Recommender systems

Nathalie Denos

Mosig IAR

December 5th, 2016

15:30-17:00

Personalize / Recommend

Why recommend

- increasing value of knowing
  - the right information
  - at the right moment
  - as soon as it is available
- increasing amount
  - of available information,
  - of information consumption

The paradox of choice – Barry Schwartz (2005)

The long tail – Chris Anderson (2008)

The value of recommendation

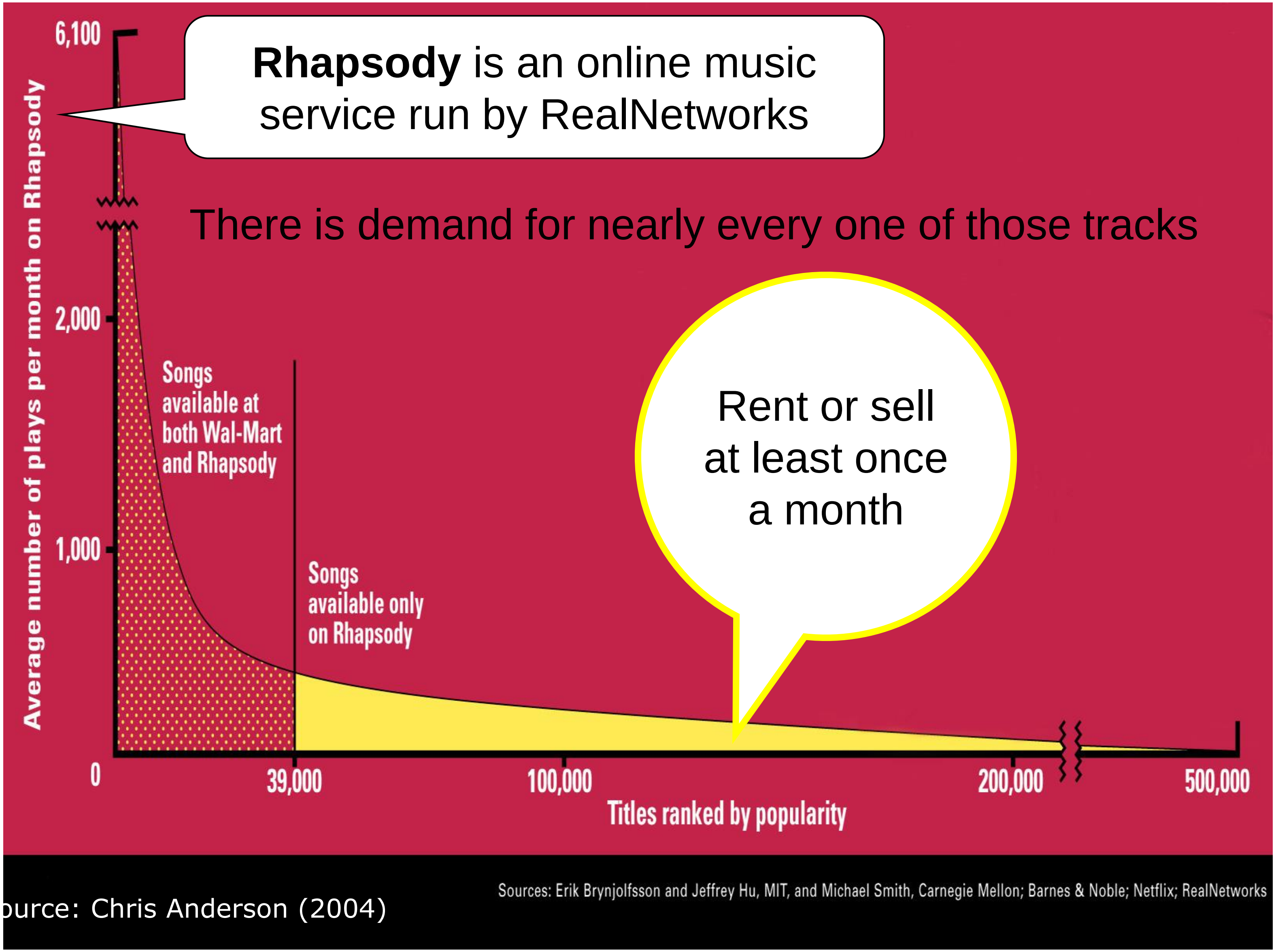
- Google News: more clickthrough
- Amazon: more sales
- Netflix: more movies watched
  - at least many movies that are watched were recommended

From scarcity to abundance

- Shelf space is a scarce commodity for traditional retailers
  - Also for: TV networks, movie theaters, musicians,…
- The web enables near-zero-cost dissemination of information about products
  - From scarcity to abundance
- More choice necessitates better filters
  - Recommendation engines
  - How Into Thin Air (1998) made Touching the Void (1988) a bestseller…

