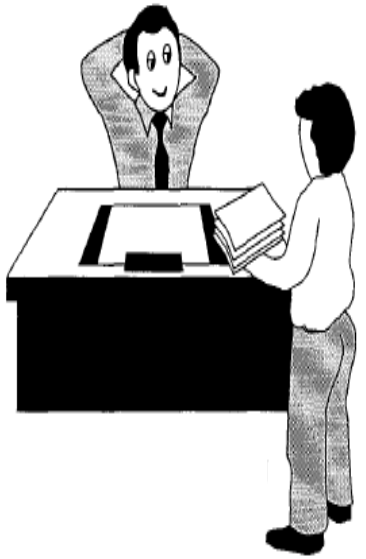
Collaborative Filtering Approaches for Recommendation Systems

Joonseok Lee

2013/11/15

Why recommendation?



Examples

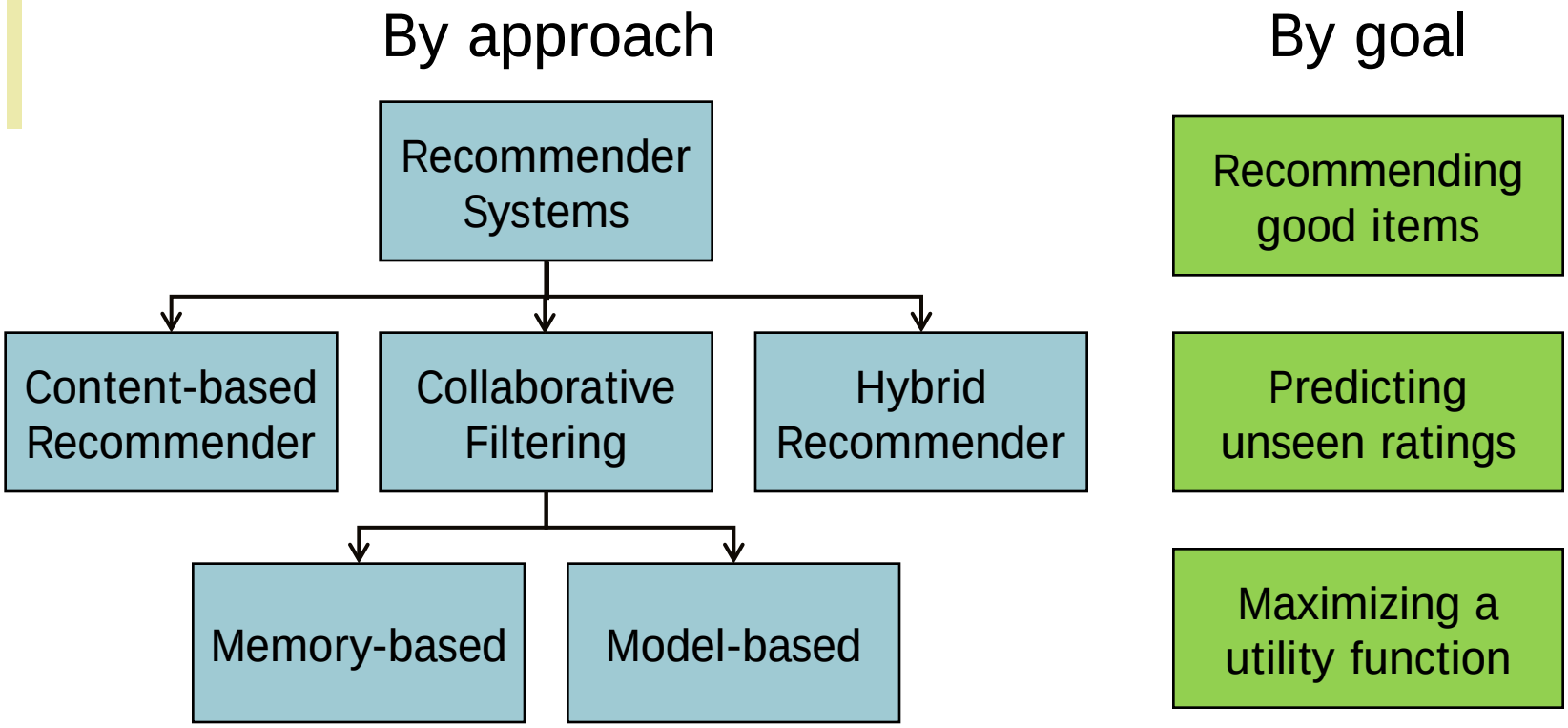
- Product recommendation

- Friend recommendation

- Rating prediction

- Personalized web search

Types of Recommender Systems



Contents-based Filtering

- Contents information
  - Demographic data about users
    - Age, gender, geographic location
  - Product attributes
    - Movie: genre, director, release year
    - Book: author, language, published year, genre

- Depends on domain
  - Domain-specific modeling is necessary.
  - Algorithm is also different domain by domain.
  - With abundant domain knowledge or data, can be very powerful.

Collaborative Filtering

- Make use of rating data from users only.
  - Direct feedback: rating,  , 
  - Indirect feedback: click through, page view

- Collaborative Filtering definition
  - People collaborate to help one another perform filtering, by recording their reactions to products they consumed.
  - Use other users' feedback to fit my preference!

- Independent of domain
  - Many models and algorithms can work regardless of the domain.

Matrix View

- Matrix completion problem: given a partially-observed noisy matrix M, we would like to approximately complete it.

- In recommendation systems,
  - $M_{u,i}$ is a rating on item i by user u.
  - Naturally sparse.
  - We want to estimate unrated items.

