# Recent Advances in Recommender Systems and Future Directions

George Karypis
Department of Computer Science & Engineering
University of Minnesota

# OVERVIEW OF RECOMMENDER SYSTEMS

# Recommender Systems

Recommender systems have emerged as a key enabling technology for ecommerce.

- Virtual experts that are keenly aware of a customer's preferences.
- They are used to filter vast amounts of data in order to identify information that is relevant to a user.

# Data sources

- Information about the items:
  - Product descriptions, product specifications, article content, media attributes, etc.
- Information about the users:
  - Profile data such as demographic, social-economic, and behavioral information.
  - Social/trust networks; either implicit or explicit.
- Transactional information:
  - History of user-item purchases.
  - Product ratings, product reviews.
  - Implicit vs. explicit information.

# Problems & approaches

**Recommendation problems**

- Rating prediction
  - Predict the rating that a user will give to a particular item.
- Top-N recommendation
  - Identify a set of items that the user will like.
- Cold-start recommendations
  - Compute recommendations for previously unseen users and/or items.
- Group recommendations
  - Compute recommendations that satisfy a group of users.
- Assortment recommendations
  - Recommend an automatically constructed assortment of items: e.g., vacation package, outfit, playlist.
- Context aware recommendations.

**Recommendation approaches**

- Non-personalized methods
  - Rule-based systems, popularity-based models, methods based on global models.
- Methods relying on a user's historical profile
  - Predictive models estimated using a user's individual preferences.
- Collaborative filtering methods
  - Approaches that leverage information from other users and items.

# Collaborative Filtering (CF)

- Derives recommendations by exploiting the “wisdom of the crowd”.
- It can be viewed as an instance of multi-task learning and transfer learning.
- It is the most prominent approach today:
  - Used by large, commercial e-commerce sites.
  - Well-understood, various algorithms and variations exist.
  - Applicable in many domains (book, movies, DVDs, ..).
  - It can operate both in a content agnostic and content aware setting.
  - It can handle both explicit and implicit preference information.

# Overview of CF methods

- User-, item-, and graph-based neighborhood methods.
  - Lazy learners.
- Methods that use various machine learning approaches to learn a predictive model from historical data.
  - Latent-space models based on matrix/tensor completion.
  - Linear and non-linear multi-regression models.
  - Probabilistic models.
  - Auto-encoder-based neural networks.
  - ….

# Item-based nearest-neighbor (1)

| | The Matrix | Titanic | Die Hard | Forrest Gump | Wall-E |
|---|---|---|---|---|---|
| John | 5 | 1 | | 2 | 2 |
| Lucy | 1 | 5 | 2 | 5 | 5 |
| Eric | 2 | ? | 3 | 5 | 4 |
| Diane | 4 | 3 | 5 | 3 | |

<u>Key assumptions:</u>

- Items belong into (overlapping) groups that elicit similar likes/dislikes.
- Users' preferences remain stable and consistent over time.

The basic technique:

– Given Eric ("active user") and Titanic ("active item") that Eric has not yet seen.

– Find a set of items (nearest neighbors) that Eric already rated such that other users liked each of them in a similar fashion as they liked Titanic.

– Use Eric's ratings on these items to predict how much Eric will like Titanic.

For top-N, do this for every unrated item and return the *N* items with the highest predicted rating.

# Item-based nearest-neighbor (2)

- Item-based methods pre-compute and store the *k* most similar other items for each item.
- The prediction scores are estimated using only those most similar items.

$$\hat{r}_{ui} = f(r_{u:}, s_{:i}) \text{ (e.g., } \hat{r}_{ui} = r_{u:}\, s_{:i})$$

where

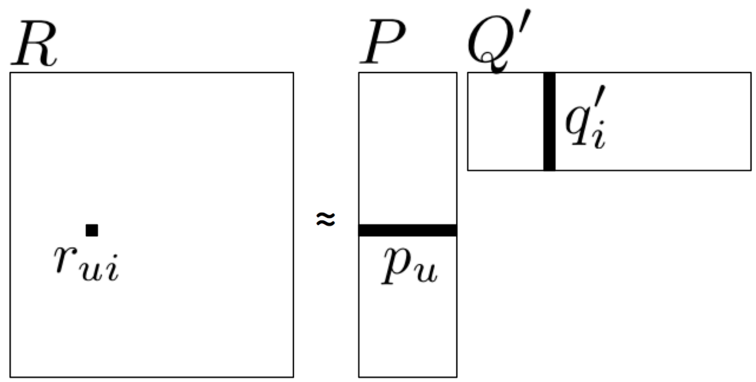
$R \in \mathbb{R}^{n \times m}$ is the matrix of historical ratings and

$S \in \mathbb{R}^{m \times m}$ is the *item-item* similarity matrix storing the $k$ highest similarities along each row.

