Analyzing Recommender System's Performance Fluctuations across Users

Charif Haydar, Azim Roussanaly, Anne Boyer

Analyzing recommender system's performance fluctuations across users.

Charif Haydar[1], Azim Roussanaly[1], and Anne Boyer

Université de Lorraine, Laboratoire Loria, Bâtiment C, Equipe KIWI
615, rue de jardin botanique
54600 Vandœuvres-lès-Nancy, France
{charif.alchiekhhaydar,azim.roussanaly,anne.boyer}@loria.fr,
Home page: http://kiwi.loria.fr

Abstract. Recommender systems (RS) are designed to assist users by recommending them items they should appreciate. User based RS exploit users behavior to generate recommendations. As a matter of fact, RS performance fluctuates across users. We are interested in analyzing the characteristics and behavior that make a user receives more accurate/inaccurate recommendations than another.
We use a hybrid model of collaborative filtering and trust-aware recommenders. This model exploits user's preferences (represented by both item ratings and trusting other users) to generate its recommendations. Intuitively, the performance of this model is influenced by the number of preferences the user expresses. In this work we focus on other characteristics of user's preferences than the number. Concerning item ratings, we touch on the rated items popularity, and the difference between the attributed rate and the item's average rate. Concerning trust relationships, we touch on the reputation of the trusted users.

Key words: Recommender system, collaborative filtering, trust-aware, trust, reputation

1 INTRODUCTION

Recommender systems (RS) [5] aim to recommend to users some items they should appreciate, over a list of items. RS exploits the user's ratings of items, and/or his explicit/implicit relationships with other users, to generate recommendations to him. Intuitively, the more the user is connected to other users and items, the better the quality of recommendation is. In this paper, we treat the question of RS performance from two different points of view. The first is the structural point of view, where we try to improve the RS performance by hybridizing two recommendation approaches. The second is the user's behavior point of view, where we study the impact of several characteristics of user behavior on the system performance.

We use the epinion.com[1] dataset. epinion.com is a consumers opinion website where users can rate items in a range of 1 to 5, and write reviews about them. Users can also express their trust towards reviewers whose reviews seem to be interesting to them.

In [9], two recommendation approaches have been tested on this dataset separately: collaborative filtering (CF) [11] and trust-aware [9, 12]. CF relies on user-item ratings to compute similarity between users, whereas trust-aware replaces this similarity by explicit trust relationships between users. Trust-aware performance surpasses that of CF, but CF is still better for some categories of users. In a previous work [27], we applied several hybridization strategies of both CF and trust-aware recommenders on this dataset. We found that hybrid models can cover a larger set of users.

In this paper, we focus on the recommendation accuracy by user. We consider its fluctuations across users as a result of user's ratings and trusting strategies. We touch on the following questions: Which type of items should user rate in order to assist the system to satisfy him? What if the user rates frequently opposite to the orientation of the community? Is trusting more users always beneficial to the user? Is there a link between the reputation of the users that a user trust and the quality of the recommendations he receives?

The outline of the paper is organized as follows: in section 2 we discuss recommendation approaches. In section 3 we explain the details of the used dataset, the context of the experiments, and both structure and user strategies based performance evaluations. Finally, the last section is dedicated to conclusion and future works.

2 STATE OF ART

Diverse techniques were used to build recommender systems. Our current explanation is restricted to the needs of our recommendation model. We employ a hybrid RS [1] composed of Collaborative filtering (CF) [11] and trust-aware recommenders [9, 12]. In the following subsections we explain both approaches and the chosen hybridization strategy.

2.1 Collaborative filtering recommenders

CF is the most popular recommendation approach. The prediction function in CF (which is the key element of any RS) is based on the similarity of users' preferences (usually expressed by rating items). Users' ratings are stored in a rating matrix, which is a $m \times n$ matrix, where m is the number of users, and n is the number of items. An element $v_{u_a i}$ of this matrix represents the rating given by the user u_a to the item i. This matrix assists compute the similarity between any two users. Many similarity metrics are available [7], we use Pearson correlation coefficient [11], which is one of the most popular and the most efficient

[1] http://www.epinion.com

in the RS domain [13], its value varies within the range $[-1, +1]$, where -1 means that the two users are completely opposite to one another, and $+1$ means that they are completely similar.

