Social Regularization

SEMINAR RECOMMENDER SYSTEMS - 10.01.17

Maurício Camargo / Guilherme Holdack / Zafar Mahmood
Agenda

- Introduction to Social Recommendation
- Social Recommendation Methods
- SoRec: Social Recommendation Using Probabilistic Matrix Factorization
- SocialMF: A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks
- RSR: Recommended Systems With Social Regularization
- Conclusions and Comparison
1. Intro to Social Recommendation

Definition

• Traditional recommender systems assume that users are independent and identically distributed (i.i.d. assumption);

• However, online users are inherently connected via various types of relations such as friendships and trust relations;

• Users in social recommender systems are connected, providing social information.
1. Intro to Social Recommendation

Narrow x Broad Definition

• **Narrow Definition**: any recommendation with online social relations as an additional input, i.e., augmenting an existing recommendation engine with additional social signals.

• **Broad Definition**: recommender systems recommending any objects in social media domains such as items (the focus under the narrow definition), tags, people, and communities.

The *narrow definition* is used in the context of this presentation.
1. Intro to Social Recommendation

Reasons to use

• Connected users are more likely to share similar interests in topics than two randomly selected users;

• In the physical world, we usually ask suggestions from our friends (tend to be similar and also know our tastes);

• Provides an independent source of information about online users (specially useful on Cold Start);

• Exploiting social relations can potentially improve recommendation performance.
1. Intro to Social Recommendation

Representation

• In addition to the rating matrix in traditional recommender systems, there is also a second matrix to map the relations:
2. Social Recommendation Methods

Overview of the Methods

- **Memory based**: for social recommendation, it takes both the rating information and social information to find similar users (ex: TidalTrust, MoleTrust, TrustWalker).

- **Model based**: uses matrix-factorization methods which also take into account the social relations. A unified framework can be stated as:

\[
\min_{U, V, \Omega} \| W \odot (R - U^T V) \|_F^2 + \alpha \cdot Social(T, S, \Omega) + \lambda (\| U \|_F^2 + \| V \|_F^2 + \| \Omega \|_F^2)
\]
2. Social Recommendation Methods

Model Based Methods

• **Co-factorization methods**: performs a co-factorization in the user-item matrix and the user-user social relation matrix by sharing the same user preference latent factor (Ex: SoRec and LOCABAL).

• **Ensemble methods**: a missing rating for a given user is predicted as a linear combination of ratings from the user and the social network (Ex: STE, mTrust).

• **Regularization methods**: For a given user, regularization methods force his preference to be closer to that of users in his social network. (Ex: SocialMF and Social Regularization).
SoRec: Social Recommendation Using Probabilistic Matrix Factorization

Maurício Camargo
Motivation

Problems with current recommender systems:

• Ignores the social interactions or connections among users;

• Bad results on users who have made very few ratings or even none at all;

• Some existing approaches fail to handle very large datasets;

In reality, we always turn to friends we trust for movie, music or book recommendations;
Current Scenario

Collaborative Filtering

• Memory Based:
  - user-based and item-based approaches;
  - trust-based recommender systems – also use trust to calculate similarity (does not scale well).

• Model-based:
  - clustering model, aspect models and the latent factor model.
  - considers users independent and identically distributed.

No model-based approach to deal with social relations.
SoRec

Proposed Solution

SoRec (Social Recommendation):

- predict the missing values of the user-item by employing two different data sources.

- factorize the social network graph and user-item matrix simultaneously using $U^T Z$ and $U^T V$

$U$ – low-dimensional user latent feature space

$Z$ – factor matrix in the social network graph

$V$ – low-dimensional item latent feature space
SoRec

How it works

1 - By analysing both the social relations and the ratings, we get two different tables:
How it works

2 – Both resulting tables can be factorized into its latent features:

\[ U_1^{TV} \quad U_2^{TZ} \]

\( U_1 \) and \( U_2 \) – low-dimensional user latent feature space

\( Z \) – factor matrix in the social network graph

\( V \) – low-dimensional item latent feature space
SoRec

How it works

3 – The trick is to force both factorizations to share the same $U$:

$$U^T V$$

Same matrix

$$U^T Z$$

$U$ – low-dimensional user latent feature space

$Z$ – factor matrix in the social network graph

$V$ – low-dimensional item latent feature space

U will be influenced by the user x item ratings AND its social network.
SoRec

How it works

To improve the model:

- trust value should decrease if user i trusts lots of users;
- trust value should be increase if user k is trusted by lots of users.

The original equation becomes:

\[
p(C|U, Z, \sigma_C^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \mathcal{N}\left[\left(c_{ik} g(U_i^T Z_k), \sigma_C^2\right)\right]^{I_{ik}}
\]
SoRec

How it works

For SoRec, the general equation:

$$\min_{U,V,\Omega} \| W \odot (R - U^T V) \|_F^2 + \alpha \ Social(T, S, \Omega) + \lambda (\| U \|_F^2 + \| V \|_F^2 + \| \Omega \|_F^2)$$

$\| \cdot \|_F^2$ - denotes the Frobenius norm

Becomes:

$$\min_{U,V,Z} \| W \odot (R - U^T V) \|_F^2 + \alpha \sum_{i=1}^{n} \sum_{u_k \in N_i} (S_{ik} - u_i^T z_k)^2 + \lambda (\| U \|_F^2 + \| V \|_F^2 + \| Z \|_F^2)$$

$$Social(T, S, \Omega) = \min \sum_{i=1}^{n} \sum_{u_k \in N_i} (S_{ik} - u_i^T z_k)^2$$
SoRec

How it works

Other notation:

\[ \mathcal{L}(R, C, U, V, Z) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R(r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_C}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} I_{ik}^C(c_{ik}^* - g(U_i^T Z_k))^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 + \frac{\lambda_Z}{2} \|Z\|_F^2 \]

In order to reduce the model complexity: \( \lambda U = \lambda V = \lambda Z \)

