Final Report

April 22, 2011

1 Introduction

Recommender System is becoming a crucial part of our online experience these days. It is widely used in various websites to recommend all kinds of products to customers: movies[1], books[2], music[3], etc. The quality of a recommender system is not only related to user experience, but also revenues of online business.

Different recommendation methods have been proposed in both industry and academia. The performance of recommender systems is usually measured with two metrics: accuracy and coverage. Higher accuracy means the estimated rating of one item for a given user is more likely to match the user's own rating of that item; higher coverage means the system is able to predict more items' ratings for more users.

One of the most popular recommendation method is Collaborative Filtering (CF) [1]. CF compares rating profiles of all items and identify *similar items* using Pearson Correlation. When asked to predict the rating of a target item for a target user, CF first finds items which are similar to the target item, then it returns the target user's average ratings of these similar items (weighted by their similarities to the target item). Though CF has been adopted by many recommender systems, it performs poorly for cold-start users (i.e. new users of a website). Since cold-start users have rated only a few items, it is often difficult for CF to find similar items rated by the target user. Therefore, coverage of CF is usually very low for cold-start users.

To overcome the weakness of CF on cold-start users, Mohsen Jamali and Martin

[1]www.flixster.com
[2]www.amazon.com
[3]www.pandora.com

Ester proposed Trustwalker [2]. Trustwalker combines trust-based method and CF by performing a random walk on the target user's neighbours (both direct and indirect). Trustwalker managed to achieve lower Root Mean Square Error (RMSE) and higher coverage compared to traditional CF.

To explore potential improvement in these recommender systems mentioned above, we propose a new recommendation method, which utilizes users' expertise in different product categories. The rationale behind this method is that experts' reviews tend to be more insightful and cover more items. Besides, according to consumer psychology, consumers tend to trust more on experts when they are purchasing more expensive products, such as cars, electronics, etc [3]. We first calculate all users' expertise based on the number and qualities of their reviews, then we pick users with the highest expertise and call them *experts*. Lastly, we use experts' ratings to predict the ratings of other users.

The rest of the report is organized as follows: section 2 introduces the methodology of our experiments; section 3 reports the dataset we crawled from Epinions.com and our experiment results on that dataset; section 4 concludes the project.

2 Methodology

Methodology.

2.1 Identifying Experts

Given a category C and a set of users $U = \{u_1, ..., u_n\}$ who have written at least one review in category C, we primarily consider two factors in calculating a user's expertise: (1) the average quality of the user's reviews, and (2) the total number of user's reviews. We use the following formulas to calculate a user's expertise in a category C:

$$E(u) = \frac{\sum_{i=1}^{n} R_{u,i}}{n} \times f(n) \tag{1}$$

$$f(n) = \frac{1}{1 + e^{\frac{-10n}{MAX\_REVIEWS(C)}}} \tag{2}$$

where $R_{u,i}$ is the rating for user u's review i in category C and $f(n)$ is a sigmoid function to avoid favoring the number of reviews too much. It is based on the assumption that a user written 100 reviews has more expertise than a user written 10 while a user written 1000 reviews is almost as good as a user written 900 reviews. $MAX\_REVIEWS(C)$ is the maximum number of reviews per user in category C. $f(n)$ is tailored for every category, i.e., the user who has written the most reviews in a category have $f(n) = 1$.

2.2 Top k Experts Recommendation

Previous research has demonstrated that recommending items to users based on *expert* opinions can be as good as traditional user-based *Collaborative Filtering* [4]. We propose a similar expert recommendation algorithm: given an item i that belongs to a category C, the recommender first picks out the top k percent experts in category C; then for each expert, if the expert has rated item i, the exact rating is returned; otherwise a weighted average of similar items of i is returned; finally the recommender averages the ratings of all selected experts by their expertise as the predicted rating for item i. Item similarity is calculated using *Pearson Correlation*:

$$corr(i,j) = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_u)} \sqrt{\sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_u)}} \tag{3}$$

where $U_{i,j}$ is the set of common users who have rated both items i and j, and $\bar{r}_u$ denotes the average of ratings expressed by u.