Amazon recommendations based on buying patterns

The Long Tail



Searching, Filtering, Recommending

Filtering / recommendation approaches

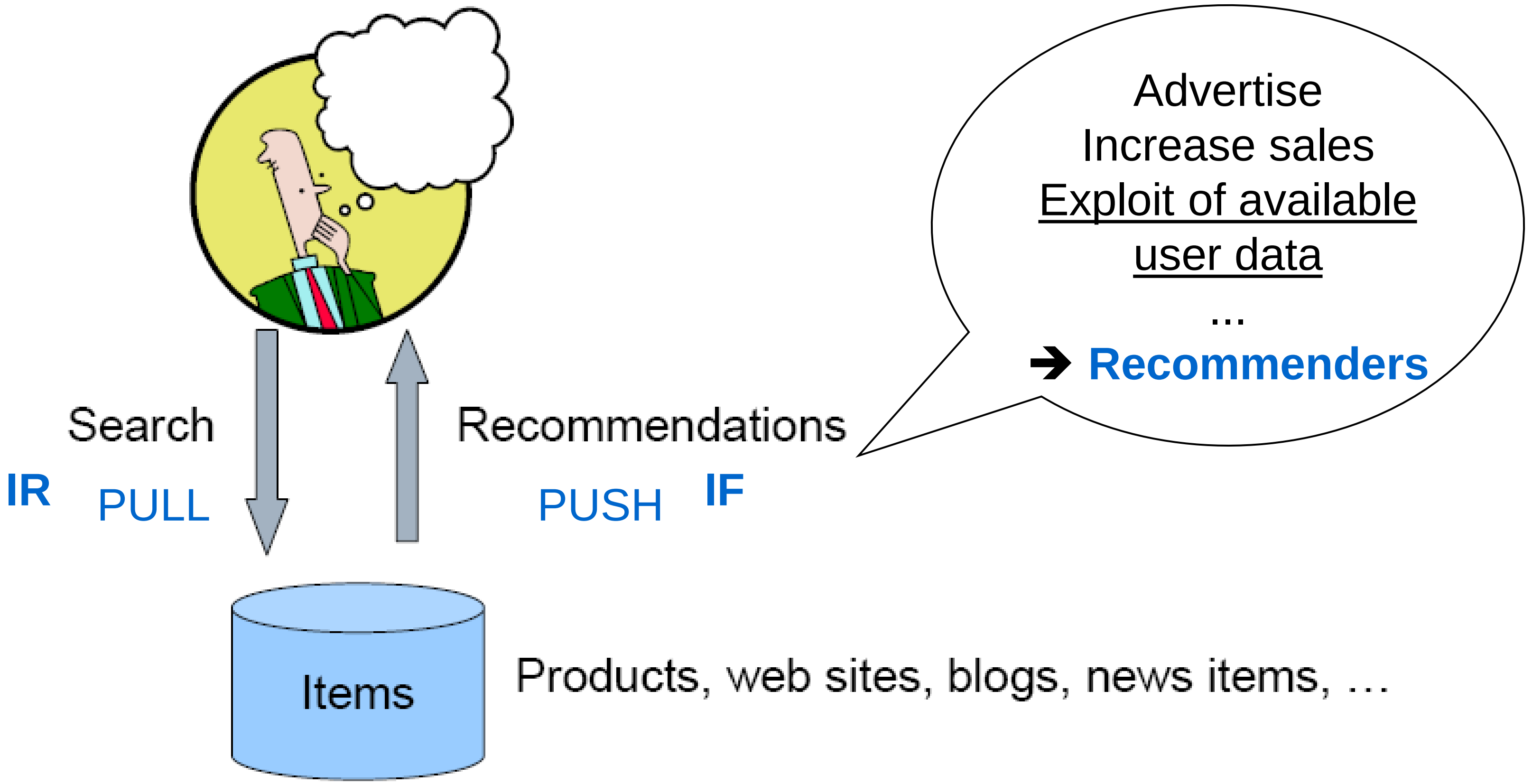
Personalization / Recommendation

Personalized information access

- personalized information retrieval
- information filtering $\leftrightarrow$ recommendation

Recommendation: no query

IR vs IF: PULL and PUSH



History

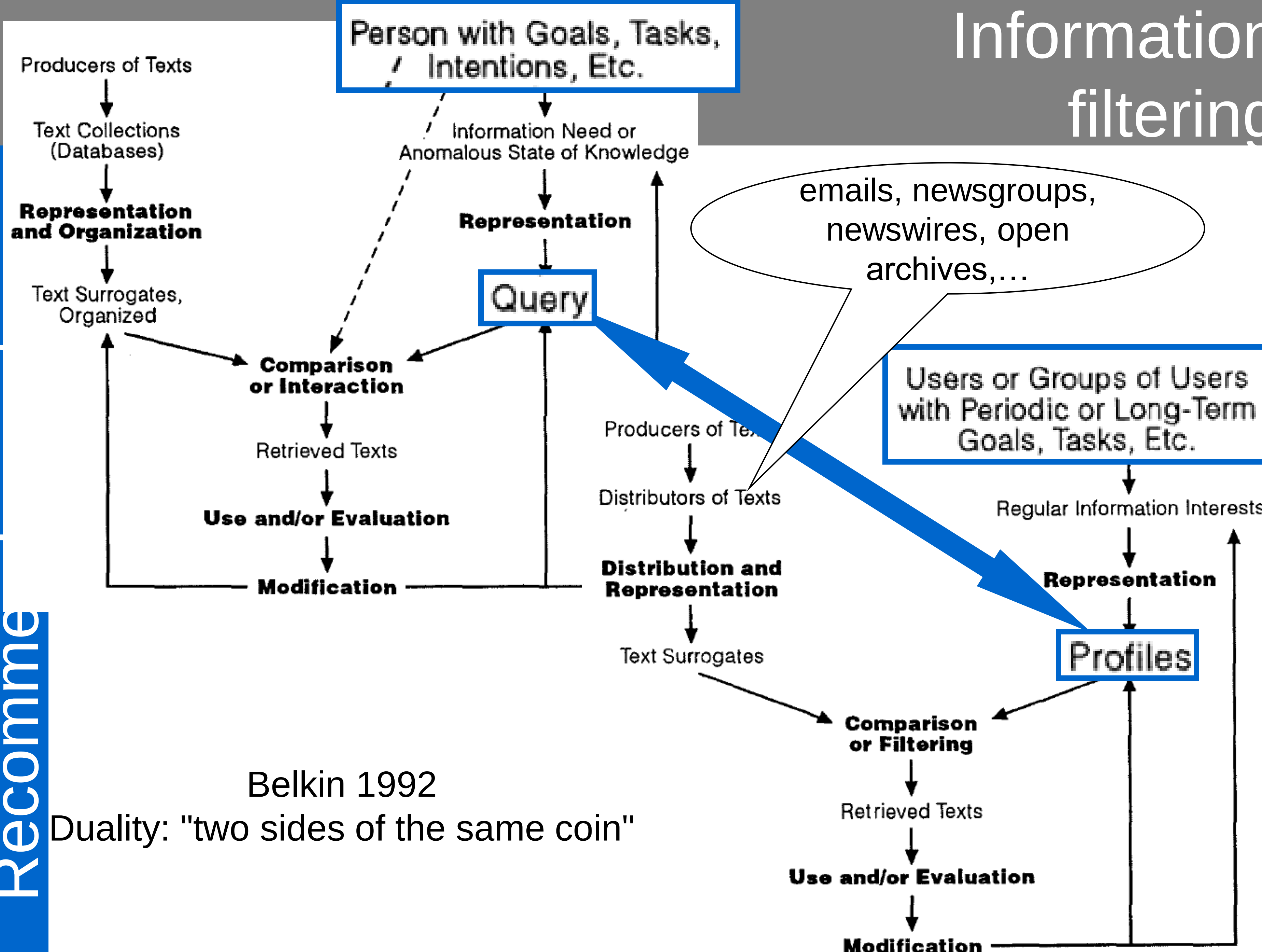
- Information filtering
  - ~1985, email filtering (junk mail)
  - Naming/clarifying:
    - Nick Belkin 1992, Doug Oard 1995