The minimum can be found through gradient descent:

\[
\begin{align*}
\frac{\partial \mathcal{L}}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})V_j + \lambda_C \sum_{k=1}^{m} I_{ik}^C g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^*)Z_k + \lambda_U U_i \\
\frac{\partial \mathcal{L}}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^R g'(U_i^T V_j)(g(U_i^T V_j) - r_{ij})U_i + \lambda_V V_j, \\
\frac{\partial \mathcal{L}}{\partial Z_k} &= \lambda_C \sum_{i=1}^{m} I_{ik}^C g'(U_i^T Z_k)(g(U_i^T Z_k) - c_{ik}^*)U_i + \lambda_Z Z_k
\end{align*}
\]

\( g'(x) = \frac{\exp(x)}{(1 + \exp(x))^2} \) derivative of the logistic function
Even though user 4 does not rate any items, the approach still can predict reasonable ratings.
**SoRec**

**How it works – pseudocode**

**Input:** The rating information \( r \), the social information \( c \), the number of latent factors \( k \), \( \lambda_C \) and \( \lambda \) (regularization parameters)

**Output:** The user preference matrix \( U \) and the item characteristic matrix \( V \)

1: Initialize \( U \), \( V \) and \( Z \) randomly (with \( k \) factors)

2: while Not convergent do

3: Calculate \( \frac{\partial J}{\partial U} \), \( \frac{\partial J}{\partial V} \) and \( \frac{\partial J}{\partial Z} \)

4: Update \( U \leftarrow U - \gamma u \frac{\partial J}{\partial U} \)

5: Update \( V \leftarrow V - \gamma v \frac{\partial J}{\partial V} \)

6: Update \( Z \leftarrow Z - \gamma z \frac{\partial J}{\partial Z} \)

7: Evaluate LossFunction

\[
L(R, C, U, V, Z) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^r (r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_C}{2} \sum_{i=1}^{m} \sum_{k=1}^{n} I_{ik}^c (c_{ik} - g(U_i^T Z_k))^2 + \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2 + \|Z\|_F^2)
\]

8: end while
Complexity Analysis

\[ \mathcal{L} = O(\rho_R l + \rho_C l) \]

Where \( \rho_R \) and \( \rho_C \) are the numbers of nonzero entries in matrices \( R \) and \( C \).

\[ \frac{\partial \mathcal{L}}{\partial U} = O(\rho_R l + \rho_C l) \]
\[ \frac{\partial \mathcal{L}}{\partial V} = O(\rho_R l) \]
\[ \frac{\partial \mathcal{L}}{\partial Z} = O(\rho_C l) \]

Total computational complexity in one iteration is: \( O(\rho_R l + \rho_C l) \)

Computational time of the method is linear with respect to the number of observations in the two sparse matrices. Thus, the approach can scale on large datasets.
Epinions was selected as the data source

- well known knowledge sharing site and review site.
- Users submit their opinions on topics such as products, companies, movies, or reviews issued by other users.
- Users can also assign products or reviews integer ratings from 1 to 5.
- Members maintain a “trust” and a “block (distrust)” list
- 40,163 users who have rated at least one of a total of 139,529 different items. The total number of reviews is 664,824

\[
\text{Density} = \frac{664824}{40163 \times 139529} = 0.01186\%. \text{ (very sparse)}
\]
Experimental Analysis

Epinions was selected as the data source.

Figure 3: Degree Distribution of User Social Network

<table>
<thead>
<tr>
<th>Statistics</th>
<th>User</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min. Num. of Rated</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Max. Num. of Rated</td>
<td>1022</td>
<td>2018</td>
</tr>
<tr>
<td>Avg. Num. of Rated</td>
<td>16.55</td>
<td>4.76</td>
</tr>
</tbody>
</table>
Experimental Analysis
Comparison to other methods

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Dimensionality = 5</th>
<th>Dimensionality = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMMF</td>
<td>PMF</td>
</tr>
<tr>
<td>90%</td>
<td>1.008</td>
<td>0.9971</td>
</tr>
<tr>
<td>80%</td>
<td>1.0371</td>
<td>1.0277</td>
</tr>
<tr>
<td>50%</td>
<td>1.1147</td>
<td>1.0972</td>
</tr>
<tr>
<td>20%</td>
<td>1.2532</td>
<td>1.2397</td>
</tr>
</tbody>
</table>

MMF - Maximum Margin Matrix Factorization
PMF - Probabilistic Matrix Factorization
CPMF - Constrained Probabilistic Matrix Factorization

On average, the approach improves the accuracy by 11.01%, 9.98%, and 7.82% relative to MMMF, PMF and CPMF, respectively.
Experimental Analysis

Impact of $\lambda_C$

Figure 4: Impact of Parameter $\lambda_C$
Experimental Analysis

Performance on Different Users

(a) Performance Comparison on Different User Rating Scales (99% as Training Data)

(b) Distribution of Testing Data (96% as Training Data)
Experimental Analysis

Efficiency Analysis

Figure 6: Efficiency Analysis
Conclusion and Future Work

Conclusion

• Experimental results: the approach outperforms the other state-of-the-art collaborative filtering algorithms.
• Complexity analysis: it is scalable to very large datasets.
• Can also be used to predict connections on social network.

Future Work:

• Investigate whether the distrust information is useful to increase the prediction quality, and how to incorporate it.
• Consider the diffusion process between users.
References


Social Recommendation: A Review Jiliang Tang · Xia Hu · Huan Liu
https://pdfs.semanticscholar.org/fff4/4f028044dd6ee79b7c9c26a90a23dc8d4438.pdf

MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS
https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf

A MATRIX FACTORIZATION TECHNIQUE WITH TRUST PROPAGATION FOR RECOMMENDATION IN SOCIAL NETWORKS.
Social MF
INTEGRATING TRUST PROPAGATION
OUTLINE