In order to predict how much the current user u_a will rate an item r, the system exploits the ratings of similar users to u_a (equation 1) out of the set of users who rated r (U_r).

$$p(u_a, r) = \overline{v_{u_a}} + \frac{\sum_{u_j \in U_r} f_{simil}(u_a, u_j) \times (v_{(u_j,i)} - \overline{v_{u_j}})}{card(U_r)} \tag{1}$$

Where:
$f_{simil}(u_a, u_j)$: the similarity between u_a and u_j.
U_r: the set of users who have rated r.
$card(U_r)$: is the number of users in U_r.
This is called Resnick formula. Neighbors in this approach are identified automatically by the prediction function, consequently the approach is sensible to the user's rating strategy. Cold start [14] is one of the essential drawbacks of this approach. It consists in the difficulty to generate recommendations to users who did not rate enough items, because it is difficult to find neighbors to them. The same difficulty can also results from certain ratings strategies such as: rating items which are not frequently rated by other users, or appreciating items that are globally detested by the community.

2.2 Trust aware recommenders

Trust-aware approaches have the advantages of reducing the impacts of the major weaknesses of CF recommenders such as the cold start [14], data sparsity [8], recommendation acceptability [15] and robustness to malicious attacks [16, 2, 17, 18], without bringing the recommendation accuracy down [9].

A correlation between trust and users similarity was found in [19] and [20]. Replacing user similarity with trust relationships has been proposed by [12, 25]. This approach is applied only in social systems where users can rate each other.

In order to compute recommendations, a trust-aware RS interrogates the friends of A, if the result was not satisfying the system interrogates the friends of A's friends and so on.

Trust-aware prediction function is the same as that of CF, with replacing the similarity value by trust value.

Commonly, trust propagation algorithms represent the dataset as a directed weighted graph, where users represent the nodes, the trust relationships represent the edges, and the trust values represent the weights. Trust propagation problem becomes a graph traversal problem. The main difference between those algorithms is about their strategies in traversing the graph, and selecting the path between the source and destination nodes.

In our studied case trust is a binary value. That is why we choose the model MoleTrust for our experiments. This algorithm is adapted and tested to our dataset. In MoleTrust, each user has a domain of trust where he adds his trustee users. In this context, user can either fully trust other user or not trust him at all. The model considers that trust is transitive, and that its value is inversely proportional to the distance between the source user and the destination user. The only initializing parameter is the maximal propagation distance d.

If user A added user B to his domain, and B added C, then the trust of A in C is given by the equation:

$$Tr(A,C) = \begin{cases} \frac{(d-n+1)}{d} & \text{if } n \leq d \\ 0 & \text{if } n > d \end{cases} \quad (2)$$

Where n is the distance between A and C ($n = 2$ as there two steps between them; first step from A to B, and the second from B to C).
d is the maximal propagation distance.

Consider $d = 4$ then: $Tr(A,C) = (4 - 2 + 1)/4 = 0.75$.

2.3 Hybridization

In [1] the author identifies seven strategies to hybridize multiple recommendation approaches, he argues that there is no reason why recommenders from the same type could not be hybridized.

In [28], authors propose to enhance Resnick formula by adding a global trust (reputation) value to the similarity score. To compute reputation score, they apply a CF recommender with one neighbor at a time. The global trust of a user is the number of correct recommendations that he could produce (while neighbor), divided by the global number of recommendations in which he was involved. A recommendation is considered correct when the difference between its value and the real one is smaller than a given threshold.

Authors argue that trust here represents the competence of the user to generate recommendations, i.e the usefulness of the user to the system. Trust in this model is computed implicitly. Like in CF, neighbors are still chosen automatically.

Giving more weight to users identified as more useful improves the accuracy compared to classical CF, but it has no impact neither on the coverage nor on the cold start problem (while user still needs to rate a considerable number of items before receiving good recommendations).

In [27], we applied five hybridization strategies on epinion dataset. Compared to trust-aware and CF recommenders, most hybrid models could improve the prediction coverage, without a serious decrease in the prediction accuracy. The best score was obtained by applying a weighted hybridization strategy, shown in the equation 3, with ($\alpha = 0.3$).

$$score(u_a, u_j) = \alpha \times simil(u_a, u_j) + (1 - \alpha) \times trust(u_a, u_j) \quad (3)$$

2.4 users behavior analysis

The fluctuations across users is a common issue in RSs, so the system can be accurate for some users while inaccurate for others. This is usually explained by quantitative variance of user activeness or behavior, i.e the number of ratings the user (for CF), and the number of trust phrases (in trust-aware RS).