Belkin, N. J. and Croft, W. B. 1992. Information filtering and information retrieval: two sides of the same coin?. Commun. ACM 35, 12 (Dec. 1992), 29-38.

  - A variety of processes involving the delivery of information to people who need it
  - Generally, the goal of an information filtering system is to sort through large volumes of dynamically generated information and present to the user those which are likely to satisfy his or her information requirement.
- 1st ACM RecSys conference 2007

Information filtering

Recomme



Belkin 1992

Duality: "two sides of the same coin"

- Content-based recommendation…

  …is an outgrowth and continuation of information filtering research (Belkin & Croft 1992)

  » Burke 02
- But…
  - the collaborative approach came first
  - born in 1992 > recommending Usenet news,…
    - David Goldberg, David Nichols, Brian Oki, and Douglas Terry, Using collaborative filtering to weave an information tapestry, *Communications of the ACM*, vol. 35, No. 12, 1992, p. 61-70.
    - Konstan, J. A. Miller, B. N. Maltz, D. Herlocker, J. L. & Gordon, L. R. Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News' in Special section: recommendation systems in CACM March 1997, Vol. 40, No. 3, pp77-87.

Recommender systems today…

- Defined in a very broad way (Burke 2002)

  Any system that

  - *produces individualized recommendations as output,*

  or

  - *has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options*

- Fully integrated in e-business Web sites

  > users are often "customers"

  > items are often "products"

Examples

Recommender systems

Recommendation Types

- Editorial
- Simple aggregates
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users
  - Amazon, Netflix, …

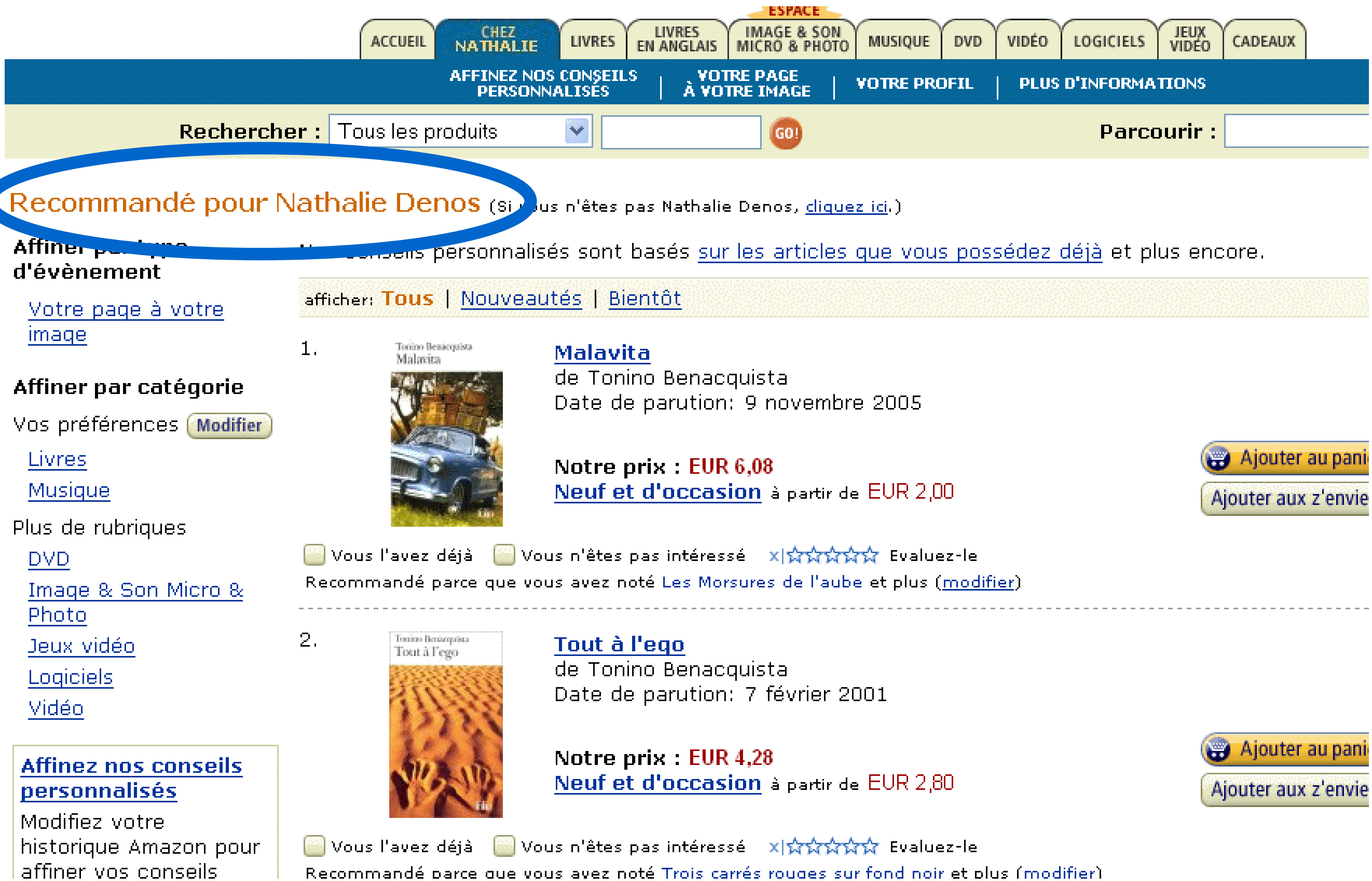
Amazon

- Personalized?

