AALBORG UNIVERSITY

Recommender Systems: Introduction & Content-based Filtering

Bo Thiesson
thiesson@cs.aau.dk
http://people.cs.aau.dk/~thiesson

Based on "Content-Based Recommendation Systems" by Michael J. Pazzani & Daniel Billsus (2007), and on Chap. 3 from the book "Recommender Systems: An Introduction" by Dietmar Jannach, Markus Zanker, Alexander Felfernig & Gerhard Friedrich (2011) with many slides copied or adapted from the teaching material supporting the latter. A few slides also provided by Peter Dolog, AAU.

cc BY SA
Bo Thiesson

Outline

- Motivation
- Basic paradigms
  - Content-based
  - Collaborative
  - Hybrid
- Content-based
  - General principle
  - Similarity measures
  - Rating feedback
  - Memory- and model-based approaches

cc BY SA
Bo Thiesson

Motivation

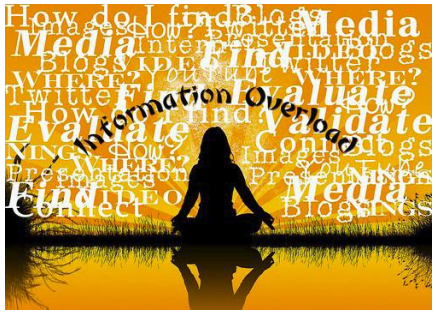
Statistics
Machine Learning

Biology
Medicine

Psychology
Readings

BY SA
Bo Thiesson

Problems with General Search



- Too many sources
- Different people have different tastes
- Even with a general social agreements about authoritative sources of information, some people might disagree or have different specific needs at different times

⇒ **Information overload**

- Low precision: retrieved information is not what you need
- Low recall: not enough of the correctly relevant information is returned

cc BY SA

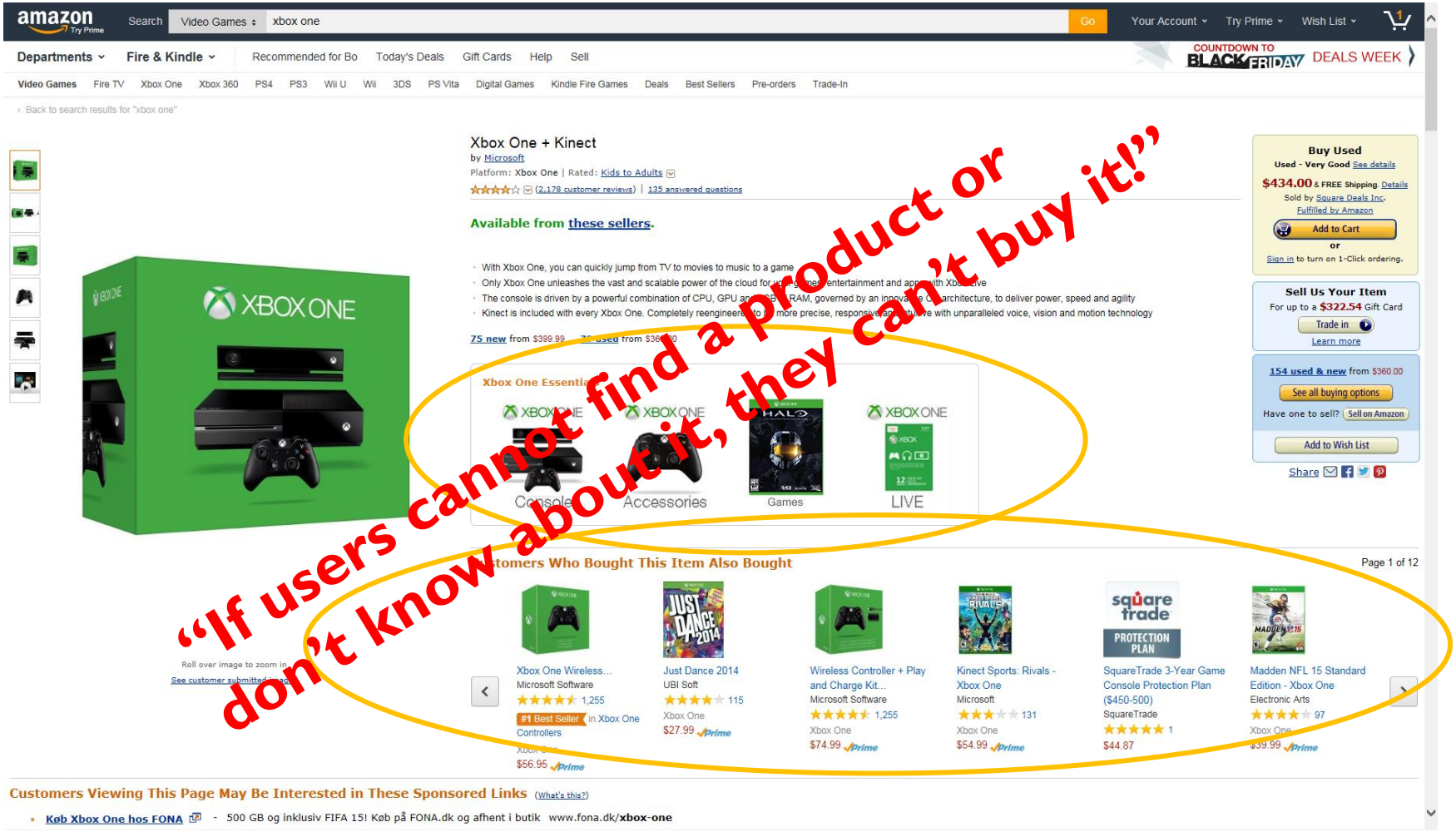
Context

- **Information Retrieval (IR)**: static content, dynamic query
  → modeling content (organized with index)
- **Information Filtering (IF)**: dynamic content, static query
  → modeling query (organized as filters)

Recommendation is between IR and IF since the content varies slowly and so does queries. Methods of both IR and IF are then used to reduce computation at query time.

