ITEM-LEVEL TURST-BASED COLLABORATIVE FILTERING FOR RECOMMENDER SYSTEMS

Te- Min Chag
Department of Information Management,
National Sun Yat-sen University
temin@mail.nsysu.edu.tw

Wen- Feng Hsiao
Department of Information Management,
National PingTung Institute of Commerce
Wfhsiao@mail.npic.edu.tw

Chia-Ju Lu
Department of Information Management,
National Sun Yat-sen University

Abstract: With the dramatic growth of the Internet, it is much more convenient for users to acquire information than before. It is, however, relatively difficult to extract desired information through the huge information pool due to the information overload problem. Just like the situation that people rely on recommendation in their daily decision making process, recommender systems that filter unnecessary information are contributive to alleviating users' information processing loads. A popular recommendation approach is referred to as collaborative filtering (CF), which compares novel information with users' common interests shared by a group of people. This approach, however, suffers from the major problem of sparsity, i.e., the number of ratings obtained from users is usually small and the coverage of ratings appears to be very sparse. Several researches on CF variations have been proposed to tackle this problem. In this research, we developed an item-level trust-based CF approach (ITBCF) that incorporates item similarity into trust-based CF (TBCF) to improve the performance. Two experiments are conducted accordingly. We observe that the proposed ITBCF outperforms TBCF as shown from the experimental results. It therefore confirms our conjecture that the item-level trusts that consider item neighbors can stabilize derived trust values and achieve better performance. The feasibility of ITBCF approach is thus justified.
Keywords: recommender systems, collaborative filtering, trust-based CF, item similarity

1. INTRODUCTION

With the rapid growth of Internet, more and more information is disseminated in the World Wide Web. It is therefore not an easy task to acquire desired information from the Web environment due to the information overload problem. To overcome this difficulty, recommender systems that employ information filtering techniques emerge. The idea of recommender systems is simple: just like the situation that we rely on recommendation in our daily decision-making process, the recommender systems can recommend desired information through analysis of our past preferences and/or through other people's valuable suggestions.

It is therefore natural to classify information filtering techniques into content-based filtering and collaborative filtering. The former refers to comparing novel information with users' profile of past interests to make the recommendation decision while the latter refers to comparing novel information with common interests shared by a group of people (families, friends, neighbors).

In fact, collaborative filtering is more popular than content-based filtering because they do not suffer from the content limitation problem and the over-specification problem in content-based filtering. It, however, has its own major problem of sparsity. This problem refers to the situation that the number of ratings obtained from users is usually small and the coverage of ratings appears very sparse. With few data available, the user similarity employed in collaborative filtering becomes unstable and thus unreliable in the recommendation process.

Recently, several collaborative filtering variations emerge to tackle the sparsity problem. One of them is referred to as the trust-based collaborative filtering. In such an approach, a second component, trust, is taken into account in the recommendation process. It assumes that trust values among users can be derived either from a "web of trust" model or from the user-item rating matrix. Such data are then employed to predict recommendations. Since trust data are derived for all users, they are inherently denser than the user similarity data and thus lessen the sparsity problem.

Of the trust models that derive trust values from user-item rating matrix, two levels of trust can be obtained: the profile-level trust for a user and the item-level trust for a user toward a specific item. Due to its granularity characteristic, it is shown that item-level trusts have better recommendation results in the recommendation process. It is noted, however, that item-level trusts can be unstable and unreliable with fewer data considered in trust derivation. The purpose of this research is thus to propose a modified approach to compensating for such a weakness. In fact, in our opinion, item similarity can be incorporated into the trust value derivation because resulting item neighbors can provide valuable and robust information toward an item and stabilize the derived trust values.

The rest of this paper is organized as follows. In Section 2, the literatures relevant to collaborative filtering recommender systems and various collaborative filtering techniques are reviewed. Our proposed ITBCF approach is detailed in Section 3, followed by the experiment design and experiment results to justify its feasibility. Finally, conclusions and future work are described in Section 5.

