Local Item-Item Models for Top-N Recommendation

Evangelia Christakopoulou and George Karypis

Computer Science & Engineering

University of Minnesota, Twin Cities

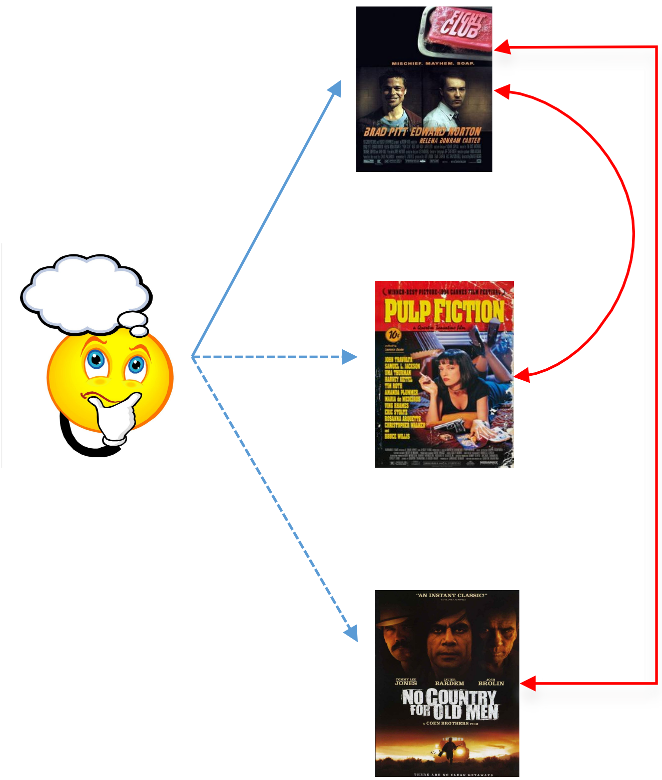
Overview

- Motivation
- Our Method
- Experimental Evaluation
- Experimental Results
- Conclusion

Motivation

Item-based Methods for Top-N Recommendation

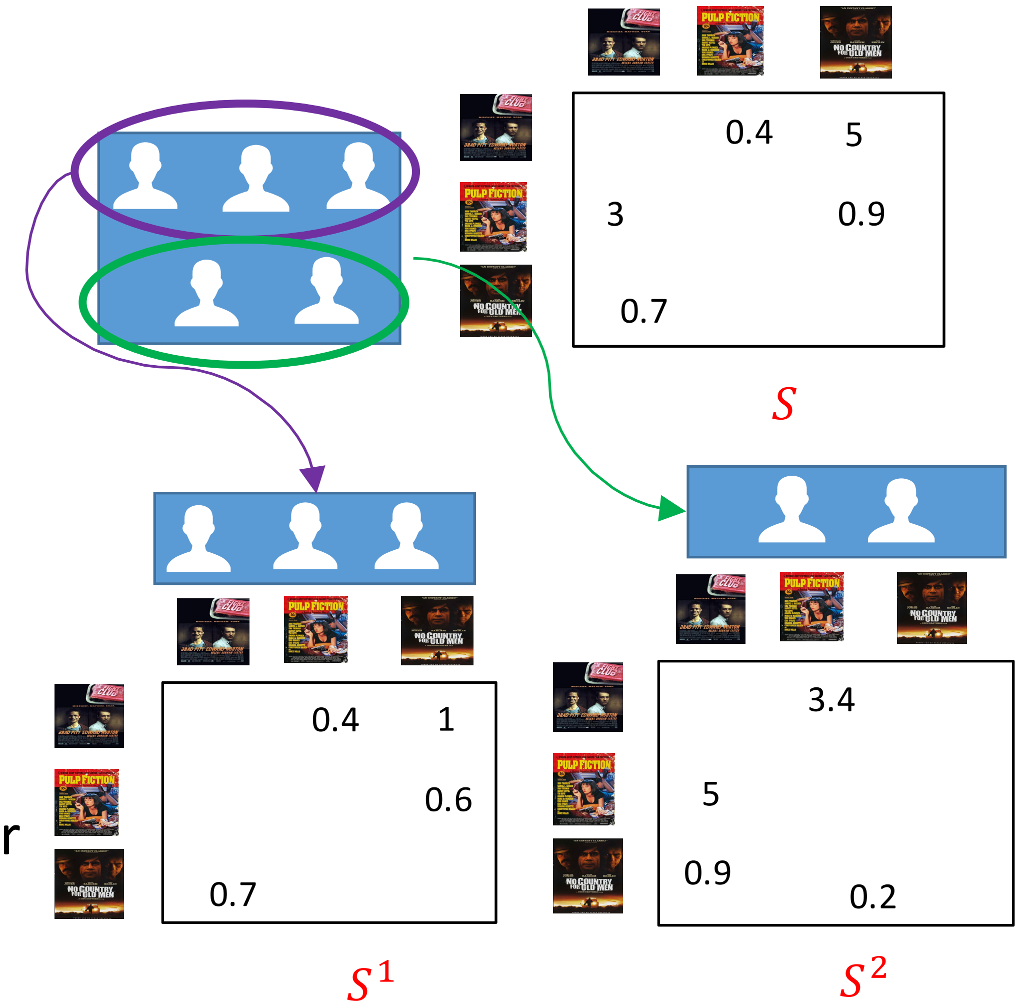
- The neighborhood methods identify similar users or items.
- The *item-based* are well-suited for the top-N recommendation task.
- Examples of item-based methods: k-NN and SLIM.



