Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks

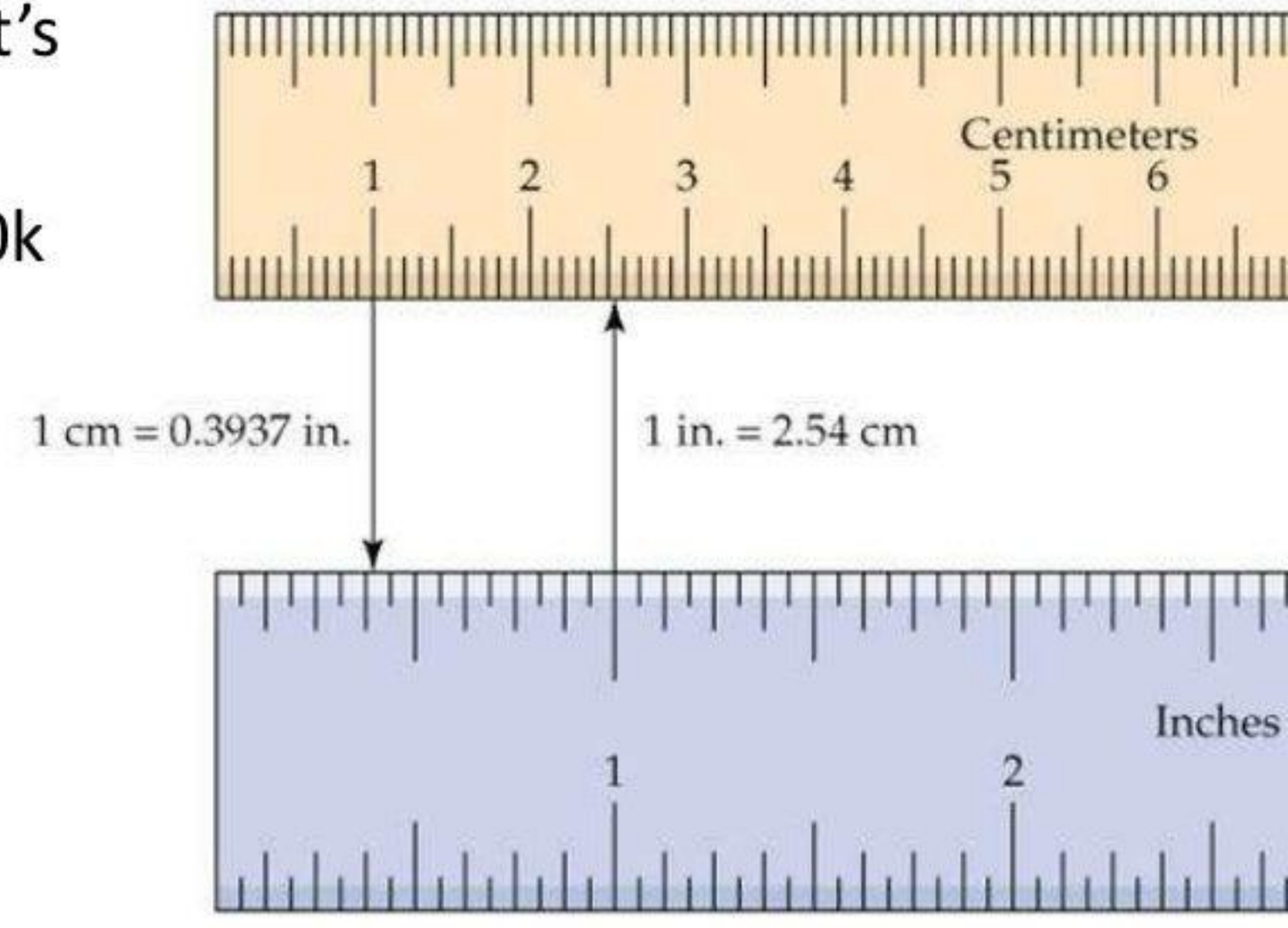
Alan Said
@alansaid
TU Delft

Alejandro Bellogín
@abellogin
Universidad Autónoma de Madrid

ACM RecSys 2014
Foster City, CA, USA

A RecSys paper outline

- We have a new model – it's great
- We used %DATASET% 100k to evaluate it
- It's 10% better than our baseline
- It's 12% better than [Authors, 2010]



Benchmarking

Python-recsys

Recommendable

LensKit

mahout

mrec

W:

LibRec

MyMedia

SLIM

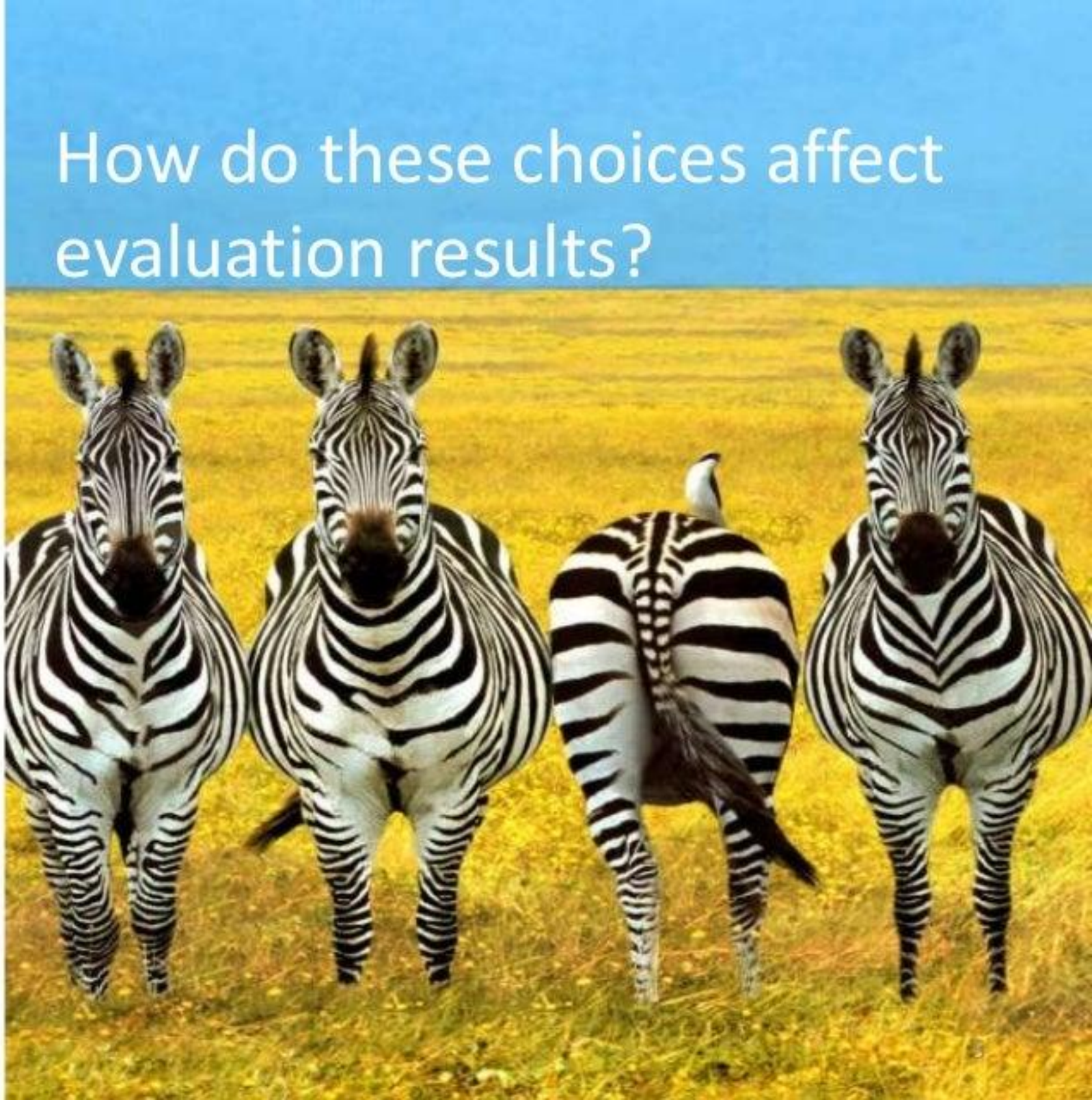
SVDFeature

What are the differences?

Some things just work differently

- Data splitting
- Algorithm design (implementation)
- Algorithm optimization
- Parameter values
- Evaluation
- Relevance/ranking
- Software architecture
- etc

Different design choices!!



Evaluate evaluation

- Comparison of frameworks
- Comparison of implementation
- Comparison of results
- Objective benchmarking

Algorithmic Implementation

Framework	Class	Similarity	
	Item-based		
LensKit	ItemItemScorer	CosineVectorSimilarity, PearsonCorrelation	
Mahout	GenericItemBasedRecommender	UncenteredCosineSimilarity, PearsonCorrelationSimilarity	
MyMediaLite	ItemKNN	Cosine, Pearson	
	User-based		**Parameters**
LensKit	UserUserItemScorer	CosineVectorSimilarity, PearsonCorrelation	SimpleNeighborhoodFinder, NeighborhoodSize
Mahout	GenericUserBasedRecommender	UncenteredCosineSimilarity, PearsonCorrelationSimilarity	NearestNUserNeighborhood, neighborhoodsize
MyMediaLite	UserKNN	Cosine, Pearson	neighborhoodsize
	Matrix Factorization		
LensKit	FunkSVDItemScorer	IterationsCountStoppingCondition, factors, iterations	
Mahout	SVDRecommender	FunkSVDFactorizer, factors, iterations	
MyMediaLite	SVDPlusPlus	factors, iterations	