2. LITERATURE REVIEW

2.1. Recommender systems

Recommender systems arise due to the emergence of the World Wide Web. The WWW channel provides an easy and inexpensive way to disseminate information. The amount of information distributed and acquired, however, is usually more than what our processing capability can handle. It therefore becomes a difficult task for us to distinguish what we desire from what we do not. Recommendation systems that serve as a tool to help users filtering unnecessary information and yield the right information they need becomes an essential research issue nowadays.

The growth of recommender systems also concerns e-commerce. Due to the popularity of doing business over the Internet, it is crucial for vendors to find consumers' preferences for products under e-commerce environments [0]. Schafer et al. [0] suggested that recommender systems could improve e-commerce sales in three ways: converting browsers into buyers, increasing cross-sell, and building loyalty. That is, recommender system can recommend products to visitors appropriately so that visitors may turn themselves into buyers to obtain what they need. Recommender systems can also recommend additional suitable products for a customer to increase the average order size. Finally, recommender systems can create a value-added relationship between the website and the customer so that the customer loyalty is enhanced.

Resnick [0] classified the recommender techniques into three categories: content-based filtering, collaborative filtering, and economic filtering. Content-based filtering compares an item's attributes with those extracted from users' past preferences, and makes recommendation based on their similarity. For example, a person who likes horror movies may be recommended to watch ghost movies, very similar to his/her previous preferences. Collaborative filtering, on the other hand, compares users' similarity and makes recommendation based on the group of users who share similar interests. For example, the person who likes horror movies may be recommended to try action movies if similar users like horror movies and action movies as well. Economic filtering recommends items based on cost-benefit evaluation of doing so. That is, if an item can be recommended to a lot of users, then it is beneficial to make such a recommendation. The optimal item recommendation to each user, however, may be sacrificed due to less personalization in this way.

2.2. Collaborative filtering techniques

Collaborative filtering is widely applied in recommender systems [0]. Traditional collaborative filtering approach is to employ user similarity to make recommendation for a given user. Namely, the recommendation is based on the interests among those users who share high similarity to that given user. In this kind of user-based collaborative filtering, user-user similarity is computed according to such similarity measures as Pearson correlation, cosine vector similarity, Spearman correlation, entropy-based uncertainty, and mean-squared difference [0]. After the similarity is calculated, the system can predict a rating for a given user using the following weighting formula:

$$P_{u,i} = \frac{\sum_{all\ \text{users},\ j} (sim(u,j) * R_{j,i})}{\sum_{all\ \text{users},\ j} (|\ sim(u,j)\ |)} \tag{1}$$

where $R_{j,i}$ is the rating of the i^{th} item given by the j^{th} user.

User-based collaborative filtering, however, suffers from the sparsity problem. We often observe that users only rate a low proportion of the total items. This phenomenon is called sparsity because the resultant user-item ratings matrix is far away from full. It causes an unreliable estimate of the similarity, and sometimes leads to low (prediction) coverage because there are no sufficient ratings to perform the prediction.

To lessen the sparsity problem, several variations are proposed in literature. Item-based collaborative filtering, for example, computes item-item similarity instead of user-user similarity. The item-item similarity is based on how often items are co-rated. It thus alleviates the sparsity problem because users hardly rate the same items, but items are much more easily co-rated. The resultant item-item similarity is more reliable and stable than user-user similarity, and the coverage of predicted ratings is thus much higher. The reliability characteristic can improve the recommendation quality in collaborative filtering.

On the other hand, trust-based collaborative filtering also arises. In this approach, a second component, trust, is employed in the recommendation process. Trust values of users are either explicitly given by users or derived from the user-item rating matrix. Such data can then replace or be combined with user-user similarities to make appropriate recommendations. It is obvious that trust-based collaborative filtering can lessen the sparsity problem in that trust data are relatively dense compared with the user-user similarity data.

Massa and Bhattacharjee [0] developed a trust-aware system that considered a “web of trust” to alleviate problems of sparsity, cold start, and vulnerability to attacks. The web of trust was constructed based on explicitly specified trust values provided by users (e.g. data from epinion.com website) and their propagations using trust inferences. As stated, it reduced data sparsity due to denser trust values in use. In addition, the trust inference property eased the cold start user problem. Finally, shilling attack problem was naturally relieved since trust values were employed to facilitate the recommendation process. Those users who intentionally used false ratings can be ruled out with low trust values given or derived.