BY SA
Bo Thiesson

Recommendations



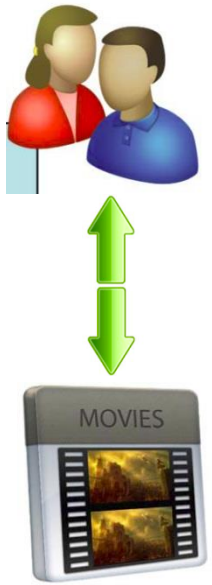
"If users cannot find a product or don't know about it, they can't buy it!"

BY SA

Problem Domain

Recommendation systems (RS) help to match users with items

- Ease information overload
  - How many books on Amazon?
  - How many tracks on iTunes?
  - How many videos on YouTube?



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 2007[1])*

(1) Xiao and Benbasat, E-commerce product recommendation agents: Use, characteristics, and impact, MIS Quarterly 31 (2007), no. 1, 137–209

BY SA
Bo Thiesson

An often-cited problem characterization

(Adomavicius & Tuzhilin, TKDE, 2005)

Given

- The profile of the "active" user and possibly some situational context
- Items, possibly with description of item characteristics

Compute

- A relevance (ranking) score for each recommendable item

The profile …

- … can include past user ratings (explicit or implicit), demographics and interest scores for item features

The problem …

- … is to learn a function that predicts the relevance score for a given (typically unseen) item

BY SA
Bo Thiesson

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
- …

cc BY SA
Bo Thiesson

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)
- News recommendation at Forbes.com (plus 37% CTR)

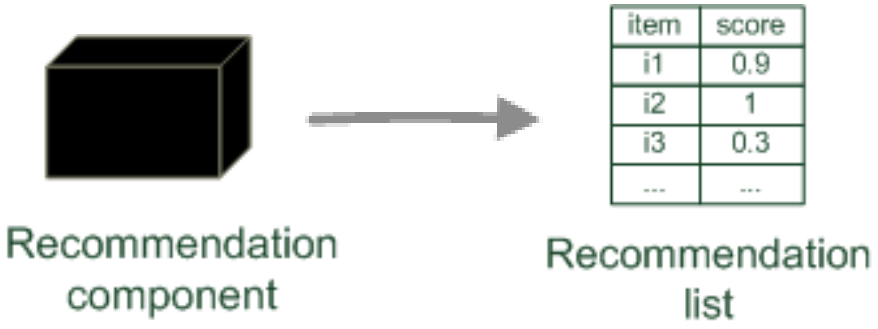
There must be some value in it

- See recommendation of groups, jobs or people on LinkedIn
- Friend recommendation and ad personalization on Facebook
- Song recommendation at last.fm
- Video recommendation at YouTube

