Matrix Factorization Recommender Systems and Cold Start

Aaron Tuor and Brian Hutchinson, Computer Science Department, Western Washington University

Overview

Recommender Systems

Created to direct users to items of interest, given a dataset, e.g., user ratings or product descriptions.

Cold Start Problem

How to give ratings for new users and new items for which no ratings are recorded in the dataset?

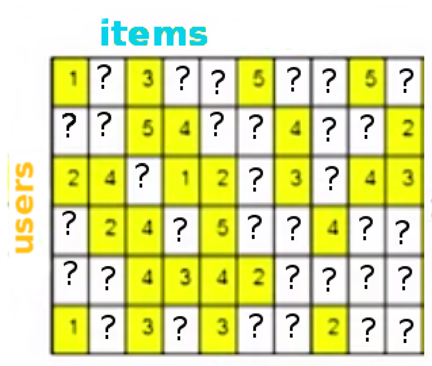
Purpose

- Survey state of the art Matrix Factorization solutions to the Cold Start problem.
- Outline our own current research.

User-Item Utility Matrix

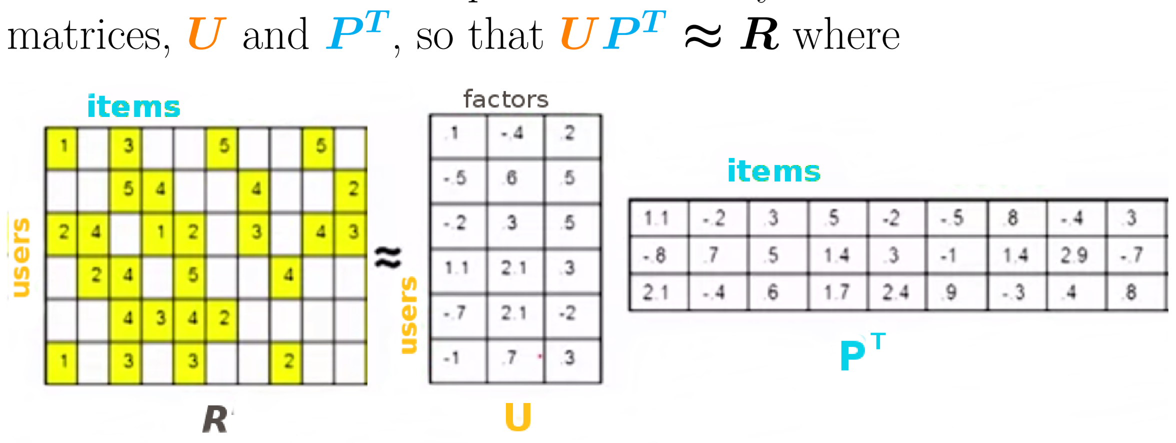
Ratings for m users and n items in $m \times n$ matrix, R.
$R_{i,j}$ = user i's rating of product j.

R =



Matrix Factorization Model [1]

Assume low rank to complete R. R may be factored into two matrices, U and P^T, so that $UP^T \approx R$ where



$U_{i,k}$ = user i's association with latent factor k

$P_{j,k}$ = item j's association with latent factor k.

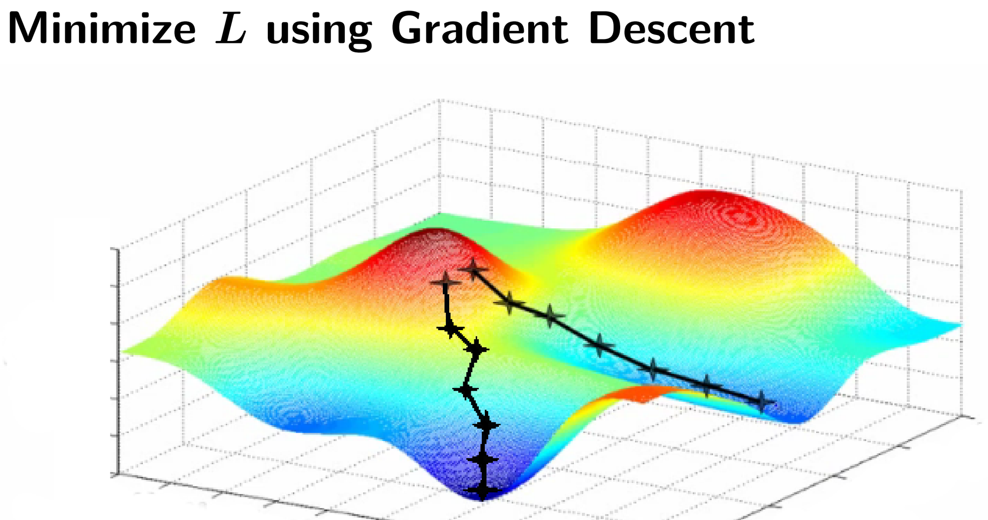
$p_j^T u_i$ = User-Item interaction.