DEFINITION
 RELATED WORK
 PSEUDOCODE
 DATASETS
 EXPERIMENTS
 CONCLUSION + FUTURE WORK

what is it?
OUTLINE

DEFINITION

RELATED WORK

PSEUDOCODE

DATASETS

EXPERIMENTS

CONCLUSION + FUTURE WORK

who are the competitors?
OUTLINE

DEFINITION
RELATED WORK
PSEUDOCODE
DATASETS
EXPERIMENTS
CONCLUSION + FUTURE WORK

Let's have a look on it!
OUTLINE

DEFINITION
RELATED WORK
PSEUDOCODE
DATASETS
EXPERIMENTS
CONCLUSION + FUTURE WORK

which data are we dealing with?
OUTLINE

DEFINITION
RELATED WORK
PSEUDOCODE
DATASETS
EXPERIMENTS
CONCLUSION + FUTURE WORK

how does it behave?
OUTLINE

DEFINITION
RELATED WORK
PSEUDOCODE
DATASETS
EXPERIMENTS

CONCLUSION + FUTURE WORK

where to go from here?
quick reminder

MATRIX FACTORIZATION
MATRIX FACTORIZATION

ONE OF THE MOST COMMON TECHNIQUES FOR MODEL BASED RECOMMENDATION.
MATRIX FACTORIZATION

ONE OF THE MOST COMMON TECHNIQUES FOR MODEL BASED RECOMMENDATION.

LEARNs LATENT FEATURES FOR BOTH USERS AND ITEMS.
MATRICES FACTORIZATION

model and conditional probability

\[ p(R|U, V, \sigma^2_R) = \prod_{u=1}^{N} \prod_{i=1}^{M} \mathcal{N}(R_{u,i} | g(U_u^T V_i), \sigma_r^2) \]

EACH RATING IS NORMALLY DISTRIBUTED AROUND THE PRODUCT OF BOTH, USERS AND ITEMS, FEATURE VECTORS - CONSIDERING STANDARD DEVIATION.
social relations
WHY PROPAGATING TRUST
WHAT IS IT ALL ABOUT?

direct neighbours
WHAT IS IT ALL ABOUT?

social influence
WHAT IS IT ALL ABOUT?

social relations
related work
PAPERS “COVERING” SAME TOPIC
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOGATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOGATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE
Finds all raters with the shortest path distance from the source user and aggregates their ratings weighted by the trust between the source user and these raters.
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOCATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE
Similar to TidalTrust, but receives a parameter called maximum-depth. This way, only raters connected up to a maximum degree are considered.
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOCATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE
RECEIVES AN INTEGER INPUT, WHICH IS THE NUMBER OF MEMBERS TO TRUST. THIS NUMBER IS INDEPENDENT OF USERS OR ITEMS, SO IT IS NOT AN APPROPRIATE APPROACH FOR TRUST-BASED RECOMMENDATION.
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOGATO
TRUSTWALKER
SOCIAL TRUST ENSEMBLER - STE
Focuses on trust-based and item-based recommendations. There is a probability of considering the rating of a similar item instead of the rating for the target item itself, depending on the length of the walk.
RELATED WORK

TIDAL TRUST
MOLE TRUST
ADVOGATO
TRUSTWALKER

SOCIAL TRUST ENSEMBLER - STE
Partner used as comparison to this paper. This model affects the ratings of a given user, making use of the feature vectors of the direct neighbours; it does not handle trust propagation, as it does not affect the feature vectors of the target user.
OR...
\[ \hat{R}_{u,i} = g\left(\alpha U_u^T V_i + (1 - \alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i \right) \]
Pseudocode

LET'S ANALYSE IT AND THEN... SEE SOME CODING..!
INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.
INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.
INCORPORATING TRUST

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

THE PROPOSED EQUATION:

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

THE BEHAVIOUR OF A GIVEN USER IS AFFECTED BY THE INFLUENCE OF HIS DIRECT NEIGHBOURS.

THE PROPOSED EQUATION:

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

all social networks analysed in this paper are binary, so: \( T_{u,v} \) can only be zero or one!
AND, HAVING ALL ROWS OF THE TRUST MATRIX NORMALIZED:
THE SUM OF ALL “TRUSTS” IN A ROW WOULD BE:  \[ \sum_{v=1}^{N} T_{u,v} = 1 \]

THIS WAY:

\[ \hat{U}_u = \frac{\sum_{v \in N_u} T_{u,v} U_v}{\sum_{v \in N_u} T_{u,v}} \]
The sum of all “trusts” in a row would be: \[ \sum_{v=1}^{N} T_{u,v} = 1 \]

This way:

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

becomes

\[ \hat{U}_u = \sum_{v \in N_u} T_{u,v} U_v \]
So it becomes easy to conclude, that the estimate of the latent features vector of a given user is nothing more than the weighted average of the latent feature vectors of his direct neighbours!
THE ESTIMATED FEATURE VECTOR OF A GIVEN USER CAN BE INFERRED THIS WAY:

\[
\begin{pmatrix}
\hat{U}_{u,1} \\
\hat{U}_{u,2} \\
\vdots \\
\hat{U}_{u,K}
\end{pmatrix}
= 
\begin{pmatrix}
U_{1,1} & U_{2,1} & \cdots & U_{N,1} \\
U_{1,2} & U_{2,2} & \cdots & U_{N,2} \\
\vdots & \vdots & \ddots & \vdots \\
U_{1,K} & U_{2,K} & \cdots & U_{N,K}
\end{pmatrix}
\begin{pmatrix}
T_{u,1} \\
T_{u,2} \\
\vdots \\
T_{u,N}
\end{pmatrix}
\]
The estimated feature vector of a given user can be inferred this way:

\[
\begin{pmatrix}
\hat{U}_{u,1} \\
\hat{U}_{u,2} \\
\vdots \\
\hat{U}_{u,K}
\end{pmatrix} =
\begin{pmatrix}
U_{1,1} & U_{2,1} & \ldots & U_{N,1} \\
U_{1,2} & U_{2,2} & \ldots & U_{N,2} \\
\vdots & \vdots & \ddots & \vdots \\
U_{1,K} & U_{2,K} & \ldots & U_{N,K}
\end{pmatrix}
\begin{pmatrix}
T_{u,1} \\
T_{u,2} \\
\vdots \\
T_{u,N}
\end{pmatrix}
\]

Question to the audience:
hey, audience! what's the main difference between STE and Social MF, by now?
AGAIN...
AGAIN...
\[ \hat{R}_{u,i} = g(\alpha U_u^T V_i + (1 - \alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i) \]
AND SOCIAL MF?
WAIT. NOT SO FAST!
WAIT. NOT SO FAST!