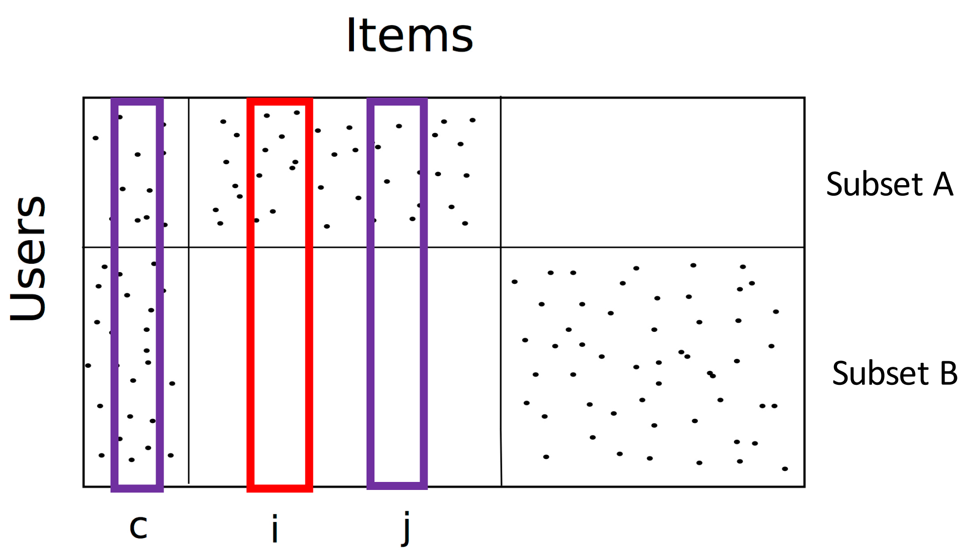
Limitation of the existing item-based approaches

Item-based methods have the drawback of estimating only a <u>*single*</u> model for all users.