- Similarities: cosine, Jaccard, correlation coeeficient, conditional probability, …

# Low-rank matrix approximation

The rating matrix can be approximated as the product of two low-rank matrices:

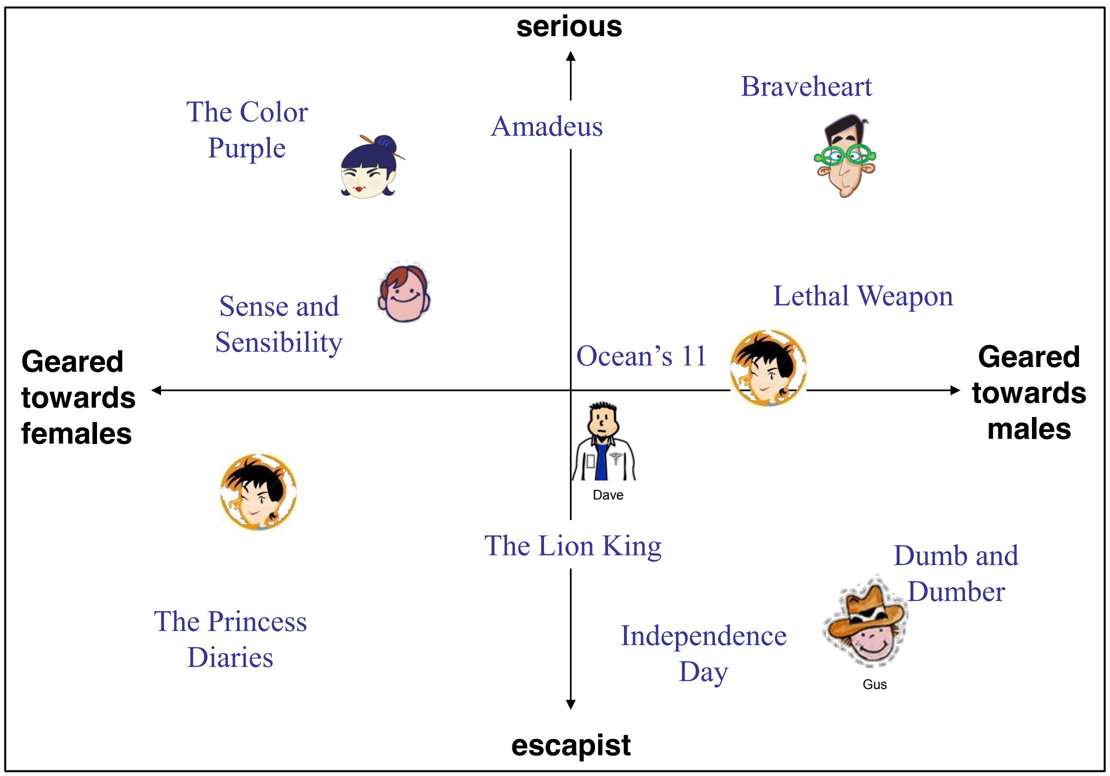


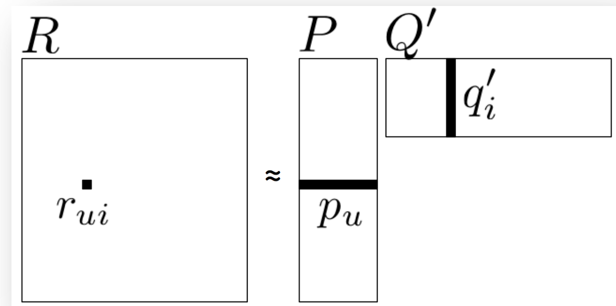

This low-rank decomposition can be used to compute the ratings for unseen items as:

$$\hat{r}_{ui} = p_u q_i'$$

# Interpretation of latent factors

There is a low dimensional feature space (whose dimensionality is the rank of the decomposition) on which both users and items can be embedded in.




- The item vector can be thought of as capturing how much of each of the latent features the item possess.
- The user vector can be thought of as capturing the user's preference for these latent features.

# Latent factors via matrix completion

- Estimate the latent factor matrices based only on the observed entries of the matrix.

Let $\Omega$ be the set of observed entries in the rating matrix $R$. The $P$ and $Q$ factors are estimated by solving the following optimization problem:

$$\underset{P,Q}{\text{minimize}} = \sum_{(u,i)\in\Omega} (r_{ui} - p_u q_i')^2.$$

- This is a non-convex optimization problem and can converge to a local minima.

# Improving accuracy of latent factor models

The performance of the rating prediction can be improved by

- Explicitly modeling global, user, and item biases
  - Biases: the baseline/expected ratings for all items by all users, for the items rated by a user, and for the users for a given item.
- Reducing model over-fitting via regularization:

$$\underset{P,Q,\mu,b_*}{\text{minimize}} = \sum_{(u,i)\in\Omega} (r_{ui} - \mu - b_u - b_i - p_u q_i')^2 + \lambda(\mu + ||b_*||_2^2 + ||P||_F^2 + ||Q||_F^2).$$

The estimated rating for user $u$ on item $i$ is given by:

$$\hat{r}_{ui} = \mu + b_u + b_i + pq'.$$

# Recent trends

- The boundaries between traditional neighbor and latent factor models have become less separated.
- Neighbor schemes rely on the latent space to compute item-item similarities.

$$R \approx PQ', \quad \mathsf{sim}(i,j) = q_i q_j'.$$

- Latent factorization methods rely on similar items to derive a latent representation of a user.

$$\hat{r}_{ui} = \mu + b_u + b_i + \left( \frac{1}{|\mathcal{U}|^\alpha} \sum_{j \in \mathcal{U}} r_{ui} q_j \right) q_i'.$$

# A SMALL DETOUR

# Chemical genetics (genomics)

***Chemical Genetics (Genomics):*** The research field that is designed to discover and synthesize protein-binding small organic molecules that can alter the function of all the proteins and use them to study biological systems.

