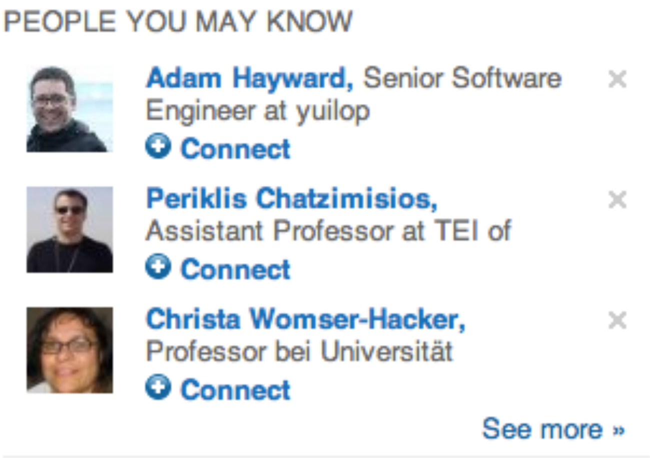
CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering

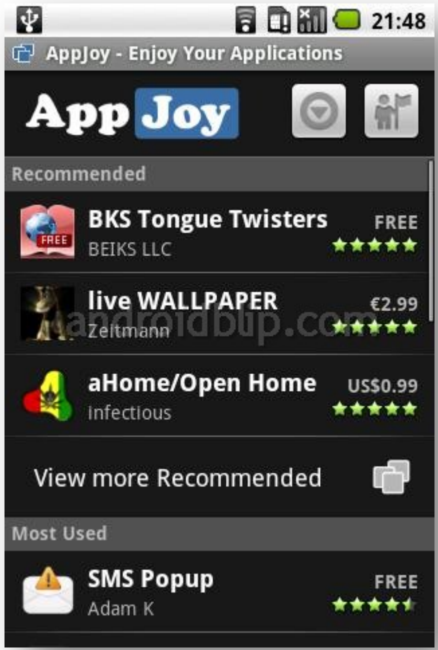
Yue Shi[a], Alexandros Karatzoglou[b] (Presenter), Linas Baltrunas[b], Martha Larson[a], Alan Hanjalic[a], Nuria Oliver[b]

[a]Delft University of Technology, Netherlands

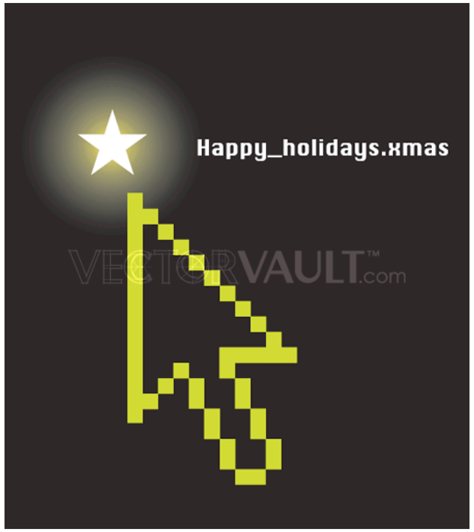
[b]Telefonica Research, Spain

Top-k Recommendations





Implicit feedback



Models for Implicit feedback

- Classification or Learning to Rank
- Binary pairwise ranking loss function (Hinge, AUC loss)
- Sample from non-observed/irrelevant entries

Friend

>

Not a Friend

Learning to Rank in CF

- The Point-wise Approach $f(user, item) \rightarrow \mathbb{R}$
  - Reduce Ranking to Regression, Classification, or Ordinal Regression problem, [OrdRec@Recys 2011]
- The Pairwise Approach $f(user, item_1, item_2) \rightarrow \mathbb{R}$
  - Reduce Ranking to pair-wise classification [BPR@UAI 2010]
- List-wise Approach $f(user, item_1, \ldots, item_n) \rightarrow \mathbb{R}$
  - Direct optimization of IR measures, List-wise loss minimization [CoFiRank@NIPS 2008]

Ranking metrics

List-wise ranking measures for implicit feedback:

Reciprocal Rank:

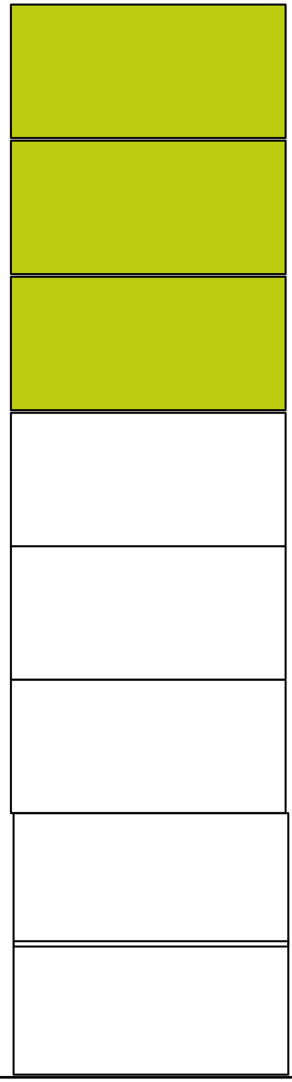
$$RR = \frac{1}{rank_i}$$

Average precision:

[TFMAP@SIGIR2012]