Papagelis et al. [0] proposed a trust inference mechanism to alleviate the sparsity and cold-start problems. They viewed the users associated with “co-rated item” similarity as a social network. Starting with the given user-item rating matrix, similarity degree between two users with direct link (i.e. they had co-rated items) was assigned to be their trust value. For those users who were not directly linked, their trust values could be inferred through trust propagation. In addition, confidence and uncertainty (the complement of confidence) properties were defined along the paths between any two users. Finally, the trust value between two users was inferred either by composition (aggregation) of trusts along their multiple paths, or by selection of the trust in the most confident path among all paths. The predicted rating is therefore based on the trust similarity.

Hwang and Chen [0] presented an effective recommender system that included trust computation module, similarity computation module, and rating prediction module. In the trust computation module, two levels of trust were computed: the global trust metric and the local trust metric. The global metric measured a user’s overall trustworthiness while the local metric calculated a user’s trustworthiness with respect to the active user. The local trust metric was derived first with a simplified Resnick’s prediction rating toward a specific item and then taking the average of the complement of difference between the predicted ratings and the actual ratings provided by the active user toward all co-rated items. A user’s global trust was computed by taking the average of the local trust values from his/her neighbors who were directly connected to the user. At last, trust values could be propagated and further composed (aggregated) if multiple paths existed. The similarity computation module employed Pearson correlation coefficient as the similarity measure. Finally, the rating prediction module used standard Resnick’s prediction formula to yield the predicted ratings. The weights in the formula could be similarity degrees, local trust values, or the global trust values, respectively.

Unlike the above research works that derived trust values between users and utilized them to replace user similarity in the prediction formula, O’Donovan and Smyth [0] proposed to measure trust values of individual users and incorporate them into the user similarity computation. The trust value of an individual was calculated based on consideration of all his/her rated items and all users who co-rated those items. Viewing a user as the producer (*p*) and all other users as the consumers I, they established *p*’s recommendation set to include all (*i*, *c*) pairs where item *i* was rated by *p* and co-rated by consumer *c*. In addition, the (*i*, *c*) pairs were called correct if the predicted rating of *i* considering *p* exclusively is close enough to the rating provided by *c* toward *i*. The profile-level trust for a producer *p* was therefore defined as the proportion of correct elements within this set. In addition, an item-level trust for a producer *p* toward an item could also be defined by the proportion of correct elements toward this item within this set. Finally, such (profile-level or item-level) trust values were incorporated into the user similarity computation as weighting or filtering to yield the predicted ratings.

3. PROPOSED APPROACH

The objective of this research is to propose an approach to improving performance of trust-based collaborative filtering. According to the literature, O'Donovan and Smyth [0] presented a trust-based filtering approach to deriving the trust value toward a user by considering all his/her rated items and all users who co-rated those items. This approach differs from other trust-based approaches mainly in that the trust values are derived for each user rather than between users. Without considering trust between users, this approach is unnecessary to propagate trusts (trust inferences) by transitivity or compose trusts when multiple paths exist between two users. It therefore not only simp ids an essential issue whether trust inference that bases itself on trust transitivity[1] between these properties, their work seems promising in recommender applications.

In addition, according to their research, two levels of trust can be obtained, i.e., the profile-level trust for a user and the item-level trust for a user toward a specific item. In thei st results in better recommendation due to its granularity characteristic. We, however, not ng item-level trusts than profile-level trusts, which may cause the results unstable and unr mation to stabilize derived trust values seems essential in this kind of collaborative filtering approaches.

Based on the above statements, we therefore propose an approach, called ITBCF (item-level trust-based collaborative filtering), that incorporates item similarity into the ite n that similar items (neighbors) can provide valuable and robust information toward an ite process of ITBCF is shown in Figure 1.