(National Institute of General Medical Sciences)

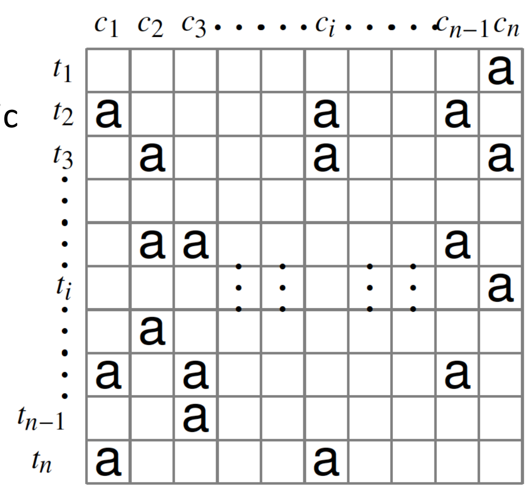
- Chemical genetics is a promising approach for studying biological systems.
- It has a number of key advantages over approaches based on molecular genetics:
  - small molecules can work rapidly,
  - their action is reversible,
  - can modulate single functions of multi-function proteins,
  - can disrupt protein-protein interactions, and
  - if the target is pharmaceutical relevant, it can lead to the discovery of new drugs.

# RS & CG – Data similarities

**Target-ligand activity matrix**

Ligand-intrinsic information
- chemical structure, pharmacophores, etc.

Target-intrinsic information
- sequence, structure, homology, etc.

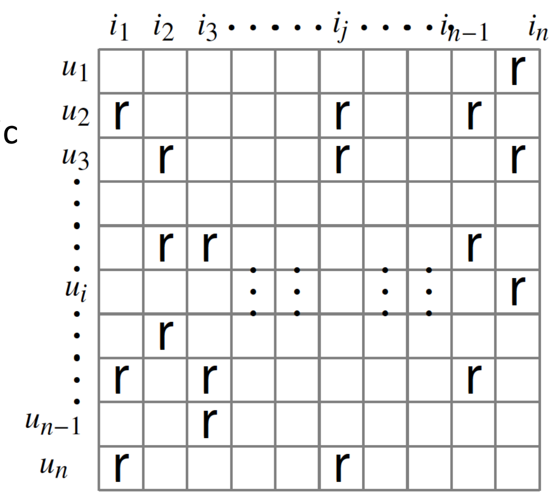



Activity information
- IC50, dose-response, etc.

**User-item rating matrix**

Item-intrinsic information
- descriptions, content, specification, attributes, etc.

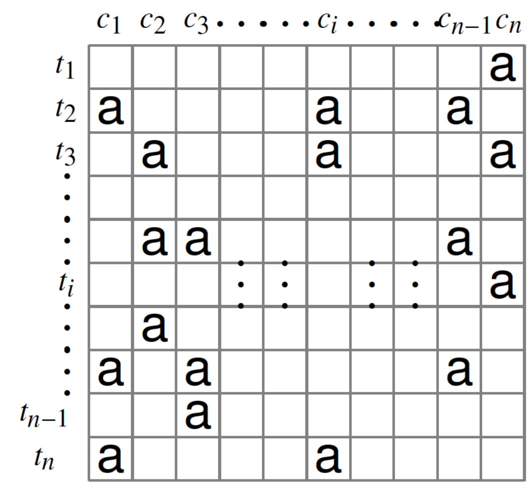
User-intrinsic information
- profile, social network, etc.

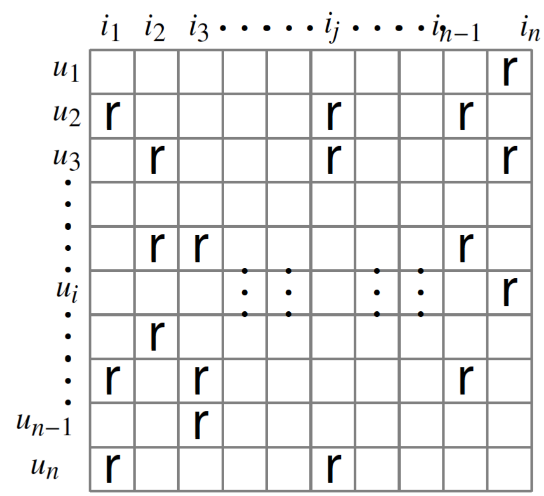



Transaction information
- rating, view, review, etc.

# RS & CG – Similar problems

- Activity prediction
  => Rating prediction
- Secondary screening library design
  => Top-N recommendation
- Library design for a novel target
  => Cold-start recommendation
- De novo compound design
  => Assortment recommendation



# RS & CG – Similar principles

- Ligand binding is a process that involves:
  - Structure of protein's binding site and the structure of the ligand.
  - Non-covalent interactions between the atoms of the protein's binding site and that of the ligand.

- As a result:
  - The same ligand will bind to similar targets.
  - Similar ligands will bind to the same target.
  - Similar ligands will bind to similar targets.

- These are the same principles and assumptions behind recommender systems.

# RS & CG – Similar methods

- Target-specific Structure-Activity Relationship (SAR) models.
  - The biological activity of a chemical compound is mathematically expressed as a function of its chemical structure.
- Chemogenomic approaches.
  - Proteins of the same family tend to bind to ligands with certain common characteristics.
- Multi-task learning approaches.
  - Models that estimate relations between protein- and ligand-derived features.

- Personalized content-based recommender systems.
  - The user's past transactions are used to derive content-based models for like/dislike prediction.
- Cluster-based recommender systems.
  - Content-based models for similar groups of users.
- Collaborative filtering approaches with side information.

# RECENT DEVELOPMENTS IN ITEM-BASED APPROACHES

# Estimating the similarity matrix

Goal:

Instead of using a pre-determined function to compute the item-item similarities, “learn” them directly from the data.