Few studies were dedicated to qualitative evaluations of user activeness. [3] is an example where authors are interested in the popularity of rated items. They consider item's popularity as: the ratio between the number of 5 stars notes the item receives in the training corpus, and the number of 5 stars notes it receives in the whole corpus after prediction. This definition was useful to improve recommender accuracy, by orienting RS towards more popular items considering that they are more probable to be accepted by the users. At the opposite, we think that the item is popular when many people rate it regardless the value of their notes, as we shall see in 4.1.

Other factors that we propose are: ratings abnormality, number of trust relations, and reputation of trustee friends. To the best of our knowledge, no other definitions were proposed to these factors.

3 Experiments and performance evaluation

3.1 DataSet

Epinion dataset contains 49,290 users who rated a total of 139,738 items. users can rate items in a range of 1 to 5, the total number of ratings is 664,824. Users can also express their trust towards others (binary value), the dataset contains 487,182 trust ratings. It is important also to mention that 3,470 users have neither rated an item nor trusted a user, these users are eliminated from our statistics, thus the final number of users is 45,820 users.

In [26], authors showed on this corpus how to improve both accuracy and coverage (number of predicted ratings) by replacing similarity metrics with trust-aware metrics. The improvement of coverage was limited because of the fact that some users are active in rating items but not in rating reviewers. 11,858 users have not trusted anybody in the site (25.8% of users). Those users have made 75,109 ratings, averagely 6.3 ratings by user. This high average means that recommendations could be generated to this category by a similarity based approach. On the other hand, 5,655 users have not rated any item in the site (12.3% of the total number of users). The average of trust relationships by user in this set is 4.07 which is not negligible, those users suffer from the same problem with the similarity approachs while trust based approach can generate recommendations to them.

We divide the corpus to two parts randomly, 80% for training and 20% for evaluation (a classical ratios in the literature). We took into consideration that every user has 80% of his ratings in the training corpus and 20% in the evaluation corpus, this is important to analyse the recommendation accuracy by user.

3.2 Structural performance Evaluation

Our test consists in trying to predict the ratings value of the test corpus. Our performance evaluation includes two aspects; accuracy and coverage.

To measure accuracy, we employ the mean absolute error metrics (MAE) [24]. MAE is a widely used predictive accuracy metrics. It measures the average absolute deviation between the predicted values and the real values. MAE is given by the following equation:

$$MAE = \frac{\Sigma_{i=1}^{N} |p_i - r_i|}{N} \tag{4}$$

Where: p_i is the rating value predicted by the recommender to the item i. r_i is the real rating value supplied by the user to the item i.

MAE focuses on ratings but not on users [25]. Take the case of a user who rated 100 items, received 20 good predictions, while other 5 users, each of whom has rated 5 items, received 1 bad prediction by user. MAE still consider the system successful in 80% of cases. Truth is; this system is able to satisfy one over 6 users. User mean absolute error (UMAE) [25] is the version of MAE which consider users' satisfaction. It consists in computing the MAE by user, before computing the average of these values. We call this average global UMAE or GUMAE.

With regard to the coverage aspect, we employ two forms of coverage metrics: Coverage of prediction is the ratio between the number of predicted ratings to the size of the test corpus. Coverage of users is the number of users who received predictions divided by the total number of users.

Table 1 illustrates the MAE, GUMAE and both forms of coverage for the three recommendation approaches (CF, Trust and hybrid). It is obvious that the hybrid model surpasses both CF and trust-aware approaches in both forms of coverage, without a serious lose in accuracy. This is because hybrid system uses each approach to predict ratings unpredictable by the other approach.

Strategy	**MAE**	**coverage**	**GUMAE**	**users coverage**
Pearson correlation	0.84	61.15%	0.8227	47.46%
MoleTrust	0.8165	69.28%	0.8079	52.21%
Weighted ($\alpha = 0.3$)	0.8210	76.38%	0.8124	62.22%

Table 1. Accuracy and coverage of RS

Intuitively, the more the user rates items, the more the CF is able to recommend items to him. The same role is applied for the trust-aware recommender and the number of users a user trusts.

