Skewness-aware Clustering Social Recommender

Chaofei Fan STFAN@STANFORD.EDU
Le Yu BILLYUE@STANFORD.EDU
Huanglong Ren RHLCHIEF@STANFORD.EDU
Computer Science Deptment, Stanford University

Abstract

Recommender systems have been a hot research area recently. One of the most widely used methods is Collaborative Filtering(CF), which selects items for an individual user from other similar users. However, CF may not fully reflect the procedure of how people choose an item in real life, for users are more likely to ask friends for opinions instead of asking strangers with similar interests. Recently, some recommendation methods based on social network have been proposed. These approaches assume that data follow Gaussian distribution, and then incorporate social network into the CF algorithms. Therefore, users preferences can be influenced by the flavors of their friends. In this paper, we propose the Skewness-aware Clustering Social Recommender. Our contributions are three-fold: (1) we show that data do not follow Gaussian distribution and then correct the bias and skewness of ratings; (2) we propose the social cluster model based on cluster coefficient; (3) our improvement outperforms the baseline by 3.34%.

1. Introduction

We are coming into recommendation age: providing personalized recommendations has been an important problem for more than ten years, e.g. Google News, Netflix DVDs and LastFM music. In the meantime, researches on recommender systems also attract a lot of attention in the last decade.

One of the most successful technologies for recommender systems is Collaborative Filtering (CF)(Salakhutdinov & Mnih, 2007), which predicts items for certain users based on the items preferred by similar users. However, traditional CF techniques do not consider social relationships between users. Therefore the recommender systems may not reflect the real life recommendation, i.e. when we are planning for movies, we might turn to our friends for recommendation instead of strangers with similar tastes. Another problem of traditional recommender system is cold-start users. In this paper, cold-start users are referred to those users who have no ratings or few ratings. Therefore, the system can hardly recommend items for these cold-start users, as the system records little rating from the users.

Recently, the advent of social network information provides new topics for recommender systems. Some recommendation algorithms based on probabilistic models (Ma et al., 2008; Jamali & Ester, 2011) with social relationship have emerged. These new approaches utilize the social network among users and make recommendations for a certain user with the aid of the ratings of the user's friends. The experiments show these methods outperform the recommender system without social information.

Another advantage of social network information is that it helps to solve the cold-start problem. Although cold start users have no or few ratings, some of them participate in social network. In that case, we can recommend for these social-active cold-start users. For example, since new users on Amazon.com may have no purchase record before, so they have no personal preference. But they might buy items purchased by their friends.

One problem in social probabilistic models (Ma et al., 2008; Jamali & Ester, 2011) is that they make the assumption that rating data follows Gaussian distribution. However, as in most real world data, this assumption is not satisfied. For example, because people are not willing to rate the movie if they think the movie is bad, the number of '5-star' ratings is much more than that of '1-star' ratings. However, previous social models are based on the assumptions that the

Appearing in *CS229 Machine Learning Project*, Stanford, 2011.

number of ratings follows the Gaussian distribution.

The rest of this paper is organized as follows. Section 2 provides the formal definition of recommender system problem. Section 3 introduces the state-of-the-art of recommender systems. Section 4 details our method for social recommendation. The performance of our approach is presented in Section 5. Finally, Section 6 concludes the paper.

2. Problem Definition

The recommender systems consist of three entities: M users in the systems $U(user) = \{u_1, ..., u_M\}$, N items for recommendations $V(items) = \{v_1, ..., v_N\}$ and the user-item ratings matrix R, where R_{ij} denotes the score u_i votes for v_j. R represents the ratings. Matrix U and V are defined as the user latent matrix and item latent matrix in D dimension. $U = (U_{ij})_{M\times D}$, $V = (V_{ij})_{N\times D}$, $D < min(M, N)$. We use U_i as the user i's feature vector, and V_j as item j's feature vector.

The problem discussed in this paper incorporates social network information between users into the CF methods of the traditional recommender system to improve the recommendations. In social recommender systems, prediction for a user is not only judged by the user himself/herself, but also influenced by his/her friends. The problem we investigate in this paper is how to predict the missing values of the user-item matrix by employing the social relation data. In this paper, S denotes the social network between users. S_{ij} is a binary value where $S_{ij} = 1$ means u_i trusts u_j.

3. Related Work

In this section, we introduce major approaches for recommendation, i.e. traditional recommender systems and social recommender methods.