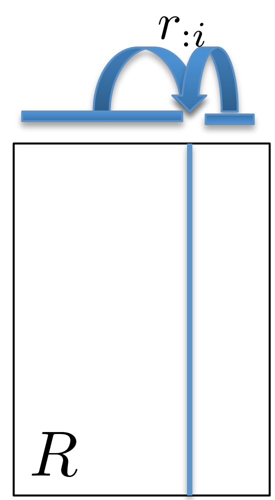
Learning problem formulation:

Estimate a linear model to predict each item based on the historical information of the other items.

Let $w_i$ be the linear model for item $i$, let $R_{\neg i}$ be the $n \times (m-1)$ sub-matrix of $R$ obtained by removing column $i$ (i.e., the historical information for item $i$), and let $r_{:i}$ be the $i$th column of $R$.
The $w_i$ is estimated as:

$$\underset{w_i}{\text{minimize}} \left( \frac{1}{2} ||r_{:i} - R_{\neg i} w_i||^2 + \text{reg}(w_i) \right).$$

This model becomes the $i$th column of the item-item similarity matrix $S$.



# SLIM – Sparse Linear Method

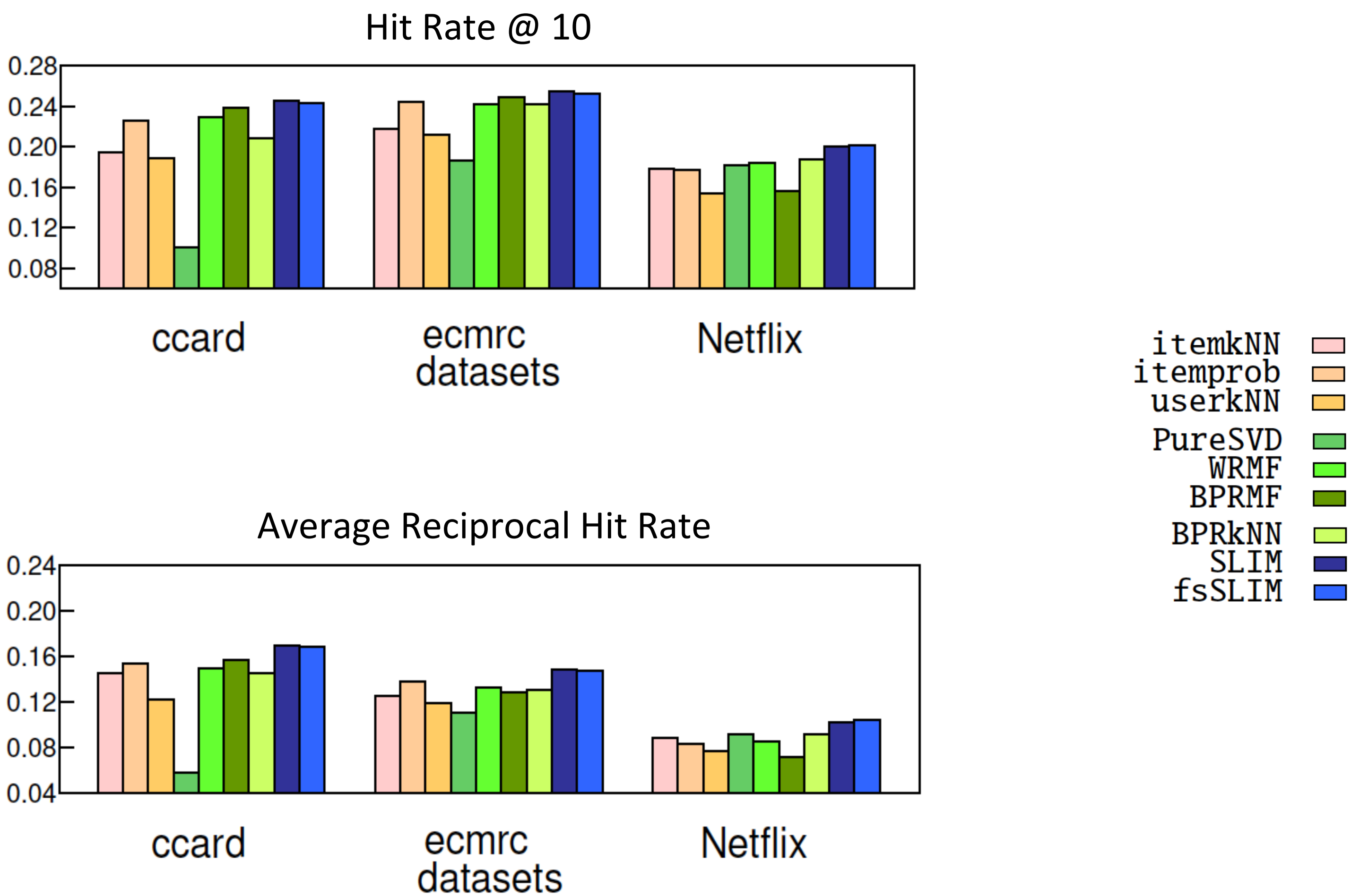
- The model (*S*) is estimated as:

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \frac{1}{2}||R - RS||_F^2 + \frac{\beta}{2}||S||_F^2 + \lambda||S||_1 \\ \text{subject to} \quad & S \geq 0 \\ & \text{diag}(S) = 0 \end{aligned}$$

> This is a Structural Equation Model (SEM) with no exogenous variables.
> It can also be viewed as a network inference model.

