Intro to recommender systems

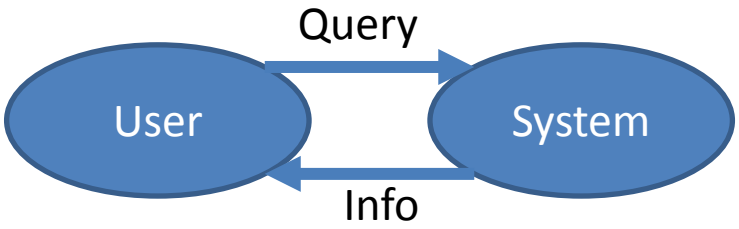
Nisheeth

Recommender systems

- Algorithms
- Applications
- Evaluation
- Usability/interface issues
- Research Directions
- Reading material
  - Main text: Recommender Systems Handbook, pdf available on HCC course page
    - I will mostly cover material from chapters 1,3,4,5,8,11
  - Bobadilla et al (2013) Recommender Systems survey
  - Dietmar et al (2013) Recommender Systems: an Introduction
    - Lots of materials in my slides from them

Taxonomy of information retrieval

- Up front cost: what it costs the user to accept the information
- Error cost: Cost to the system of retrieving bad information
- Heterogeneity: Are different users likely to want different things
- Frequency: How frequently do users use the service
- Scale: What is the size of the corpus being queried per user



Taxonomy of factual web search

	Low	Medium	High
Upfront cost	X		
Error cost		X	
Heterogeneity	X		
Frequency			X
Scale			X

Product recommendations

amazon.com

Recommended for You

Amazon.com has new recommendations for you based on items you purchased or told us you own.

Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop

Google Apps Administrator Guide: A Private-Label Web Workspace

Googlepedia: The Ultimate Google Resource (3rd Edition)

Taxonomy of product recommendations

	Low	Medium	High
Upfront cost		X	
Error cost			X
Heterogeneity	X		
Frequency	X		
Scale			X

Consumption recommendations

Taxonomy of consumption recommendations

	Low	Medium	High
Upfront cost			X
Error cost		X	
Heterogeneity			X
Frequency		X	
Scale		X	

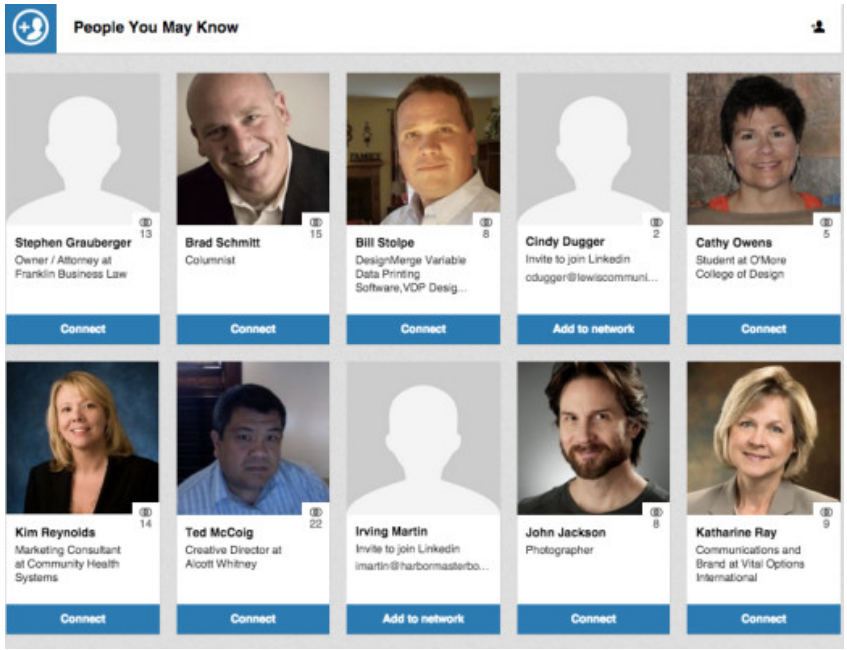
Social recommendations

Create an Ad

Facebook Suggestion: Matthew Moore

Matthew Moore and Taj Moore are mutual friends.

Add as Friend [?]



I have never seen a tinder profile quite like yours. You went all out haha

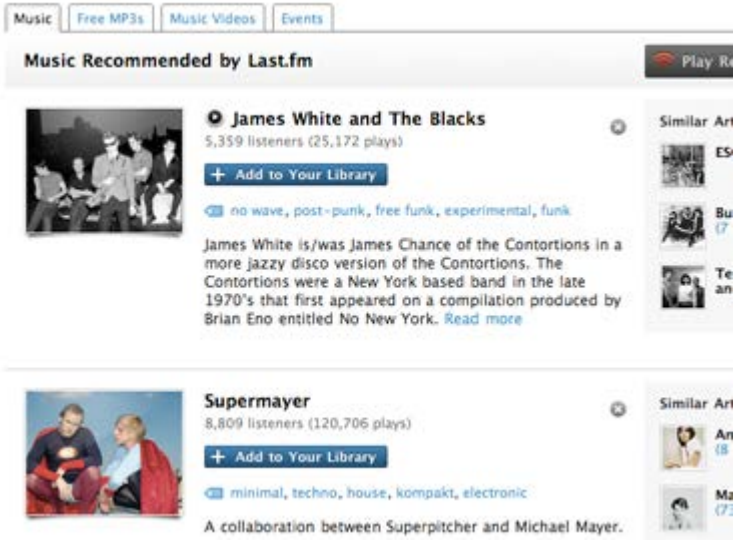
For a sec I thought your "match of the day" was something that Tinder did, but I see what you did there. Good job!

