Recommender Systems: Practical Aspects, Case Studies

Radek Pelánek

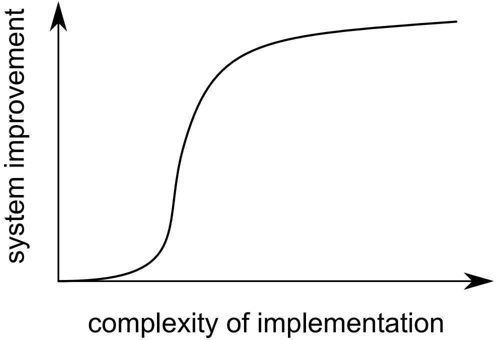
This Lecture

- "practical aspects": attacks, shared accounts, context, ...
- case studies, illustrations of application
- illustration of different evaluation approaches
- specific requirements for particular domains

focus on "ideas", quick discussion (consult cited papers for technical details)

Focus on Ideas

even simple implementation often brings most of the advantage



Focus on Ideas

potential inspiration for projects, for example:

- taking context into account
- highlighting specific aspects of each domain
- specific techniques used in case studies
- analysis of data, visualizations
- evaluation

Attacks on Recommender System

- Why?
- What type of recommender systems?
- How?
- Countermeasures?

Attacks

susceptible to attacks: collaborative filtering

reasons for attack:

- make the system worse (unusable)
- influence rating (recommendations) of a particular item
  - *push attacks* – improve rating of "my" items
  - *nuke attacks* – decrease rating of "opponent's" items

Example

Items

Users	1	2	3	4	5	6	7	
a	+	−		+	+		+	Authentic profiles
b	−	+	+	−	−		−	
c	+	−	+		−	−	−	
d	−	+	+	−				
e	−		−	−	−		−	
f	+	−	+	+	+		+	
g		−	+	+	−	−	+	
h	+	−	+	+	+		?	Target profile
i	+	−	+		−	−	−	Attack profiles
j	−	+	+	−			−	
k	−		−	−	−		−	
l	+	−	+	+	+		−	
m		−	+	+	−	−	−	

Fig. 2 Simplified system database showing authentic user profiles and a number of attack profiles inserted. In this example, user *h* is seeking a prediction for item 7, which is the subject of a product nuke attack.

Robust collaborative recommendation, Burke, O'Mahony, Hurley

Types of Attacks

more knowledge about system $\rightarrow$ more efficient attack

random attack
: generate profiles with random values (preferably with some typical ratings)

average attack
: effective attack on memory-based systems (average ratings $\rightarrow$ many neighbors)

bandwagon attack
: high rating for "blockbusters", random values for others

segment attack
: insert ratings only for items from specific segment

special nuke attacks
: love/hate attack, reverse bandwagon

Example

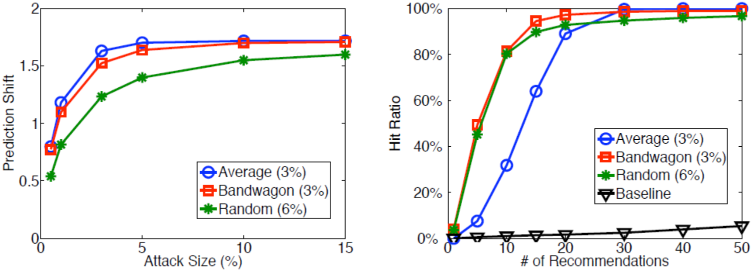


Fig. 3 Prediction shift (left) and hit ratio (right) for product push attacks mounted against the user-based collaborative recommendation algorithm. Hit ratio results relate to a 10% attack size.

Robust collaborative recommendation, Burke, O'Mahony, Hurley

Countermeasures

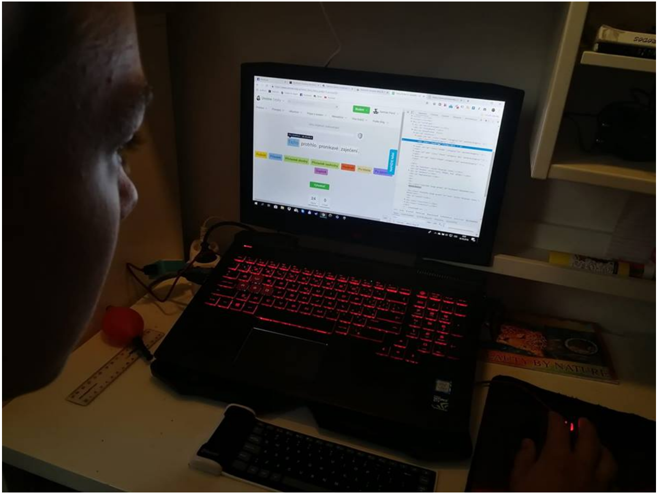
- more robust techniques: model based techniques (latent factors), additional information
- increasing injection costs: Captcha, limited number of accounts for single IP address
- automated attack detection

Attacks and Educational Systems

- cheating $\sim$ false rating
- gaming the system – using hints as solutions

can have similar consequences as attacks
breaks models that (implicitly) assume honest students

Cheating Using Page Source Code



Users and IDs

common (implicit) assumption in recommender system:

database ID $\sim$ one person

when violated?

Shared Accounts

Top-N Recommendation for Shared Accounts (2015)

typical example: family sharing single account

Is this a problem? Why?

Shared Accounts

Top-N Recommendation for Shared Accounts (2015)

typical example: family sharing single account

Is this a problem? Why?

