Evaluating Recommender Systems: *Ensuring Replicability of Evaluation*

Alejandro Bellogín (UAM, Spain)

Alan Said (TU-Delft, The Netherlands)

Tutorial at Hypertext 2014

European Science Foundation

UAM Universidad Autónoma de Madrid

TU Delft Delft University of Technology

ERCIM European Research Consortium for Informatics and Mathematics

About me

- 2014: Lecturer at UAM (Spain)
- 2013: Post-doctoral Marie Curie fellow at CWI (The Netherlands)
- 2007-2012: PhD student at UAM

- Topics (recommender systems): algorithms (probabilistic, hybrid, trust-based, social-based, graph-based), evaluation (methodologies, biases), user analysis (user clarity, coherence, performance)

Outline

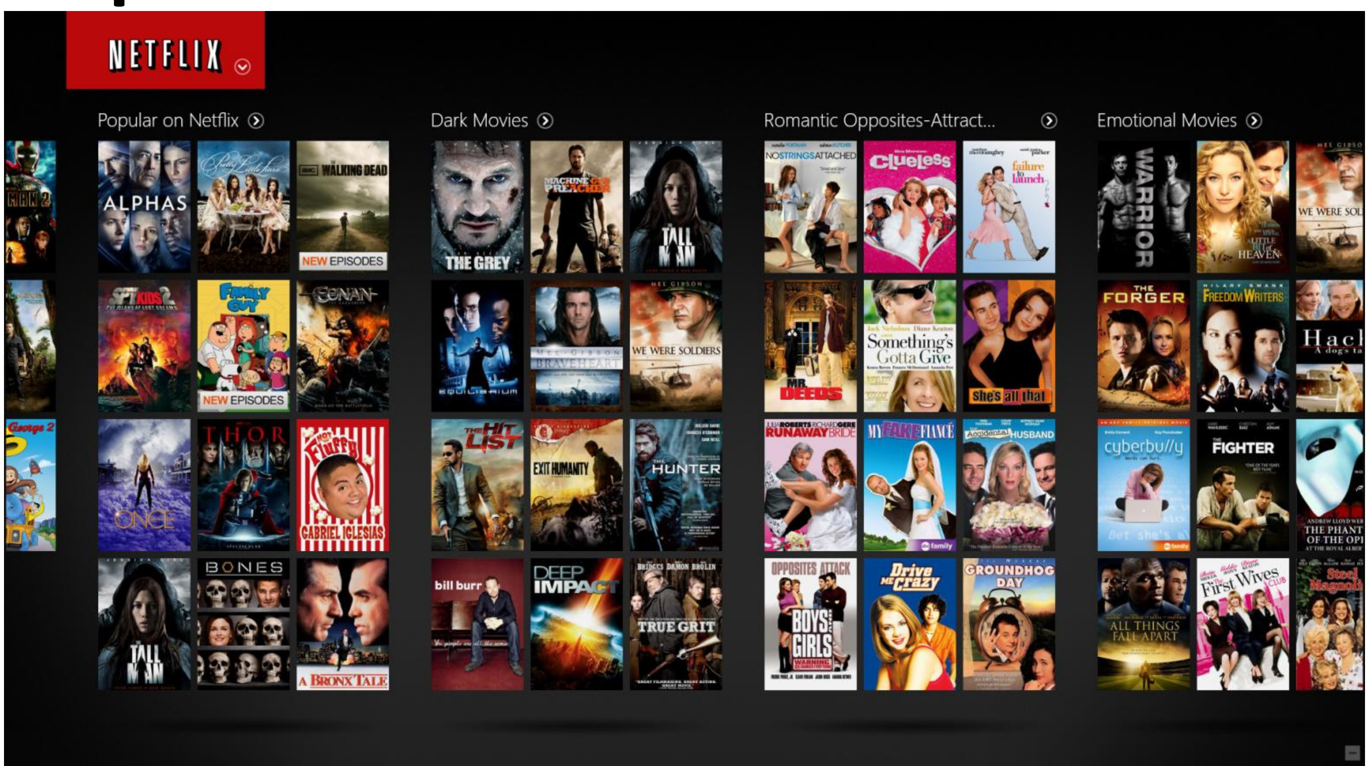
- Background and Motivation
- Evaluating Recommender Systems
- Reproducible Experimental Design
- Summary

Outline

- **Background and Motivation**
- Evaluating Recommender Systems
- Reproducible Experimental Design
- Summary

Background

- A <u>recommender system</u> aims to find and suggest items of **likely interest** based on the **users' preferences**



Background

- A <u>recommender system</u> aims to find and suggest items of **likely interest** based on the **users' preferences**

Background

- A recommender system aims to find and suggest items of **likely interest** based on the **users' preferences**
- Examples:
  - **Netflix**: tv shows and movies
  - **Amazon**: products
  - **LinkedIn**: jobs and colleagues
  - **Last.fm**: music artists and tracks

Background

- Typically, the interactions between user and system are recorded in the form of ratings
  - But also: clicks (implicit feedback)
- This is represented as a user-item matrix:

	i_1	…	i_k	…	i_m
u_1	★★★★★		★★★★★		★☆☆☆☆
⋮					
u_j	★★★☆☆		?		★★☆☆☆
⋮					
u_n	★★☆☆☆		★★★★☆		★★☆☆☆

Motivation

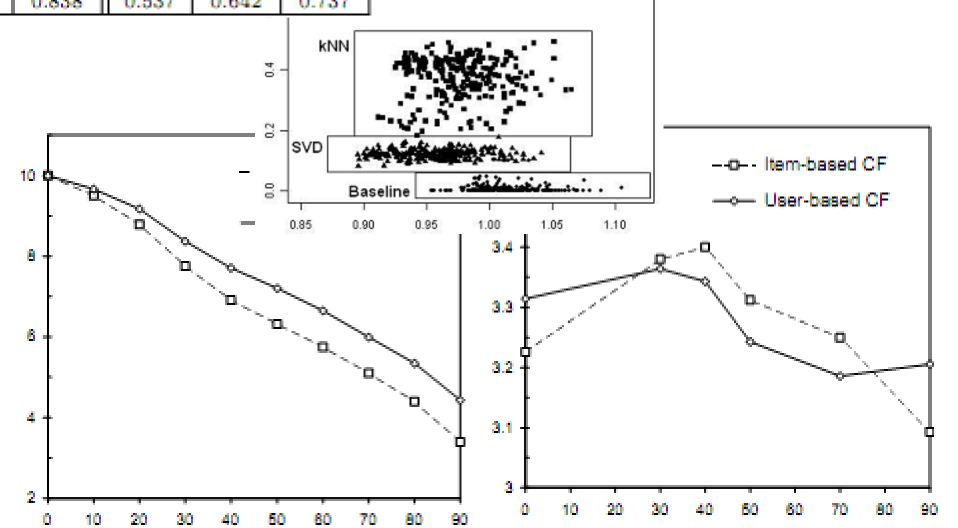
- Evaluation is an integral part of any experimental research area
- It allows us to compare methods…

Methods	MAP	Gain in UCF	MPR %	Gain in UCF %
UCF	0.0513	-	28.5	-
UCFWithCT	0.0856	0.0343	18.1	10.4
UCFWithCT +SchKW	0.1022	0.0509	15.4	13.1
UCFWithCT +SchKW+CT	0.1037	**0.0524**	15.0	**13.5**

	MovieLens			LastFM		
K	r@5	r@10	r@20	r@5	r@10	r@20
1	0.529	0.691	0.84	0.541	0.643	0.737
2	0.539	0.699	0.846	0.543	0.657	0.752
5	0.531	0.690	0.841	0.544	0.658	0.749
10	0.525	0.691	0.839	0.530	0.639	0.736
25	0.525	0.689	0.838	0.537	0.642	0.737

Model	50 factors	100 factors	200 factors
SVD	0.9046	0.9025	0.9009
Asymmetric-SVD	0.9037	0.9013	0.9000
SVD++	0.8952	0.8924	0.8911

	Lastfm		YahooMusic		BookCrossing	
SVDR	0.113	0.083	0.237	0.207	0.078	0.063
ASVDR	0.114	0.087	0.237	0.210	0.078	0.062
NMFR	0.114	0.090	0.218	0.189	0.073	0.054
ANMFR	0.110	0.089	0.211	0.190	0.071	0.056
SVDN	0.002	0.003	0.001	0.001	0.005	0.003
ASVDN	0.003	0.002	0.007	0.005	0.005	0.003
HSVD	**0.180**	**0.158**	**0.258**	**0.227**	0.075	0.065



Motivation

- Evaluation is an integral part of any experimental research area
- It allows us to compare methods…
- … and decide a winner (in competitions)

ACM RecSys Challenge

CLEF INITIATIVE

plista

eswc14

Motivation

- A proper evaluation culture allows advance the field

Improvements That Don't Add Up:
Ad-Hoc Retrieval Results Since 1998

Timothy G. Armstrong, Alistair Moffat, William Webber, Justin Zobel
Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
{tgar,alistair,wew,jz}@csse.unimelb.edu.au

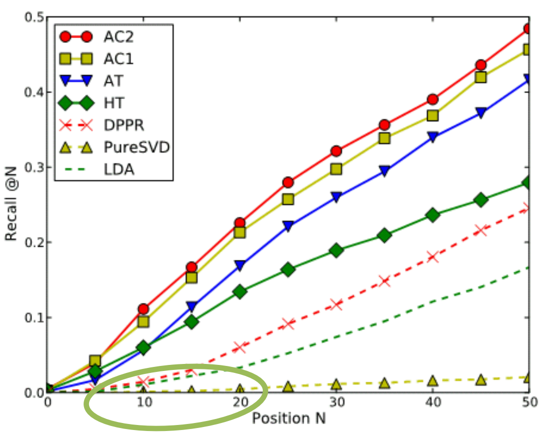
- … or at least, identify when there is a problem!

