Design and Analysis of a Gossip-based Trust Recommender System

Stefan Magureanu, Nima Dokoohaki,
Shahab Mokarizadeh and Mihhail Matskin

{magur,nimad,shahab,misha}@kth.se

KTH Royal Institute of Technology, Sweden
Introduction

- Prominently used in online stores and social networks
- Emerging interest from Big Data Analysis
- Two main approaches:
  - Neighborhood-based Collaborative Filtering (CF)
  - Matrix Factorization-based (MF)
- Important setback: *Scalability*
Background - Trust

- **Trust propagation:**
  - Rely on a pre-defined network of \((social)\) trust
  - Compute trust between nodes not directly connected using propagation over existing edges

- **Trust inference from ratings:**
  - Computed solely from user ratings
  - Usually associated with similarity - not quite
Background - Gossip Algorithms

- **T-Man:**
  - Network overlay construction algorithm
  - Backbone of our system
  - Uses Cyclon to increase convergence speed
  - Versatile

- **Basic Mechanism:**
  - Nodes maintain neighborhoods
  - Iteratively exchange neighborhood information with a chosen neighbor
  - Keep most *useful* nodes in neighborhood as described by a *utility function* (or *distance function*)
Approach - Steps

- Reorganize network, clustering similar users together
- Compute *trust* values towards neighbors
- Improve coverage if needed - recurrent predictions
Approach -
T-Man Distance Metric

- **Goal:**
  - Gather similar users in the neighborhood
  - Use the T-Man neighborhood as the "filter" in CF

- **Intuitive metric: Pearson Correlation**

\[
Similarity(u_1, u_2) = \frac{\sum_i r_{u_1,i} \times r_{u_2,i}}{\sqrt{\sum_i r_{u_1,i}^2 \times \sqrt{\sum_i r_{u_2,i}^2}}}
\]

- **Drawback:**
  - Does not take into account number of items in common
  - Users with more rated items in common are more likely to be interested in same items and have rated items the active user is interested in
Approach - T-Man Distance Metric

- Our variation of the Pearson Correlation:

\[
\text{Similarity}(u_1, u_2) = \frac{\sum_i r_{u_1,i} \times r_{u_2,i}}{\sqrt{\sum_i C \times r_{u_1,i}^2} \times \sqrt{\sum_i r_{u_2,i}^2}}
\]

where \( C = 1 \) if \( u_2 \) rated \( i \) else \( C = 0.5 \).

- Now:
  - Metric penalizes users proportionally to the number of items rated by the active user that they have not rated
  - Compromise between similarity of ratings and similarity of items rated
Approach - Dealing with sparsity
Approach - Modeling trust

Basic idea:

- Use the neighbors to make predictions on the active user’s rated item
- Obtain a system of equations with the trust values being the variables
- Solve the system to discover the trusts towards each neighbor
Approach - Modeling trust

- For the prediction formula:
  \[ \frac{\sum_n (r_{n,i} - \bar{r}_n) \times w_n}{\sum_n w_n} = r_i - \bar{r} \]

- The resulting system would be:
  \[
  \begin{cases}
  \sum_n (r_{n,0} - \bar{r}_n - r_0 + \bar{r}) \times w_n = 0 \\
  \ldots \\
  \sum_n (r_{n,i} - \bar{r}_n - r_i + \bar{r}) \times w_n = 0
  \end{cases}
  \]

- Problem:
  - Solutions must be positive
  - We require positive coefficients to approximate positive solutions
Approach - Modeling trust

- Convert the system to a more useful form by adding a new equation:

\[
\begin{align*}
\sum_n \left( r_{n,0} - \bar{r}_n - r_0 + \bar{r} \right) \times w_n &= 0 \\
\ldots \\
\sum_n \left( r_{n,i} - \bar{r}_n - r_i + \bar{r} \right) \times w_n &= 0 \\
\sum_n w_n &= N \times Trust_{\text{mean}}
\end{align*}
\]

where \( N \) is the number of variables and \( Trust_{\text{mean}} \) is the average desired trust value.
Approach - Modeling trust

- Obtain positive coefficients:
  \[
  \sum_n \left( 2 \times R_{\text{max}} + r_{n,0} - \bar{r}_n - r_0 + \bar{r} \right) \times w_n = C
  \]
  \[
  \ldots
  \]
  \[
  \sum_n \left( 2 \times R_{\text{max}} + r_{n,i} - \bar{r}_n - r_i + \bar{r} \right) \times w_n = C
  \]
  \[
  \sum_n w_n = N \times \text{Trust}_{\text{mean}}
  \]
  where \( C = 2 \times R_{\text{max}} \times \sum_n w_n \) and \( N \) is the number of variables.

- We can now chose a convex method to approximate the solutions.
Approach - Approximating trust values

- Our solving algorithm is proposed by D. Cartwright in 2011
- Uses Expectation Maximization to converge to approximations of positive solutions
- Can be reduced to the following form:

```latex
while(!done)
    for each n in neighbors
        \[ w_n = w_n \times \frac{\sum_{i\in Items} \frac{C_i}{C} \times coef_{i,n}}{\sum_{i\in Items} coef_{i,n}} \]
```
Approach - Approximating trust values

Obtained Trust Distribution:

![Bar chart showing trust distribution for Yahoo with various trust values ranging from 0.0 to 0.9 on the x-axis and number of occurrences on the y-axis.]}
Results - Setup

- Leave one out
  - Hide random item from each user
  - Run T-Man for X rounds
  - Predict hidden item ratings
  - Coverage = % of hidden items for which the system could generate a prediction

- 3 Datasets
  - Movielens - 6,040 users, 1,000,000 ratings, 4,000 items
  - Yahoo Webscope! - 15,400 users, 300,000 ratings, 1,000 items
  - Epinions - 49,290 users, 664,824 ratings, 140,000 items
Results - MAE

Movielens (top) and Yahoo (bottom) datasets.
Results - MAE

Epinions dataset

![Graph showing MAE for different methods across varying number of neighbors.](image)
Results - Coverage

Movielens dataset

Coverage (% of total prediction attempts)

T-Man rounds
Results - Coverage Search Range

Epinions dataset - effect of search range on coverage.
Results - Coverage
Search Range

Epinions dataset - effect of search range on coverage.
Results - Coverage Effects on MAE

Movielens dataset - effect of distance metric on MAE.
Conclusions

- Proposed a fully decentralized CF recommender system
- Presented methods of increasing coverage
- Proposed new trust inference model
- Future work:
  - Evaluate diversity of recommendation
  - Evaluate trust models based on boosting
  - Implement item-relevance
  - Model trust distribution in the system
Questions?

Contact information:
Stefan Magureanu, magur@kth.se