Expanding User Features with Social Relationships in Social Recommender Systems

Chengjie Sun, Lei Lin, Yuan Chen, and Bingquan Liu

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
{cjsun,linl,ychen,liubq}@insun.hit.edu.cn

Abstract. Although recommender system has been studied for many years, the research of social recommender system is just beginning. Plenty of information can be used in social networks to improve the performance of recommender system. However, some information is very sparse when used as features. We call this feature sparsity problem. In this paper, we aimed at solving feature sparsity problem. A new strategy was proposed to expand user features by social relationships. Experiments on two real world datasets demonstrated that our method can significantly improve the recommendation performance.

Keywords: social recommender system, feature expanding, social relationships, feature sparsity.

1 Introduction

With the rapid development of the Internet, social networking services such as Twitter, Facebook, Last.fm and Tencent Weibo become more and more popular, even become an important part of many people's daily life. There are thousands of new users joining in these social networks and huge amount of information being generated from there. For example, there are more than 200 million registered users currently and 40 million messages being generated each day on Tencent Weibo [1]. People make friends, get information, and express their views on social network day and night. While benefiting us a lot, it also wastes us too much time to search useful information from massive information. So how to choose good information sources is very important for users in social networks. In different social networks, information sources are different, such as users in microblog, artists in music online system and sellers in e-commerce website. There are many ways to choose information sources, like getting from friend's recommendation or using search engine. But getting from friend's recommendation is inefficient and sometimes it is hard to express the requirements by a few keywords for many users when using search engine in social networks. Therefore, to deal with the problem of choosing information sources efficiently, it is important to develop effective social recommender system to help users easily find the potential items they want to follow.

Social recommender system is the kind of recommender system whose target domain is social media. Although all kinds of recommender systems' common goal is modeling the relation between users and items, social recommender system has much

G. Zhou et al. (Eds.): NLPCC 2013, CCIS 400, pp. 247–254, 2013.

more social information to use than traditional recommender system such as e-commerce recommender system and music recommender system. Social information includes social relationships, user's profile, user's social action and so on. How to take advantage of these information to improve the performance of social recommender system is both a good way and a big challenge. In this paper, we concentrate on exploring the methods of combining social relationships into recommendation algorithm to improve the performance of social recommender system.

The rest of this paper is organized as follows. Section 2 will introduce the background and related works of our research. Section 3 will discuss the methods of modeling social relationships with recommendation algorithm. In section 4， we will present our experiments and results on two real world datasets. Then we will make our conclusion in section 5.

2 Related Work

We will review the related works of social recommender system from two aspects: the popular methods of recommender system and the research of social recommendation.

The goal of recommender system is suggesting users with items which may fit the users' tastes. Recommendation methods can be classified into the following three categories: content-based, collaborative filtering (CF), and hybrid recommendation approaches [2]. Recently, as one of the most successful methods, CF has been widely used in business and deeply studied in academia. CF is a kind of algorithm whose fundamental assumption is that if user A and user B rate k items similarly, they share similar tastes, and hence will rate other items similarly. Approaches differ in how they define a "rating", how they define k, and how they define "similarly" [3]. One of the major factors which limit the performance of CF is data sparsity. In real commercial recommender system, the data density is often lower than 1% [4]. To deal with the data sparsity challenge, many algorithms of CF have been proposed by recent years, such as Singular Value Decomposition (SVD) [5] and SVD++ [6]. SVD is a method of matrix factorization which can reduce the dimension of rating matrix. As a result, it represents the original rating matrix information by several latent factors with lower dimension and higher data density. However, this is at the cost of information loss. SVD++ is an extension of SVD, which can handle external information of rating matrix to improve the performance.

Social recommendation aims at forming a recommender system by using recommendation methods with information from social networks such as user profile, user action, and social relationships [7]. User recommendation in social networks has become one of the hottest topics [8]. The task of KDD Cup 2012 Track1 is to develop such a system aiming at recommending items (celebrities) which could be persons, organizations, or groups in Tencent Weibo. In [9-11], researchers have tried to improve the recommender system's performance by sufficiently making use of social information, such as the items user followed in the past, user age and gender, user's click rate and so on. For another task, artist recommendation in Last.fm online music system, [12] proposes a novel hierarchical Bayesian model which jointly incorporates topic

modeling and probabilistic matrix factorization of social networks. However, in the above situations, researchers have not taken full use of one important kind of information in social networks, the social relationships among users. In our paper, we will present how to utilize social relationships to expand user features to improve the performance of social recommender system.