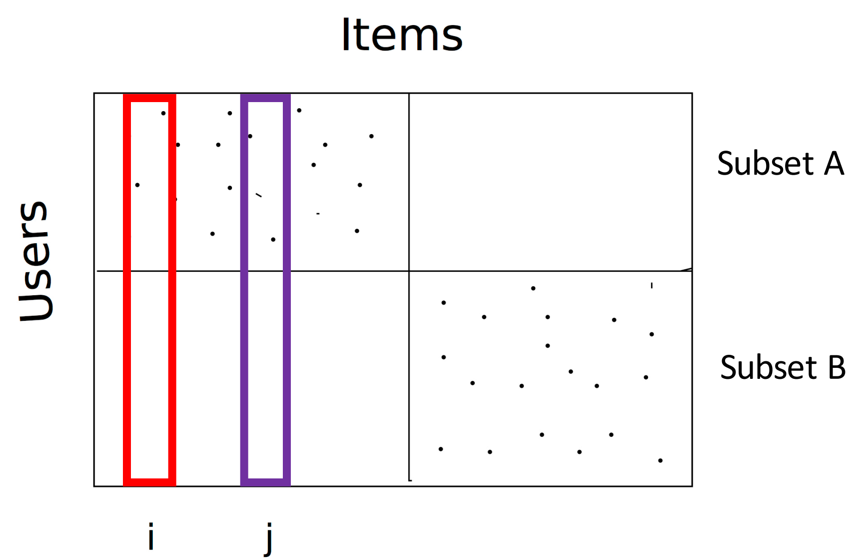
However, there could be differences in users' behaviors, which cannot be captured by a single model.
Instead, we need <u>*multiple*</u> item-item models, each for every user subset!



Example of when local item-item models are beneficial



Dataset a



Dataset b

Local item-item models improve upon global item-item model.

Global item-item model and local item-item models yield the same results.

i: item for which we will compute predictions

Sneak Preview

Our method is an item-item method that computes top-N recommendations by learning a global item-item model and user-subset specific item-item models and it automatically identifies the user subsets .

Our Method
GLSLIM

A few words on SLIM (Sparse LInear Method)

- Computes the item-item relations, by estimating an items $\times$ items sparse aggregation coefficient matrix S.
- The recommendation score of an unrated item i for user u is:

$$\hat{r}_{ui} = \mathbf{r}_u^T \mathbf{s}_i .$$

$$\begin{array}{ll} \underset{S}{\text{minimize}} & \frac{1}{2} \sum_{u,i} (r_{ui} - \hat{r}_{ui})^2 + \frac{\beta}{2} ||S||_F^2 + \lambda ||S||_1, \\ \text{subject to} & S \geq 0, \text{and} \\ & \text{diag}(S) = 0. \end{array}$$

GLSLIM model

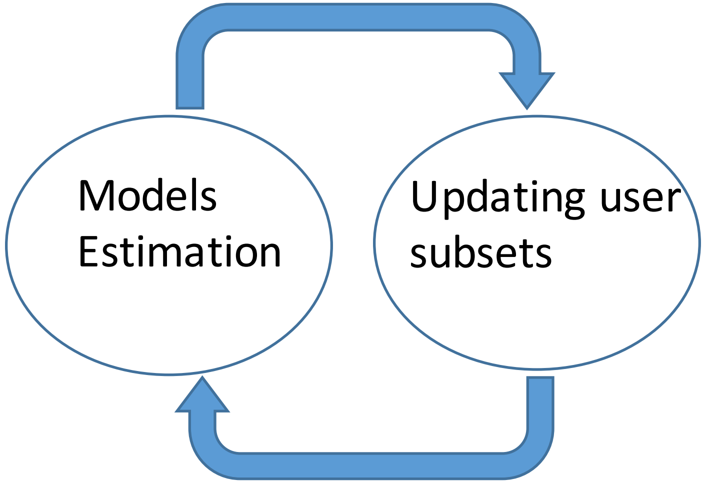
If user u belongs to user subset p_u, then the predicted rating is:

$$\hat{r}_{ui} = \mathbf{r}_u^T(\underbrace{\boxed{g_u \mathbf{s}_i}}_{\text{global}} + \underbrace{\boxed{(1 - g_u)\mathbf{s}_i^{p_u}}}_{\text{local}}).$$