Let's get some maths done!
\[ p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) \propto \]
\[ p(R | U, V, \sigma_R^2)p(U | T, \sigma_U^2, \sigma_T^2)p(V | \sigma_V^2) \]
\[ = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[ \mathcal{N}\left( R_{u,i} | g(U_i^T V_i), \sigma_r^2 \right) \right] I_{u,i}^{R_u} \]
\[ \times \prod_{u=1}^{N} \mathcal{N}\left( U_u | \sum_{v \in N_u} T_{u,v} U_v, \sigma_T^2 \mathbf{I} \right) \]
\[ \times \prod_{u=1}^{N} \mathcal{N}\left( U_u | 0, \sigma_U^2 \mathbf{I} \right) \times \prod_{i=1}^{M} \mathcal{N}\left( V_i | 0, \sigma_V^2 \mathbf{I} \right) \]
\[ p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) \propto \]
\[ p(R | U, V, \sigma_R^2)p(U | T, \sigma_U^2, \sigma_T^2)p(V | \sigma_V^2) \]
\[ = \prod_{u=1}^{N} \prod_{i=1}^{M} \left[ N \left( R_{u,i} | g(U_u^T V_i), \sigma_r^2 \right) \right]^{I_{u,i}} \]

**THE PRODUCT OF THE PROBABILITIES OF OBSERVED RATINGS ARE THE SAME AS IN MATRIX FACTORIZATION.**
NORMAL PROBABILITY OVER THE LATENT USER FEATURES INFLUENCED BY THE LATENT FEATURES OF THE DIRECT NEIGHBORS.
YES, RATING VALUES ARE DEPENDENT ON LATENT FEATURES OF USERS AND ITEMS. BUT, REMEMBER... LATENT FEATURES OF USERS ARE INFLUENCED BY DIRECT NEIGHBOURS!
AND ALSO...
BEING THE OBJECTIVE FUNCTION STATED BEFORE:

\[ \hat{R}_{u,i} = g(\alpha U_u^T V_i + (1 - \alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i) \]
A GRADIENT DESCENT CAN BE PERFORMED TO FIND A LOCAL MINIMUM FOR ALL USERS AND ITEMS:

\[
\frac{\partial \mathcal{L}}{\partial U_u} = \sum_{i=1}^{M} I_{u,i} V_i g'(U_u^T V_i) (g(U_u^T V_i) - R_{u,i}) + \lambda_U U_u + \lambda_T (U_u - \sum_{v \in N_u} T_{u,v} U_v) - \lambda_T \sum_{\{v | u \in N_v\}} T_{v,u} (U_v - \sum_{w \in N_v} T_{v,w} U_w) \\
\frac{\partial \mathcal{L}}{\partial V_i} = \sum_{u=1}^{N} I_{u,i} V_i g'(U_u^T V_i) (g(U_u^T V_i) - R_{u,i}) + \lambda_V V_i
\]

in the experiments:

\[
\lambda_U = \frac{\sigma^2_U}{\sigma^2_R}, \lambda_V = \frac{\sigma^2_V}{\sigma^2_V} \\
\lambda_T = \frac{\sigma^2_T}{\sigma^2_T} \\
\lambda_U = \lambda_V \\
g'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}.
\]
**PSEUDOCODE**

**Inputs:** observed ratings $R$, users $U$, items $V$ and trust information $T$

**Output:** the latent feature vectors

1: U and V initialization – samples from normal noises with zero mean

2: while not converged do

3: update $U$: \[
\frac{\partial \mathcal{L}}{\partial U_u} = \sum_{i=1}^{M} I_{u,i}^R V_i g'(U^T u_i V_i) (g(U^T u_i V_i) - R_{u,i}) + \lambda_U U_u + \lambda_T (U_u - \sum_{v \in N_u} T_{u,v} U_v) - \lambda_T \sum_{\{v | v \in N_u\}} T_{v,u} (U_v - \sum_{w \in N_v} T_{v,w} U_w)
\]

4: update $V$: \[
\frac{\partial \mathcal{L}}{\partial V_i} = \sum_{u=1}^{N} I_{u,i}^R U_u g'(U^T u_i V_i) (g(U^T u_i V_i) - R_{u,i}) + \lambda_V V_i
\]

5: Evaluate LossFunction \[
\mathcal{L}(R,T,U,V) = \frac{1}{2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{u,i}^R ((R_{u,i} - g(U^T u_i V_i))' + \frac{\lambda_U}{2} \sum_{u=1}^{N} U_u^T U_u + \frac{\lambda_V}{2} \sum_{i=1}^{M} V_i^T V_i + \lambda_T \sum_{\{v | v \in N_u\}} T_{v,u}^T (U_v - \sum_{w \in N_v} T_{v,w} U_w)
\]
datasets

which data are we dealing with
# Datasets

<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>

Cold-start users represent more than 50% of each dataset. For both datasets, all ratings have been normalized to a scale from zero to one.
experiments
RUNNING IT
EXPERIMENTS

COMPARING WITH:
PLAIN MATRIX FACTORIZATION
COLLABORATIVE FILTERING
STE $\alpha = 0.4$

ERROR MEASURE: RMSE.
5-FOLD CROSS VALIDATION; 80% TRAIN, 20% TEST
$\lambda_U = \lambda_V = 0.1$
EXPERIMENTS

TAKING INTO ACCOUNT DIFFERENT SIZES OF K - 5 AND 10.