Items

Users

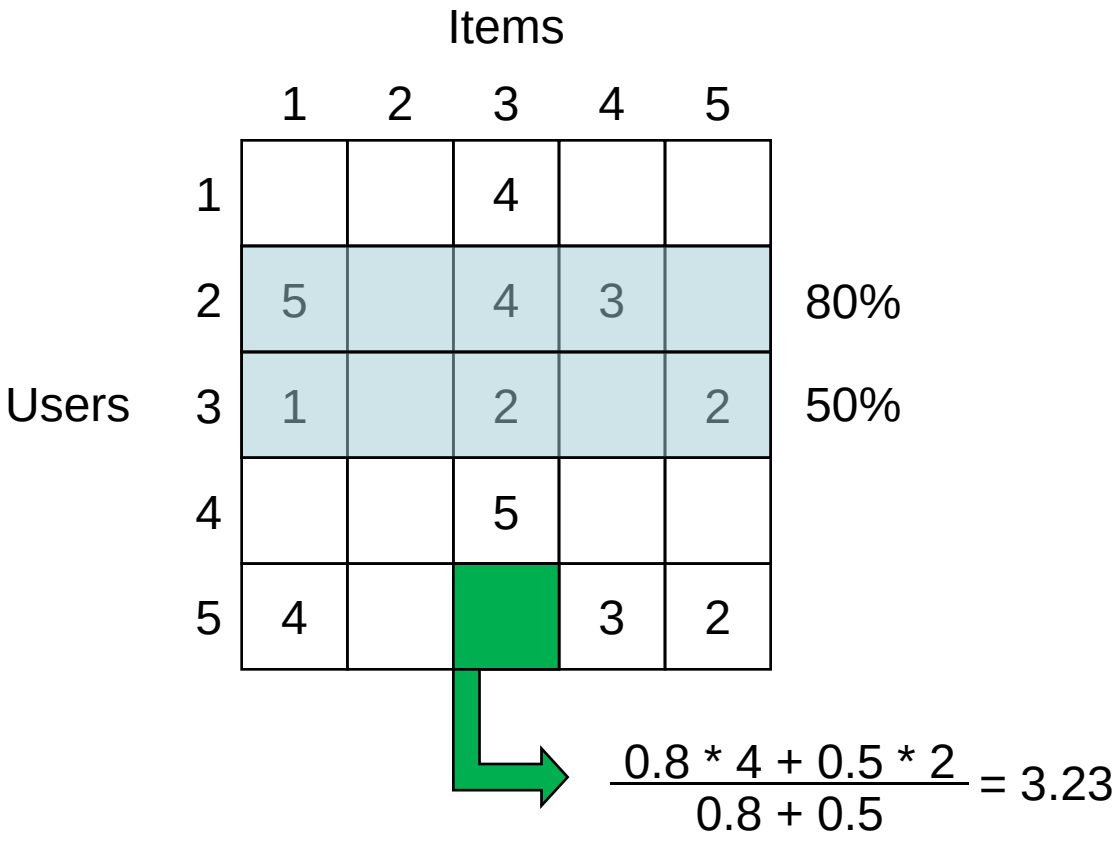
		3		
3				5
	5		4	
		2	5	
1			2	

User-based Collaborative Filtering

- Find preferred items by similar users to me.
  - S_u: a set of similar users to user u.
  - Trust other user's opinion proportional to the similarity between (s)he and I.

$$\hat{r}_{ui} = \frac{1}{\sum_{v \in S_u} sim(u, v)} \sum_{v \in S_u} sim(u, v)\, r_{vi}$$

User-based Collaborative Filtering



Item-based Collaborative Filtering

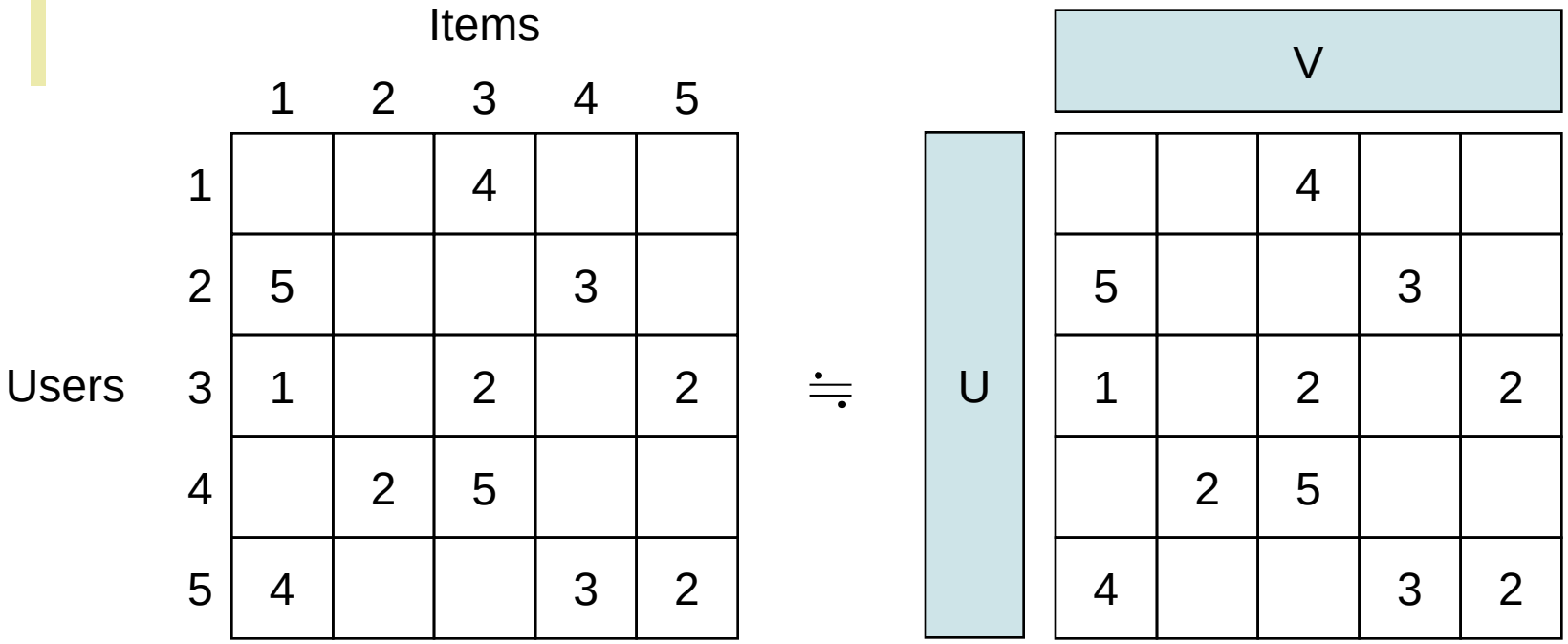
- Same way to user-based CF, but in column-wise.