There's more than algorithms though

There's the data, evaluation, and more

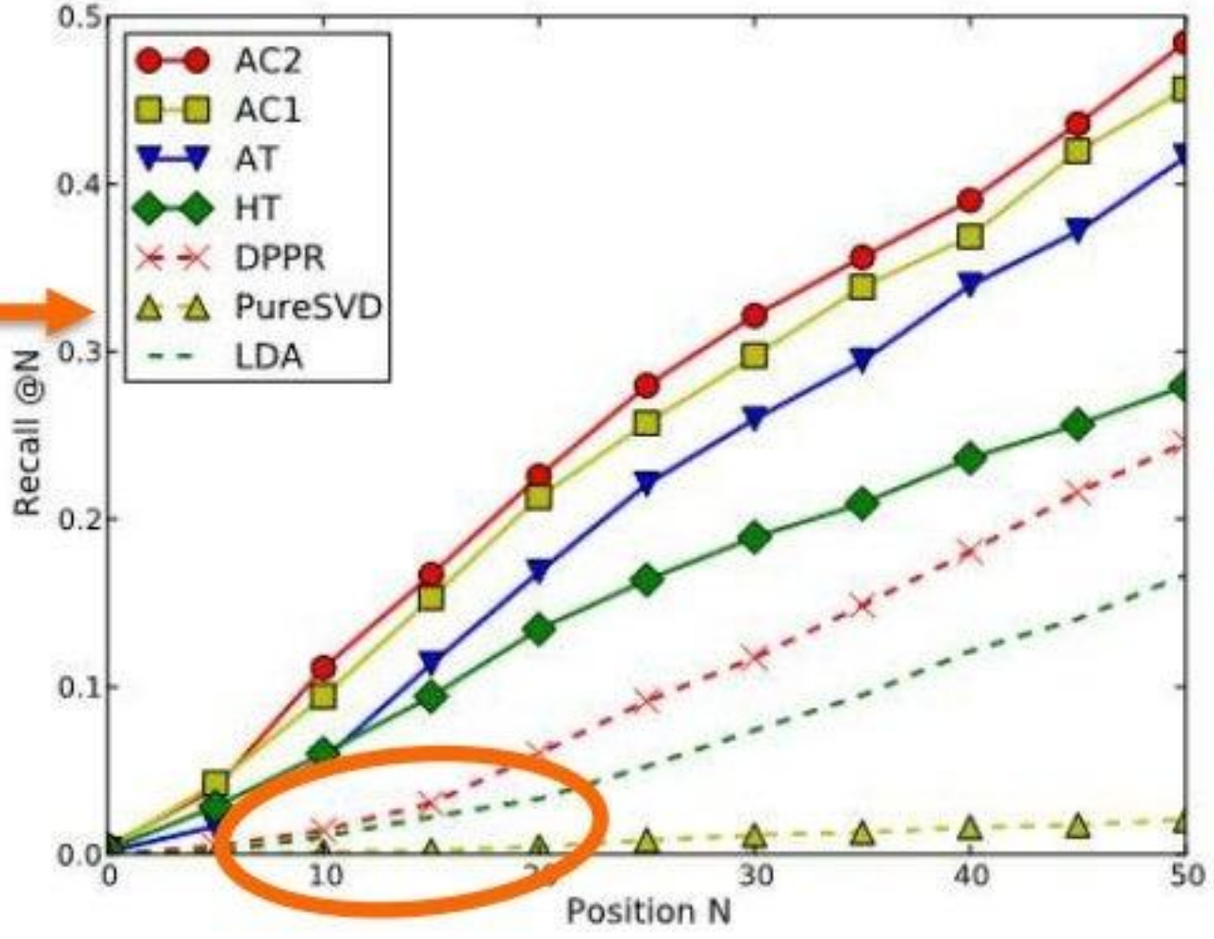
Data splits

- 80-20 Cross-validation
- Random Cross-validation
- User-based cross validation
- Per-user splits
- Per-item splits
- Etc.