- **Objective Function**

  Let $\omega = \{(i,j) :$ user i has rated item $j\}$.

  $$L(P,U) = \sum_{(i,j)\in\omega}(R_{i,j} - p_j^T u_i)^2 + \lambda(\|p_j\|^2 + \|u_i\|^2)$$

- **Minimize L using Gradient Descent**



Andrew NG
Stanford Machine Learning Online Lectures

User Cold Start

User Cold Start Problem

How to predict ratings for new users in the MF framework?

Functional Matrix Factorization [2]

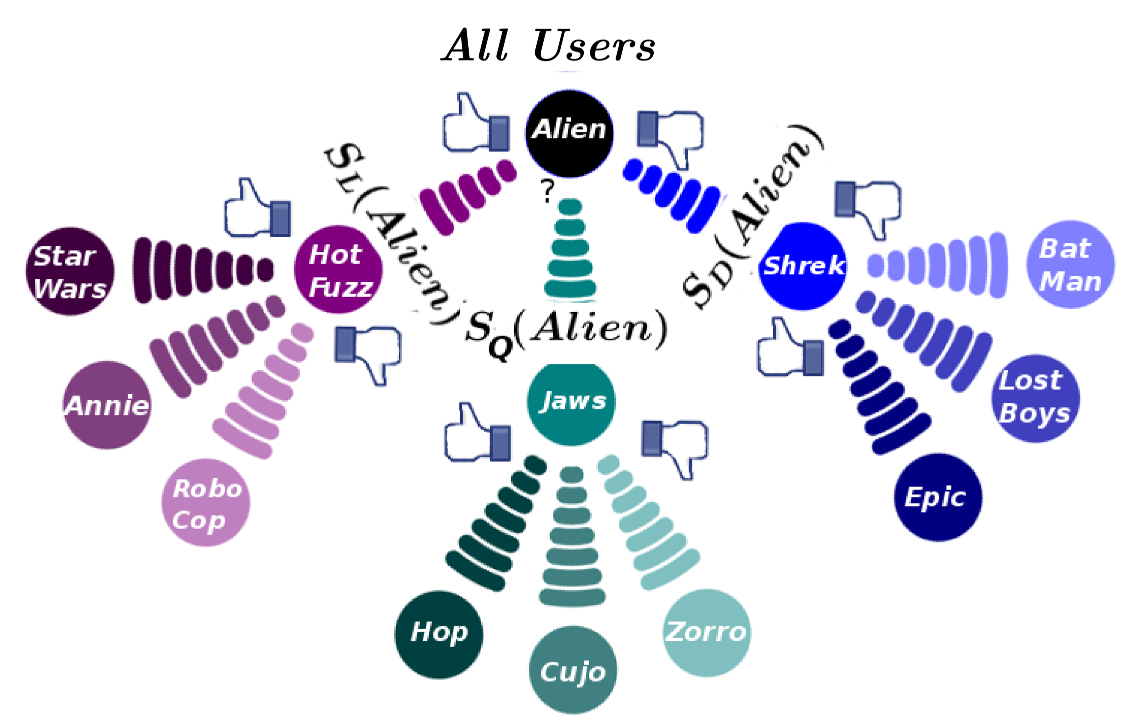
- Hybrid MF-Decision Tree Recommender System
- Uses MF to find best queries to estimate user profiles
- **Recommender Decision Trees**
  - Query new users about preferences for important items
  - a_i = user i's set of responses to queries.
  - $T(a_i) \approx u_i$ (function for approx. user profiles)
- **Objective Function**

