YAHOO!
RESEARCH

Item Cold-Start Handling in Collaborative Filtering Recommenders

DMBI’2016, Beer-Sheva Univ.

PRESENTED BY **Oren Somekh** May 19, 2016

Introduction and background

- Recommendation technologies are ubiquitous!
  - Media, eCommerce, mobile app stores, advertising and more…
- Internet users will find it very hard to navigate through the overwhelming content volume and find what they like without recommender systems
- Two main technologies:
  - **Content based** - uses users and content attributes
  - **Collaborative filtering** (CF) - relays solely on historical user's interactions
    - Interactions can be implicit (e.g., clicks, skips) or explicit (e.g., ratings)
- We focus on CF recommenders
  - Requires no domain knowledge
  - Detects popularity trends
  - Reveals complex and unexpected patterns

Introduction, background, and motivation (Cont'd)

- The principle of CF technique through a toy example

	Item A	Item B	Item C	Item E	Item F
User 1	👍	👎	👍	👍	?
User 2	👍	👎	👍	?	👎

- *Latent factor models* (LFM) is one of the leading CF techniques
  - Entities (users and items) are represented by vectors in a low dimensional latent space
- *Matrix Factorization* (MF) is a popular and successful realization of LFM
  - Low rank factorization of the user-item interaction matrix (usually sparse)

$$\mathcal{R}_{n\times m} \approx P^T_{k\times n} Q_{k\times m}$$
$$k \ll n,\ m$$

$$R = \begin{pmatrix} ? & 3 & 5 & ? \\ 1 & ? & ? & 1 \\ 2 & ? & 3 & 2 \\ ? & ? & ? & 5 \\ 5 & 2 & 3 & 4 \end{pmatrix}$$

- The entities vectors may be learnt by minimizing some **cost function** using gradient descent (SGD) over the **known** entries

The Cold-Start Problem

- Inherent problem of CF recommenders:
    - It is hard to characterize new entities with few or no historical data - **cold-start problem**
- Common solutions for generating initial characterization of new entities
    - New users may be “interviewed” when joining a service
    - Content information (e.g., item attributes) may be combined into the model - hybrid recommender
        - May not be available
    - A small portion of the traffic may be devoted for **random exploration** of new items
        - Inefficient, not scalable, and costly

What else can be done to mitigate the inherent **item cold-start problem**?

Assumption: we have a mature CF-MF model for all users

Selecting reviewers to rate new items

O. Anava, et al., *"Budget-Constrained Item Cold-Start Handling in Collaborative Filtering Recommenders via Optimal design"*, WWW'2015

- Use case:
  - eCommerce site gets a new book and would like to recommend it to its users
  - The operator has a budget to order book reviews from B users
- Two questions:
  - How to select the "best" reviewers for the job?
  - How to combine their reviews to generate an "optimal" characterization?
- We adopt an **optimal design** approach
  - Borrowed from statisticians originally using it to select a subset of experiments

Selecting reviewers to rate new items

- Applying an *optimal design* approach:
  - We assume that our **mature** model is correct up to an additive noise term

$$r_{ui} = \mu + b_i + b_u + Q_i^\top P_u + \varepsilon_{ui}$$

  - We use a *mean square error* (MSE) cost function

$$\min_{\mathcal{U}_B^i \subset \mathcal{U}^i} \left\{ \mathbb{E}\left[\frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\tilde{r}_{ui} - r_{ui} \right)^2 \right] \right\}$$

  - The optimal new item latent vector estimate is the *least square* (LS) solution of the MSE of the reviewers rating errors - answering the second question
  - Minimizing the MSE is equivalent to

$$\min_{\mathcal{U}_B^i \subset \mathcal{U}^i} \left\{ \sigma^2 tr\left(\left(P_B P_B^\top \right)^{-1} \right) + \sigma^2 \right\}$$

Independent of the actual ratings!