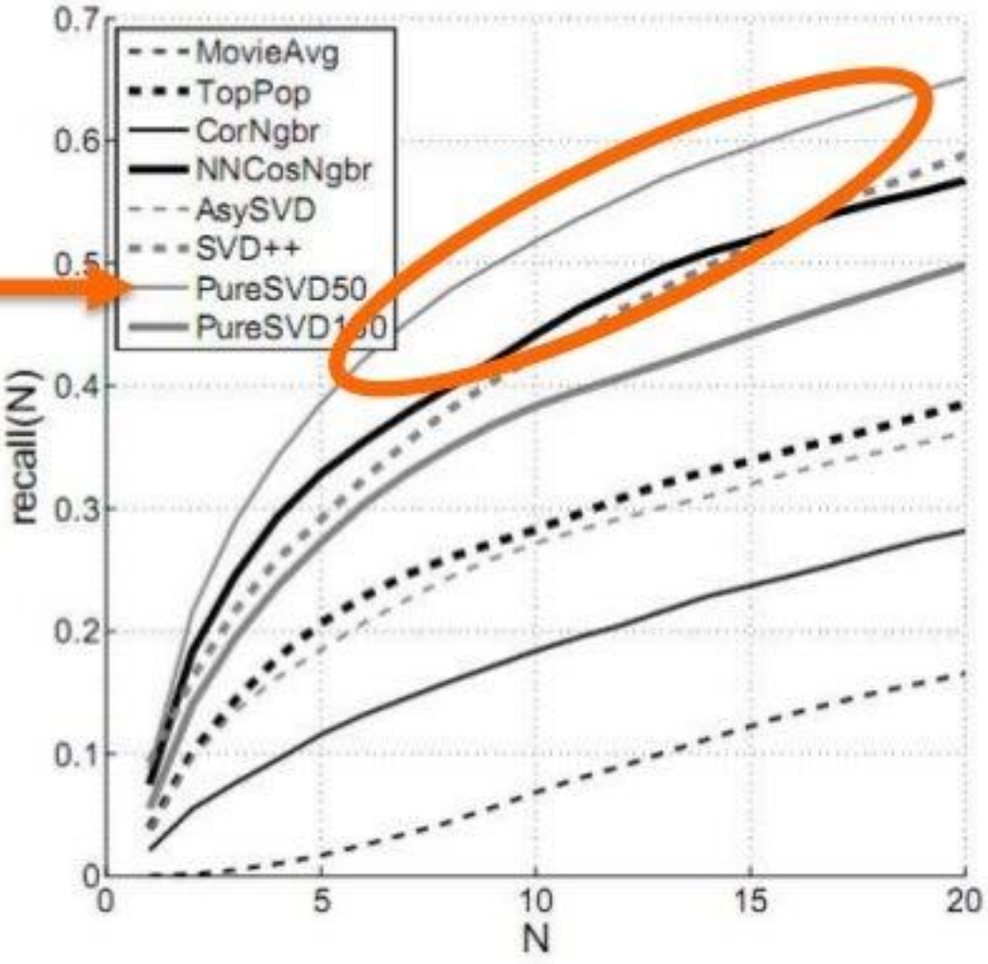
Evaluation

- Metrics
- Relevance
- Strategies

Real world examples

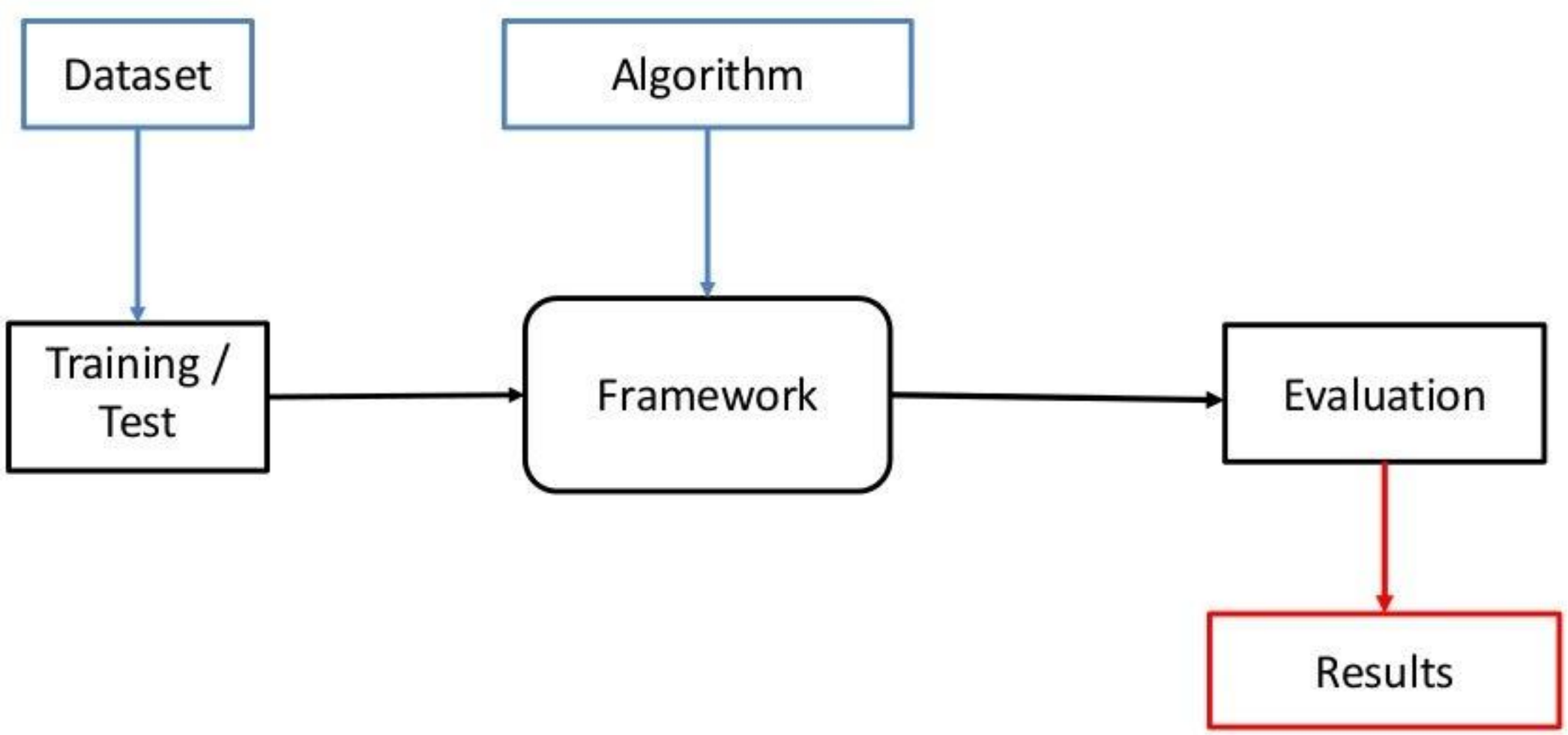


Movielens 1M
[Yin et al, 2012]