cc BY SA
Bo Thiesson

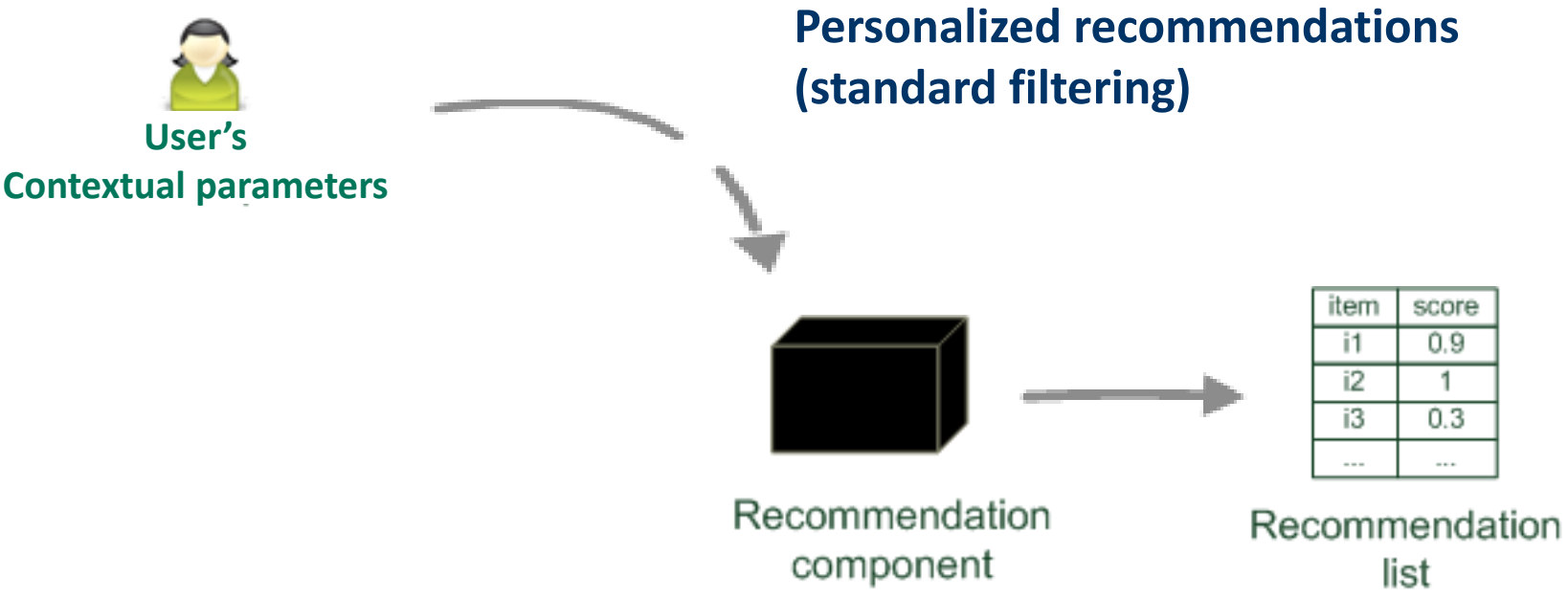
Paradigms of recommender systems

Recommender systems reduce information overload by estimating relevance



BY SA
Bo Thiesson

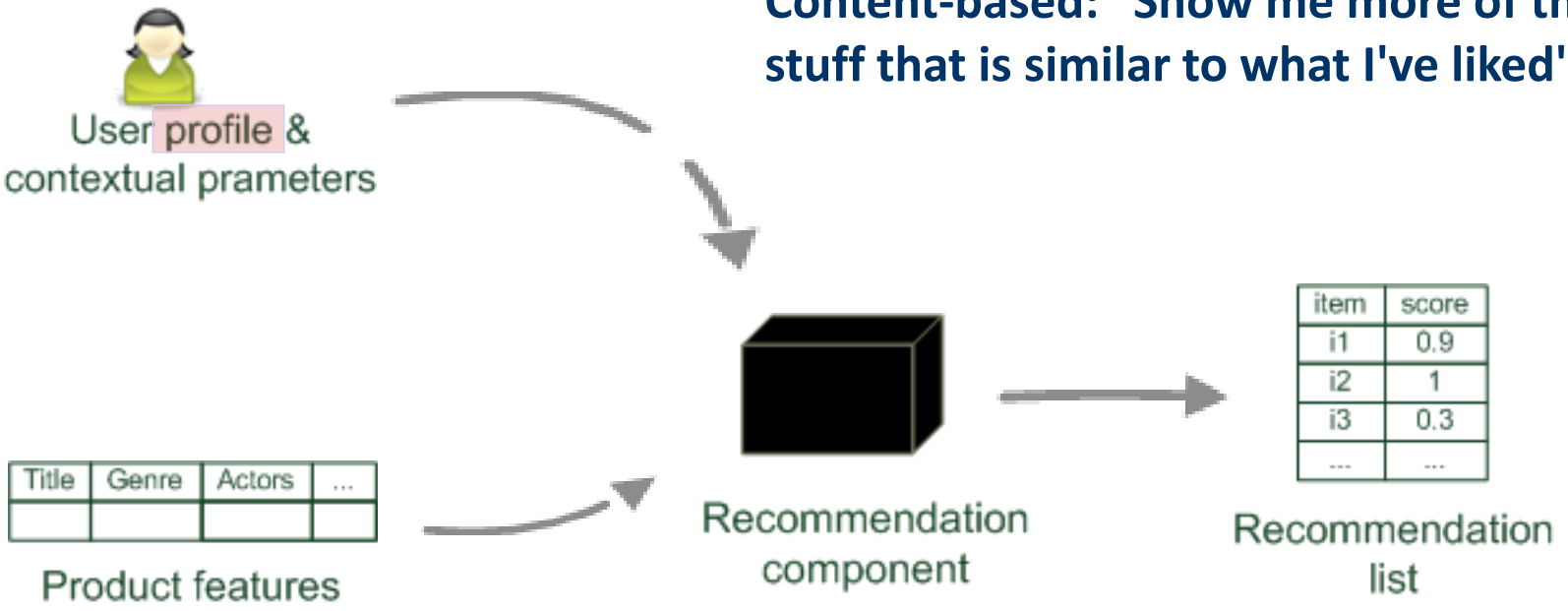
