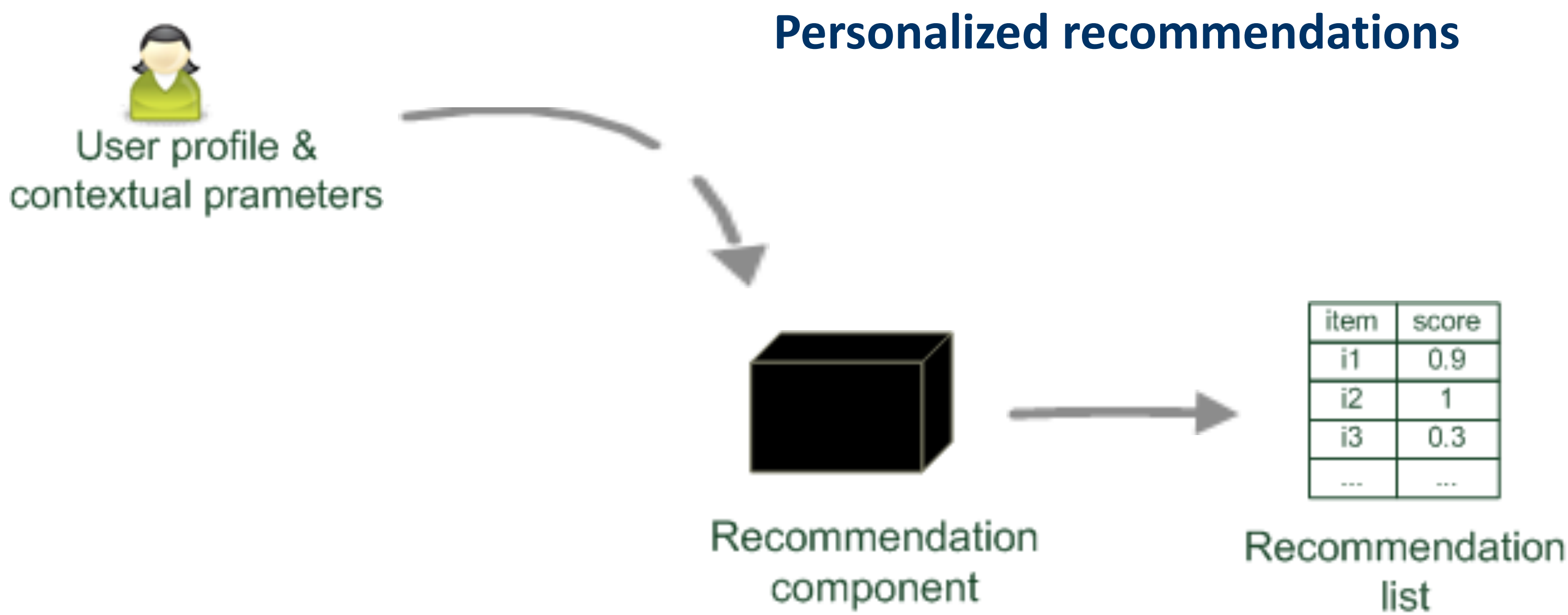
Recommender systems: task definition

- **Given:**
  - User model and profile (e.g. ratings, preferences, demographics, situational context)
  - Items (with or without description of item characteristics)
- **Find:**
  - Relevance score for items. Used for ranking.
- **Purpose:**
  - Recommend items that are assumed to be relevant for the user
- **But:**
  - Remember that “relevance” might be context and user dependent (recommending songs is not useful in a tourism context)
  - Characteristics of the recommendation itself might be important (saliency, diversity: see later)

Saliency and diversity : first intuitive definition

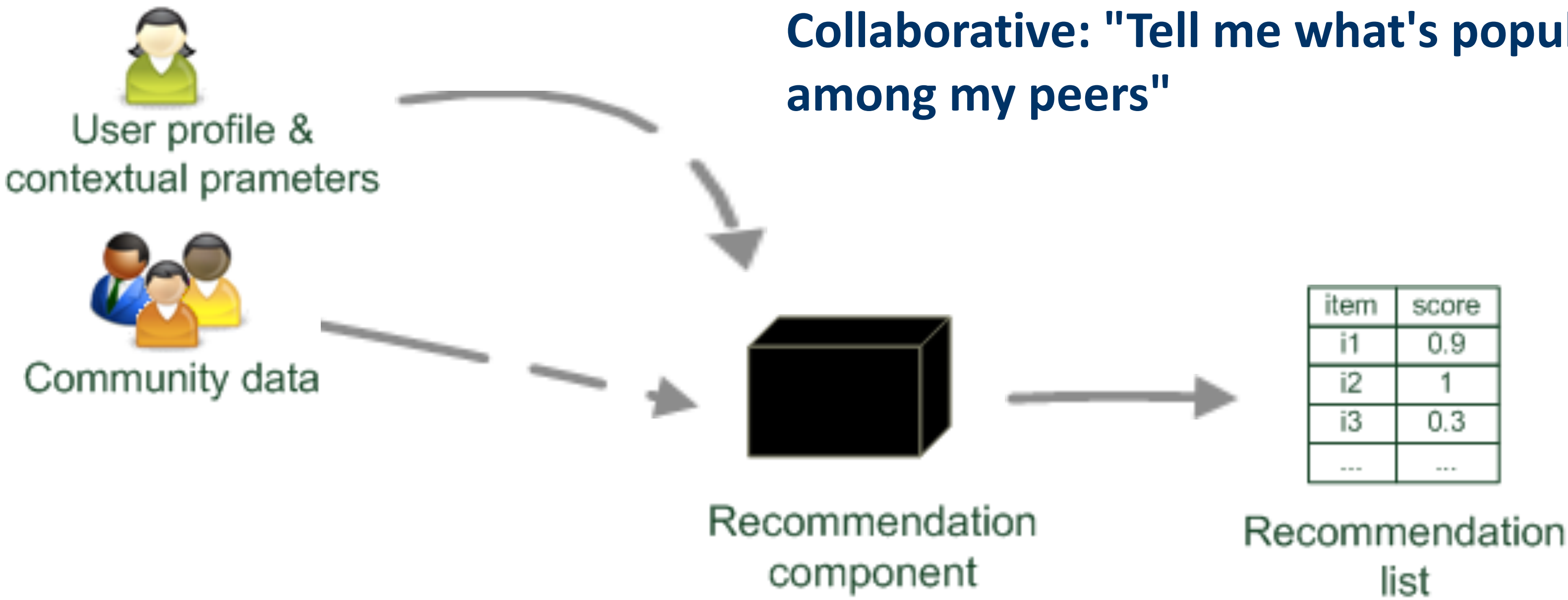
- A recommended item is SALIENT is it is truly relevant wrt a user's needs
- A recommended item is "diverse" or serendipitous IF is is also "unexpected" – we should not recommend what is obvious
- We will see later how to formally measure saliency and serendipity