**EPINIONS:**

<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>

**FLIXSTER:**

<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>

**LAMBDA T = 5**

**LAMBDA T = 1**
EXPERIMENTS

OVERALL OBSERVATIONS:

EPINIONS:

FOR $K = 5$:
* 6.2% OF GAIN OVER STE

FOR $K = 10$:
* 5.7% OF GAIN OVER STE

FLIXSTER:

FOR $K = 5$ AND FOR $K = 10$:
* 5.0% OF GAIN OVER STE
EXPERIMENTS

THE DIFFERENT VALUES FOR LAMBDA T:

EPINIONS:
EXPERIMENTS

THE DIFFERENT VALUES FOR LAMBDA T:

FLIXSTER:
COLD-START USERS?
COLD-START USERS?

those with less than 5 ratings!
EXPERIMENTS

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>

**K = 5 FOR BOTH DATASETS**
EXPERIMENTS

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>

11.5% GAIN OVER STE

K = 5 FOR BOTH DATASETS
# Experiments

## 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>

8.5% gain over STE

K = 5 for both datasets
conclusion + future

WHAT COMES NEXT
RELEVANT POINTS

OUTPERFORMS ALL OTHER METHODS COMPARED. EVEN FOR COLD-START USERS!
RELEVANT POINTS

OUTPERFORMS ALL OTHER METHODS COMPARED. EVEN FOR COLD-START USERS!

WHAT ABOUT NEGATIVE TRUST?
HOW COULD SOCIAL MF DEAL WITH IT?
references
GOOD SOURCES
Propagation of Trust and Distrust.
Guha, R., Kumar R., Prabhakar, R., Tomkins, A.
https://pdfs.semanticscholar.org/3911/6b28f1a94e7d0aec082fb325ffdeae430012.pdf

Weidele, D.
https://kops.uni-konstanz.de/bitstream/handle/123456789/29251/Weidele_0-259317.pdf

A Generative Bayesian Model for Item and User Recommendation in Social Rating Networks with Trust Relationships
C. Gianni, Manco G., Ortale R.
http://www.academia.edu/23622275/A_Generative_Bayesian_Model_for_Item_and_User_Recommendation_in_Social_Rating_Networks_with_Trust_Relationships
Recommended System with Social Regularization

Hao Ma, Dengyoung Zhou, Chao Liu, Micheal R.Lyu, Irwin King

Microsoft Research & Chinese University of Hong Kong 2011
Outline

• Motivation and Introduction
  • Trust and Social Aware System Difference
• Traditional Systems
• Problem Definition
• Matrix Factorization
• Social Regularization
• Data Sets
• Comparisons and Results
• Pseudo Code
• Conclusion and Future work
• References
Introduction and Motivation

- Widely studied for information retrieval
- For production Recommendation, used in Amazon, Itunes, Netflix etc
- We always ask friends for recommendation in different products
- We Used Trust aware Systems
- Previous methods ignores social relationship in process,
Trust Aware And Social Friends (1)

• Different Approaches

• “Trust aware” doesn’t have to know each other, ... SoundCloud, twitter etc
  • Based on the Assumption that user have similar taste

• “Social aware” to interact and connect with their friends in the real life, ... facebook etc
  • Need to incorporate social information
Traditional Systems

Collaborative Filtering

- Neighborhood Approaches (User or Items)
- Model Based approaches
Problem Definition

Predict the missing terms of user-item matrix by incorporate the social network information

- Bidirectional social connection (User – Item Matrix)
- Unidirectional trust connection

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>1</td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$u_2$</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$u_3$</td>
<td></td>
<td></td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>$u_4$</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>$u_5$</td>
<td>2</td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Low Rank Matrix Factorization

- We have User and Item Matrix, approx rating matrix by multiplying $l$-rank factors

$$R \approx U^T V \quad \text{Extremely Sparse} \quad \text{.................................................. (1)}$$

- Traditionally, we use Single Value Decomposition (SVD) for minimization of $R$

$$\frac{1}{2} \| R - U^T V \|_F^2 \quad \text{.............................................................. (2)}$$

- Due to sparsity we only need factorize the observed rating in matrix

- So, we use Indicator function for missing value's ----> $I = \{1,0\}$
  - when user rated the item = 1, else = 0

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} (R_{i,j} - U_i^T V_j)^2 \quad \text{.................................................. (3)}$$
Now to avoid overfitting, we add normalization

\[
\min_{U,V} 1/2 \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} (R_{i,j} - U_i^T V_j)^2 - \lambda_1/2 \| U \|^2_F + \lambda_2/2 \| V \|^2_F
\]

\[
\lambda_1, \lambda_2 > 0
\]

- Now we can use Gradient Approach to Find the minimum
Social Regularization

Two models are used for social Regularization

• Average Based Model
• Individual Based Model
Average Based Model

We always ask our friend for recommendation using (....4) Matrix Factorization

$$\min_{U,V} L_1(R, U, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} (R_{i,j} - U_i^T V_j)^2$$

$$+ \frac{\alpha}{2} \sum_{i=1}^{m} ||U_i - \frac{1}{|F^+(i)|} \sum_{f \in F^+(i)} U_f||_F^2$$

$$+ \frac{\lambda_1}{2} ||U||_F^2 + \frac{\lambda_2}{2} ||V||_F^2$$

$\alpha > 0, \lambda_1, \lambda_2 > 0, F^+(i), \ldots, (i)$

$|F^+(i)| = |F^-(i)|, \ldots, (ii)$

In social Network, Facebook etc
• In (……5) we have given the average taste users friends, which doesn’t seems right, due to diverse taste nature …. changing it by introducing a similarity function

\[
\min_{U,V} L_1(R, U, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (R_{ij} - U_i^T V_j)^2 \\
+ \frac{\alpha}{2} \sum_{i=1}^{m} \|U_i - \frac{\sum_{f \in \mathcal{F}^+(i)} \text{Sim}(i, f) \times U_f}{\sum_{f \in \mathcal{F}^+(i)} \text{Sim}(i, f)} \|_F^2 \\
+ \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2
\]

.................(6)

• As similarity is more accurate than our previous approach,
• Now to find the local minima, we just take the derivative
\[
\frac{\partial L_1}{\partial U_i} = \sum_{j=1}^{n} I_{ij} (U_i^T V_j - R_{ij}) V_j + \lambda_1 U_i \\
+ \alpha (U_i - \frac{\sum_{f \in F^+(i)} \text{Sim}(i, f) \times U_f}{\sum_{f \in F^+(i)} \text{Sim}(i, f)}) \\
+ \alpha \sum_{g \in F^-(i)} \frac{-\text{Sim}(i, g)(U_g - \frac{\sum_{f \in F^+(g)} \text{Sim}(g, f) \times U_f}{\sum_{f \in F^+(g)} \text{Sim}(g, f)})}{\sum_{f \in F^+(g)} \text{Sim}(g, f)}
\]

\[
\frac{\partial L_1}{\partial V_j} = \sum_{i=1}^{m} I_{ij} (U_i^T V_j - R_{ij}) U_i + \lambda_2 V_j.
\]