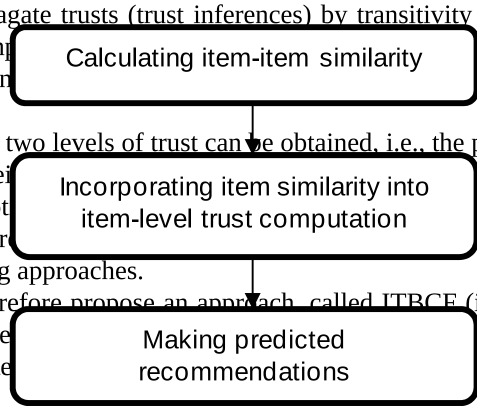


Figure 1 Process of ITBCF

3.1. Calculating item-item similarity

First, we compute item-item similarity based on the Pearson correlation, which is calculated as follow:

$$sim(i, j) = \frac{\sum(R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum(R_{u,i} - \overline{R}_i)^2}\sqrt{\sum(R_{u,j} - \overline{R}_j)^2}} \tag{2}$$

where $\overline{R}_i$ ($\overline{R}_j$) is the average of the i^{th} (j^{th}) item's ratings. Note that the item-level trust will only be extended with similar items (neighbors). Therefore, a predetermined threshold is set for the similarity degree to filter distinct items.

3.2. Incorporating item similarity into item-level trust computation

The second step is to compute each user's trust value on the item level. Let an active user be the one whom we would like to make recommendations for. Viewing the active user as the producer (p) and all other users as the consumers (c), we first define correct

---

[1] That is, if A trusts B and B trusts C, then A trusts C with some mathematical inference mechanism.

predicted score if the predicted rating[2] solely determined by *p* toward an item *i* is close enough to the rating provided by *c* toward *i*, which is expressed as follows.

$$\text{Correct}(i, p, c) \Leftrightarrow |\hat{p}(i) - c(i)| < \varepsilon \tag{3}$$

where ε is the threshold parameter to determine correctness.

For each *p*, we then construct a recommendation set (RecSet (*p*)) whose elements are pairs of (*i*, *c*) where *i* is rated by *p*, and co-rated by *c*. Within this set, we therefore can define a subset (CorrectSet(*p*)) whose elements satisfy correct conditions as above. The number of CorrectSet(*p*) elements divided by the number of RecSet(*p*) is the profile-level trust for *p*.

To further obtain the item-level trust for *p* toward an item *i*, we can simply extract from CorrectSet(*p*) and RecSet(*p*) elements that are relevant with item *i*. The item-level trust formula is defined as follows:

$$Trust^I(p,i) = \frac{|\{(c_j, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_j, i_k) \in RecSet(p) : i_k = i\}|} \tag{4}$$

In our study, however, we would like to extend this formula with similar items (neighbors) as explained before. Then equation (4) is thus modified as

$$Trust^I(p,i) = \frac{|\{(c_j, i_k) \in CorrectSet(p) : i_k = i \text{ and its neighbors}\}|}{|\{(c_j, i_k) \in RecSet(p) : i_k = i \text{ and its neighbors}\}|} \tag{5}$$

where neighbors of the item *i* are obtained from the first step.

3.3. Making predicted recommendations

In the final step, we employ Resnick's prediction formula to predict the ratings for a given user. We have three methods to incorporate trust value into prediction computation: filtering, weighting, and combining.

For the filtering method, we will filter out the user's recommendation from prediction computation if $Trust^I(p, i)$ is below a predetermined threshold. The prediction formula is as follow:

$$c(i) = \bar{c} + \frac{\sum_{p \in P^T(i)} (p(i) - \bar{p})\, sim(c, p)}{\sum_{p \in P^T(i)} |sim(c, p)|} \tag{6}$$

where $P^T(i)$ is the set of users with item level trust on item *i* over the threshold T, as expressed by

$$P^T(i) = \{p \in P(i) : Trust^I(p, i) > T\} \tag{7}$$

and *P*(*i*) is the user set who has rated item *i*.