- More powerful than user-based CF.
  - Item neighbors tend to be more stable than user neighbors.

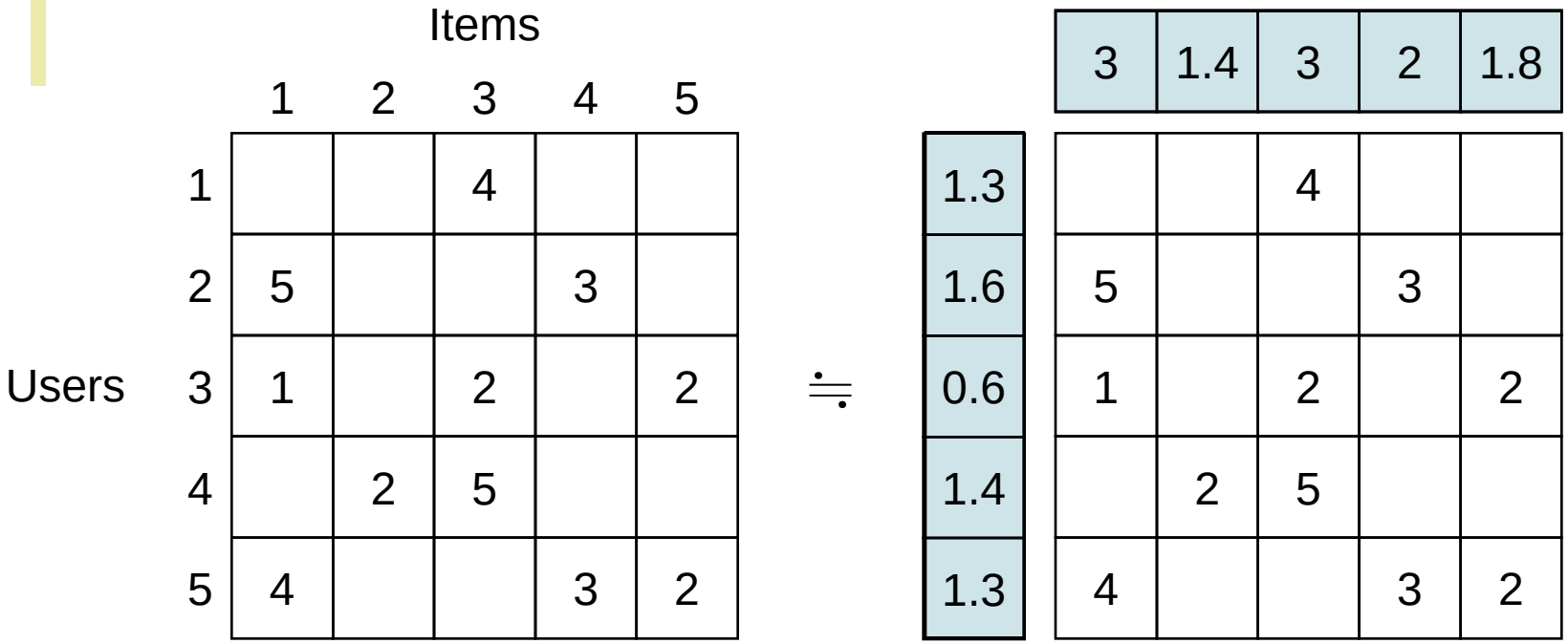
Memory-based CF Summary

- Pros
  - Simple, easy to implement.
  - Can explain reason of recommendation.
- Cons
  - Huge memory consumption.
  - Not scalable.