Paradigms of recommender systems



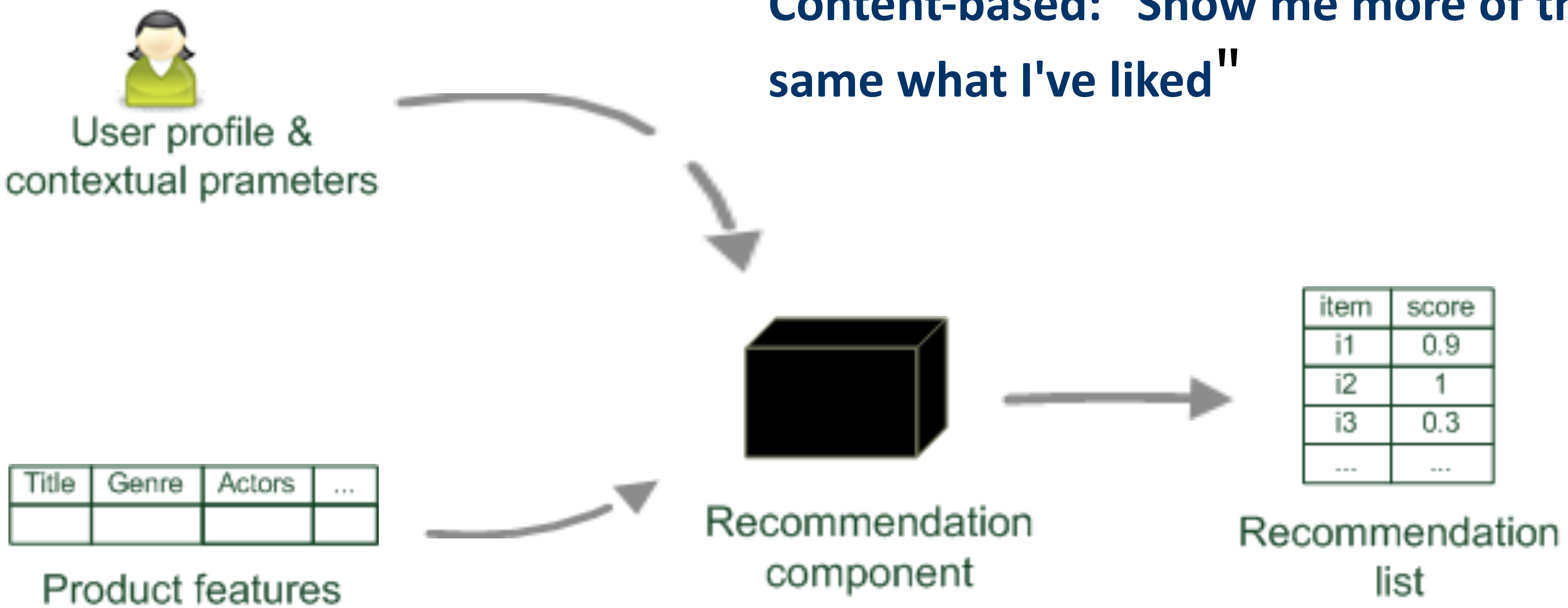
Paradigms of recommender systems

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



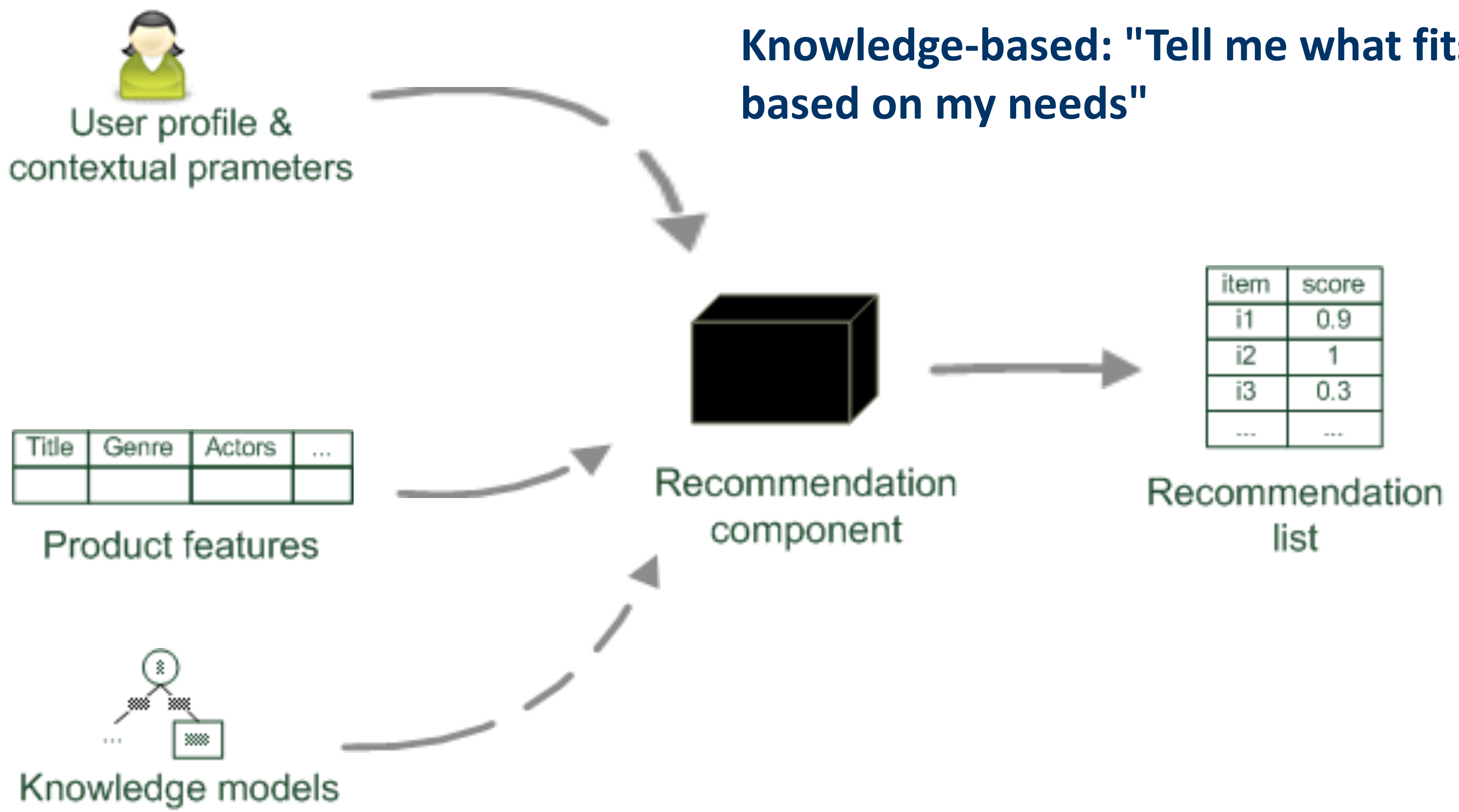
Paradigms of recommender systems

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



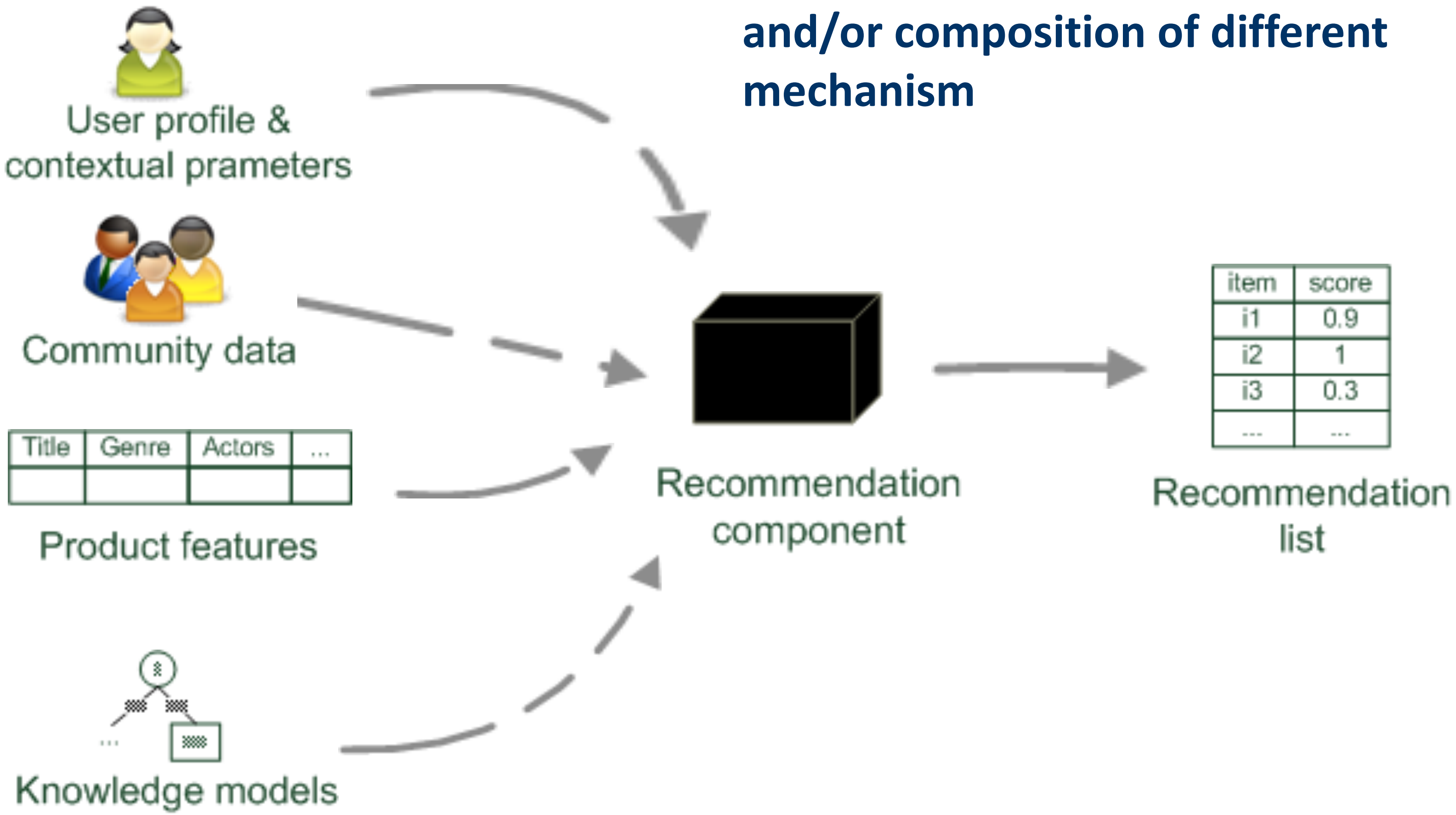
Paradigms of recommender systems

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	Content descriptions necessary, cold start for new users, no surprises
Knowledge-base	old-	Knowledge engineering effort to b n (u le

Unexpectedness of what is recommended wrt previous user's choices

Difficult unless you already collected much information about other users

Memory-based (user-based) and model-based (item-based) collaborative approaches

- **User-based recommenders are said to be "memory-based"**
  - the rating matrix is directly used to find “similar” users to make predictions at run time
  - does not scale for most real-world scenarios (unless we know something about the users, other than the previous purchases)
  - large e-commerce sites (Amazon, Netflix) have tens of millions of customers and millions of items (but they are just a few companies, while many companies are interested in recommending but have cold-start problem)
- **Model-based CF 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 (recently, deep ML models)
  - model-building and updating can be computationally expensive

Collaborative Filtering
a.k.o. memory-based

Collaborative Filtering (CF)

- **The most prominent approach to generate recommendations**
  - used by large, commercial e-commerce sites (eg, Amazon)
  - 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

User-based nearest-neighbor collaborative filtering (1)

- **The basic technique:**
  - 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.:
    - 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

	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

User-based nearest-neighbor collaborative filtering (2)

- **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

Measuring user similarity

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

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

a, b : users

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

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

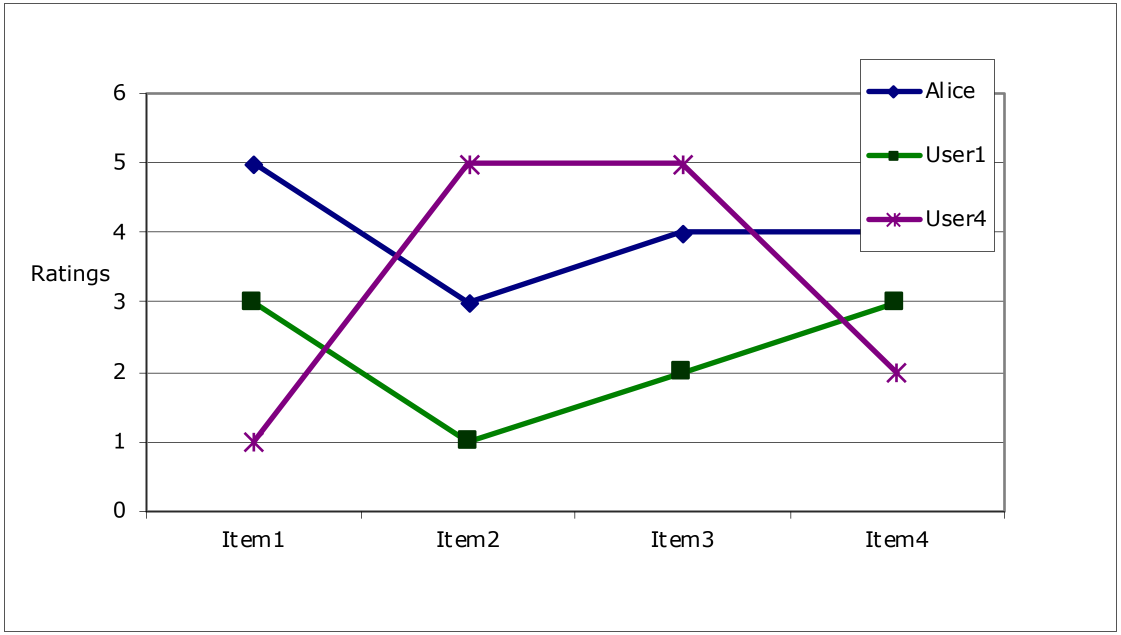
Possible similarity values between -1 and 1; $\bar{r}_a, \bar{r}_b$ = user's average ratings