For the weighting method, every user's recommendation is adjusted with a weight that is combines user-user similarity and item-level trust derived above. We adopt the harmonic mean of these two figures because inherently the harmonic mean is high only if both values are high, and low if any one of them is low. The weight formula is shown as follows.

$$w(c, p, i) = \frac{2(sim(c, p))(trust(p, i))}{|sim(c, p)| + trust(p, i)} \tag{8}$$

The prediction formula is as follows.

$$c(i) = \bar{c} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p})\, w(c, p, i)}{\sum_{p \in P(i)} |w(c, p, i)|} \tag{9}$$

Finally, for the combination method, we combine the above filtering and weighting schemes. A user's recommendation will be filtered out if his trust value toward the item is too low. Once accepted, his/her trust value will be adjusted with the user similarity as the harmonic mean. The prediction formula is then as follows.

$$c(i) = \bar{c} + \frac{\sum_{p \in P^T(i)} (p(i) - \bar{p})\, w(c, p, i)}{\sum_{p \in P^T(i)} |w(c, p, i)|} \tag{10}$$

where $P^T(i)$ is from equation (7) and *w*(*c, p, i*) from equation (8).

---

[2] This predicted rating $\hat{p}$ (i) from the Resnick's formula considering p exclusively is $\bar{p}$.

4. EXPERIMENTS AND RESULTS

In this section, we conduct two experiments to examine our proposed approach, ITBCF. Specifically, we compare performance of ITBCF with that of trust-based collaborative filtering (TBCF) as proposed by O'Donovan and Smyth [0]. Since three different methods (filtering, weighting and combination) can be utilized in the final predicted recommendation step, they are called ITBCF_FT, ITBCF_WT and ITBCF_CB, respectively; and their counterparts of TBCF are called TBCF_FT, TBCF_WT and TBCF_CB, respectively.

To evaluate the performance, we first describe the experimental design that includes data descriptions, performance measures, evaluation scheme, and parameter settings as follows.

4.1. Setup

4.1.1. Data Collection

In our experiments, we employ data from MovieLens (http://movielens.umn.edu/), as commonly adopted in collaborative filtering researches [0, 0]. MovieLens is a web-based recommender system that is provided by GroupLens Research at the University of Minnesota since 1997. Users visit MovieLens to rate and receive recommendations for movies.

The website offers a dataset that consists of 100,000 ratings by 943 users on 1682 movies collected from September, 1997 through April, 1998. All ratings are in a five-star scale (ratings 1-5) and each user has made at least twenty movies. The sparsity level of this dataset (called Data LS) is 0.9369 ($1 - 100{,}000 / (943 \times 1682)$)[3]. It is defined as 1 - density. Furthermore, to manipulate different degrees of sparsity, we extract data from Data LS to form several subsets that are relatively small in scale. Statistics of the extracted datasets are summarized in Table 1. It is obvious that sparsity degree decreases from Data 1 to Data 7 with the highest of 94.22% and the lowest of 82.38% (still sparse though).

	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
No. of Users	207	221	231	217	191	178	165
No. of Movies	269	284	289	416	180	178	175
No. of Rating 1	154	236	296	291	156	195	377
No. of Rating 2	336	493	771	718	453	441	576
No. of Rating 3	870	1205	1831	1783	1221	1117	1470
No. of Rating 4	1142	1745	2395	2159	1735	1726	1671
No. of Rating 5	714	1123	1370	1234	1172	1123	991
Sparsity degrees	0.94	0.92	0.90	0.88	0.86	0.85	0.82

Table 1: Statistics of the small datasets

4.1.2. Performance Measure

In our study, we adopt the commonly used measure in recommender systems: mean absolute error (MAE). MAE is one of the statistical accuracy metrics that compare prediction results (numerical recommendation scores) with actual outcomes (user ratings). It measures the average magnitude of the errors of the recommendation results without considering the direction (the sign), which is defined as follows:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N} \tag{11}$$

---

[3] The formula for sparsity calculation is 1– no. of ratings / (no. of users × no. of items).