\(R_{ij} = \text{UserItem matrix}\)
\(I_{ij} = \text{Indicator Function}\)
\(F^+ = \text{Out link friends}\)
\(F^- = \text{In link friends}\)
\(U_i = \text{first person}\)
\(U_f = \text{first person friend}\)
\(U_g = \text{second person friend}\)
Individual-based Regularization

• Previously, we used similarity average of friends
• In reality users have diverse taste, so this could cause information loss so, add another regularization term,
• Constraint between user and their friends, individually

\[
\frac{\beta}{2} \sum_{i=1}^{m} \sum_{f \in \mathcal{F}+(i)} \text{Sim}(i, f) \| U_i - U_f \|_F^2, \quad \text{......... (iii)}
\]

Now putting in equation ( ........ 5)
\[
\min_{U, V} \mathcal{L}_2(R, U, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} (R_{i,j} - U_i^T V_j)^2 \\
+ \frac{\beta}{2} \sum_{i=1}^{m} \sum_{f \in \mathcal{F}^+ (i)} \text{Sim}(i, f) \| U_i - U_f \|^2_F \\
+ \lambda_1 \| U \|^2_F + \lambda_2 \| V \|^2_F.
\] ........................ (6)

- Also deal with 2nd degree friends
- Like U(i) and U(g) are not friends but indirectly minimizing the distance between the feature vectors ..... ( expanding.....(iii) )

\[
\text{Sim}(i, f) \| U_i - U_f \|^2_F \text{ and } \text{Sim}(f, g) \| U_f - U_g \|^2_F.
\]

- Now for local minima we again use the gradient descent ( ........ 6 )
\[
\frac{\partial L_2}{\partial U_i} = \sum_{j=1}^{n_i} I_{ij} (U_i^T V_j - R_{ij}) V_j + \lambda_1 U_i \\
+ \beta \sum_{f \in F^+(i)} \text{Sim}(i, f)(U_i - U_f) \\
+ \beta \sum_{g \in F^-(i)} \text{Sim}(i, g)(U_i - U_g),
\]

\[
\frac{\partial L_2}{\partial V_j} = \sum_{i=1}^{m_j} I_{ij} (U_i^T V_j - R_{ij}) U_i + \lambda_2 V_j
\]

\( R_{ij} = \text{UserItem matrix} \)

\( I_{ij} = \text{Indicator Function} \)

\( F^+ = \text{Out link friends} \)

\( F^- = \text{In link friends} \)

\( U_i = \text{first person} \)

\( U_f = U_g = \text{first person friend} \)
Similarity Function

We have User's rating, for similarity two methods are used.

- Two popular methods raging [0,1] Vector Space Similarity (VSS), ignore the individual rating behavior

\[
Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{ij} \cdot R_{fj}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{ij}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{fj}^2}}
\]

- Pearson Correlation Coefficient (PCC) [-1,1], considers individual Rating behavior

\[
Sim(i, f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R_i}) \cdot (R_{fj} - \overline{R_f})}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{ij} - \overline{R_i})^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{fj} - \overline{R_f})^2}}
\]

To map in [0,1] we will do \( f(x) = (x + 1)/2 \)
Datasets

two data-sets

- Douban
  - Rating and Recommendation about movies, books, music
  - Provides information about social friends
  - In Movie Group, Users = 129,490, Movies = 58,541, total rated cells in matrix = 16,830,839

<table>
<thead>
<tr>
<th>Table 1: Statistics of User-Item Matrix of Douban</th>
<th>Table 2: Statistics of Friend Network of Douban</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>User</td>
</tr>
<tr>
<td>Min. Num. of Ratings</td>
<td>1</td>
</tr>
<tr>
<td>Max. Num. of Ratings</td>
<td>6,328</td>
</tr>
<tr>
<td>Avg. Num. of Ratings</td>
<td>129.98</td>
</tr>
</tbody>
</table>
Epinions

- Visitors read review of other users for item selection
- Each user maintains a Trust list
- Users = 51,670; items = 83,509; total rating cells in matrix = 631,064

<table>
<thead>
<tr>
<th>Statistics</th>
<th>User</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Num. of Ratings</td>
<td>1960</td>
<td>7082</td>
</tr>
<tr>
<td>Avg. Num. of Ratings</td>
<td>12.21</td>
<td>7.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Trust per User</th>
<th>Be Trusted per User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Num.</td>
<td>1763</td>
<td>2443</td>
</tr>
<tr>
<td>Avg. Num.</td>
<td>9.91</td>
<td>9.91</td>
</tr>
</tbody>
</table>
Comparison’s

Comparison with previous three other different methods

• NMF
  • For image analysis, also used in Collaborative Filtering
• Probabilistic Matrix Factorization (PMF)
  • User-item matrix for recommendation
• RECOMMENDATION WITH SOCIAL TRUST ENSEMBLE (RSTE) *
  • Trust aware recommendation user’s rating

Parameters

In Douban and Epinions, \( \lambda = 0.001 \)

\[ \begin{align*}
\text{Alpha} &= 0.001 \quad \text{on Douban} \\
\text{Beta} &= 0.01 \quad \text{on Epinions}
\end{align*} \]
Result By Doubian

- Results given by Different Previous Methods and Our Present Method SR_1 and SR_2,