$$\text{AP} = \frac{\sum_{k=1}^{|S|} P(k)}{|S|}$$

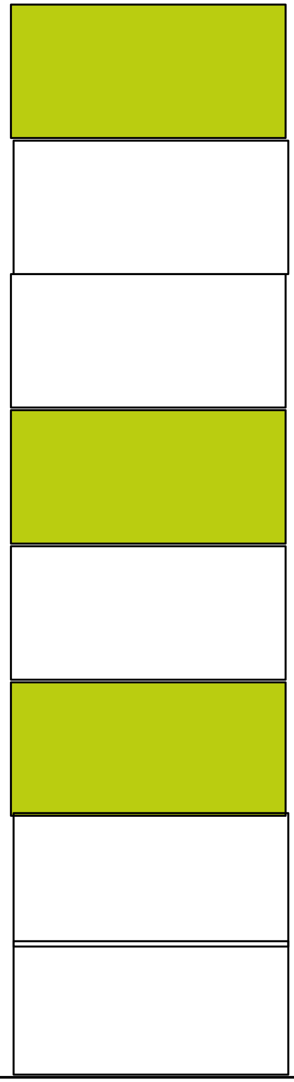
Ranking metrics



RR = 1

AP = 1

Ranking metrics



RR = 1

AP = 0.66

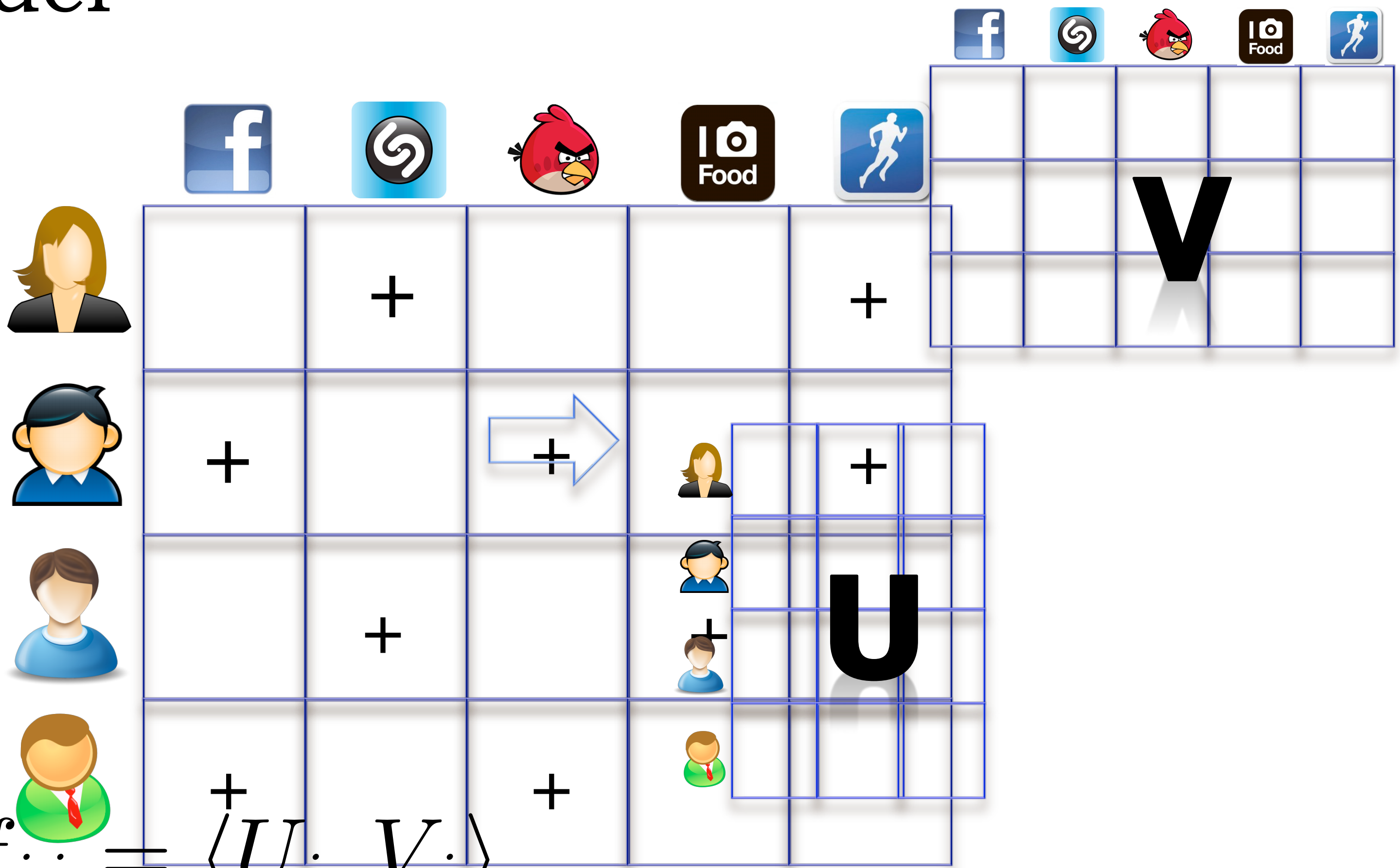
Less is more

- Focus at the very top of the list
- Try to get at least one interesting item at the top of the list
- MRR particularly important measure in domains that usually provide users with only few recommendations, i.e. Top-3 or Top-5

Ingredients

- What kind of model should we use?
  - Factor model
- Which Ranking measure do we need to optimize to have a good Top-k recommender?
  - MRR captures the quality of Top-k recommendations
- But MRR is not smooth so what can we do?
  - We can perhaps find a smooth version of MRR
- How to ensure the proposed solution scalable?
  - A fast learning algorithm (SGD), smoothness -> gradients

Model



$$f_{ij} = \langle U_i, V_j \rangle$$

The Non-smoothness of Reciprocal Rank

- Reciprocal Rank (RR) of a ranked list of items for a given user

$$RR_i = \sum_{j=1}^{N} \frac{Y_{ij}}{R_{ij}} \prod_{k=1}^{N} (1 - Y_{ik}\mathbb{I}(R_{ik} < R_{ij}))$$

Non-smoothness

F_i

0.81	2
0.75	1
0.64	3
0.61	4
0.58	5
0.55	6
0.49	7
0.43	8

RR = 0.5

Non-smoothness

F_i

0.84	2
0.82	1
0.56	3
0.50	4
0.45	5
0.40	6
0.32	7
0.31	8

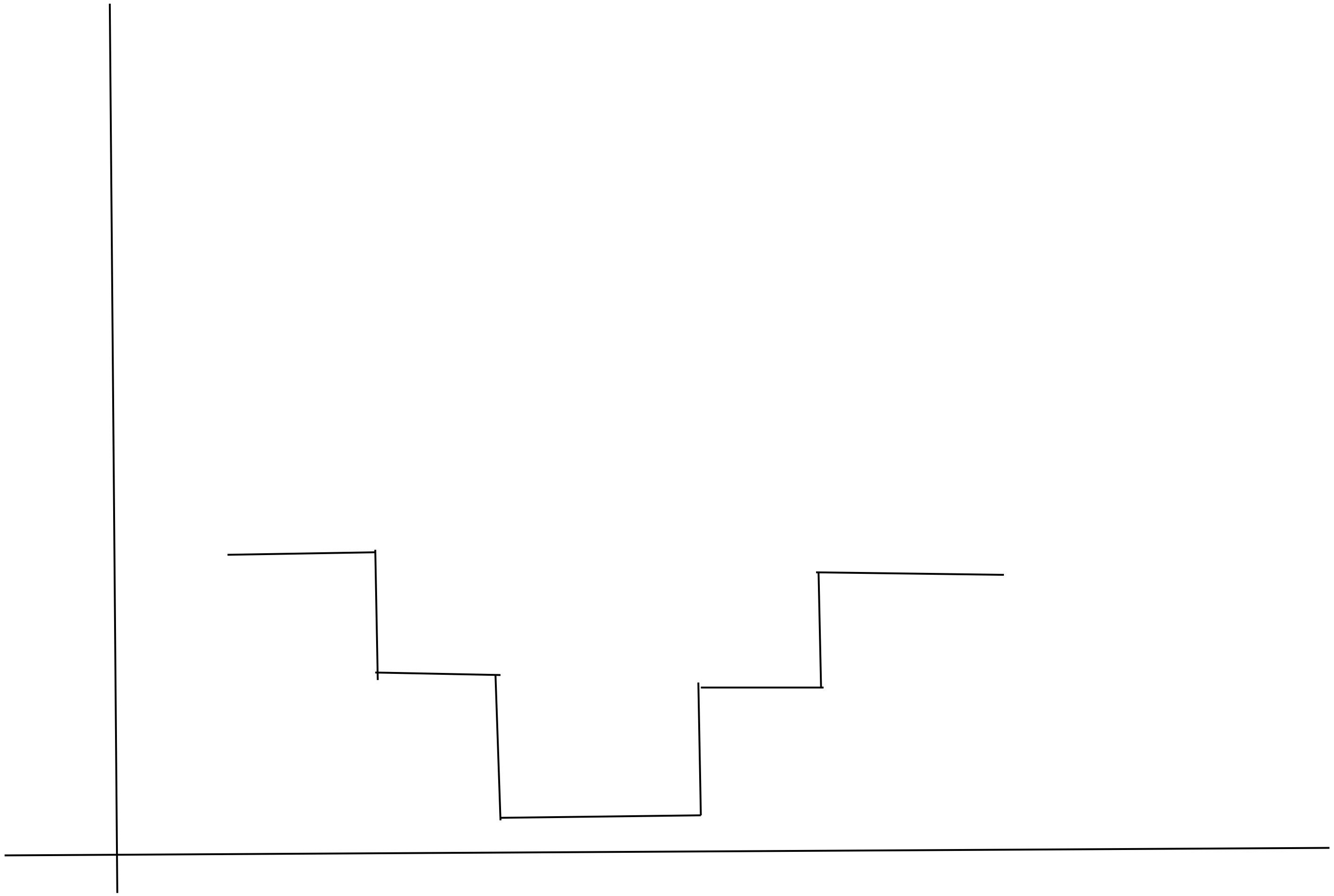
RR = 0.5

Non-smoothness

F_i

0.85	1
0.84	2
0.56	3
0.50	4
0.45	5
0.40	6
0.32	7
0.31	8

RR = 1



How can we get a smooth-MRR?

