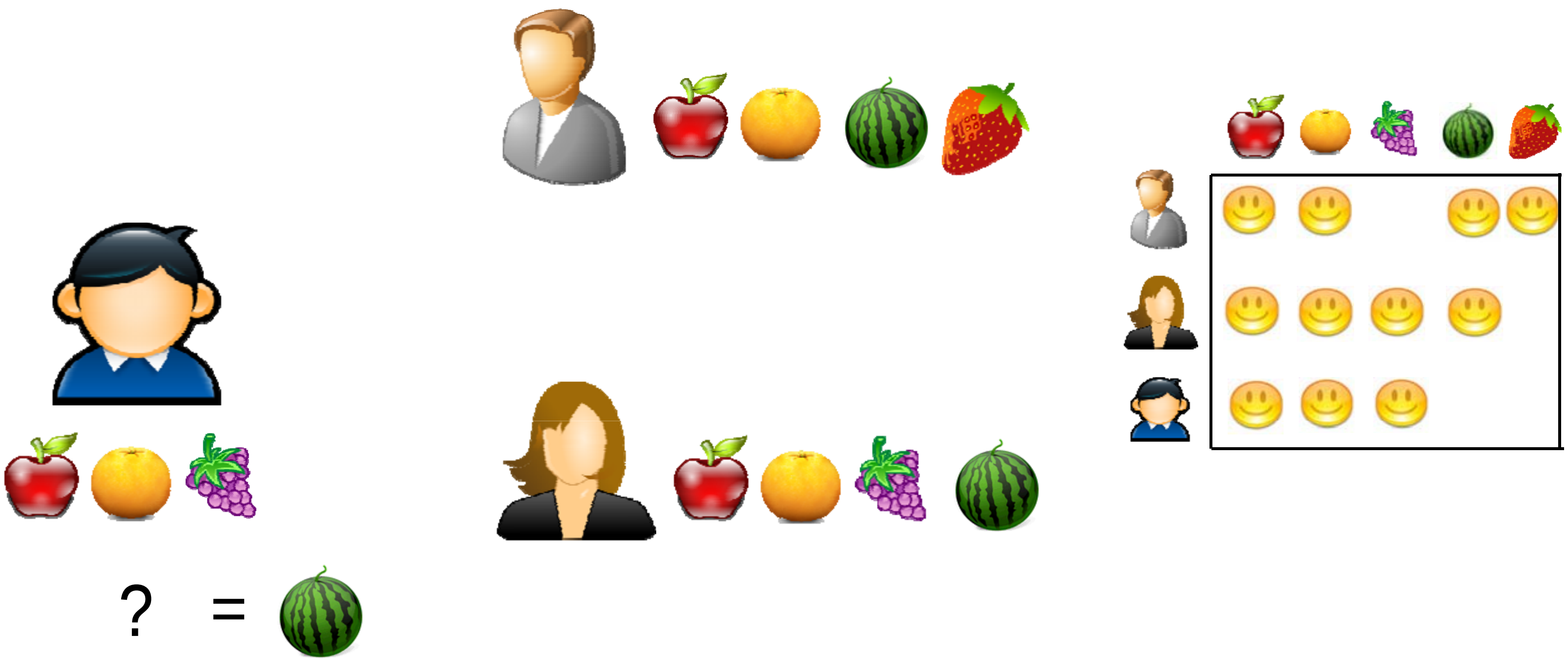
TFMAP: Optimizing MAP for Top-N Context-aware Recommendation