3.1. Traditional Recommender Systems

Traditional recommender systems refer to the systems only use rating matrix for recommendation. Among all the algorithms, Collaborative Filtering(CF) is the most efficient in dealing with large data sets. Certain user's preference can be predicted by information from other users who have similar interests.

The state-of-the-art CF algorithm (Salakhutdinov & Mnih, 2007) is Probabilistic Matrix Factorization (PMF), which factorizes the user features matrix and item features matrix. Rating R_{ij} by user i for item j is $\hat{R_{ij}} = U_iV_j^T$. The probability of the rating data can be displayed as

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^{M}\prod_{j=1}^{N}[\mathcal{N}(R_{ij}|g(U_iV_j^T), \sigma_R^2)]^{I_{ij}^R}$$

where we define indicator function $I = (I_{ij})_{M\times N}$, if $R_{ij} \neq 0$, $I_{ij} = 1$; else $I_{ij} = 0$.

The model assumes that rating data follow the Gaussian distribution. In order to set the data within the rating range, we use logistic function as $g(x)$. As for the user features and item features, PMF applies the Gaussian priors to those features, we have:

$$p(U|\sigma_U^2) = \prod_{i=1}^{M}\mathcal{N}(U_i|0, \sigma_U^2 I); p(V|\sigma_V^2) = \prod_{j=1}^{N}\mathcal{N}(V_j|0, \sigma_V^2 I)$$

According to bayesian rules, the probability of user features U and item features V can be represented as:

$$p(U, V|R, \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto p(R|U, V, \sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \quad (1)$$

3.2. Social Recommender System

Although CF methods have achieved great success in recommendation applications, there still exist some problems which limit its performance. One of the biggest problems is data sparsity. To solve this problem, (Yu et al., 2011; Ma et al., 2011) utilize social network information to provide more data for recommendation. These methods explore the similar users to generate recommendations. User i will ask friends for ideas. So the predicted rating R_{ij} by user i for item j is $\hat{R_{ij}} = U_iV_j^T + \beta \sum_{k\in F(i)} S_{ik}U_kV_j^T$, where $F(i)$ are the friends that user i trusts, while S_{ik} is the relation between user i and k, and β describes how much that user i trust the friends.

Therefore, social recommendation suggested in (Ma et al., 2008) have the following rating probabilistic model:

$$\begin{aligned} &p(R|S, U, V, \sigma_R^2) &(2)\\ &= \prod_{i=1}^{M}\prod_{j=1}^{N}[\mathcal{N}(R_{ij}|g(U_iV_j^T + \sum_{k\in F(i)} S_{ik}U_kV_j^T, \sigma_R^2))]^{I_{ij}^R} \end{aligned}$$

In order to calculate the user and item features, we want to maximize the likelihood of

$$\begin{aligned} &p(U, V|R, S, \sigma_R^2, \sigma_U^2, \sigma_V^2)\\ &\propto p(R|S, U, V, \sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \quad (3) \end{aligned}$$

4. Skewness-aware Clustering Social Recommender

In this section, we analyze the social recommendation problem based on matrix factorization framework. Then we propose a new social relationship similarity function.

4.1. Skewness Analysis

Previous models have the assumption that the rating data is under Gaussian distribution. However, most real-world problems do not follow the assumption(Cao et al., 2010). In the real life, while users are unwillingly to rate bad movies, they are more likely to give ratings to good movies. Therefore, the rating data are asymmetric, which do not satisfy model assumptions. In Figure 1, it shows the number of different ratings.

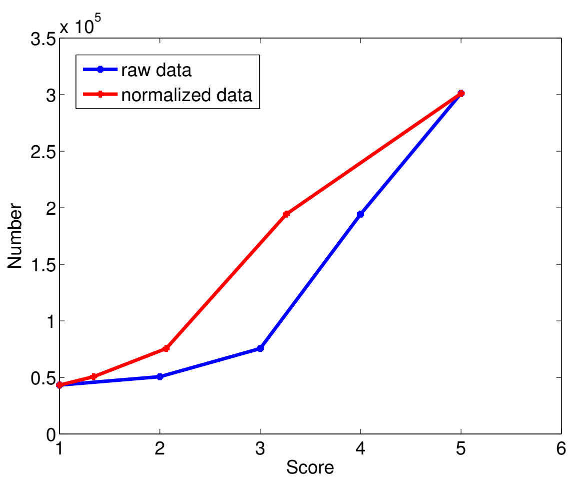

Figure 1. Rating skewness analysis

The skewness γ can be calculated by:

$$\gamma = \frac{E[(Z - E[Z])^3]}{E[(Z - E[Z])^2]^{\frac{3}{2}}} \tag{4}$$

where Z is the random variable and $E[Z]$ is the expectation of Z. The skewness of the Epinion dataset is -1.087. The skewness shows that the data violates the Gaussian assumptions, which is the foundations of the probabilistic model. In order to normalize the rating score we use the normalization function:

$$Normal(x) = cx^{1+\alpha} + b \tag{5}$$

where α, b, c are parameters in the transformation. α is the parameter to ensure skewness γ to be close to 0; c, and b are parameters to ensure the rating range does not change, i.e. mapping the original ratings of scale 1 to 5 to new ratings of the same scale. The corrected rating distribution is shown in Figure 1.

4.2. Social Recommender

From the social influence, user u will be affected by each of his/her friends $F(i)$. The social network will change the user latent features, while it will not influence the equation of the rating distribution or item distribution. Therefore, user features U_i will be affected by the friends features:

$$\hat{U}_i = S_{i.}U \tag{6}$$

where $\hat{U}_i$ is the combination of users i's friends' features, and $S_{i.}$ is the social network of user i.

The user features not only follow the Gaussian distribution, but also are influenced by users that he/she trusts. So we have

$$p(U|S, \sigma_S^2, \sigma_U^2) = \prod_{i=1}^{M} \mathcal{N}(U_i|\hat{U}_i, \sigma_S^2 I) \prod_{i=1}^{M} \mathcal{N}(U_i|0, \sigma_U^2 I) \tag{7}$$

According to bayesian rules, the probability of user features and item features can be represented as:

$$\begin{aligned}
&p(U, V|R, S, \sigma_R^2, \sigma_S^2, \sigma_U^2, \sigma_V^2) \\
&\propto p(R|U, V, \sigma_R^2) p(U|S, \sigma_S^2, \sigma_U^2) p(V|\sigma_V^2) \\
&= \prod_{i=1}^{M} \prod_{j=1}^{N} [\mathcal{N}(R_{ij}|g(U_i V_j^T), \sigma_R^2)]^{I_{ij}^R} \times \prod_{i=1}^{M} \mathcal{N}(U_i|0, \sigma_U^2 I) \\
&\times \prod_{i=1}^{M} \mathcal{N}(U_i|\hat{U}_i, \sigma_S^2 I) \times \prod_{j=1}^{N} \mathcal{N}(V_j|0, \sigma_V^2 I)
\end{aligned} \tag{8}$$

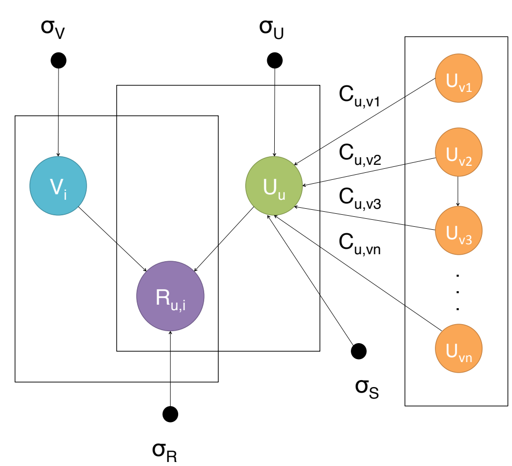

Figure 2. Probabilistic Graphic Model

The graphical model of Social Network is shown in Figure 2. We want to maximize the log-likelihood of the posterior distribution, we get:

$$log\ p(U, V|R, S, \sigma_R^2, \sigma_S^2, \sigma_U^2, \sigma_V^2) =$$

$$\begin{aligned} & - \frac{1}{2\sigma_R^2} \sum_{i,j} I_{ij}(R_{ij} - U_i V_j^T)^2 \\ & - \frac{1}{2\sigma_U^2} ||U||_F^2 - \frac{1}{2\sigma_V^2} ||V||_F^2 \\ & - \frac{1}{2\sigma_S^2} ||U - SU||_F^2 + C \end{aligned} \tag{9}$$

where C is a constant irrelevant to U and V. In order to maximize the log-likelihood, we can then minimize the following objective loss function:

$$\begin{aligned} \mathcal{L}(U,V) \quad = \quad & \sum_{i,j} I_{ij}(R_{ij} - U_i V_j^T)^2 + \\ & \lambda_1 ||U||_F^2 + \lambda_2 ||V||_F^2 + \\ & \beta \sum_i \sum_{f \in F(i)} S_{if} ||U_i - U_f||_F^2. \end{aligned} \tag{10}$$