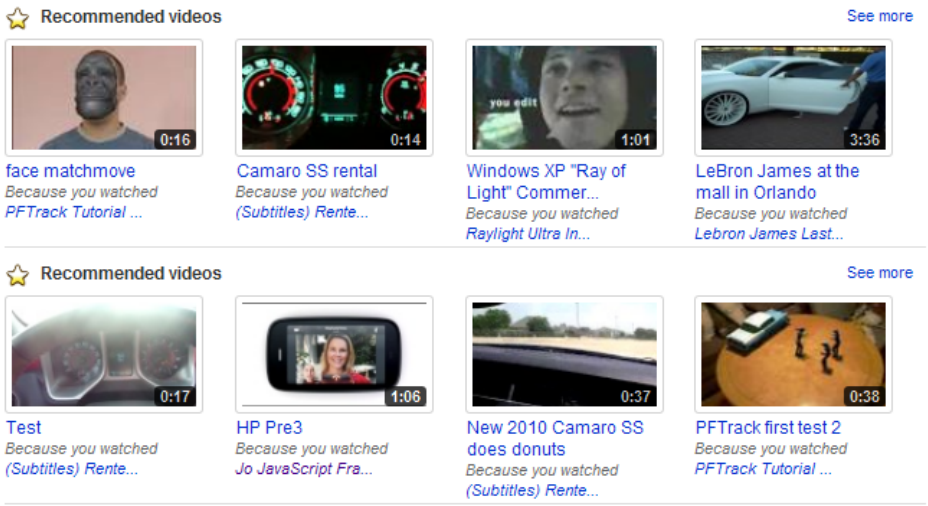
Hello Match of the Day! How did you get that title? :-)

Taxonomy of social recommendations

	Low	Medium	High
Upfront cost	X		
Error cost		X	
Heterogeneity		X	
Frequency		X	
Scale	X		

Experience recommendations





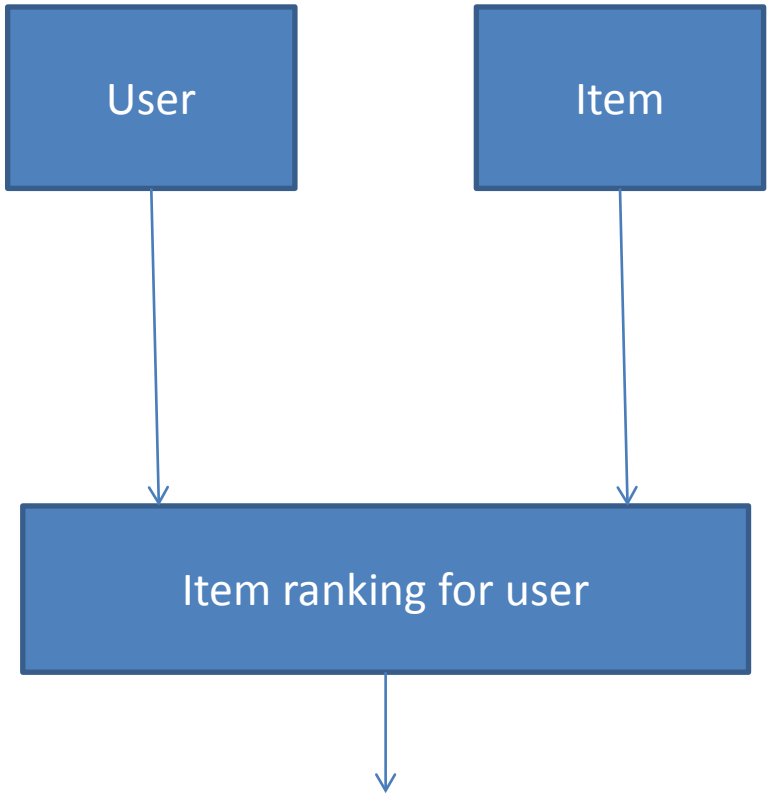
Taxonomy of experience recommendations

	Low	Medium	High
Upfront cost	X		
Error cost			X
Heterogeneity			X
Frequency		X	
Scale	X		