3 Methods

3.1 Latent Factor Model

Recently, latent factor models have been widely used in rating prediction problem. The main idea of latent factor model is to predict a rating $\tilde{r}_{u,i}$ by the dot product of a user latent factor p_u and an item latent factor q_i which can be learned from history rating matrix. Singular Value Decomposition (SVD) is one of the most popular latent factor models. The basic SVD model with user bias, item bias and global bias can be described in the following equation:

$$\tilde{r}_{u,i} = p_u^T \cdot q_i + b_u + b_i + b_g \tag{1}$$

Here b_g is the global bias, b_u is the user bias, b_i is the item bias, p_u is the user latent factor and q_i is the item latent factor. Instead of L2 loss function, we choose pairwise loss function in this work. So the above parameters can be trained by minimizing the following cost function:

$$\min \sum_{\substack{<u,k,h> \\ \in D}} \ln(1+e^{-(\tilde{y}_{u,k}-\tilde{y}_{u,h})}) + \lambda_1 \| p_u \|^2 + \lambda_2 \| q_i \|^2 + \lambda_3 \| b_u \|^2 + \lambda_4 \| b_i \|^2 + \lambda_5 \| b_g \|^2 \tag{2}$$

Here D is the set all $\langle u, k, h \rangle$ pairs in training set, k is item u gives positive review and h is item u gives negative review. In the equation, the first term is used to learn the best parameters, the other terms are regularizations with constants to avoid overfitting.

Stochastic Gradient Descent algorithm (SGD) [5] is an efficient and popular strategy to solve the above optimization problem. The main idea of SGD is to minimize the error by taking a small step on parameters along the direction of gradient descent in each loop. SGD uses the following update rules to learn the model parameters:

$$p_u \leftarrow p_u + \eta(e_{ui} p_u - \lambda_1 p_u) \tag{3}$$

$$q_i \leftarrow q_i + \eta(e_{ui} q_i - \lambda_2 q_i) \tag{4}$$

$$b_u \leftarrow b_u + \eta(e_{ui} b_u - \lambda_3 b_u) \tag{5}$$

$$b_i \leftarrow b_i + \eta(e_{ui} b_i - \lambda_4 b_i) \tag{6}$$

$$b_g \leftarrow b_g + \eta(e_{ui} b_g - \lambda_5 b_g) \tag{7}$$

Here e_{ui} is the prediction error and η is the learning rate.

3.2 Incorporate Implicit Feedback Information as Features

The basic SVD model is good at handling simple dataset which only includes rating matrix information. However, the datasets are usually much more complicated for building recommender system in the real world. For example, in a movie recommender system, except for the rating matrix information, there are many other kinds of information which may include movie tags, movie category, user profile, user social network and so on. These information can be called implicit feedback information [5]. How to utilize these diverse implicit feedback information is beyond the ability of SVD, while SVD++ can do it well by modeling them as latent features. The only thing has to be done is extending the SVD equation to the following form:

$$\tilde{r}_{u,i} = (p_u^T + \sum_{j \in N(u)} \alpha_j y_j) \cdot q_i + b_u + b_i + b_g \tag{8}$$

The adding sum term represents the perspective of implicit feedback. $N(u)$ can be user's neighbors, tags or other implicit feedbacks. y_j is the latent factor and α_j is its weight.

3.3 Expand Features with Social Relationships

Social relationships are not usually used in the same way as other information such as tags and profiles which are used via a factorization process, while social relationships are used via a regularization process [13]. Using social relationships as factorization has been proved to be effective, but to my best knowledge this has just been used in the basic SVD model and the social network is not sufficient enough. This is probably because in SVD++ model or in big social network there are too many latent factor parameters to learn with social regularization. So how to use social relationships of big social network in SVD++ model is a still a big challenge.

