Collaborative Factorization for Recommender Systems

Chaosheng Fan[†*], Yanyan Lan[‡], Jiafeng Guo[‡], Zuoquan Lin[†], Xueqi Cheng[‡]
†School of Mathematical Sciences, Peking University, Beijing, P.R.China
‡Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R.China
fcs@pku.edu.cn, {lanyanyan,guojiafeng}@ict.ac.cn
lz@pku.edu.cn, cxq@ict.ac.cn

ABSTRACT

Recommender system has become an effective tool for information filtering, which usually provides the most useful items to users by a top-k ranking list. Traditional recommendation techniques such as Nearest Neighbors (NN) and Matrix Factorization (MF) have been widely used in real recommender systems. However, neither approaches can well accomplish recommendation task since that: (1) most NN methods leverage the neighbors' behaviors for prediction, which may suffer the severe data sparsity problem; (2) MF methods are less sensitive to sparsity, but neighbors' influences on latent factors are not fully explored, since the latent factors are often used independently. To overcome the above problems, we propose a new framework for recommender systems, called collaborative factorization. It expresses the user as the combination of his own factors and those of the neighbors', called collaborative latent factors, and a ranking loss is then utilized for optimization. The advantage of our approach is that it can both enjoy the merits of NN and MF methods. In this paper, we take the logistic loss in RankNet and the likelihood loss in ListMLE as examples, and the corresponding collaborative factorization methods are called CoF-Net and CoF-MLE. Our experimental results on three benchmark datasets show that they are more effective than several state-of-the-art recommendation methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Recommender System, Nearest Neighbors, Matrix Factorization, Learning to Rank

*The work was performed when the first author was a visiting student at Institute of Computing Technology, Chinese Academy of Sciences.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

1. INTRODUCTION

Recently, recommender system has become an effective tool for information filtering, and has played an important role in many popular web services, such as Amazon, YouTube, Netflix, Yahoo! etc. In most applications, the core task of a recommender system is to well predict the utilities of different items, and then return to users top-k useful items by a ranking list.

In literature, the main recommendation techniques can be divided into two categories, i.e. Nearest Neighbors (NN) and Matrix Factorization (MF). Nearest neighbors methods [2], leverage user's neighbors behavior for recommendation. However, this approach may fail as the severe data sparsity in recommendation makes the computation of of neighbors not accurate any more. Compared with NN, MF methods, such as SVD [9] and SVD++ [4], adopt the latent factors model to uncover the interests of a user, thus it is less sensitive to data sparsity. However, the latent factors of different users are usually used independently, and the neighbors' influences are rarely explored.

In order to take the advantages of both techniques, we propose a new framework called collaborative factorization (CoF for short). The central idea is to introduce the collaborative philosophy into matrix factorization by defining the latent factors of a user as the combination of his own factors and those of his neighbors' in traditional MF, called collaborative latent factors. This is also in accordance with the intuition that when making a decision, the user's own opinion and his/her friends' suggestions may both play an important role.

In this framework, we treat recommendation as a ranking problem, which is more consistent with the typical output of recommender systems (i.e. a ranking list of most useful items). A ranking loss defined on the basis of the collaborative latent factors is then utilized for optimization. To avoid overfitting, regularizations are further taken in CoF. In this paper, we use both pairwise and listwise losses since they are two dominant ranking losses in the literature of learning to rank. Specifically, we use the logistic loss in RankNet [5] and the likelihood loss in ListMLE [6], and the corresponding methods are called CoF-Net and CoF-MLE, respectively.

Finally, we conduct extensive experiments on the datasets of Movielens and Yahoo!Rand. The experimental results show that our proposed CoF-Net and CoF-MLE can significantly outperform several state-of-the-art recommendation methods, including NN, MF and the Hybrid methods.

To sum up, the contribution of this paper lies in that:

(1) The introduction of collaborative latent factors, which can be viewed as a novel approach to leverage collaborative philosophy in matrix factorization;

(2) The proposal of a unified framework to solve the recommendation problem, which is convenient to accommodate any form of ranking losses.