$\sum_i \sum_{f \in F(i)} S_{if} ||U_i - U_f||_F^2$ is the social regularization term to constrain user's feature vectors. β is the influencing impact of social network. $F(i)$ are users whom user u_i trust. Taste difference between two friends can be described as $S_{if}||U_i - U_f||_F^2$: U_i means the feature vector of user u_i. If interest of user u_i is not similar to his friend u_f, $||U_i - U_f||_F^2$ will be large. In other words, u_i is less likely to follow the taste of u_f.

Then we need to minimize the loss function. By performing gradient descent on U_i and V_j for user i and item j, we obtain

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial U_i} \quad = \quad & \sum_{j=1}^{N} I_{ij}(R_{ij} - U_i V_j^T) + \lambda_1 U_i + \\ & \beta \sum_{f \in F(i)} S_{if}(U_i - U_f), \end{aligned} \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial V_j} \quad = \quad \sum_{i=1}^{M} I_{ij}(R_{ij} - U_i V_j^T) + \lambda_2 V_j. \tag{12}$$

4.3. Clustering Coefficient

(Ma et al., 2011) proposes the idea of calculating the similarities between friends based on the common movies both users watch. However, as (Yu et al., 2011) pointed out, the friendship would be lost when two friends do not watch the same movie together. Therefore, we propose the clustering coefficient between user i and j as:

$$C(u_i, u_j) = \frac{\sum_{u_k \in F(i)} 1\{S_{ik} \cap S_{kj}\}}{|F(i)| - 1}, \tag{13}$$

where $F(i)$ are the friends that user i trusts. $|F(i)|$ is the number of friends that user i trusts. $1\{S_{ik} \cap S_{kj}\} = 1$ means user k is trusted by user i and user j. We then apply Laplace smoothing to replaces the estimate by

$$C(u_i, u_j) = \frac{\sum_{u_k \in F(i)} 1\{S_{ik} \cap S_{kj}\} + 1}{2|F(i)| - 1} \tag{14}$$

5. Experiment

In this section, we report our experiments described in Section 4 and compare the results with traditional recommendation and social recommendation methods.

5.1. Datasets

Our experiment is conducted on a social networking dataset: Epinions (Massa & Avesani, 2007). Epinions is a well-known rating network for research. Users rate the product or service on a rating scale from 1 to 5 stars. Epinions member maintains a trust list, and it shows a network of trust relationships between users. It provides "Twitter-like" social network. The Epinions dataset consists of 49,290 users who have rated a total of 139,738 different items. The total number of ratings is 664,824. As to the user social trust network, the total number of issued trust statements is 511,799.

5.2. Evaluation Measurement

We use 5-fold Cross Validation to estimate the performance of different algorithms. In each fold, the validation datasets are divided into training sets and testing sets randomly. The training set contains 80% known positive examples and the other 20% elements of the matrix are treated as unknown. The known positives in the training set are excluded in the test process.

For all the experiments in this paper, the values of λ_1 and λ_2 are set to 0.1. The experimental results using 5 and 10 as dimensions to represent the latent features are shown in Table 1.

The evaluation metric we use in the experiment is the Root Mean Square Error (RMSE), which is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j} (R_{ij} - \hat{R}_{ij})^2}{T}}, \tag{15}$$

where $\hat{R_{ij}}$ means predicted score for user u_i on item i_j and T means the number of pairs of (i, j) in the test set. Notice smaller RMSE value means a better performance.

5.3. Comparisons

In this section, we compare the recommendation results of the following algorithms:

- PMF: Probabilistic Matrix Factorization (Salakhutdinov & Mnih, 2007). This

Table 1. RMSE for Epinions with different D

Users	Dim.	PMF	STE	CSR	$SCSR$
All	D=5	1.1435	1.1295	1.1250	1.0975
All	D=10	1.1524	1.1379	1.1304	1.0987
Cold	D=5	1.1970	1.1833	1.1765	1.1432
Cold	D=10	1.1947	1.1811	1.1760	1.1408

method only uses rating data to recommend.