	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 = 0,85
sim = 0,70
sim = -0,79

Pearson correlation

- **Takes differences in rating behavior into account**

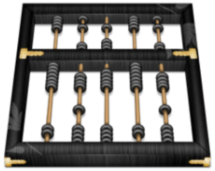


- **Works well in usual domains, compared with alternative measures**
  – such as cosine similarity

Generating recommendations (2)

- A common prediction function (*"will user a buy product p?"):*

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



- Calculate, whether the other users' ratings for the unseen item ***i*** are higher or lower than their average
- Combine the rating differences – use the similarity as a weight
- **Add/subtract the users' bias from the active user's average** and use this as a prediction

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 of prediction 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
  - More recently, **social recommenders** use social relations (e.g. friendship) to select “similar” users rather than the full set of users

Item-based recommenders

A.K.O. MODEL-BASED RECOMMENDERS

Item-based CF approaches

- **Basic idea: "Item-based CF exploits relationships between items first, instead of relationships between users”**
- **Relation between items can be computed off-line (model-based approach)**
- **Item similarities are supposed to be more stable than user similarities**

Item-based collaborative filtering

- **Basic idea:**
  - Use the similarity between items (and not users) to make predictions
  - But we need to know something about the items (item descriptions, categories..)
- **Example:**
  - Look for items that are similar to Item5 (as for rating)

Item5

	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

The cosine similarity measure

- **Ratings are seen as vector in n-dimensional space**
- **Similarity is calculated based on the cosin-similarity (or jaccard)**

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

- **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 (note: now a and b are items, u are users)

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

Note in comparison to previous user-based formula **here we vary users, not items**

Pre-processing for item-based filtering

- **Item-based filtering does not solve the scalability (sparse matrix) 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
- **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)

More on ratings

- **Pure CF-based systems only rely on the rating matrix**
- **Explicit ratings**
  - Most commonly used in e-commerce
  - Research topics
    - Augmenting available information with social data, knowledge bases, ecc
    - Extend to multi-domain (rather than just one single domain, e.g. movies, books..)
- **Challenge: the cold start problem**
    - Users not always willing to rate many items; **sparse** rating matrices
    - What if we have a new user? What if we have just few users and can't reliably compute similarities?

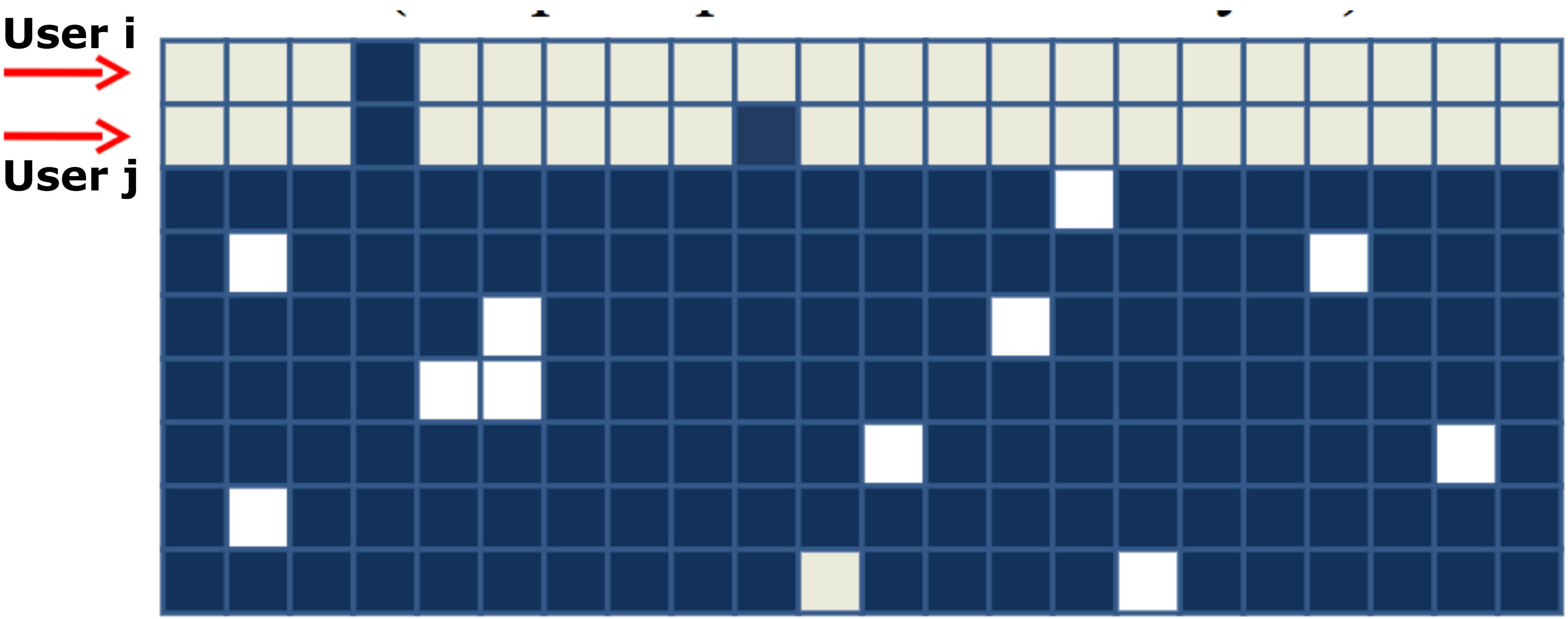
The cold start problem

- **Cold start users**: if a user is new, and we have no or little information about his/her interests, therefore we cannot reliably find his/her "similar ones"
- **Cold start item**: if we have a new item, the user-item table includes no info about the appreciation of this item by other users
- **Cold start user-item table (sparsity problem)**: only big players have millions of rating on millions of items, like Amazon. Smaller companies have a very "sparse" user-item matrix, which limits the effectiveness of the simple collaborative mechanism that we previously introduced
- The cold-start problem is mitigated in different ways depending on the approach (collaborative vrs content-based)

Problems with user-based collaborative filtering (1)

- **User Cold-Start problem (empty rows)**

not enough known about new users, to decide who is similar to whom



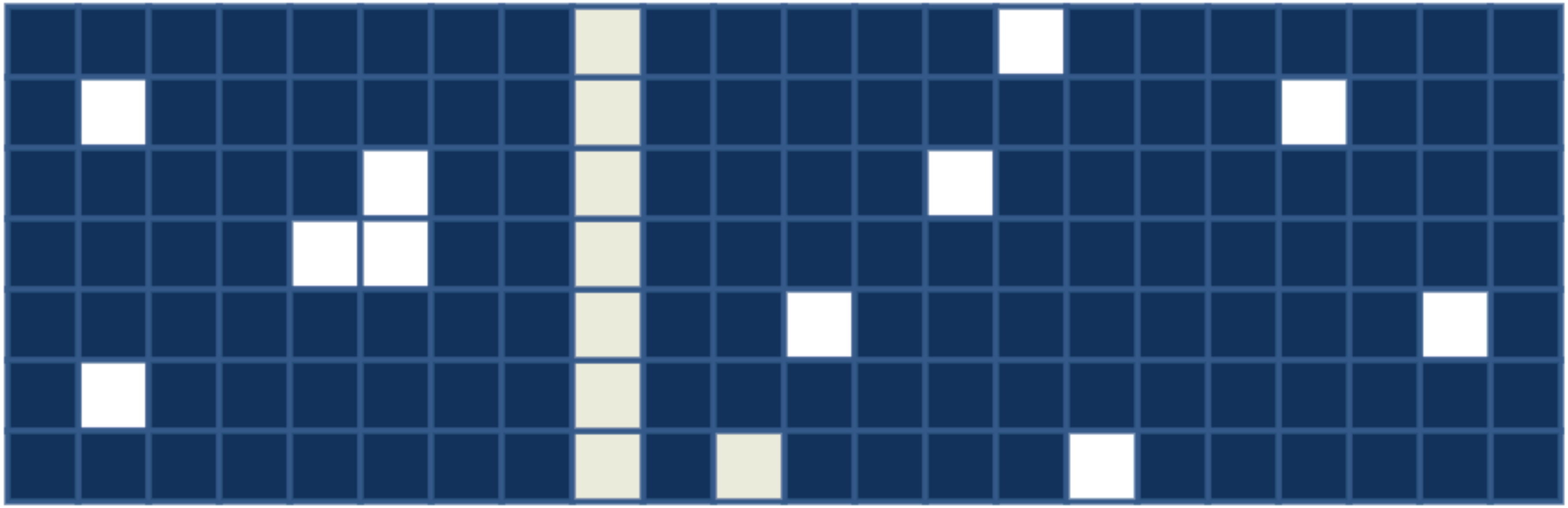
*** White cells are empty cells**

Problems with collaborative filtering (2)