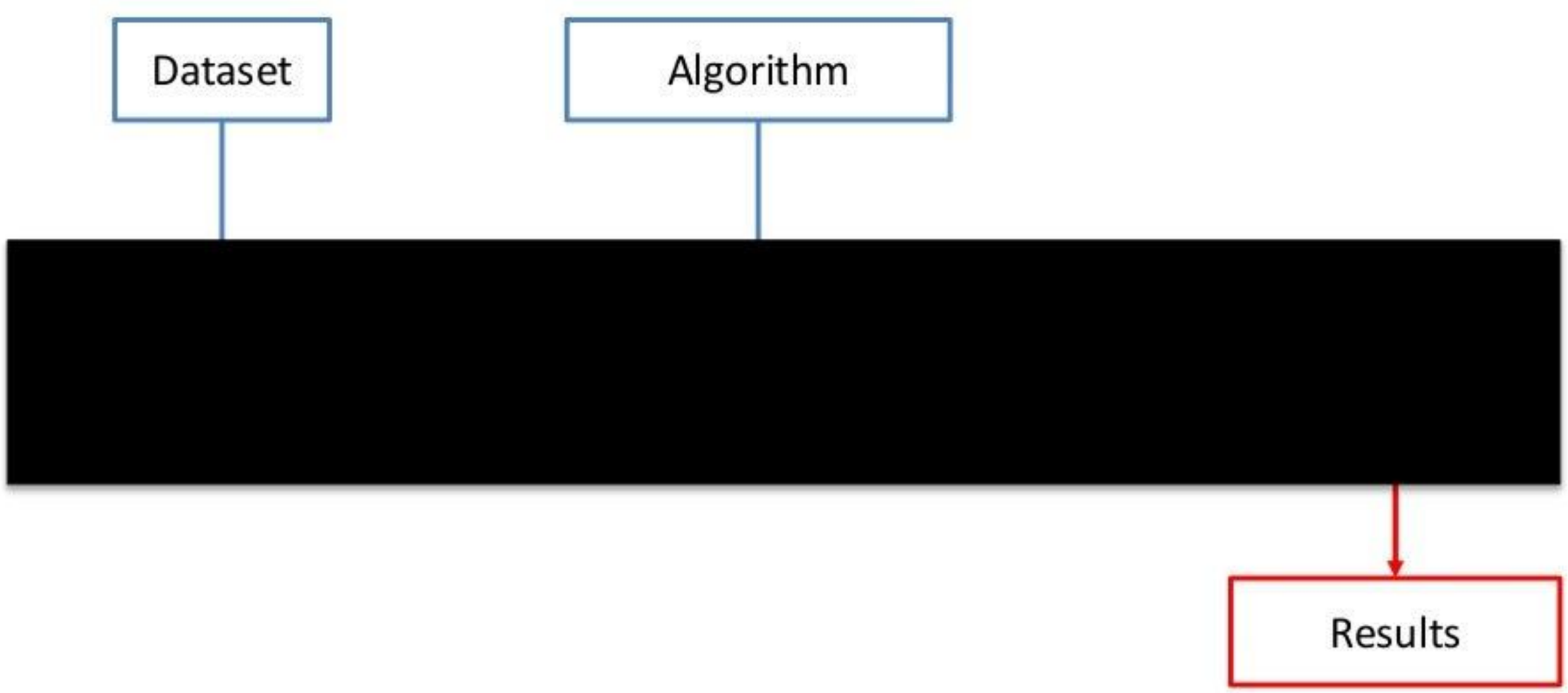


(a) recall
Movielens 1M
[Cremonesi et al, 2010]

Evaluation


Dataset
Algorithm
Training / Test
Framework
Evaluation
Results

Internal Evaluation



Internal Evaluation Results

Algorithm	Framework	nDCG
IB Cosine	Mahout	0,00041478
IB Cosine	Lenskit	0,94219205
IB Pearson	Mahout	0,00516923
IB Pearson	Lenskit	0,92454613
SVD50	Mahout	0,10542729
SVD50	Lenskit	0,94346409
UB Cosine	Mahout	0,16929545
UB Cosine	Lenskit	0,94841356
UB Pearson	Mahout	0,16929545
UB Pearson	Lenskit	0,94841356

Algorithm	Framework	RMSE
IB Cosine	Lenskit	1,01390931
IB Cosine	MyMediaLite	0,92476162
IB Pearson	Lenskit	1,05018614
IB Pearson	MyMediaLite	0,92933246
SVD50	Lenskit	1,01209290
SVD50	MyMediaLite	0,93074012
UB Cosine	Lenskit	1,02545490
UB Cosine	MyMediaLite	0,93419026


We need a fair and common evaluation protocol!