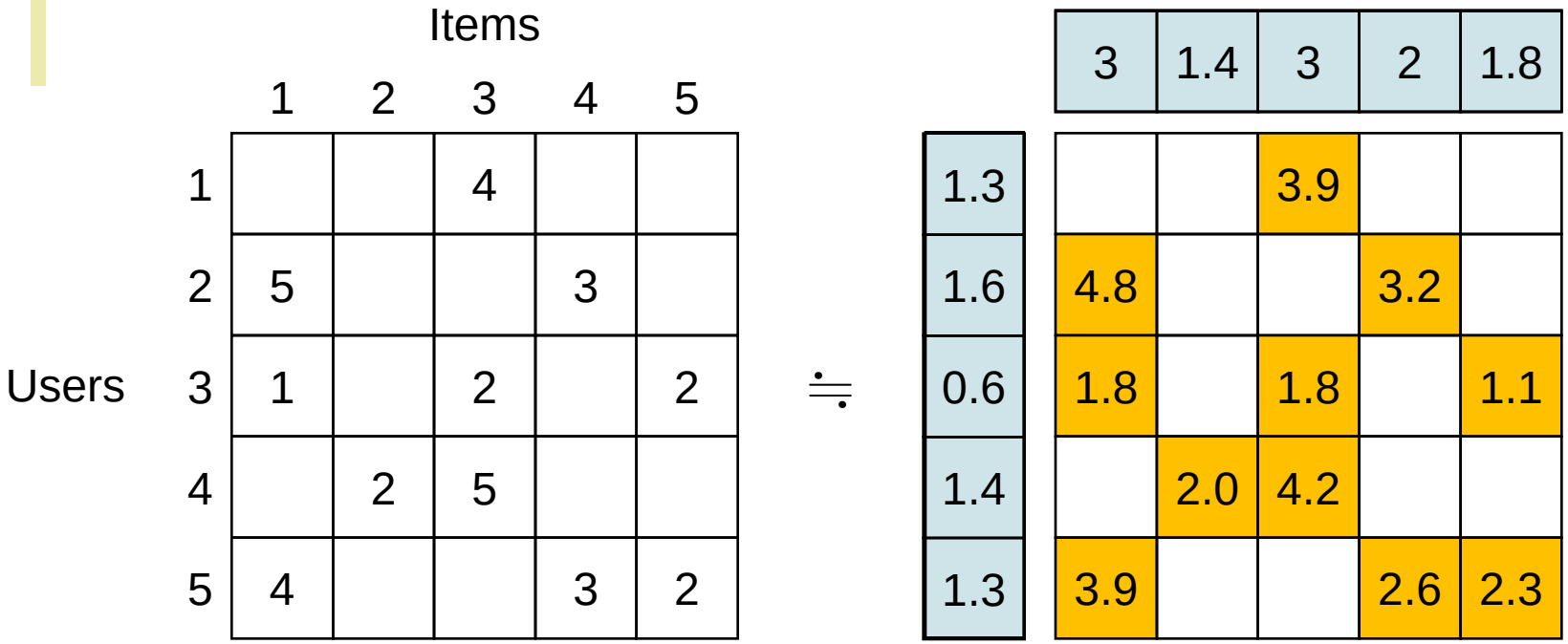
Matrix Factorization



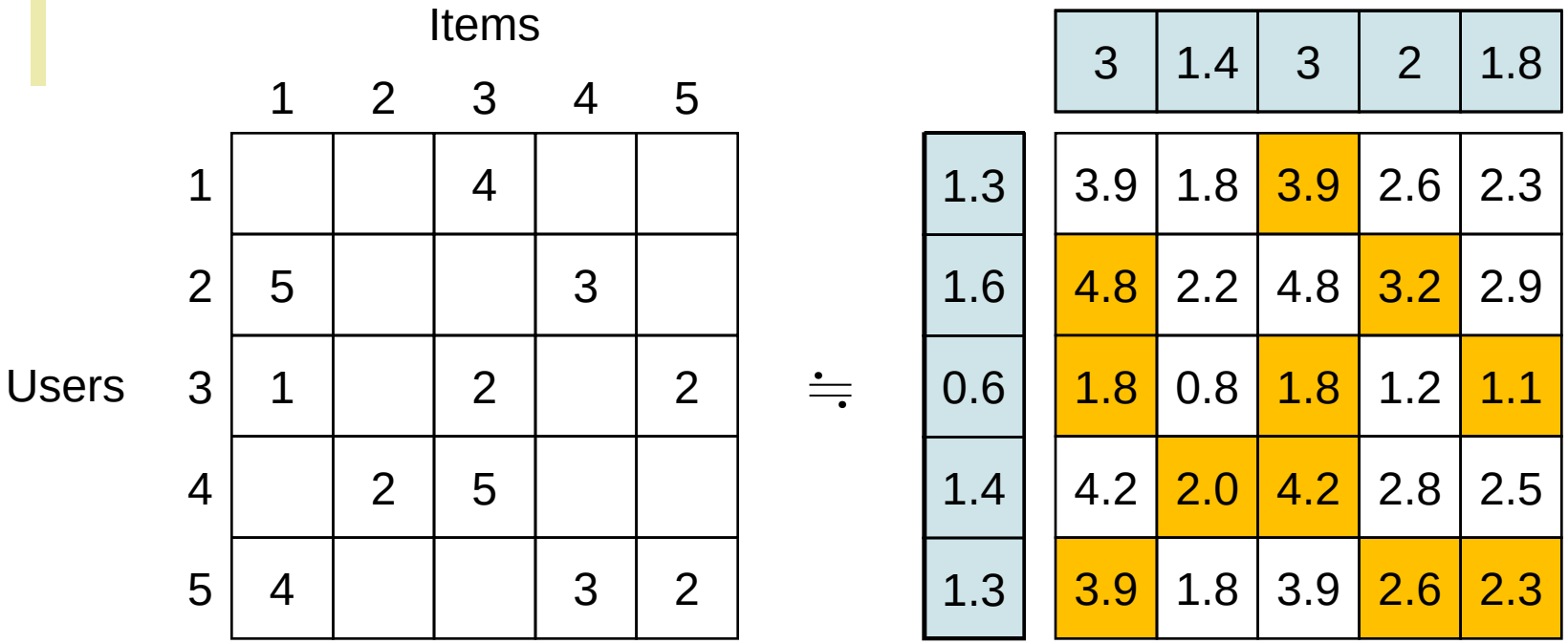
Matrix Factorization



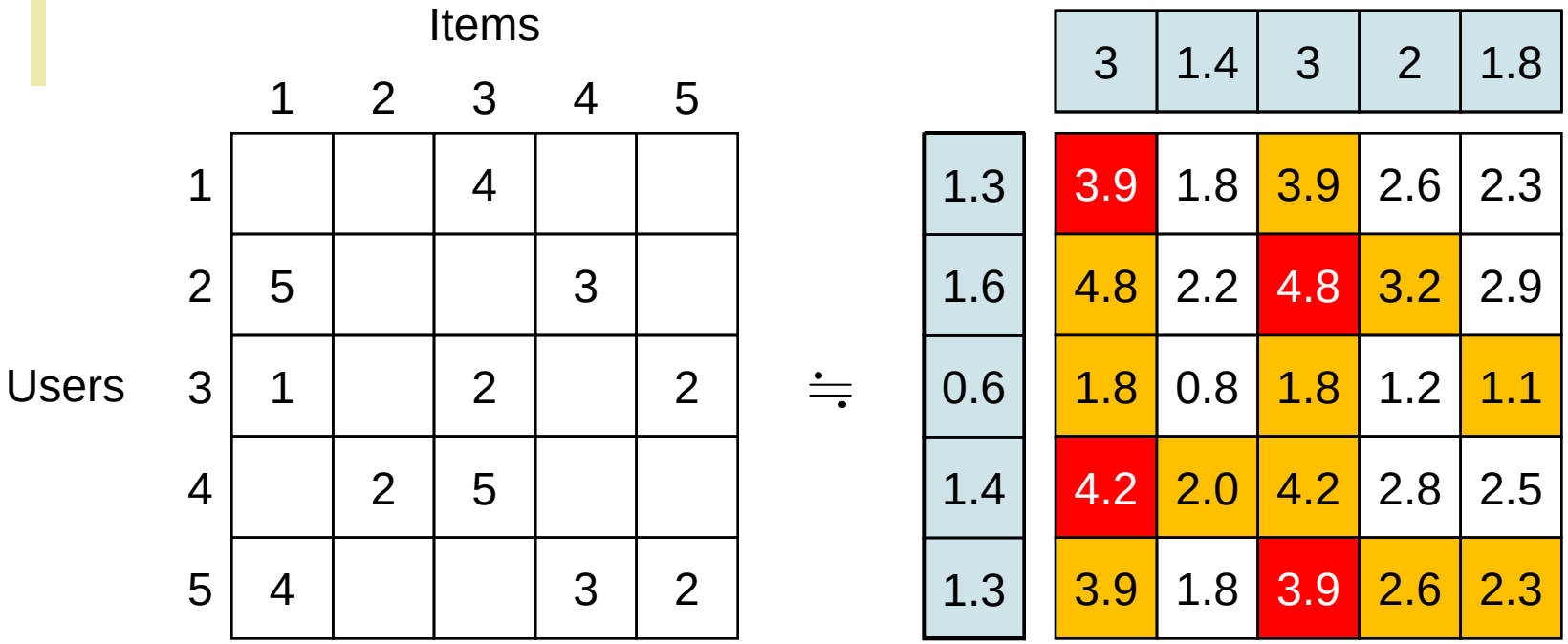
Matrix Factorization



Matrix Factorization



Matrix Factorization



Matrix Factorization Summary

- Pros
  - Prediction is (most) accurate.
- Cons
  - Computationally expensive.
  - Difficult to explain why we recommend.

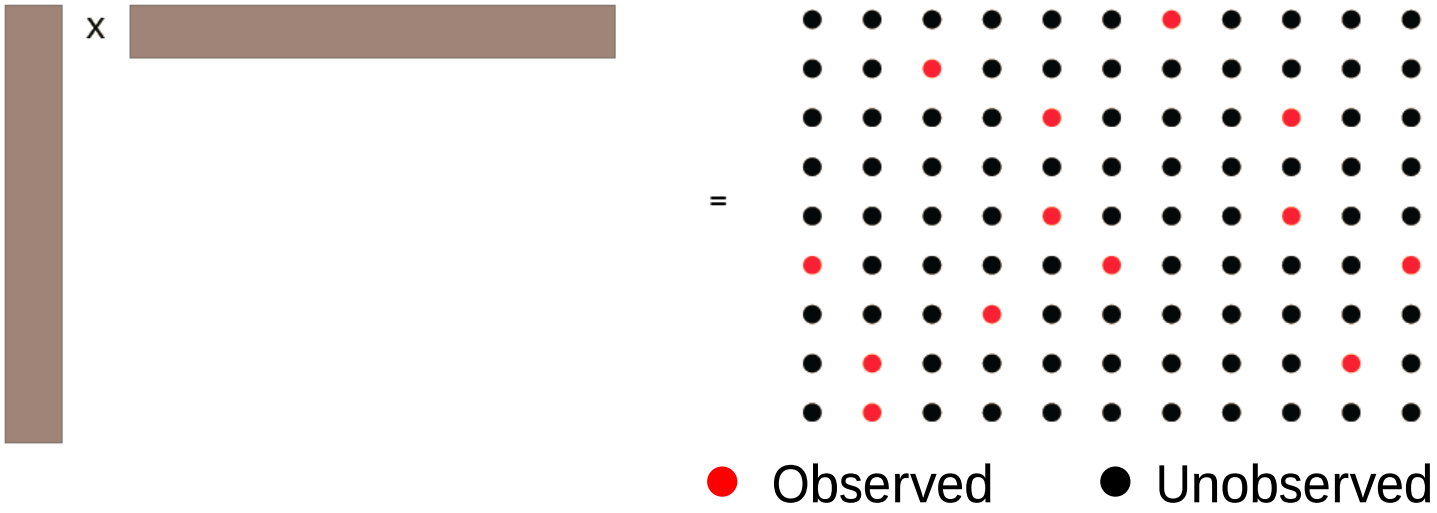
Incomplete SVD