Motivation

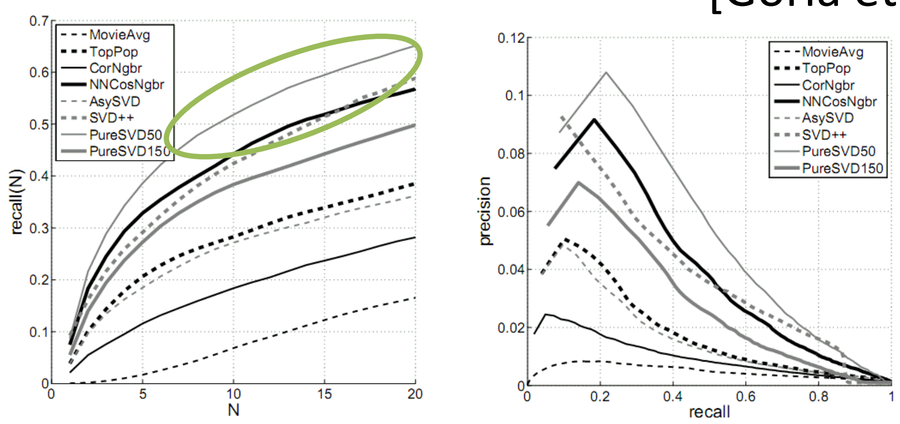
- In recommendation, we find inconsistent evaluation results, for the "same"
  - Dataset
  - Algorithm
  - Evaluation metric

	Algorithm				
Metric	k-Item	k-User	PureSVD	*Pop-item*	IMM
P@5	0.00135	0.006	0.067	0.227	0.267
NDCG@5	0.0036	0.0091	0.0566	0.216	0.245
MAP	0.013	0.041	0.061	0.119	0.156

Movielens 100k
[Gorla et al, 2013]



Movielens 1M
[Yin et al, 2012]



(a) recall (b) precision vs recall

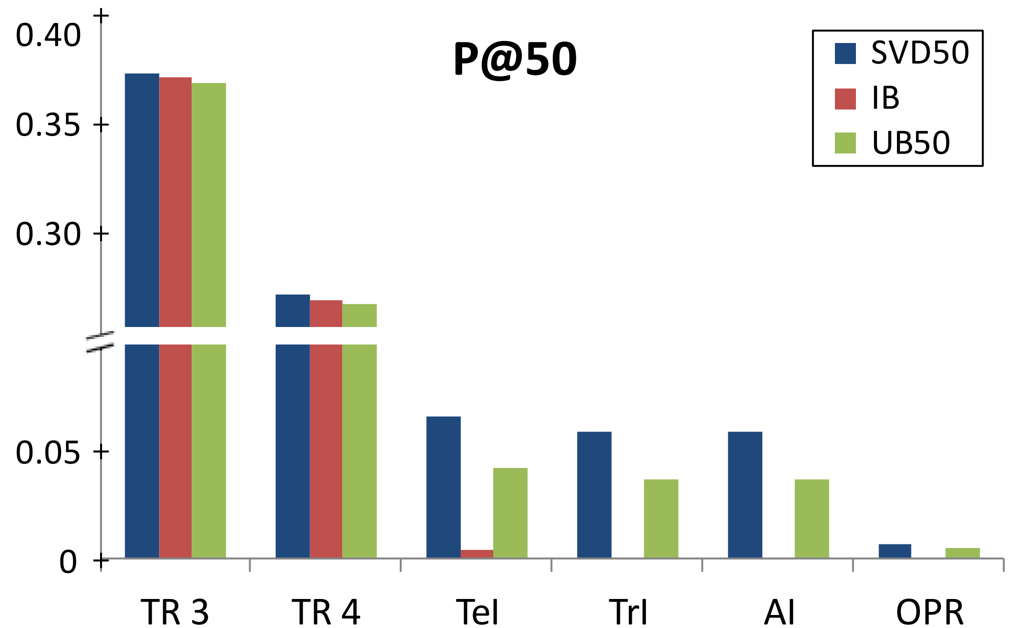
Movielens 1M
[Cremonesi et al, 2010]

	Baseline(Test)
MAP	0.447
MRR	0.889
NDCG@1(	0.720
NDCG@5	0.570
NDCG@3	0.447

Movielens 100k, SVD
[Jambor & Wang, 2010]

Motivation

- In recommendation, we find inconsistent evaluation results, for the "same"
  - Dataset
  - Algorithm
  - Evaluation metric



[Bellogín et al, 2011]

Motivation

- In recommendation, we find inconsistent evaluation results, for the "same"
  - Dataset
  - Algorithm
  - Evaluation metric

We need to understand why this happens

In this tutorial

- We will present the basics of evaluation
  - Accuracy metrics: error-based, ranking-based
  - Also coverage, diversity, and novelty
- We will focus on reproducibility
  - Define the context
  - Present typical problems
  - Propose some guidelines

NOT in this tutorial

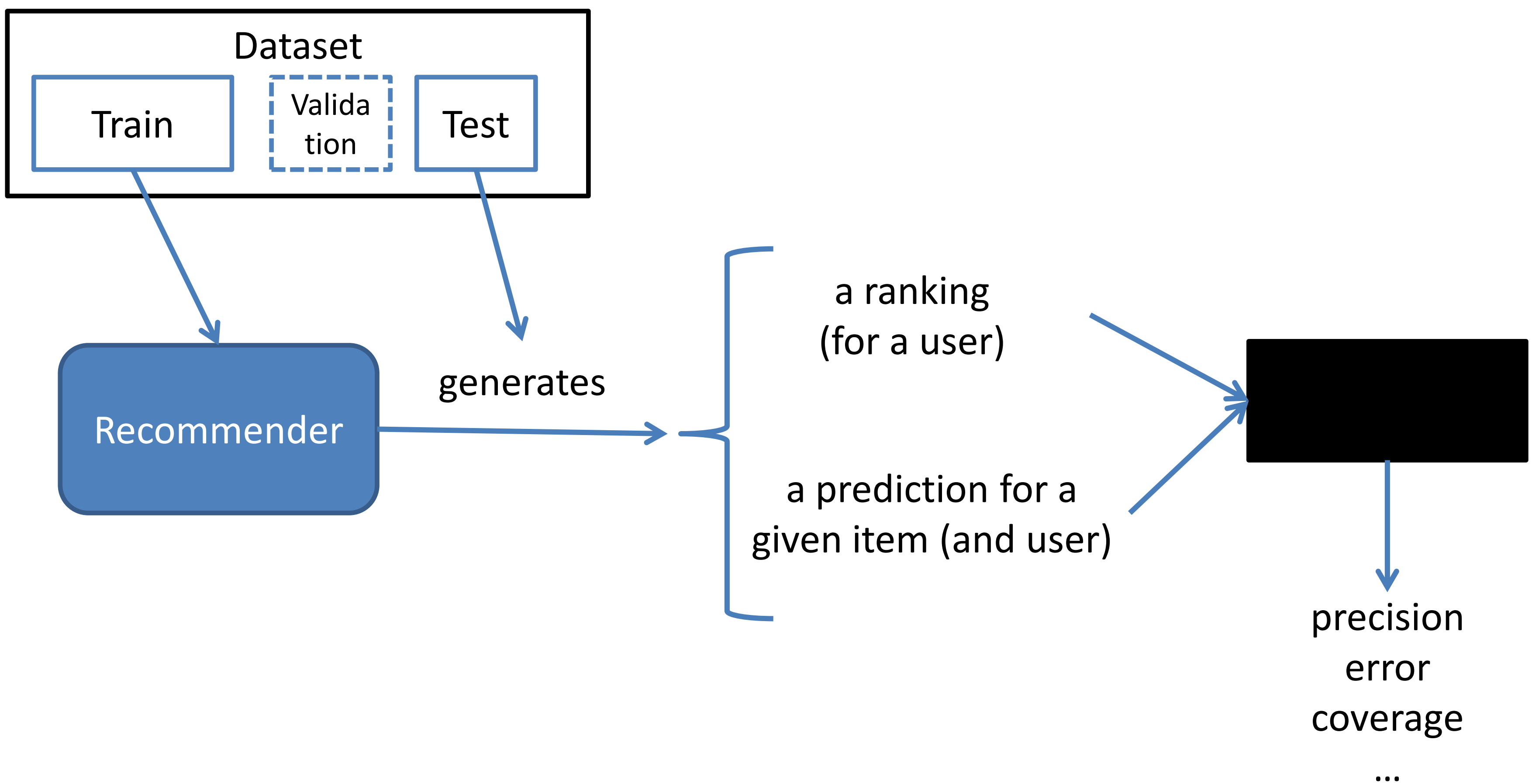
- In-depth analysis of evaluation metrics
  – See chapter 9 on handbook [Shani & Gunawardana, 2011]
- Novel evaluation dimensions
  – See tutorial at WSDM ’14 and SIGIR ‘13 on diversity and novelty
- User evaluation
  – See tutorial at RecSys 2012 by B. Knijnenburg
- Comparison of evaluation results in research
  – See RepSys workshop at RecSys 2013