Typical data sources

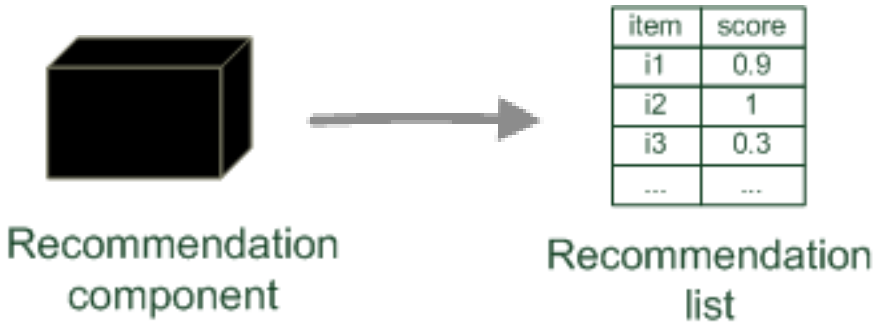
- Preference information
  - Implicit, e.g. dwell time
  - Explicit, e.g. Ratings
- Content information
  - Implicit, e.g., user trends, item trends
  - Explicit, e.g. demographics, item features
- Social information
  - Implicit, e.g. Friend graph, retailer brand
  - Explicit, e.g., Verified profiles, review counts
- Context information
  - Implicit, e.g. Location, time, venue