- dominance: recommendations dominated by one user
- generality: too general items, not directly relevant for individual users
- presentation

Shared Account: Evaluation

- hard to get "ground truth" data
- log data insufficient

How to study and evaluate?

Shared Account: Evaluation

- hard to get "ground truth" data
- log data insufficient

How to study and evaluate?

- artificial shared accounts – mix of two accounts
- not completely realistic, but "ground truth" now available
- combination of real data and simulation

Shared Account: Example

Table 3: Example of user 562 suffering from sharing an account with user 4385.

user ID	562	4385
$I(u)$	Wes Craven's New Nightmare, The Exorcist III, Serial Mom, Scream, Scream 2, The Blair Witch Project, Good Will Hunting, Misery, Interview with the Vampire, Candyman, Freddy's Dead: The Final Nightmare	American Beauty, The Shawshank Redemption, Being John Malkovich, L.A. Confidential, Boys Don't Cry, Croupier, Dogma, Cider House Rules, Girl Interrupted, Saving Grace, The Talented Mr. Ripley
individual top-5: IB, $k = 25$	A Nightmare on Elm Street, Halloween, Halloween:H20, The Shining, Seven	Pulp Fiction, Fargo, The Sixth Sense, The Silence of the Lambs, Shindler's List
$R_{sa} = \text{IB}$	The Silence of the Lambs, Fargo, Pulp Fiction, The Sixth Sense, Saving Private Ryan, The Usual Suspects, Shindler's List, Shakespeare in Love, Star Wars: Episode V, The Matrix	
$R_{sa} =$ DAMIB-COVER (p=0.75)	The Silence of the Lambs, Fargo, Schindler's List, A Nightmare on Elm Street, Halloween:H20, Pulp Fiction, Shakespeare in Love, The Shining, The Exorcist, Sleepy Hollow	

Case Studies: Note

- recommender systems widely commercially applied
- nearly no studies about "business value" and details of applications (trade secrets)

Context Aware Recommendations

taking context into account – improving recommendations

- when relevant?
- what kind of context?

Context Aware Recommendations

context:

- **physical** – location, time
- **environmental** – weather, light, sound
- **personal** – health, mood, schedule, activity
- **social** – who is in room, group activity
- **system** – network traffic, status of printers

Context – Applications

- tourism, visitor guides
- museum guides
- home computing and entertainment
- social events

Contextualization

- pre- post- filtering
- model based
  - multidimensionality: user $\times$ item $\times$ time $\times \ldots$
  - tensor factorization

Context – Specific Example

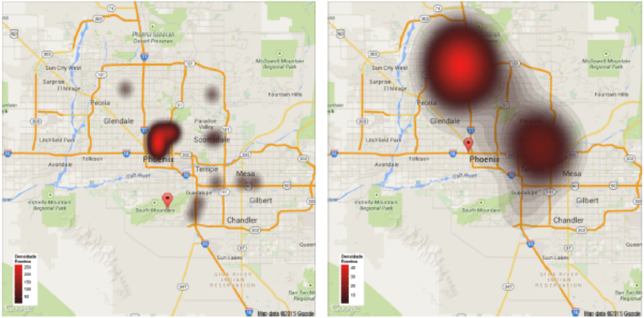
Context-Aware Event Recommendation in Event-based Social Networks (2015)

- social events (meetup.com)
- inherent item cold-start problem
  - short-lived
  - in the future, without "historical data"
- contextual information useful

Contextual Models

social	groups, social interaction
content	textual description of events, TF-IDF
location	location of events attended
time	time of events attended

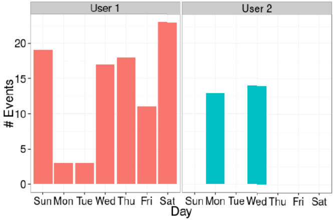
Context: Location



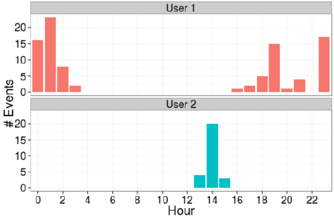
(a) User 1 (b) User 2

Figure 1: Geographical densities of two users.

Context: Time



(a) Distribution per day.



(b) Distribution per hour.

Learning, Evaluation

- machine learning feature weights (Coordinate Ascent)
- historical data, train-test set division
- ranking metric: normalized discounted cumulative gain (NDCG)

Case Studies

- Game Recommendations
- App Recommendations
- YouTube
- Google News
- Yahoo! Music Recommendations
- Book Recommendations for Children

Personalized Game Recommendations

"textbook case study, focusing on basic algorithms"

- Recommender Systems - An Introduction book, chapter 8 Personalized game recommendations on the mobile internet
- *A case study on the effectiveness of recommendations in the mobile internet*, Jannach, Hegelich, Conference on Recommender systems, 2009

Personalized Game Recommendations

setting:

- mobile Internet portal, telecommunications provider in Germany
- catalog of games (nonpersonalized in the original version):
  - manually edited lists
  - direct links – teasers (text, image)
  - predefined categories (e.g., Action&Shooter, From 99 Cents)
  - postsales recommendations

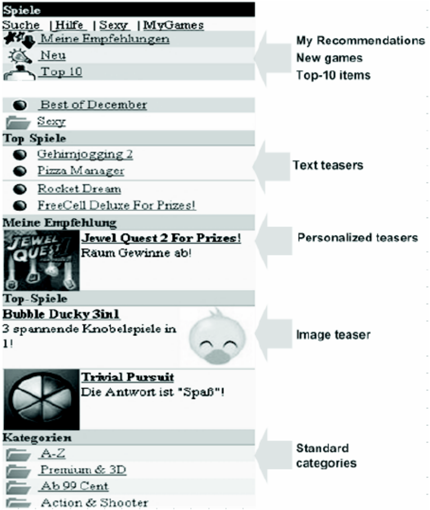


Figure 1: Catalog navigation and categories

Personalized Game Recommendations

personalization:

- new “My Recommendations” link
- choice of teasers
- order of games in categories
- choice of postsales recommendations

Algorithms

- nonpersonalized:
  - top rating
  - top selling
- personalized:
  - item-based collaborative filtering (CF)
  - Slope One (simple CF algorithm)
  - content-based method (using TF-IDF, item descriptions, cosine similarity)
  - hybrid algorithm (< 8 ratings: content, ≥ 8 ratings: CF)

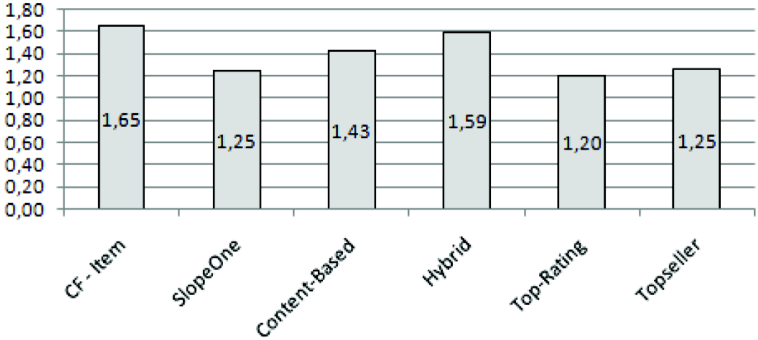


Figure 2: Average number of item detail views per "My Recommendations" visits

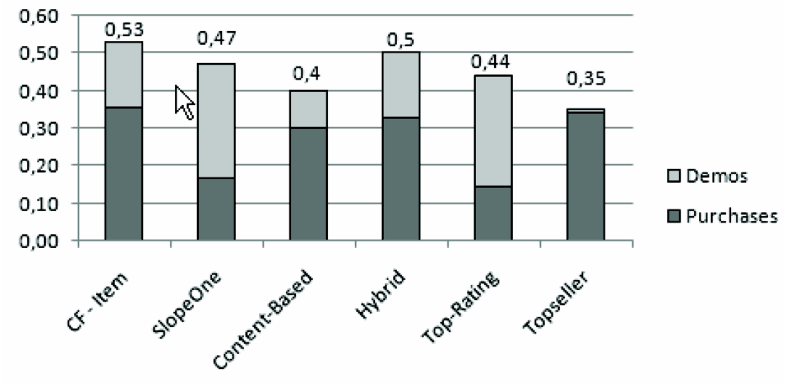


Figure 3: Average number of downloads per "My Recommendations" visit

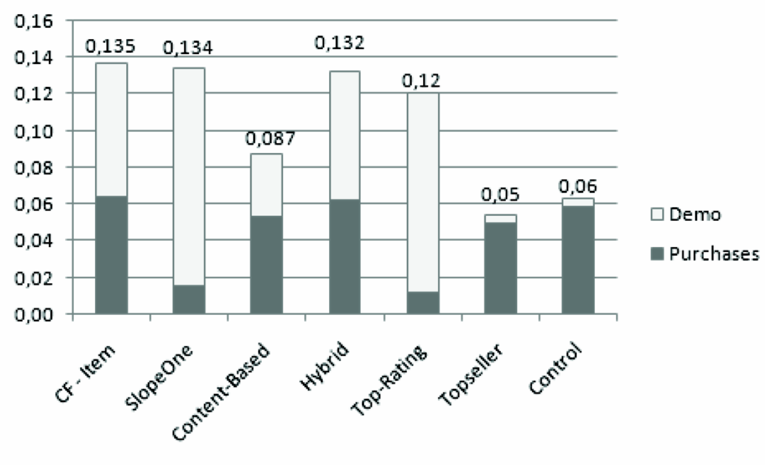


Figure 4: Average number of game purchases and demo downloads in post-sales situation.

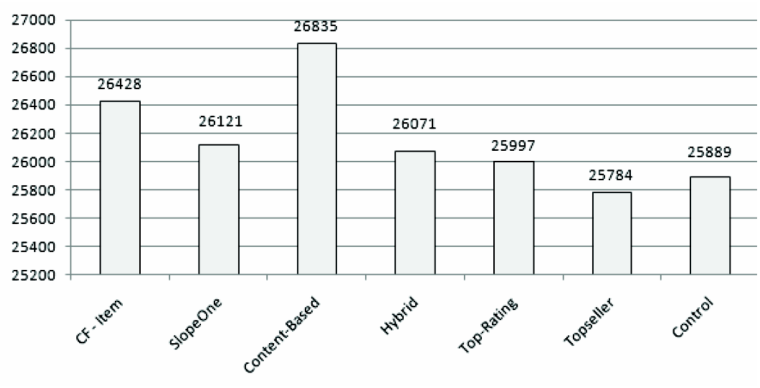


Figure 5: Total number of non-free game downloads.

App Recommendations

App recommendations (e.g., Google Play, Apple App store)

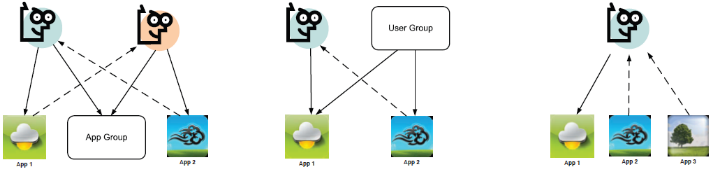
- What are the main differences? (e.g., compared to movies/book recommendations)
- Why the basic application of recommendation techniques may fail?