$$\begin{aligned}
\underset{S,\{S^1,\ldots,S^k\},\mathbf{p},\mathbf{g}}{\text{minimize}} \quad & \tfrac{1}{2}\textstyle\sum_{u,i}(r_{u,i} - \hat{r}_{u,i})^2 + \\
& \underbrace{\boxed{\tfrac{1}{2}\beta_g||S||_F^2 + \lambda_g||S||_1}}_{\text{global}} + \underbrace{\boxed{\textstyle\sum_{p_u=1}^{k}\left[\tfrac{1}{2}\beta_l||S^{p_u}||_F^2 + \lambda_l||S^{p_u}||_1\right]}}_{\text{local}}, \\
\text{subject to} \quad & 0 \leq g_u \leq 1, \ \forall u \\
& p_u \in \{1, \ldots, k\}, \ \forall u \\
& S \geq 0, \ S^1 \geq 0, \ldots, \ S^k \geq 0 \\
& \mathrm{diag}(S) = 0, \ \mathrm{diag}(S^1) = 0, \ldots, \ \mathrm{diag}(S^k) = 0.
\end{aligned}$$

How the variables are estimated

We use Alternating Least Squares.



The models are *jointly* optimized with the user assignments and the personalized weight.

Experimental Evaluation

Datasets

Name	#Users	#Items	#Transactions	Density
groceries	63,034	15,846	2,060,719	0.21%
ml	69,878	10,677	10,000,054	1.34%
flixster	29,828	10,085	7,356,146	2.45%
netflix	274,036	17,770	31,756,784	0.65%

Evaluation Methodology

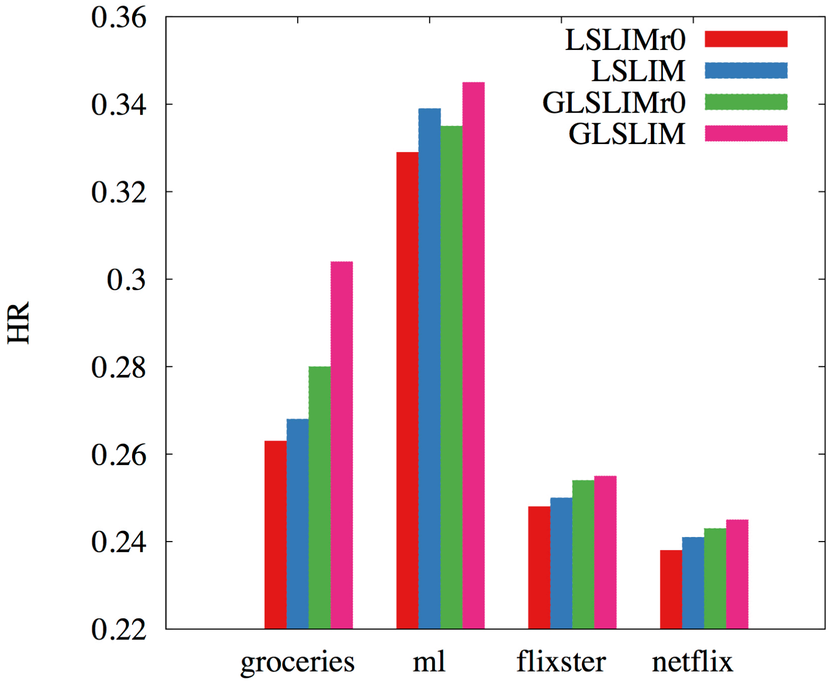
- Leave-one-out cross-validation.
- Quality measures: $HR = \frac{\#hits}{\#users}$ $ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{p_i}$
- Comparison algorithms: *PureSVD, BPR-MF, SLIM.*
- Extensive search over the parameter space.