Recommender system workflow

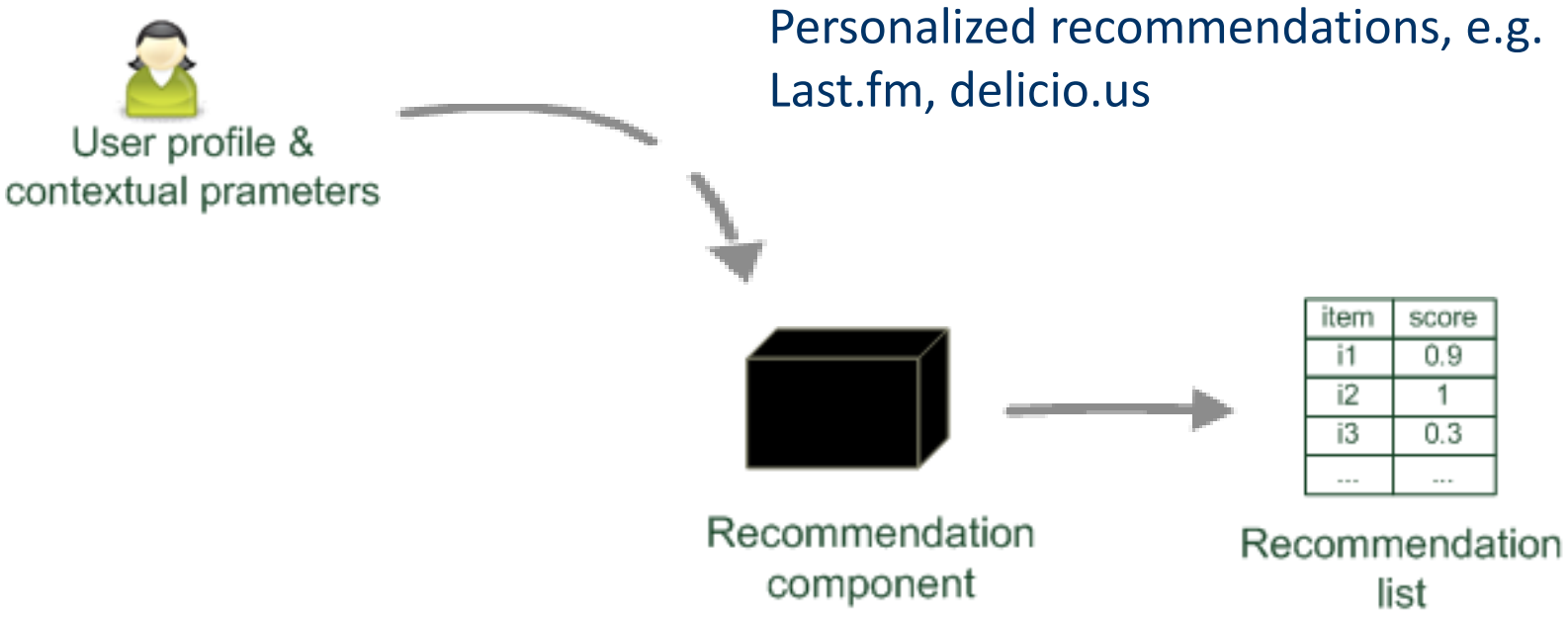


Types of recommender systems

Important algorithmic differences in recommender systems

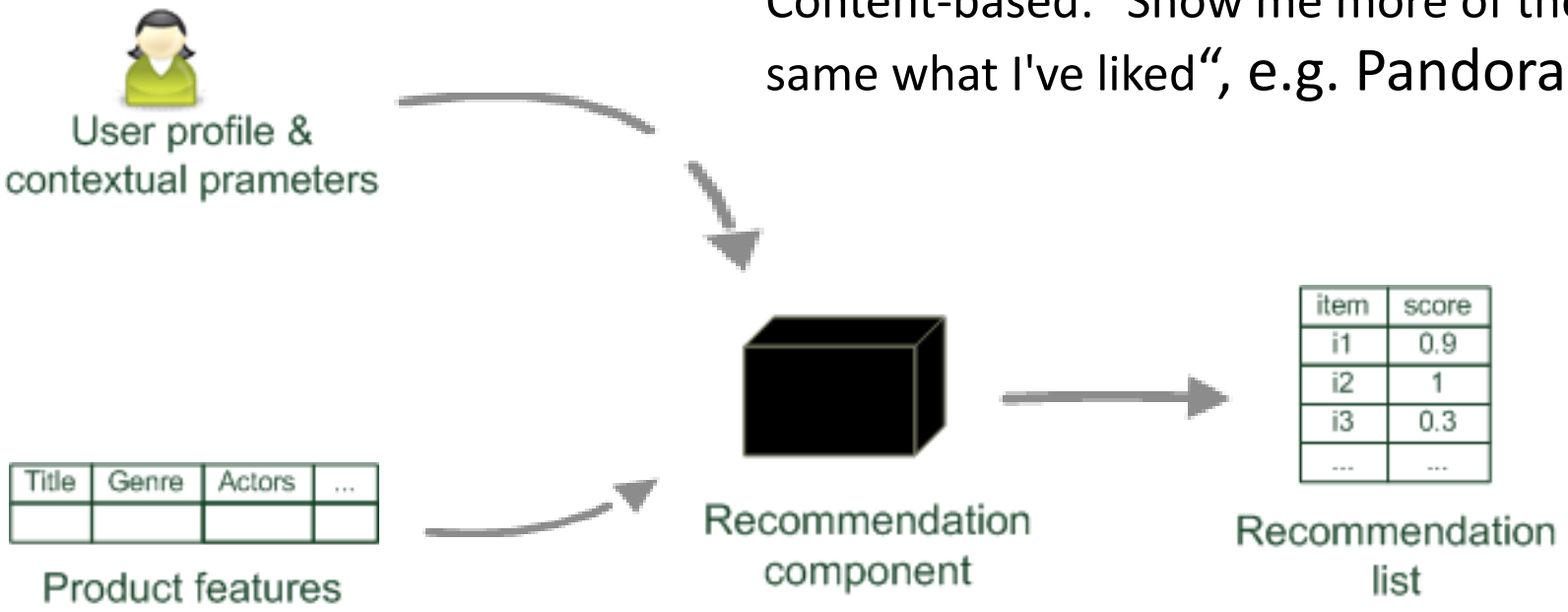


Types of recommender systems

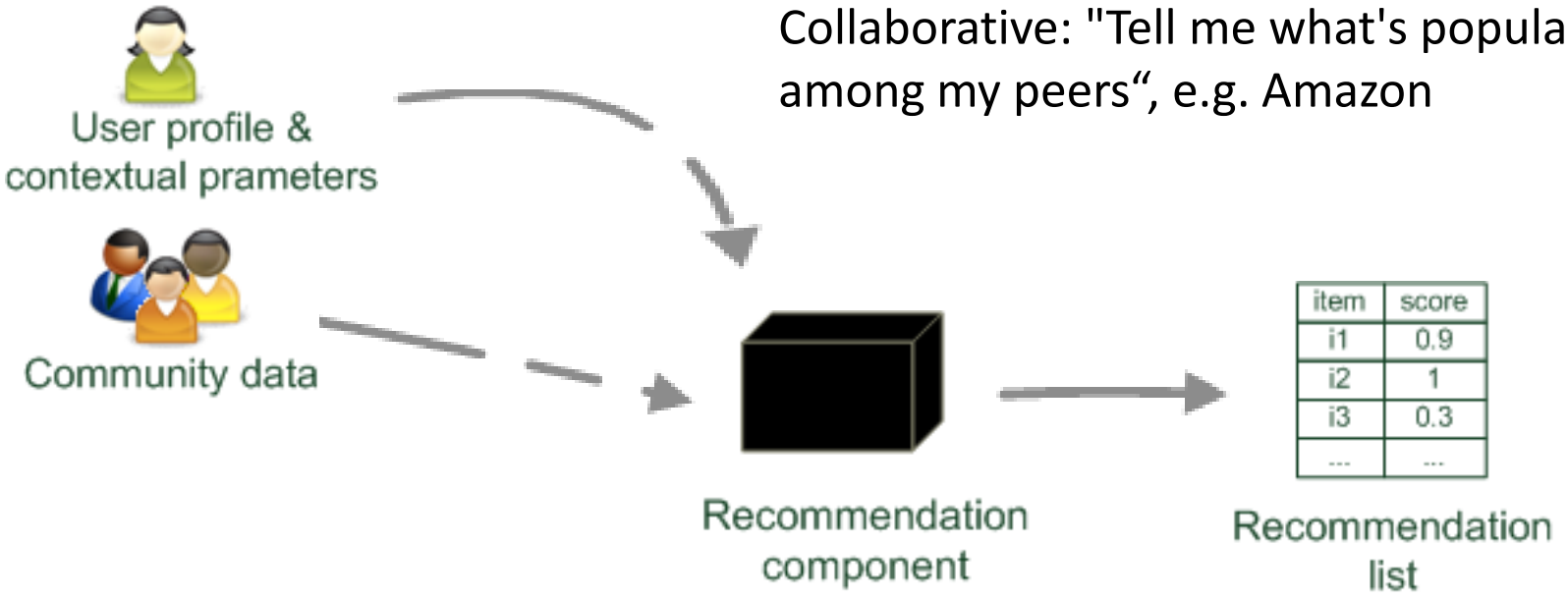


Types of recommender systems

