Accumulative Influence Weight Collaborative Filtering Recommendation Approach

Nan Li and Chunping Li

Abstract. Memory-based collaborative filtering algorithms are widely used in practice. But most existing approaches suffer from a conflict between prediction quality and scalability. In this paper, we try to resolve this conflict by simulating the ”word-of-mouth” recommendation in a new way. We introduce a new metric named influence weight to filter neighbors and weight their opinions. The influence weights, which quantify the credibility of each neighbor to the active user, form accumulatively in the process of the active user gradually provides new ratings. Therefore, when recommendations are requested, the recommender systems only need to select the neighbors according to these ready influence weights and synthesize their opinions. Consequently, the scalability will be significantly improved without loss of prediction quality. We design a novel algorithm to implement this method. Empirical results confirm that our algorithm achieves significant progress in both aspects of accuracy and scalability simultaneously.

1 Introduction

Memory-based collaborative filtering approaches are widely used in practice to help people cope with the problem of information overload. But most of them suffer from problems such as poor prediction quality or poor scalability.

Recently many methods [5] [3] have been proposed to improve the prediction quality by alleviating data sparsity. The evaluations of these approaches showed that it is really effective. But some extra effort required in these approaches, such as smoothing the missing data, makes the scalability of them worse.

Nan Li
School of Software, Tsinghua University, Beijing 100084, China
e-mail: ln8209@gmail.com

Chunping Li
School of Software, Tsinghua University, Beijing 100084, China
e-mail: cli@tsinghua.edu.cn

B.-C. Chien and T.-P. Hong (Eds.): Opportunities and Challenges, SCI 214, pp. 73–78.
springerlink.com

Most approaches aiming at good scalability achieve their goals by taking advantage of some model-based strategies [5] or conducting some precomputation offline [4] in order to reduce the searching space of users (items) or similarity computation online. But these methods either would limit the diversity of users (items) or could not keep pace with the change in the user-item matrix. Thus they would definitely bring negative influence to prediction quality.

Unfortunately, there seems to be a conflict between prediction quality and scalability in the general framework of most existing memory-based approaches.

In this paper, we try to resolve this conflict by simulating the ”word-of-mouth” recommendation in a new way. We introduce a new metric named influence weight to filter neighbors and weight their opinions. The influence weights, which quantify the credibility of each neighbor to the active user, form accumulatively in the process of the active user gradually provides new ratings. Therefore, when recommendations are requested, the recommender systems only need to select the neighbors according to these ready influence weights and synthesize their opinions. Consequently, the scalability will be significantly improved without loss of prediction quality. We design a novel algorithm to implement this method. Empirical results confirm that our algorithm achieves significant progress in both aspects of accuracy and scalability simultaneously.

2 Related Work

Memory-based collaborative filtering approaches, which include user-based and item-based, search the most similar neighbors of the active user in the entire user-item database whenever they make predictions. User-based approaches [2] [5] predict for the active user based on the opinions of similar users, and item-based approaches [1] [4] make prediction based on the information of similar items.

Recently many methods have been proposed to improve the prediction quality by alleviating data sparsity problem. [5] proposes a novel framework for collaborative filtering which combines the strengths of memory-based approaches and model-based approaches in order to enable recommendation by groups of closely related individuals. [3] proposes an effective missing data prediction algorithm which predict the missing data by exploiting information of both user neighbors and item neighbors whose similarities are all higher than some thresholds.

To improve scalability, [4] proposes item-based collaborative filtering recommender algorithm which reduces online computation by utilizing the relatively static relationships between items. [5] exploits clustering techniques to reduce the searching space of potential neighbors and improve the scalability consequently.

3 “Word-of-Mouth” Recommendation

When a person, Tom, thinks about whether going to watch a new movie, he would ask friends with similar movie taste for advices. And then Tom will synthesize these advices according to the credibility of each friend. The credibility of each friend formed according to the validity of his past advices. After watching this movie, Tom will obtain his own opinion. Then he would adjust the credibility of his friends based on his opinion. The credibility of a friend will increase if his advice accord with Tom's opinion. Otherwise his credibility will decrease according to the deviation between his advice and Tom's opinion. It is natural that the lager the deviation is, the more credibility the giver loses.