- Borrow techniques from learning-to-rank:

$$\mathbb{I}(R_{ik} < R_{ij}) \approx g(f_{ik} - f_{ij})$$

$$\frac{1}{R_{ij}} \approx g(f_{ij})$$

$$g(x) = 1/(1 + e^{-x})$$

MRR Loss function

$$RR_i \approx \sum_{j=1}^{N} Y_{ij} g(f_{ij}) \prod_{k=1}^{N} \big(1 - Y_{ik} g(f_{ik} - f_{ij})\big)$$

$$f_{ij} = \langle U_i, V_j \rangle$$

$$U_i, V = \arg\max_{U_i, V} \{RR_i\}$$

$$O(N^2)$$

MRR loss function II

- Use concavity and monotonicity of log function,

$$L(U_i, V) = \sum_{j=1}^{N} Y_{ij} \left[\ln g(f_{ij}) + \sum_{k=1}^{N} \ln \left(1 - Y_{ik} g(f_{ik} - f_{ij})\right) \right]$$

$$O(n^{+2})$$

Optimization

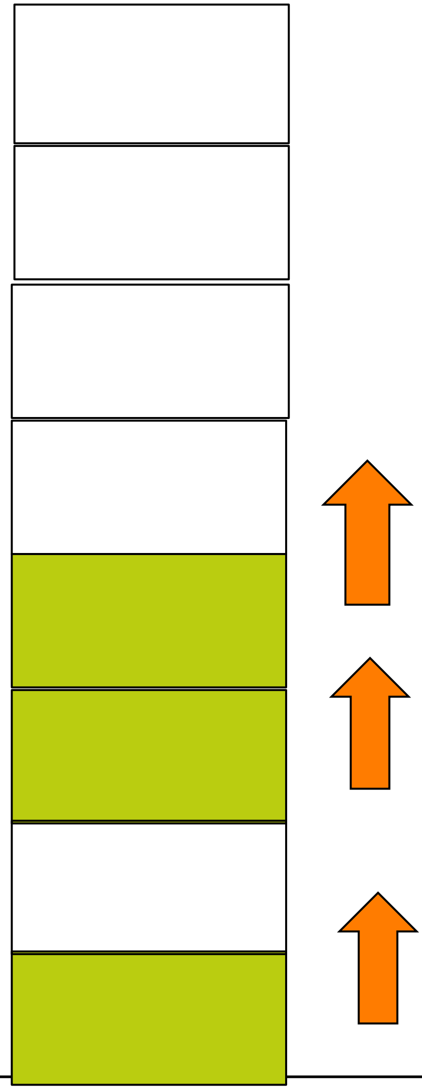
$$E(U,V) = \sum_{i=1}^{M}\sum_{j=1}^{N} Y_{ij}\Big[\ln g(U_i^T V_j)$$
$$+ \sum_{k=1}^{N} \ln\Big(1 - Y_{ik} g(U_i^T V_k - U_i^T V_j)\Big)\Big]$$
$$- \frac{\lambda}{2}(\|U\|^2 + \|V\|^2)$$

- Objective is smooth we can compute: $\frac{\partial E}{\partial U_i}$ $\frac{\partial E}{\partial V_j}$
- We use Stochastic Gradient Descent
- Overall scalability linear to # of relevant items $O(dS)$

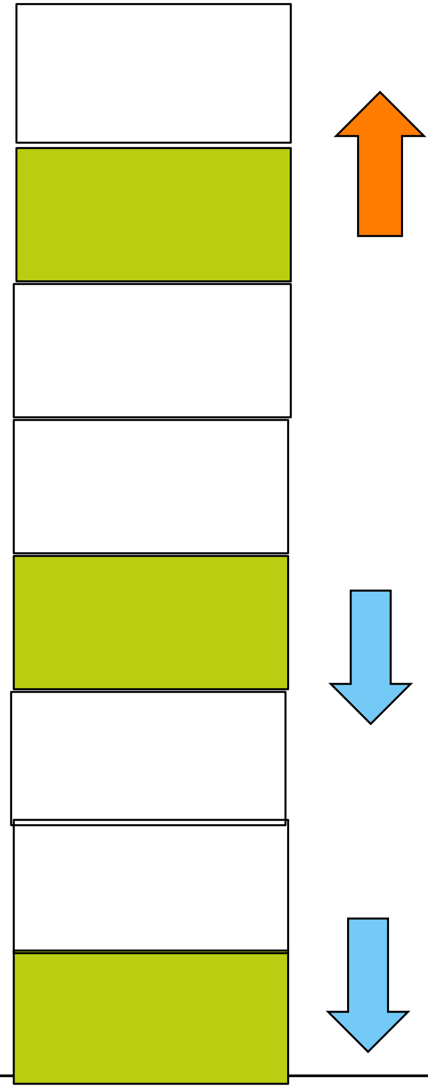
What’s different ?

- CLiMF reciprocal rank loss essentially pushes relevant items apart
- In the process at least one items ends up high-up in the list

Conventional loss



CLiMF MRR-loss



Experimental Evaluation

Data sets

- Epinions :
  - 346K observations of trust relationship
  - 1767 users; 49288 trustees
  - 99.85% Sparseness
  - Avg. friends/trustees per user 73.34

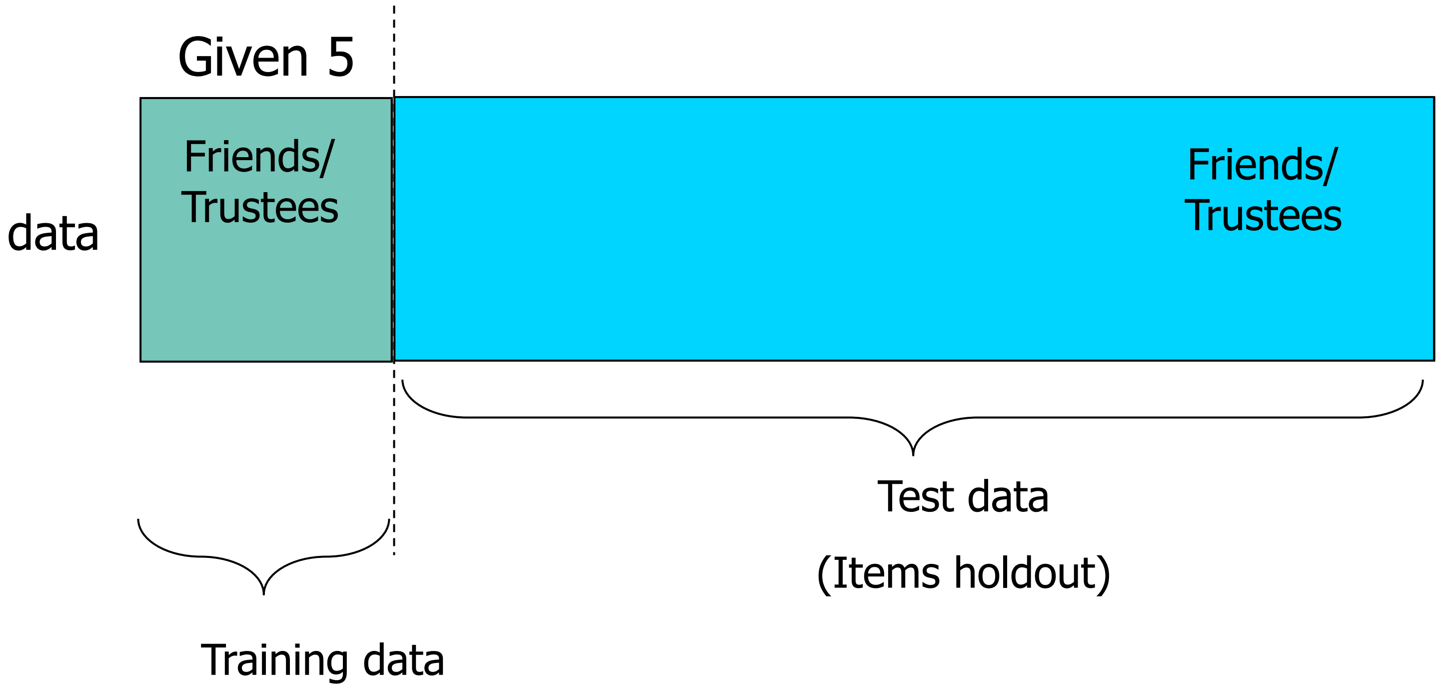
Experimental Evaluation

Data sets

- Tuenti :
  - 798K observations of trust relationship
  - 11392 users; 50000 friends
  - 99.86% Sparseness
  - Avg. friends/trustees per user 70.06