As for a hybrid recommender, both roles are applied. Nevertheless, we note that some users who have a considerable number of ratings/trust relation still have a larger UMAE than others who have less number of ratings/trust relations. This lead us to analyze their ratings/trusting strategies in order to answer this question.

4 User strategies analysis

In this section we analyze many characteristics of user behavior and rating strategy. The aim of which is to explain the recommendation accuracy fluctuation across users. In this context, we represent user behavior by four criteria, one of which is quantitative (number of trusted friends), and the three others are qualitative. We need also to say that the first two criteria (user Ratings' popularity and abnormality) consider the quality of user's item ratings. The last two criteria (number of trusted friends and their reputation) are dedicated to the social influence and the quality of trust relation that the user does.

In the four following subsection, we illustrate the relations between of each criterion and the UMAE value of users, trying to explain the impact of this criterion on the performance of RS.

4.1 user Ratings' popularity

We define item's popularity as the number of ratings that the item gets. Users tend to rate popular items more than unpopular item [3], this behavior creates an important bias in items popularity. By consequence, RS tends to recommend popular items more than others. This can limit the choices of users and reduce the serendipity in the RS.

The question here is about the user choice of items to rate, and how can this influence the performance of the recommender.

Now we define user's ratings' popularity as the average of the popularity of items who have been rated by this user. We compute the user's ratings popularity value for all users, then in figure 1 we show the relation between it and UMAE.

In order to have a readable figures, we categorize the population into 20 categories, users are grouped in function of their increasing ratings' popularity value, with regarding that every category contains about 5% of the population. This percentage is not fix, because we are conscious to keep users having the same ratings' popularity value in the same category. We compute, then, the average of UMAE of the members of the category. Therefor every point in the curve represent the average of UMAE of nearly 5% of the whole population.

Note that in figure 1, the more UMAE is low the more accurate are the recommendations. Thus we can find that users who have a high ratings' popularity value (more than 100) are receiving the less accurate recommendations. This results from the fact that those very popular items are usually less discriminant and less informative to RS because they are appreciated by almost everybody.

4.2 Abnormality coefficient

This measurement distinguish users with particular taste. we tend to study user's rating strategies versus the global orientation of the community.

Formally: we compute the average rate of the item, then the difference between the rate supplied by the current user and this average. The Abnormality

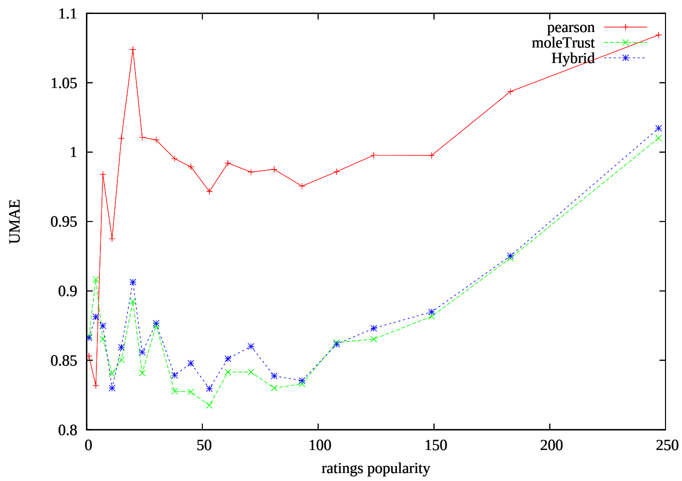


Fig. 1. UMAE and ratings popularity

coefficient of the user is the average of differences between his ratings and the average rate of each item he rates.

$$abn(u) = \sum_{i}^{N} \left(\frac{|r_{ui} - \overline{r_i}|}{N} \right) \tag{5}$$