- **Item Cold-Start problem (empty columns)**

– Cannot predict ratings for new item until some similar users have rated it

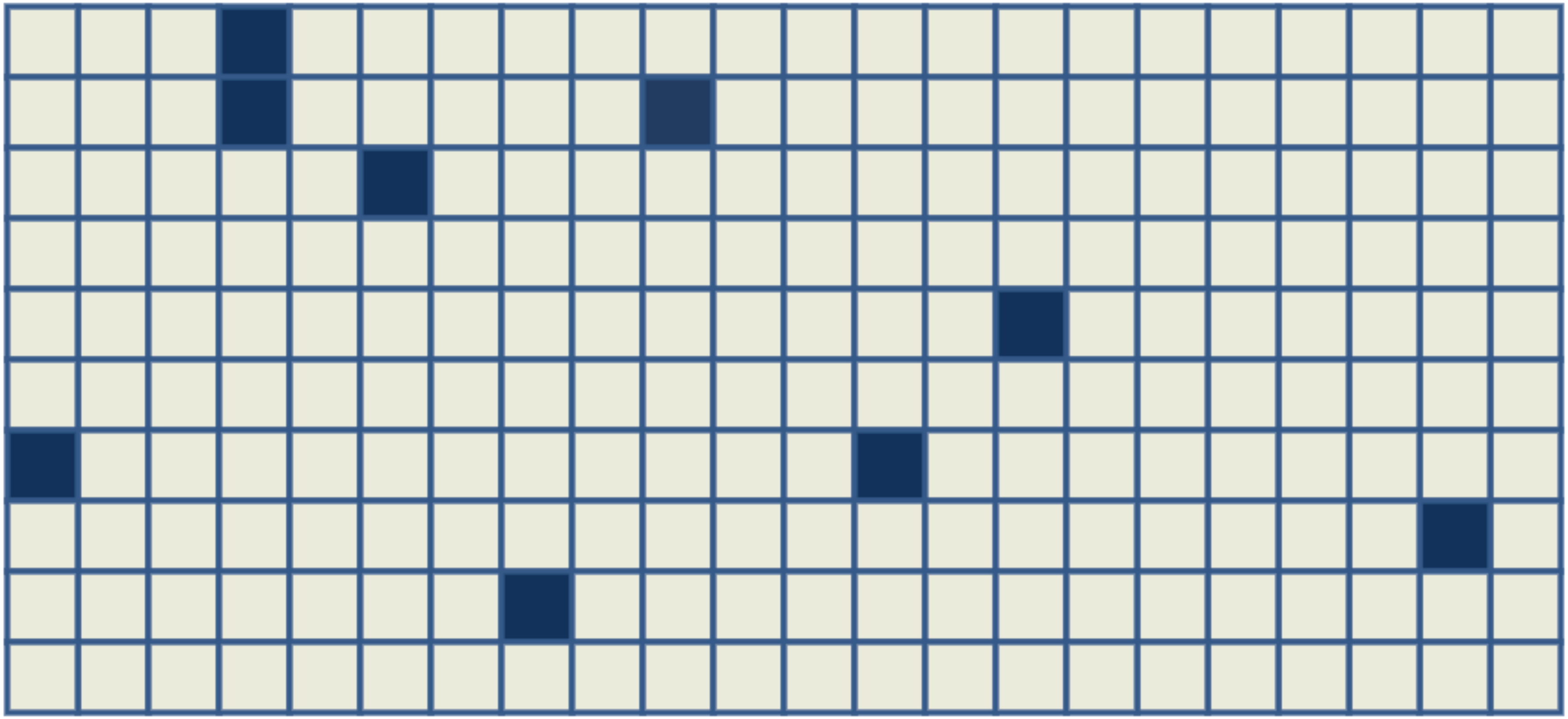
Item k



Problems with item-based collaborative filtering (3)

- **Sparsity (many zeros in user-item matrix)**

when recommending from a large item set, users will have rated only some of the items (makes it hard to find similar users)



Cold start problem in “pure” collaborative filtering systems

- **Collaborative filtering – when the only information available is the user-item matrix M, can hardly cope with cold start problems.**
- **Algorithmic solutions are available for the sparsity problem, when we have a very sparse matrix M, with few item ratings.**
- **Algebraic solutions to mitigate sparsity**
  - **Matrix factorization (**singular value decomposition, principal component (principal eigenvector) analysis (*you know this already..)*
  - **Association rule mining** *(you should know from ML course..)*
    - (extract rules from data e.g. IF I_a&I_b THEN I_c)
  - **Probabilistic models**
    - clustering models, Bayesian networks, probabilistic Latent Semantic Analysis
  - Various other machine learning approaches, including **deep methods**
  - **Most recently, augmentation with social data or knowledge bases**

Example: Dimensionality Reduction /Matrix factorization

- **Singular Value Decomposition for dimensionality reduction of rating matrices**
  - SVD is a form of clustering: detects latent dimensions in user/item matrix
  - Captures important factors/aspects and their weights in the data
  - Assumption is that k dimensions capture the “semantic” signals and filter out noise
- **General Method:**
  - The past ratings can be represented as a (sparse) matrix M . Through matrix factorization, one can learn a **low-dimensional latent vector $\mathbf{u_i}$** for each user and a **low-dimensional latent vector $\mathbf{v_j}$** for each item.
  - User u ’s rating on item j can be predicted as $\mathbf{u^T_i v_j}$, where $\mathbf{u_i}$ and v_j are the low-dimensional vectors associated with user i and item j , respectively.

Matrix factorization example

Decompose the (sparse) user-item matrix M into two (dense) matrixes – the project of users (items) onto a dense item (user) latent space.

$$M = U^T\mathbf{V}$$

Rating Matrix (User × Item)

User \ Item	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

=

User Matrix

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

X

Item Matrix

W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

Alternative: SVD dimensionality reduction

Start from user/item rating matrix M and apply SVD with rank k approximation

- **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

Movies

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

How will Alice rate EPL?

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

= 3 + 0.84 = 3.84

Collaborative Filtering Issues (summary)

- **Pros:**
  - well-understood, works well in some domains, no knowledge engineering required
- **Cons:**
  - requires user community, has sparsity problems, no integration of other knowledge sources, no explanation of results
- **What is the best CF method?**
  - In which situation and which domain? Inconsistent findings; always the same domains and data sets; differences between methods are often very small (1/100)
- **How to evaluate the prediction quality?**
  - (will analyze later on)
- **What about multi-dimensional ratings?**

More recent approaches

- **Two additional major paradigms of recommender systems**
  - Content-based
  - Knowledge-based
- **In a sense, both can be grouped into a unique category of "augmented" recommenders**

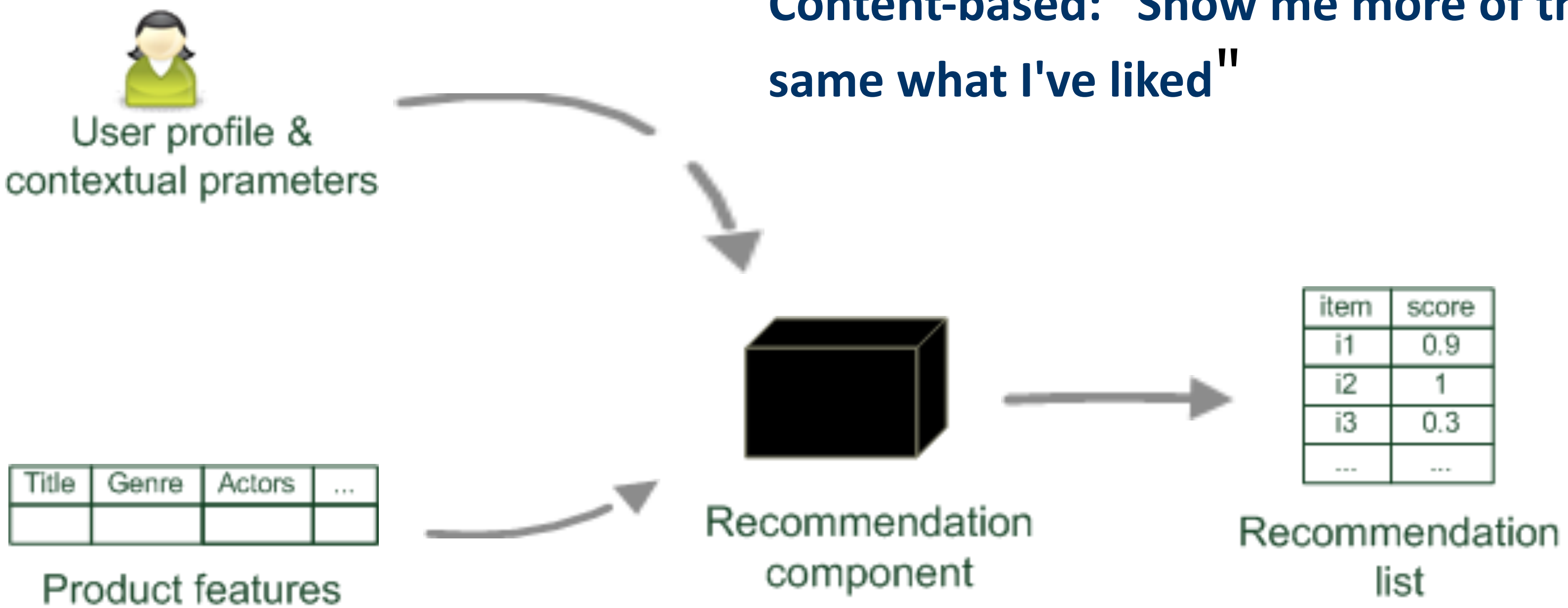
Content-based recommendation

Content-based recommendation

- **Collaborative filtering does NOT require any information about the items,**
  - However, it might be reasonable to exploit such information
  - E.g. recommend new “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 (for movies, books), or a short description (meta-data /structured /unstructured)
  - what the user likes in general (the preferences,profiles..)
- **The task:**
  - Learn user preferences, learn item descriptions (e.g., bag of words)
  - Locate/recommend items that are "similar" to the user preferences

Paradigms of recommender systems

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



What is the "content"?

- **Most CB-recommendation methods originate from Information Retrieval field:**
  - The item descriptions are usually automatically extracted (e.g., important words)
  - Or, we can extract descriptions from other sources (users' messages, wikipedia descriptions, movie databases..)
  - Goal is always to find and rank interesting items, but now items (and users) are associated with some textual description

- **If we have text, then classical IR methods can be used:**
  - Classical IR-based methods based on keywords
  - No expert recommendation knowledge involved
  - Users' preferred items are rather learned than explicitly elicited

Content-based systems provide a way to cope with sparsity

- **Implicit ratings: induce users' interests from other sources, e.g.:**
  - "topical" friends in social networks
  - Access to lists, groups, etc. (always in social networks)
  - Extract preferences from messages (e.g. for music: Spotify)
  - Other users' actions, clicks, page views, downloads..
  - Can be used **in addition to explicit ones**; question of correctness of interpretation

Content representation and item similarities (e.g., movies)

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
...					