App Recommendations

App recommendation: a contest between satisfaction and temptation (2013)

- one-shot consumption (books, movies) vs continuous consumption (apps)
- impact on alternative (closely similar) apps, e.g., weather forecast
- when to recommend alternative apps?

App Recommendations: Failed Recommendations



(a) User-based Collaborative Filtering (b) Item-based Collaborative Filtering (c) Content-based Recommendation

Figure 2: Three scenarios of failed recommendation. The solid arrow means the user downloads the app while the dashed arrow indicates the particular app is recommended to the user.

Actual Value, Tempting Value

- actual value – “real satisfactory value of the app after it is used”
- tempting value – “estimated satisfactory value” (based on description, screenshots, ...)

computed based on historical data:
users with installed App i who view description of App j and decide to (not) install j

Actual Value minus Tempting Value

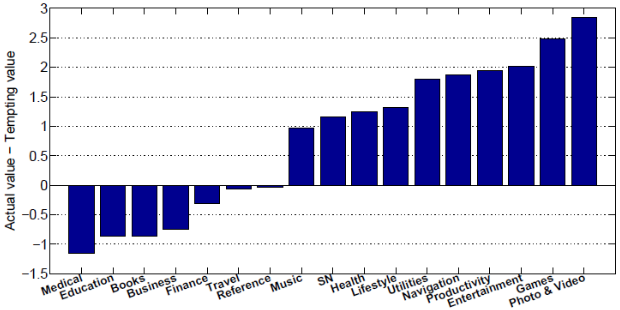


Figure 5: Actual-tempting difference with regarding to app category. Note that negative value means the app's actual value is smaller than its tempting value and vice versa.

Recommendations, Evaluation

- AT model, combination with content-based, collaborative filtering
- evaluation using historical data
- relative precision, recall

YouTube

- *The YouTube video recommendation system* (2010)
  - description of system design (e.g., related videos)
- *The impact of YouTube recommendation system on video views* (2010)
  - analysis of data from YouTube
- *Video suggestion and discovery for YouTube: taking random walks through the view graph* (2008)
  - algorithm description, based on view graph traversal
- *Deep neural networks for youtube recommendations* (2016)
  - use of context, predicting watch times

YouTube: Challenges

YouTube videos compared to movies (Netflix) or books (Amazon)

What are the specifics? Challenges?

YouTube: Challenges

YouTube videos compared to movies (Netflix) or books (Amazon)

What are the specifics? Challenges?

- poor meta-data
- many items, relatively short
- short life cycle
- short and noisy interactions

Input Data

- content data
  - raw video streams
  - metadata (title, description, ...)
- user activity data
  - explicit: rating, liking, subscribing, ...
  - implicit: watch, long watch

in all cases quite noisy

Related Videos

goal: for a video v find set of related videos

relatedness score for two videos v_i, v_j:

$$r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$$

- c_{ij} – co-visitation count (within given time period, e.g. 24 hours)
- $f(v_i, v_j)$ – normalization, "global popularity", e.g., $f(v_i, v_j) = c_i \cdot c_j$ (view counts)

top N selection, minimum score threshold

Generating Recommendation Candidates

- seed set S – watched, liked, added to playlist, ...
- candidate recommendations – related videos to seed set

$$C_1(S) = \cup_{v_i \in S} R_i$$

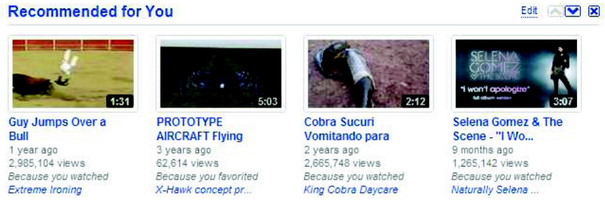
$$C_n(S) = \cup_{v_i \in C_{n-1}} R_i$$

Ranking

1. video quality
   - "global stats"
   - total views, ratings, commenting, sharing, ...
2. user specificity
   - properties of the seed video
   - user watch history
3. diversification
   - balance between relevancy and diversity
   - limit on number of videos from the same author, same seed video

User Interface

screenshot in the paper:



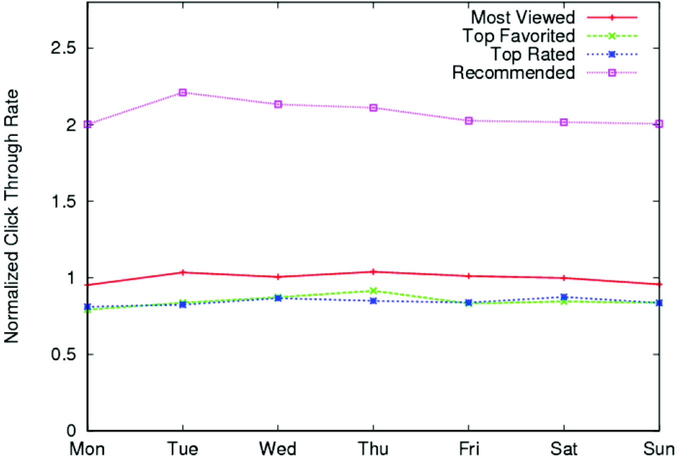
Note: explanations "Because you watched..." – not available in the current version

System Implementation

"batch-oriented pre-computation approach"

1. data collection
   - user data processed, stored in BigTable
2. recommendation generation
   - MapReduce implementation
3. recommendation serving
   - pre-generated results quickly served to user

Evaluation



Most Viewed
Top Favorited
Top Rated
Recommended
Normalized Click Through Rate
0
0.5
1
1.5
2
2.5
Mon
Tue
Wed
Thu
Fri
Sat
Sun

News Recommendations

recommending news stories

- What are the specifics?
- What approach would you use?