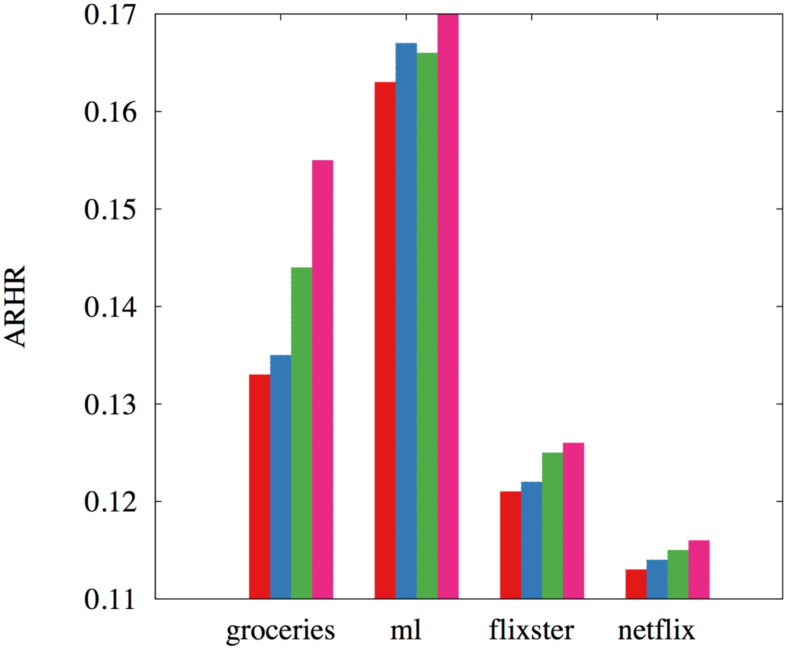
Proposed Methods

- LSLIMr0: Local SLIM without refinement.
- LSLIM: Local SLIM with refinement.
- GLSLIMr0: Global and Local SLIM without refinement.
- GLSLIM: Global and Local SLIM with refinement.