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia Barry, Ken Follet, ..	Paperback	25.65	detective, murder, New York

- **Simple approach**
  - Compute the similarity of an unseen item with other items in the user profile based on the keyword overlap (e.g. using Jaccard)
  - $$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

Limitations of content-based recommendation methods

- **Keywords alone may not be sufficient to judge quality/relevance of a document or web page**
  - Up-to-dateness, usability, aesthetics, writing style
  - Content may also be limited / too short (this is often the case, exception are movies and books databases)
  - Content may not be automatically extractable (e.g., multimedia)
- **Ramp-up phase required**
  - Some **training data** is still required
  - Web 2.0: Use other sources to learn the user preferences
- **Overspecialization**
  - Algorithms tend to propose "more of the same"
  - E.g. too similar news items (low serendipity)

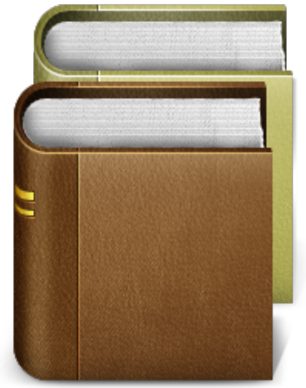
Social recommenders

- They use social content to improve recommendations
- For example, a user's friendship list
- Two users are similar if they share many friends – based on the notion of homophily (friends tend to share the same tastes)

$$sim(u, u') = cosin\_sim(F_u, F_{u'})$$

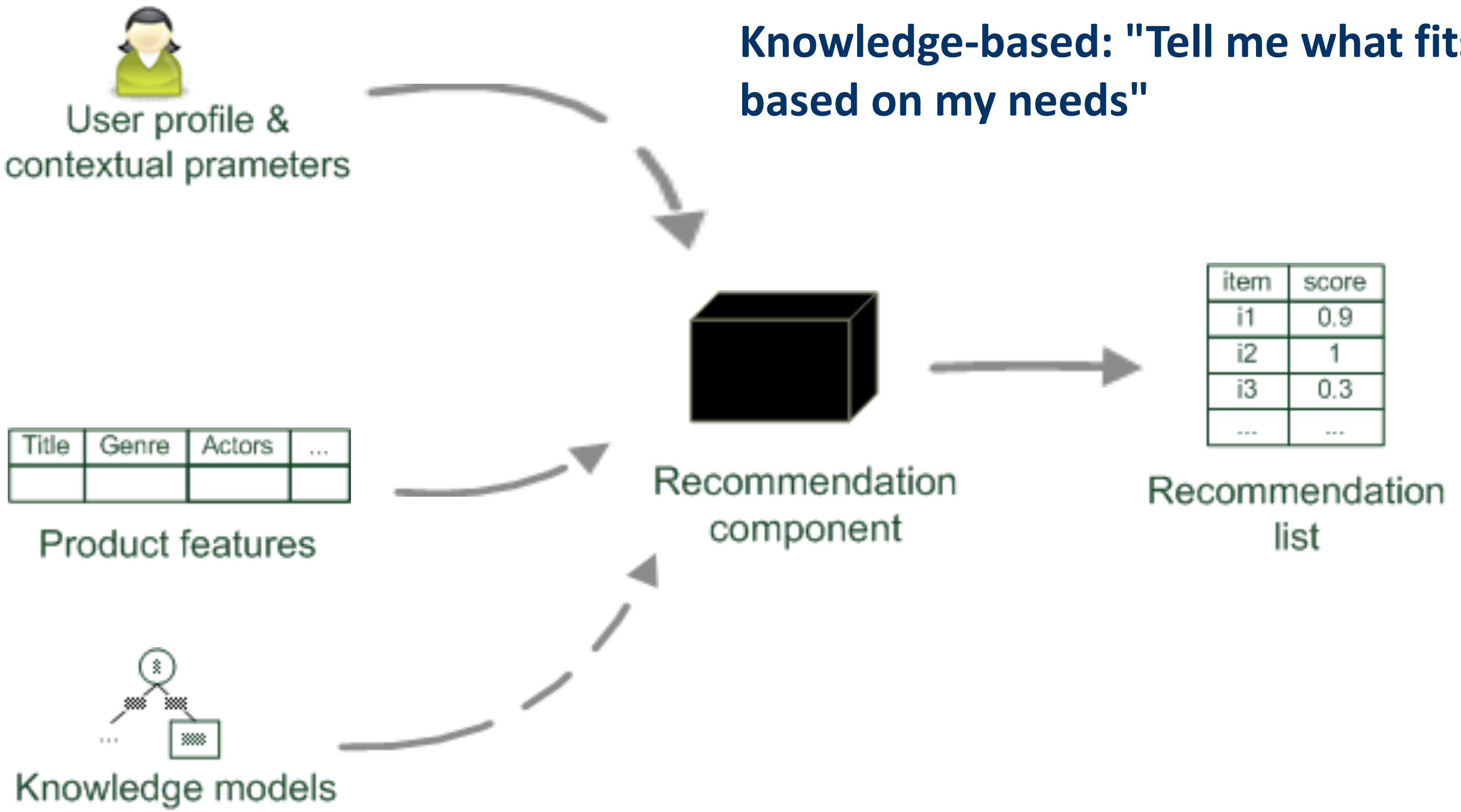
- Or we can use the Jaccard similarity
- Advantage: not dependent on keyword extraction
- Advantage: can solve the user cold-start problem: we can predict tastes of a brand new user exploiting knowledge on his/her similar-ones.

Knowledge-Based Recommender Systems



Knowledge-based recommendation

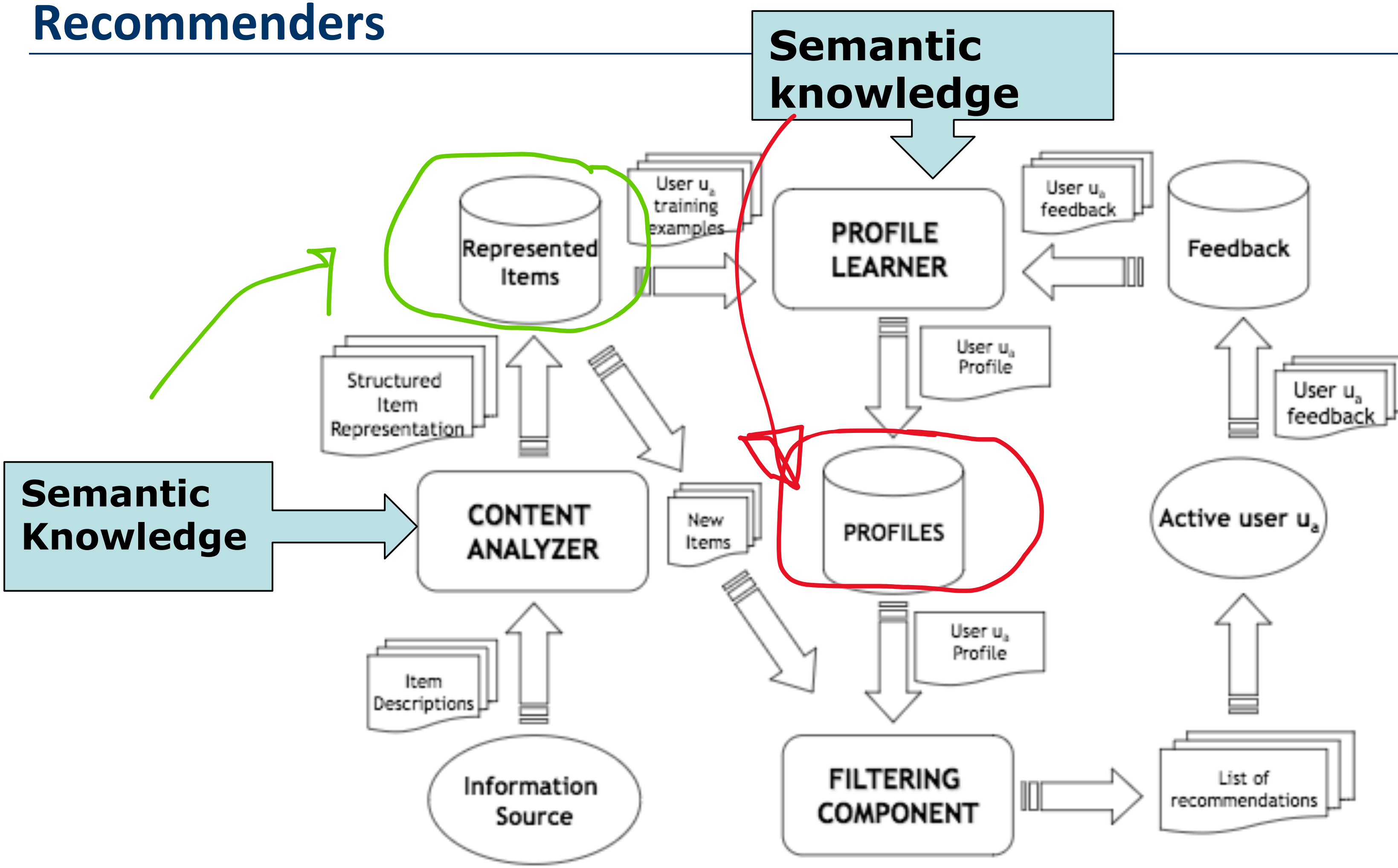
Knowledge-based: "Tell me what fits based on my needs"



Why semantic profiling is better?