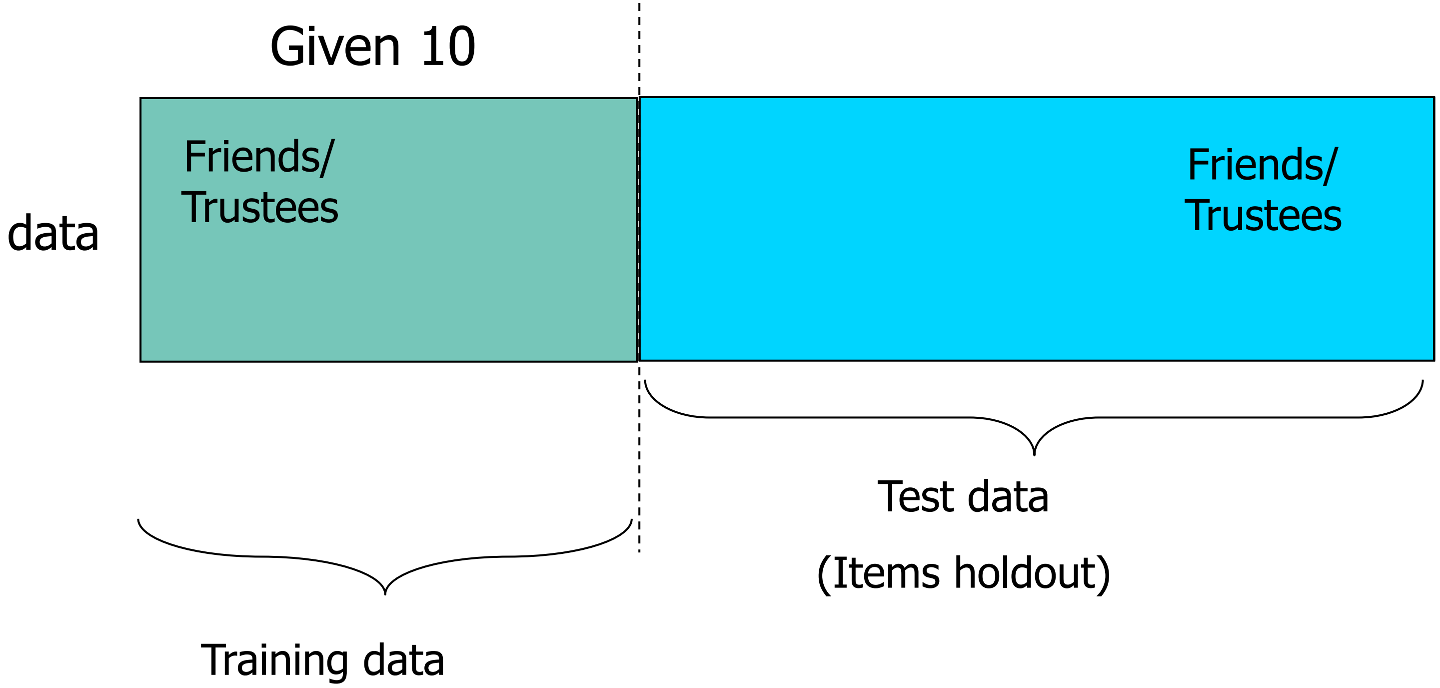
Experimental Evaluation

Experimental Protocol



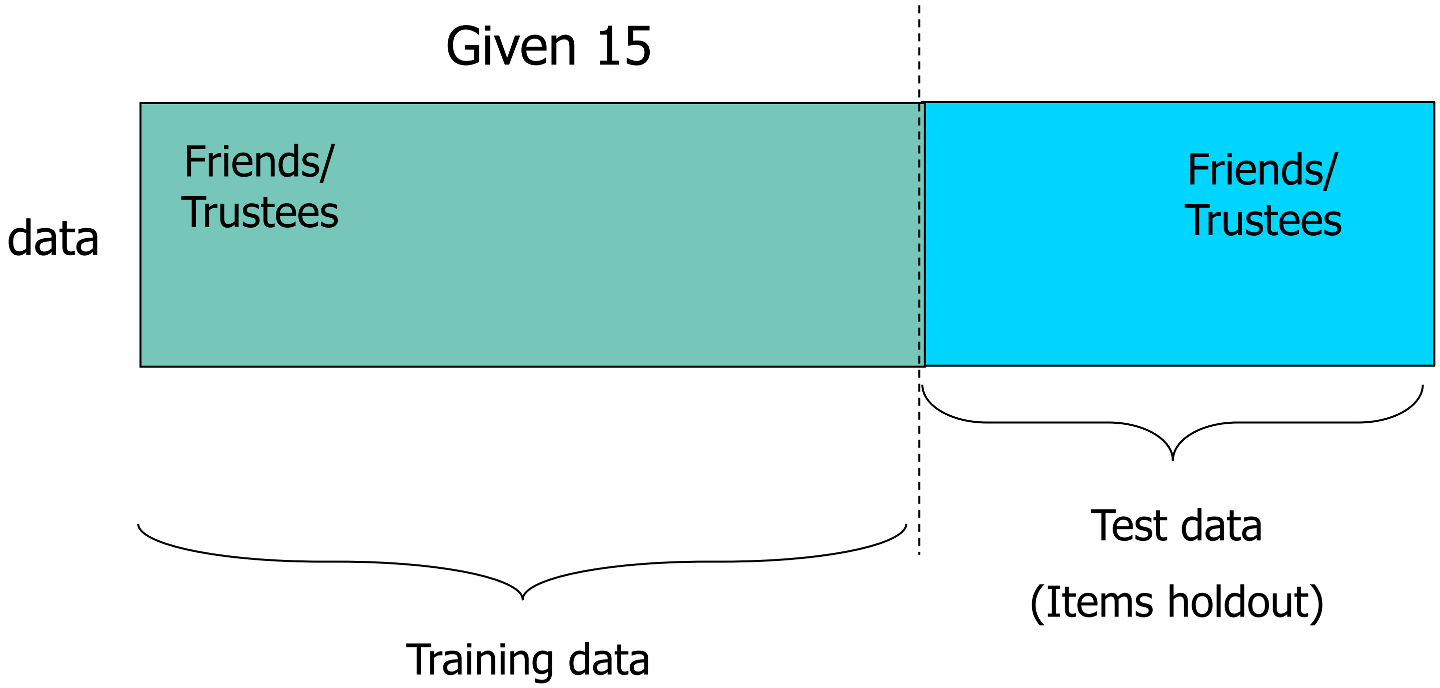
Experimental Evaluation

Experimental Protocol



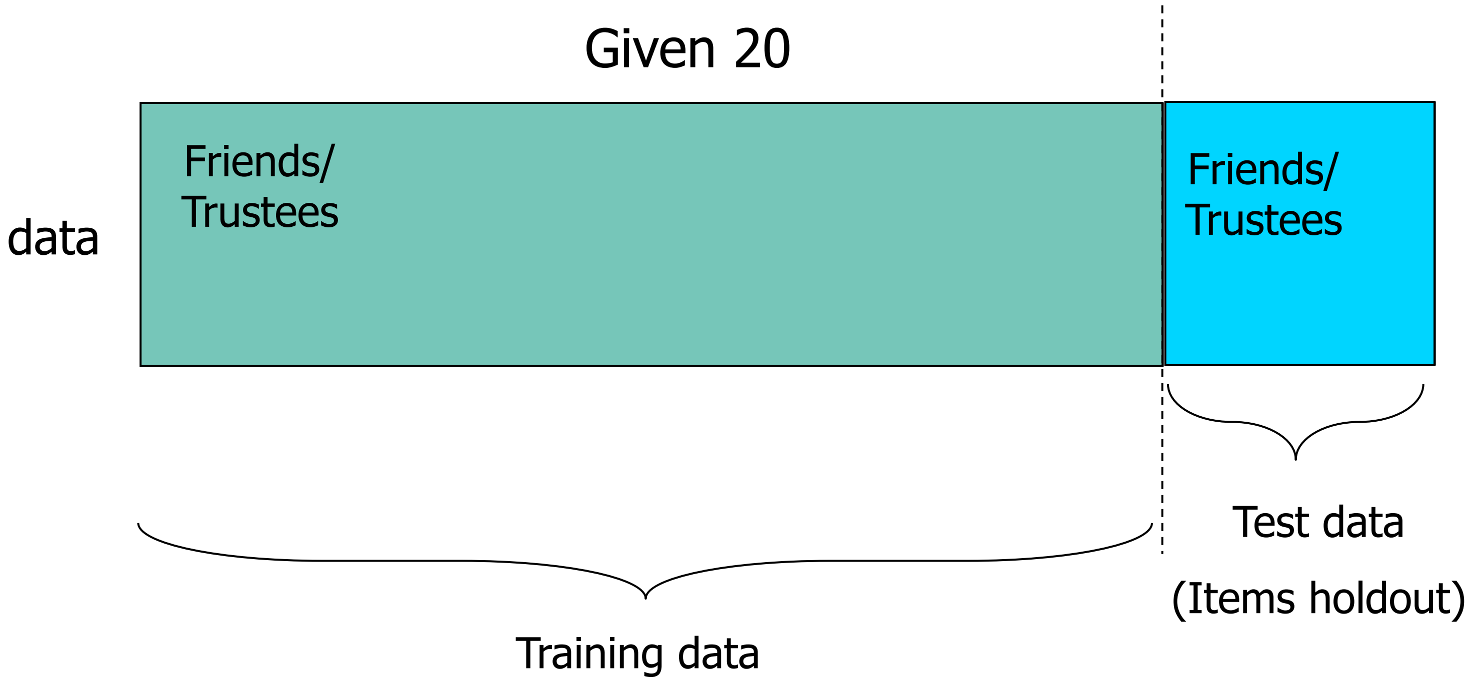
Experimental Evaluation