Outline

- Background and Motivation
- **Evaluating Recommender Systems**
- Reproducible Experimental Design
- Summary

Recommender Systems Evaluation

- Typically: as a black box



Evaluation metrics

- Accuracy metrics: typically reported in the literature (and usually, only these)
- Non accuracy metrics: related to other evaluation dimensions
  - Coverage
  - Diversity
  - Novelty
  - …

Accuracy metrics

- Error-based
  - RMSE, MAE
- Ranking-based
  - Precision, recall, MAP, nDCG
- Other accuracy metrics
  - AUC, NDPM, correlation

Error-based metrics

- Assumption: more accurate predictions, better
- Pre-assumption: we are predicting ratings
- Conclusion: not useful for implicit feedback

$$\mathrm{MAE} = \frac{1}{|\mathrm{Te}|} \sum_{(u,i) \in \mathrm{Te}} |\tilde{r}(u,i) - r(u,i)|$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathrm{Te}|} \sum_{(u,i) \in \mathrm{Te}} \left(\tilde{r}(u,i) - r(u,i)\right)^2}$$

MAE = Mean Absolute Error

RMSE = Root Mean Squared Error

Error-based metrics

- Variations:
  - Normalize RMSE or MAE by the range of the ratings (divide by $r_{max} - r_{min}$)
  - Average RMSE or MAE to compensate for unbalance distributions of items or users

$$\mathrm{uMAE} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathrm{Te}_u|} \sum_{i \in \mathrm{Te}_u} |\tilde{r}(u,i) - r(u,i)|$$

uMAE = user-averaged Mean Absolute Error

Error-based metrics

- Limitations:
  - Depend on the ratings range (unless normalized)
  - Depend on the recommender output's range
  - Not valid for recommenders that produce a score (not a rating): probability, similarity, etc.
  - Do not distinguish errors on top items and the rest

User-item pairs	Real	Rec1	Rec2	Rec3
(u_1, i_1)	5	4	8	5
(u_1, i_2)	3	2	4	1
(u_1, i_3)	1	1	2	1
(u_2, i_1)	3	2	4	2
MAE/RMSE		0.75/0.87	1.5/1.73	0.75/1.12

Ranking-based metrics

- Assumption: users only care about errors in the item rank order provided by the system
- They are usually computed up to a ranking position or cutoff *k*

$$P@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathrm{Rel}_u@k|}{k}$$

$$R@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathrm{Rel}_u@k|}{|\mathrm{Rel}_u|}$$

P = Precision (Precision at k)

R = Recall (Recall at k)

Ranking-based metrics

- Assumption: users only care about errors in the item rank order provided by the system
- They are usually computed up to a ranking position or cutoff *k*

$$\mathrm{MAP} = \frac{1}{|\mathcal{U}|} \sum_{u} \frac{1}{|\mathrm{Rel}_u|} \sum_{i \in \mathrm{Rel}_u} \mathrm{P@rank}(u, i)$$

MAP = Mean Average Precision

Ranking-based metrics

- Assumption: users only care about errors in the item rank order provided by the system
- They are usually computed up to a ranking position or cutoff *k*

$$\mathrm{nDCG} = \frac{1}{|\mathcal{U}|}\sum_{u}\frac{1}{\mathrm{IDCG}_u^{p_u}}\sum_{p=1}^{p_u}\mathrm{f_{dis}}(\mathrm{rel}(u, i_p), p)$$

$$f_{\mathrm{dis}}(x, y) = (2^x - 1)/\log(1 + y)$$

$$f_{\mathrm{dis}}(x, y) = x/\log y \ \text{ if } y > 1$$

nDCG = normalized Discounted Cumulative Gain

Ranking-based metrics

- There are many others:
  - Rank score (half-life utility): like nDCG but with a different discount function
  - Mean percentage ranking
  - Mean reciprocal rank: only takes into account where the first relevant result occurs
  - Average rank of correct recommendation
  - Average reciprocal hit-rank

Ranking-based metrics

- Limitations:
  - Performance is, probably, underestimated (since real preferences are scarce and unknown preferences are assumed to be not relevant)
  - Implementation-dependent when there are ties in the scores that generate the ranking
  - Different results depending on the cutoff…
  - … And no agreement about which cutoff is best: 1, 3, 5, 10, 50, …?

Other accuracy metrics

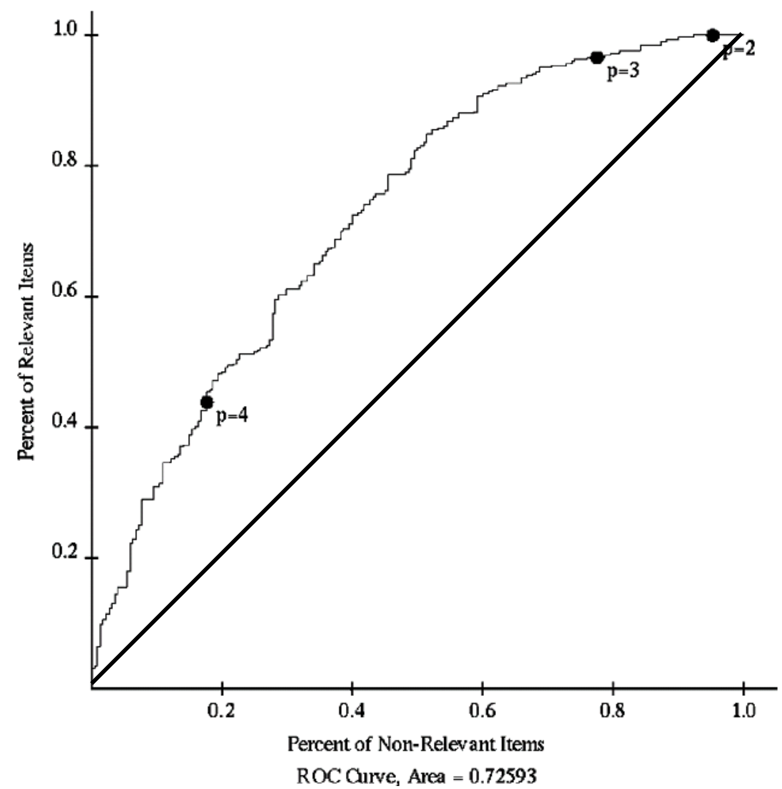
- AUC: area under the (ROC) curve

  At each rank position:

  – If item relevant: curve up

  – Otherwise: curve right

- Random recommender
  – straight diagonal line
  – AUC = 0.5
- Variations
  – Global ROC
  – Customer ROC: same number of items to each user



[Herlocker et al, 2004]

Other accuracy metrics

- NDPM: normalized distance-based performance measure
- It compares two weakly ordered rankings

$$\mathrm{NDPM} = \frac{1}{|\mathcal{U}|}\sum_{u}\frac{2C_u^{\mathrm{con}} + C_u^{\mathrm{tie}}}{2C_u}$$

  - *con*: number of discordant item pairs
  - *tie*: number of compatible item pairs
  - normalized by the number of pairs not tied in the real ranking