http://www.amazon.fr/

Amazon

Recommender systems



http://www.amazon.fr/ (connecté)

MovieLens

- Personalized

http://movielens.umn.edu

The recommender problem

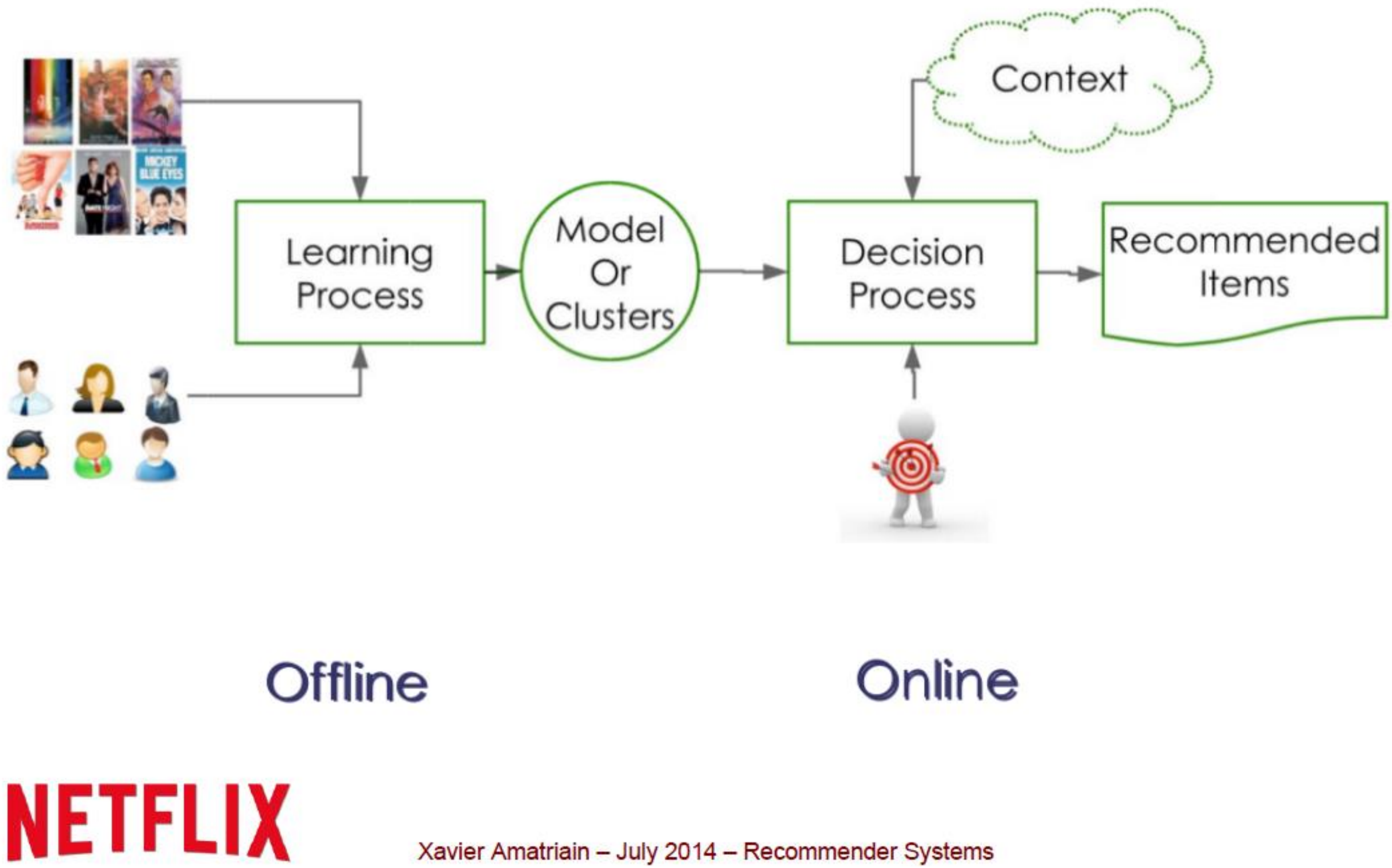
- Estimate a utility function that automatically predicts how a user will like an item
- Based upon… whatever is available!
  - past behavior
  - relations to other users
  - item similarity
  - context
  - …

The recommender problem

- U = set of Users (customers,…)
- I = set of Items
- Utility function u: $U \times I \to R$
  - R = set of ratings
  - R is a totally ordered set
  - e.g., 0-5 stars, real number in [0,1]

- for each current user, choose items that maximize u

A two-step process



What matters

What matters for recommenders?