  $$L(P,T) = \sum_{(i,j)\in\omega}(R_{i,j} - p_j^T T(a_i))^2 + \lambda\|p_j\|^2)$$

- **Fix T and Optimize w.r.t. P**

  $$\forall j, p_j = \Big(\sum_{(i,j)\in\omega} T(a_i)T(a_i)^T + \lambda I\Big)^{-1}\Big(\sum_{(i,j)\in\omega} R_{i,j}T(a_i)\Big)$$

- **Fix P and Optimize w.r.t. T**
  - Starting at root, partition users into sets by query h:



- **Find Optimal Profiles for users in Child Nodes**

  Let γ be a variable ranging over subscripts L, D, Q.

  $$u_\gamma = \underset{u}{\mathrm{argmin}} \sum_{i\in S_\gamma(h)}\sum_{(i,j)\in\omega}(R_{i,j} - p_j^T u)$$

- **Find Queries to Optimize Profiles**

  $$\min_h \sum_{i\in S_L(h)}\sum_{(i,j)}(R_{i,j} - p_j^T u_L) + \sum_{i\in S_Q(h)}\sum_{(i,j)}(R_{i,j} - p_j^T u_Q) + \sum_{i\in S_D(h)}\sum_{(i,j)}(R_{i,j} - p_j^T u_D)$$

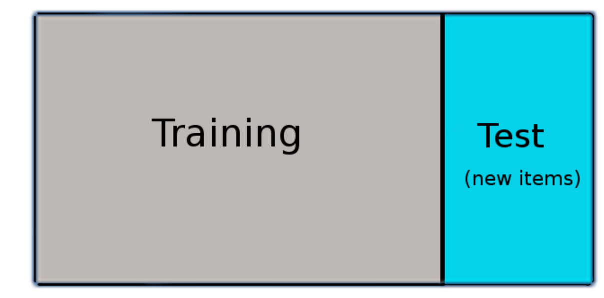
User Cold Start Summary

For new user z, use decision tree and queries to derive user profile u_z. User z's predicted rating for item j is then $p_j^T u_z$.

Item Cold Start

Item Cold Start Problem

How to recommend new unrated items in MF framework?