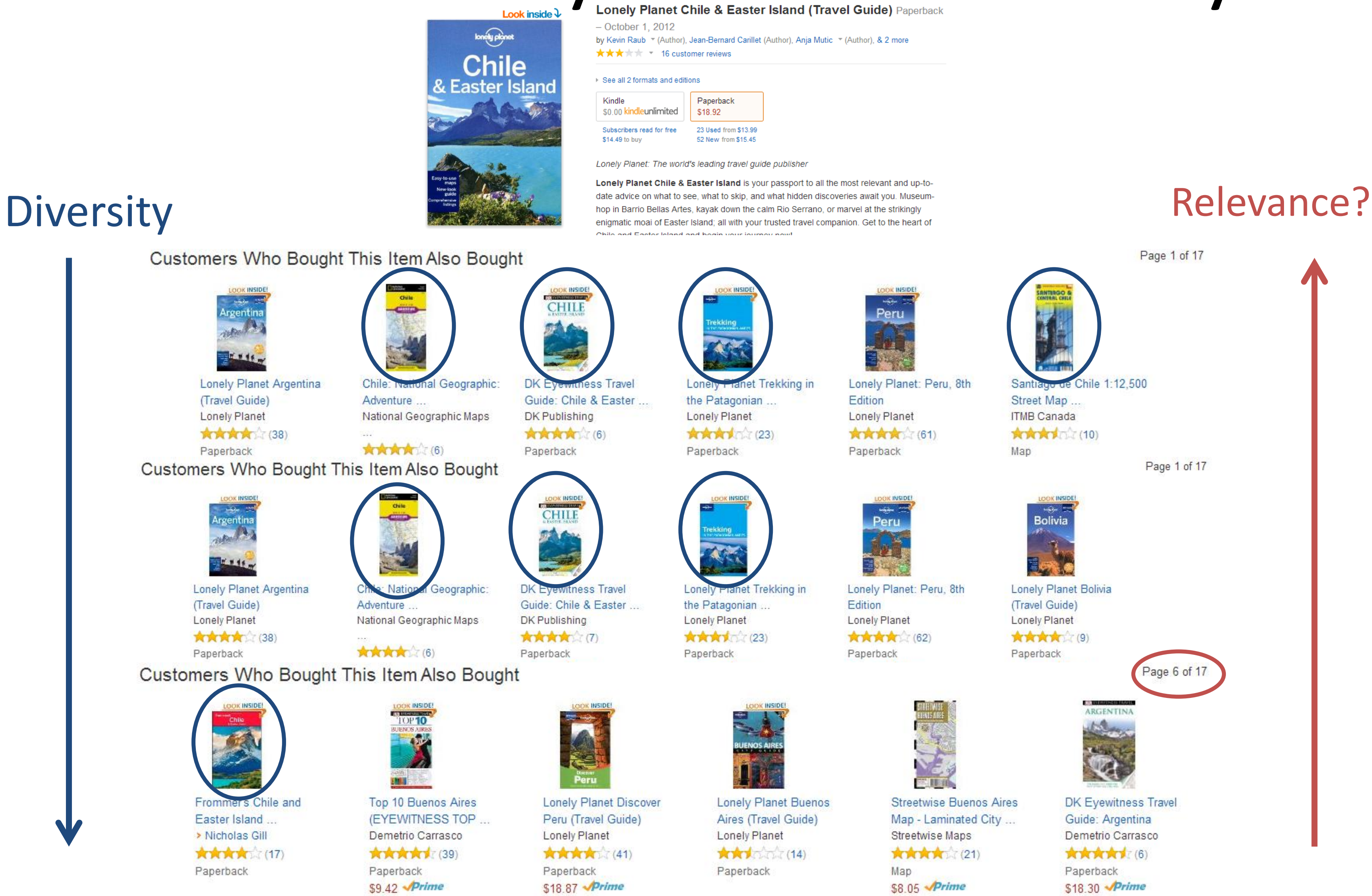
Other accuracy metrics

- Rank correlation coefficients between predicted and ideal ranking:
  - Spearman
  - Kendall
- NDPM is similar but provides a more accurate interpretation of the effect of tied user ranks
- Limitation: interchange weakness
  - Interchanges at the top of the ranking have the same weight as in the bottom

Non accuracy metrics: Coverage

- User coverage
- Catalog/item coverage
  - Simple ratio [Ge et al, 2010]
  - Based on Gini’s index [Shani & Gunawardana, 2011]
  - Based on Shannon’s entropy [Shani & Gunawardana, 2011]
- “Practical accuracy of a system”: combination of coverage and accuracy
  - A system with low coverage is less useful

Non accuracy metrics: Diversity



Non accuracy metrics: Diversity

- How to measure diversity? Several proposals:
  - Using a distance/dissimilarity function [Zhang & Hurley, 2008]
  - Measuring the intra-list similarity [Ziegler et al, 2005]
  - Using statistics to analyze the item distribution (concentration curve) [Zhang & Hurley, 2009]
  - Based on entropy [Bellogín et al, 2010]
  - Based on Gini's index [Fleder & Hosanagar, 2009]

- Formal framework in [Vargas & Castells, 2011]

Non accuracy metrics: Novelty

- Novel recommendations: items the user did not know prior to the recommendation
- Directly measured in online experiments
- Not clear how to do it in offline experiments:
  - Using a taxonomy (items about novel topics) [Weng et al, 2007]
  - New items over time [Lathia et al, 2010]
  - Based on entropy, self-information and Kullback-Leibler divergence [Bellogín et al, 2010; Zhou et al, 2010; Filippone & Sanguinetti, 2010]

Recommender Systems Evaluation: Summary

- Usually, evaluation seen as a black box
- Mostly focused on metrics
  - Especially, on accuracy metrics
- But there are other dimensions worth of interest
- No metric is perfect
- We should agree on standard implementations, parameters, instantiations, …
  - Example: trec_eval in IR

Outline

- Background and Motivation
- Evaluating Recommender Systems
- **Reproducible Experimental Design**
- Summary

Reproducible Experimental Design

- We need to distinguish
  - Replicability
  - Reproducibility

- Different aspects:
  - Algorithmic
  - Published results
  - Experimental design

Definition: Replicability

To *copy* something

- The results
- The data
- The approach

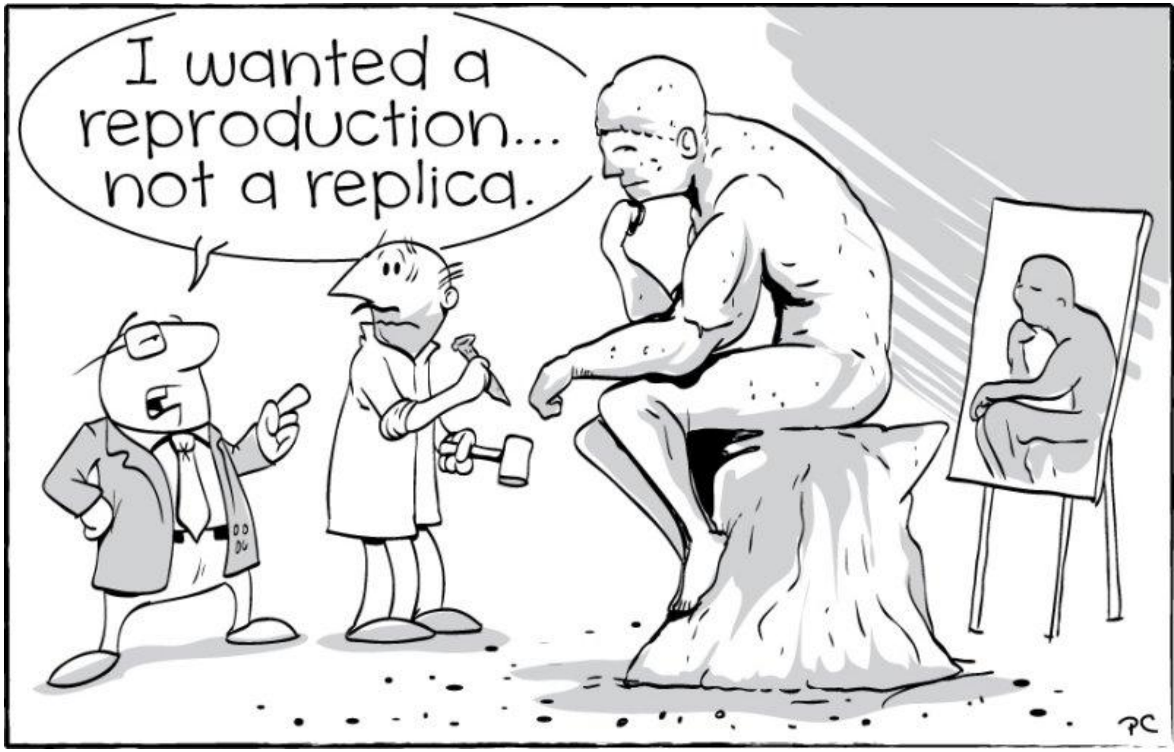
Being able to evaluate in <u>the same</u> setting and obtain <u>the same</u> results