Content-based: "Show me more of the same what I've liked", e.g. Pandora

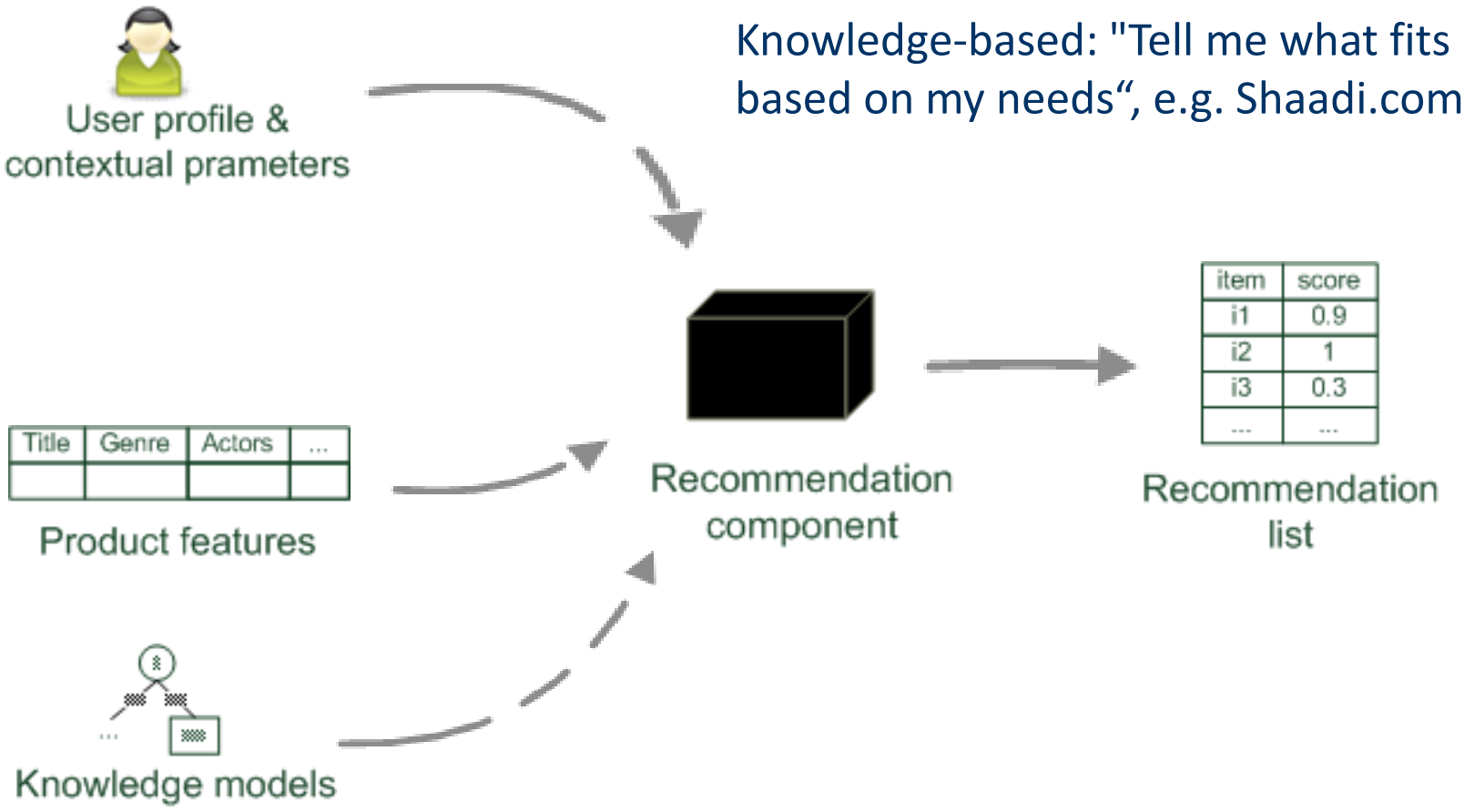


Types of recommender systems



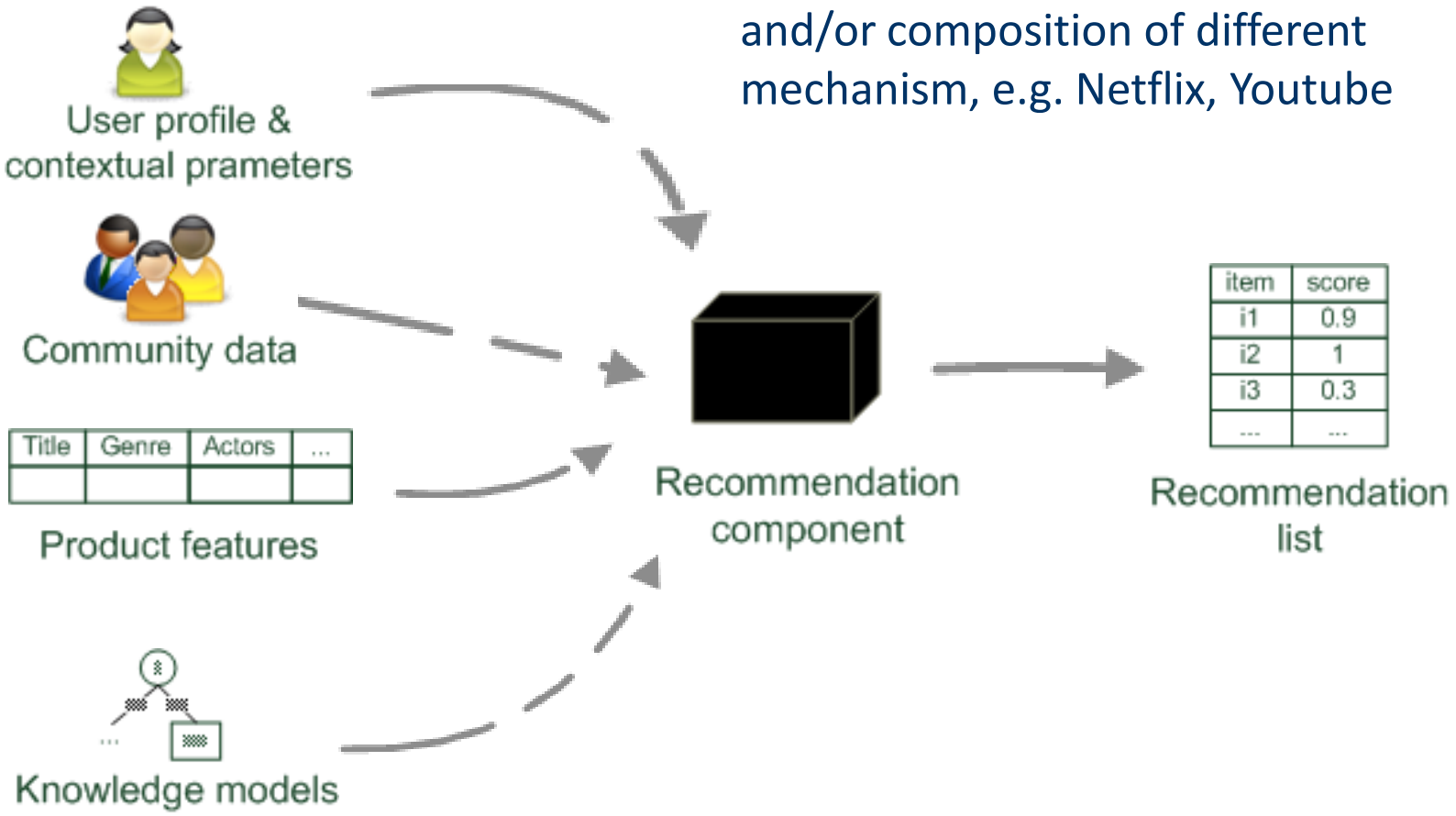
Collaborative: "Tell me what's popular among my peers“, e.g. Amazon

Types of recommender systems