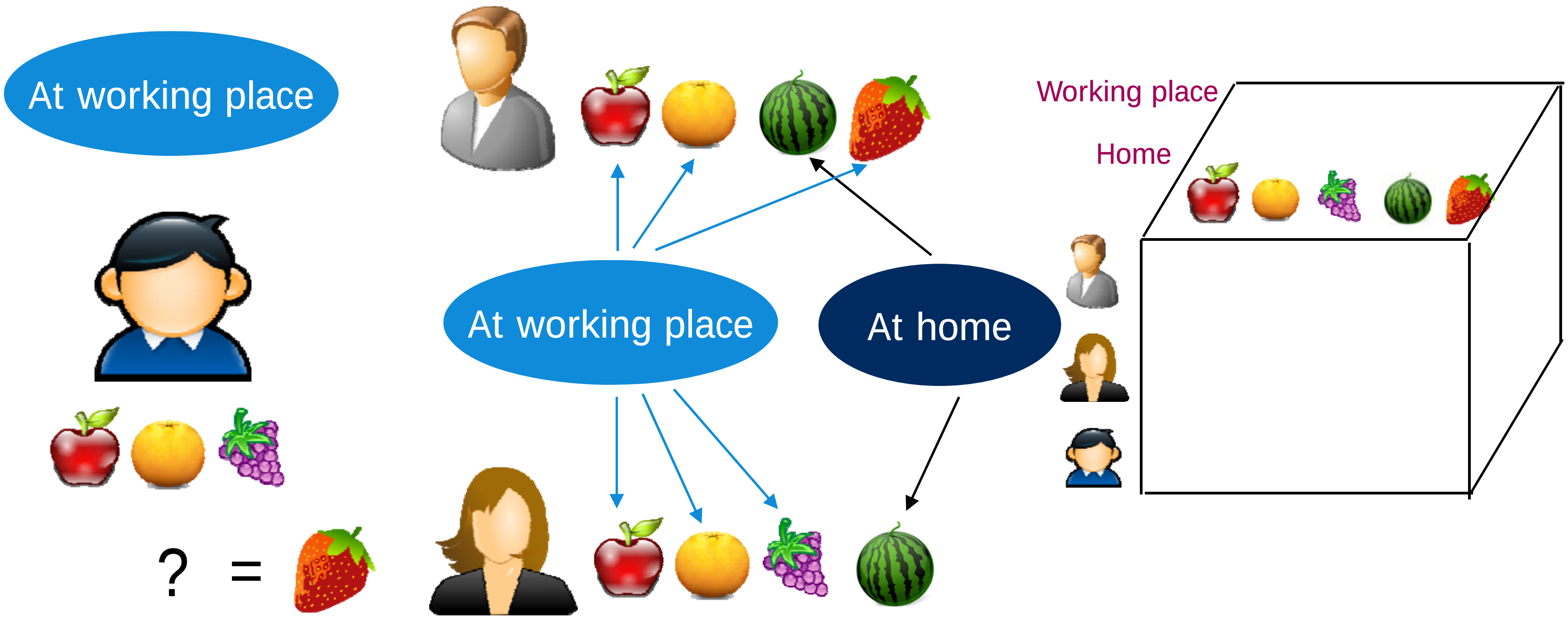
Yue Shi[a] (Presenter), Alexandros Karatzoglou[b], Linas Baltrunas[b], Martha Larson[a], Alan Hanjalic[a], Nuria Oliver[b]
[a]Delft University of Technology, Netherlands
[b]Telefonica Research, Spain

Introduction to Collaborative Filtering



Recommending based on the target user's past behavior and other users' interest

Motivation



Not only personalized, but also context-aware

Motivation

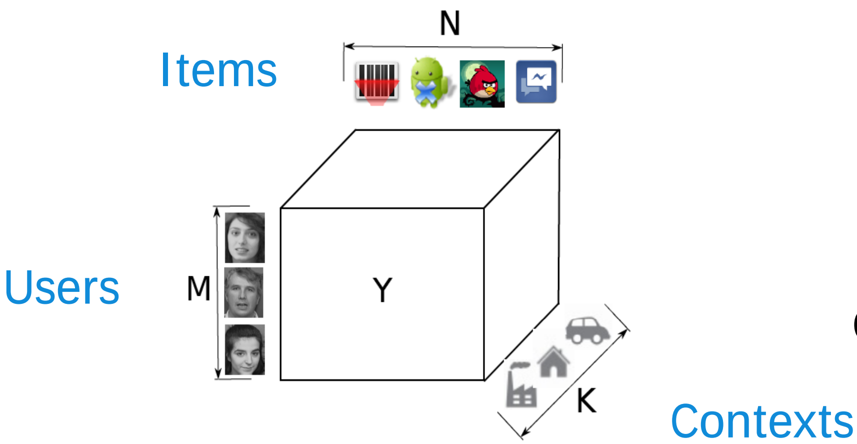
Not only context-aware, but also suitable for implicit feedback data

What is New!