- Common practice: low-rank assumption.

$$M \approx UV^T \in \mathbb{R}^{n_1 \times n_2}, \qquad U \in \mathbb{R}^{n_1 \times r}$$

$$V \in \mathbb{R}^{n_2 \times r}$$

$$r \ll \min(n_1, n_2)$$

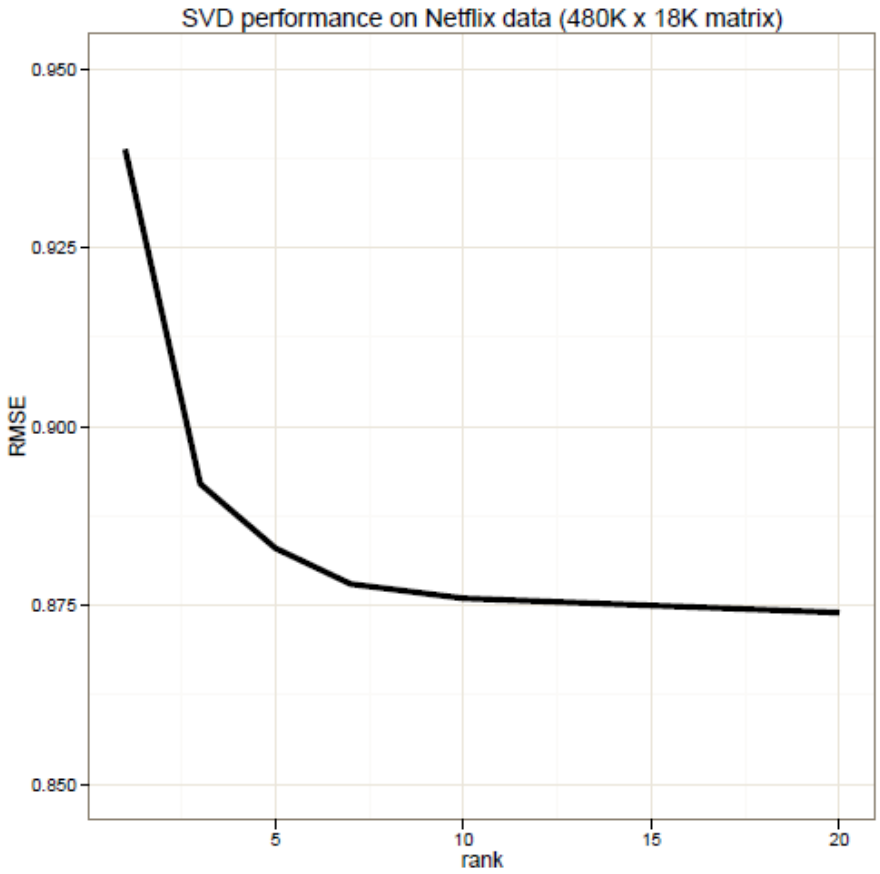


Incomplete SVD

- Minimizing Frobenius norm

$$\min_{U,V} \sum_{(u,i)\in A} \left([UV^T]_{u,i} - M_{u,i}\right)^2$$

- Diminishing returns:
  - Small decrease in approximated loss as rank increases.