- learning process
- user interface, user interaction (user studies)
- information browsing, presentation, visualization, …

Approaches to recommendation

- Collaborative Filtering: based on users past behavior only
  - User-based: find similar users and recommend what they liked
  - Item-based: find similar items to those that were previously liked
- Content-based: based on item features
- Demographic: based on user features
- Social recommendations: trust-based
- Hybrid: combine

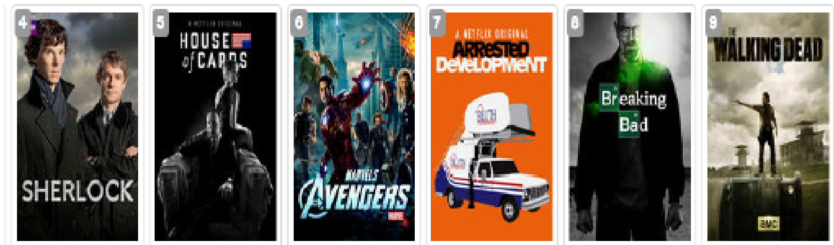
Recommender systems

The recommender problem

- U = set of Users (customers,…)
- I = set of Items
- Utility function u: U x I → R
  - R = set of ratings
  - R is a totally ordered set
  - e.g., 0-5 stars, real number in [0,1]

- for each current user, choose items that maximize u

Utility matrix



2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

NETFLIX

Key Problems

- Gathering "known" ratings for matrix
- Extrapolate unknown ratings from known ratings
  - Mainly interested in high unknown ratings
- Evaluating extrapolation methods

Gathering Ratings

- Explicit
  - Ask people to rate items
  - Doesn't work well in practice – people can't be bothered
- Implicit
  - Learn ratings from user actions
  - e.g., purchase implies high rating
  - What about low ratings?

Favor implicit: easier to get, less noisy

Extrapolating utilities

- Key problem: matrix U is sparse
  - most people have not rated most items

  Netflix prize (1M$ 2009)
  - 500 000 users x 17 000 items
    - 8 500 M slots
    - 100 M ratings
- Main approaches
  - Content-based
  - Collaborative

Content-based filtering

Content-based recommenders

- Main idea: recommend items i to user u similar to previous items rated highly by u
- Movie recommendations
  - recommend movies with same actor(s), director, genre, …
- Websites, blogs, news
  - recommend other sites with “similar” content

Item Profiles

- For each item, create an item profile
- Profile is a set of features
  - movies: author, title, actor, director,…
  - text: set of "important" words in document
- How to pick important words?
  - Usual heuristic is TF.IDF (Term Frequency times Inverse Doc Frequency)

TF.IDF

f_{ij} = frequency of term t_i in document d_j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

n_i = number of docs that mention term i

N = total number of docs

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF score $w_{ij} = TF_{ij} \, x \, IDF_i$

Doc profile = set of words with highest TF.IDF scores, together with their scores

User profiles and prediction

- User profile possibilities:
  - Weighted average of rated item profiles
  - Variation: weight by difference from average rating for item
  - …
- Prediction heuristic
  - Given user profile **u** and item profile **i**, estimate $u(\mathbf{u},\mathbf{i}) = \cos(\mathbf{u},\mathbf{i}) = \mathbf{u}.\mathbf{i}/(|\mathbf{u}||\mathbf{i}|)$

Model-based approaches

- For each user, learn a classifier that classifies items into rating classes
  - liked by user and not liked by user
  - e.g., Bayesian, regression, SVM
- Apply classifier to each item to find recommendation candidates

Limitations of content-based approach

- Finding the appropriate features
  - e.g., images, movies, music
- Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
- Recommendations for new users
  - How to build a profile?

Collaborative filtering

Collaborative Filtering

- given target user u
- find set D of other users whose ratings are “similar” to u’s ratings (neighbors)
- identify the items neighbour users liked
- generate a prediction (rating) that would be given by u to each of these items
- recommend the top N items

Ingredients for CF

- List of **m Users** and a list of **n Items**
- Each user has a **list of items** with associated **opinion**
  - **Explicit opinion** - a rating score
  - Sometime the rating is **implicitly** – purchase records or listen to tracks
- **Active user** for whom the CF prediction task is performed
- **Metric** for measuring **similarity between users**
- Method for selecting a subset of **neighbors**
- Method for **predicting a rating** for items not currently rated by the active user.

Similar users

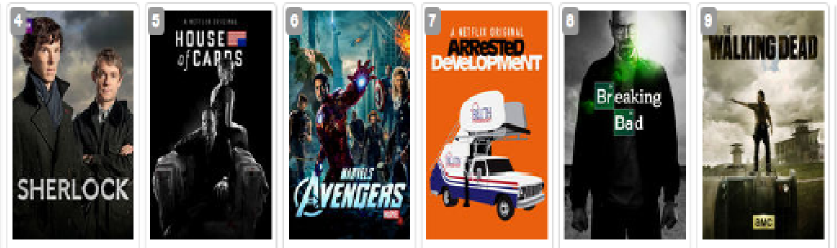
- Let r_x be the vector of user x's ratings
- Cosine similarity measure
  - sim(x,y) = cos(r_x , r_y)
- Pearson correlation coefficient
  - S_{xy} = items rated by both users x and y

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r_x})(r_{ys} - \bar{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r_x})^2 (r_{ys} - \bar{r_y})^2}}$$

Rating predictions

- Let D be the set of *k* users most similar to *u* who have rated item *i*
- Possibilities for prediction function (item i):
  - $r_{ui} = K \sum_{d \in D} r_{di}$
  - $r_{ui} = K \sum_{d \in D} sim(u,d)\ r_{di})$
  - $r_{ui} = r_u + K (\sum_{d \in D} sim(u,d)\ (r_{di} - r_d))$
  - Other options…

User-based CF

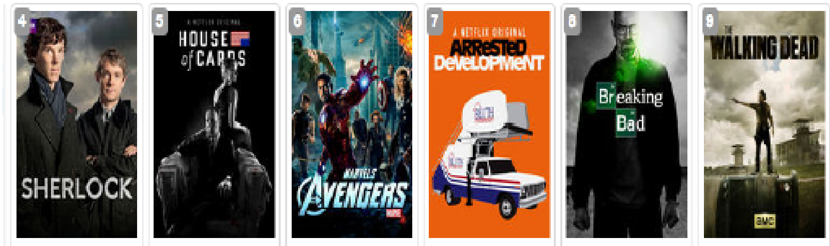


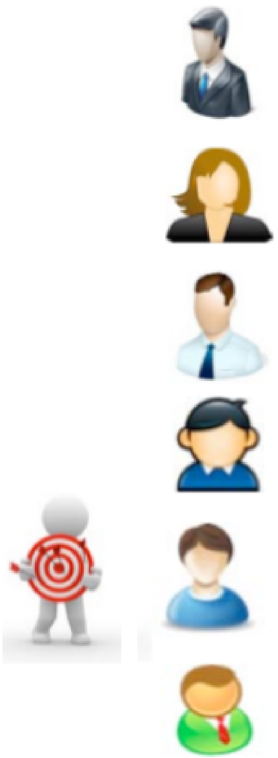
						sim(u,v)
2			4	5		NA
5		4			1	
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