- First work on context-aware recommendation for implicit feedback domains
- Taking MAP optimization from learning-to-rank to recommendation models with a new fast learning algorithm

Problem

- Given: Users' implicit feedback on items under different contexts
- Target: To recommend a list of items to each user under any given context, as accurate as possible



Top-N recommendation

Context-aware

Optimal in terms of a ranking measure

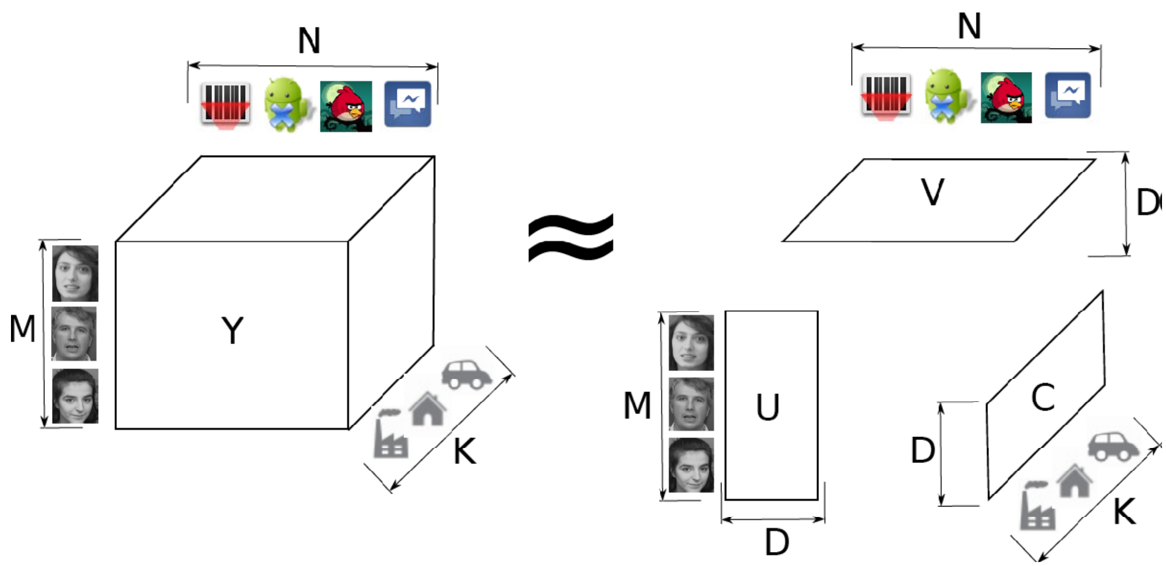
Challenges

- How to incorporate contextual information?
  - A tensor factorization model
- What to optimize for training the recommendation model? And How?
  - MAP capturing the quality of recommendation list based on implicit feedback data
  - but ***MAP is non-smooth, thus not able to be directly optimized***
    - *A smoothed version of MAP*
- How to ensure the proposed solution scalable?
  - A fast learning algorithm

How to incorporate contextual information?

- CP tensor factorization

$$f_{mik} = \langle U_m, V_i, C_k \rangle = \sum_{d=1}^{D} U_{md} V_{id} C_{kd}$$



U, V, C are latent factors (parameters to be learned)

U, V, C not optimized for y_{mik}, but for MAP

The Non-smoothness of MAP

- Average precision (AP) of a ranked list of items for a given user (user *m*) and a given context (context type *k*)

$$AP_{mk} = \frac{1}{\sum_{i=1}^{N} y_{mik}} \sum_{i=1}^{N} \frac{y_{mik}}{r_{mik}} \sum_{j=1}^{N} y_{mjk} \mathbb{I}(r_{mjk} \leq r_{mik})$$

Mobile app	y (Obs)	f (pred)	r (rank)
Angry birds	1	0.6	3
Draw something	0	0.8	2
Fruit ninja	0	0.2	4
ibook	0	0.1	5
DragonVale	1	0.9	1

- AP(y,r) non-smooth over model parameters
- MAP: Mean AP across users and contexts

Problem: r is a non-smooth function of f, thus, MAP non-smooth over model parameters

How to smooth MAP?

- Borrow techniques from learning-to-rank:

$$\mathbb{I}(r_{mjk} \le r_{mik}) \approx g(f_{mjk} - f_{mik}) = g(\langle U_m, V_j - V_i, C_k \rangle)$$

$$\frac{1}{r_{mik}} \approx g(f_{mik}) = g(\langle U_m, V_i, C_k \rangle)$$

- Smoothed MAP:

$$MAP \approx L(f, Y) = L(U, V, C, Y)$$

Smooth over *U*, *V* and *C*

- Updating *U, V, C* by gradient-based method to optimize MAP
- Theoretically, optimal *U, V, C* can be obtained.