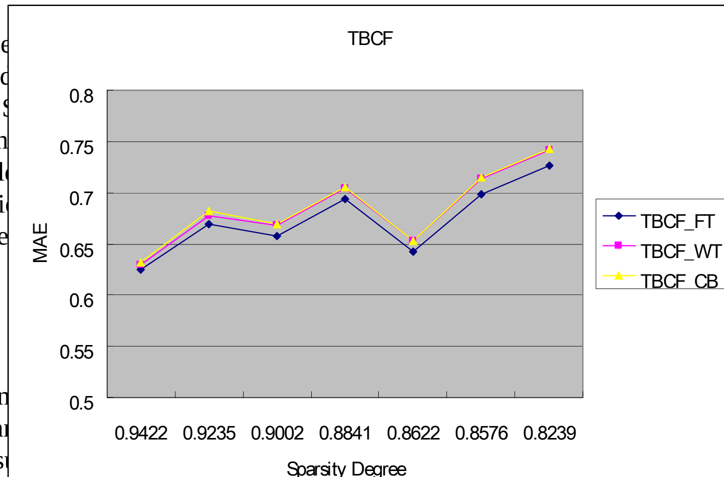
where p_i is the predicted rating, q_i is the actual rating and N represents the total number of predicted recommendations. The lower the MAE, the better performance the recommender system achieves.

4.1.3. Evaluation Scheme

The evaluation scheme we adopt in this study is the hold-out validation. Hold-out validation randomly splits the data into a training set and a test set according to a certain percentage. Training data are used exclusively in the recommendation process to predict whether a novel item should be recommended to a user, while test data are used to examine the recommendation performance by matching the predicted and the actual result. For small datasets (Data 1 to Data 7), the ratio of training data to test data is 9:1 while for Data LS, the ratio is 8:2.

4.1.4. Parameter Settings

Finally, three parameters ne we consider the parameter ε that determines whether the predicted st values. We set it to be 1.8, as pointed out by O' Donovan and S ems (see equation (2)) in ITBCF. We set it to be 0.8 since we do n trust value compositions. Finally, we need to specify the threshol ommendations with filtering and combination methods (see equati ut users with extremely low trust values and allow most users to be



4.2. Experiment I

The objective of Experimen ferent predicted recommendation methods (weighting, filtering, a 7). We first consider the TBCF approach and its performance resu

We observe that the MAI parsity, which is an interesting phenomenon. Recall that TBCF ust value of a user is derived by considering (*item*, *user*) pairs wh to the one who co-rates the item. With data relatively less sparse, overestimated due to the large ε (=1.8) setting. With overestimat s or wrong weights in predicted recommendation calculations, an

Furthermore, we notice that that we set the trust threshold to be 0.1 and thus few users are fi on the user similarity or weights derived. As stated, the item-level in trust derivation, and thus may easily bias the weights. Therefore We then turn to the ITBCF appro

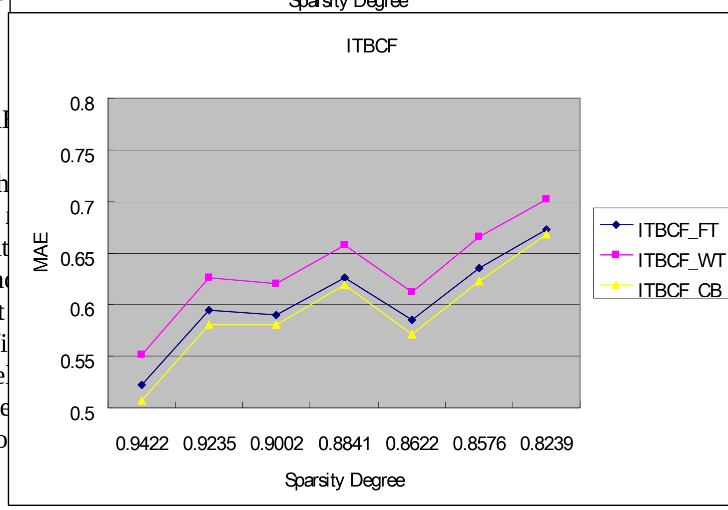


Figure 3: Performance result for ITBCF

From Figure 3, we again observe that the MAE performance has an increasing trend with decreasing data sparsity. The same explanation on TBCF can be also applied here. However, the second phenomenon differs significantly: ITBCF_CB outperforms ITBCF_WT and ITBCF_FT. This is because in ITBCF, the item-level trust values consider neighbors (similar items) to improve the stability and reliability of the trust values. With stable and reliable trust values obtained, we may easily incorporate right users or accurate weights in predicted recommendation calculations, and thus results in performance improvement in both the weighing

---

4 This is different from what was reported in O'Donovan and Smyth [0] where TBCF_CB always performed best. However, if weights are biased due to unreliable trust values, then filtering scheme that employs user-user similarity exclusively could be better.

method and the filtering method. T ... ion performs best among the three methods.
To further compare performance be ... under each method and draw them into Figure 4.