Definition: Reproducibility

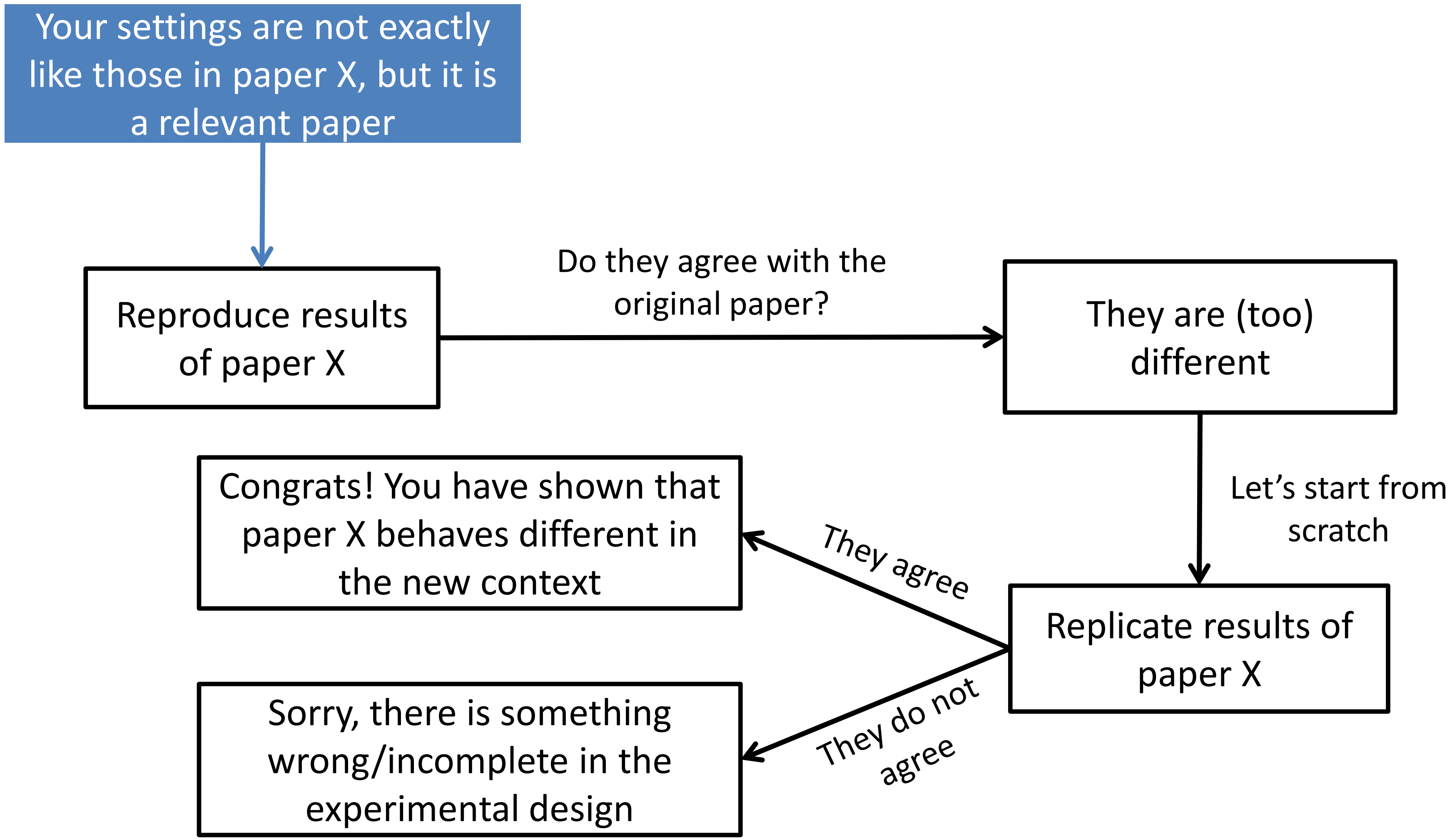
To recreate something

- The (complete) set of experiments
- The (complete) set of results
- The (complete) experimental setup

To (re) launch it in production with the same results



Comparing against the state-of-the-art



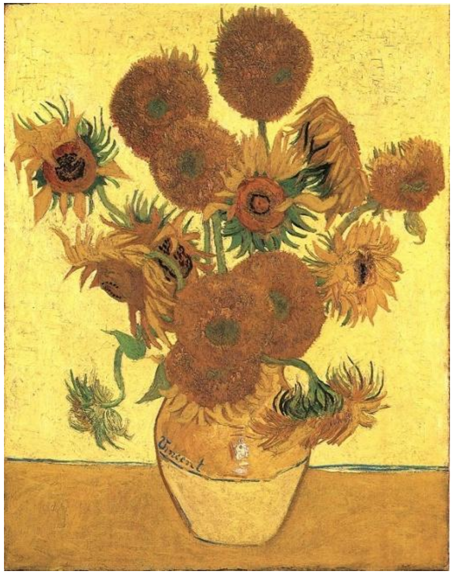
What do they mean?

Can we recreate them?

Replicability

- Why do we need to replicate?




Replicability

- Making sure your results were not a fluke
- Can others repeat/validate your experiments, results, conclusions?

REPRODUCIBILITY INITIATIVE

http://validation.scienceexchange.com

Reproducibility

Why do we need to reproduce?

Because these two are not the same

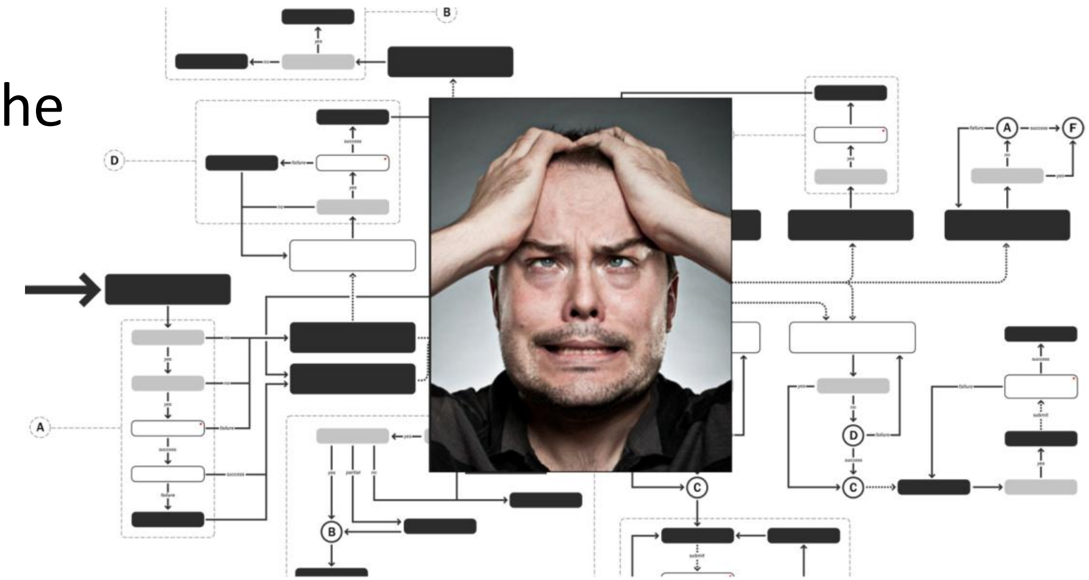
Reproducibility

- In order to ensure that our experiments, settings, and results are:
  - Valid
  - Generalizable
  - Of use for others
  - etc.

we must make sure that others can reproduce our experiments in their setting

Making reproducibility easier

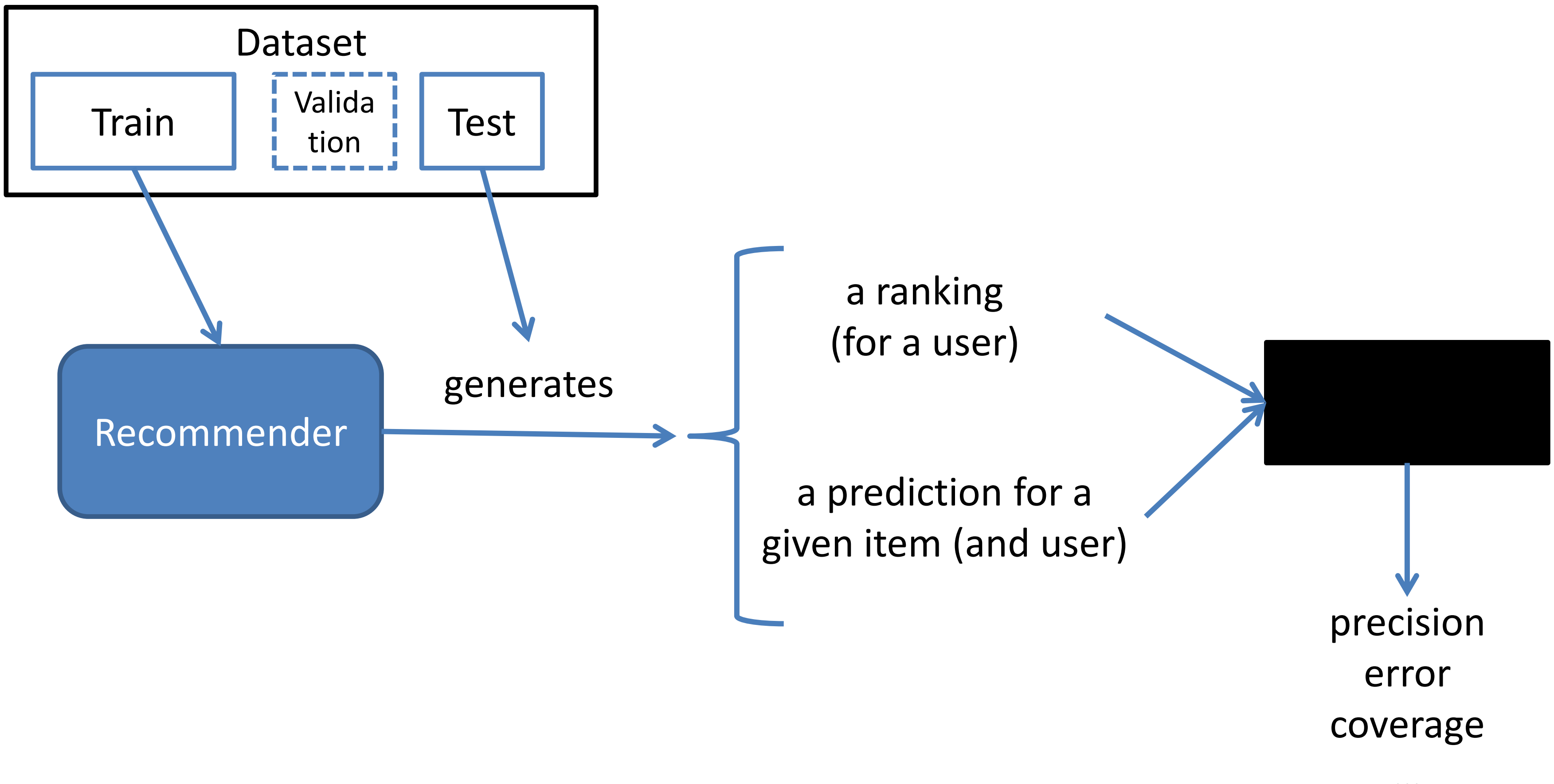
- Description, description, description
- No magic numbers
- Specify the value for all the parameters
- Motivate!
- Keep a detailed **protocol**
- Describe process **clearly**
- Use **standards**
- Publish code (nobody expects you to be an awesome developer, you’re a researcher)