We could quantify this credibility and use it as a metric to select neighbors and weight their opinions. Since the more similar with the active user a neighbor is, the larger his credibility would be, the concept of credibility actually reflect the similarity between each pair of users from another angle. Thus it may play the role as good as or even better than the similarity metrics such as Pearson Correlation Coefficient (PCC) or Vector Similarity (VS). More importantly, the credibility has two advantages over those broadly used similarity metrics.

On the one hand, to generate a recommendation, those algorithms using similarity metrics have to compute similarity scores tens of thousands of times, but when next recommendation is requested by the same user, they have to do the same heavy work again even if only a small portion of the user-item matrix has changed. In contrast, the credibility form accumulatively in the process of the active user continually provides new ratings. And the adjustment to the credibility would get done once and for ever whenever a new rating is provided. This means the overall work of a system adopting the credibility is approximately linear with the number of the ratings no matter how many recommendations would be requested. It is obviously that this would improve the efficiency significantly.

On the other hand, the limitation of responding latency requires those heavy computations to be completed in real time, which is exactly the bottleneck of performance and scalability in most existing memory-based approaches. Moreover, the recommender systems would be jammed by these real-time tasks in the rush hours but be free in most other time, which result in that the systems have to waste quite a lot of resource. In contrast, adjusting the credibility is not a real-time task. It could be conducted on background when the active user is providing new ratings. Or the systems could schedule these tasks freely just before next recommendation is requested by this user. When a recommendation is requested, all the necessary data are ready and the systems only need to conduct some simple retrieval and computation online. In this way, the systems could not only achieve very short responding latency to provide better user experience but also balance their load effectively to take use of the resource efficiently.

4 Accumulative Influence Weight Algorithm

We design a novel memory-based algorithm named Accumulative Influence Weight (AIW) to implement the method discussed above. We first formally define a metric named influence weight to quantify the credibility described above. AIW maintain a table named IW table to record the influence weights between each pair of existing users. When a new user registers, AIW add him into the IW table and set all the original influence weights relating to him 0. Then, whenever this user provides a new rating, AIW would adjust the influence weights between him and all the other existing users who have rated the same item with a zero-sum mechanism as follows:

Whenever a user u provides a new rating $r_{u,i}$ on item i,
1. select the users who have rated on i and divide them into two subsets:
S_1 contains the users whose rating on i are unequal to $r_{u,i}$
S_2 contains the users whose rating on i are equal to $r_{u,i}$
2. for each user a in S_1:
decrease the influence weight of a to u according to the deviation between $r_{a,i}$ and $r_{u,i}$,adjust the influence weight of u to a accordingly , and accumulate the deviations
3. for each user b in S_2
increase the influence weight of b to u with an average share of the accumulation of deviations and adjust the influence weight of u to b accordingly

In this paper, AIW directly convert the numerical value of the rating into the value of the increase or decrease on influence weight. Formally, the influence weights relating to u will be adjusted as follows:

$$IW_{a,u} = \begin{cases} IW_{a,u} - \left|r_{a,i} - r_{u,i}\right| & a \in S_1 \\ IW_{a,u} + \frac{\sum\limits_{b \in S_1} \left|r_{b,i} - r_{u,i}\right|}{|S_2|} & a \in S_2 \end{cases} \tag{1}$$

where $IW_{a,u}$ denotes the influence weight of a to u.

When predicting for the active user u on item i, AIW would firstly retrieve the IW table and select neighbors with the Top-N largest positive influence weights to u. If some selected credible neighbors have not rated on i, AIW would predict those missing data. The procedure of this smoothing is a little different: AIW would choose all the neighbors with a positive influence weight in order to alleviate the impact of the sparsity to the prediction quality of the missing data.

Both the prediction for missing data and the active user would use the weighted sum equation as follows:

$$P_{u,i} = \frac{\sum\limits_{a \in S(u)} (IW_{a,u} * r_{a,i})}{\sum\limits_{a \in S(u)} IW_{a,u}} \tag{2}$$

where $S_{(u)}$ contains the selected neighbors according to their influence weights, $P_{u,i}$ denotes the prediction for u on item i and the $IW_{a,u}$ denotes the influence weight of a to u.

