SoRank: Incorporating Social Information into Learning to Rank Models for Recommendation

Weilong Yao *, Jing He†, Guangyan Huang†, Yanchun Zhang*†

*University of Chinese Academy of Sciences, Beijing, China
†Centre for Applied Informatics, Victoria University, Melbourne, Australia
yaoweilong12@mails.ucas.ac.cn,
{Jing.He,Guangyan.Huang,Yanchun.Zhang}@vu.edu.au

ABSTRACT

Most existing learning to rank based recommendation methods only use user-item preferences to rank items, while neglecting social relations among users. In this paper, we propose a novel, effective and efficient model, SoRank, by integrating social information among users into listwise ranking model to improve quality of ranked list of items. In addition, with linear complexity to the number of observed ratings, SoRank is able to scale to very large dataset. Experimental results on publicly available dataset demonstrate the effectiveness of SoRank.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; J.4 [**Computer Applications**]: SOCIAL AND BEHAVIORAL SCIENCES

Keywords

Learning to Rank, Matrix Factorization, Social Relation

1. INTRODUCTION

Motivated by the analogy between query-document relations and user-item relations in recommender systems, several learning to rank [3] based recommendation methods have been proposed to improve the performance of traditional collaborative filtering. These models assume that all users are independent and identically distributed, while ignoring the social connections among users.

However, in real-world situation, users can be easily affected by the friends they trust, and prefer their friends' recommendations. The reality in the case of item ranking is that the rank position of an item is determined by the active user's own taste, indirectly from his/her trusted friends' taste as well. Based on the above intuition, we propose a straightforward but effective way to incorporate social information into listwise learning to rank model for recommendation.

Copyright is held by the author/owner(s).
WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577333.

2. SORANK

In this section, we introduce the listwise learning to rank with social information for item recommendation.

2.1 Preliminaries

Top one probability [1] of an item represents the probability of its being ranked on the top of a list. Given user i's ranked list $l_i = (R_{i1}, R_{i2}, ..., R_{iK})$, where R_{ik} is i's rating score for item k, the top one probability of item j is given by:

$$P_{l_i}(R_{ij}) = \frac{exp(R_{ij})}{\sum_{k=1}^{K} exp(R_{ik})}, \tag{1}$$

where $exp(x)$ denotes the exponential function of x.

2.2 Objective function

Based on matrix factorization framework, the predicted ranking score of item j with respect to user i is expressed by:

$$\widehat{R}_{ij} = g(U_i^T V_j), \tag{2}$$

where U_i and $V_j \in \mathbb{R}^D$ denote the latent factors of user i and item j, respectively, where D is the dimensionality of latent vectors. $g(x)$ is logistic function. To produce a ranked list of items for a user i items are scored using Eq. 2 and ranked according to the scores. Furthermore, the following Eq. 3 will be employed to derive the final ranking score by integrating users with their trusted friends' tastes via an ensemble parameter α:

$$\widehat{R}_{ij} = g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in T(i)} \tilde{S}_{ik} U_k^T V_j), \tag{3}$$

where $\tilde{S}_{ik}$ denotes the normalized trust strength between user i and k, and $T(i)$ represents the set of users trusted by user i. The second term of Eq. 3 is interpreted as following: if the user k ranks item j highly (i.e., $U_k^T V_j$ is large), and user i expressed trust on k, then the ranking score between i and j tend to be increased. The ranking score of item j is a balance between the active user's and his/her trusted friends' favors, smoothed by the parameter $\alpha \in [0, 1]$, which controls how much the user is affected by his/her friends.

With the use of top one probability, we adopt cross entropy to measure the distance between two ranked list of items: the ground truth list l_i and the predicted list $\widehat{l}_i$ generated by Eq. 3. Hence the latent vectors are learned to fit ranked item lists by minimizing the following objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \underbrace{-\sum_{j=1}^{M} I_{i,j} P_{l_i}(R_{ij}) log(\widehat{P}_{l_i}(\widehat{R}_{ij}))}_{\text{Cross entropy for user } i} + \frac{1}{2}\lambda(||U||_F^2 + ||V||_F^2), \quad (4)$$

where I_{ij} is an indicator function, which equals 1 if user i rated item j, and 0 otherwise. Parameter λ is regularization coefficient used to reduce over-fitting, while $||\cdot||_F^2$ denotes Frobenius norm. Note that, Eq. 3 can be extended to any ranking based objective function for including social information, in addition to cross entropy loss.