- **Using semantics (categories rather than items) to represent users' interests enables**
  - the inference of incomplete information about users,
  - the generalization of their interests, and
  - the interplay among different domains.
- **For example, knowing that a user is interested in American television series (rather than observing that he/she likes Robin Wright, Aaron Paul and Homeland) may enable better recommendations on new series to follow (or movies with the same actors or genre), new social links to establish, the participation in related live events, the purchase of gadgets, and more.**
- **Furthermore, semantic interests solve the volatility problem: specific interests (e.g. the series *Homeland*) may change even frequently, while generalized interests are more stable.**

Architecture of Semantic Recommenders

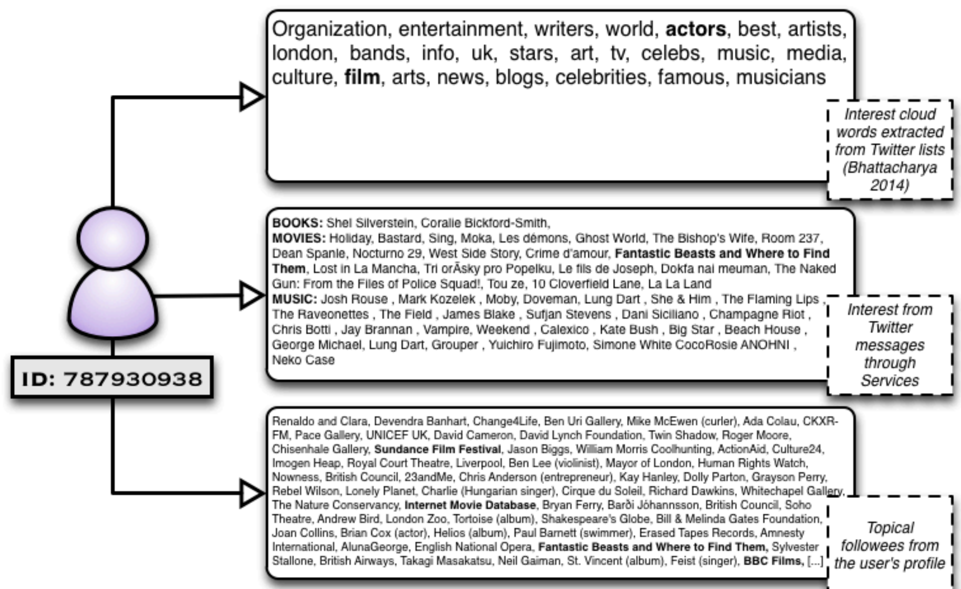


De Gemmis et al. 2015

Example of Semantic Profiles learning (1)

- Extract from users' messages, topical friendships, subscription to lists.. sets of named entities
- For example, non-reciprocated friendship with "popular" Twitter users

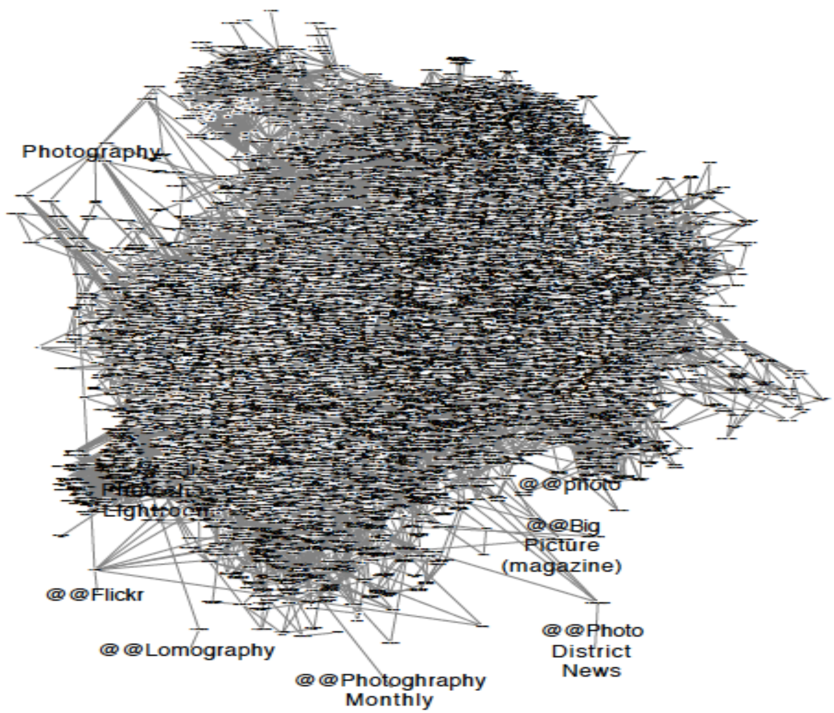
Example: primitive interests extracted from different source for the same users



Faralli et al. IJCAI 2015, Di Tommaso et al. ISWC 2018

Example of Semantic Profiles learning (2)

- **Map interests to Wikipedia articles (as for in Wiki-MED dataset)**
- **Consider the graph induced starting from these articles and travelling towards top categories of the Wikipedia category Graph**



Faralli et al. IJCAI 2018

h

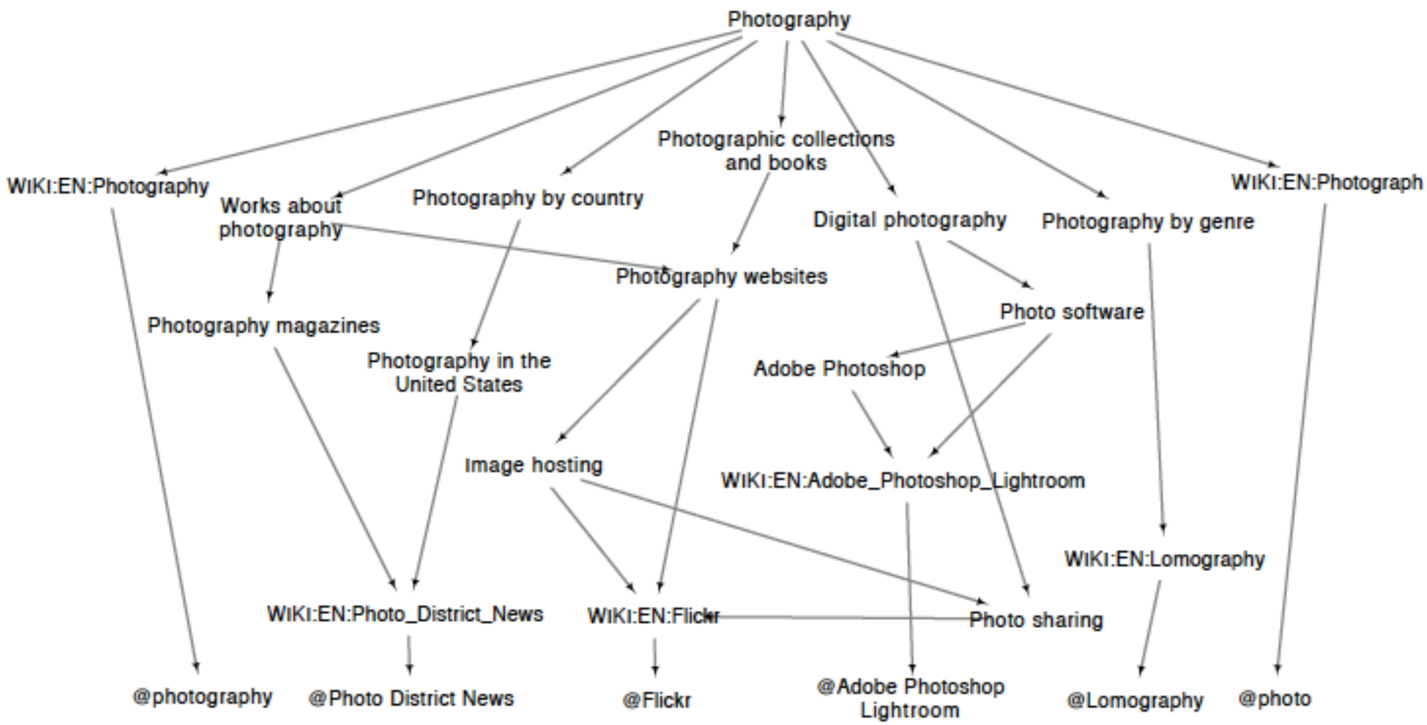
Example of Semantic Profiles learning (3)

- **Wikipedia Category Graph is highly ambiguous!**

Wikipage	1st level categories	2nd level categories	Top categories
John Turturro	People from Brooklyn, American television actors, American people of Italian descent, State University of New York at New Paltz alumni, American stage actors, Obie Award recipients, David di Donatello winners, Actors from New York City, American people of Sicilian descent, American film actors, Yale School of Drama alumni, Emmy Award winners	American people of Italian descent, Italy-United States relations, Brooklyn, People from New York City by occupation, American actors by medium, Theatre in the United States, Television award winners, People of Sicilian descent, Actors from New York, State University of New York alumni, People from New York City by borough, Film actors by nationality, **(20 more categories..)**	Geography, Humans, World, History, Information, Knowledge, Arts, Industry, Language, Employment, Technology, Education, Mind, Behavior, Structure, Culture, Nature, Humanities, Architecture, Government, People, Creativity, Systems, Environment, Politics
MIT Media Lab	Massachusetts Institute of Technology, MIT Media Lab, Modernist architecture in Massachusetts, Fumihiko Maki buildings	Massachusetts Institute of Technology, Land-grant universities and colleges, Modernist architecture in the United States by state, Universities and colleges in Cambridge, Buildings and structures by Japanese architects, Architecture in Massachusetts, Postmodern architecture by architect, Universities and colleges in Massachusetts, Technical	Geography, Humans, Science, World, History, Knowledge, Arts, Industry, Education, Technology, Employment, Mind, Agriculture, Behavior, Structure, Culture, Nature, Humanities, Architecture, Government, People, Universe, Systems, Creativity, Environment

Example of Semantic Profiles learning (4)