Experimental Protocol



Experimental Evaluation

Experimental Protocol



Experimental Evaluation
Evaluation Metrics

$$MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{rank_i}$$

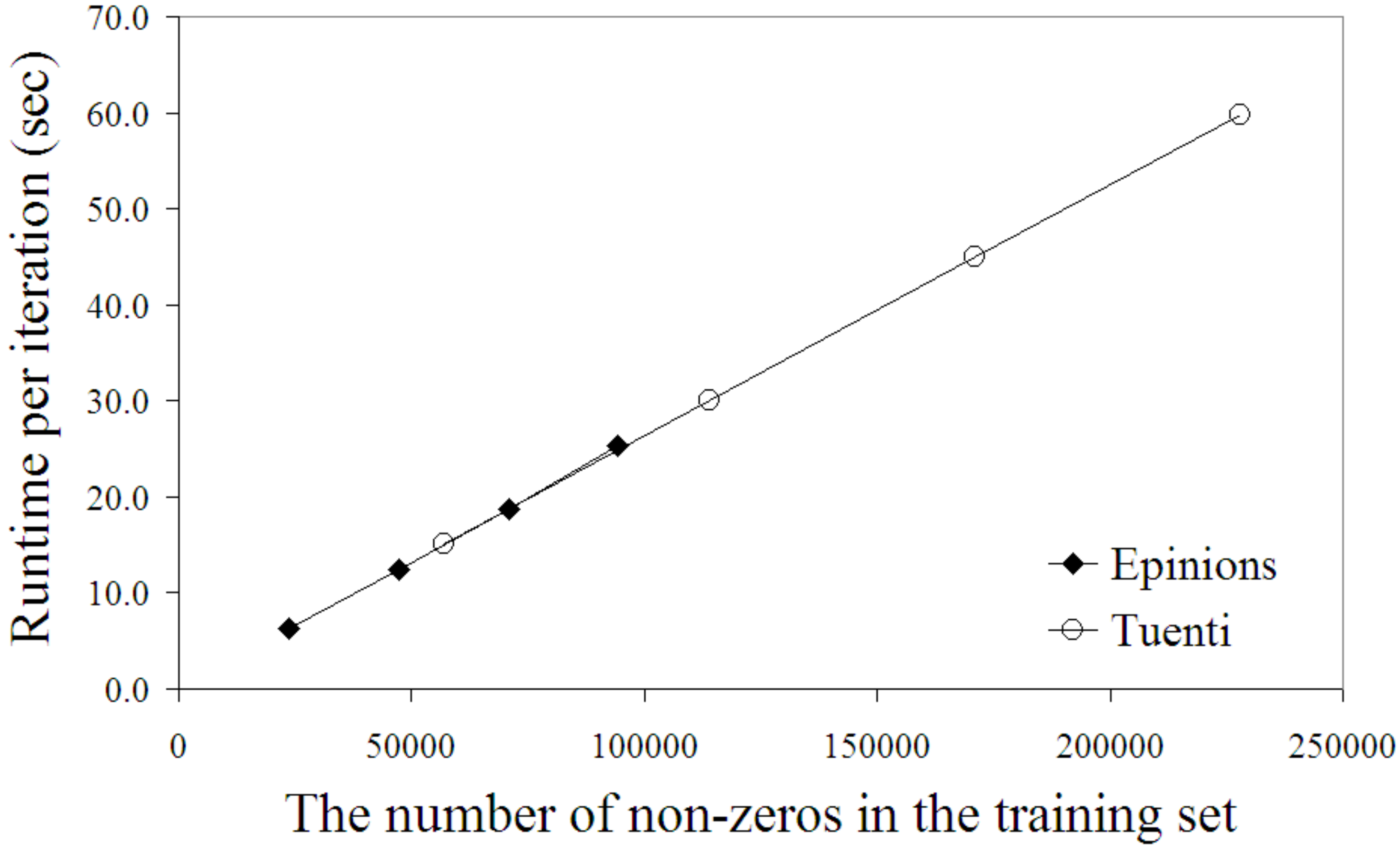
$$P@5$$