- Good performance is often achieved with 50-200 non-zeros per column.
  - Low storage requirements and low recommendation time.
- The model can be estimated efficiently:
  - Each column of *S* can be computed independently.
  - The solution of a column can "warm start" other similar columns.
  - Regularization space can be explored efficiently.
  - Item neighbors can be used for initial "feature" selection and restrict sparsity structure.

# SLIM – Performance

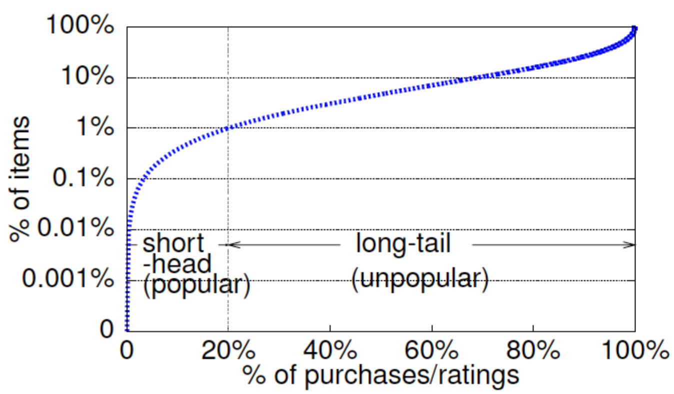



$$\text{Hit Rate (HR) @ N} = \frac{tp}{N}$$

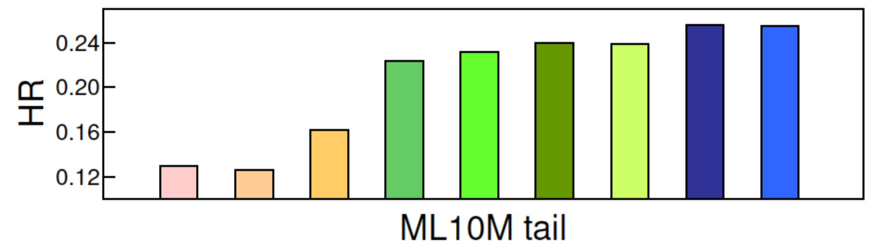
$$\text{ARHR}(L) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{\text{rank}(i_u, L(u))}$$

# SLIM – Long-tail performance

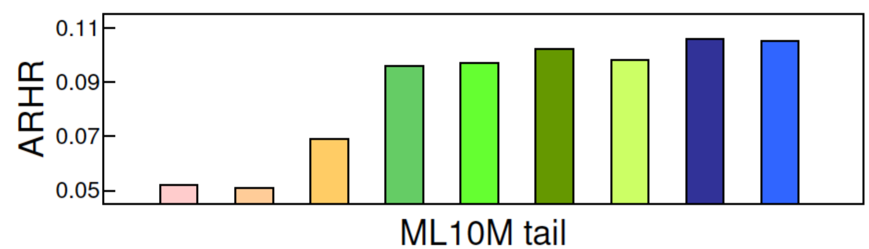
Rating Distribution in ML10M

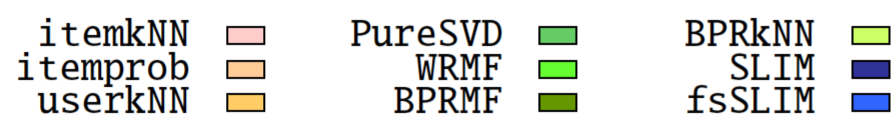



HR@10 in ML10M Long Tail




ARHR in ML10M Long Tail



# SLIM extensions

$$\begin{aligned} \underset{S}{\text{minimize}} \quad & \frac{1}{2}||R - RS||_F^2 + \frac{\beta}{2}||S||_F^2 + \lambda||S||_1 \\ \text{subject to} \quad & S \geq 0 \\ & \text{diag}(S) = 0 \end{aligned}$$

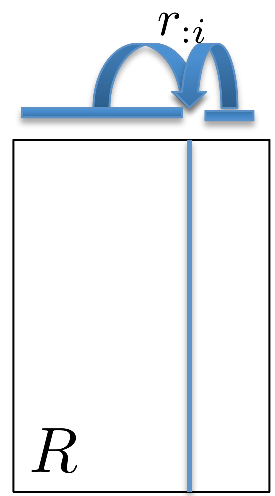
- Low-rank constraints on *S* as an alternate way to control model complexity.
  - Either via the product of two low rank matrices or by minimizing the nuclear norm of *S*.
- Incorporation of item side information.
- Incorporation of different contexts.
- Incorporation of temporal information.
- Higher-order regression models.
- Fusion of global and local SLIM models.

# Factored similarity matrix

- SLIM (and traditional item-item approaches) cannot compute/learn relations between pairs of items that are not co-rated.
  - The estimated similarities for such pair of items will always be 0.

- They cannot produce meaningful recommendations that rely on transitivity within the item-item similarity graph.

- Can lead to poor recommendations, especially for sparser datasets, in which there are few pairs of co-rated items.

