A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks

Mohsen Jamali
Martin Ester
Introduction

- Recommender Systems are systems that help users make decisions
  - What book to buy (Amazon)
  - What movie to watch (Netflix)
  - What music to listen to (Spotify)
- Can be Personalized or Non-Personalized
- Two types:
  - Memory-based recommenders
  - Model-based recommenders
Memory-Based recommenders: Collaborative Filtering

- Maintain a database of many users’ ratings of a variety of items.
- For a given user, find other similar users whose ratings strongly correlate with the current user.
- Recommend items rated highly by these similar users, but not rated by the current user.
- Almost all existing commercial recommenders use this approach (e.g. Amazon)

\[
P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} w_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} w_{a,u}}
\]
Model based recommenders & Social networks

- Learn the parameters of a model and store only those parameters
- Very fast once the parameters are learnt
- Memory based recommenders have to explore the rating matrix each time during testing
- Training bottleneck
- Matrix Factorization
Social Networks in Recommendation

- Makes recommendations for users based on ratings of users who have direct or indirect relations with the given user
- People tend to relate to people with similar attributes
- Due to social influence, related people in a social network influence each other and become more similar
- Cold start users: users who haven’t rated much items
  - Similarity based approaches fail
  - Social network based approach works as long as user is connected to others

Figure 1: A sample social rating network.
Matrix Factorization & Dimensionality Reduction

- Ratings matrices can have thousands, millions of dimensions
- Ratings Matrix is an overfit representation of user taste and item descriptions
  - King Lear, Othello, Macbeth
- High computational complexity, potentially poor results
- Find a more compact representation of user tastes and item descriptions
- Keyword vectors in information retrieval
Singular Value Decomposition

- Reduce space to smaller representation that is compact and robust
  \[ R = M \Sigma U^T \]
- SVD is an algorithm that takes the matrix \( R \) as an input, and it gives you \( M, \Sigma \) and \( U \), such that:
  - \( R \) is an \( m \times n \) ratings matrix
  - \( M \) is \( m \times k \) user-feature affinity matrix
  - \( U \) is \( n \times k \) item-feature relevance matrix
  - \( \Sigma \) is \( k \times k \) diagonal feature weight matrix
- Features are optimized for predictive power
  - Not necessarily interpretable
Singular Value Decomposition

- Truncated SVD is the best rank $k$ approximation

\[ r_{ui} = p_u \cdot q_i = \sum_{f \text{ latent factors}} \text{affinity of } u \text{ for } f \times \text{affinity of } i \text{ for } f \]

<table>
<thead>
<tr>
<th></th>
<th>Alice</th>
<th>Bob</th>
<th>Titanic</th>
<th>Toy Story</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>10%</td>
<td>50%</td>
<td>20%</td>
<td>30%</td>
</tr>
<tr>
<td>Comedy</td>
<td>+10%</td>
<td>+30%</td>
<td>+00%</td>
<td>+60%</td>
</tr>
<tr>
<td>Romance</td>
<td>+50%</td>
<td>+10%</td>
<td>+70%</td>
<td>+00%</td>
</tr>
</tbody>
</table>

\[ p_{Alice} = (10\%, 10\%, 50\%, \cdots) \]
\[ p_{Bob} = (50\%, 30\%, 10\%, \cdots) \]
\[ q_{Titanic} = (20\%, 00\%, 70\%, \cdots) \]
\[ q_{Toy Story} = (30\%, 60\%, 00\%, \cdots) \]
Probabilistic Matrix Factorization

- Alternative to linear algebra approach
- Assume that data is generated by a random process with known structure
- Learn parameters that would generate data that looks like what you have
- Model ratings are drawn from normal distributions
  - Mean determined by user and item via features
  - Gaussian Mixture Model
Probabilistic Matrix Factorization

- Learn the latent characteristics of users and items and predict the unknown ratings using these latent characteristics
- \( U \in \mathbb{R}^{K \times N} \) and \( V \in \mathbb{R}^{K \times M} \) are latent user and item feature matrices
- The conditional probability of the observed ratings is defined as:

\[
p(R|U, V, \sigma_R^2) = \prod_{u=1}^{N} \prod_{i=1}^{M} \mathcal{N}\left(R_{u,i} | g(U_u^T V_i), \sigma_R^2 \right) \]_{u,i}'

- Zero mean Gaussian priors are assumed for user and item feature vectors:

\[
p(U|\sigma_U^2) = \prod_{u=1}^{N} \mathcal{N}(U_u|0, \sigma_U^2 I),
\]

\[
p(V|\sigma_V^2) = \prod_{i=1}^{M} \mathcal{N}(V_i|0, \sigma_V^2 I)
\]
Probabilistic Matrix Factorization