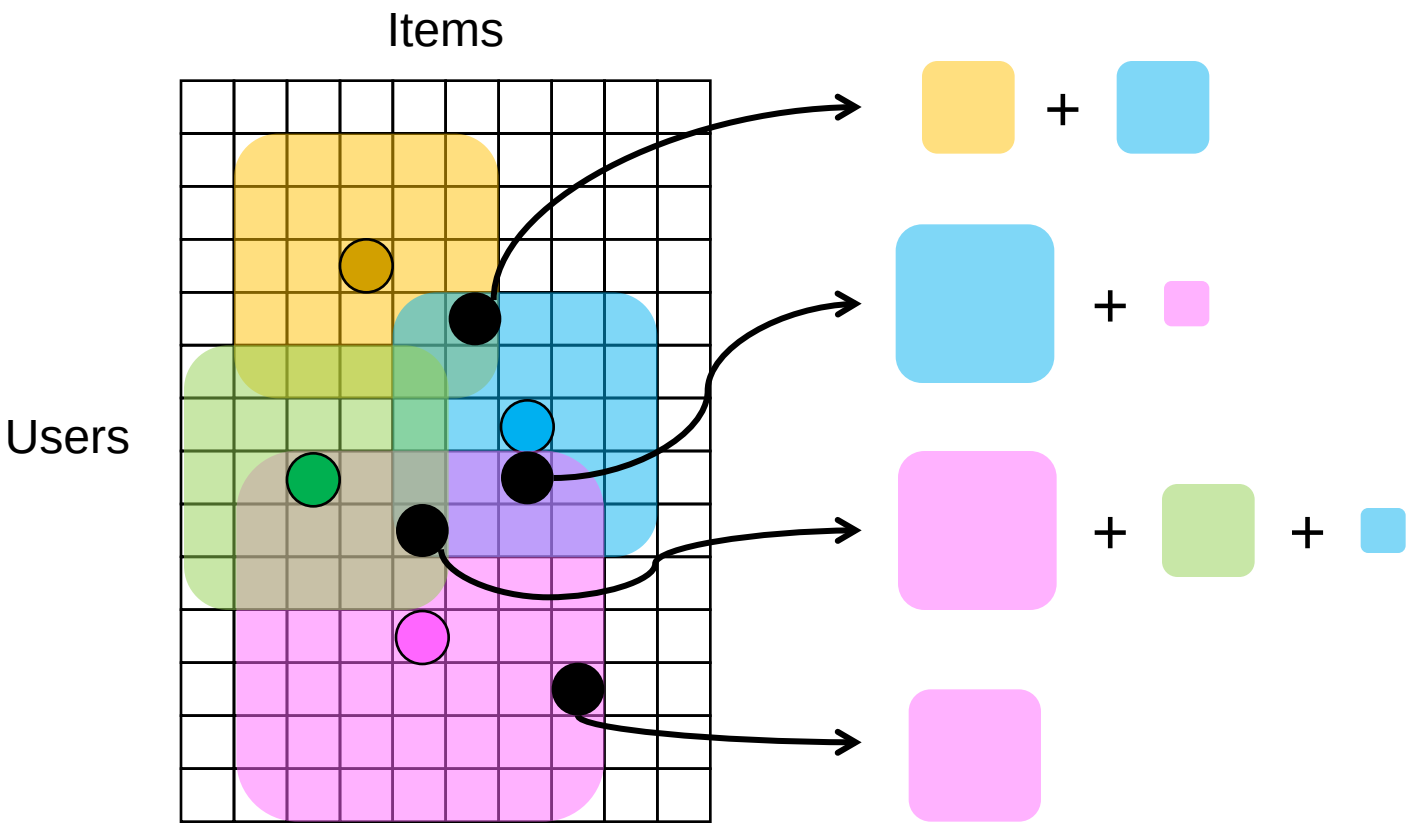
Incomplete SVD

- Diminishing returns?
  - H1: M has low rank; it reflects best possible prediction.
  - H2: M has high rank; diminishing returns due to over-fitting, or convergence to a poor local maximum.

- In recommendation systems,
  - H2 is a realistic assumption.
  - H1 is unrealistic globally, but it's realistic locally.

- The rating matrix is only locally of low-rank.
  - Build Low-Rank Matrix Approximation for local groups.
  - Combine them.

LLORMA: Illustration



LLORMA: Algorithm

- Learning: Smoothing kernels.

$$\min_{U,V} \sum_{(u,i)\in A} K((u^*,i^*),(u,i))\left([UV^T]_{u,i} - M_{u,i}\right)^2$$