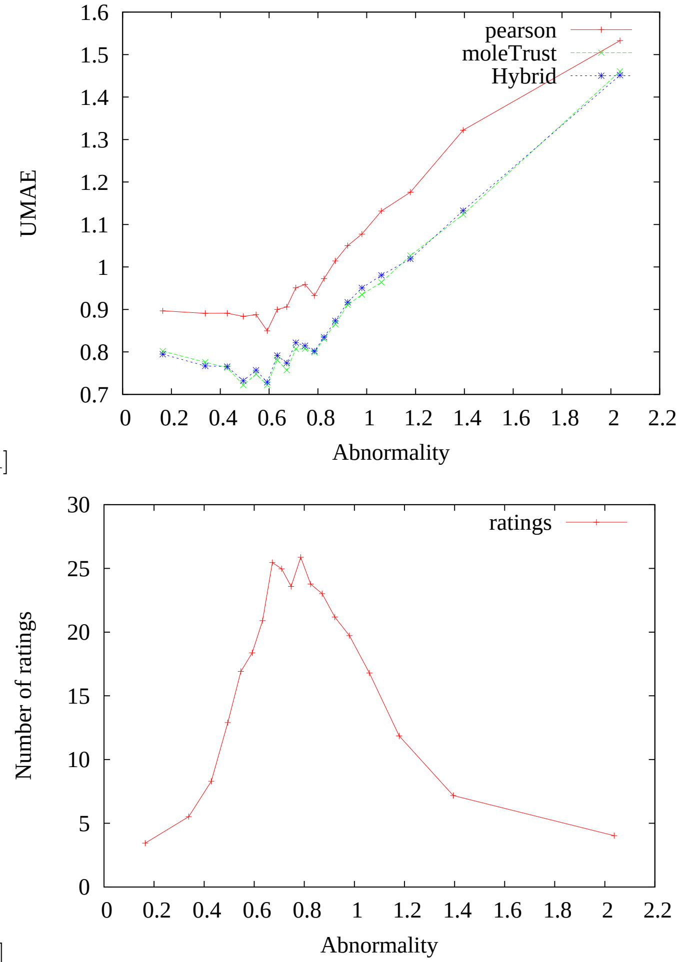
Where: N: is the number of items rated by the user u.
r_{ui}: is the rate given by the user u to the item i.
$\overline{r_i}$: is the average rate of the item i.

Figure 2 has the same structure as figure 1 with one difference is that users' categorizing is done in function of their increasing abnormality coefficient.

Regarding to figure 2 [A], UMAE is relatively very high for users with large abnormality coefficient, which means that users whose ratings are close to the average rates of the rated items receive more accurate recommendations than those whose ratings is opposite to the tendency of the community. The part [B] of the same figure illustrates the distribution of average number of ratings in the abnormality categories. Users in categories with high abnormality (more than 1.4) and categories with low abnormality (less than 0.4) have nearly the same number of ratings. Looking at those same categories in the figure [A], we notice that they are on both extremes of UMAE. It is obvious here that, for users with small quantity ratings, abnormality is a discriminant factor of RS performance, rather than number of ratings.

4.3 Number of trusted users

This factor links the number of trusted users with the UMAE. It is intuitive that the more the user trusts people, the more the system can recommend items to



[A]

[B]

Fig. 2. Abnormality and UMAE

him. Even though, we find it is important to have a close look on the details of this correlation. The curve in figure 3 represents a Hyperbolic cosecant function. This means that trusting more users is in general beneficial for any user, but it is more beneficial for users with a low number of trust relations, while it becomes slightly beneficial for those with numerous relations.

4.4 Reputation of trusted users

In 4.3, we discussed the number of people the user trusts, but we think that this is not the only factor, derived from a trust relationships, to influence the performance of RS. The reputation of the trusted persons is a key issue for RS. In this section, we illustrate the impact of trusting reputed /not reputed people

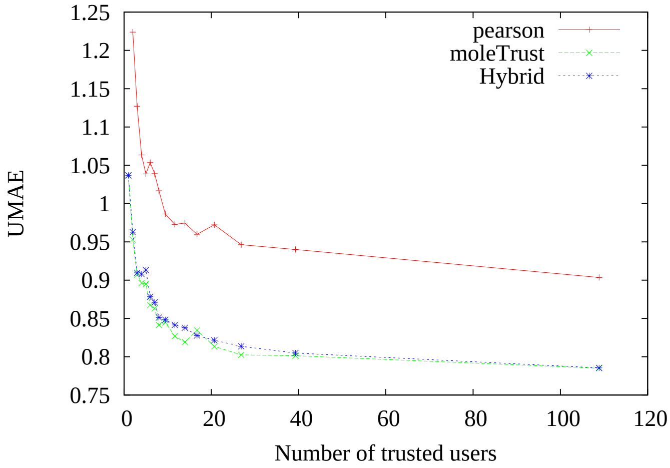


Fig. 3. Number of trusted users and UMAE

on the quality of recommendations.ted /not reputed people on the quality of recommendations.