- Through a Bayesian inference, the posterior probability of the latent variables $U$ and $V$ can be obtained as follows:

\[
p(U, V | R, T, \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto p(R | U, V, \sigma_R^2)p(U | \sigma_U^2)p(V | \sigma_V^2)
\]

\[
= \prod_{u=1}^{N} \prod_{i=1}^{M} \left[ \mathcal{N}(R_{u,i} | g(U_u^T V_i), \sigma_R^2) \right] I_{u,i}^R
\times \prod_{u=1}^{N} \mathcal{N}(U_u | 0, \sigma_U^2 I) \times \prod_{i=1}^{M} \mathcal{N}(V_i | 0, \sigma_V^2 I)
\]  

(3)
Social Influence

- Due to social influence, the behavior of a user $u$ is affected by his direct neighbors $N_u$
- The latent feature vector of $u$ is dependent on the latent feature vectors of all his direct neighbors $v \in N_u$

\[
\hat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}} = \frac{\sum_{v \in N_u} T_{u,v} U_v}{|N_u|}
\]

- We normalize each row of the trust matrix
SocialMF Model

- Approach to incorporate trust propagation into a matrix factorization
- Conditional Probability of observed ratings remains the same as PMF
- For the user latent features, we have two factors:
  - The zero-mean Gaussian prior to avoid over-fitting
  - The conditional distribution of user latent features given the latent features of his direct neighbors

$$p(U|T, \sigma_U^2, \sigma_T^2) \propto p(U|\sigma_U^2) \times p(U|T, \sigma_T^2)$$

$$= \prod_{u=1}^{N} \mathcal{N}(U_u|0, \sigma_U^2) \times \prod_{u=1}^{N} \mathcal{N}(U_u| \sum_{v \in N_u} T_{u,v} U_v, \sigma_T^2)$$
SocialMF Model

- Posterior probability of latent feature vectors given the rating and social trust matrices

\[
p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) \propto \]
\[
p(R | U, V, \sigma_R^2)p(U | T, \sigma_U^2, \sigma_T^2)p(V | \sigma_V^2)
\]
\[
= \prod_{u=1}^{N} \prod_{i=1}^{M} \left[ \mathcal{N} \left( R_{u,i} | g(U_u^T V_i), \sigma_r^2 \right) \right] I_{u,i}^{R}
\]
\[
\times \prod_{u=1}^{N} \mathcal{N} \left( U_u | \sum_{v \in N_u} T_{u,v} U_v, \sigma_T^2 \mathbf{I} \right)
\]
\[
\times \prod_{u=1}^{N} \mathcal{N} \left( U_u | 0, \sigma_U^2 \mathbf{I} \right) \times \prod_{i=1}^{M} \mathcal{N} \left( V_i | 0, \sigma_V^2 \mathbf{I} \right)
\]
SocialMF Model (Maximal Likelihood Estimation)

\[
\mathcal{L}(R, T, U, V) = \frac{1}{2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i} - g(U_{u}^{T}V_{i}))^2 \\
\quad + \frac{\lambda_U}{2} \sum_{u=1}^{N} U_{u}^{T}U_{u} + \frac{\lambda_V}{2} \sum_{i=1}^{M} V_{i}^{T}V_{i} \\
\quad + \frac{\lambda_T}{2} \sum_{u=1}^{N} \left( (U_{u} - \sum_{v \in N_u} T_{u,v}U_{v})^{T}(U_{u} - \sum_{v \in N_u} T_{u,v}U_{v}) \right) \tag{12}
\]

In the above equation, \( \lambda_U = \sigma_R^2 / \sigma_U^2 \), \( \lambda_V = \sigma_R^2 / \sigma_V^2 \), and \( \lambda_T = \sigma_T^2 / \sigma_R^2 \).

\[
\frac{\partial \mathcal{L}}{\partial U_{u}} = \sum_{i=1}^{M} I_{u,i}^{R} V_{i} g'(U_{u}^{T}V_{i})(g(U_{u}^{T}V_{i}) - R_{u,i}) + \lambda_U U_{u} \\
\quad + \lambda_T (U_{u} - \sum_{v \in N_u} T_{u,v}U_{v}) - \lambda_T \sum_{\{v|u \in N_v\}} T_{v,u} \left( U_{v} - \sum_{u \in N_v} T_{v,u}U_{u} \right) \tag{13}
\]

\[
\frac{\partial \mathcal{L}}{\partial V_{i}} = \sum_{u=1}^{N} I_{u,i}^{R} U_{u} g'(U_{u}^{T}V_{i})(g(U_{u}^{T}V_{i}) - R_{u,i}) + \lambda_V V_{i} \tag{14}
\]
STE Model