Complexity issue

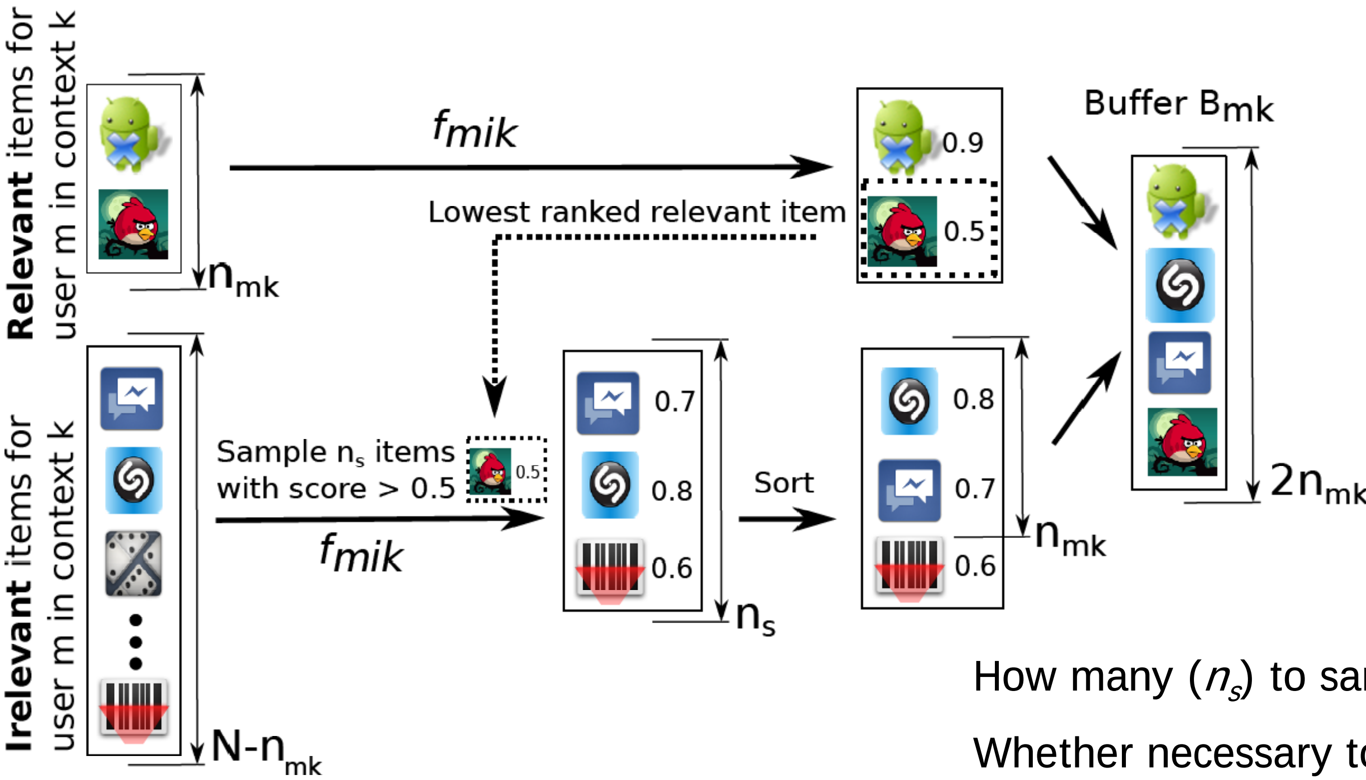
$$L(U,V,C) = \sum_{m=1}^{M} \sum_{k=1}^{K} \frac{1}{\sum_{i=1}^{N} y_{mik}} \sum_{i=1}^{N} y_{mik} g(\langle U_m, V_i, C_k \rangle) \times \sum_{j=1}^{N} y_{mjk} g(\langle U_m, V_j - V_i, C_k \rangle) - \frac{\lambda}{2}(\|U\|^2 + \|V\|^2 + \|C\|^2)$$

- Updating U and C: $\frac{\partial L}{\partial U}$ and $\frac{\partial L}{\partial C}$
  - Linear to the number of observations in the tensor data Y
- Updating V: $\frac{\partial L}{\partial V}$
  - Quadratic to the number of items!
  - Not scalable in the case of large number of items!