The rest of the paper is organized as follows. In Section 2 we briefly discuss related work. The collaborative factorization framework is presented in Section 3. The experimental results are discussed in Section 4. We summarize our work in the last section.

2. RELATED WORK

Nearest Neighbors [2] is a classical recommendation approach. It leverages users' neighbors who have similar rating behavior for prediction. However, the rating data is often highly sparse, which makes the chosen neighbors unreliable.

Matrix factorization [1, 4, 8, 9, 10, 11] is another popular recommendation technique. It explains ratings as the inner product of the latent factors of items and users. The usage of low rank latent factors makes it less sensitive to sparsity, but the neighbors' influences are highly ignored.

There are also some work trying to combine the two techniques to enhance recommendation, such as re-ranking model and hybrid model [4, 12]. For example, Koren et al. [4] proposed to directly combine matrix factorization model and item-based neighborhood model from algorithm-level to make more accurate model. Our work is different from their methods: (1) they aim at minimizing rating prediction error, while we use ranking based loss; (2) they do not fully explore neighbors' influences, while we introduce the collaborative idea in expressing user latent factors.

Please note that Ma etc. [7] conduct similar user representation, however, the objective function in their work is regression-based loss, and the social connections are explicitly given. Compared with their approach, our setting is more general and the motivation is different.

3. OUR APPROACH

Our approach uses latent factors model (i.e. matrix factorization) as its centric form. A latent factors model for recommendation can be formalized in the following way. Given a set of m users $\mathcal{U}$, a set of n items $\mathcal{I}$, and an observed rating matrix R, one aims to learn two low rank matrixes P and Q for users and items respectively by optimizing the following objective function:

$$\min_{P,Q} L(R,P,Q)+\lambda_P\|P\|_F^2+\lambda_Q\|Q\|_F^2, \quad (1)$$

where $L(R,P,Q)$ is the loss function, P is a $k\times m$ matrix, Q is a $k\times n$ matrix, and k is the dimension of latent factors. $\|P\|_F^2$ and $\|Q\|_F^2$ are regularization items to overcome overfitting and λ_P and λ_Q are parameters. The idea behind latent factors model is that ratings can be explained as the inner product of the latent factors of items and users.

In traditional matrix factorization [9], recommendation task is viewed as a regression problem where one aims to approximate the observed rating matrix R as the inner product of matrix P and Q. Therefore, the loss function takes the form of mean square error as following

$$L(R,P,Q)=\sum_{u\in\mathcal{U},i\in\mathcal{I}}(R_{ui}-p_u^Tq_i)^2. \quad (2)$$

where column vector p_u in P stands for the factors of user u, and column vector q_i stands for the factors of item i.

From the above process, we can see that the latent factors are used independently, and the neighbors' influences is ignored in MF methods.

In our approach, we show how to introduce collaborative philosophy into MF and propose a novel framework for recommendation, namely collaborative factorization. Here we first introduce the collaborative latent factors.

3.1 Collaborative Latent Factors

Inspired by the intuition that when making a decision, the user's own opinion and his friends' suggestions are both taken into consideration, we propose to model the latent factors of a user as the combination of his/her own factors and those of his/her neighbors', as defined as follows, called collaborative latent factors. The formalization is as follows

$$\tilde{p}_u=\alpha p_u+(1-\alpha)\sum_{v\in F(u)}\frac{S_{uv}}{\sum_{w\in F(u)}S_{uw}}p_v, \quad (3)$$

where α is the parameter to show the trade-off between the user's own interest and his neighbors' influences, $F(u)$ stands for the set of neighbors of user u, and S_{uv} (S_{uw}) stands for the similarity between user u and v (w). Any user-user similarity metric can be used in the above definition.