A local minimum of the loss function given by Eq. 4 can be found by performing gradient descent in latent vectors U_i and V_j. Note that, unlike the constant learning step size as used in [1], we set it in SoRank to be as large as possible in each iteration, as long as it leads to a decrease in the loss function Eq. 4.

2.3 Complexity analysis

Evaluating loss function $\mathcal{L}$ and its gradients comprise the computation process of SoRank: (1) The computational complexity of evaluating the loss function is $O(\rho D + \rho \bar{k} D)$, where ρ is the number of observed ratings, and $\bar{k}$ is the average number of friends that a user trust. (2) Computing the gradients $\frac{\partial \mathcal{L}}{U_i}$ and $\frac{\partial \mathcal{L}}{V_j}$ are of complexity $O(\rho D + \rho \bar{k} D + \rho \bar{p} D + \rho \bar{k} \bar{p} D)$ and $O(\rho D + \rho \bar{u} \bar{k} D)$, respectively, where $\bar{p}$ is the average number of friends who trust a user, and $\bar{u}$ is the average number of items that a user rated. Therefore, the total computational complexity in one iteration is in the order of $O(\rho \bar{m} \bar{k} D)$, where $\bar{m} = max(\bar{u}, \bar{p})$. Since we usually have $\bar{m}, \bar{k} \ll \rho$, the complexity is linear to the number of observed ratings. Overall, our analysis shows that SoRank is suitable for large dataset.

3. EXPERIMENTS

The dataset used in our experiments is Epinions dataset[1], which contains 22,166 users who expressed 922,267 ratings for 296,277 items. In addition to rating information, trust relations are also available in Epinions. The total number of issued trust statements is 355,754. We use the same strategy with [2] [3] to generate training and test set: we randomly select 10, 20 and 30 items for each user for training and use the remaining rated items for testing. To perform a direct and fair comparison, we also adopt the evaluation metric used in [2] [3], i.e., NDCG@10. For SoRank, D is set to 10, and $\lambda = 0.01$. α is set to 0.4, 0.5 and 0.9 in the three training set, respectively. These parameters are tuned in order to yield the best performance based on a validation set of the dataset.

We compare SoRank with CofiRank [3] and ListRank [2]. CofiRank is a learning to rank model which directly optimizes NDCG metric for ranking. ListRank also adopts cross entropy loss to learn a ranking function but without counting social relations among users.

The experimental results, in terms of NDCG@10, are shown in Fig. 1, from which we obtain one key observation. In all scenarios, SoRank significantly outperforms CofiRank and ListRank, which only utilize user-item information. This approves the motivation of this paper that incorporating social information will improve quality of ranked list.

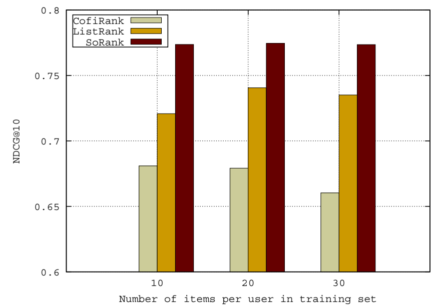

Figure 1: NDCG@10 comparison.

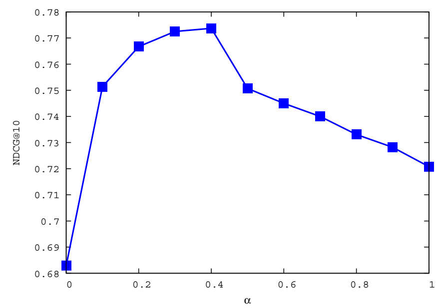

Figure 2: Impact of parameter α.

We illustrate the trend of NDCG@10 of SoRank with the increase of α in Fig. 2 under the condition with 10 rated items per user for training. The general observation is that the value of NDCG@10 increases first until reaching the peak, then decreases. This demonstrates that using either rating information or utilizing trusted friends' tastes only for recommendation can not generate better results than fusing these two information resources together.

4. CONCLUSION

Based on the assumption that user's behaviors can be affected by his/her socially connected friends, we explore a new improvement space for learning to rank models, with application to item recommendation. To the best of our knowledge, it is the first attempt that adapts social learning to rank for recommendation.

5. ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (Grant No. 61272480).

6. REFERENCES

[1] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136. ACM, 2007.

[2] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys*, pages 269–272. ACM, 2010.

[3] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. Cofi rank-maximum margin matrix factorization for collaborative ranking. In *NIPS*, pages 1593–1600, 2007.

[1]http://www.public.asu.edu/ jtang20/datasetcode