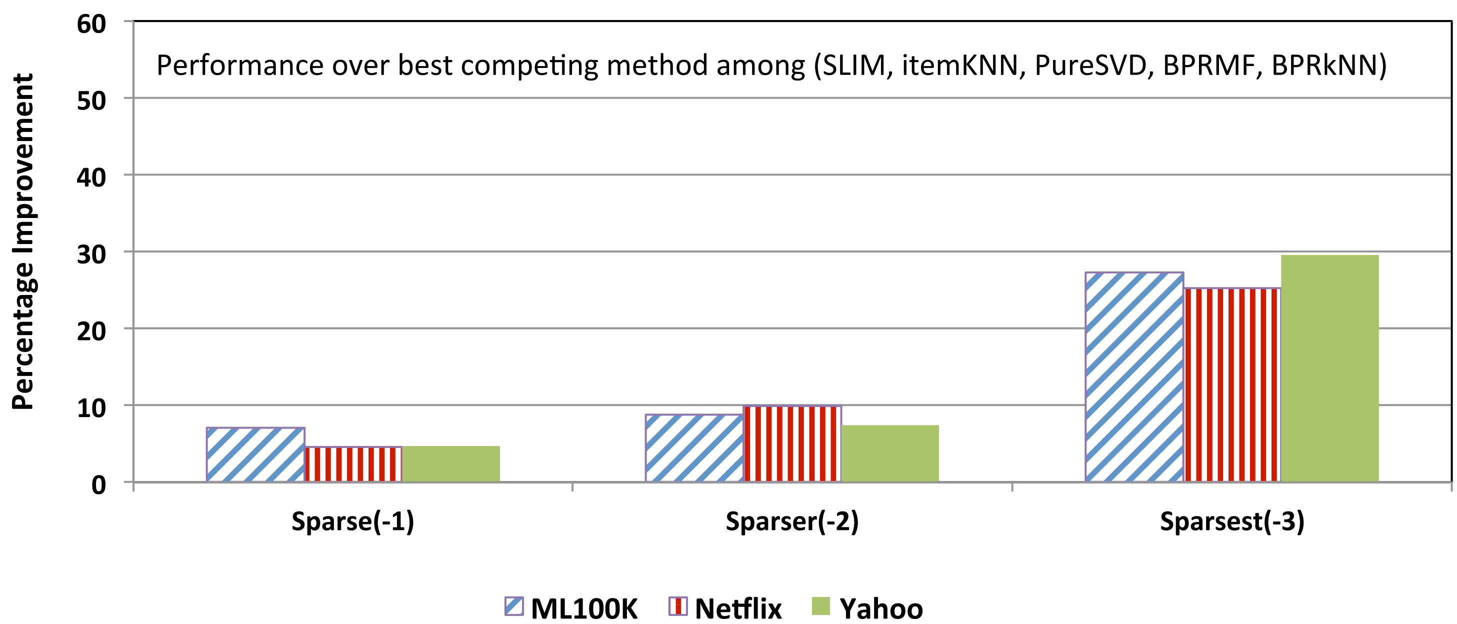


# FISM – Factored SLIM

$$\underset{P,Q}{\text{minimize}} \left( \frac{1}{2} ||R - RPQ^T||_F^2 + \frac{\beta}{2}(||P||_F^2 + ||Q||_F^2) \right)$$

$$\hat{r}_{ui} = \sum_{j \in \mathcal{R}_u} r_{uj} p_j q_i^T$$

Since the overall problem is not convex, optimization and search over the regularization space becomes considerably more expensive.
Scalable approaches rely on SGD.



# Learning from higher-order relations

- If a customer buys a certain group of items, they are more (or less) likely to buy some other items.


- The joint distribution of a set of items can be different from the distributions of the individual items in the set.
- Higher order models are needed to capture such relations.

# HOSLIM—Higher-order SLIM

- Key Idea: Use frequent itemsets to capture higher order relations.
- Two-step approach:
  - Given a user-item purchase matrix *R*, find all frequent itemsets and create a user-itemset matrix $R^f$.
  - Learn two similarity matrices *S* and $S^f$ that capture item-item and itemset-item relations.
- Predict new items by combining information from *S* and $S^f$.

Model estimation:

$$\underset{S}{\text{minimize}} \quad \frac{1}{2}||R - RS - R^f S^f||_F^2 + \frac{\beta}{2}(||S||_F^2 + ||S^f||_F^2) + \lambda(||S||_1 + ||S^f||_1)$$

$$\text{subject to} \quad S \geq 0,\ S^f \geq 0,$$

$$\text{diag}(S) = 0,\ \text{and} \quad s_{ji}^f = 0 \text{ for } i \in \mathcal{I}_j$$

Prediction: $\hat{r}_{ui} = r_{u:}\, s_{:i} + r_{u:}^f\, s_{:i}^f$

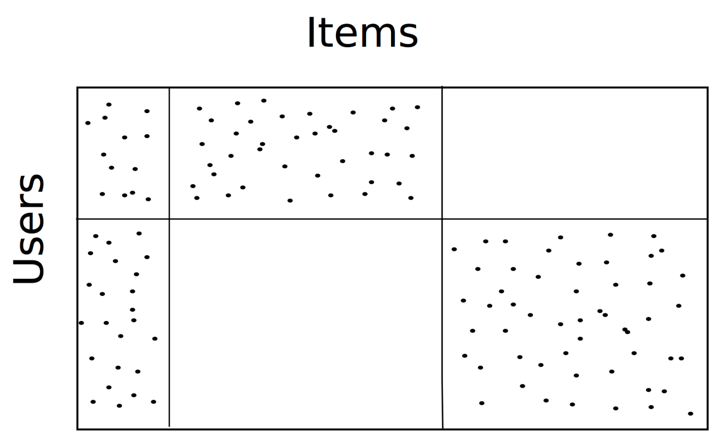
# HOSLIM performance

| | SLIM models | | |
|---|---|---|---|
| Dataset | SLIM HR | HOSLIM HR | Improved % |
| groceries | 0.259 | **0.338** | 32.03 |
| synthetic | 0.733 | **0.860** | 17.33 |
| delicious | 0.148 | **0.156** | 5.41 |
| ml | 0.338 | **0.349** | 3.25 |
| retail | 0.310 | **0.317** | 2.26 |
| bms-pos | 0.502 | **0.509** | 1.39 |
| bms1 | 0.588 | **0.594** | 1.02 |
| ctlg3 | 0.581 | **0.582** | 0.17 |

Considerable gains can be obtained for datasets that have such characteristics.