Recommender systems

NETFLIX

User-based CF

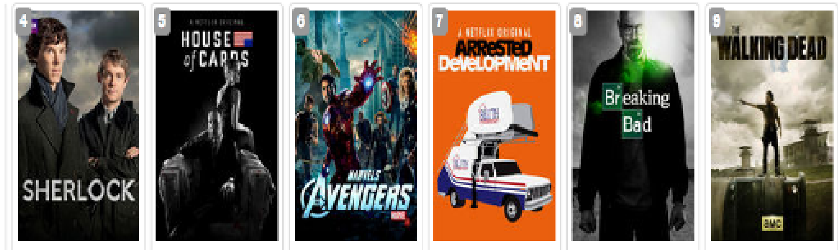




						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		
	1		5		4	
		4			2	
4	5		1			NA

NETFLIX

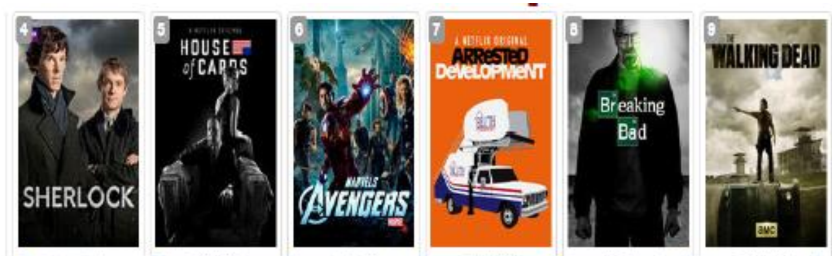
User-based CF



						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	
		4			2	
4	5		1			NA

NETFLIX

User-based CF



						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	-1
		4			2	
4	5		1			NA

User-based CF

						sim(u,v)
2			4	5		NA
5		4			1	0.87
		5		2		1
	1		5		4	-1
3.51*	3.81*	4	2.42*	2.48*	2	
4	5		1			NA

NETFLIX

Xavier Amatriain – July 2014 – Recommender Systems

Scalability

- Expensive step is finding k most similar customers
  - worst case O(N |U|)
  - O(N + |U|)
- Too expensive to do at runtime
  - Need to pre-compute
- Can use clustering

Challenges for user-based CF

- Sparsity – evaluation of large item sets, users purchase under 1%
- Scalability – nearest neighbour computation grows with both users and items
- Poor relationship between likeminded but sparse-rating users
- Solution: reduce dimensional space
- Try item-based CF

Item-based Collaborative Filtering

- So far: User-based collaborative filtering
- Another view
  - For target item i
  - Compute how similar it is to items rated by target user
    - only based on past ratings from other users !
  - Selet k most similar items
  - Predict rating as weighted average on target user's ratings on most similar items
- Can use same similarity metrics and prediction functions as in user-based model
- In practice, it has been observed that item-based often works better than user-based

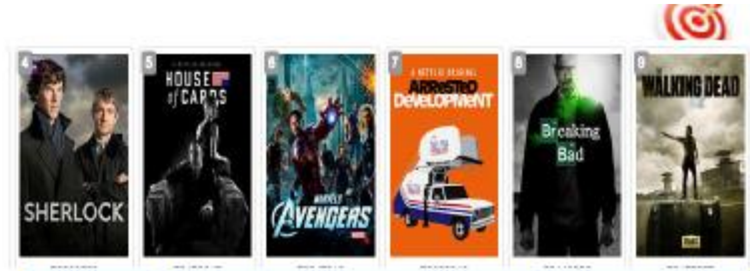
Item-based CF

	SHERLOCK	HOUSE of CARDS	AVENGERS	ARRESTED DEVELOPMENT	Breaking Bad	WALKING DEAD
	2			4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