With the collaborative latent factors, ratings R_{ui} can be represented as the inner product of the corresponding factors, formulated as follows.

$$R_{ui}=\tilde{p}_u^Tq_i=(\alpha p_u^T+(1-\alpha)\sum_{v\in F(u)}\frac{S_{uv}}{\sum_{w\in F(u)}S_{uw}}p_v^T)q_i. \quad (4)$$

3.2 Collaborative Factorization

With the collaborative latent factors defined above, we now formalize our collaborative factorization framework. Since the typical output of a recommender system is usually a top-k ranking list, we naturally treat recommendation as a ranking problem. Therefore, the goal of collaborative factorization is to minimize the following subject:

$$\min_{P,Q}\sum_{u\in\mathcal{U}}L(R_{u1},\cdots,R_{un})+\lambda_P\|P\|_F^2+\lambda_Q\|Q\|_F^2. \quad (5)$$

where $L(R_{u1},\cdots,R_{un})$ is a ranking loss of user u's ratings on item set $\mathcal{I}$, and the other notations are the same as mentioned before.

We can see that the proposed collaborative factorization is a general framework which can accommodate any kind of ranking losses. In this paper, we adopt the pairwise and listwise losses, which are two dominant ranking losses in the literature of learning to rank. We call the corresponding approach as pairwise collaborative factorization and listwise collaborative factorization.

3.2.1 Pairwise Collaborative Factorization

In pairwise collaborative factorization, a pairwise loss L_p is used, and the ranking loss on the item sets are defined as the sum of losses on all the pairs:

$$L(R_{u1},\cdots,R_{un})=\sum_{(i,j)\in D_u}L_p(R_{ui},R_{uj}). \quad (6)$$

where D_u stands for the pairs constructed from user's rating profile. For example, if a user has rated item i higher than j, then the pair (i,j) is included in D_u.

Any pairwise ranking loss in learning to rank can be used in Eq. (6), such as the hinge loss in RankSVM, the exponential loss in RankBoost, and the logistic loss in RankNet. In this paper, we take the logistic loss in RankNet as an example, and obtain the following method, named CoF-Net.

$$-\sum_{u\in\mathcal{U}}\sum_{(i,j)\in D_u}\log(\frac{1}{1+\exp\{-(R_{ui}-R_{uj})\}}). \quad (7)$$

3.2.2 Listwise Collaborative Factorization

In listwise collaborative factorization, a listwise loss L_l is used, and the ranking loss on the item sets are defined as the loss on the generated list:

$$L(R_{ui},\cdots,R_{un}) = L_l(R_{ui},\cdots,R_{un},\sigma_u). \quad (8)$$

where σ_u stands for the permutation generated from user's rating profile. For example, if a user has rated item i higher than j, and j higher than k, then the generated permutation is (i,j,k).

Any listwise ranking loss in learning to rank can be used in Eq. (8), such as the likelihood loss in ListMLE, the entropy loss in ListNet, and the cosine loss in RankCosine. In this paper, we take the likelihood loss in ListMLE as an example, and obtain the following method, named CoF-MLE.

$$-\sum_{u\in\mathcal{U}}\sum_{i=1}^{|R_u|-1}\log\left(\frac{\exp\{R_{u\sigma^{-1}(i)}\}}{\sum_{j=i}^{|R_u|}\exp\{R_{u\sigma^{-1}(j)}\}}\right). \quad (9)$$

where $\sigma^{-1}(i)$ represents the item ranked in position i of σ, and $|R_u|$ is the length of user u's rating profile.

For optimization, we just use stochastic gradient descent in this paper. We omit the details of algorithm iteration framework due to the lack of space and the reader can refer to [8, 10] for similar description.

4. EXPERIMENT

In this section, we conduct extensive experiments on benchmark datasets to show the superiority of our proposed collaborative factorization methods.

4.1 Experimental Settings

Datasets: We use three benchmark datasets for experiments: Movielens100K, Movielens1M[1] and Yahoo!Rand[2].

The Movielens100K dataset consists of about 100 K ratings made by 943 users on 1628 movies. The Movielens1M dataset consists of about 1 M ratings made by 6040 users on 3706 movies. Following the experimental strategies used in [10, 13], for each user, we sample N ratings for training, and sample 10 ratings from training set to tune hyper parameters (validation set). After that, we fix the hyper parameters and the validation set are added to the original training set, then we retrain the model. In our experiments, we set N as 20 and 50, and users with less than 30 and 60 ratings are removed to ensure that NDCG@10 can be obtained.