<table>
<thead>
<tr>
<th>Training</th>
<th>Metrics</th>
<th>UserMean</th>
<th>ItemMean</th>
<th>NMF</th>
<th>PMF</th>
<th>RSTE</th>
<th>SR1_vss</th>
<th>SR1_pcc</th>
<th>SR2_vss</th>
<th>SR2_pcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>80%</td>
<td>MAE</td>
<td>0.6809</td>
<td>0.6288</td>
<td>0.5732</td>
<td>0.5693</td>
<td>0.5643</td>
<td>0.5579</td>
<td>0.5576</td>
<td>0.5548</td>
<td>0.5543</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>18.59%</td>
<td>11.85%</td>
<td>3.30%</td>
<td>2.63%</td>
<td>1.77%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.8480</td>
<td>0.7898</td>
<td>0.7225</td>
<td>0.7200</td>
<td>0.7144</td>
<td>0.7026</td>
<td>0.7022</td>
<td>0.6992</td>
<td>0.6988</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>17.59%</td>
<td>11.52%</td>
<td>3.28%</td>
<td>2.94%</td>
<td>2.18%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>MAE</td>
<td>0.6823</td>
<td>0.6300</td>
<td>0.5768</td>
<td>0.5737</td>
<td>0.5698</td>
<td>0.5627</td>
<td>0.5623</td>
<td>0.5597</td>
<td>0.5593</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>18.02%</td>
<td>11.22%</td>
<td>3.03%</td>
<td>2.51%</td>
<td>1.84%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.8505</td>
<td>0.7926</td>
<td>0.7351</td>
<td>0.7290</td>
<td>0.7207</td>
<td>0.7081</td>
<td>0.7078</td>
<td>0.7046</td>
<td>0.7042</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>17.20%</td>
<td>11.15%</td>
<td>4.20%</td>
<td>3.40%</td>
<td>2.29%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40%</td>
<td>MAE</td>
<td>0.6854</td>
<td>0.6317</td>
<td>0.5899</td>
<td>0.5868</td>
<td>0.5767</td>
<td>0.5706</td>
<td>0.5702</td>
<td>0.5690</td>
<td>0.5685</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>17.06%</td>
<td>10.00%</td>
<td>3.63%</td>
<td>3.12%</td>
<td>1.42%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>0.8567</td>
<td>0.7971</td>
<td>0.7482</td>
<td>0.7411</td>
<td>0.7295</td>
<td>0.7172</td>
<td>0.7169</td>
<td>0.7129</td>
<td>0.7125</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>16.83%</td>
<td>10.61%</td>
<td>4.77%</td>
<td>3.86%</td>
<td>2.33%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- SR_1 = Average Based Model
- SR_2 = Individual Based Model
<table>
<thead>
<tr>
<th>Training</th>
<th>Metrics</th>
<th>UserMean</th>
<th>ItemMean</th>
<th>NMF</th>
<th>PMF</th>
<th>RSTE</th>
<th>SR1vss</th>
<th>SR1pcc</th>
<th>SR2vss</th>
<th>SR2pcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>MAE</td>
<td>0.9134</td>
<td>0.9768</td>
<td>0.8712</td>
<td>0.8651</td>
<td>0.8367</td>
<td>0.8290</td>
<td>0.8287</td>
<td>0.8258</td>
<td>0.8256</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>9.61%</td>
<td>15.48%</td>
<td>5.23%</td>
<td>4.57%</td>
<td>1.33%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>1.1688</td>
<td>1.2375</td>
<td>1.1621</td>
<td>1.1544</td>
<td>1.1094</td>
<td>1.0792</td>
<td>1.0790</td>
<td>1.0744</td>
<td>1.0739</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>8.12%</td>
<td>13.22%</td>
<td>7.59%</td>
<td>6.97%</td>
<td>3.20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td>MAE</td>
<td>0.9285</td>
<td>0.9913</td>
<td>0.8951</td>
<td>0.8886</td>
<td>0.8537</td>
<td>0.8493</td>
<td>0.8491</td>
<td>0.8447</td>
<td>0.8443</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>9.07%</td>
<td>14.83%</td>
<td>5.68%</td>
<td>4.99%</td>
<td>1.10%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>1.1817</td>
<td>1.2584</td>
<td>1.1832</td>
<td>1.1760</td>
<td>1.1256</td>
<td>1.1016</td>
<td>1.1013</td>
<td>1.0958</td>
<td>1.0954</td>
</tr>
<tr>
<td></td>
<td>Improve</td>
<td>7.30%</td>
<td>12.95%</td>
<td>7.42%</td>
<td>6.85%</td>
<td>2.68%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Impact Of Parameters

We keep the values of beta low

- Only uses second model
- Douban
Epinions

![Graphs showing MAE and RMSE for different values of \( \beta \)]

(c) Epinions (MAE)

(d) Epinions (RMSE)
Impact Of Similarity Functions

- Also test the similarity function by few alterations (random & set all to 1)
- As we used PCC and VSS for evaluation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training</th>
<th>Metrics</th>
<th>SR2 Sim=1</th>
<th>SR2 Sim=Ran</th>
<th>SR2_vss</th>
<th>SR2_pcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Douban</td>
<td>80%</td>
<td>MAE</td>
<td>0.5579</td>
<td>0.5592</td>
<td>0.5548</td>
<td>0.5543</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>0.7034</td>
<td>0.7047</td>
<td>0.6992</td>
<td>0.6988</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>MAE</td>
<td>0.5631</td>
<td>0.5643</td>
<td>0.5597</td>
<td>0.5593</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>0.7083</td>
<td>0.7098</td>
<td>0.7046</td>
<td>0.7042</td>
</tr>
<tr>
<td></td>
<td>40%</td>
<td>MAE</td>
<td>0.5724</td>
<td>0.5737</td>
<td>0.5690</td>
<td>0.5685</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>0.7195</td>
<td>0.7209</td>
<td>0.7129</td>
<td>0.7125</td>
</tr>
<tr>
<td>Epinions</td>
<td>90%</td>
<td>MAE</td>
<td>0.8324</td>
<td>0.8345</td>
<td>0.8258</td>
<td>0.8256</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>1.0794</td>
<td>1.0809</td>
<td>1.0744</td>
<td>1.0739</td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>MAE</td>
<td>0.8511</td>
<td>0.8530</td>
<td>0.8447</td>
<td>0.8443</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>1.1002</td>
<td>1.1018</td>
<td>1.0958</td>
<td>1.0954</td>
</tr>
</tbody>
</table>
Pseudo Code: Averaging Method