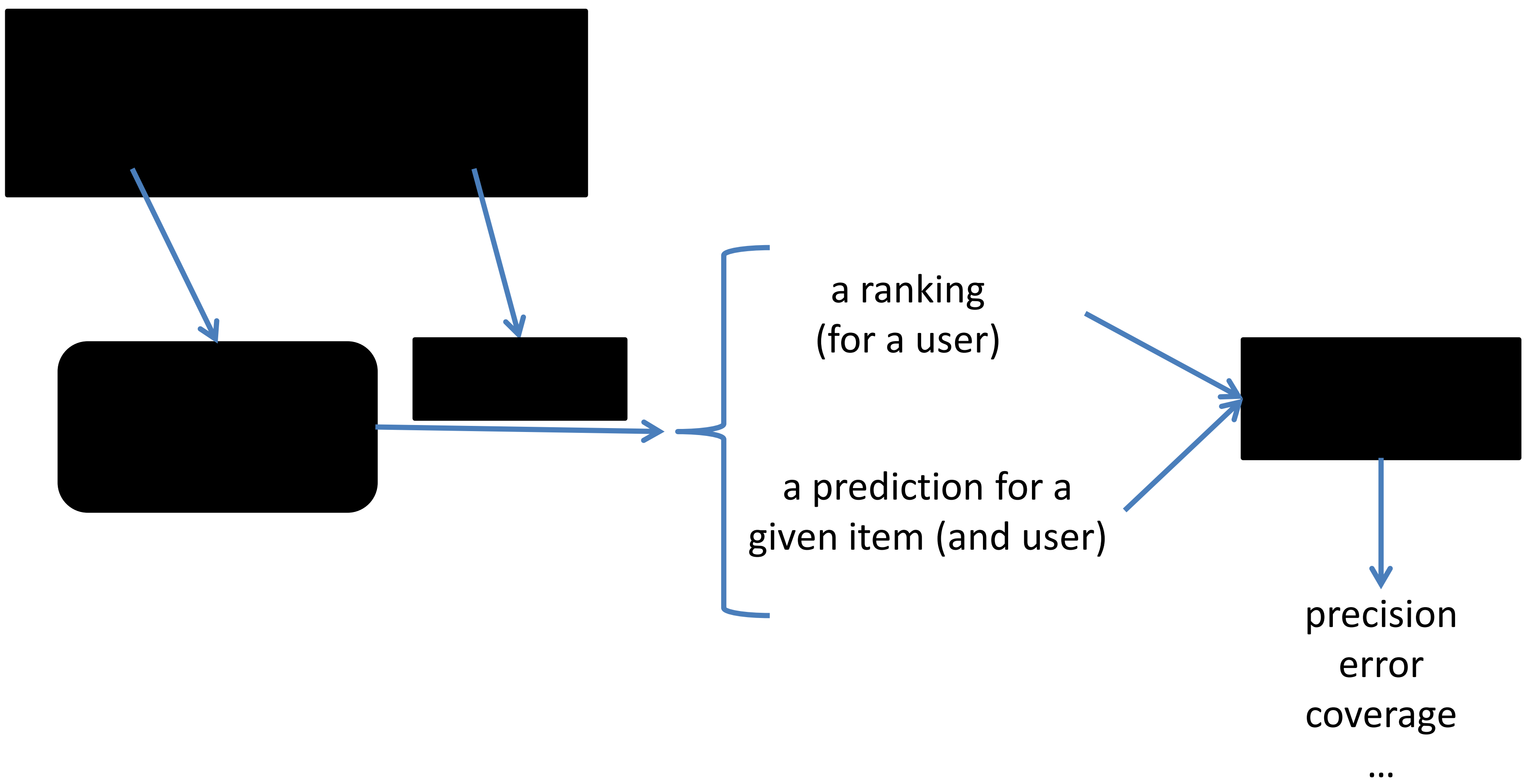
Replicability, reproducibility, and progress

- Can there be actual progress if no valid comparison can be done?
- What is the point of comparing two approaches if the comparison is flawed?
- How do replicability and reproducibility facilitate actual progress in the field?

Evaluation as a black box



Evaluation as black boxes



An experiment

- We used internal evaluation methods in Mahout (AM), LensKit (LK), and MyMediaLite (MML)

(a) nDCG for AM and LK

Alg.	F.W.	nDCG
IBCos	AM	0.000414780
	LK	0.942192050
IBPea	AM	0.005169231
	LK	0.924546132
SVD50	AM	0.105427298
	LK	0.943464094
UBCos50	AM	0.169295451
	LK	0.948413562
UBPea50	AM	0.169295451
	LK	0.948413562

(b) RMSE values for LK and MML.

Alg.	F.W.	RMSE
IBCos	LK	1.01390931
	MML	0.92476162
IBPea	LK	1.05018614
	MML	0.92933246
SVD50	LK	1.01209290
	MML	0.93074012
UBCos50	LK	1.02545490
	MML	0.95358984
UBPea50	LK	1.02545490
	MML	0.93419026

[Said & Bellogín, 2014]

Evaluation as black boxes

PROS

- Easy
- Don't reinvent the wheel

CONS

- Cherry-picking
  - Good results
  - Wrong (not optimal) setting
- Not comparable
  - Add/remove bias from data
- Difficult to disclose all the details
  - Is step N important?
  - What did I do after step M?

Some problems with "black boxes"

- What do you do when a recommender cannot predict a score?
  - This has an impact on coverage

Alg.	F.W.	Time (sec.)	RMSE	nDCG@10 RPN	nDCG@10 UT	User cov.(%) RPN	User cov.(%) UT	Cat. cov.(%) RPN	Cat. cov.(%) UT
IBCos	AM	238	1.041	0.003	0.501	98.16	100	99.71	99.67
	LK	44	0.953	0.199	0.618	98.16	100	99.88	99.67
	MML	75	NA	0.488	0.521	98.16	100	100	99.67
IBPea	AM	237	1.073	0.022	0.527	97.88	100	86.66	99.31
	LK	31	1.093	0.033	0.527	97.86	100	86.68	99.31
	MML	1,346	0.857	0.882	0.654	98.16	100	2.87	99.83
SVD50	AM	132	0.950	0.286	0.657	98.12	100	99.88	99.67
	LK	7	1.004	0.280	0.621	98.16	100	100	99.67
	MML	1,324	0.848	0.882	0.648	98.18	100	2.87	99.83
UBCos50	AM	5	1.178	0.378	0.387	35.66	98.25	6.53	27.80
	LK	25	1.026	0.223	0.657	98.16	100	99.88	99.67
	MML	38	NA	0.519	0.551	98.16	100	100	99.67
UBPea50	AM	6	1.126	0.375	0.486	48.50	100	10.92	39.08
	LK	25	1.026	0.223	0.657	98.16	100	99.88	99.67
	MML	1,261	0.847	0.883	0.652	98.18	100	2.87	99.83

Some problems with "black boxes"

- What do you do when a recommender cannot predict a score?
  - This has an impact on coverage
  - It can also affect error-based metrics

User-item pairs	Real	Rec1	Rec2	Rec3
(u_1, i_1)	5	4	NaN	4
(u_1, i_2)	3	2	4	NaN
(u_1, i_3)	1	1	NaN	1
(u_2, i_1)	3	2	4	NaN
MAE/RMSE, ignoring NaNs		0.75/0.87	2.00/2.00	0.50/0.70
MAE/RMSE, NaNs as 0		0.75/0.87	2.00/2.65	1.75/2.18
MAE/RMSE, NaNs as 3		0.75/0.87	1.50/1.58	0.25/0.50

Some problems with "black boxes"

- NDCG has at least two discounting functions
  - Which one are you using: linear or exponential decay?