The Yahoo!Rand dataset is a much newer dataset. It contains ratings for songs collected from two different sources. The first one consists of 300,000 ratings supplied by 15,400 users with at least 10 existing ratings during normal interaction with Yahoo! Music services between 2002 and 2006. The second one consists of ratings for randomly selected songs collected during an on-line survey conducted by Yahoo! Research and hosted by Yahoo! Music between August 22, 2006 and September 7, 2006. 5400 Participants were asked to rate 10 songs selected at random from a fixed set of 1000 songs. All the ratings from the first source are used for training, and ratings from the second source is used for testing. The experiments on this dataset are more convincing, since the test data are collected on random sampled songs.

Baseline Methods: Besides the typical collaborative filtering methods such as user-based K-Nearest-Neighbors (KNN) [2] and matrix factorization methods aim to minimize RMSE (Root Mean Square Error), e.g. Probabilistic Matrix Factorization (PMF) [9], we also use state-of-the-art ranking based methods including BPR [8] and ListMLE-MF [11, 10] as baselines. Furthermore, we also conduct the experiments on a hybrid method (HYBRID) (Equation 16 in [4]). Here we modify the HYBRID method by learning user-user weight (similarity) other than the original item-item weight, since we focus on investigating the user-wise neighborhood effect throught this paper.

We find that further precision improvements can be achieved by extending global average rating, user rating bias, item rating bias, item-based implicit feedbacks and item-based neighbors information to our framework (we verify this in separated experiments not reported here). But we omit these terms like most of the collaborative ranking methods [1, 8, 9, 10, 11, 13, 14] to keep fairly comparison with our baselines.

Hyper Parameters: Latent factors dimension is fixed as 5 in all methods. Learning rate and regularization values are chosen from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for every model. The size of nearest neighbors (chosen by Pearson correlation coefficient) is set to 100 in HYBRID. For CoF-Net and CoF-MLE, we choose neighbors by cosine similarity and set the nearest neighbors size as 50 which makes them work very well. We don't investigate the effect of other neighborhood size and other similarity in detail due to the page limitation. The best value of combination coefficient α is 0.15 for CoF-Net, and 0.65 for CoF-MLE.

Evaluation Measures: The traditional evaluation measure RMSE places equal emphasis on high rating and low rating. Therefore, it is not suitable for the top-k recommendation scenario. In this paper, we choose Normalized Discount Cumulative Gain (NDCG) [3] as the evaluation measure, which is a popular evaluation metric in the community of information retrieval. NDCG can leverage the relevance judgment in terms of multiple ordered categories, and has an explicit position discount factors in its definition as follows:

$$\text{NDCG}(u,\pi)@\text{K} = \frac{1}{N_K(u)}\sum_{p=1}^{K}\frac{2^{R_{u,\pi^{-1}(p)}}-1}{\log_2(p+1)}, \quad (10)$$

where $\pi : \{1,2,...,N\} \to \{1,2,...,N\}$ is a permutation generated in the descending order of the predicted ratings, and $\pi^{-1}(p)$ stands for the item that is ranked in position p of the ranking list π. $N_K(u)$ is the normalization factors. We set K to 10 like in [1, 10, 14], and we report the average NDCG@10 over all users in our experiment.

4.2 Experimental Results

The experiment results are listed in Table 1. From the results, we can see that:

[1]http://www.grouplens.org/node/73

[2]http://webscope.sandbox.yahoo.com/

Table 1: Ranking Performance

Dataset	Movielens100K		Movielens1M		Yahoo!Rand
Algorithm	N = 20	N = 50	N = 20	N = 50	|Test Items| = 10
KNN	0.640	0.675	0.643	0.657	0.802
PMF	0.681	0.691	0.729	0.737	0.823
HYBRID	0.710	0.714	0.750	0.754	0.830
ListMLE-MF	0.707	0.711	0.741	0.748	0.825
BPR	0.710	0.714	0.742	0.755	0.835
CoF-MLE	0.712	0.715	0.752	0.757	0.828
CoF-Net	**0.722**	**0.726**	**0.756**	**0.775**	**0.849**