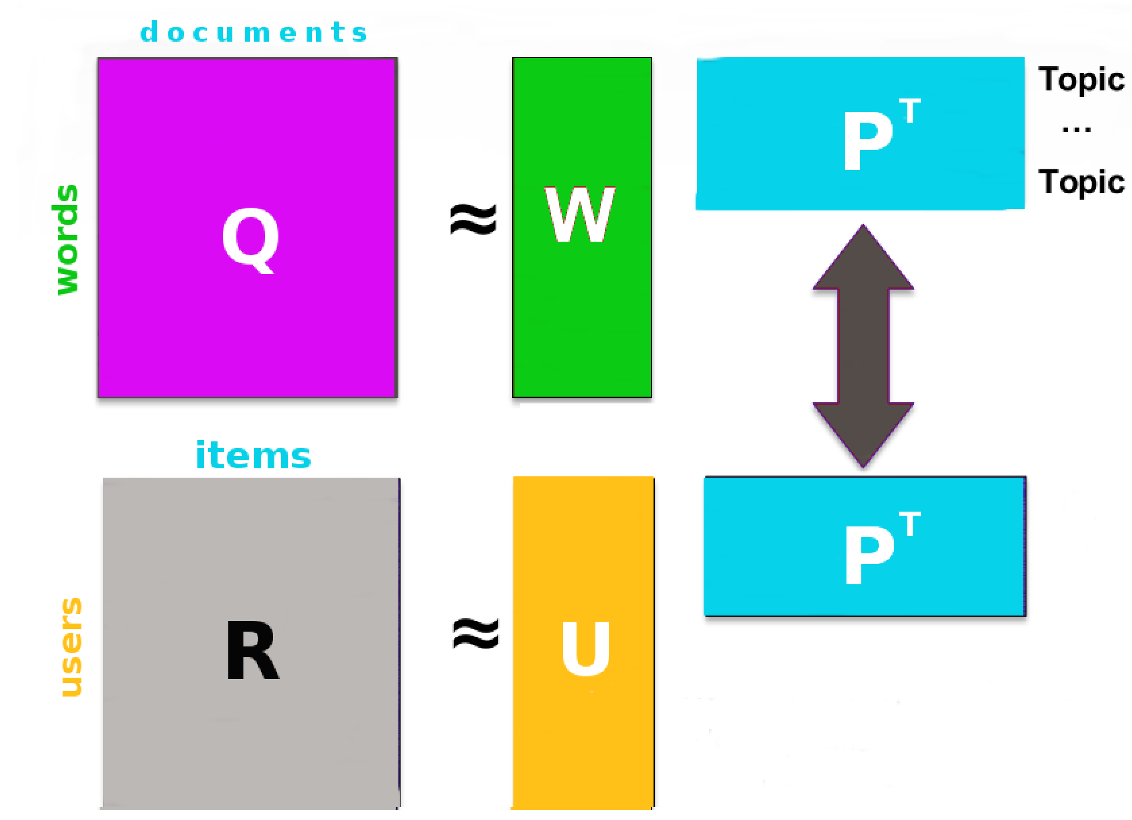


Local Collective Embeddings [3]

- Uses collective factorization to link user ratings to item text.
- **Collective Factorization**

  Document j is a collection of texts associated with item j.

  $Q_{i,j} = 1$ if word i occurs in document j, 0 otherwise.



- **Exploiting Locality**
  - If two columns in document term matrix, Q are similar, the corresponding pair in matrix P should be close.
  - Measurement of local smoothness for factor P:

  $$S = \frac{1}{2}\sum_{(i,j)}\|p_i - p_j\|_F^2 \frac{\mathrm{q_i} \cdot \mathrm{q_j}}{\|\mathrm{q_i}\|\|\mathrm{q_j}\|}$$

- **Objective Function**

  $$L(P,U,W) = \frac{1}{2}[\alpha\sum_{(i,j)\in\omega}(R_{i,j} - p_j^T u_i)^2]$$
  $$+\frac{1}{2}[(1-\alpha)\sum_{(h,j)}(Q_{h,j} - p_j^T w_h)^2] + \beta S$$
  $$\text{s.t. } W \geq 0,\ P \geq 0$$

- **Optimize L using Multiplicative Updates**

Item Cold Start Summary

For new item z and document-term vector q_z, derive item profile p_z by solving $Wp_z = q_z$. User i's estimated rating for item z is then $q_z^T u_i$.

Current Research

- Benchmark state of the art methods
- Implement and refine new models for item cold start.
- Explore models which handle both user and item cold start.
- Explore metrics for Coldish Start (when users or items have some but fewer ratings)

Datasets

Amazon Movies Dataset (Leskovec, McAuley. 2013)

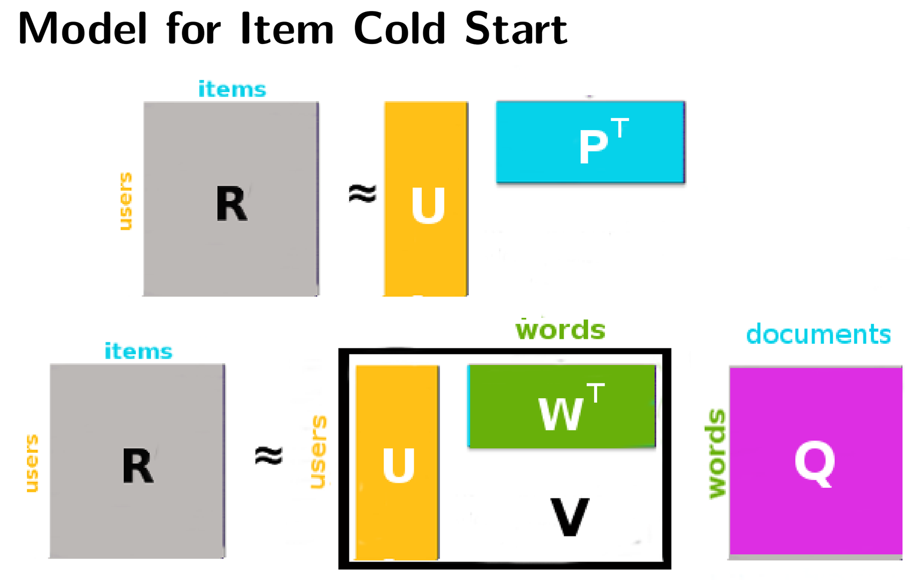
- 7,850,072 reviews (text & ratings)
- Plain text descriptions for all reviewed items