- STE: Social Trust Ensemble. This method adds social network to the PMF model (Ma et al., 2008).
- CSR: Clustering Social Recommender, which is proposed in Section 4. It also uses social network, and it uses cluster coefficient function.
- $SCSR$: Skewness-aware Clustering Social Recommender. The model corrects the bias in the data and then applies the CSR method.

We run the recommendation algorithms on Epinions dataset. The result is in Table 1. We see that by incorporating social networks, STE outperforms PMF by 1.2% for $D = 5$ and 1.15% for $D = 10$. This is similar to the results in (Ma et al., 2008), meaning that social network is indeed useful in recommendation. Comparing CSR with STE, the improvement is 0.4% for $D = 5$ and 0.7% for $D = 10$. This shows that it is useful to consider the clustering effect in the social network rather than consider all friends are the same. Our last algorithm $SCSR$ outperforms STE by 2.83% for $D = 5$ and 3.34% for $D = 10$. It also outperforms PMF by 4.02% for $D = 5$ and 4.26% for $D = 10$. This is great improvement since the Netflix Prize awards 1 million dollars for 10% improvement.

We also run the experiment on cold-start users. Cold-start users are those who rate less than 10 movies. It is generally a hard problem to predict ratings for them because the system has little information about them. The results for cold-start users are similar to the results of all users. Thus by incorporating social networks we can improve the performance for both kinds of users.

6. Conclusion

In this paper, we present the skewness-aware social recommender. We first identify that user ratings do not follow Gaussian distribution. We then adjust the skewness of the data and also consider the clustering effect in social network. This results in 4.26% improvement over the state-of-the-art Collaborative Filtering algorithm. Our algorithm performs well for both normal users and cold-start users.

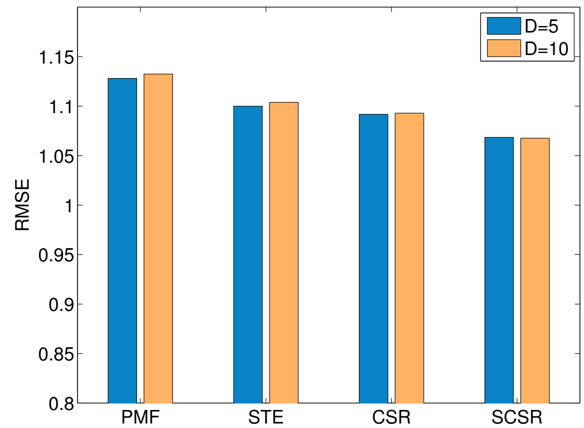

Figure 3. RMSE result of all users.

For the future work, we would like to test our algorithm on a more popular dataset (e.g. Netflix) to have a more solid comparison with other algorithms. Also we would like to consider the problem of cold-start items (i.e. items rated by few users).

References

Cao, Bin, Liu, Nathan Nan, and Yang, Qiang. Transfer learning for collective link prediction in multiple heterogenous domains. In *ICML*, pp. 159–166. Omnipress, 2010.

Jamali, Mohsen and Ester, Martin. A transitivity aware matrix factorization model for recommendation in social networks. In *IJCAI*, pp. 2644–2649. IJCAI/AAAI, 2011. ISBN 978-1-57735-516-8.

Ma, Hao, Yang, Haixuan, Lyu, Michael R., and King, Irwin. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pp. 931–940. ACM, 2008. ISBN 978-1-59593-991-3.

Ma, Hao, Zhou, Dengyong, Liu, Chao, Lyu, Michael R., and King, Irwin. Recommender systems with social regularization. In King, Irwin, Nejdl, Wolfgang, and Li, Hang (eds.), *WSDM*, pp. 287–296. ACM, 2011. ISBN 978-1-4503-0493-1.

Massa, Paolo and Avesani, Paolo. Trust-aware recommender systems. In Konstan, Joseph A., Riedl, John, and Smyth, Barry (eds.), *RecSys*, pp. 17–24. ACM, 2007. ISBN 978-1-59593-730-8.

Salakhutdinov, Ruslan and Mnih, Andriy. Probabilistic matrix factorization. In *NIPS*. MIT Press, 2007.

Yu, Le, Pan, Rong, and Li, Zhangfeng. Adaptive social similarities for recommender systems. In *RecSys*, pp. 257–260. ACM, 2011. ISBN 978-1-4503-0683-6.