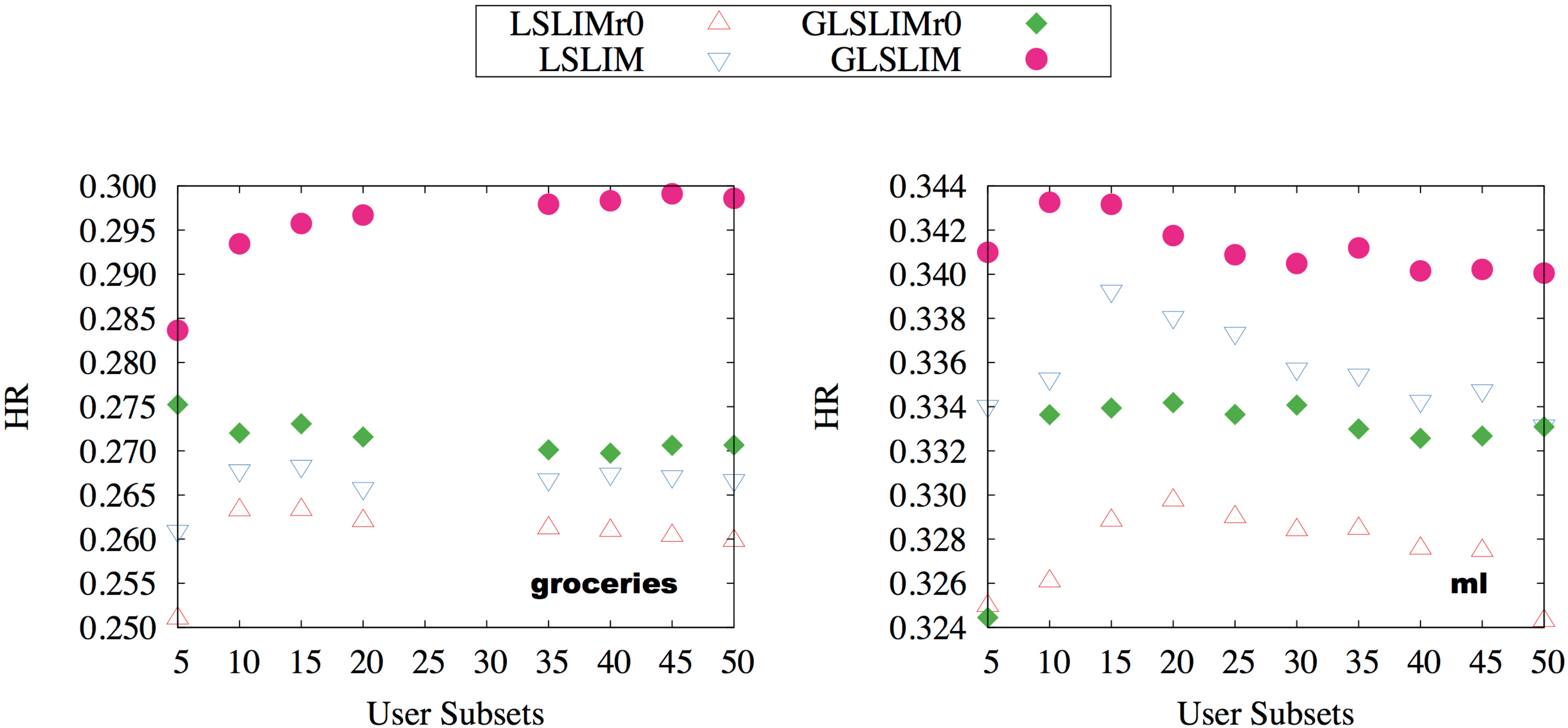
Experimental Results

Performance of the proposed methods

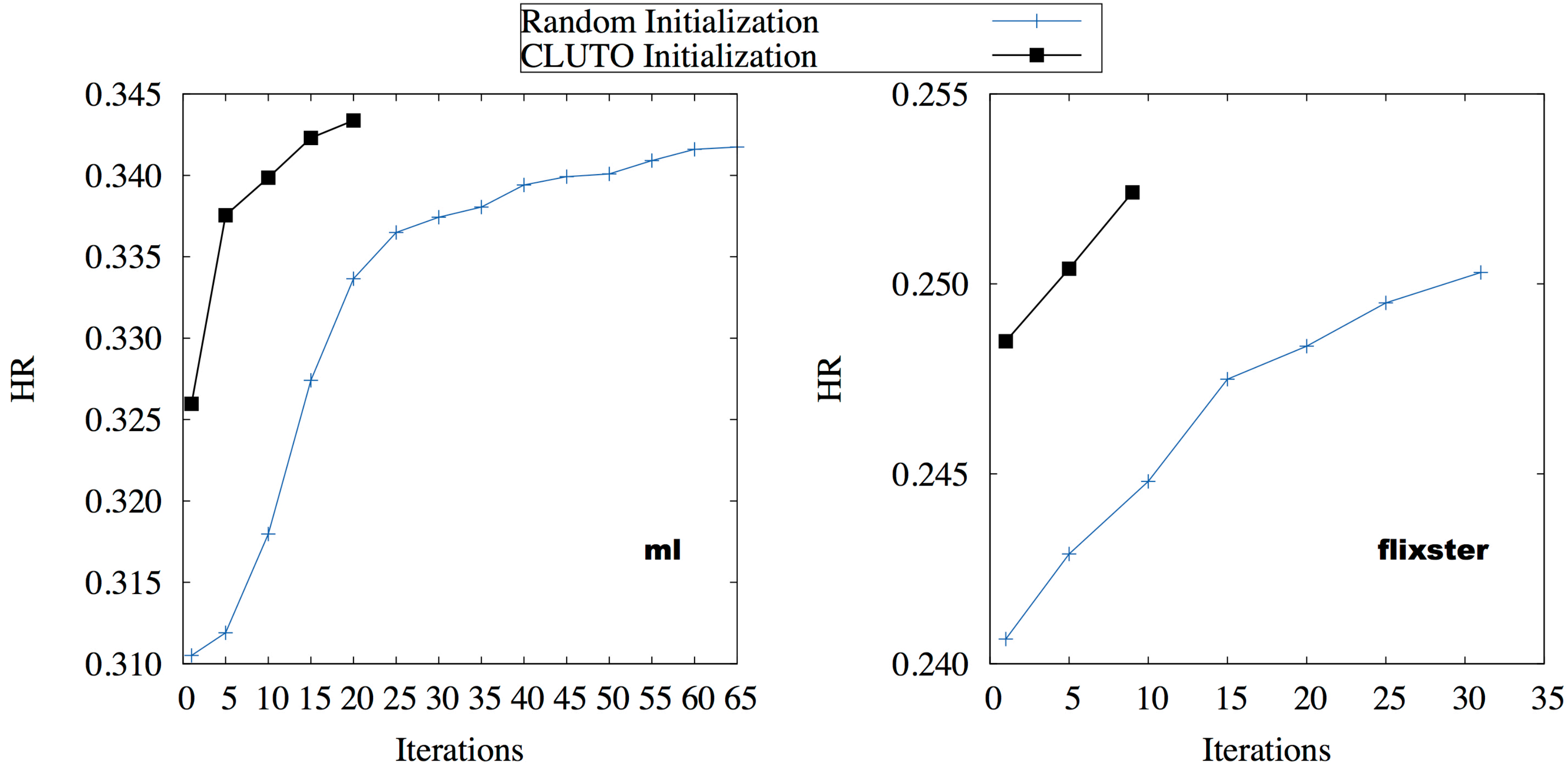




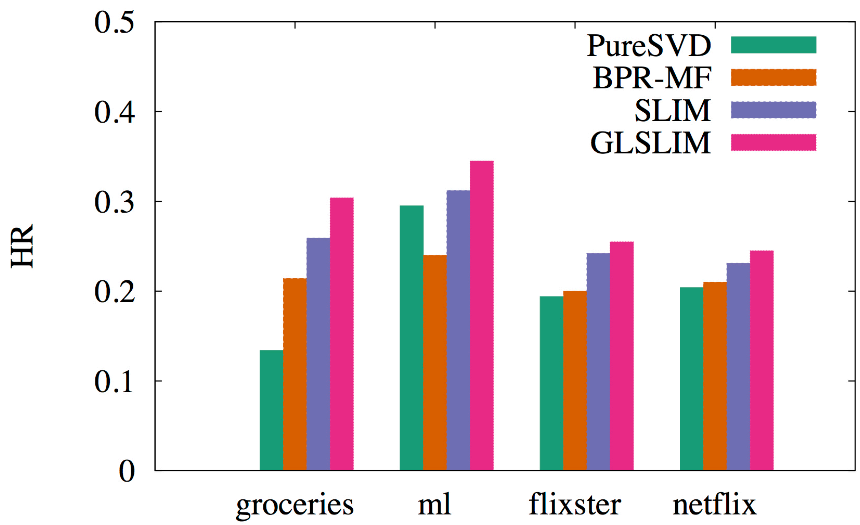
Sensitivity on the number of User Subsets

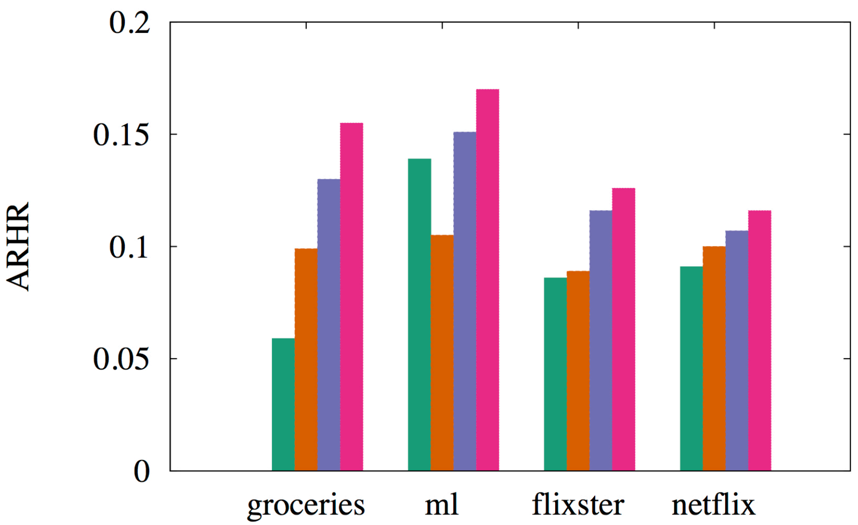


Initializing with Random User Subsets



Performance against Competing Approaches





Conclusion

Conclusion

- GLSLIM improves upon item-based schemes, by capturing the differences in the user preferences.
- Experiments show that GLSLIM outperforms competing top-N recommender methods.
- Using multiple item-item models is valuable!

Thank you!