How to ensure scalability?

- Fast learning
  - Per combination of user *m* and context *k*, update *V* of a set of representative items (Buffer)
    - Relevant items
    - Top-ranked irrelevant items
  - Using an AP property
    - Updating positions of items that are ranked below the lowest ranked relevant item would not improve AP

Fast Learning



How many (n_s) to sample?

Whether necessary to select representative irrelevant items?

How beneficial from using the lowest ranked item?

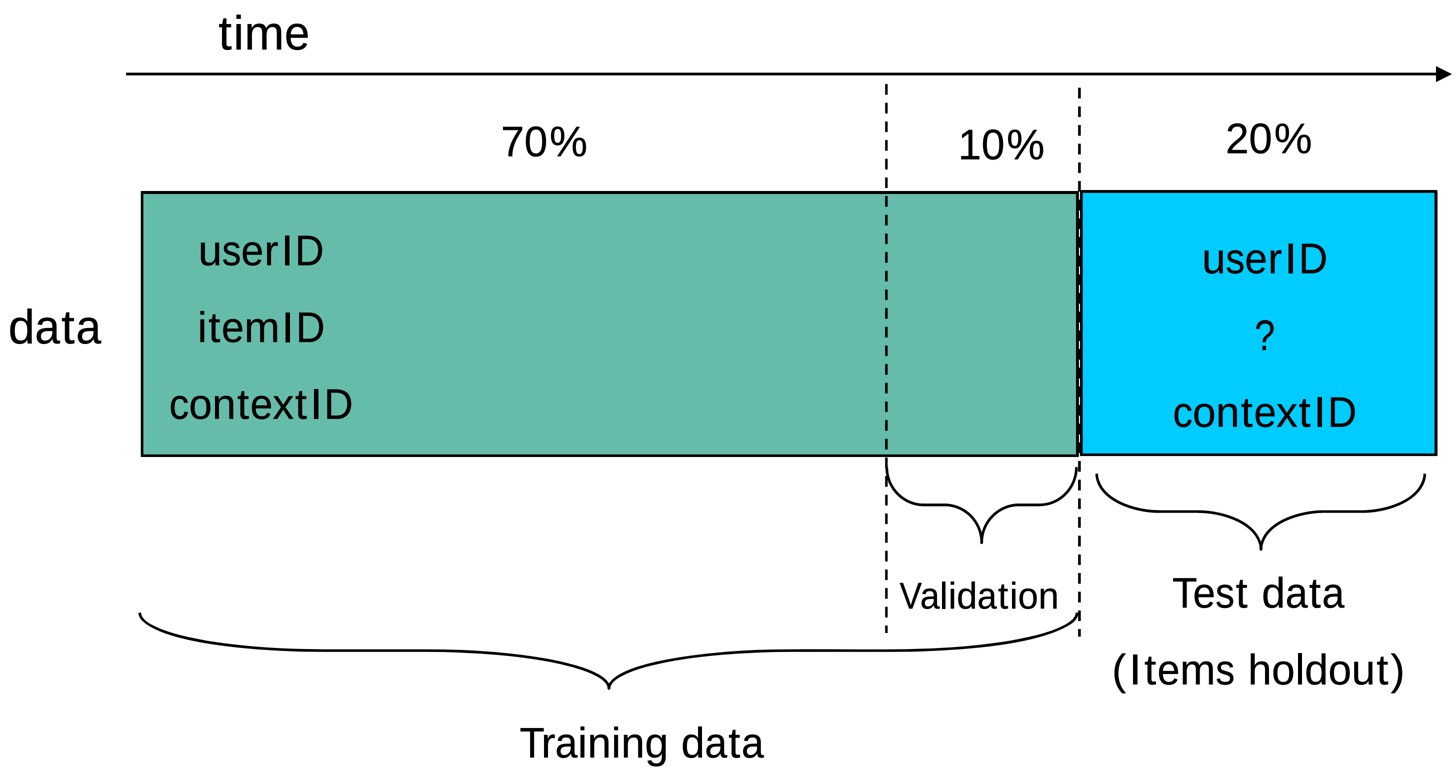
Experimental Evaluation

Data sets

- Appazaar (Main):
  - 300K observations of implicit feedback
  - 1767 users; 7701 mobile apps/items; 9 context types
  - Context defined by motion speed (3 possible states) and location (3 possible states)
  - < benchmarking datasets; but > other datasets in context-aware recommendation