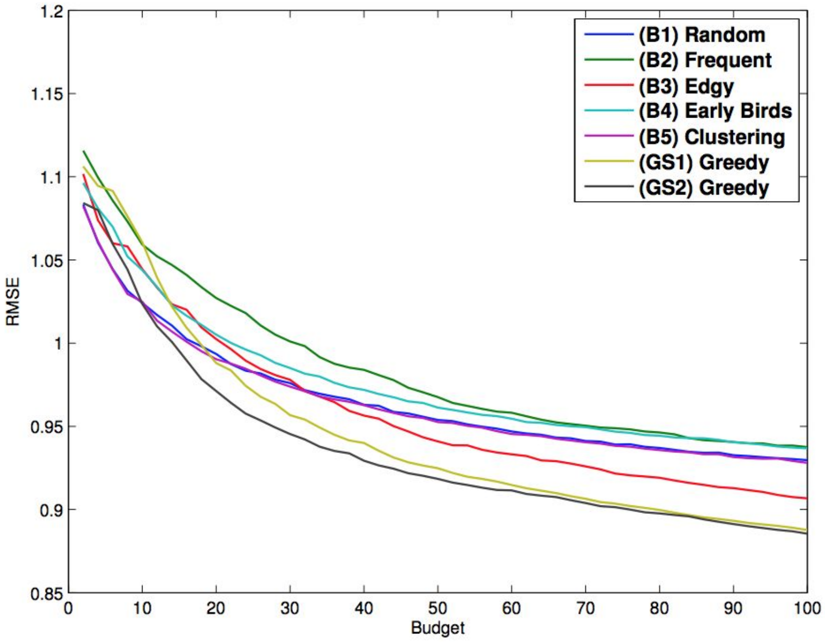
Still a hard problem!

YAHOO!

Selecting reviewers to rate new items (Cont'd)

- We showed that the equivalent cost function is **supermodular monotone decreasing**
  - The reviewer selection problem is solvable yielding $\frac{e^t - 1}{t}$ approximation to the optimal solution using **greedy algorithms!**

- Experimental results
  - Netflix dataset
  - Performance metric - RMSE
  - Different reviewer selection approaches
  - Greedy is much better than random selection
  - Edgy is the best baseline



YAHOO!

Smart exploration of new items

D. Drachsler, et al., *"ExcUseMe: Asking Users to Help in Item Cold-Start Recommendations"*, RecSys'2015

- Use case:
  - Media site uses random exploration to characterize new items for its CF-MF recommender
  - Random exploration is not scalable in the number of new items and fixed traffic
- Question:
  - Can we explore new items more efficiently?
- We propose a **smart exploration** approach where *K* exploring users are carefully selected
  - Users arrive randomly one after the other (online setting)
  - The system has to immediately decide whether the incoming user will explore the new item
- Our problem setting fits nicely to the *K secretaries problem*
  - An extension of the well known *secretary problem*