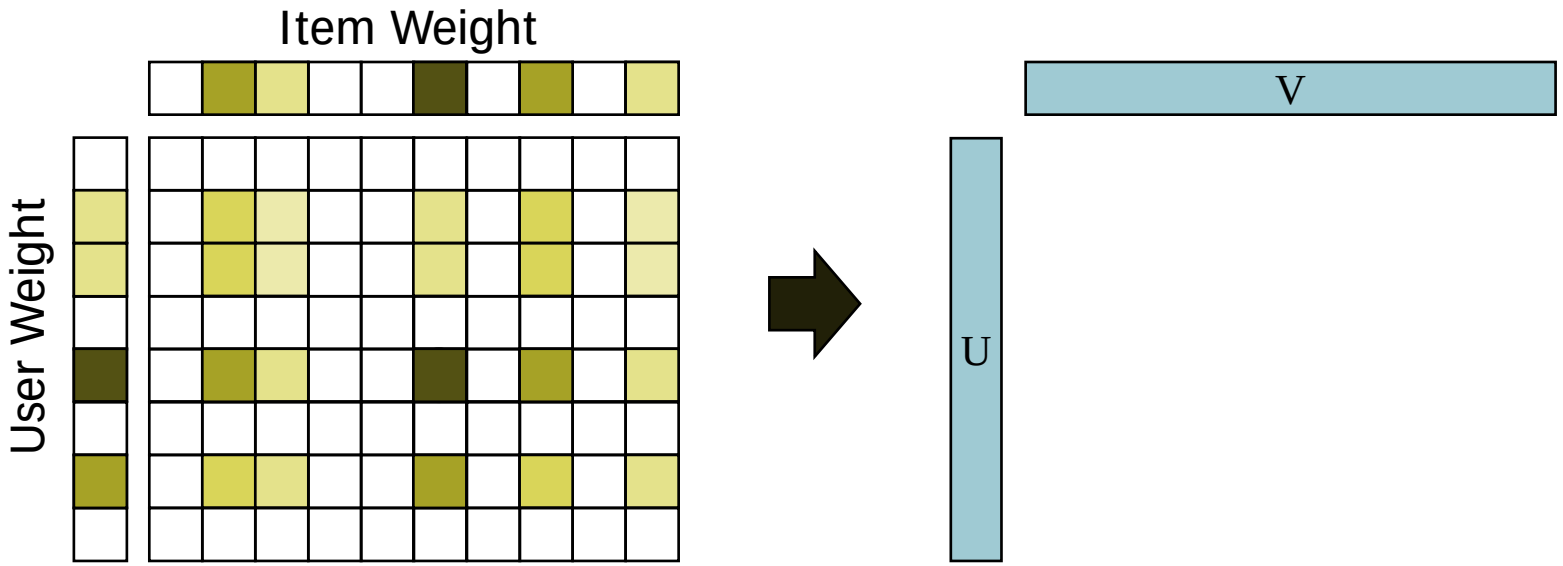
- Predicting: Nadaraya-Watson local regression.

$$\hat{M}_{u,i} \sim \sum_{t=1}^{q} K(u_t,u)K(i_t,i)\ [U_t V_t^T]_{u,i}$$

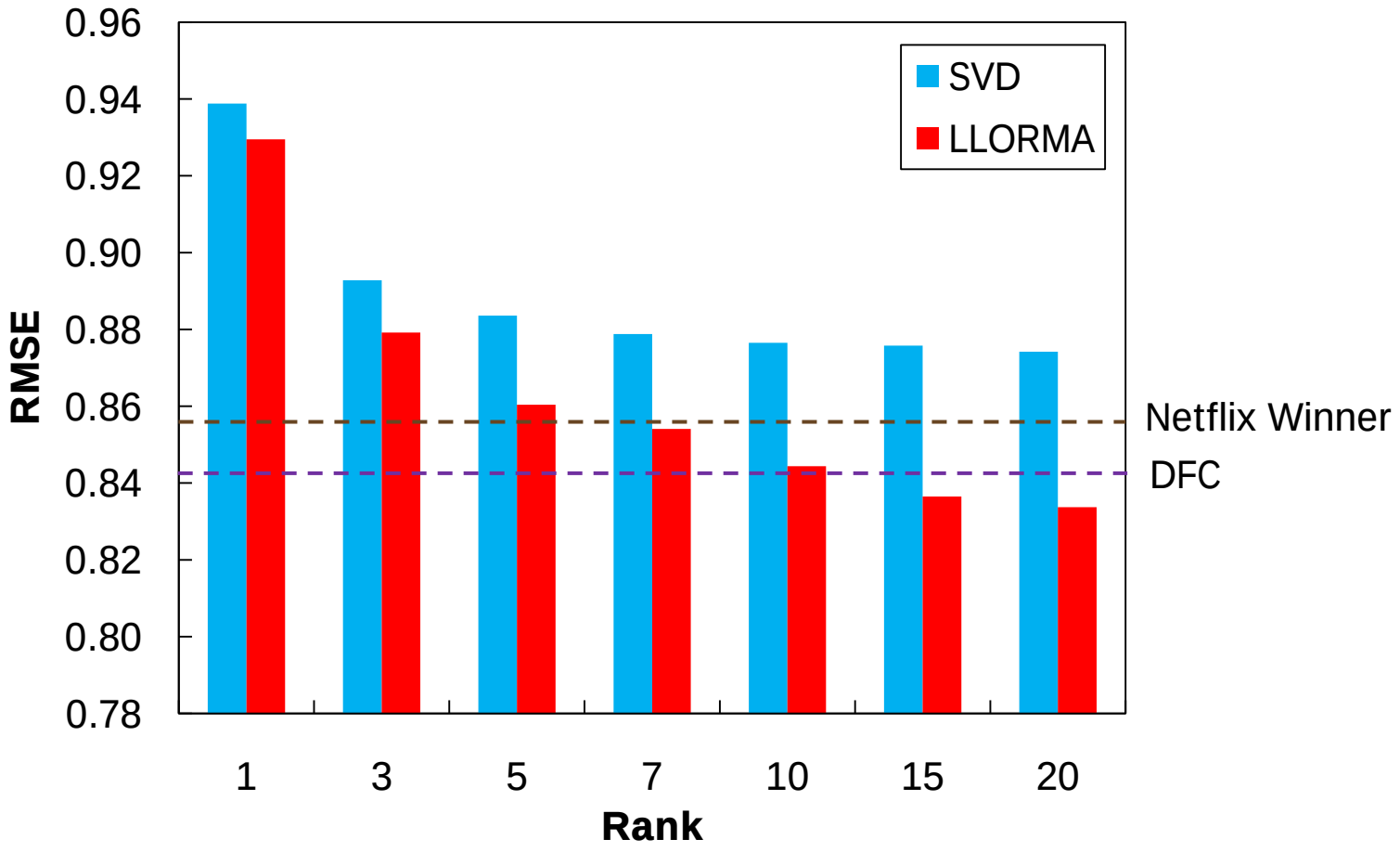
  - Values of local models at indices close to queried point contribute more than indices further away from it.

Learning Algorithm

- Run in Parallel:
  - Step 1: Select an anchor point.
  - Step 2: Calculate user/item weight with kernel smoothing.
  - Step 3: Solve a weighted incomplete SVD problem.