Paradigms of recommender systems



cc BY SA
Bo Thiesson

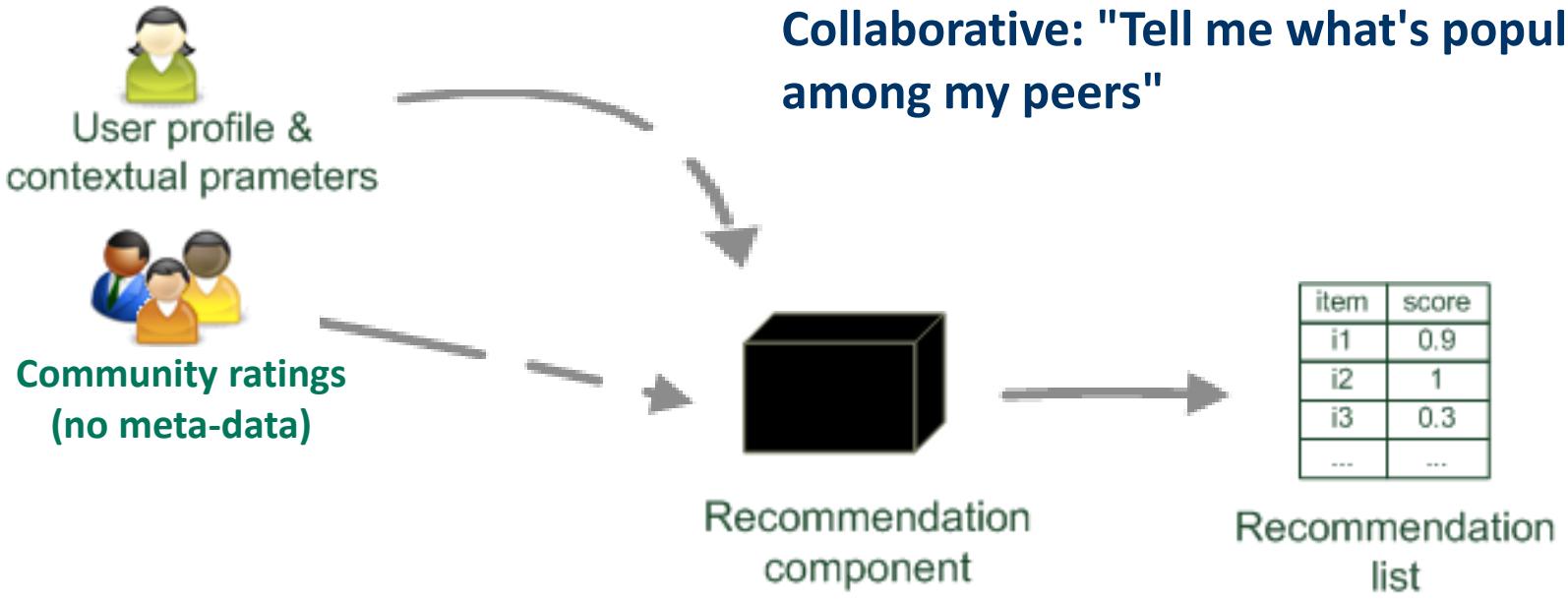
Paradigms of recommender systems

Content-based: "Show me more of the stuff that is similar to what I've liked"