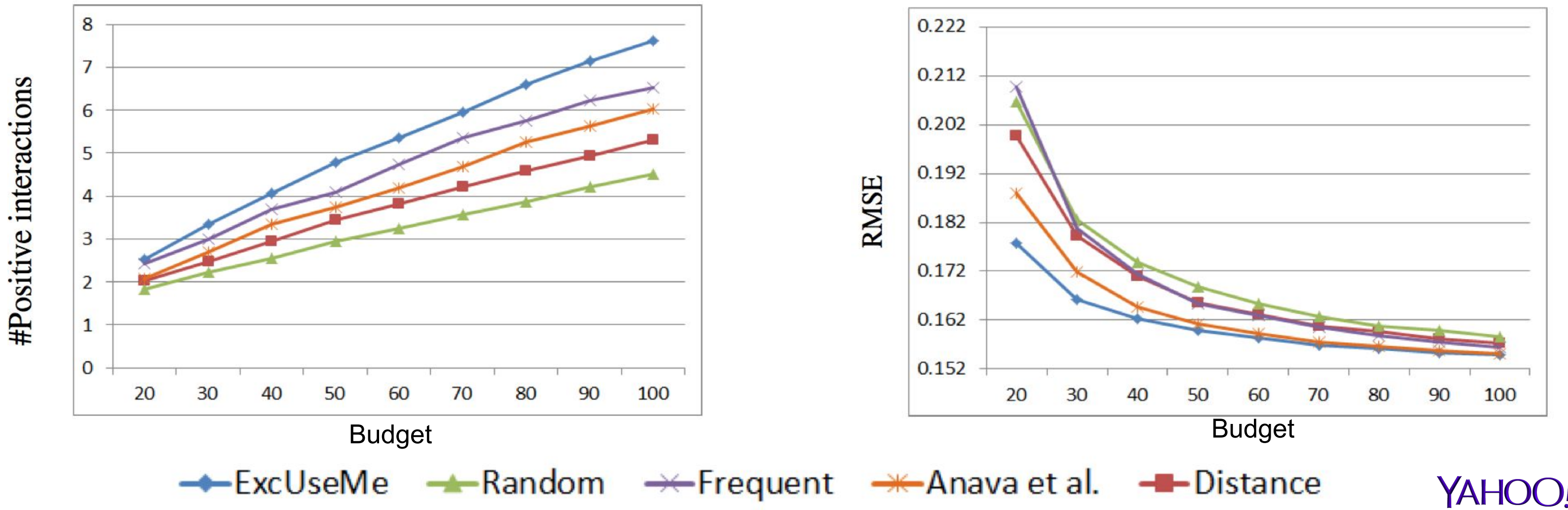
Smart exploration of new items

- We adopt the following *K secretaries* framework [A. Bentai et al., ACM Trans. Alg., 2013]
  - Split the set of arriving users into *K* portions and select one exploring user per portion
- For each portion of incoming users:
  - We further split the portion into two phases - *learning* and *selecting* phases
  - **Learning phase**: look for the maximal user value *Sm*
  - **Selection phase**: select the first user with value higher or equal than *Sm* or the last user
- How to evaluate the value of the arriving user w.r.t. the new item?
- We propose ExcUseMe user evaluator:
  - Initialize an auxiliary vector $V_{\text{aux}} = 0$
  - The user value $F_s(u, V_{\text{aux}}) = V_u^T \; V_{\text{aux}} + b_u$
    - In general the value is higher if the user is more “similar” to the auxiliary vector
  - After each portion update the auxiliary vector using all the users picked so far (e.g., LS for MSE)
  - After K portions the auxiliary vector is the new item vector

YAHOO!

Smart exploration of new items

- Experimental results
  - MovieLens1M dataset
  - Performance metric - RMSE
  - The exploring users are selected from the first 25% arriving users
  - ExcUseMe achieves the lowest RMSE
  - ExcUseMe gets the highest number of positive interactions - highest user engagement



YAHOO!

Boosting new items exploration via attribute-to-feature mapping

D. Cohen, et al., “Boosting Exploration by Attribute-to-Feature Mapping for Cold-Start Recommendation”, submitted

- Use case:
  - Media site uses random exploration to characterize new items for its CF-MF recommender
  - Item attributes are available but the operator wishes to keep the current recommender
- Example: movie genres

Index	Movie	Genres
1	The Butterfly Effect	Drama
2	Toy Story	Animation,Comedy,Fantasy

$$a_1 = [0,0,1,0]^T$$
$$a_2 = [1,1,0,1]^T$$

- Question:
  - How can we use the items attributes without changing the recommender?
- Answer [S. Rundle, et al., ICDM’10]:
  - Learn a linear mapping between the item attribute space and the model latent space