sim(i,j) -1

Xavier Amatriain – July 2014 – Recommender Systems

Item-based CF

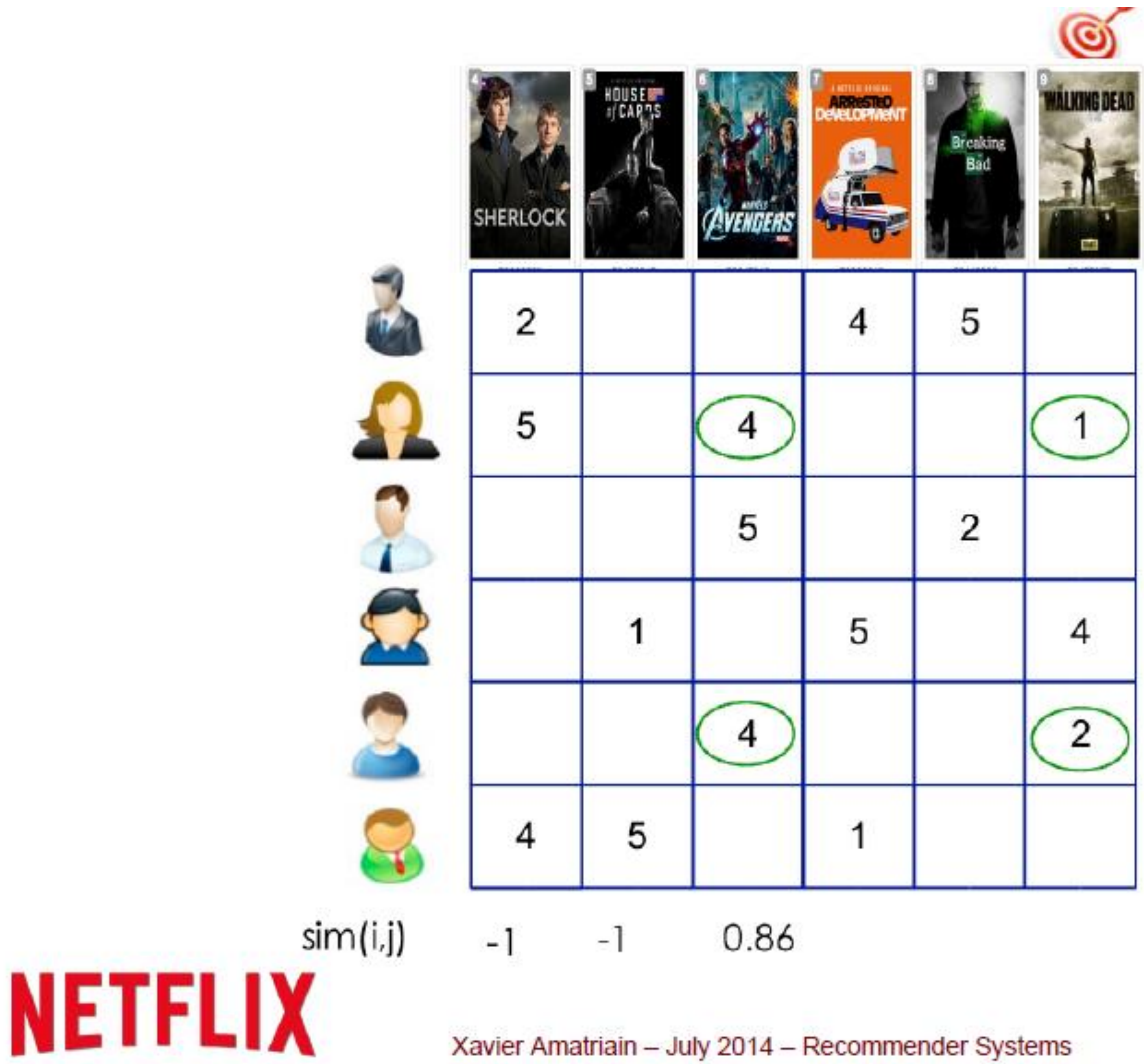


2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

sim(i,j) -1 -1

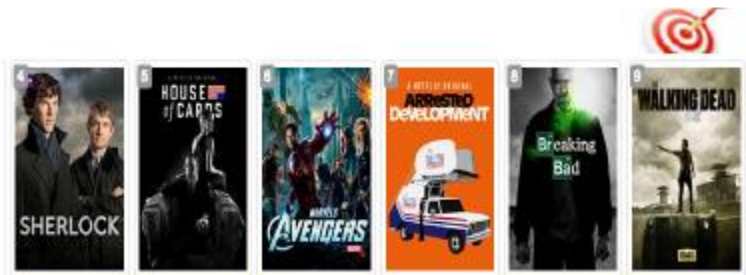
Xavier Amatriain – July 2014 – Recommender Systems

Item-based CF



	Sherlock	House of Cards	Avengers	Arrested Development	Breaking Bad	Walking Dead
User 1	2			4	5	
User 2	5		4			1
User 3			5		2	
User 4		1		5		4
User 5			4			2
User 6	4	5		1		
sim(i,j)	-1	-1	0.86			

Item-based CF

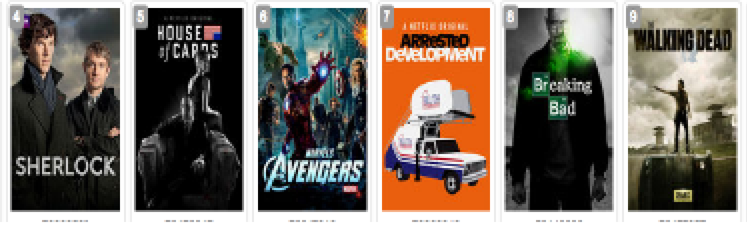


	2			4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		
sim(i,j)	-1	-1	0.86	1		

NETFLIX

Xavier Amatriain – July 2014 – Recommender Systems

Item-based CF

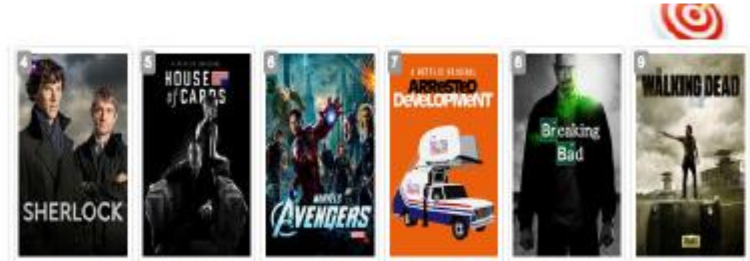


	2			4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		
sim(i,j)	-1	-1	0.86	1	NA	

sim(6,5) cannot be calculated

NETFLIX

Item-based CF



	2			4	5	2.94*
	5		4			1
			5		2	2.48*
		1		5		4
			4			2
	4	5		1		1.12*
sim(i,j)	-1	-1	0.86	1	NA	

Xavier Amatriain – July 2014 – Recommender Systems

Pros and cons of collaborative filtering

- Pros
  - No domain knowledge
  - No item features (and no feature selection)
  - Good enough in most cases
- Cons
  - Bootstrap / cold start (new user / new item)
  - Standardized items (and sparsity)
    - dimensionality reduction techniques
  - Assumption: prior behavior determines current
  - Bottleneck of scalability for similarity computation
    - neighbourhood offline

Hybrid Methods

- Implement two separate recommenders and combine predictions
- Add content-based methods to collaborative filtering
  - item profiles for new item problem
  - demographics to deal with new user problem

Evaluation

General thoughts / Baseline

- Serendipity
- Personalized vs. non personalized
  - personalized
    - neighbour users are different for each user
  - non-personalized
    - neighbours = all users
- Popularity as a baseline