We consider a primitive metrics of reputation; the reputation of a user is the number of users who trust him.

$$Rep(u_i) = Nb.trusters_{u_i} \tag{6}$$

Where: $Nb.trusters_{u_i}$ is the number of people how trust u_i.

We think that even when a user trusts few people, this can be more informative to RS when these people are well reputed. Therefore, our current factor $Trep(u_a)$ is the average of the reputations of users that the user u_a trusts, shown in the equation 7. Figure 4 illustrates the relationship between this average and the UMAE. Like precedent factors, users are categorized in groups. This categorization is based on the values of our $Trep$ value metrics.

$$Trep(u_a) = \frac{\sum_i^N Rep(u_i)}{N} \tag{7}$$

$u_i \in D(U_a)$ (the group of users who are trusted by u_a).

The curve in figure 4 shows that UMAE is relatively high when average of reputation is very low (less than 10), whereas it is almost stable after that. This shows that gaining reputation is not a complicated issue in this context, it is sufficient that the user shows positive intention to a few users to have a sufficient reputation in the community.

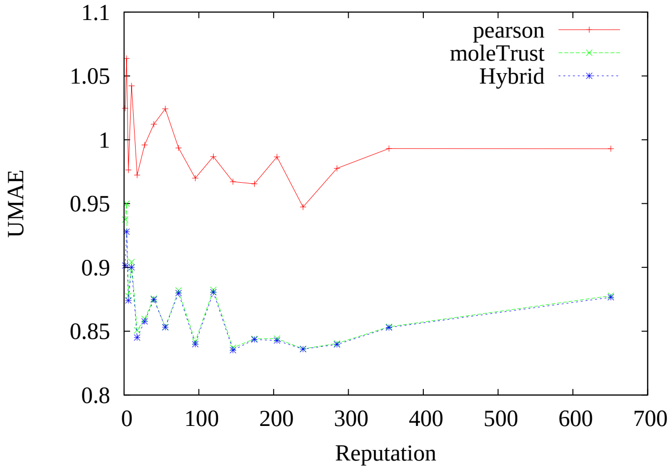


Fig. 4. reputation of trusted users

5 Conclusion and future works

In this paper we showed that, even though trust-aware recommenders improve the accuracy of CF recommenders, the hybrid model can, once again, make use of both approaches to surpass their performances, and generate recommendation to a wider set of users community without a serious decrease in accuracy.

We also showed that it is important to analyze the performance of the system regarding to various users behavior, which can lead in the future to build a model aware to different users ratings and trusting strategies.

In this paper, we analyzed the behavior criteria separately, it will be interesting in the future to elaborate an analysis by clustering users according all criteria together, and to build user profile in function of his own strategies.

Even though epinion is a known corpus in the literature, we think that is important to test our model on other corpora, and to elaborate the same analysis in order to generalize our results.

The nature of current corpus restricted our choice of trust metrics. We hope that upcoming tests be done on datasets with numeric trust values, which allow to test other trust metrics.

References

1. R. Burke. 2007. Hybrid web recommender systems. In The adaptive web, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Lecture Notes In Computer Science, Vol. 4321. Springer-Verlag, Berlin, Heidelberg 377-408.
2. R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik. Identifying attack models for secure recommendation. In in Beyond Personalization: A Workshop on the Next Generation of Recommender Systems, 2005.