Types of recommender systems

Hybrid: combinations of various inputs and/or composition of different mechanism, e.g. Netflix, Youtube

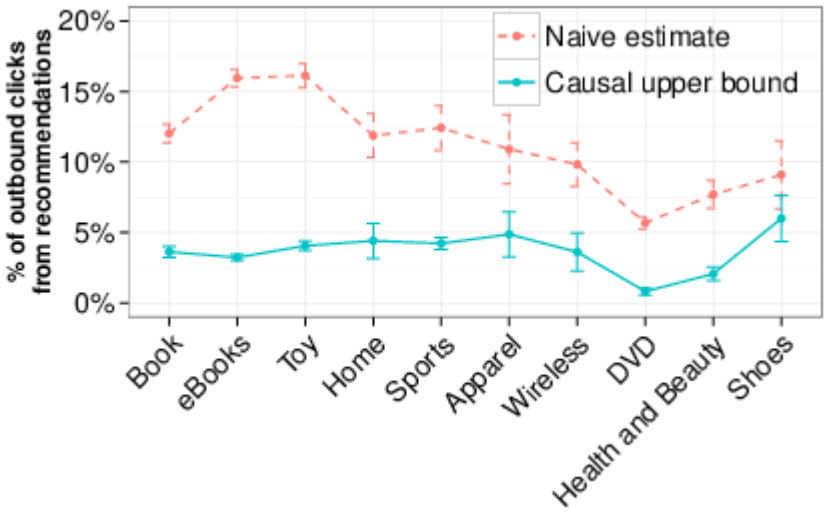


Value proposition

- To user
  - Reduce information search time
  - Discover new things
- To server
  - Sell more
  - Know customers better