$$1 - call@5$$

ratio of test users who have at least one relevant item in their Top-5

Experimental Evaluation

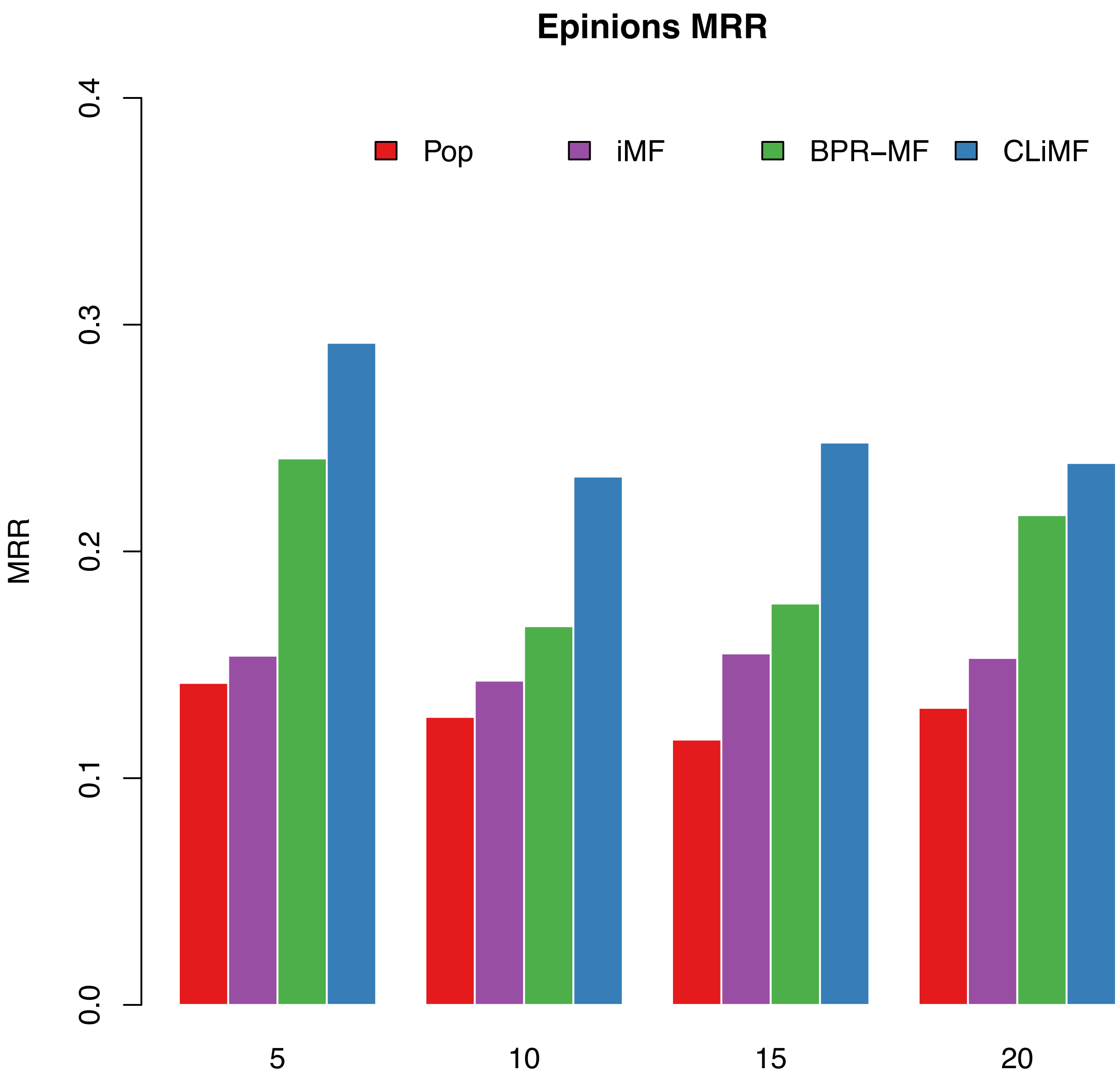
Scalability

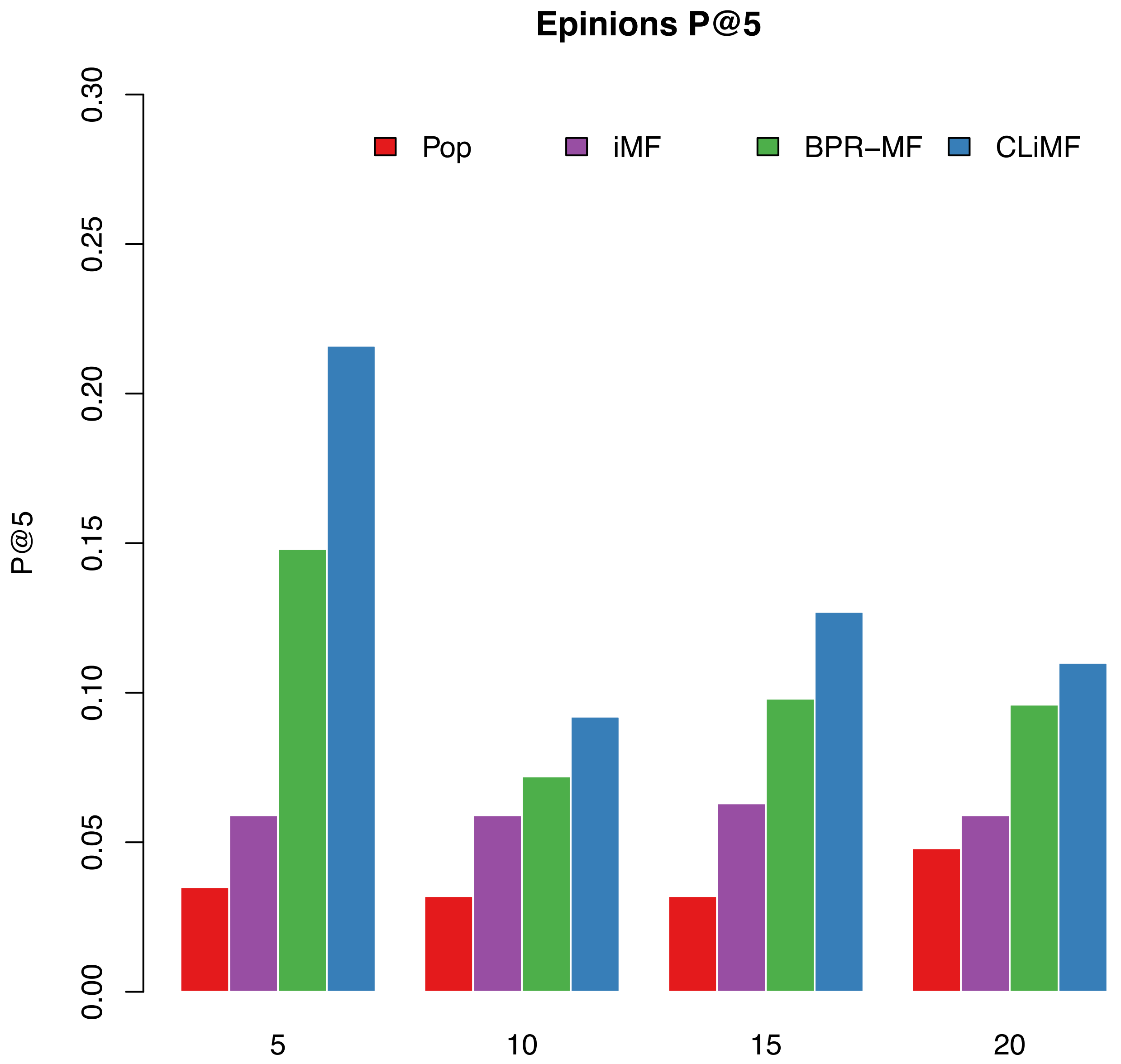


Experimental Evaluation