Experiment with Netflix data



Nuclear Norm Minimization

- Nuclear norm: sum of singular values.
  - a good surrogate for $\min_X rank(X)$. (Compressed Sensing)

- An alternative matrix completion:

$$\min_X \sum_i \sigma_i(X) \quad \text{s.t.} \quad \sum_{(u,i)\in A} (X_{u,i} - M_{u,i})^2 < \alpha$$

- Local variation:

$$\min_X \sum_i \sigma_i(X) \quad \text{s.t.} \quad \sum_{(u,i)\in A} K((u^*,i^*),(u,i))(X_{u,i} - M_{u,i})^2 < \alpha$$

Nuclear Norm Minimization

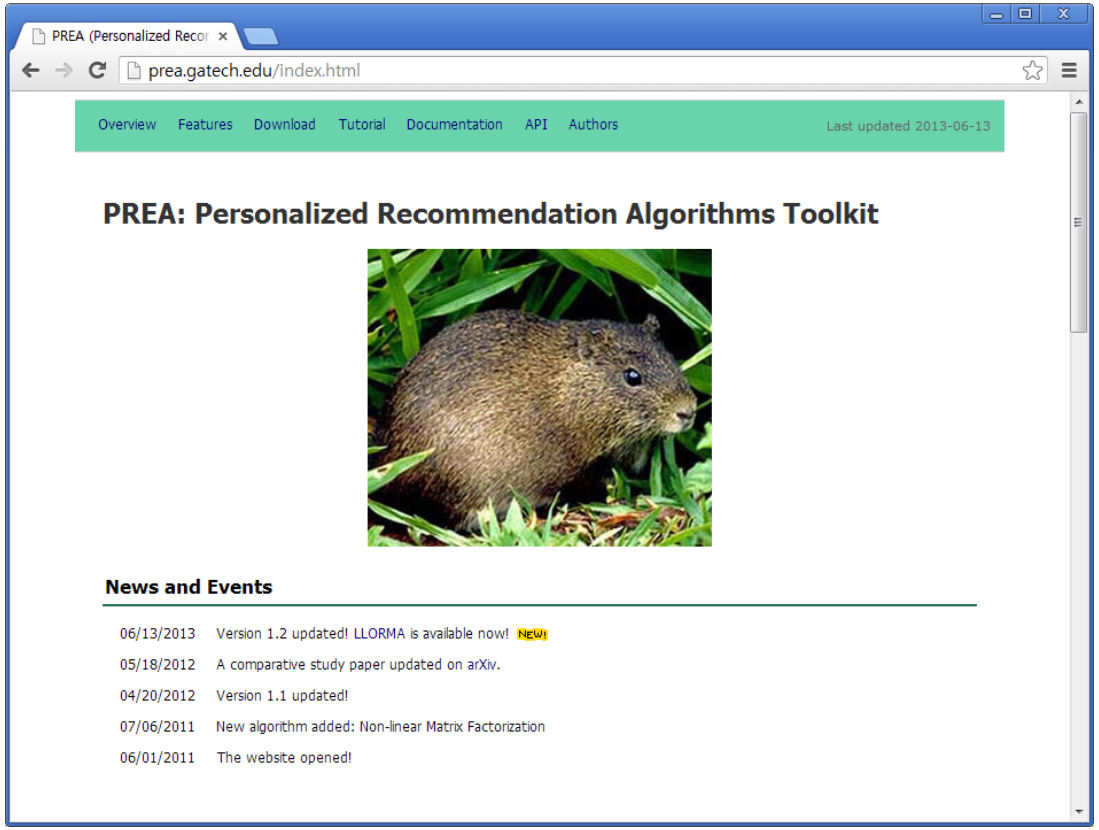
- Pros
  - Convex problem.
  - No need to specify rank in advance.
- Cons
  - Not scalable due to computational overhead.

Summary

- We proposed a matrix approximation under the local low-rank assumption.
- Our algorithm runs completely in parallel, leading to superior scalability.
- Experimental result supports that LLORMA outperforms several state-of-the-art methods without heavy computational overhead.
- With a formal analysis, we provide a theoretical bound in terms of matrix size, training set size, and locality.

PREA Toolkit

- Source code is online. http://prea.gatech.edu



Why PREA?

- Since Netflix Prize (2006), a lot of state-of-the-art algorithms were suggested.
  - They are implemented only by the authors.
  - Different language, different dataset, different evaluation measures.

- Standardized implementation is needed for fair comparison of CF algorithms.
  - PREA implements those algorithms on the same interface.

Features

Category	Feature	PREA	Mahout	Duine	Cofi	MyMedia
Baselines	Constant	O			O	O
	User/Item Average	O	O	O	O	O
	Random	O				O
Memory-based CF	User-based CF (Su and Khoshgoftaar, 2009)	O	O	O	O	O
	Item-based CF (Sarwar et al., 2001)	O	O	O	O	O
	Default Vote, Inv-User-Freq (Breese et al., 1998)	O	O			
	Slope-One (Lemire and Maclachlan, 2005)	O	O			O
Matrix Factorization	SVD (Paterek, 2007)	O	O		O	O
	NMF (Lee and Seung, 2001)	O				
	PMF (Salakhutdinov and Mnih, 2008a)	O				
	Bayesian PMF (Salakhutdinov and Mnih, 2008b)	O				
	Non-linear PMF (Lawrence and Urtasun, 2009)	O				
Other methods	Fast NPCA (Yu et al., 2009)	O				
	Rank-based CF (Sun et al., 2011, 2012)	O				
Evaluation Metric	(N)MAE	O	O	O	O	O
	RMSE	O	O		O	O
	HLU/NDCG	O				O
	Kendall's Tau, Spearman	O				
	Precision/Recall/F1		O			O
Miscellaneous	Sparse Vector/Matrix	O	O	O	O	O
	Wrapper for other languages	O			O	O
	Item Recommender for Positive-only Data					O
	Release Year	2011	2005	2009	2004	2009
	Language	Java	Java	Java	Java	C#
	License	GPL	LGPL	LGPL	GPL	GPL

THE END

Thank you for your attention!