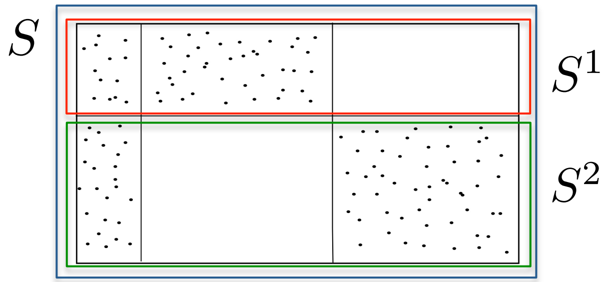
# One model does not fit all

- In item-based methods, personalization occurs based on the items that the user has previously acted on.
  – The “recommendations” that these items trigger are not specific to each user but are global.




- Better performance can potentially be achieved by having user-specific item-based models.

# GLSLIM – Fusion of global and local SLIM models




$$\hat{r}_{ui} = \sum_{j \in \mathcal{R}_u} r_{uj} \left( g_u s_{ji} + (1 - g_u) s_{ji}^{c_u} \right)$$

user's membership

global model

local model

$$\underset{S, S^1, \ldots, S^K, c, g}{\text{minimize}} \quad \frac{1}{2} \sum_{ui} (r_{ui} - \hat{r}_{ui})^2 + \frac{\beta}{2} (||S||_F^2 + \sum_k ||S^k||_F^2) + \lambda (||S||_1 + \sum_k ||S^k||_1)$$

subject to

$S \geq 0, \; S^k \geq 0$ for $k = 1, \ldots, K,$

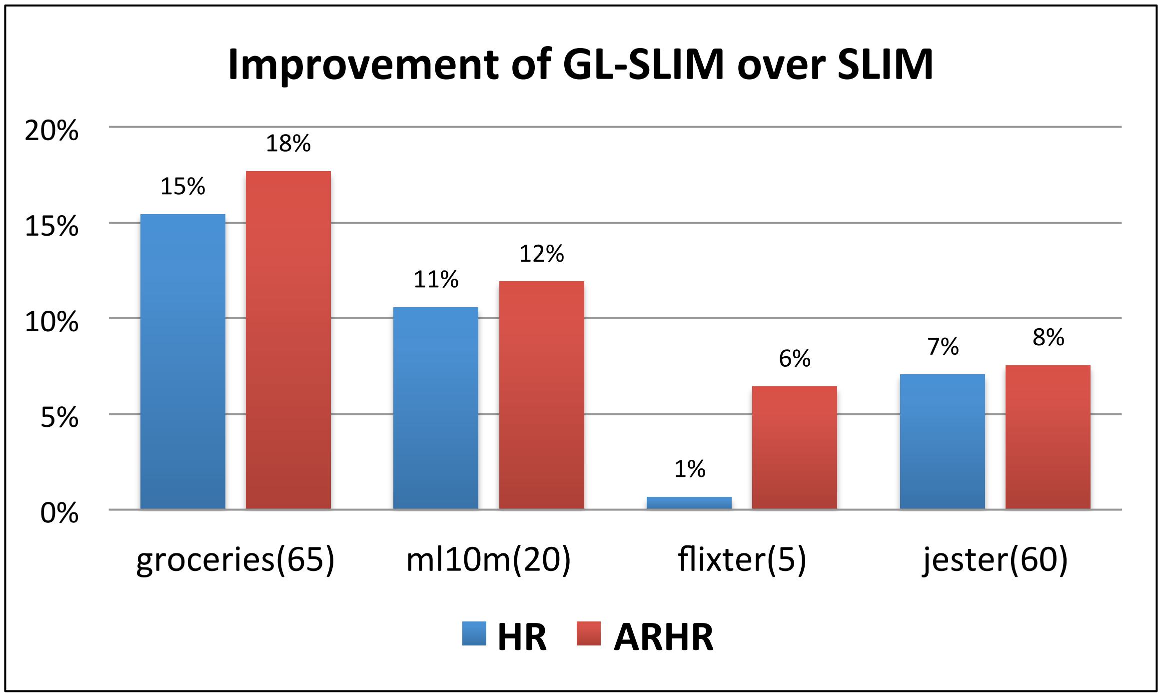
$\text{diag}(S) = 0, \text{diag}(S^k) = 0$ for $k = 1, \ldots, K,$