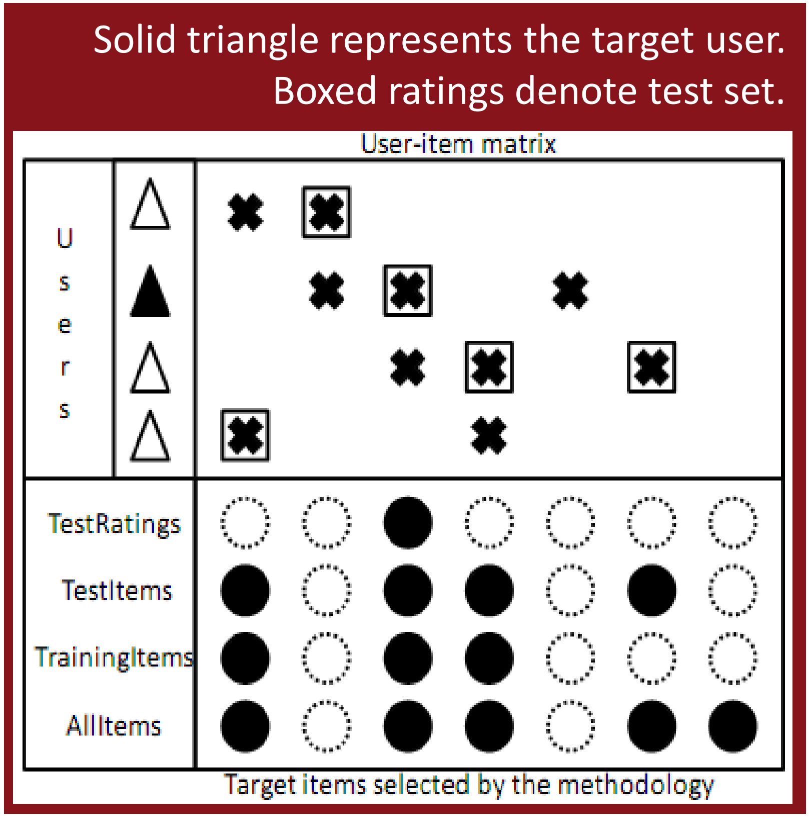
$$\mathrm{nDCG} = \frac{1}{|\mathcal{U}|} \sum_{u} \frac{1}{\mathrm{IDCG}_u^{p_u}} \sum_{p=1}^{p_u} \mathrm{f}_{\mathrm{dis}}\left(\mathrm{rel}\left(u, i_p\right), p\right)$$

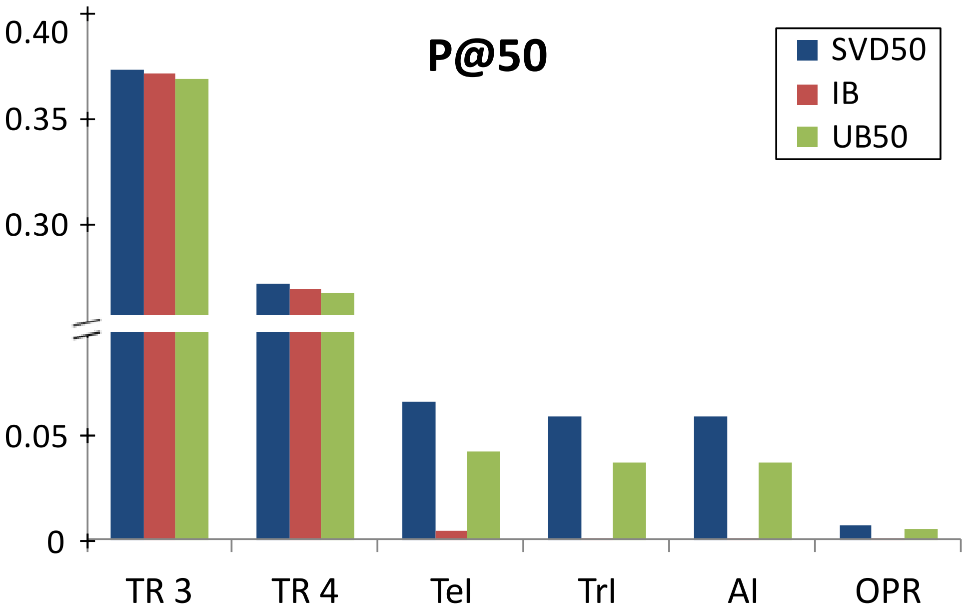
$$f_{\mathrm{dis}}(x, y) = (2^x - 1)/\log(1 + y)$$

$$f_{\mathrm{dis}}(x, y) = x/\log y \text{ if } y > 1$$

Some problems with “black boxes”

- How do you select the candidate items to be ranked?





Some problems with "black boxes"

- How do you select the candidate items to be ranked?

Alg.	F.W.	Time (sec.)	RMSE	nDCG@10 RPN	nDCG@10 UT	User cov.(%) RPN	User cov.(%) UT	Cat. cov.(%) RPN	Cat. cov.(%) UT
IBCos	AM	238	1.041	0.003	0.501	98.16	100	99.71	99.67
	LK	44	0.953	0.199	0.618	98.16	100	99.88	99.67
	MML	75	NA	0.488	0.521	98.16	100	100	99.67
IBPea	AM	237	1.073	0.022	0.527	97.88	100	86.66	99.31
	LK	31	1.093	0.033	0.527	97.86	100	86.68	99.31
	MML	1,346	0.857	0.882	0.654	98.16	100	2.87	99.83
SVD50	AM	132	0.950	0.286	0.657	98.12	100	99.88	99.67
	LK	7	1.004	0.280	0.621	98.16	100	100	99.67
	MML	1,324	0.848	0.882	0.648	98.18	100	2.87	99.83
UBCos50	AM	5	1.178	0.378	0.387	35.66	98.25	6.53	27.80
	LK	25	1.026	0.223	0.657	98.16	100	99.88	99.67
	MML	38	NA	0.519	0.551	98.16	100	100	99.67
UBPea50	AM	6	1.126	0.375	0.486	48.50	100	10.92	39.08
	LK	25	1.026	0.223	0.657	98.16	100	99.88	99.67
	MML	1,261	0.847	0.883	0.652	98.18	100	2.87	99.83

[Said & Bellogín, 2014]

Summary

- Important issues in recommendation
  - Validity of results (replicability)
  - Comparability of results (reproducibility)
  - Validity of experimental setup

- We need to incorporate reproducibility and replication to facilitate the progress in the field