3 Experiment

We tested different recommender systems on a comprehensive dataset from Epinions.com, which is a product reviewing site for consumers. This section will report our findings in the experiments.

3.1 Dataset

Epinions.com is a website where users can rate products and write reviews. Its product catalogue covers a large number of product categories, including Music, Books, Movies, Automobiles, Electronics, etc. For each product, consumers provide ratings on a scale from 1 to 5 (5 being the best), write detailed reviews on products

Data Type	Size
Users	91,584
Reviews	605,066
Trust Links	1,152,095

(a) Dataset Size

Rating	Percentage
5	33%
4	27%
3	20%
2	13%
1	7%

(b) Rating Distribution

Review Helpfulness	Percentage
Very Helpful	74.3%
Helpful	15.4%
Somewhat Helpful	6.8%
Show	2.4%
Not Yet Rated	1%

(c) Review Helpfulness Distribution

Table 1: Dataset Summary

and express the helpfulness of existing reviews. Besides, users can express directed trust among each other: if user A trusts user B, user B does not have to trust user A. In other words, there is a implicit directed trust graph among all users.

We crawled a comprehensive dataset from Epinions.com, which includes user info, product reviews, review helpfulness, trust links, etc. The size of the crawled dataset is shown in Table 1a. Among all the 91,584 users, about 1000 users have written more than 100 reviews. These top 1000 most productive users have contributed to 55.9% reviews (338,173 out of 605,066). Table 1b shows the distribution of ratings among all products. One third of the ratings are 5, and about 60% of the ratings are 4 or 5. The distribution is very different from another popular dataset – MovieLens, whose rating distribution is more diverse[1]. Table 1c shows the helpfulnesses of all reviews. Similarly, about 75% of all reviews are considered *very helpful*.

Table 2 shows the number of reviews in the top 5 categories. The top 5 categories account for more than half of all reviews on Epinions.com.

Category	Number of Reviews	Percentage
Movies	102,461	16.9%
Books	72,940	12.1%
Music	63,424	10.5%
Hotels & Travel	32,348	5.3%
Electronics	31,853	5.3%

Table 2: Product Categories of Reviews

To test the performance of recommender systems, we split the dataset into a training

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

set and a testing set. Testing set is chosen randomly from all ratings and takes about 20% of all ratings; training set is the rest, which is about 80%. Then we used the training set to predict the ratings in the testing set. To make the prediction more challenging, we pick out the ratings for controversial items from the testing set and measure the performance separately for these items. We define *controversial items* to be the items whose ratings' Standard Deviations are greater than 1.5. These items' ratings are harder to predict, because users' opinions towards these items varies wildly from 1 to 5.

We used two most adopted metrics to measure the performance of recommender systems: Root Mean Square Error (RMSE) and coverage. In addition, we combined RMSE and coverage into a single metric called F Measure [2]. We define precision and F Measure similarly:

$$Precision = 1 - \frac{RMSE}{4} \tag{4}$$

$$FMeasure = \frac{2 \times Precision \times Coverage}{Precision + Coverage} \tag{5}$$

The next subsection reports the performance of different recommender systems in the Epinions.com dataset.

3.2 Experimental Results

Besides two *Top-k expert* methods, we also implemented item-based CF and Trust-walker to compare the performance. The four methods in our experiments are:

1. *itemCF*. This is the traditional CF method, which returns the weighted average ratings of similar items from target user's rating profile.
2. *trustwalker*. This is the random walk model proposed by Mohsen Jamali and Martin Ester [2]. We set the *max_depth* to 3 and *epsilon* to 0.001 in their *Trustwalker* model.
3. *expert*. This method finds the top-k experts in each category, then calculates weighted average of experts' ratings for target item and also similar items. The estimated rating is adjusted according to target user's rating scale.

4. *expert-simple.* This is the simpler version of *expert* method. It finds the top-k experts in each category, and returns the experts' weighted average ratings for the exact target item. The estimated rating is adjusted according to target user's rating scale.