After predicting all the items which the active user has not rated, AIW will select the ones with the biggest prediction values to generate a recommendation.

Table 1 MAE comparison with benchmarks under different sparsity

Num. of Training Users	100			200			300		
Ratings Given for Active users	10	20	30	10	20	30	10	20	30
AIW	0.823	0.801	0.787	0.808	0.783	0.764	0.795	0.774	0.753
EMDP	0.906	0.834	0.794	0.883	0.818	0.778	0.865	0.802	0.765
UBPCC	0.87	0.827	0.797	0.855	0.815	0.789	0.841	0.807	0.782
IBVS	0.887	0.838	0.805	0.863	0.816	0.792	0.849	0.805	0.783

5 Experiment and Evaluation

We use Movielens (http://www.grouplens.org/) dataset in our experiments. AIW here normalize the ratings involved by subtracting the mean value of all the ratings the same user has provided before adjusting influence weights and predicting for the active user. The normalization would convert the original discrete ratings to continuous values; consequently the ”equal rating” in the basic algorithm is changed into ”the abstract deviation of two ratings is less than 0.5”.

1. Prediction Quality

We follow the evaluation procedures described in [3] to test our algorithm and compare it with other state-of-the-art approaches: Effective Missing Data Prediction (EMDP) [3], standard user-based PCC(UBPCC) and item-based VS(IBVS). We vary the number of items in the profiles of the active users from 10, 20 to 30 and then predict for all the rest items they had rated in order to test the prediction quality under different data sparsity. In consideration of both accuracy and performance, we set the neighbor size of AIW to 20.

Table 1 show that AIW outperforms those benchmarks in various configurations.

2. Performance and Scalability

We first randomly select a number of users and all their ratings to form a sequence according to the timestamps of these ratings. Then we divide this sequence into a series of user sessions based on an assumption that every single session last no

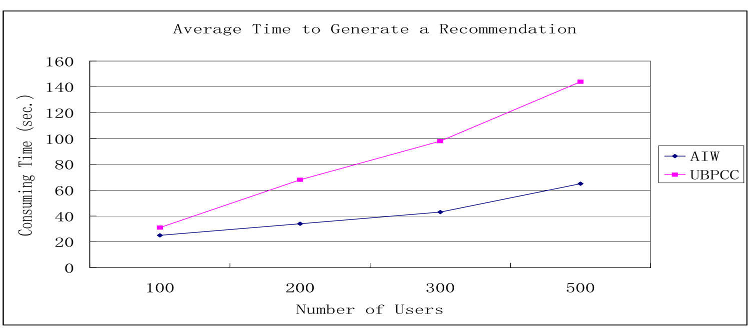

Fig. 1 Average recommendation time comparison

longer than 1 hour. And we assume that in each session the user would request only one recommendation, which means predicting for all the existing items this user has not rated. At last we apply the algorithms to deal with this interaction sequence and record the total consuming time. In this way, we could get an average consuming time for an algorithm to generate a recommendation. Then we gradually increase the number of the selected users to evaluate the scalability with different scales.

Fig.1 shows that the consuming time of AIW to generate a recommendation is less than UBPCC in all scales. In addition, the consuming time of AIW increases much more gently as the scale expands. This confirms that AIW is more scalable.

6 Conclusion

In this paper, we simulate the ”word-of-mouth” recommendation in a novel way. The online computation of similarity, which is the bottleneck of scalability in most existing approaches, is completed gradually in the form of accumulation of credibility. The algorithm designed accordingly achieves significant improvement in both aspects of accuracy and scalability simultaneously. It confirms this method is effective to resolve the conflict between improving prediction quality and scalability.

References

1. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
2. Herlocker, J.L., Konstan, J.A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 230–237 (1999)
3. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 39–46. ACM, New York (2007)
4. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp. 285–295 (2001)
5. Xue, G.R., Lin, C., Yang, Q., Xi, W.S., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 114–121 (2005)