News Recommendations

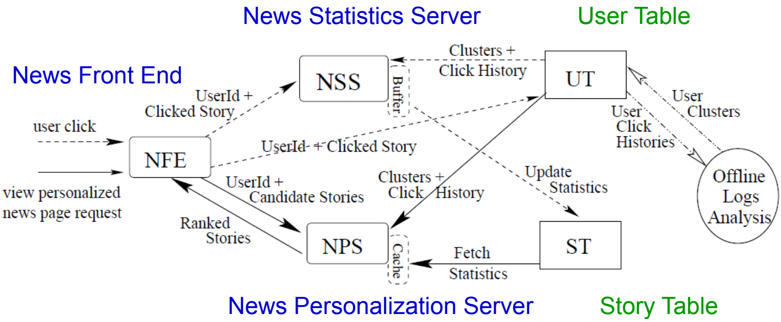
specific aspects:

- value of immediacy
- short time span of items (high churn)
- scale, timing requirements

Google News Personalization: Scalable Online Collaborative Filtering (2007)
basic idea: clustering

another example: *Scene: a scalable two-stage personalized news recommendation system*

Google News – System Setup



Google News: Algorithms

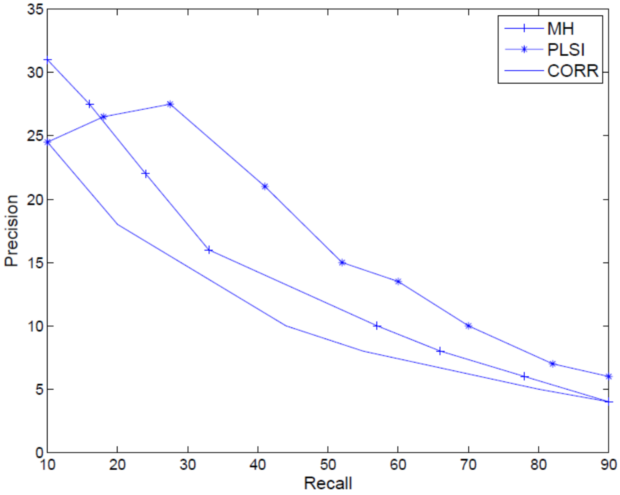
- collaborative filtering using MinHash clustering
- probabilistic latent semantic indexing
- covisitation counts

MapReduce implementations

Evaluation

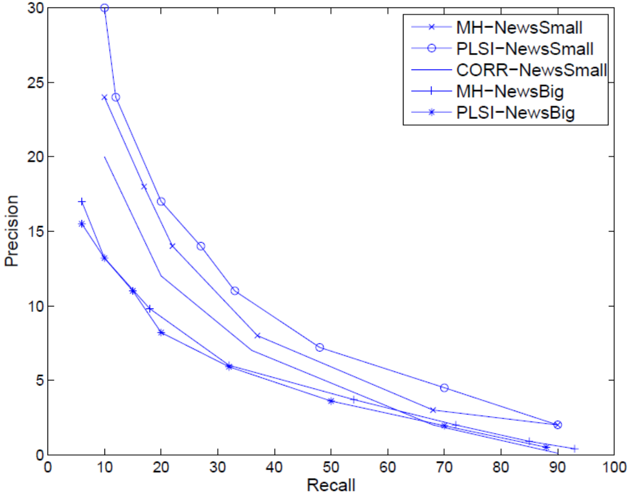
- datasets:
  - MovieLens $\sim$ 1000 users; 1700 movies; 54,000 ratings
  - NewsSmall $\sim$ 5000 users; 40,000 items; 370,000 clicks
  - NewsBig $\sim$ 500,000 users, 190,000 items; 10,000,000 clicks
- repeated randomized cross-validation (80% train set, 20% test set)
- metrics: precision, recall

Evaluation



MH
PLSI
CORR
Precision
Recall
0
5
10
15
20
25
30
35
10
20
30
40
50
60
70
80
90

Evaluation

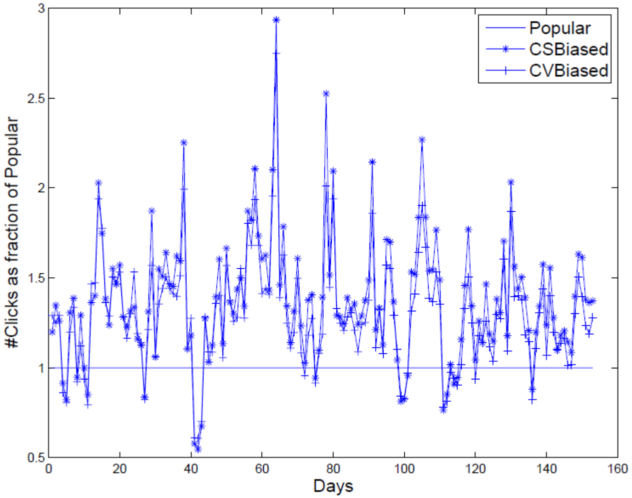


MH−NewsSmall
PLSI−NewsSmall
CORR−NewsSmall
MH−NewsBig
PLSI−NewsBig
Precision
Recall
0
5
10
15
20
25
30
0
10
20
30
40
50
60
70
80
90
100