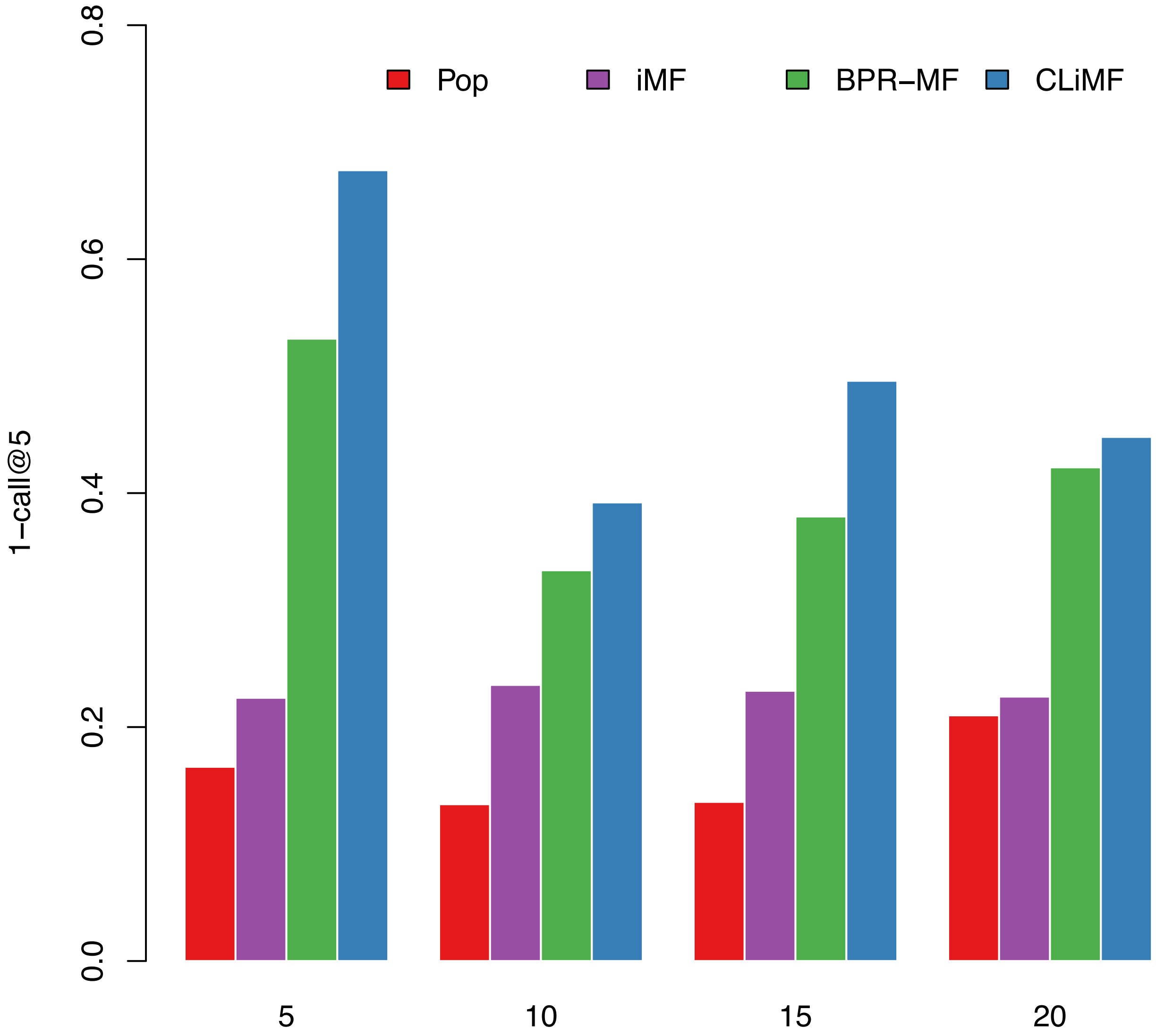
Competition

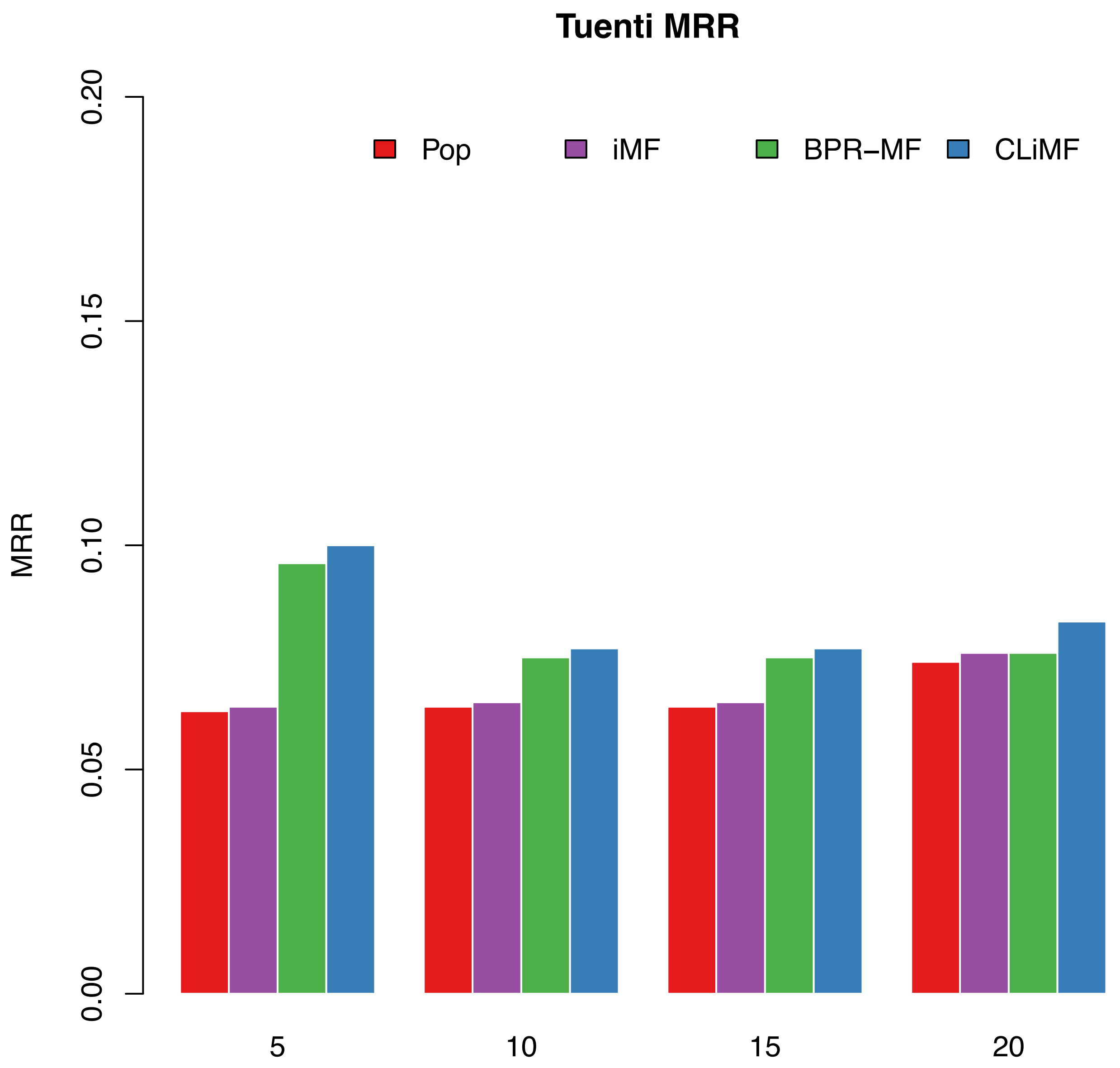
- Pop: Naive, recommend based on popularity of each item
- iMF (Hu and Koren, ICDM' 08): Optimizes Squared error loss
- BPR-MF (Rendle et al., UAI' 09): Optimizes AUC