Measures

- Compare predictions with known ratings
  - Mean Average Error $\frac{1}{N}\sum_{p,m}|\mathrm{pred}(p,m) - \mathrm{test}(p,m)|$
  - Root-mean-square error

$$\sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}$$

Personalized vs. non personalized

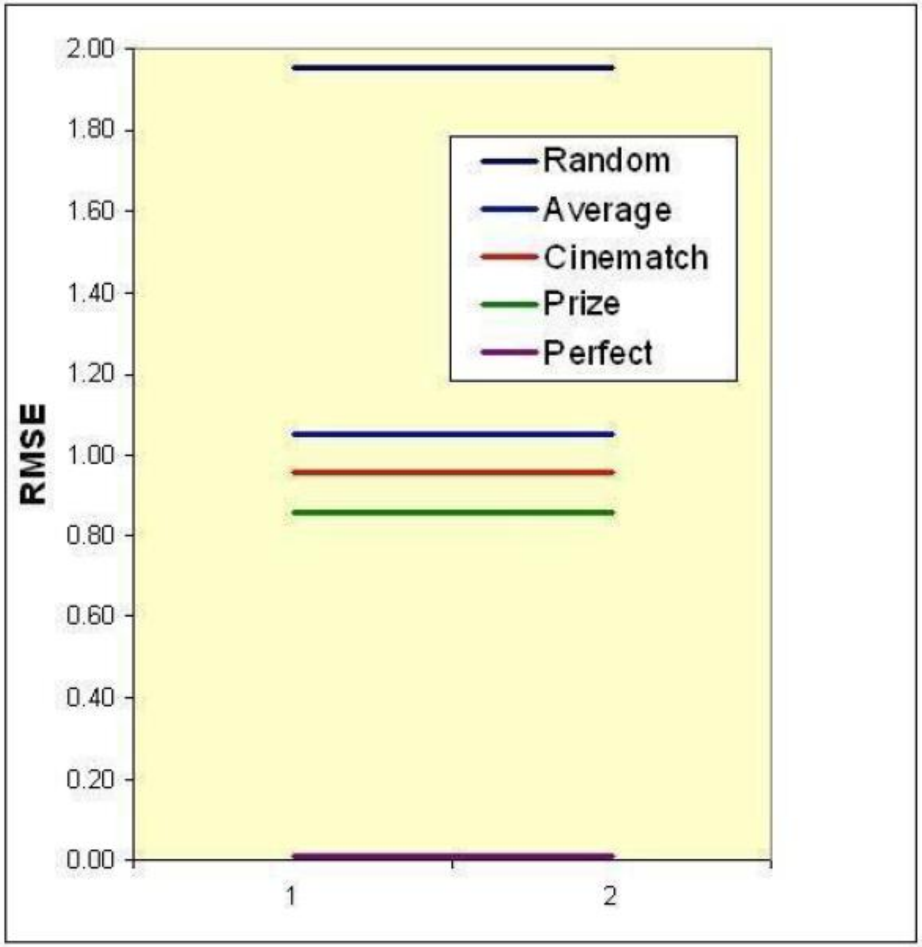
Data Set	users	items	total	density	MAE Non Pers	MAE Pers
Jester	48483	100	3519449	0,725	0,220	0,152
MovieLens	6040	3952	1000209	0,041	0,233	0,179
EachMovie	74424	1649	2811718	0,022	0,223	0,151

$$MAE_{NP} = \frac{\sum_{i,j} |v_{ij} - v_j|}{num.ratings}$$

v_{ij} is the rating of user i for product j and v_j is the average rating for product j

Random / Average / Best

- Netflix Prize's first conclusion: it is really extremely simple to produce "reasonable" recommendations and extremely difficult to improve them.



Problems with Measures

- Narrow focus on accuracy sometimes misses the point
  - Coverage
    - Number of items/users for which system can make predictions
  - Prediction Diversity
  - Prediction Context

Wrapping up

Recommendation: what works

- Collaborative filtering
  - agnostic to domain
  - good performance in general
- Implicit ratings
  - easier to get
  - less noisy
- Dealing with
  - sparsity: dimensionality reduction
    - matrix factorization, clustering, projection (PCA…)
  - scalability: O(mn) worst case > O(m+n)
    - clustering techniques (K-means)
  - cold-start: hybridize

Stakes

- economique / e-commerce
  - suggest items to buy
    - precision… honesty ? attacks ?
    - confidence in the system, explanations, transparence, control
  - limit user effort?
    - personalized service has a cost for the user
- diversity
  - give access…
    - more information, rare things?
    - only at the price of a nice balance between… precision / novelty / diversity
- privacy
  - to be controlled
    - I say everything for a better service
    - securing, integrating profiles

References

References

- Books
  - Recommender systems: an introduction – Dietmar Janach (2010)
  - Recommender Systems: The Textbook – Charu C. Agarwal (2016)
  - Recommender systems handbook – Francesco Ricci (2015)
- Articles
  - Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 4 (Nov. 2002), 331-370, 2002.
  - and… Belkin 1992, Goldberg 1992
- Some slides from
  - Anand Rajaraman and Jeffrey D. Ullman. *Course CS 345 on Data Mining*, Stanford University, California, Autumn 2006.
  - Xavier Amatriain MLSS'14 tutorial http://technocalifornia.blogspot.fr/2014/08/introduction-to-recommender-systems-4.html

Research communities involved

- user modeling
- machine learning
- (adaptive hypermedia)
- (digital libraries)
- the Semantic Web
- human-computer interaction
- information visualization
- information retrieval
- recommender systems

Academic milestones

- Specialized conferences
  - User Modelling
  - Adaptive Hypermedia
  - UM + AH = UMAP (since 2009)
  - Recommender Systems (RecSys)
  - IR in Context (IRIX)
- Journal
  - User Modeling and User-Adapted Interaction, Éditeur Springer Netherlands