cc BY SA
Bo Thiesson

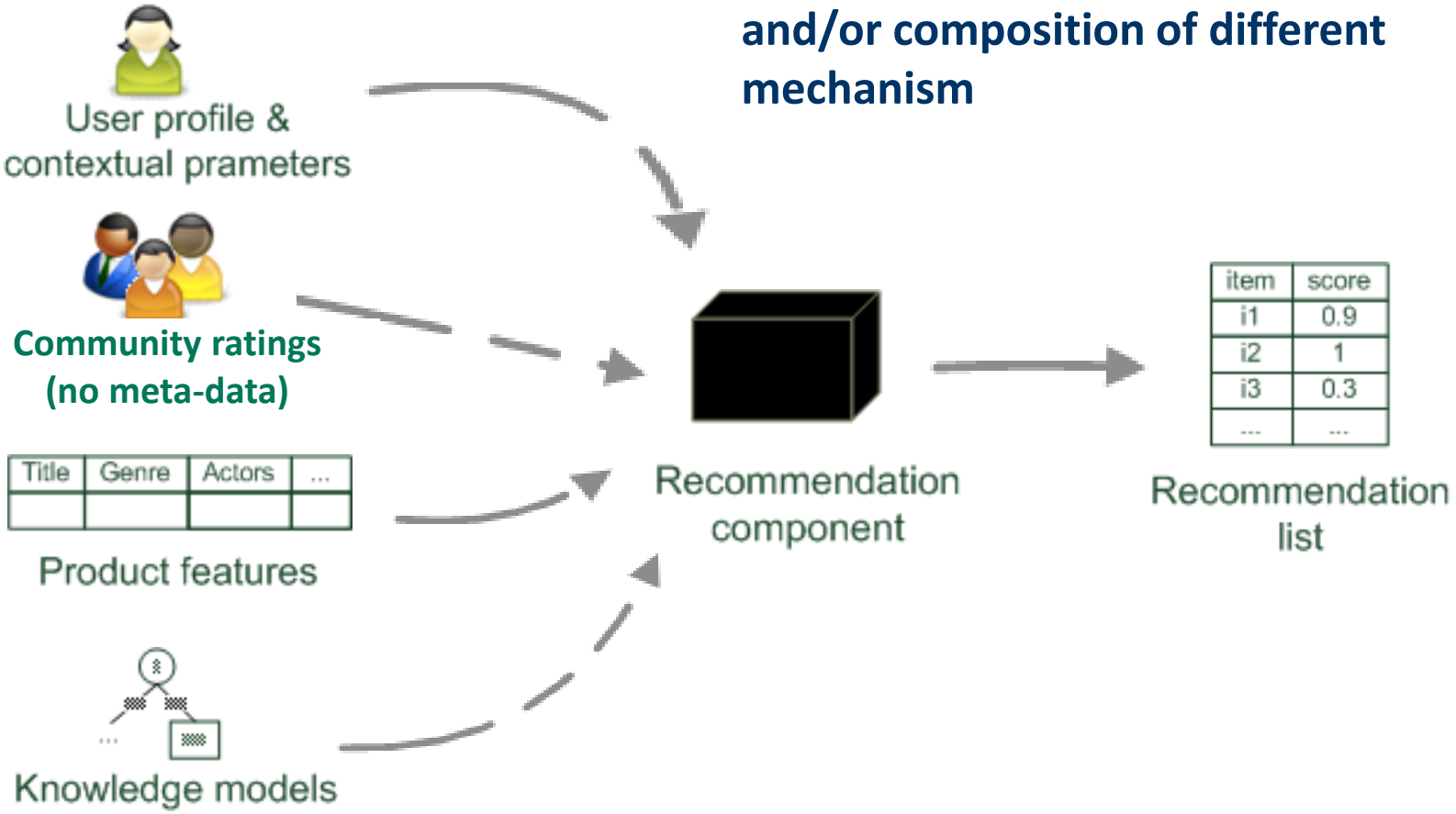
Paradigms of recommender systems



cc BY SA
Bo Thiesson

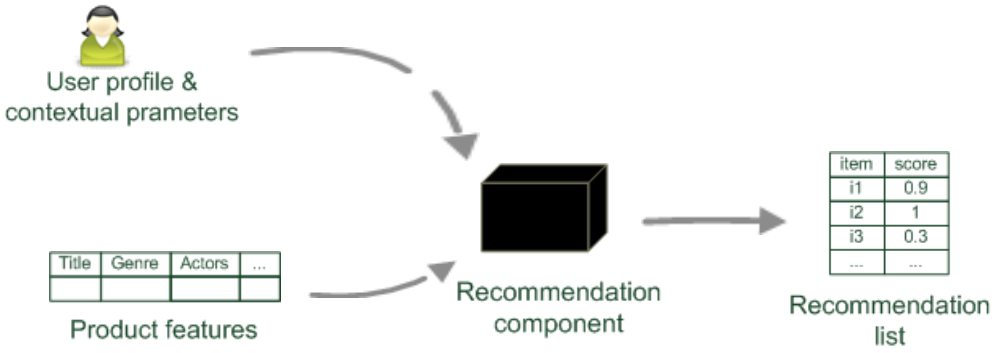
Paradigms of recommender systems

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



Content-based Filtering

"show me more of the stuff that is similar to what I've liked"



BY SA
Bo Thiesson

Content-based Filtering

What do we need:

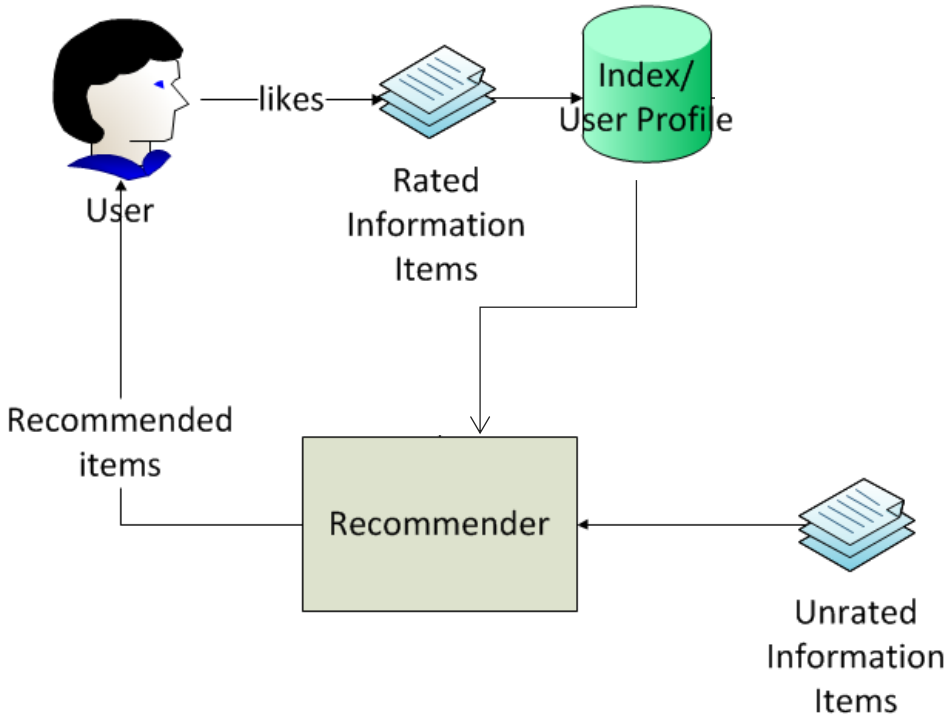
- some information about the available items such as the genre ("content")
- some sort of user profile describing what the user likes (the preferences)
  - Implicit: look at what the user liked / not liked (purchased, viewed, tagged,…)
  - Explicit: ask the user

The task:

- learn user preferences
- locate/recommend items that are "similar" to the user preferences
- recommend fantasy novels to people who liked fantasy novels in the past (but not the whole story)

BY SA
Bo Thiesson

It is an adaptive process…



BY SA
Bo Thiesson

What is the "content"?

Most CB-recommendation techniques were applied to recommending text documents.

- Like web pages or newsgroup messages for example.
- But can also be used for products, restaurants, and other specialized services
- Hybrid CB and CF even better!

Content of items can typically be represented in a semi-structured way

- Structured:
  - Each item is described by the same set of attributes
  - Some attributes represent "standard" boolean, categorial, nominal, or continous features
- Unstructured:
  - Some attributes holds free-text descriptions.

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

BY SA
Bo Thiesson

Content-based filtering: Basics

Represent items and users in the same way

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

Items

Title	Genre	Author	Type	Price	Keywords
…	Fiction	Brunonia, Barry, Ken Follett	Paperback	25.65	Detective, murder, New York

User profile

Basic method

- Compute the similarity of an unseen item with the user profile
- Combine multiple measures:
  - E.g., $sim(U, I) = \alpha\, sim_{genre}(U, I) + \beta\, sim_{keywords}(U, I)$
- Suggest most similar item(s) to user

BY SA
Bo Thiesson

Similarity – why do we combine measures?