Figure 1a and 1b shows the RMSE of the four methods over the entire testing set. The horizontal axis represents the percentage of users that we call experts. If the percentage is k%, it means that we take the top k% of users with the highest expertise in each category. Since y axis is only related to *expert-simple* and *expert* method, the performance of *itemCF* and *trustwalker* remains the same.

As shown in Figure 1a, *expert-simple* and *expert* almost perform better than *itemCF* and *trustwalker.* This result is very surprising, since *expert-simple* and *expert* are relatively simple and naive compared to *itemCF* and *trustwalker.* Besides, though the estimated ratings in *expert-simple* and *expert* are not personalized, the RMSE over all users are quite good.

Figure 1b shows the RMSE of the four methods over cold-start users in the testing set. For cold-start users, *itemCF* performs poorly, since it could not find many rated similar items from the target user's rating profile. *trustwalker* performs better than both *expert-simple* and *expert.* This result is expected, since *trustwalker* was designed to mitigate the cold-start user's problem.

One of the other most widely used metrics is coverage. Figure 2a and 2b shows the precision, coverage and F measure of all four methods over all users and cold-start users.

4 Conclusion

Conclusion.

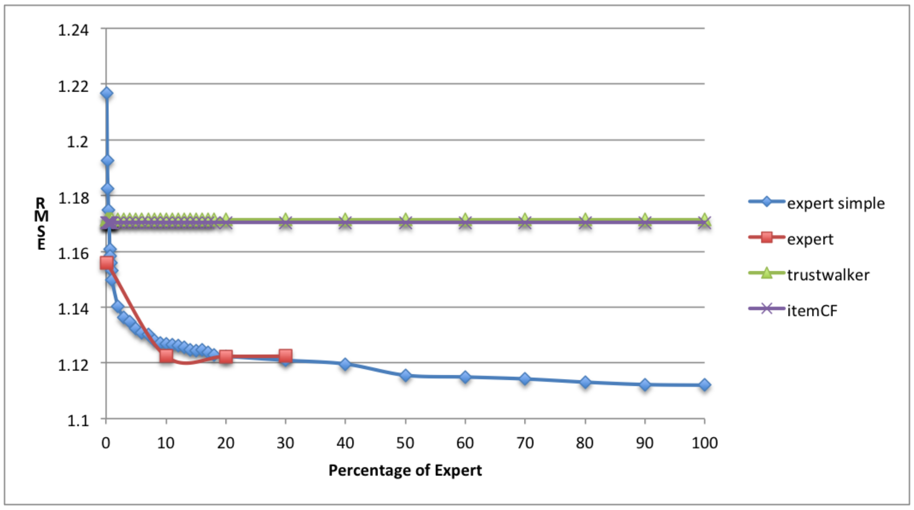
References

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

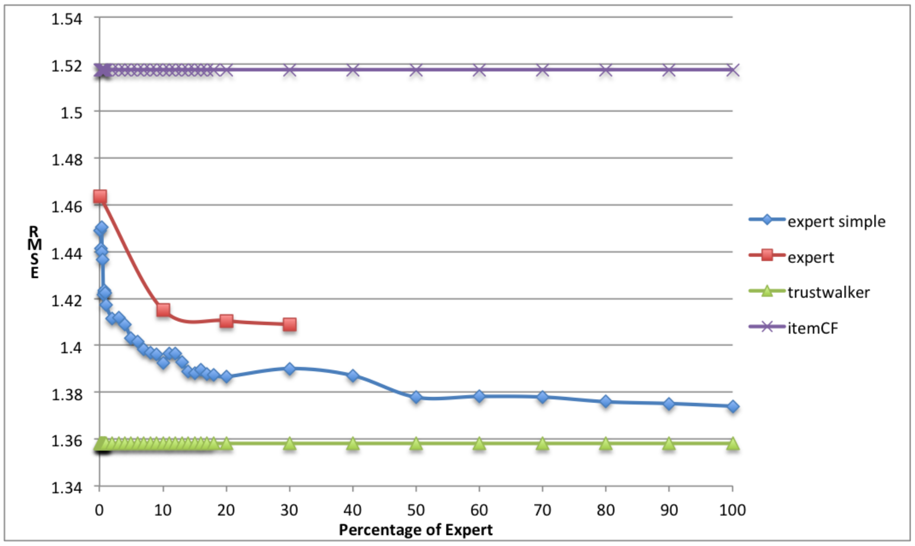
[2] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 397–406, 2009.