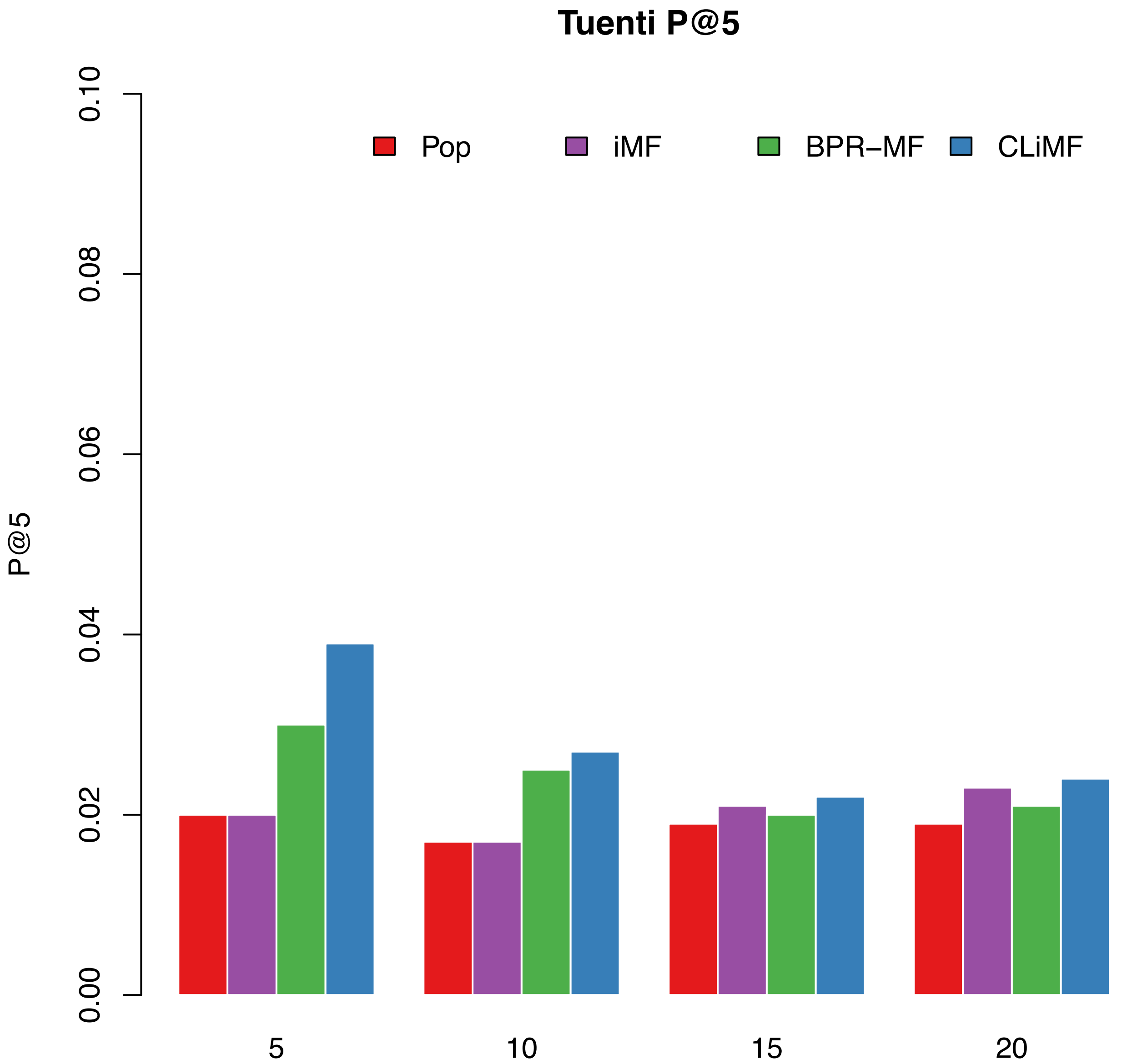

Epinions MRR
Pop
iMF
BPR−MF
CLiMF
MRR
0.0
0.1
0.2
0.3
0.4
5
10
15
20

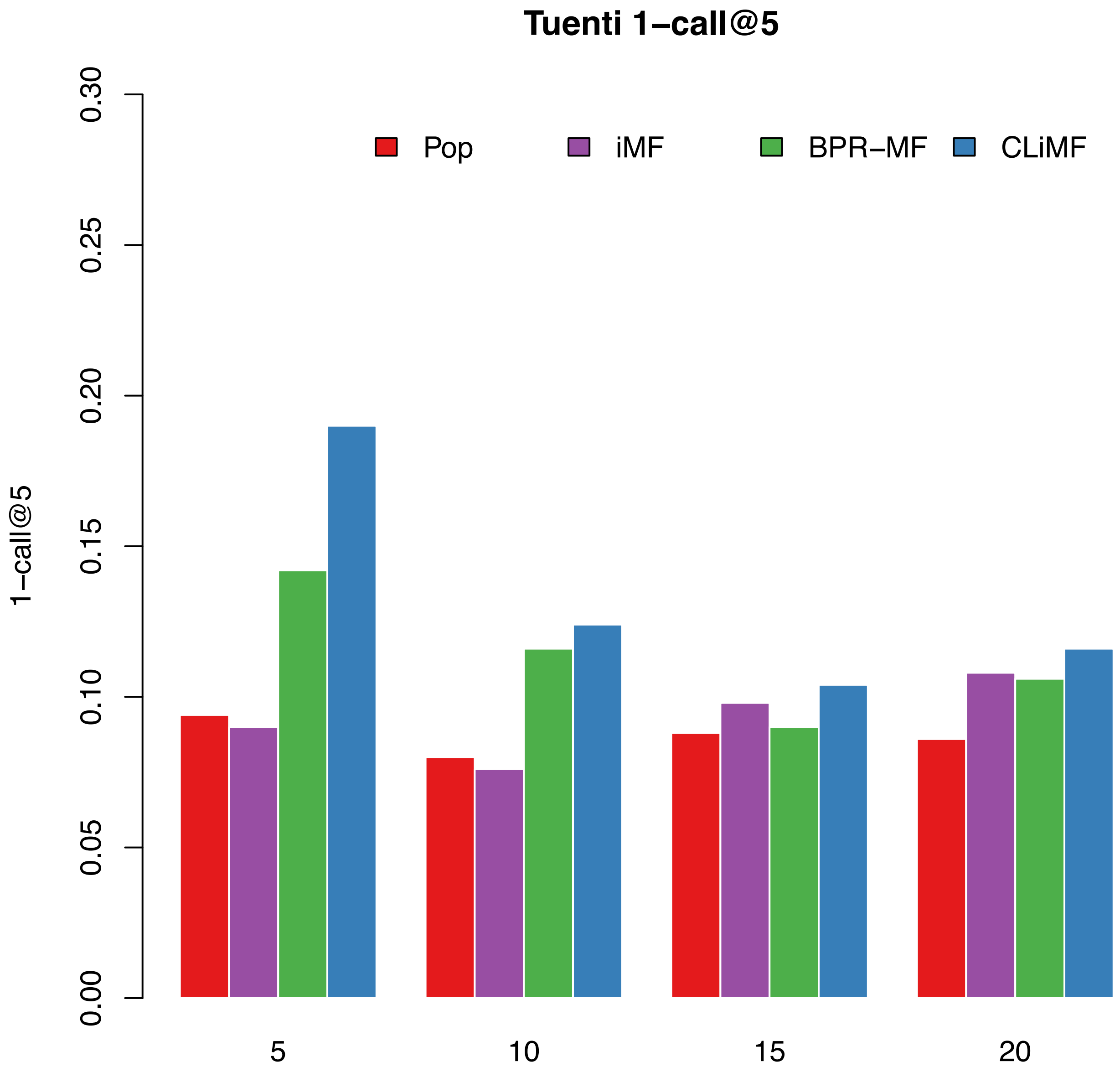

Epinions P@5
Pop
iMF
BPR−MF
CLiMF
P@5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
5
10
15
20

Epinions 1-call@5




Tuenti MRR
MRR
0.00
0.05
0.10
0.15
0.20
Pop
iMF
BPR–MF
CLiMF
5
10
15
20


Tuenti P@5
Pop
iMF
BPR−MF
CLiMF
P@5
0.00
0.02
0.04
0.06
0.08
0.10
5
10
15
20


Tuenti 1−call@5
Pop
iMF
BPR−MF
CLiMF
1−call@5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
5
10
15
20

Conclusions and Future Work

- Contribution
  - Novel method for implicit data with some nice properties (Top-k, speed)
- Future work
  - Use CLiMF to avoid duplicate or very similar recommendations in the top-k part of the list
  - To optimize other evaluation metrics for top-k recommendation
  - To take the social network of users into account

Thank you !

Telefonica Research is looking for interns!

Contact: alexk@tid.es or linas@tid.es