Some attributes may not be equally important

- apriori knowledge
- Statistical analysis

A similarity measure for one type of attribute make not make sense for another type. E.g:

- Continuous, nominal attributes (price, lat/long location,…) → reciprocal Euclidian distance (or more general Minkowski distance)
- Boolean, categorical attributes (book cover type, genre,…) → simple equality measure or Jaccard similarity "$(\frac{intercept}{union})$" for binarized featurization
- Unstructured text → cosine similarity

One similarity measure may cover multiple attributes at the same time and, hence, capture correlations – though!

cc BY SA
Bo Thiesson

Looking at the unstructured text…

…we re-use all the good stuff that we have learned about in document modeling and sentiment analysis, but in a slightly different context

BY SA
Bo Thiesson

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

Simple keyword representation has its problems

- in particular when automatically extracted as
  - 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 in multi-dimensional Euclidian space
  - weighted term vector
- TF: Measures, how often a term appears in a document (item)
  - assuming that important terms appear more often
- IDF: Measures, how often a term appears across documents (items)
  - Aims to reduce the weight of terms that appear in many documents

BY SA
Bo Thiesson

TF-IDF (cont)

- **Given a keyword i and a document j**
- $TF(i,j)$
  - term frequency of keyword i in document j
- $IDF(i)$
  - inverse document frequency calculated as $\boldsymbol{IDF(i) = log\frac{N}{n(i)}}$
    - N : number of all recommendable documents
    - $n(i)$: number of documents from N in which keyword i appears
- $TF - IDF$
  - is calculated as: $\boldsymbol{TF\text{-}IDF(i,j) = TF(i,j) * IDF(i)}$

cc BY SA
Bo Thiesson

Example TF representation

Term frequency:

- Each document is a count vector in $\mathbb{N}^{|v|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	1.51	0	3	5	5	1
worser	1.37	0	1	1	1	0

Vector v with dimension $|v| = 7$

Example taken from http://informationretrieval.org

cc BY SA
Bo Thiesson

Example TF-IDF representation

Combined TF-IDF weights

- Each document is now represented by a real-valued vector of $TF\text{-}IDF$ weights $\in \mathbb{R}^{|v|}$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4					
Caesar	232					
Calpurnia	0					
Cleopatra	57					
mercy	1.51					
worser	1.37					

	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

Example taken from http://informationretrieval.org

cc BY SA
Bo Thiesson

Improving the vector space model

Vectors are usually long and sparse

remove stop words

- They will appear in nearly all documents.
- e.g. "a", "the", "on", …

use stemming

- Aims to replace variants of words by their common stem
- e.g. "went" ⟹ "go", "stemming" ⟹ "stem", …

size cut-offs

- only use top n most representative words to remove "noise" from data
- e.g. use top 100 words

cc BY SA
Bo Thiesson

Improving the vector space model (cont.)

Use lexical knowledge, use more elaborate methods for feature selection

- Remove words that are not relevant in the domain

Detection of phrases as terms

- More descriptive for a text than single words
- e.g. "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.."
  - The word "vegetarian" will receive a higher weight then desired

    ⇨ an unintended match with a user interested in vegetarian restaurants
  - *Could use the NOT_ prefix from sentiment analysis lecture*

BY SA
Bo Thiesson

Comparing the vectors (users/items)

Usual **similarity metric** to compare vectors:

Cosine similarity (angle)

- Cosine similarity is calculated based on the angle between the vectors
- Compensates for the effect of different document lengths

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

Query "Caesar Calpurnia"

- Similarity between query and documents

Norm. TF-IDF	Antony and Cleopatra	Julius Caesar	Hamlet	Query
Caesar	0.83	0	1	0.35
Calpurnia	0	1	0	0.94
Cleopatra	0.55	0	0	0
Similarity to query	0.29	**0.94**	0.35	1

BY SA

Memory- and model-based filtering/ranking approaches

Memory-based approached

- E.g., kNN based methods
- Non-parametric, items are "memorized"
- At runtime, raw items are used for the predictions
- May not scale
- Trivial adaptation as new ratings appear, just add the rated items

Model-based approaches

- E.g., Naïve Bayes, logistic regression, SVM,…
- Parametric, items are represented by a compressed parameterization in a statistical model
- At runtime, only the learned model is used to make predictions
- Scales well
- May or may not be simple to adapt model over time (depends on model). Worst case, models are updated/re-trained periodically

BY SA

Ratings

Explicit ratings

- Most commonly used
  - Binary (like, not like)
  - 1 to 5, 1 to 7 Likert response scales
- Typically only one rating per user and item, including time-stamp

Some research topics

- Data sparsity
  - Users not always willing to rate many items
  - How to stimulate users to rate more items?
- Which items have (not) been rated?
  - Ratings not missing at random
- Optimal granularity of scale
- Multidimensional ratings
  - multiple ratings per movie (acting, directing, …)

BY SA

Ratings (cont.)