$\forall u, 0 \leq g_u \leq 1,$ and

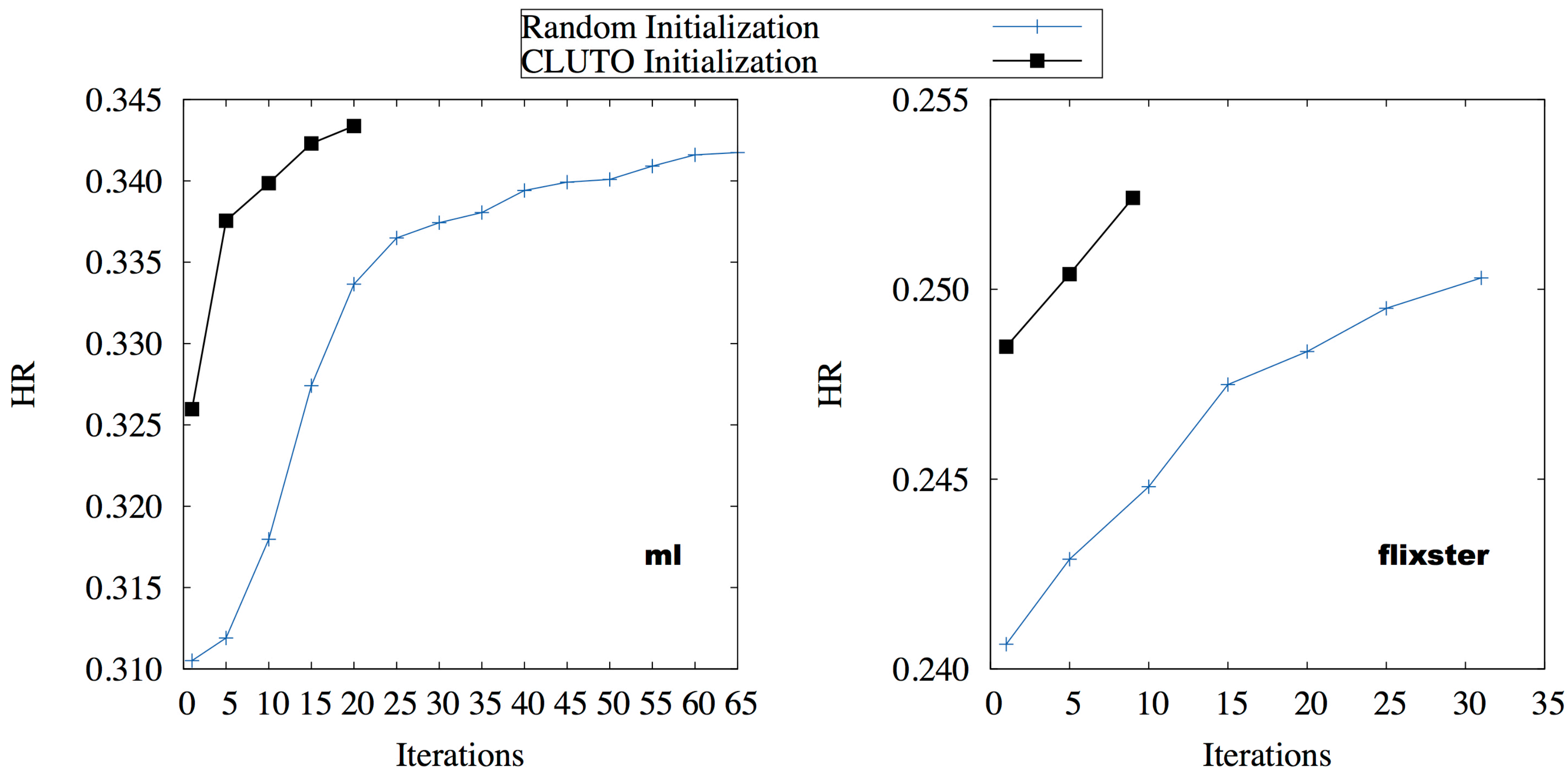
$\forall u, c_u \in \{1, \ldots, K\}.$

Optimized using alternate optimization between solving for the various SLIM models and solving for the clustering and user membership.

# GLSLIM – Fusion of global and local SLIM models



# GLSLIM – Fusion of global and local SLIM models




Smart initialization reduces the number of iterations but does not significantly impact the final solution.

# WRAPPING UP

# Current state of the art

- Rating prediction
  - Factorization-based approaches are highly effective:
    - Good prediction performance, fast training time, they can incorporate side information, diverse objectives, etc.

- Top-N recommendation methods
  - No clear winning strategy has emerged.
    - Item-based methods outperform significantly more sophisticated methods.
  - There is significant ongoing research on ranking loss functions.

# Scaling up

- Estimate only item factors from a subset of users & compute user factors on the fly.

- Warm-start the search over the regularization space:
  - For MF use SVD to eliminate bad local minima.
  - For SLIM use previous solutions and/or solutions for similar items.

- Sample the users. The goal is to get a reliable set of item-based models:
  - Item factors or item-item models.

# Future directions

- Deep personalization
  - Context, location, time, etc.
- Behavior steering
  - Acquiring taste, modeling state, future benefits, etc.
- Content creation
  - Package recommendation, article generation, etc.
- Live evaluations
  - Open A/B testing platforms.

# Final words…

- Recommender systems have extensive applications.
  - Both commercial, scientific, and societal applications.
- There are already high-quality software implementations of many of these algorithms.
  - They can be used as is, or used to quickly experiment with new modeling approaches and data sources.
- Multi-task learning underlies many of the learning methods.
  - A very active area of research with broad applications.
- The field is ripe for new methodological advances that will get us to the next level.

# Acknowledgments

Students (current & former)

- Xia Ning
- Evangelia Christakopoulou
- Santosh Kabbur
- Asmaa El Badrawy
- Agoritsa Polyzou

Funding

- NSF, ARO, NIH, PayPal, Samsung, Intel.

References:

Recommender Systems Handbook, 2015
http://www.springer.com/us/book/9781489976369

Papers at
http://www.cs.umn.edu/~karypis