Myth vs. reality

- Myth: 35% of Amazon product landings from recommender system
- Reality: <10% really caused by recommender system



(Sharma, Hofman & Watts, 2015)

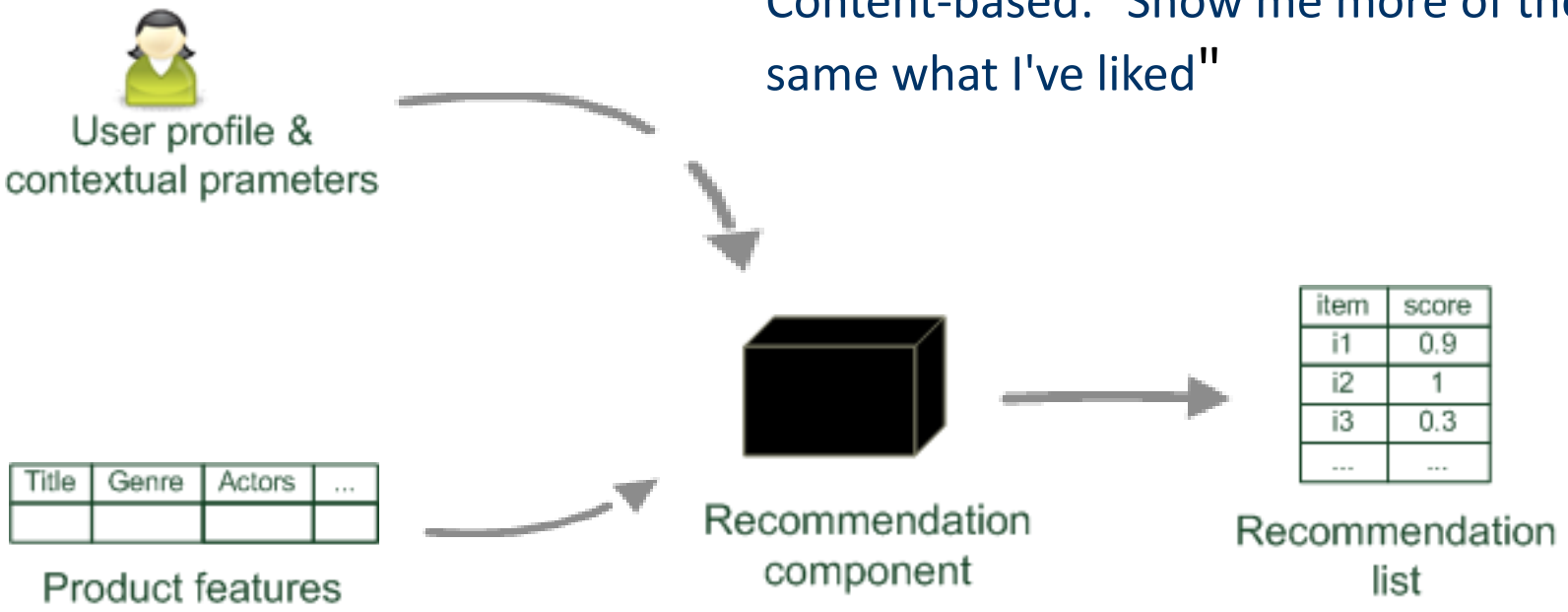
Content-based recommendation

Content-based recommendation

- 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

Types of recommender systems

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



What is the "content"?

- The genre is actually not part of the content of a book
- Most CB-recommendation methods originate from Information Retrieval (IR):
  - The item descriptions are usually automatically extracted (important words)
  - Goal is to find and rank interesting text documents (news articles, web pages)

- Here:
  - Classical IR-based methods based on keywords
  - No expert recommendation knowledge involved
  - User profile (preferences) are rather learned than explicitly elicited

Content representation and item similarities

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 the user profile based on the keyword overlap (e.g. using the Dice coefficient)
  - $\text{sim}(b_i, b_j) = \frac{2 * |keywords(b_i) \cap keywords(b_j)|}{|keywords(b_i)| + |keywords(b_j)|}$

Term-Frequency - Inverse Document Frequency (TF-IDF)

- Simple keyword representation has its problems
  - In particular when automatically extracted because
    - Not every word has similar importance
    - Longer documents have a higher chance to have an overlap with the user profile
- Standard measure: TF-IDF
  - Encodes text documents as weighted term vector
  - TF: Measures, how often a term appears (density in a document)
    - Assuming that important terms appear more often
    - Normalization has to be done in order to take document length into account
  - IDF: Aims to reduce the weight of terms that appear in all documents

Example TF-IDF representation

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Figure taken from http://informationretrieval.org

More on the vector space model

- Vectors are usually long and sparse
- Improvements
  - Remove stop words ("a", "the", ..)
  - Use stemming
  - Size cut-offs (only use top n most representative words, e.g. around 100)
  - Use additional knowledge, use more elaborate methods for feature selection
  - Detection of phrases as terms (such as United Nations)
- Limitations
  - Semantic meaning remains unknown
  - Example: usage of a word in a negative context
    - "there is **nothing** on the menu that a vegetarian would like.."
- Usual similarity metric to compare vectors: Cosine similarity (angle)