Implicit ratings (feedback)

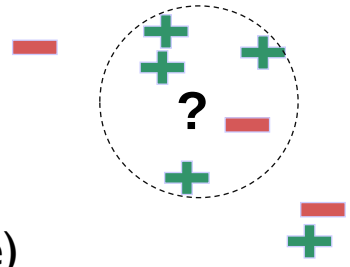
- Typically collected by the web shop or application in which the recommender system is embedded
  - Clicks, page views, time spent on some page, demo downloads …
  - Multiple events over time
- Can be collected constantly and do not require additional efforts from the side of the user

Research topics

- Correct interpretation of the (strength of the) action
  - Buy something for a friend, accidental clicks
  - How to interpret shopping cart actions (recommend or not?)
- Huge amounts of data to be processed

BY SA
Bo Thiesson

Recommending items: kNN



Find item candidates: k nearest neighbors

- Given a set of documents D already rated by the user (like/dislike)
  - Either explicitly via user interface
  - Or implicitly by monitoring user's behavior
- Find the k nearest neighbors of an not-yet-seen item i in D
  - Use similarity measures (cosine similarity, etc.) to capture similarity of two documents

Rating predictions

- Take these neighbors to predict a rating for i
  - e.g. $k = 5$ most similar items to i.
    4 of k items were liked by current user ⇨ item i will also be liked by this user
- Variations:
  - Varying neighborhood size k
  - lower/upper similarity thresholds to prevent system from recommending items the user already has seen
- Good to model short-term interests / follow-up stories
- Used in combination with method to model long-term preferences

BY SA
Bo Thiesson

Relevance Feedback – Rocchio's Algorithm

Quality of Information Retrieval depends on user's capability to formulate queries with suitable keywords

Adaptive query-based retrieval: Rocchio's method

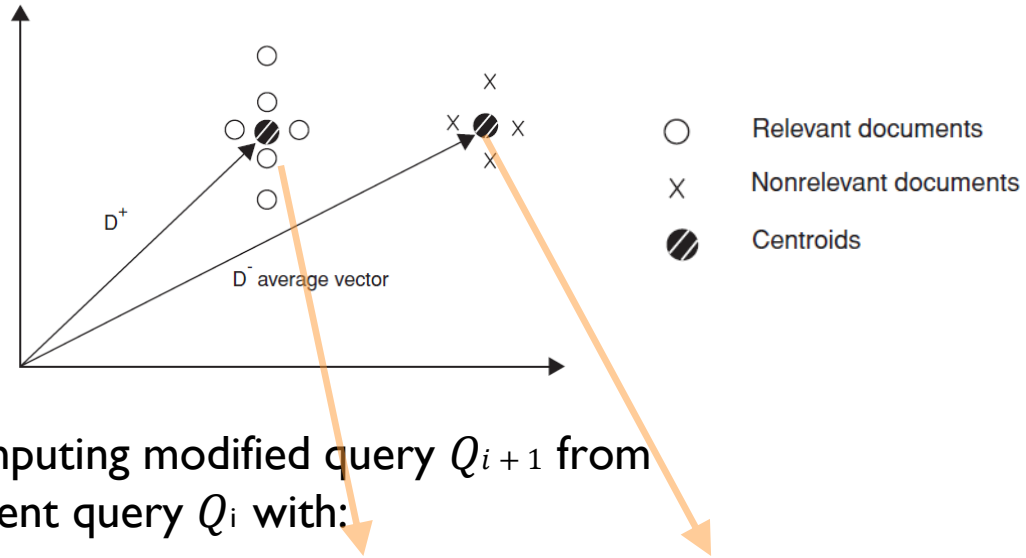
- Users are allowed to rate (relevant/irrelevant) retrieved documents (feedback)
- The system then learns a prototype of relevant/irrelevant documents
- Queries are then automatically extended with additional terms/weight of relevant documents

Intuitive, but no theoretical basis; works well in practice!

cc BY SA
Bo Thiesson

Rocchio details

- Document collections D^+ (liked) and D^- (disliked)
  - Calculate prototype vector for these categories.



- Computing modified query Q_{i+1} from current query Q_i with:

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

  - **Often only positive feedback is used**
  - More valuable than negative feedback ("little to like and much to dis-like")

α, β, γ used to fine-tune the feedback

α weight for original query

β weight for positive feedback

γ weight for negative feedback

CC BY SA
Bo Thiesson

Naïve Bayes Classifier – Learning the Model

Prediction model:

$$score(x, c) \propto \left(\prod_{i=1}^{n} p(x_i|c)\right) p(c)$$

Distribution depends on the actual attribute x_i; here we assume Bernoulli

…is simple counting

- Estimate $p(c)$ for all $c \in C$:
  - Count the number of ratings by the user: N
  - Count the number of rated items with rating c: $N(c)$

$$p(c) = \frac{N(c)}{N}$$

- Estimate $p(x_i|c)$ for all possible words in vocabulary $x_i \in X$ (or all possible meta features; e.g. genre, director) and all possible rating classes $c \in C$
  - Count the number of times the word x_i appears across all rated items with rating c: $N(x_i, c)$

$$p(x_i|c) = \frac{N(x_i, c)}{N(c)}$$

$$p(not\ x_i|c) = 1 - p(x_i|c)$$

BY SA
Bo Thiesson

Naïve Bayes Classifier – Adapt model

…simply add one to appropriate counts

user rated item with word x_i as rating c

Prediction model:

$$score(x, c) \propto (\prod_{i=1}^{n} p(x_i|c))p(c)$$

- Estimate $p(c)$ for all $c \in C$:
  - Add one to number of ratings by the user: N
  - Add one to number of rated items with rating c: $N(c)$

$$p(c) = \frac{N(c) + 1}{N + 1}$$

- Estimate $p(x_i|c)$ for all possible words in vocabulary $x_i \in X$ (or all possible meta features; e.g. genre, director) and all possible rating classes $c \in C$
  - Add one to number of times the word x_i appears across all rated items with rating c: $N(x_i, c)$

$$p(x_i|c) = \frac{N(x_i, c) + 1}{N(c) + 1}$$

$$p(not\ x_i|c) = 1 - p(x_i|c)$$

BY SA
Bo Thiesson

Linear classifiers (for like / not-like)