- Matrix factorization approach for social network based recommendation
- Linear combination of basic matrix factorization approach and a social network based approach
- The predicted rating of user $u$ on item $i$ is as follows:

$$
\hat{R}_{u,i} = g(\alpha U_u^T V_i + (1 - \alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i)
$$

- The feature vectors of direct neighbors of $u$ affect the ratings of $u$ instead of affecting the feature vector of $u$
- Does not handle trust propagation
Desirable Properties

- Addresses the transitivity of trust in social networks.
  - STF does not support trust propagations
- Does not just depend on observed ratings
  - Despite not having any expressed ratings, the feature vectors of these users will be learnt to be close to their neighbors
  - Addresses Cold start issue for users with few ratings
- Faster than STF
Datasets

- **Flixster**
  - Social Networking service where users can rate movies
  - Can also add users to friend list

- **Epinions**:
  - Public social rating dataset
  - Items from different categories such as cameras, dvd players, music

Table 1: General statistics of the Flixster and Epinions

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Flixster</th>
<th>Epinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>1M</td>
<td>71K</td>
</tr>
<tr>
<td>Social Relations</td>
<td>26.7M</td>
<td>508K</td>
</tr>
<tr>
<td>Ratings</td>
<td>8.2M</td>
<td>575K</td>
</tr>
<tr>
<td>Items</td>
<td>49K</td>
<td>104K</td>
</tr>
<tr>
<td>Users with Rating</td>
<td>150K</td>
<td>47K</td>
</tr>
<tr>
<td>Users with Friend</td>
<td>980K</td>
<td>60K</td>
</tr>
</tbody>
</table>
Experimental Results

- Evaluation criteria: RMSE

\[
RMSE = \sqrt{\frac{\sum_{(u,i) \in R_{test}} (r_{u,i} - \hat{r}_{u,i})^2}{|R_{test}|}}
\]

Table 2: RMSE values for comparison partners on Epinions with different settings of dimensionality K.

<table>
<thead>
<tr>
<th>Method</th>
<th>K=5</th>
<th>K=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>1.180</td>
<td>1.180</td>
</tr>
<tr>
<td>BaseMF</td>
<td>1.175</td>
<td>1.195</td>
</tr>
<tr>
<td>STE</td>
<td>1.145</td>
<td>1.150</td>
</tr>
<tr>
<td>SocialMF</td>
<td>1.075</td>
<td>1.085</td>
</tr>
</tbody>
</table>

Table 3: RMSE values for comparison partners on Flixster with different settings of dimensionality K.

<table>
<thead>
<tr>
<th>Method</th>
<th>K=5</th>
<th>K=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>0.911</td>
<td>0.911</td>
</tr>
<tr>
<td>BaseMF</td>
<td>0.878</td>
<td>0.863</td>
</tr>
<tr>
<td>STE</td>
<td>0.864</td>
<td>0.852</td>
</tr>
<tr>
<td>SocialMF</td>
<td>0.821</td>
<td>0.815</td>
</tr>
</tbody>
</table>
Impact of $\lambda_t$ on results

- $\lambda_t$ controls the influence of social network on behavior of users
  - Large $\lambda_t$ : more impact from neighbors
  - Small $\lambda_t$ : Close to baseline MF
Cold Start Performance and Runtime

- In both Flixster and Epinions more than 50% of users are cold start users

<table>
<thead>
<tr>
<th>Method</th>
<th>Epinions</th>
<th>Flixster</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>1.361</td>
<td>1.228</td>
</tr>
<tr>
<td>BaseMF</td>
<td>1.352</td>
<td>1.213</td>
</tr>
<tr>
<td>STE</td>
<td>1.295</td>
<td>1.152</td>
</tr>
<tr>
<td>SocialMF</td>
<td>1.159</td>
<td>1.057</td>
</tr>
</tbody>
</table>

**Table 4:** RMSE values on cold start users (K=5).

<table>
<thead>
<tr>
<th>Model</th>
<th>Epinions</th>
<th>Flixster</th>
</tr>
</thead>
<tbody>
<tr>
<td>SocialMF</td>
<td>40 min</td>
<td>5.5 hr</td>
</tr>
<tr>
<td>STE</td>
<td>5 hr</td>
<td>9 days</td>
</tr>
</tbody>
</table>

**Table 6:** Total time required to learn the parameters of models.
Conclusions

- Matrix factorization, both probabilistic and linear algebra based, are core to many recommendation algorithms.
- Integrating trust from social networks enables better recommendations, especially for cold start users.
- Model does not consider negative trust.
THANK YOU