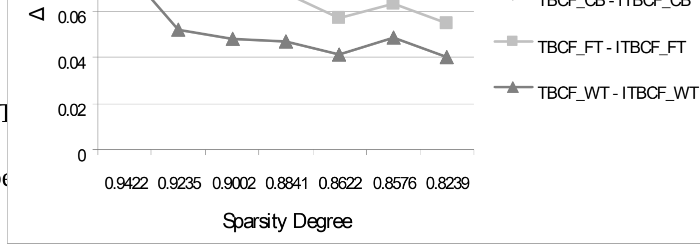


Figure 4: Performance comparison between TBCF and ITBCF

From Figure 4, the first significant observation is that results from our proposed ITBCF are better than those counterparts from TBCF under different sparsity degrees. It confirms our conjecture that the item-level trusts without considering neighbors (similar items) may easily cause the predicted recommendation results more unstable and unreliable due to fewer data utilized in trust derivation. On the other hand, item-level trusts that consider neighbors can compensate such a drawback since they provide more robust information toward the trust determination.

Second, the performance differences exhibit a decreasing trend with decreasing data sparsity. This implies that the item-level trust values that consider neighbors improve the performance significantly with data of high sparsity. The reason lies in that with sparser data, item-level trust values ... nreliable due to fewer data utilized in trust derivation. And our propo ... ty of derived trust values and thus enlarge the performance difference ... ITBCF_CB is the most significant simply because performance from ... F_CB is the best in ITBCF so that they have the largest difference gap

4.3. Experiment II

The objective of Experiment I ... he large-scale dataset, Data LS. As stated, the sparsity level of Data L ... rge as those small datasets used in Experiment I. The performance resu

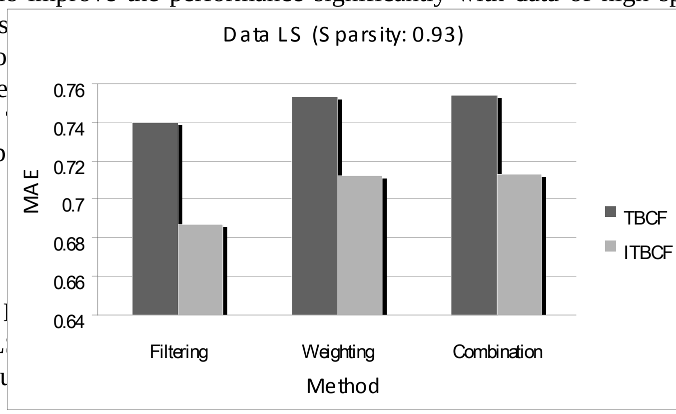


Figure 5: Performance comparison under Data LS

As can be seen from Figure 5, the results do not differ much from those observed in Experiment I. In general, ITBCF outperforms TBCF no matter what methods utilized in predicted recommendation calculations. It further justifies the feasibility of ITBCF in real applications.

Regarding TBCF, once again we observe that TBCF_FT performs slightly better than TBCF_WT and TBCF_CB. This is because the item-level trust values tend to be unreliable and thus bias the weights. As a result, the filtering method performs best among the three methods.