- Most learning methods aim to find coefficients of a linear model (hyperplane) that best separates classes

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} - b = \sum_i w_i x_i - b$$

- Rating (classification) based on checking

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} - b > 0$$

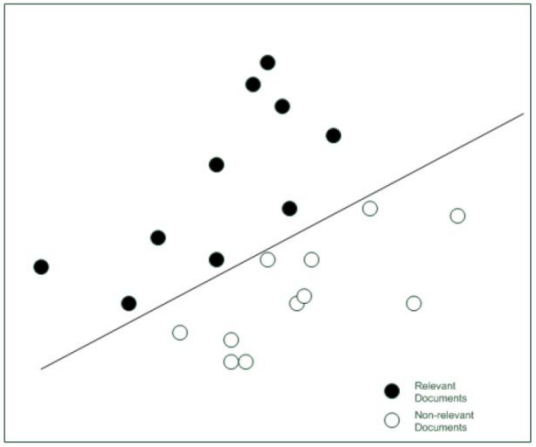
- Can be learned by minimizing squared error loss for rated items

$$Loss = \sum_{rated\ items} \left((f(\boldsymbol{x}^n) - c^n \right)^2$$

- Widrow-Hoff rule, delta rule, stochastic gradient descent can be used to incrementally update model

$$w \leftarrow w - \alpha(f(\boldsymbol{x}) - \mathrm{c}) \cdot \boldsymbol{x}$$

- Can be generalized to multiple rating classes!



Not the best loss-function, but is decent in many situations and leads to simple optimization

More of this method in next lecture

BY SA
Bo Thiesson

Other (Rating) Classifiers

- Decision trees
- Linear & quadratic discriminant analysis
- Conditional maximum entropy models / Logistic regression
- SVM
- Neural net
- …

BY SA

Advantages & Drawbacks

- Content-based techniques do not require a user community
  - They however require content information (may be limited)
  - Recent new types of Web 2.0 "content" information
    - Wikipedia, Linked Data, Social Tags, Social Media posts…
- New user problem: Significant ramp-up phase required
  - The presented approaches learn a model of the user's interest preferences based on explicit or implicit feedback
    - Deriving implicit feedback from user behavior can be problematic or at least imprecise
- Transparency: Explanations exclusively based on users rating activity
- New items not at problem
- Serendipity problem: Danger exists that recommendation lists contain too many similar items
  - Algorithms tend to propose "more of the same"
  - Or: too similar new items

BY SA
Bo Thiesson

Serendipity Problem:
A prediction with 0% error!

Past profile

- You liked Star Wars and
- you gave five stars to Star Wars I to Star Wars II

My prediction is that you

- will give five stars to Star Wars III to Star Wars Infinity
- I recommend more Star Wars movies

Exact rating predictions might not be enough

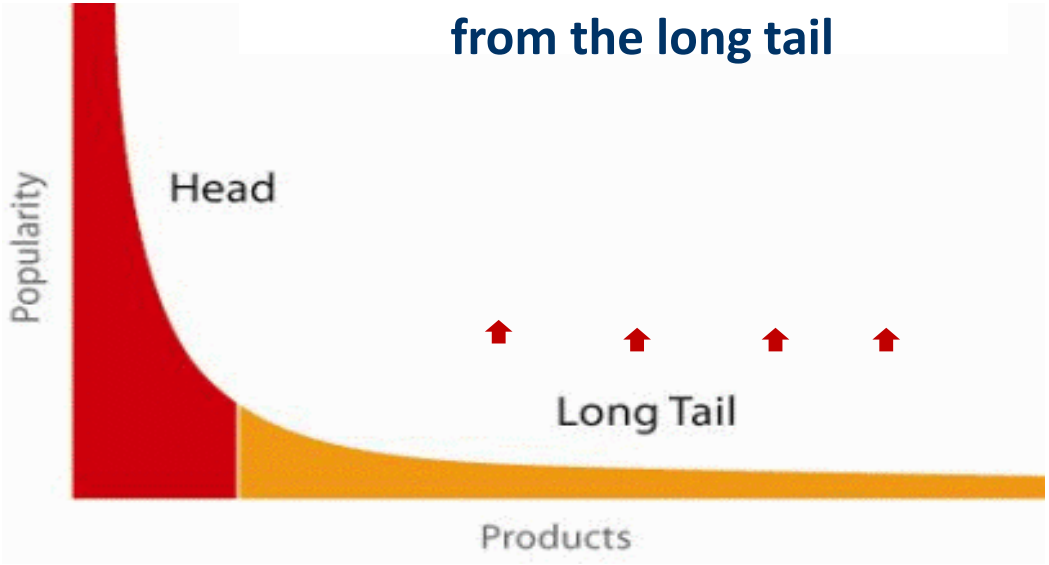
- No surprise
  - no extra sales and limited value
- No variety in recommendations …

cc BY SA
Bo Thiesson

When does a RS do its job well?

- "Recommend widely unknown items that users might actually like!"

Recommend items from the long tail



cc BY SA

Outline

- Motivation
- Basic paradigms
  - Content-based
  - Collaborative
  - Hybrid
- Content-based
  - General principle
  - Similarity measures
  - Rating feedback
  - Memory- and model-based approaches