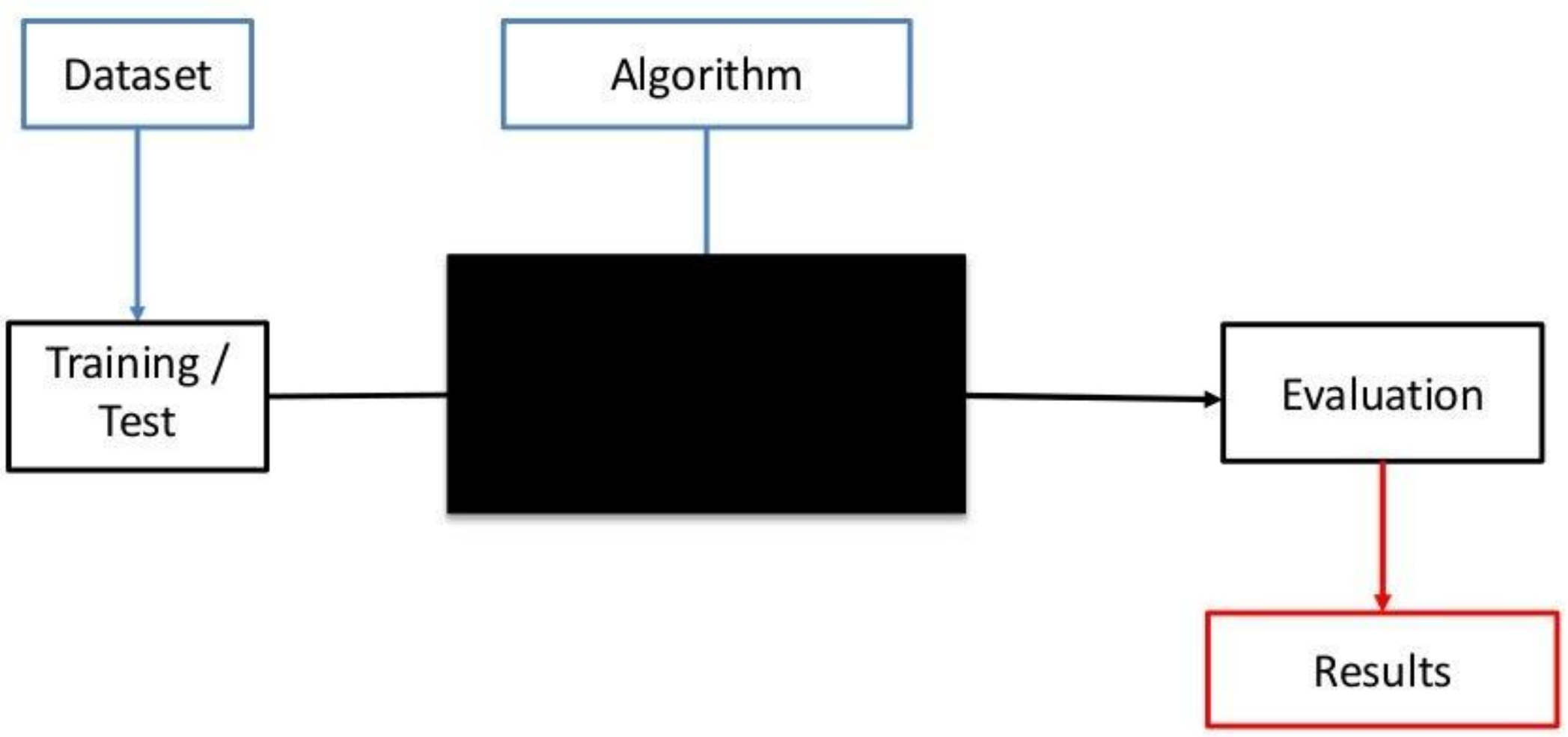
Reproducible evaluation - Benchmarking

Control all parts of the process

- Data Splitting strategy
- Recommendation (black box)
- Candidate items generation (what items to test)
- Evaluation

http://rival.recommenders.net

Controlled Evaluation

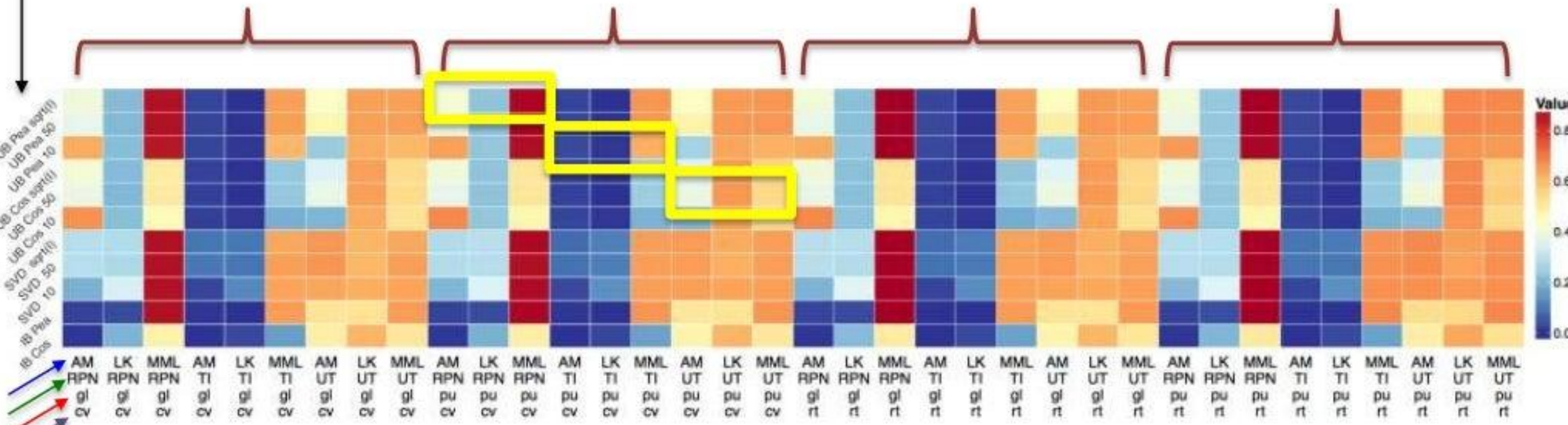


Lenskit vs. Mahout vs. MyMediaLite

Movielens 100k (additional datasets in the paper)