[3] D. F. Duhan, S. D. Johnson, J. B. Wilcox, and G. D. Harrell, "Influences on consumer use of word-of-mouth recommendation sources," *J. Academy of Marketing Science*, pp. 283–295, 1997.

[4] X. Amatriain, N. Lathia, J. Pujol, H. Kwak, and N. Oliver, "The wisdom of the few: a collaborative filtering approach based on expert opinions from the web," *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 532–539, 2009.

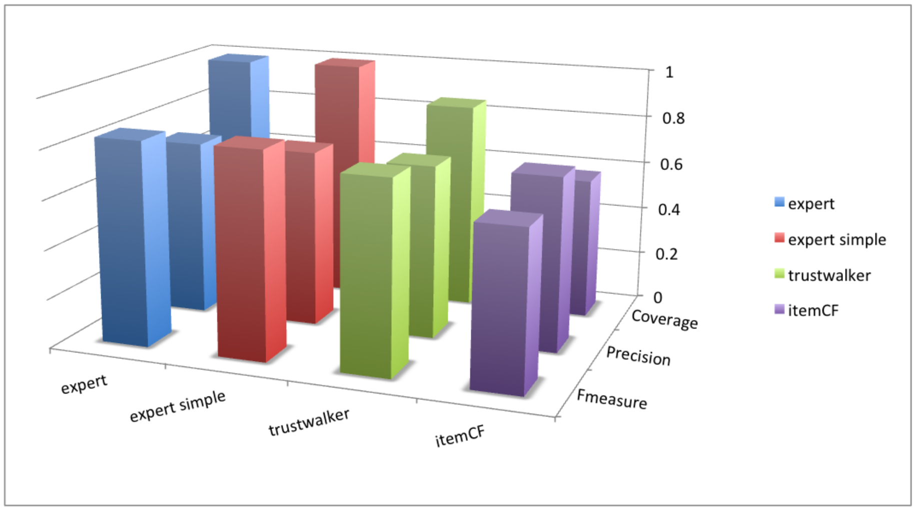


(a) RMSE for all users

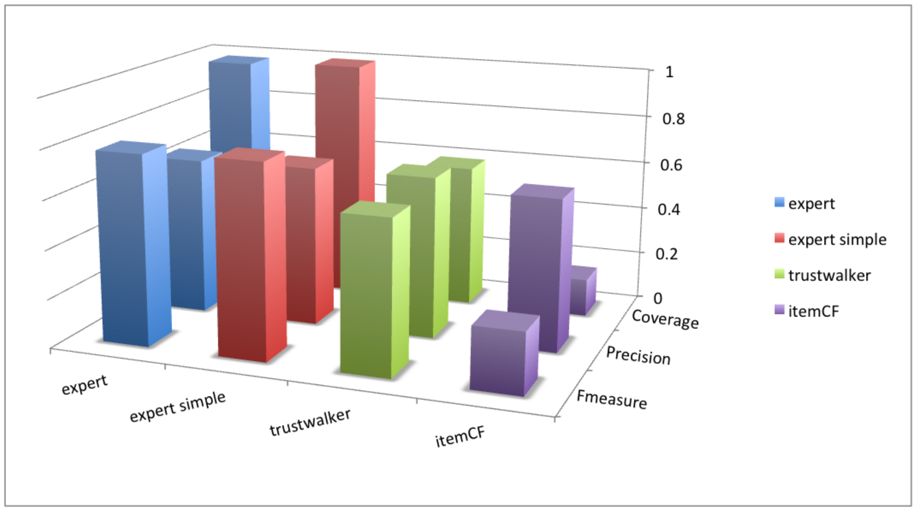


(b) RMSE for cold-start users

Figure 1: RMSE



(a) Results for all users



(b) Results for cold-start users

Figure 2: Results