Evaluation on Life Traffic

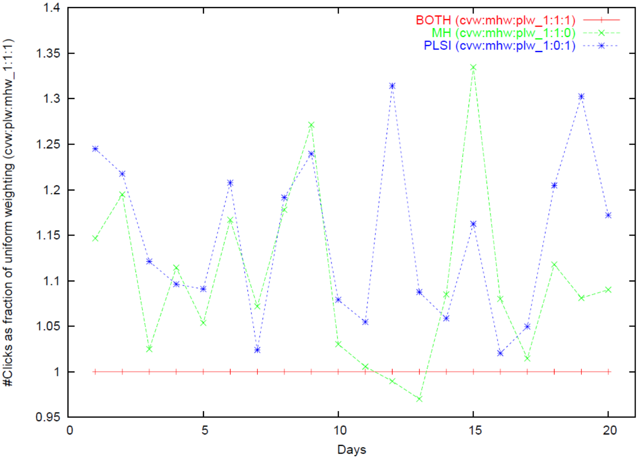
- large portion of life traffic on Google news
- comparison of two algorithms:
  - each algorithms generates sorted list of items
  - interlace these two lists
  - measure which algorithm gets more clicks
- baseline: “Popular” (age discounted click count)

Evaluation



Popular
CSBiased
CVBiased
#Clicks as fraction of Popular
Days
0.5
1
1.5
2
2.5
3
0
20
40
60
80
100
120
140
160

Evaluation



BOTH (cvw:mhw:plw_1:1:1)
MH (cvw:mhw:plw_1:1:0)
PLSI (cvw:mhw:plw_1:0:1)
#Clicks as fraction of uniform weighting (cvw:plw:mhw_1:1:1)
Days
1.4
1.35
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0
5
10
15
20

Music Recommendations

Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy (2011)

- large dataset (KDD cup 2011): 600 thusand items, 1 million users, 250 million ratings
- multi-typed items: tracks, albums, artists, genres
- taxonomy
- temporal dynamics

Ratings

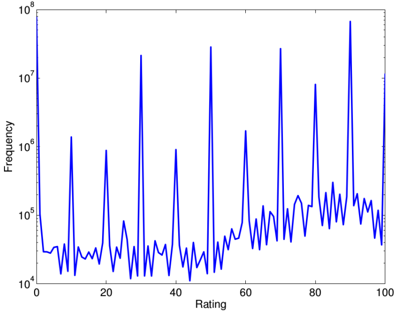


Figure 1: The distribution of ratings. The approximately discrete nature of the distribution is evident

Why the peaks?

Ratings

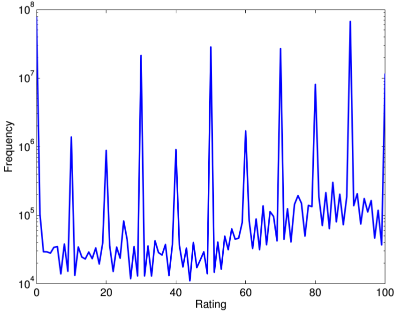


Figure 1: The distribution of ratings. The approximately discrete nature of the distribution is evident

Why the peaks?
Different widgets used for collecting ratings, including "5 stars" (translated into 0, 30, 50, 70, 90 values)

Item Mean Ratings

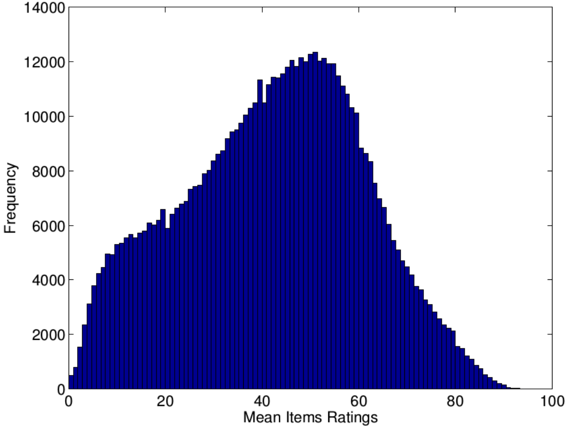


Figure 2: The distribution of item mean ratings

User Mean Ratings

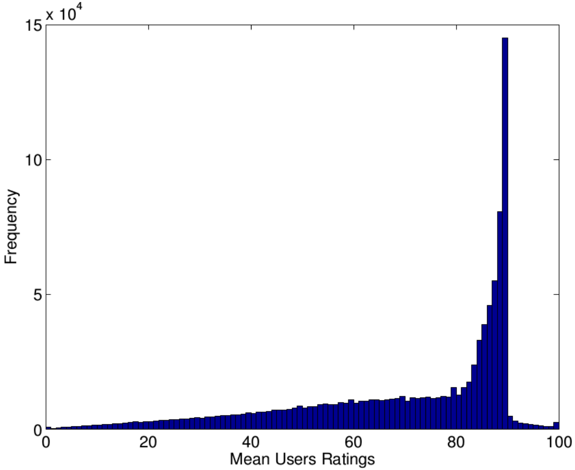


Figure 3: The distribution of user mean ratings

Item, User Mean Ratings

Item vs user means – why the discrepancy?

Item, User Mean Ratings

Item vs user means – why the discrepancy?

Users who rate less, rate higher.
Long term users are more critical.