More interestingly, ITBCF_FT also outperforms ITBCF_WT and ITBCF_CB in this case. We first notice that ITBCF that considers neighbors in item-level trust value does improve the prediction performance under all three different methods. With comparably reliable trust values, we can easily incorporate right users or accurate weights in predicted recommendation calculations. On the other hand, however, we recall that the number of ratings is 100,000, which is approximately 15 to 30 times as many as those in small datasets, despite its sparsity. With more ratings, we need to consider more pairs of (*item*, *user*) in the recommendation set (RecSet) and we can easily obtain more elements in the correct set (CorrectSet) due to the large ε (=1.8) setting. The derived item-level trust values therefore tend to be overestimated (See equation (7) in Section 3). The interactive effect of more data considered in RecSet results in weight bias in the weighting method and combination method, and therefore ITBCF_FT performs best among the three methods.

5. CONCLUSIONS

Due to the dramatic growth of the Internet, more and more people rely on the online channel to share and disseminate information. It is, however, not an easy task to acquire desired information from the Web environment due to the information overload problem. To overcome this difficulty, recommender systems emerge to recommend desired information for users and facilitate their decision-making process.

In literature, many researchers employ collaborative filtering in recommender systems. Collaborative filtering, however, suffers from the sparsity problem, i.e., the number of ratings obtained from users is usually small and the coverage of ratings appears very sparse. With few data available, the user similarity employed in collaborative filtering becomes unstable and unreliable in the recommendation process.

In this research, we propose the item-level trust-based collaborative filtering (ITBCF) approach to alleviate the sparsity problem. ITBCF utilizes the user ratings in the trust derivation process, just like TBCF as proposed by O'Donovan and Smyth [0]. Unlike TBCF, however, ITBCF incorporates item similarity into the item-level trust calculations to further enhance the stability and reliability of derived trust values.

Two experiments are conducted accordingly to examine the performance of ITBCF. Data from MovieLens are employed because of their common citations in collaborative filtering researches for evaluation purpose. The first experiment is to compare TBCF and ITBCF under different predicted recommendation methods and different sparsity degrees. We observe that ITBCF outperforms TBCF in every situation we consider. It therefore confirms our conjecture that the item-level trusts that consider neighbors can stabilize derived trust values, and thus improve the performance. Experiment II is to compare TBCF and ITBCF using the large-scale data. Similar results are obtained and further justify the feasibility of our proposed approach in real applications.

Even though our research results seem promising, there are some issues worthy of further exploration. First, in our experiments, only MovieLens data are applied. To further enhance the generalization of the experimental results, more different kinds of data should be employed to evaluate the performance of ITBCF.

Second, regarding ITBCF itself, we know it takes time because substantial user ratings are considered to derive users' trust values. This gives rise to an interesting research issue of whether it is possible to adaptively update the trust values without repeating the whole trust value calculation process every time the rating matrix is changed. It serves as another potential work we may consider in the future.

6. REFERENCES

Candillier, L., Meyer, F., & Boulle, M. (2007). Comparing State-of-the-Art Collaborative Filtering Systems. *In proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition.* Leipzig, Germany.

Goldberg, D., Nichols, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12) pp.61-70.

Hwang, C.-S. and Chen, Y.-P. (2007). Using Trust in Collaborative Filtering Recommendation. In Proceedings of 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.

Massa, P. and Bhattacharjee, B. (2004). Using Trust in Recommender Systems: An Experimental Analysis. In Proceedings of 2nd International Conference on Trust Managment. Oxford, England.

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In proceedings of the 10th international conference on Intelligent user interfaces , pp. 167-174.

Papagelis, M., Plexousakis, D., & Kutsuras, T. (2005). Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences . In proceedings of the 3rd International Conference on Trust Management , pp. 224-239.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In CSCW '94: Conference on Computer Supported Coorperative Work (Chapel Hill, 1994), ACM, pp. 175-186.

Rucker, J. and Polanco, M.J. (1997). Siteseer: personalized navigation for the Web. Communications of the ACM ,40(3) pp.73-76.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender systems--a case study. In proceedings of ACM WebKDD Workshop.

Schafer, J., Konstan, J. A., & Riedl, J. (2001). E-Commerce Recommendation Applications. Data Min.Knowl.Discov. , pp. 115-153.

Yu, K., Wen, Z., Ester, M., & Xu, X. (2001). Feature Weighting and Instance Selection for Collaborative Filtering. dexa (p. 285). IEEE Computer Society.