MovieLens Datasets (http://movielens.org)

- 20 million movie reviews
- Reviewed movies can be paired with plot descriptions from the Internet Movie Database (http://www.imdb.com/)
- Smaller Movielens dataset (100,000 reviews) has simple user demographic info (age, gender, occupation, zip)

Model

- **Base Model for Item Cold Start**



- **Objective Function**

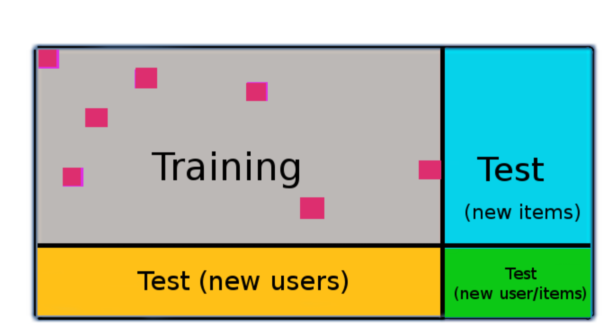
  Let, $\Omega_{i,j} = 1$ if $(i,j) \in \omega$, 0 otherwise. Then

  $$L(U,W,Q) = \mathrm{Tr}((R - U^T P)^T \Omega (R - U^T P))$$
  $$+\mathrm{Tr}((R - UW^T Q)^T \Omega (R - UW^T Q))$$

Model Summary

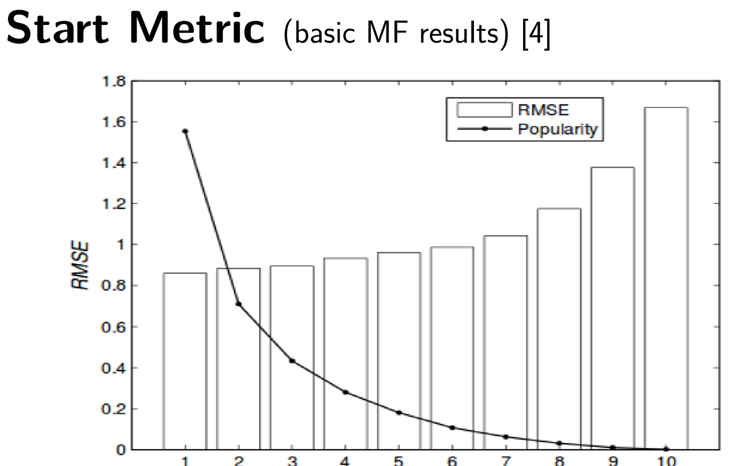
The model learns user i's user-term vector v_i. User i's estimated rating for new or established item z is then $v_i \cdot q_z$.

Train and Test Split



Test set for users and items in "known" ratings set

Coldish Start Metric (basic MF results) [4]



[1] Koren, Yehuda, and Robert Bell. "Advances in collaborative filtering.". 2012
[2] Zhou, Ke, Shuang-Hong Yang, and Hongyuan Zha. "Functional matrix factorizations for cold-start recommendation.". 2011.
[3] Saveski, Martin, and Amin Mantrach. "Item cold-start recommendations: learning local collective embeddings." ACM, 2014.
[4] Zhang, Mi, et al. "Addressing cold start in recommender systems: A semi-supervised co-training algorithm." ACM, 2014.