AN OBJECTIVE BENCHMARK

nDCG@10



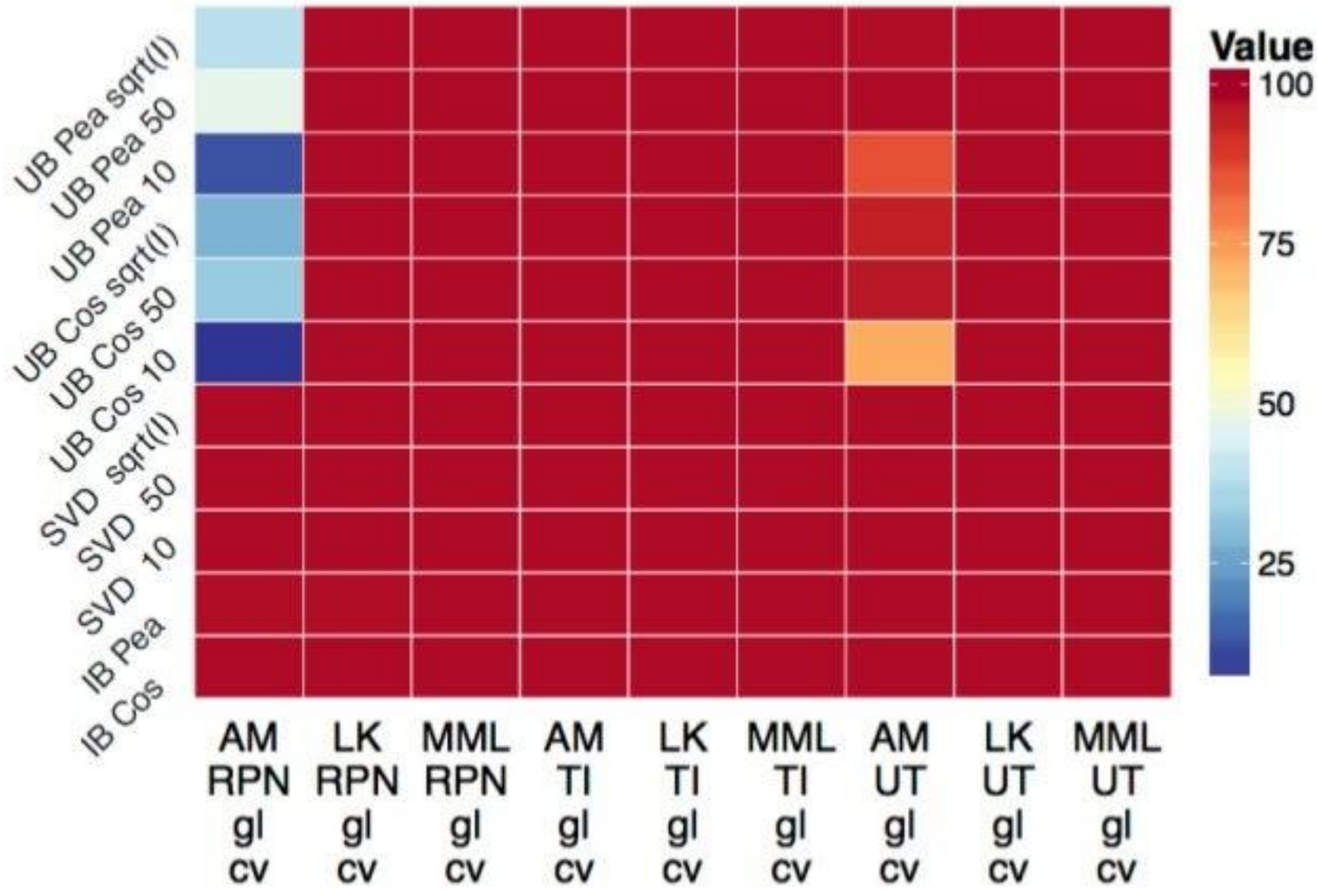
The Frameworks
AM: Apache Mahout
LK: Lenskit
MML: MyMediaLite

The Candidate Items
RPN: Relevant + N [Koren, KDD 2008]
TI: TrainItems
UT: UserTest