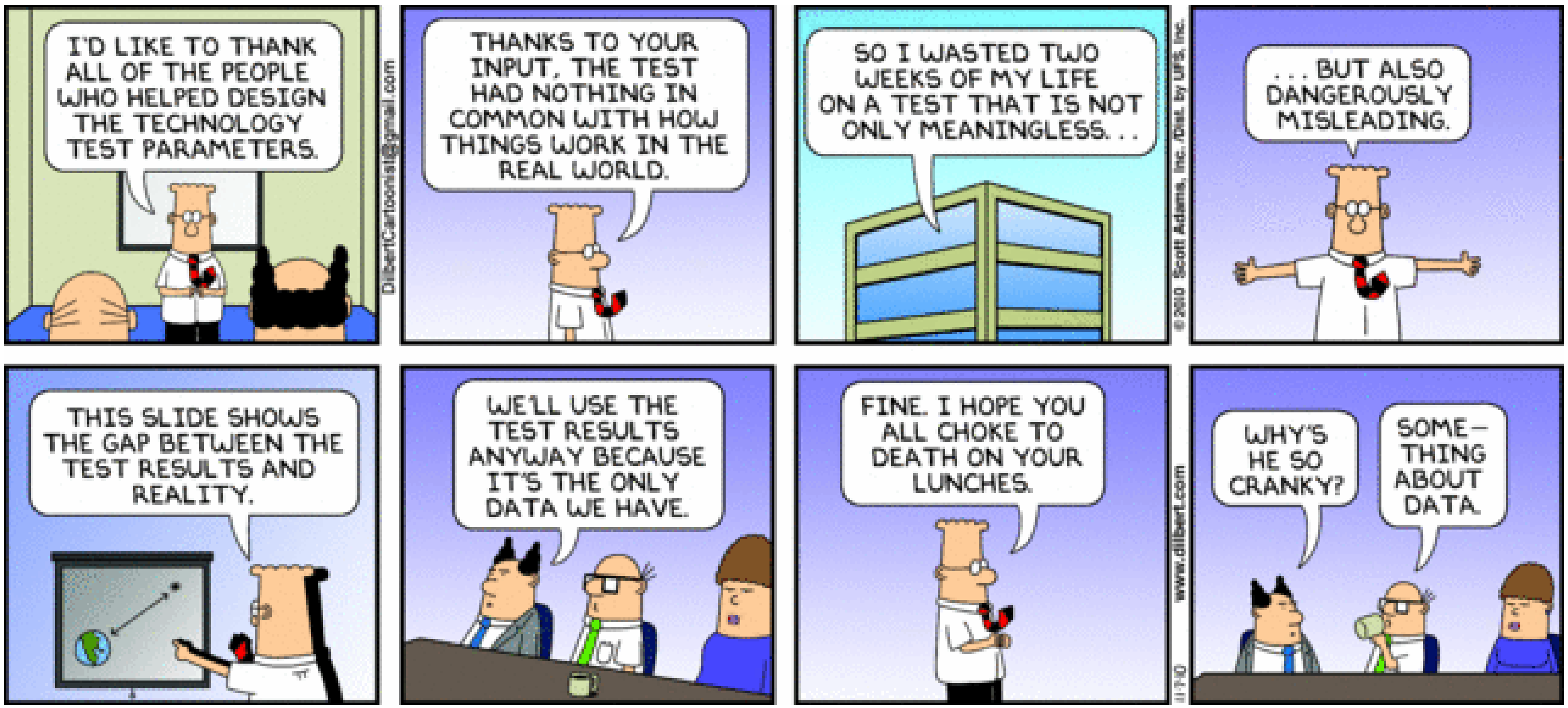
Outline

- Background and Motivation
- Evaluating Recommender Systems
- Reproducible Experimental Design
- **Summary**

Key Takeaways

- Every decision has an impact
  - We should log every step taken in the experimental part and report that log
- There are more things besides papers
  - Source code, web appendix, etc. are very useful to provide additional details not present in the paper
- You should not fool yourself
  - You have to be critical about what you measure and not trust intermediate “black boxes”

We must avoid this



From http://dilbert.com/strips/comic/2010-11-07/

Pointers

- Email and Twitter
  - Alejandro Bellogín
    - alejandro.bellogin@uam.es
    - @abellogin
  - Alan Said
    - alansaid@acm.org
    - @alansaid
- Slides:
    - In Slideshare... soon!

Thank you!

References and Additional reading

- [Armstrong et al, 2009] Improvements That Don’t Add Up: Ad-Hoc Retrieval Results Since 1998. CIKM
- [Bellogín et al, 2010] A Study of Heterogeneity in Recommendations for a Social Music Service. HetRec
- [Bellogín et al, 2011] Precision-Oriented Evaluation of Recommender Systems: an Algorithm Comparison. RecSys
- [Cremonesi et al, 2010] Performance of Recommender Algorithms on Top-N Recommendation Tasks. RecSys
- [Filippone & Sanguinetti, 2010] Information Theoretic Novelty Detection. Pattern Recognition
- [Fleder & Hosanagar, 2009] Blockbuster Culture’s Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. Management Science
- [Ge et al, 2010] Beyond accuracy: evaluating recommender systems by coverage and serendipity. RecSys
- [Gorla et al, 2013] Probabilistic Group Recommendation via Information Matching. WWW
- [Herlocker et al, 2004] Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems

References and Additional reading

- [Jambor & Wang, 2010] Goal-Driven Collaborative Filtering. ECIR
- [Knijnenburg et al, 2011] A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems. RecSys
- [Koren, 2008] Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. KDD
- [Lathia et al, 2010] Temporal Diversity in Recommender Systems. SIGIR
- [Li et al, 2010] Improving One-Class Collaborative Filtering by Incorporating Rich User Information. CIKM
- [Pu et al, 2011] A User-Centric Evaluation Framework for Recommender Systems. RecSys
- [Said & Bellogín, 2014] Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. RecSys
- [Schein et al, 2002] Methods and Metrics for Cold-Start Recommendations. SIGIR
- [Shani & Gunawardana, 2011] Evaluating Recommendation Systems. Recommender Systems Handbook
- [Vargas & Castells, 2011] Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. RecSys

References and Additional reading

- [Weng et al, 2007] Improving Recommendation Novelty Based on Topic Taxonomy. WI-IAT
- [Yin et al, 2012] Challenging the Long Tail Recommendation. VLDB
- [Zhang & Hurley, 2008] Avoiding Monotony: Improving the Diversity of Recommendation Lists. RecSys
- [Zhang & Hurley, 2009] Statistical Modeling of Diversity in Top-N Recommender Systems. WI-IAT
- [Zhou et al, 2010] Solving the Apparent Diversity-Accuracy Dilemma of Recommender Systems. PNAS
- [Ziegler et al, 2005] Improving Recommendation Lists Through Topic Diversification. WWW

Rank-score (Half-Life Utility)

Using a different discount function, the **rank score** or **half-life utility** metric (Breese et al., 1998; Herlocker et al., 2004; Huang et al., 2006) can be obtained as follows:

$$\mathrm{HL} = 100\left(\sum_{u} \mathrm{HL}_{u}^{\max}\right)^{-1} \sum_{u} \mathrm{HL}_{u}\,; \quad \mathrm{HL}_{u} = \sum_{p=1}^{p_u} \frac{\max(\tilde{r}(u, i_p) - d, 0)}{2^{(p-1)/(\alpha-1)}}$$

where d is the default ranking, and α is the half-life utility that represents the rank of the item on the list such that there is a 50% chance that the user will view that item. In (Breese et al., 1998) the authors use a value of 5 in their experiments, and note that they did not obtain different results with a half-life of 10.

Mean Reciprocal Rank

Mean reciprocal rank (MRR) favours rankings whose first correct result occurs near the top ranking results (Baeza-Yates and Ribeiro-Neto, 2011). It is defined as follows:

$$\mathrm{MRR} = \sum_{u} \frac{1}{s_r(u)}$$

where $s_r(u)$ is a function that returns the position of the first relevant item obtained for user u. This metric is similar to the **average rank of correct recommendation** (ARC) proposed in (Burke, 2004) and to the **average reciprocal hit-rank** (ARHR) defined in (Deshpande and Karypis, 2004).

Mean Percentage Ranking

Mean Percentage Ranking, which is used in [11] and [4], to measure the user satisfaction of items in an ordered list. Let $rank_{ui}$ be the percentile-ranking of item i within the ordered list of all items for user u. $rank_{ui} = 0\%$ means that item i is most preferred by user u. The higher ranking (until $rank_{ui} = 100\%$ is reached) indicates that i is predicted to be less desirable for user u. The way of calculating the MPR in our experiment setup is as the following: for each actual pair of a user and the purchased item, we randomly select 1000 other items, and produce an ordered list of these items. Then, we keep track of where the actual purchased item is ranked, and calculate the expected percentage ranking for all users and items:

$$MPR = \frac{\sum_{u,i} r_{ui} \times rank_{ui}}{\sum_{u,i} r_{ui}}$$

[Li et al, 2010]

Where r_{ui} is a binary variable indicating whether user u purchases item i. It is expected that a randomly produced list would have a MPR of around 50%.

Global ROC

We use a global ROC (GROC) curve to measure performance when we are allowed to recommend more often to some users than others. GROC curves are constructed in the following manner:

1. Order the predictions $\mathbf{pred}(p_i, m_j)$ in a list by magnitude, imposing an ordering: $(p, m)_k$.
2. Pick n, calculate hit/miss rates caused by predicting the top n $(p, m)_k$ by magnitude, and plot the point.

By selecting different n (e.g. incrementing n by a fixed amount) we draw a curve on the graph.

[Schein et al, 2002]

Customer ROC

Customer ROC (CROC) curves measure performance of a recommender system when we are constrained to recommend the same number of items to each user. Unlike the GROC curve, the CROC curve is not a special case of the ROC curve, though it is constructed in an analogous manner:

1. For each person p_i, order the predictions $\mathbf{pred}(p_i, m_j)$ in a list by magnitude imposing an ordering: $(m)_k$.
2. Pick n, calculate global hit/miss rates caused by recommending the top predicted n movies to each person and plot the point.

[Schein et al, 2002]

We vary n as in the GROC case.

In a GROC curve, the perfect recommender will generate a curve with area one, but for the CROC curve this is not the case. To see why, imagine using an omniscient recommender on a data set with three people: person a sees four movies, person b sees two movies, and person c sees six movies. When we recommend four movies to each person, we end up with two false-positives from person b, lowering the area of the curve. However, for any particular data set, we can plot the curve and calculate the area of the omniscient recommender in order to facilitate comparison.