Experimental Evaluation

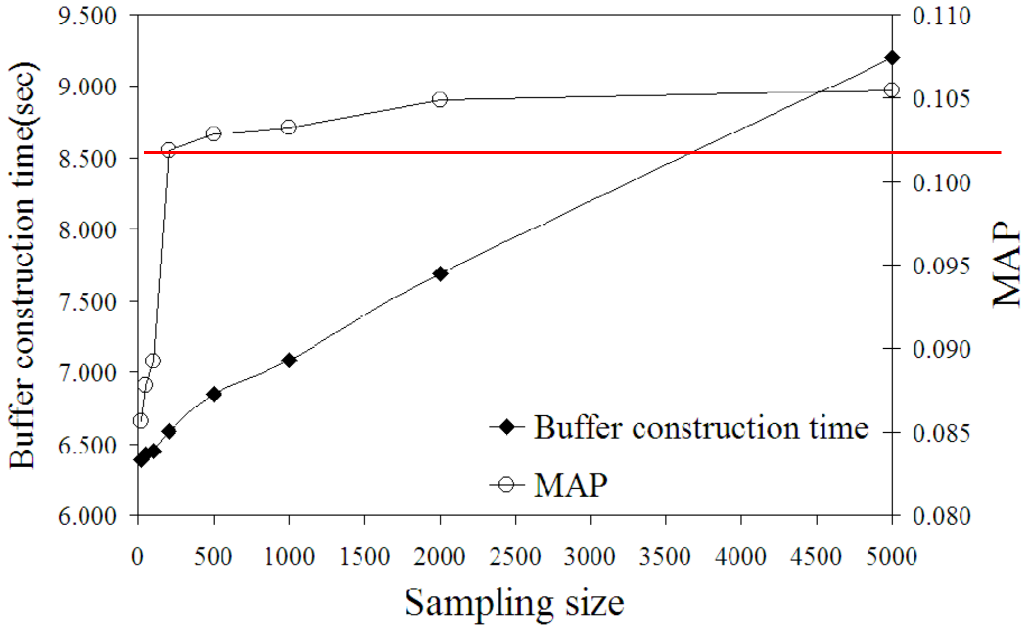
Experimental Protocol



Evaluation metrics: MAP, Precision@N

Experimental Evaluation

Impact of Fast Learning (I)