Number of Ratings and Mean Rating

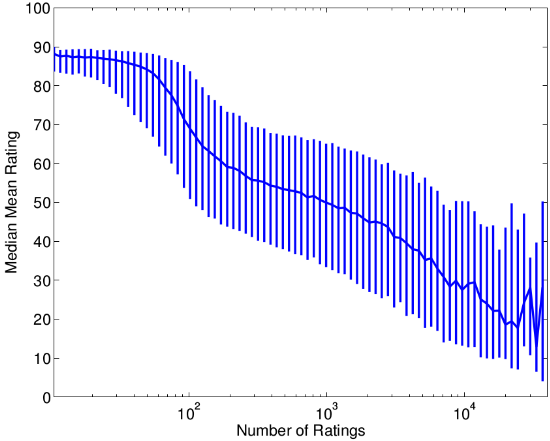


Figure 4: Median of user ratings as a function of the number of ratings issued by the user. The vertical lines represent inter-quartile range.

Types of Items

Also the type of rated items differ:

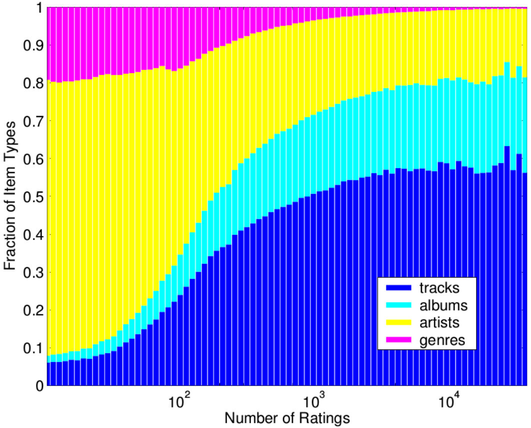


Figure 5: The fraction of ratings the four item types receive as a function of the number of ratings a user gives.

Lesson

Get to know your data before you start to use it.

Temporal Dynamics

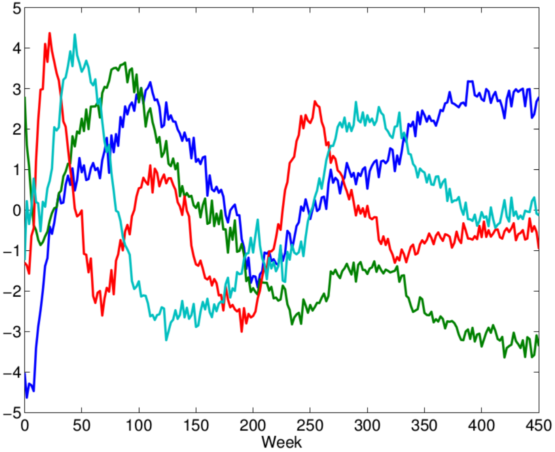


Figure 6: Items temporal basis functions $\{f_i(t)\}_{i=1}^4$ vs. time since an item's first rating measured in weeks

Evaluation

#	Model Name	RMSE
1	Mean Score	38.0617
2	Items and Users Bias	26.8561
3	Taxonomy Bias	26.2553
4	User Sessions Bias	25.3901
5	Items Temporal Dynamics Bias	25.2095
6	MF	22.9533
7	Taxonomy	22.7906
8	Final	22.5918

Table 2: Root Mean Squared Error (RMSE) of the evolving model. RMSE reduces while adding model components.

Book Recommendations for Children

What are the specific challenges compared to book recommendations for adults?

What type of data would you use? What techniques?

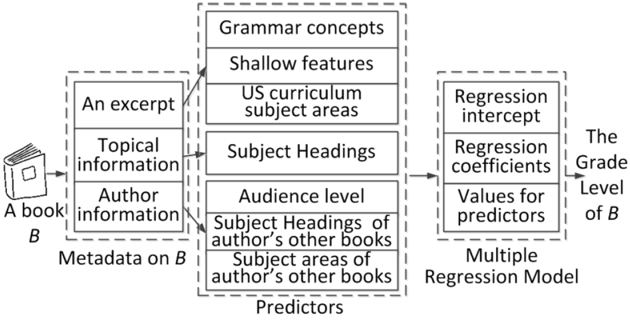
Book Recommendations for Children

What to read next?: making personalized book recommendations for K-12 users (2013)

books for children, specific aspects:

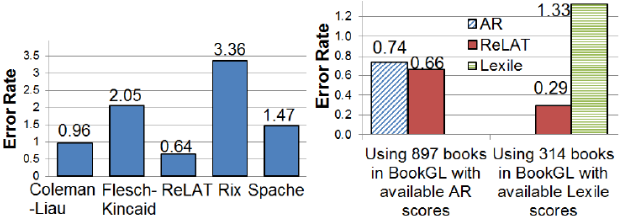
- focus on text difficulty
- less ratings available

Readability Analysis



Evaluation of Readability Analysis

dataset: $>$ 2000 books, "gold standard": publisher-provided grade level



(a) Readability Formulas (b) Analysis Tools

Figure 2: Performance evaluation of ReLAT

Book Recommender

1. identifying candidate books (based on readability)
2. content similarity measure
3. readership similarity measure
4. rank aggregation

Content Similarity

- brief descriptions from book-affiliated websites (not the content of book itself)
- cosine similarity, TF-IDF
- word-correlation factor – based on frequencies of co-occurrence and relative distance in Wikipedia documents

Content Similarity – Equations Preview

$$CSim(B,P) = \max_{P_B \in P} \frac{\sum_{i=1}^{n} VB_i \times VP_{B_i}}{\sqrt{\sum_{i=1}^{n} VB_i^2} \times \sqrt{\sum_{i=1}^{n} VP_{B_i}^2}} \qquad (3)$$

where B and P_B are represented as n-dimensional vectors $VB = <VB_1, ..., VB_n>$ and $VP_B = <VP_{B_1}, ..., VP_{B_n}>$, respectively, n is the number of distinct words in the descriptions of B and P_B, and VB_i (VP_{B_i}, respectively), which is the *weight* assigned to word B_i (P_{B_i}, respectively), is calculated as shown in the equations in Table 2.