Input

\[ U = R^{lxm} \approx lxm \]
\[ V = R^{lxn} \approx lxn \]
\[ \lambda_1 = \lambda_2 = 0.001 \]
\[ \alpha = 0.001 \]

Algorithm

\begin{align*}
\text{for } i : m \\
\quad \text{for } j : n \\
\quad \quad x &= I_{i,j} \left( U_i^T V_j - R_{i,j} \right) + \lambda U_i \\
\quad \text{for } f : f^+ \\
\quad \quad b &= b + \left( U_i - \frac{\text{sim}(i,f)^+ U_f}{\text{sim}(i,f)} \right) \\
\quad \text{for } g : f^- \\
\quad \quad c &= c + \left( -\text{sim}(i,g) \right) \frac{(U_g - \text{sim}(g,f) \times U_f)}{\text{sim}(g,f)} \\
\quad V_j &= I_{i,j} \left( U_i^T - R_{i,j} \right) \ast U_i + \lambda_2 V_j \\
\quad U_i &= x + \alpha \ast b + \alpha \ast c \\
\text{return } U, V
\end{align*}
Pseudo Code: Individual Method

Input

\[ U = R^{lxm} \approx lxm \]
\[ V = R^{lxn} \approx lxn \]
\[ \lambda_1 = \lambda_2 = 0.001 \]
\[ \beta = 0.001 \]

Algorithm

for \( i : m \)
  for \( j : n \)
    \[ x = I_{i,j} ( U_i^T V_j - R_{i,j} ) + \lambda U_i \]
    for \( f : f^+ \)
      \[ b = b + \text{sim}(i,f) \ast (U_i - U_f) \]
    for \( g : f^- \)
      \[ c = c + \text{sim}(i,g)(U_i - U_g) \]
    \[ V_j = I_{i,j} ( U_i^T - R_{i,j} ) \ast U_i + \lambda_2 V_j \]
    \[ U_i = x + \beta \ast b + \beta \ast c \]

return \( U,V \)
Conclusion and Future Work

- Two general algorithms are proposed that imposed social regularization using PCC and VSS
- Quite generic method also can be applied to trust aware recommendation problems
- Comparison shows it outperforms the state of the art RSTE method

- Make it more better if we have user’s information about Clicking behavior and Tagging Records
- To make it more realistic we can use categorical cluster wise approach
References


Conclusions and Comparison

<table>
<thead>
<tr>
<th></th>
<th>SoRec</th>
<th>SocialMF</th>
<th>SRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Based</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Method</td>
<td>Co-factorization</td>
<td>Regularization methods</td>
<td>Regularization methods</td>
</tr>
<tr>
<td>$Social(T, S, \Omega)$</td>
<td>$\min \sum_{i=1}^{n} \sum_{u_k \in N_i} (S_{ik} - u_i^T z_k)^2$</td>
<td>$\min \sum_{i=1}^{n} (u_i - \sum_{u_k \in N_i} S_{ik} u_k)^2$</td>
<td>$\min \sum_{i=1}^{n} \sum_{u_k \in N_i} S_{ik} (u_i - u_k)^2$</td>
</tr>
<tr>
<td>Dataset - Epinions03</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Dataset - Epinions02</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Dataset - Douban</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset - Flixster</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Error Metric - RMSE</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Error Metric - MAE</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>
## Conclusions and Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Best MAE</th>
<th>Best RMSE</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoRec – Epinions</td>
<td>0.8932</td>
<td>---</td>
<td>Dimensionality = 10 99% Training Data</td>
</tr>
<tr>
<td>SocialMF – Epinions [2]</td>
<td>---</td>
<td>1.075</td>
<td>80% Training Data</td>
</tr>
<tr>
<td>SocialMF - Flixter</td>
<td>0.815</td>
<td>5-fold CV.</td>
<td></td>
</tr>
<tr>
<td>SRS – Epinions</td>
<td>0.8256</td>
<td>1.0739</td>
<td>PCC, Individual Method 90% Training Data</td>
</tr>
<tr>
<td>SRS - Douban</td>
<td>0.5543</td>
<td>0.6988</td>
<td>PCC, Individual Method 80% Training Data</td>
</tr>
</tbody>
</table>
Conclusions and Comparison

Issues on Social Recommendation

Social recommendation may also perform worse than traditional recommender systems:

• social network composed of valuable friends, casual friends and event friends; users are not necessarily all that similar;

• social relations mixed with useful and noise connections;

• users with fewer ratings are likely to also have fewer connections.
QUESTIONS
Backup Slides - SRS
NMF

• Originally Used for image Analysis, But now widely used in Collaborative Filtering (For recommendation uses User Item matrix )

• algorithm for non-negative matrix factorization that is able to learn parts of faces and semantic features of text.

• *This is in contrast to other methods, such as principal components analysis and vector quantization,*

• that learn holistic, not parts-based, representations. Non-negative matrix factorization is distinguished from the other methods by its use of non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations.

• When non-negative matrix factorization is implemented as a neural network, parts-based representations emerge by virtue of two properties: the firing rates of neurons are never negative and synaptic strengths do not change sign.
PMF (Probabilistic Matrix Factorization)

- model which scales linearly with the number of observations and, more importantly, performs well on the large, sparse, and very imbalanced Netflix dataset

- users who have rated similar sets of movies are likely to have similar preferences

- When the predictions of multiple PMF models are linearly combined with the predictions of Restricted Boltzmann Machines models, we achieve an error rate of 0.8861, that is nearly 7% better than the score of Netflix’s own system.

$$p(R|U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[ \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2) \right]^{I_{ij}}$$
RSTE (RECOMMENDATION WITH SOCIAL TRUST ENSEMBLE)

- Aiming at modeling recommender systems more accurately and realistically, we propose a novel probabilistic factor analysis framework, which naturally fuses the users’ tastes and their trusted friends’ favors together.

- term Social Trust Ensemble (RSTE) to represent the formulation of the social trust restrictions on the recommender systems.

\[
p(R|U, V, \sigma_R^2) = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[ \mathcal{N} \left( R_{ij} | g(U_i^T V_j), \sigma_R^2 \right) \right]^{I_{ij}}^R.
\]

- Uses the epinion Dataset