We find that it is easy to add features in SVD++ model, while it is hard to model social relationships as features directly. Therefore we need to find a compromise way. The main idea is that we can expand some user features by using social relationships. From the social network, we can get user u 's friends set $F(u)$. For each friend i in $F(u)$, we can get the same kind of feature z_i as u 's feature and β_i is its weight. We call the friends' features as expanding features. The equation of user latent factor in SVD++ with the expanding features can be described as following:

$$p_u \leftarrow p_u^T + \sum_{j \in N(u)} \alpha_j y_j + \sum_{i \in F(u)} \beta_i z_i \tag{9}$$

For example, in the friend recommendation problem of social media, some user information such as user's tags, keywords and history following records are very sparse which will lead to the features being ineffective. Using social relationships can deal with the feature sparsity problem to some extent. We can use friends' tags, keywords and history following records as a kind of expanding features.

4 Experiments

4.1 Data Description

We conducted our experiments on two real world datasets. The first one is a subset of dataset of Track 1 in KDD Cup 2012 [1] provided by Tecent Weibo which is one of the two biggest microblog platforms in China. The dataset is a snapshot of Tencent Weibo of five days. We used the first three days records as training set and next two days records as testing set. Except training set and testing set, there are lots of other information including user profile, keywords, item categories, social action and social graph. This is a friend recommendation problem. The items in the dataset are a small group of specific users which can be celebrities, famous organizations or some well-known groups. Another dataset is hetrec2011-lastfm-2k [14] obtained from Last.fm online music system. There are listening histories, tagging histories and friend relationships in the dataset. The items in the dataset are artists. So this is an artist recommendation problem. We prepared training set and testing set by dividing listening histories into 9 to 1. Because there are only positive records, we produced 20 negative records by randomly selecting 20 artists whom have not been listened to for each positive record. Table 1 shows the statistical properties of the datasets.

Table 1. Statistical properties of the datasets

Element	Size (Tencent Weibo)	Size (Last.fm)
Training set	7,105,220	1,554,819
Testing set	4,265,397	392,910
Users	266,615	1892
Items	6,095	17,632
Edges of social group	50,655,143	25,434

4.2 Evaluation Metrics

We used the MAP@n (Mean Average Precision of Top N) and recall@n for two datasets respectively as our evaluation metrics which are the same as the metric used in the competition and other researchers. The metric is defined as:

$$ap@n = \sum_{k=1}^{n} P(k) / m \tag{10}$$

$$MAP@n = \sum_{i=1}^{N} ap@n_i / N \tag{11}$$

$$recall@n = \frac{number\ of\ items\ the\ user\ likes\ in\ Top\ n\ recommendations}{total\ number\ of\ items\ the\ user\ likes} \tag{12}$$

where $P(k)$ is the precision at cut-off k in the recommender item list, m is the number of items clicked by the user and N is the number of users.

4.3 Experiment Description

For Tencent Weibo dataset, we conducted our experiments on four user features: follow histories, tags, keywords and actions. Follow histories refer to items one user has followed. Tags are a few keywords from user's profile added by himself or herself. The number of tags is less than ten and usually three to eight. Keywords are selected from user's tweets whose number is more than that of tags and usually twenty to fifty. Actions mean the number of comments, at, retweets from a user to another user.

The definition of friends is different with environments. In Tencent Weibo dataset, the social network is directed. There are three kinds of relationships between two users: follow, be followed, and follow each other. We just used the follow relationship because it is the result of active action which is more representative of one user's interests. So friends in our experiments on Tencent Weibo dataset mean other users followed by the user.

For Last.fm dataset, we conducted our experiments on the feature of tags. There is a little different from tag feature in Tencent Weibo dataset. Tag features in Last.fm dataset are tags a user has made for artists. It is also a kind of implicit feedback information representing user's tastes.

There are many friends for some users and each friend has several features, so this will lead to too many expanding features for a user. For example, a user has 100 friends and each friend has 5 tags. The total number of the user's friends' tags is 500, which is too large to use. Therefore, we need a selection strategy. In our experiments, we chose k most common ones of user's friends' features. The weight of each feature is its count in friends' features after normalization. We tried both k=30 and k=20 and different weight strategies, but experiment results indicated that there were not much differences. So k is set to 30 in the following reported results. The SVDFeature toolkit [15] was used in our experiments.

4.4 Results and Analysis

Table 2 and Table 3 showed the results of our experiments on Tencent Weibo dataset and Last.fm dataset. We can see that all of the expanding features can achieve improvement when used independently or in combination. Except follow history in Tencent Weibo dataset, other expanding features in both datasets were even more useful than the original features. This is because the original features of tags, keywords, actions are every sparse while the expanding features are denser.