(1) Models combining MF and NN will obtain enhanced performances. For example, on Yahoo!Rand, the NDCG@10 of KNN, PMF and HYBRID is 0.802, 0.823, and 0.830, respectively. KNN obtains the lowest performance due to the unreliable neighbors computation on sparse data. PMF works better since it employ dimensionality reduction to relieve sparsity. By combining MF and NN, HYBRID obtains even better performance. Similar results can be found in the ranking based methods. By introducing the collaborative latent factors, CoF-MLE consistently outperforms ListMLE-MF (about 1%) and CoF-Net consistently outperforms BPR (about 2%).

(2) RMSE minimization is not optimal for top-k recommendations. From the results in Table 1, we can see that models which aim at minimizing rating prediction error perform worse than the ranking based models. For example, PMF obtains lower NDCG@10 scores than all of our ranking based CoF methods (i.e. CoF-MLE, CoF-Net, BPR, LISTMLE-MF).

(3) As the methods combining MF and NN, the proposed collaborative factorization (CoF) outperforms the traditional hybrid methods (HYBRID). From Table 1, CoF-Net is the best model and it significantly outperform HYBRID (about 2%). We can see that by introducing the collaborative idea to the representation of user latent factors, the neighbors' effect are better explored in CoF than HYBRID, which directly combines MF and NN from algorithm-level.

5. CONCLUSION

In this paper, we propose a new recommendation framework to enjoy both the merits of nearest neighbors and matrix factorization, called collaborative factorization. Firstly, collaborative latent factors are defined as the combination of the user's own factors and those of his neighbors'. Secondly, a ranking loss is utilized as the objective for top-k recommendation. Our experiments on several benchmark datasets show that our approach can significantly outperform the traditional methods.

As for future work, we can further study whether there exists other ways to introduce collaborative idea to matrix factorization.

Acknowledgements

This research work was funded by the National Natural Science Foundation of China under Grant No. 61232010, No. 61203298, No. 61003166, No. 60973003, 863 Program of China under Grants No. 2012AA011003, and National Key Technology R&D Program under Grant No. 2011BAH11B02, No. 2012BAH39B02, No. 2012BAH39B04. We also wish to express our thanks to Yahoo!Research for providing the Yahoo!Rand dataset to us.

6. REFERENCES

[1] S. Balakrishnan and S. Chopra. Collaborative ranking. In *Proc. of WSDM '12*, pages 143–152, New York, USA, 2012.

[2] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, Jan. 2004.

[3] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[4] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1):1:1–1:24, Jan. 2010.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender, Learning to rank using gradient descent, In *Proc. of ICML '05*, pages 89–96, 2005.

[6] F. Xia, Tie-Yan Liu, J. Wang, W. S. Zhang and H. Li, Listwise Approach to Learning to Rank - Theory and Algorithm, In *Proc. of ICML '08*, pages 1192–1199, 2008.

[7] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proc. of SIGIR '09*, pages 203–210, New York, USA, 2009.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI '09*, pages 452–461, Arlington, USA, 2009.

[9] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Proc. of NIPS '07*, 2007.

[10] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proc. of RecSys '10*, pages 269–272, New York, USA, 2010.

[11] T. Tran, D. Q. Phung, and S. Venkatesh. Learning from ordered sets and applications in collaborative ranking. *Journal of Machine Learning Research - Proceedings Track*, 25:427–442, 2012.

[12] Gideon Dror, Noam Koenigstein, Yehuda Koren, Markus Weimer The Yahoo! Music Dataset and KDD-Cup11. *Journal of Machine Learning Research - Proceedings Track*, 18:318, 2012.

[13] M. N. Volkovs and R. S. Zemel. Collaborative ranking with 17 parameters. In *Proc. of NIPS '12*, 2012.

[14] M. Weimer, A. Karatzoglou, and M. Bruch. Maximum margin matrix factorization for code recommendation. In *Proc. of RecSys '09*, pages 309–312, New York, USA, 2009.