$$W_{k\times\ell} \; : \; \mathbb{Z}_2^\ell \rightarrow \mathbb{R}^k$$

  - Use the mapping to generate an initial latent vector for new items

$$q_i^0 = W_{k\times\ell}\, a_i$$

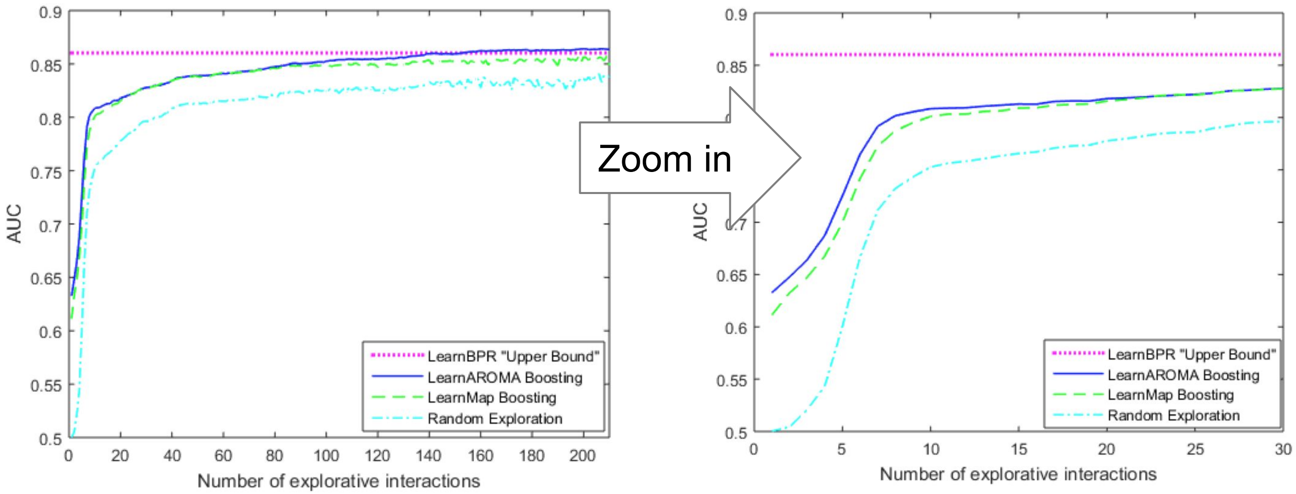
Boosting new items exploration via a2f mapping (Cont'd)

- The LearMAP algorithm [S. Rundle et al., ICDM'10]
  - The linear mapping is learnt by replacing the mature items vectors with $W_{k \times \ell} \, a_j$ in the cost function
- We proposed LearnAROMA algorithm - alternative algorithm for learning the mapping
  - Based on *adaptive regularization of matrix algorithm* (AROMA) originally proposed for classification [K. Crammer et al., ICML'12]
  - A confidence-weighted online algorithm that learns a Gaussian distribution $\mathcal{N}(\mathbf{w}, \mathbf{\Sigma})$
- Boosting random exploration
  - Use the new item attributes and mapping to generate an initial latent vector estimate
  - Update the initial estimate with each incoming exploration rating
    - Simple convex optimization
    - The initial vector is used via L2 regularization term added to the target function

YAHOO!

Boosting new items exploration via a2f mapping (Cont'd)

- Experimental results
  - MovieLense2K dataset (genres and crew information as item attributes)
  - Performance metric - *area under the curve* (AUC)
    - The probability of picking a random item pair and rank it correctly for a user using the model
  - Our boosting algorithm requires 70% less ratings than random exploration to achieve 95% accuracy
  - Our mapping learning algorithm LearnAROMA shows a 3% lift in initial estimate AUC



YAHOO!

Concluding remarks

- Collaborative-Filtering Matrix Factorization based recommenders
- Inherent item cold-start problem
- Three use cases:
  - Selecting reviewers to explore new items (offline setting)
  - Smart exploration of new items (online setting)
  - Boosting new items exploration via attribute-to-features mapping (online setting)

If you know your users and have some information on the new items you can mitigate the item cold-start problem in CF-MF recommenders

Thank You!