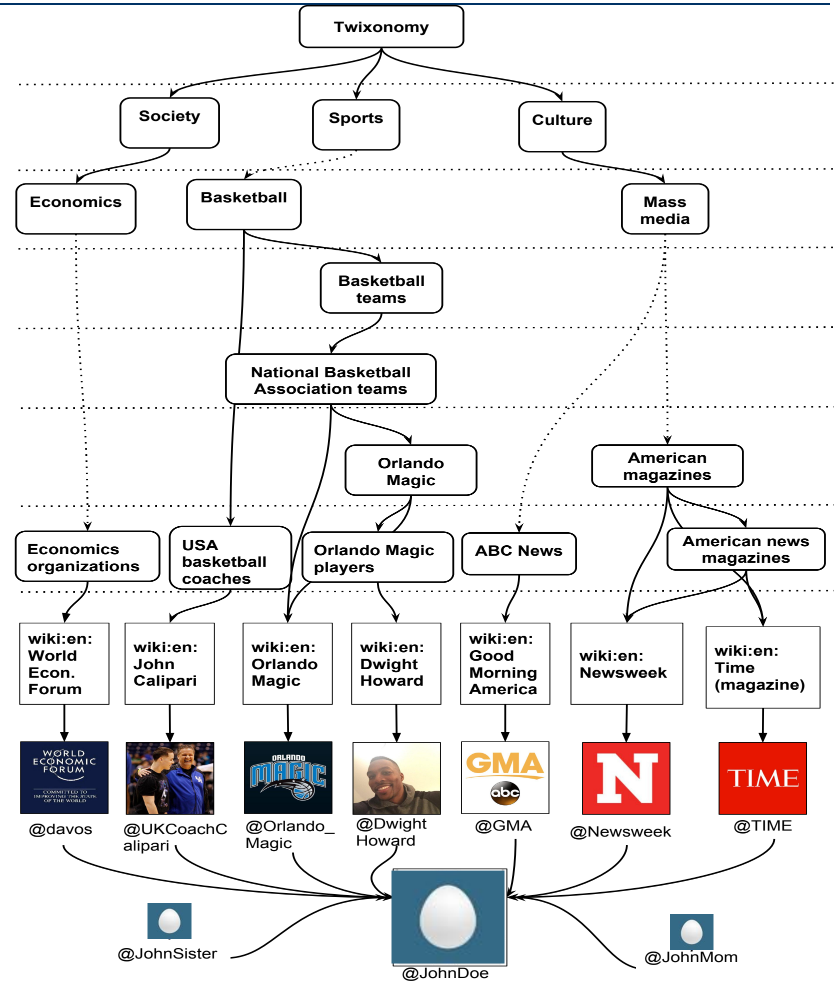
- Algorithm for bottom-up efficient pruning of the category Graph



Faralli et al. IJCAI 2018, KBS 2018

Example of Semantic Profiles learning (5)

- **The final result is a semantic profile that can be used for recommending items**

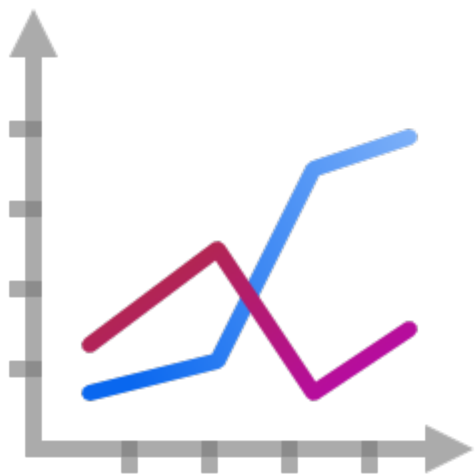


Faralli et al. IJCAI 2018, KBS 2018

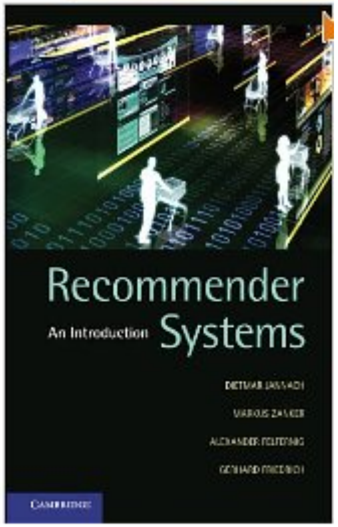
Readings

- Bobadilla et al. , Recommender systems survey Knowledge-Based Systems, Volume 46, July 2013, Pages 109–132
- Su Mon Kywe et al. A survey of recommender systems in twitter SocInfo'12 Proceedings of the 4th international conference on Social Informatics Pages 420-433, 2012
- Codina & Ceccaroni Taking Advantage of Semantics in Recommendation Systems Proceedings of the 2010 conference on Artificial Intelligence Research and Development: Pages 163-172
- DiTommaso&Faralli&Stilo&Velardi "Wiki-MID: a very large Multi-domain Interests Dataset of Twitter users with mappings to Wikipedia" ISWC 2018
- Finocchi&Faralli&Ponzetto&Velardi "Efficient Pruning of Large Knowledge Graphs" IJCAI 2018
- PLUS ALL THOSE SENT TO YOUR GOOGLE GROUP (after 2018)

Evaluation of Recommender Systems



Recommender Systems in e-Commerce

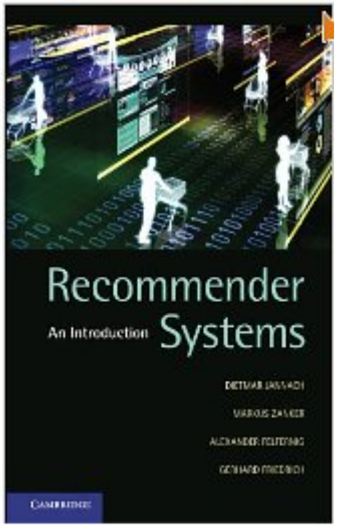


- **One Recommender Systems research question**
  - What should be in that list?

Customers Who Bought This Item Also Bought

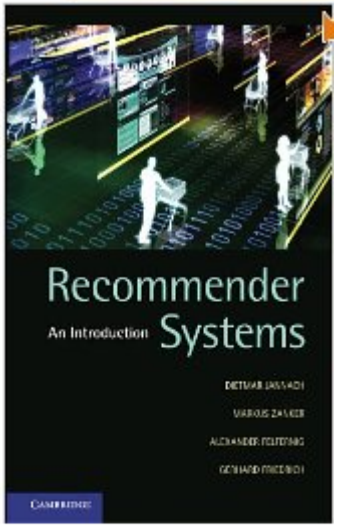
Recommender Systems Handbook	Algorithms of the Intelligent Web	Programming Collective Intelligence: ...	Machine Learning: A Probabilistic ...	Data Mining: Practical Machine Learning ...
Francesco Ricci	Haralambos Marmanis	> Toby Segaran	> Kevin P. Murphy	> Ian H. Witten
	(14)	(91)	(15)	(29)
Hardcover	Paperback	Paperback	Hardcover	Paperback
$167.73	$26.76	$25.20	$81.00	$42.61

Recommender Systems in e-Commerce



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

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?

These have been in stock for quite a while now …

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

However, these evaluation methods only work for "operative" systems, where we already have active users!
What if the domain is brand-new ? (will see later)

Purpose and success criteria (1)

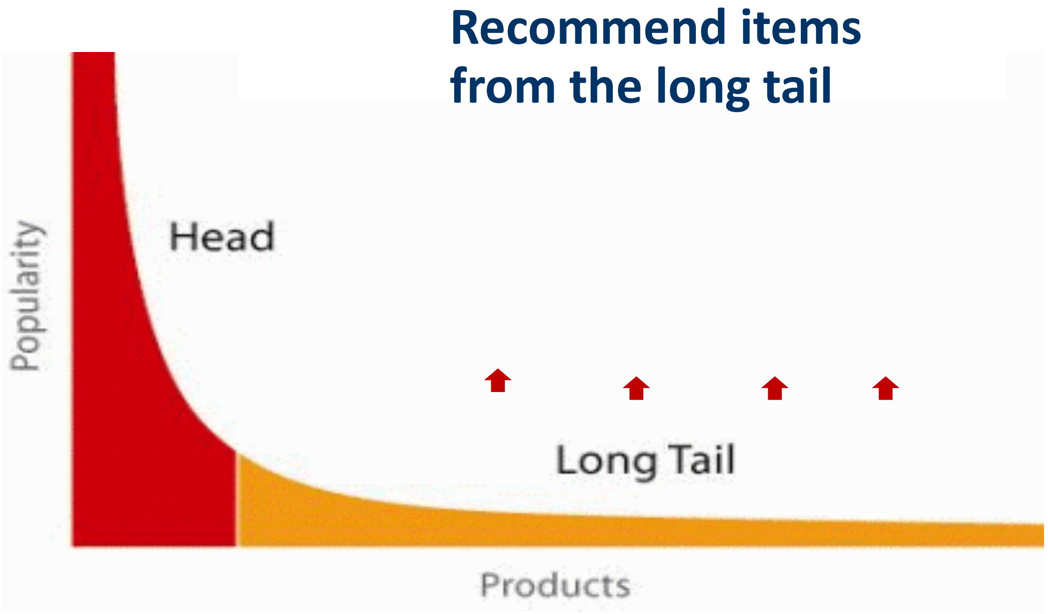
Different perspectives/aspects

- Depends on domain and purpose
- No holistic evaluation scenario exists

- **Retrieval perspective**
  - Reduce search costs
  - Provide "correct" proposals
  - Assumption: Users know in advance what they want

- **Recommendation perspective**
  - **Serendipity** – identify items from the Long Tail – not obvious recommendations!
  - Users did not know about their existence

When does a RS do its job well?