A small sample size is enough

Sampling size: 200

Rep. irrel. items

MAP: 0.102

vs

Sampling size: 200

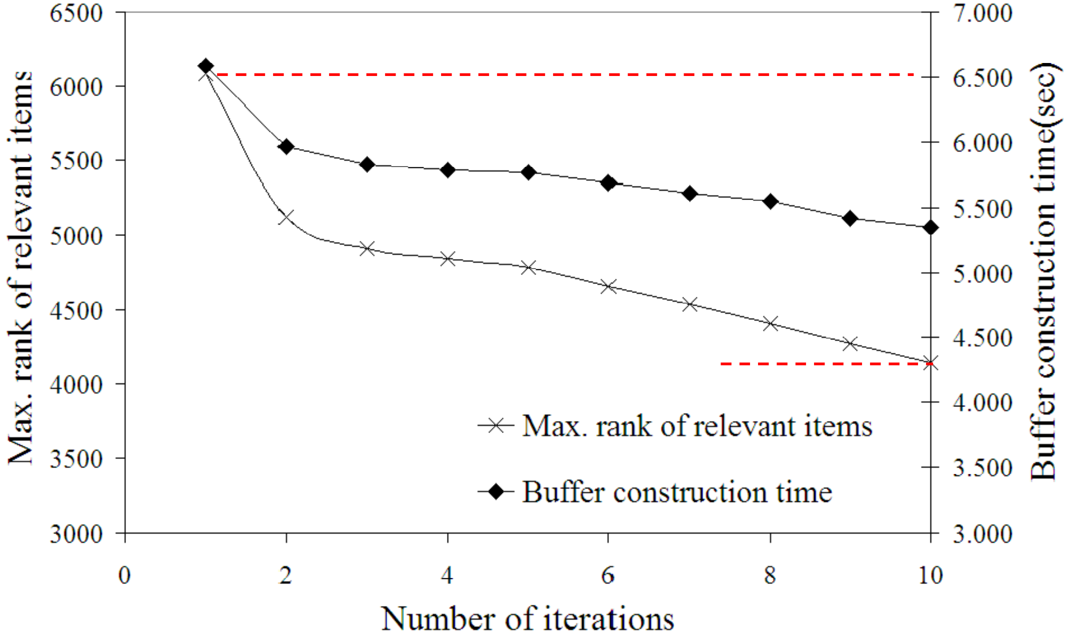
Random items

MAP: 0.083 (-18%)

Benefit from rep. irrel. items

Experimental Evaluation

Impact of Fast Learning (II)

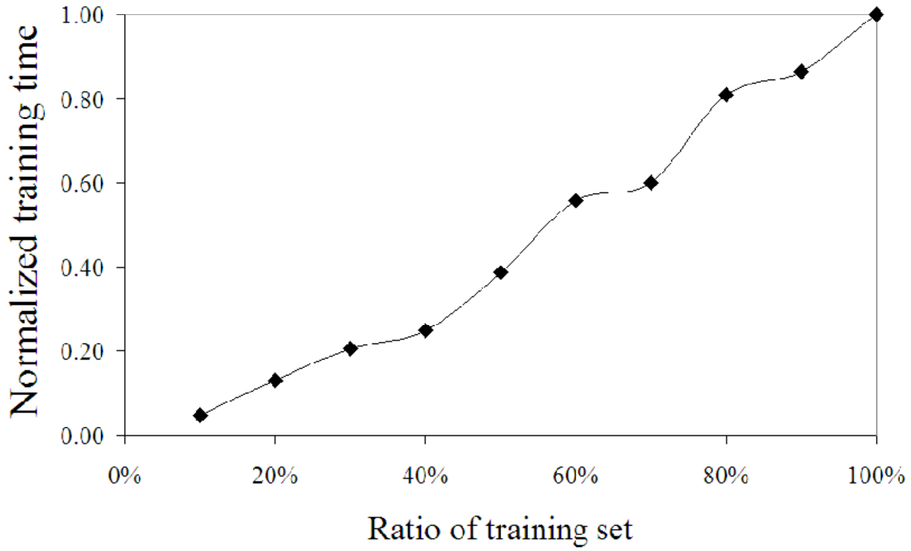


A side benefit from AP property

Using lowest-ranked relevant item help to improve the quality of rep. irrel. items, and also reduce buffer construction time

Experimental Evaluation

Impact of Fast Learning (III)



Training time per iteration at different scales of training set

Empirically validate the linear complexity of the fast learning algorithm

Experimental Evaluation

Performance

- Context-free baselines (Appazaar)
  - Pop: Naive, the popularity of each item under a given context
  - iMF (Hu and Koren, ICDM'08): SotA, no context
  - BPR-MF (Rendle et al., UAI'09): SotA, no context
  - TFMAP-noC: Variant of TFMAP, no context

Performance comparison between TFMAP and context-free baselines

	MAP	P@1	P@5	P@10
Pop	0.090	0.312	0.292	0.227
iMF	0.577	0.698	0.642	0.583
BPR-MF	0.612	0.800	0.712	0.602
TFMAP-noC	0.629	0.834	0.720	0.602
TFMAP	**0.659**	**0.879**	**0.732**	**0.611**

TFMAP-noC outperforms all the other baselines significantly. (Opt. MAP!)

TFMAP introduces another 5% improvement over TFMAP-noC. (Use context!)

Conclusions and Future Work

- Our contribution
  - First work on context-aware recommendation for implicit feedback domains
  - Propose a new recommendation model that directly optimizes MAP
  - Succeed in addressing the scalability issue of the proposed model
- Future work
  - To optimize other evaluation metrics for top-N recommendation (e.g., MRR, to appear in RecSys '12)
  - To take metadata of users and items into account

Questions & Answers

Thank you !

- Contact info: y.shi@tudelft.nl

We thank SIGIR for providing a travel grant for the first author.

Telefonica Research is looking for interns!

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