Table 2. Improvement on MAP@3 in Tencent Weibo Dataset

Feature	Model	MAP@3
	SVD	0.3595
follow history	SVD + follow history	0.3714
	SVD + friends' follow history	0.3677
	SVD + follow history + friends' follow history	0.3743
tags	SVD + user tags	0.3609
	SVD + friends' tags	0.3629
	SVD + user tags + friends' tags	0.3669
keywords	SVD + user keywords	0.3653
	SVD + friends' keywords	0.3705
	SVD + user keywords + friends' keywords	0.3770
action	SVD + user action	0.3644
	SVD + friends' retweet	0.3676
	SVD + user action + friends' retweet	0.3712
follow history + tags + keywords + action	SVD + follow history + user tags + user keywords + user action	0.3823
	SVD + friends' follow history + friends' tags + friends' keywords + friends' retweet	0.3754
	SVD + follow history + friends' follow history + user tags + friends' tags + user keywords + friends' keywords + user action + friends' retweet	0.3853

Table 3. Improvement on recall@n in Last.fm Dataset

Model + Feature	recall@50	recall@100	recall@150	recall@200	recall@250
SVD	14.91%	22.45%	27.99%	34.31%	38.65%
SVD + user tags	15.93%	25.87%	31.87%	40.46%	44.43%
SVD + friends' tags	17.23%	25.97%	33.11%	43.14%	47.84%
SVD + user tags + friends' tags	21.22%	29.67%	36.03%	43.23%	48.25%

5 Conclusion

In this paper, we explored a new way of using social relationships by expanding user features in social recommender systems. We trained SVD++ model on Tencent Weibo and Last.fm datasets. Experiment results showed that expanding features by social relationships can improve the performance of recommender system efficiently which indicated that it can help to solve features' sparsity problem.

Acknowledgments. We thank three anonymous reviewers for their helpful comments on an earlier version of this work. This work is supported by the National Natural Science Foundation of China (61100094, 61300114) and Research Fund for the Doctoral Program of Higher Education of China (20102302120053).

References

1. Niu, Y., Wang, Y., Sun, G., Yue, A., Dalessandro, B., Perlich, C., Hamner, B.: The tencent dataset and kdd-cup'12. In: KDD-Cup Workshop 2012 (2012)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
3. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant-Time Collaborative Filtering Algorithm. Information Retrieval 4(2), 133–151 (2001)
4. Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, pp. 285–295. ACM Press, New York (2001)
5. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
6. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: KDD 2008: Proceeding of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 426–434. ACM, New York (2008)
7. King, I., Lyu, M.R., Ma, H.: Introduction to social recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1355–1356. ACM, New York (2010)
8. Ma, T.L., Yang, Y.J., Wang, L.W., Yuan, B.: Recommending People to Follow Using Asymmetric Factor Models with Social Graphs. In: Proceedings of the 17th Online World Conference on Soft Computing in Industrial Applications. AISC, Springer (2012)
9. Chen, T., Tang, L., Liu, Q., Yang, D., Xie, S., Cao, X., Wu, C., Yao, E., Liu, Z., Jiang, Z., Chen, C., Kong, W., Yu, Y.: Combining factorization model and additive forest for collaborative followee recommendation. In: KDD-Cup Workshop 2012 (2012)
10. Rendle, S.: Network and Click-through Prediction with Factorization Machines. In: Workshop 2012 (2012)
11. Zhao, X.: Scorecard with Latent Factor Models for User Follow Prediction Problem. In: KDD-Cup Workshop 2012 (2012)
12. Purushotham, S., Liu, Y., Kuo, C.C.J.: Collaborative topic regression with social matrix factorization for recommendation systems. In: Proceedings of the 29th International Conference on Machinelearning. ACM, New York (2012)
13. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011), pp. 287–296 (2011)
14. Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In: Proc. of 5th ACM Conf. on Recommender Systems, RecSys 2011, pp. 387–388 (2011)
15. Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., Yu, Y.: SVDFeature: A Toolkit for Feature-based Collaborative Filtering. Journal of Machine Learning Research 13, 3619–3622 (2012)