- "Recommend widely unknown items that users might actually like!"
- 20% of items accumulate 74% of all positive ratings

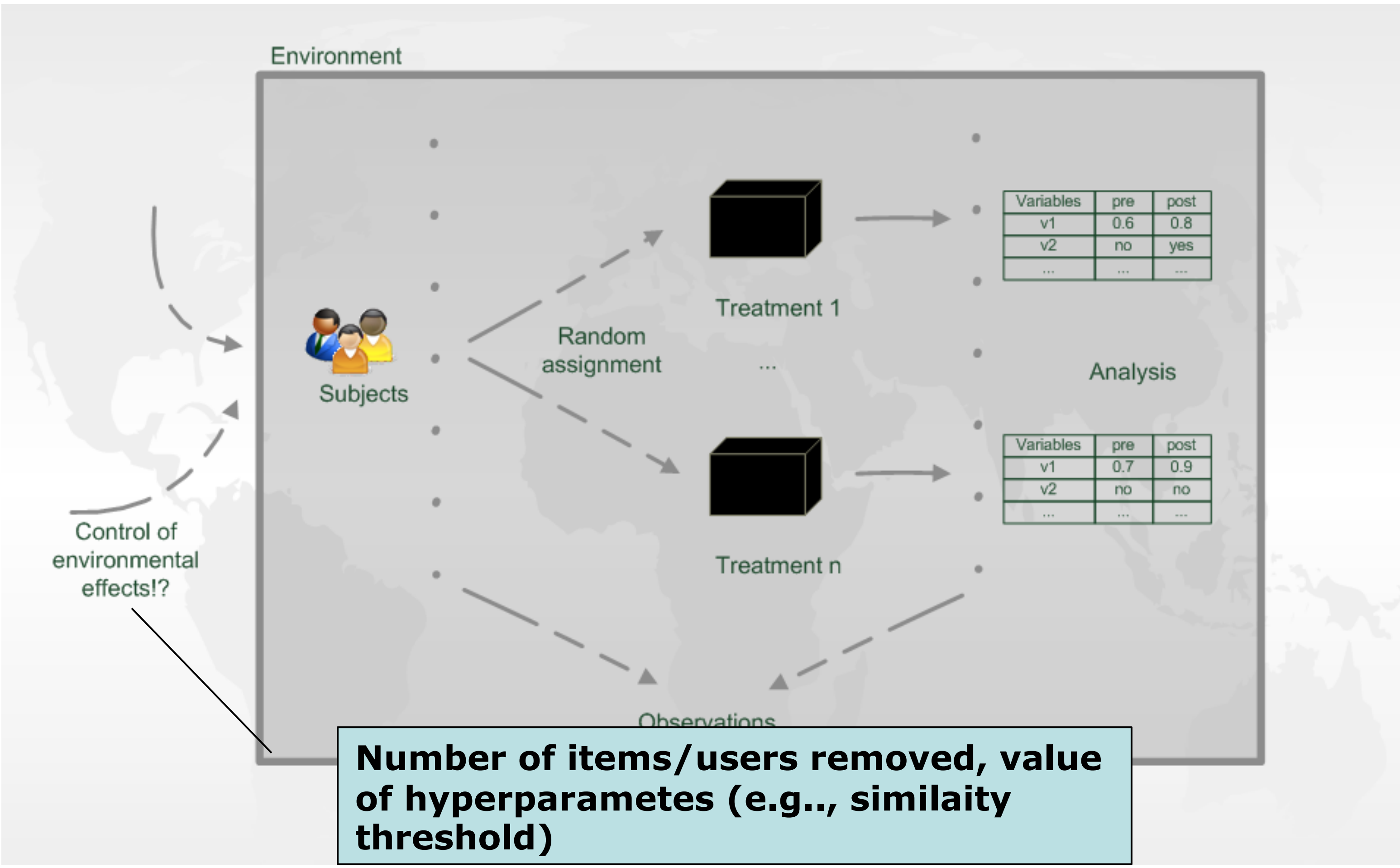
Purpose and success criteria (2)

- **Prediction perspective**
  - Predict to what degree users like an item
  - Most popular evaluation scenario in research
- **Interaction perspective**
  - Give users a "good feeling"
  - Educate users about the product domain
  - Convince/persuade users - explain
- **Finally, conversion perspective**
  - Commercial situations
  - Increase "hit", "clickthrough", "***lookers to bookers***" rates
  - Optimize sales margins and profit

How do we, as researchers, know?

- **Test with real users**
  - A/B tests
  - Example measures: sales increase, click through rates – as we said, real users are often not available for new types of recommenders (e.g., recommending places to visit during a trip)
- **Laboratory studies**
  - Controlled experiments: recruit a number of possible users
  - Example measures: satisfaction with the system (questionnaires)
- **Offline experiments**
  - Based on historical data (predict the “known” future: remove items from a user’s purchase list, learn a recommendation model based on these “purged” data, and then test if system would recommend removed items)
  - Example measures: prediction accuracy, coverage

Experiment designs



Evaluation as in information retrieval (IR)

- **Recommendation is viewed as information retrieval task:**
  - Retrieve (recommend) all items which are predicted to be "good" or "relevant".
- **Common protocol :**
  - Hide some items with known ground truth (e.g. rankings are known to evaluators, but not known to recommender system)
  - Often historic rating: System learns a model based on e.g. ratings from date d0 to date d1, and predict ratings after d1 (which are actually known)
- **Evaluation based on confusion matrix**

		Reality	
		Actually Good	Actually Bad
Prediction	Rated Good	True Positive (tp)	False Positive (fp)
	Rated Bad	False Negative (fn)	True Negative (tn)

Offline experimentation needs large datasets

- **Netflix prize dataset**
  - Web-based movie rental
  - Prize of $1,000,000 for accuracy improvement (RMSE) of 10% compared to own Cinematch system.

- **Movilens** (Harper and Konstan, 2016)
- **Million song dataset** (McFee et al., 2012)
- **Wiki-MED** (Di Tommaso et al. 2018 a, 2018b) – the largest multi-domain-

Metrics: Precision and Recall (known staff)

- **Precision: a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved**
  - E.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

- **Recall: a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items**
  - E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

Dilemma of IR measures in RS

- **IR-like measures are frequently applied, however**:
  - If we have non-unary ratings (e.g., like/dislike) precision and recall are not adequate
  - Different ways of measuring precision possible
- **Results from offline experimentation may have limited predictive power for online user behavior**.

Better accuracy metrics (1)

- **Metrics measure error rate**
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i|$$

  - Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

Better accuracy metrics (2)

For a user:

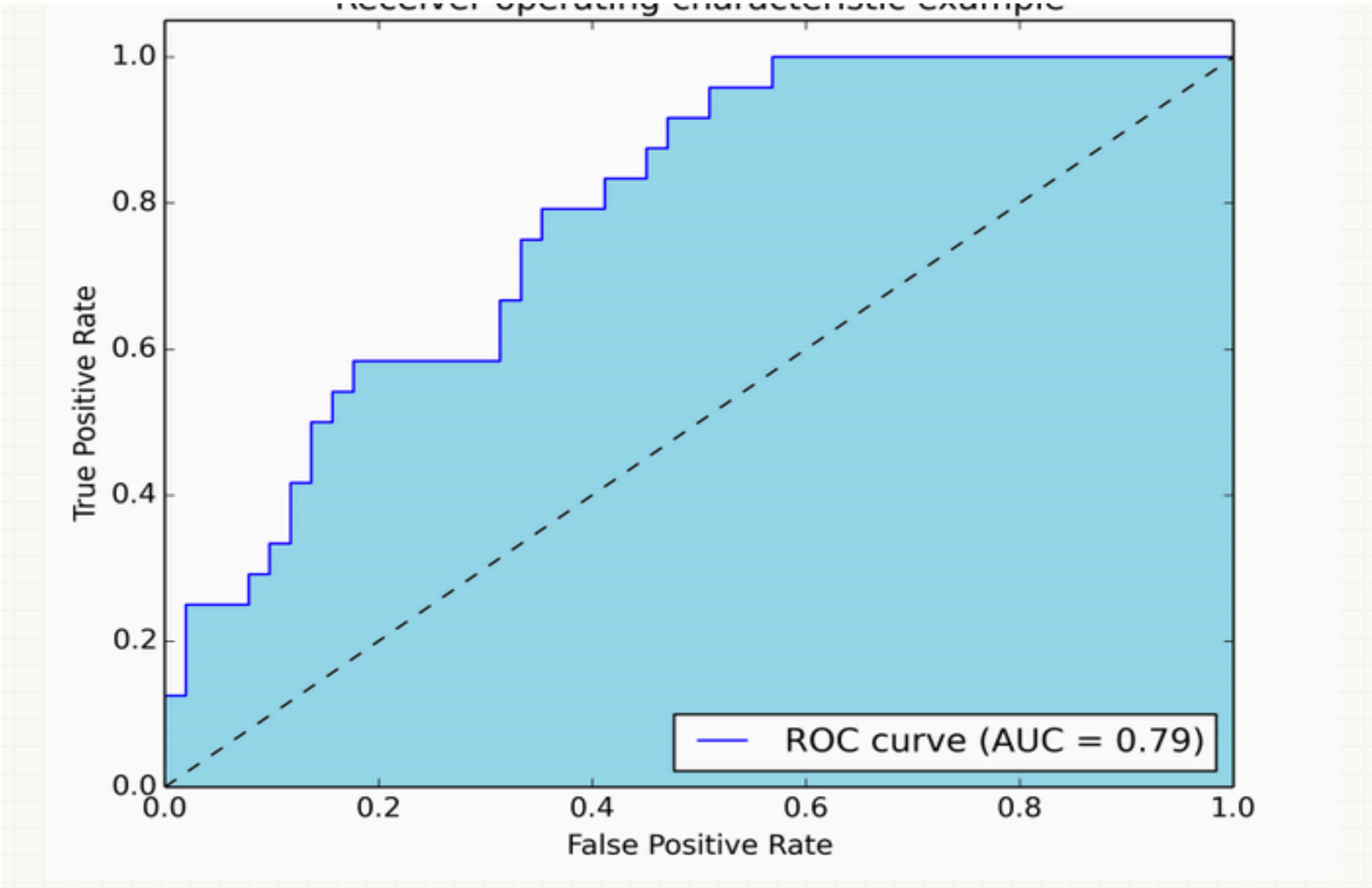
Actually good
Item 237
Item 899

hit

Recommended (predicted as good)
Item 345
Item 237
Item 187

- **Rank Score extends recall and precision to take the positions of correct items in a ranked list into account**
  - Particularly important in recommender systems, as lower ranked items may be overlooked by users
  - Learning-to-rank: define a model, a measure, and an optimization problem to optimize the model for such measures (e.g., AUC, area under the curve)

AUC (Area Under Curve – often Area Under ROC)



Alternative measures

- **Alternative and complementary measures:**
  - Diversity and Novelty (serendipity), Coverage, Familiarity, Serendipity, Popularity, Concentration effects (Long tail)
  - All these variants have the objective of prizing the most salient recommendations according to other criteria than a user's interest – of course the user must adopt the item!

Non-experimental research

- **Non-experimental / observational research**
  - Surveys / Questionnaires (also trough crowdsourced evaluation platforms, e.g. Crowdflower.com. , MechanicalTurk)
  - Longitudinal research
    - Observations over long period of time
    - E.g. customer life-time value, returning customers
  - Case studies
  - Focus group
    - Interviews
    - Think-aloud protocols