3. Harald Steck. 2011. Item popularity and recommendation accuracy. In Proceedings of the fifth ACM conference on Recommender systems (RecSys '11). ACM, New York, NY, USA, 125-132.
4. Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10). ACM, New York, NY, USA, 713-722.
5. Paul Resnick and Hal R. Varian. 1997. Recommender systems. Commun. ACM 40, 3 (March 1997), 56-58.
6. J. Ben Schafer, Joseph Konstan, and John Riedi. 1999. Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (EC '99). ACM, New York, NY, USA, 158-166.
7. Laurent Candillier, Frank Meyer, and Franoise Fessant. 2008. Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems. In Proceedings of the 8th industrial conference on Advances in Data Mining: Medical Applications, E-Commerce, Marketing, and Theoretical Aspects (ICDM '08), Petra Perner (Ed.). Springer-Verlag, Berlin, Heidelberg, 242-255.
8. Bradley N. Miller, Joseph A. Konstan, and John Riedl. 2004. PocketLens: Toward a personal recommender system. ACM Trans. Inf. Syst. 22, 3 (July 2004), 437-476.
9. P. Massa and B. Bhattacharjee. Using trust in recommender systems: An experimental analysis. In iTrust04, pages 221235, 2004.
10. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classifcation: using social and content-based information in recommendation. In Proceedings of the fifteenth national/tenth conference on Artifcial intelligence/Innovative applications of artifcial intelligence, 1998.
11. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94). ACM, New York, NY, USA, 175-186
12. J. Golbeck and J. Hendler. FilmTrust: movie recommendations using trust in web-based social networks. 2006.
13. Upendra Shardanand and Pattie Maes. 1995. Social information filtering: algorithms for automating word of mouth. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '95), Irvin R. Katz, Robert Mack, Linn Marks, Mary Beth Rosson, and Jakob Nielsen (Eds.). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 210-217.
14. David Maltz and Kate Ehrlich. 1995. Pointing the way: active collaborative filtering. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '95), Irvin R. Katz, Robert Mack, Linn Marks, Mary Beth Rosson, and Jakob Nielsen (Eds.). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 202-209.
15. Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00). ACM, New York, NY, USA, 241-250.
16. Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Trans. Internet Technol. 7, 4, Article 23 (October 2007).
17. Shyong K. Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In Proceedings of the 13th international conference on World Wide Web (WWW '04). ACM, New York, NY, USA, 393-402.

18. Michael O'Mahony, Neil Hurley, Nicholas Kushmerick, and Gunol Silvestre. 2004. Collaborative recommendation: A robustness analysis. ACM Trans. Internet Technol. 4, 4 (November 2004), 344-377.
19. Cai-Nicolas Ziegler and Georg Lausen. Analyzing correlation between trust and user similarity in online communities. In Trust Management, pages 251265. 2004.
20. Lee, D. H. & Brusilovsky, P (2009) Does Trust Influence Information similarity?, Proceedings of Workshop on Recommender Systems & the Social Web, the 3rd ACM International Conference on Recommender Systems, New York, NY, USA, October 22 25, 2009.
21. Jennifer Golbeck. Personalizing applications through integration of inferred trust values in semantic web-based social networks. In Semantic Network Analysis Workshop at the 4th International Semantic Web Conference, November 2005.
22. Ugur Kuter and Jennifer Golbeck. 2010. Using probabilistic confidence models for trust inference in Web-based social networks. ACM Trans. Internet Technol. 10, 2, Article 8 (June 2010), 23 pages.
23. Cai-Nicolas Ziegler and Georg Lausen. 2004. Spreading Activation Models for Trust Propagation. In Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04) (EEE '04). IEEE Computer Society, Washington, DC, USA, 83-97.
24. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22, 1 (January 2004), 5-53.
25. P.Massa, P.Avesani. 2004. Trust-aware Collaborative Filtering for Recommender Systems. In Proc. of Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE. pages 492-508.
26. P.Massa, P.Avesani. 2006. Trust-aware Bootstrapping of Recommender Systems. in ECAI Workshop on Recommender Systems.
27. C.Haydar, A.Boyer, A.Roussanaly. 2012. Hybridising collaborative filtering and trust-aware recommender systems.
28. John O'Donovan and Barry Smyth. 2005. Trust in recommender systems. In Proceedings of the 10th international conference on Intelligent user interfaces (IUI '05). ACM, New York, NY, USA.
29. McNally, Kevin and O'Mahony, Michael P. and Smyth, Barry. 2011. Models of Web Page Reputation in Social Search. Third IEEE International Conference on Social Computing (SocialCom2011), MIT, Boston, USA.
30. Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. 2003. The Eigentrust algorithm for reputation management in P2P networks. In Proceedings of the 12th international conference on World Wide Web (WWW '03). ACM, New York, NY, USA, 640-651.
31. Raph Levien and Alexander Aiken. 1998. Attack-resistant trust metrics for public key certification. In Proceedings of the 7th conference on USENIX Security Symposium - Volume 7 (SSYM'98), Vol. 7. USENIX Association, Berkeley, CA, USA, 18-18.