Table 2: TF-IDF weighting scheme used in the enhanced cosine similarity measure in Equation 3

Condition	Weight Assignment
$B_i \in B$ and $P_{B_i} \in P_B$	$V_{B_i} = tf_{B_i,B} \times idf_{B_i}$ and $V_{P_{B_i}} = tf_{P_{B_i},P_B} \times idf_{P_{B_i}}$
$B_i \in B$ and $P_{B_i} \notin P_B$	$V_{B_i} = tf_{B_i,B} \times idf_{B_i}$ and $V_{P_{B_i}} = \frac{\sum_{c \in HS_{B_i}} tf_{c,P_B} \times idf_c}{\lvert HS_{B_i} \rvert}$
$B_i \notin B$ and $P_{B_i} \in P_B$	$V_{B_i} = \frac{\sum_{c \in HS_{P_{B_i}}} tf_{c,B} \times idf_c}{\lvert HS_{P_{B_i}} \rvert}$ and $V_{P_{B_i}} = tf_{P_{B_i},P_B} \times idf_{P_{B_i}}$

Readership Similarity

- collaborative filtering, item-item similarity
- co-occurrence of items bookmarked by users
- Lennon similarity measure

$$RSim(B,P) = \max_{P_B \in P} \left(1 - \frac{min(|S_B - S_\cap|, |S_{P_B} - S_\cap|)}{min(|S_B - S_\cap|, |S_{P_B} - S_\cap|) + |S_\cap|} \right)$$

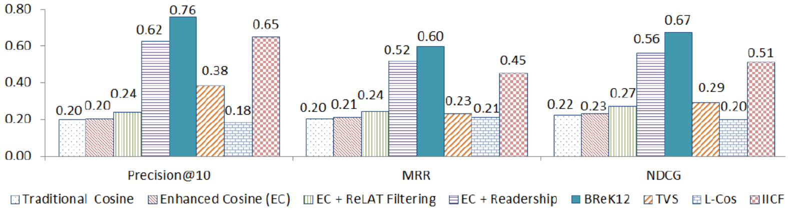
Rank Aggregation

- combine ranking from content and readership similarity
- Borda Count voting scheme
  - simple scheme to combine ranked list
  - points $\sim$ order in a list

Evaluation

- data: BiblioNasium (web page for kids), bookmarked books
- evaluation protocol: five-fold cross validation
- ranking metrics: Precision10, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (nDCG)

Evaluation



0.80
0.60
0.40
0.20
0.00
0.20 0.20 0.24 0.62 0.76 0.38 0.18 0.65
0.20 0.21 0.24 0.52 0.60 0.23 0.21 0.45
0.22 0.23 0.27 0.56 0.67 0.29 0.20 0.51
Precision@10
MRR
NDCG
Traditional Cosine
Enhanced Cosine (EC)
EC + ReLAT Filtering
EC + Readership
BReK12
TVS
L-Cos
IICF

Glimpse at Current Research

- Recommender Systems conference
- Google Scholar $\rightarrow$ metrics
- $\Rightarrow$ top cited publications from last 5 years
- lot of deep learning techniques... but also scepticism about them (2019 best paper)

ACM Conference on Recommender Systems

h5-index:50 h5-median:84

#6 Data Mining & Analysis
#11 Databases & Information Systems

Title / Author	Cited by	Year
Deep Neural Networks for YouTube Recommendations P Covington, J Adams, E Sargin Proceedings of the 10th ACM Conference on Recommender Systems, 191-198	1506	2016
Convolutional Matrix Factorization for Document Context-Aware Recommendation D Kim, C Park, J Oh, S Lee, H Yu Proceedings of the 10th ACM Conference on Recommender Systems, 233-240	483	2016
Field-aware Factorization Machines for CTR Prediction Y Juan, Y Zhuang, WS Chin, CJ Lin Proceedings of the 10th ACM Conference on Recommender Systems, 43-50	389	2016
Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks M Quadrana, A Karatzoglou, B Hidasi, P Cremonesi Proceedings of the Eleventh ACM Conference on Recommender Systems, 130-137	300	2017
Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations B Hidasi, M Quadrana, A Karatzoglou, D Tikk Proceedings of the 10th ACM Conference on Recommender Systems, 241-248	274	2016
Ask the GRU: Multi-task Learning for Deep Text Recommendations T Bansal, D Belanger, A McCallum Proceedings of the 10th ACM Conference on Recommender Systems, 107-114	238	2016
Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction S Seo, J Huang, H Yang, Y Liu Proceedings of the Eleventh ACM Conference on Recommender Systems, 297-305	236	2017
When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation D Jannach, M Ludewig Proceedings of the Eleventh ACM Conference on Recommender Systems, 306-310	228	2017
Are we really making much progress? A worrying analysis of recent neural recommendation approaches MF Dacrema, P Cremonesi, D Jannach Proceedings of the 13th ACM Conference on Recommender Systems, 101-109	224	2019

Summary

illustration of many aspects relevant in development of recommender systems:

- attacks
- context
- groups, shared accounts
- approaches to evaluation
- diversity
- differences between domains (books, movies, news...)