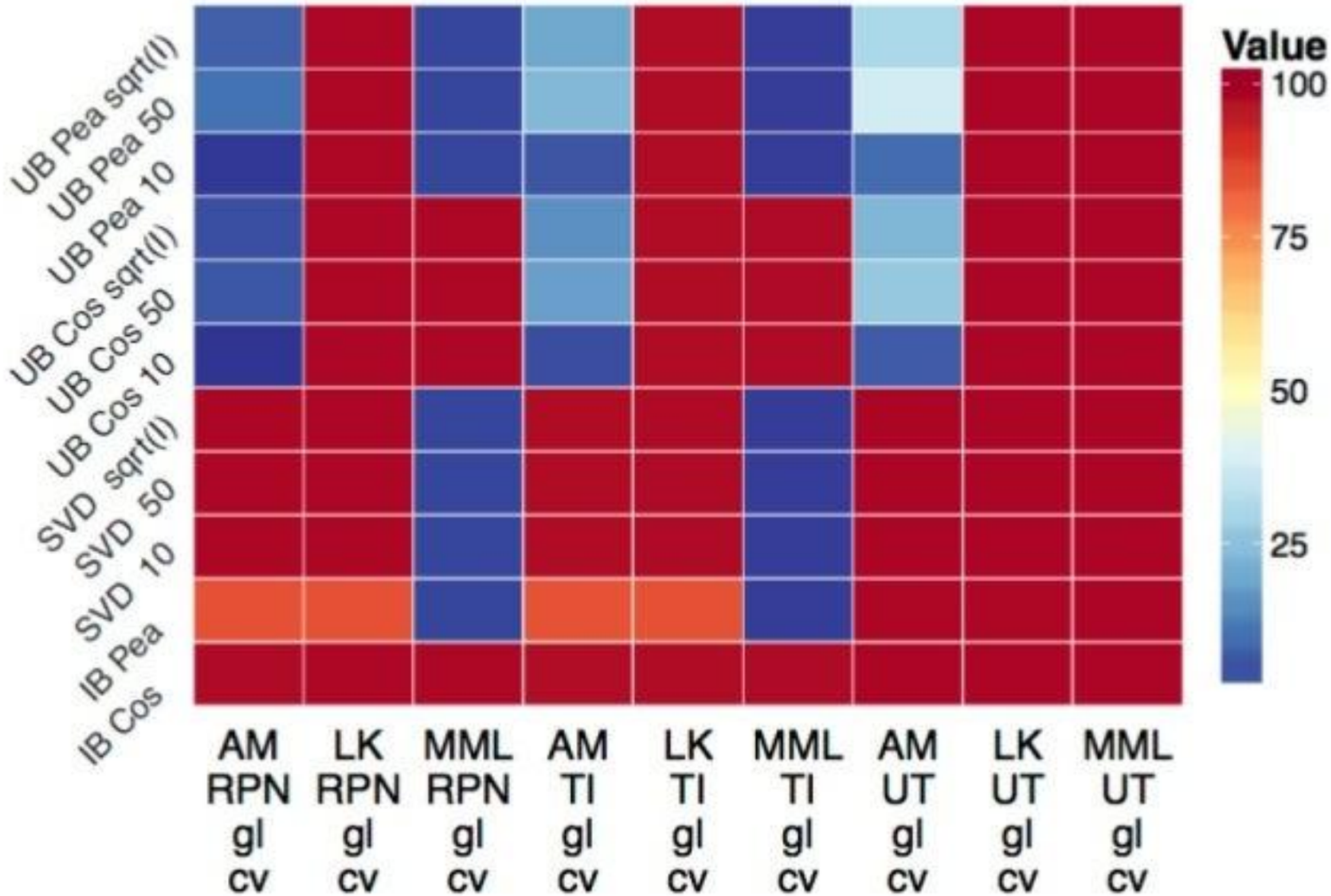
Split Point
gl: Global
pu: Per-user

Split Strategy
cv: 5-fold cross-validation
rt: 80-20 random ratio

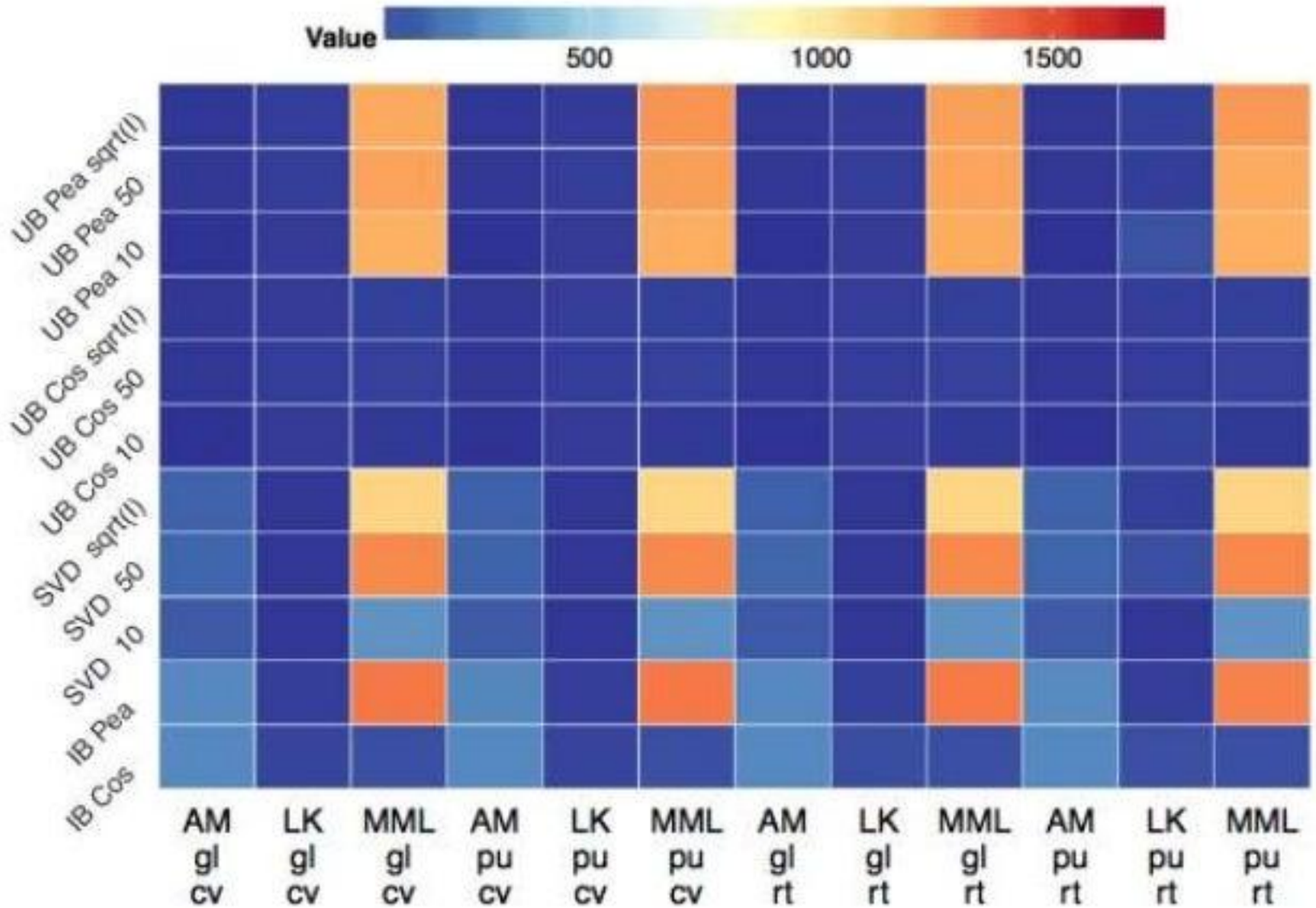
User Coverage



Catalog Coverage



Time


Value
500
1000
1500
UB Pea sqrt(I)
UB Pea 50
UB Pea 10
UB Cos sqrt(I)
UB Cos 50
UB Cos 10
SVD sqrt(I)
SVD 50
SVD 10
IB Pea
IB Cos
AM gl cv
LK gl cv
MML gl cv
AM pu cv
LK pu cv
MML pu cv
AM gl rt
LK gl rt
MML gl rt
AM pu rt
LK pu rt
MML pu rt