Recommending items

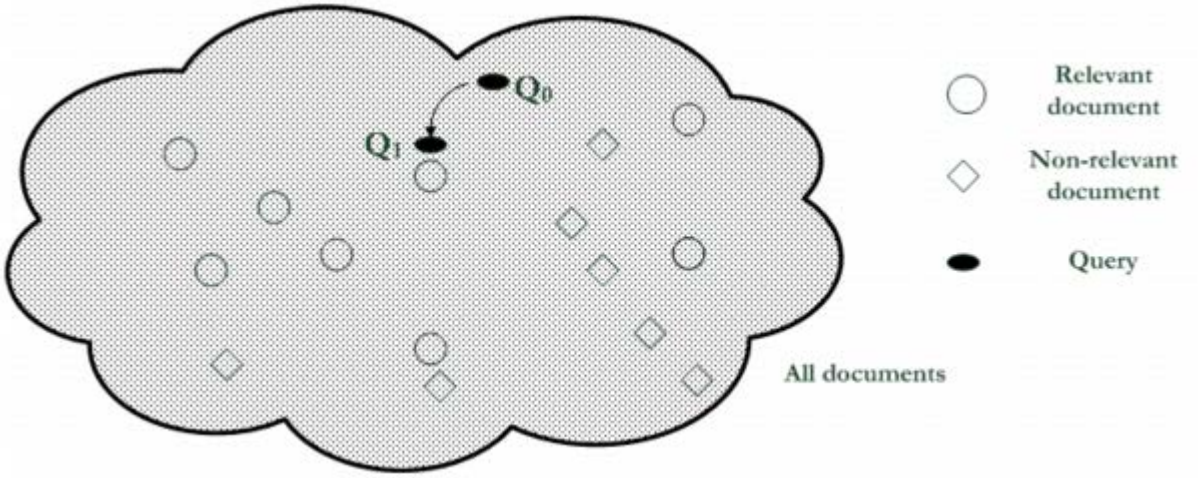
- Simple method: nearest neighbors
  - Given a set of documents D already rated by the user (like/dislike)
    - Find the n nearest neighbors of a not-yet-seen item *i* in D
    - Take these ratings to predict a rating/vote for *i*
    - (Variations: neighborhood size, lower/upper similarity thresholds)
- Query-based retrieval: Rocchio's method
  - The SMART System: Users are allowed to rate (relevant/irrelevant) retrieved documents (feedback)
  - The system then learns a prototype of relevant/irrelevant documents
  - Queries are then automatically extended with additional terms/weight of relevant documents

Rocchio algorithm

- Document collections D^+ and D^-
- α, β, γ used to fine-tune the feedback

$$Q_{i+1} = \alpha * Q_i + \beta(\frac{1}{|D^+|}\sum_{d^+ \in D^+} d^+) - \gamma(\frac{1}{|D^-|}\sum_{d^- \in D^-} d^-)$$

- often only positive feedback is used



Probabilistic methods

- Recommendation as classical text classification problem
  - Long history of using probabilistic methods
- Simple approach:
  - 2 classes: like/dislike
  - Simple Boolean document representation
  - Calculate probability that document is liked/disliked based on Bayes theorem

Remember:
$P(Label=1|X) = k*P(X|Label=1) * P(Label=1)$

Simple NB example

Doc-ID	recommender	intelligent	learning	school	**Label**
1	1	1	1	0	**1**
2	0	0	1	1	**0**
3	1	1	0	0	**1**
4	1	0	1	1	**1**
5	0	0	0	1	**0**
6	1	1	0	0	**?**

$$
\begin{aligned}
\mathrm{P}(\mathrm{X}|\mathrm{Label}{=}1) &= \mathrm{P}(\mathrm{recommender}{=}1|\mathrm{Label}{=}1) \times \\
&\quad \mathrm{P}(\mathrm{intelligent}{=}1|\mathrm{Label}{=}1) \times \\
&\quad \mathrm{P}(\mathrm{learning}{=}0|\mathrm{Label}{=}1) \times \mathrm{P}(\mathrm{school}{=}0|\mathrm{Label}{=}1) \\
&= 3/3 \times 2/3 \times 1/3 \times 2/3 \\
&\approx 0.149
\end{aligned}
$$

Improvements

- Side note: Conditional independence of events does in fact not hold
  - “New”/ “York“ and “Hong” / “Kong"
  - Still, good accuracy can be achieved
- Boolean representation simplistic
  - Keyword counts lost
- More elaborate probabilistic methods
  - E.g. estimate probability of term **v** occurring in a document of class **C** by relative frequency of **v** in all documents of the class
- Other linear classification algorithms (machine learning) can be used
  - Support Vector Machines, ..

Best fit for

	Low	Medium	High
Upfront cost	X		
Error cost	X		
Heterogeneity	X		
Frequency			X
Scale			X

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
  - Content may not be automatically extractable (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