Article

AgreeTrust, a simple Implicit Trust Inference Model for Memory-based Collaborative Filtering Recommender Systems.

Ahemd Zahir 1\*, Yuan Yuyu 2, and Krishna Moniz 3

1 Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications, Beijing 100876, China; xahiru@gmail.com

2 Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications, Beijing 100876, China; yuanyuyu@bupt.edu.cn

3 Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications, Beijing 100876, China; krishnamoniz@live.nl

**\*** Correspondence: xahiru@gmail.com; Tel.: +86-131-204-4403

Received: date; Accepted: date; Published: date

**Abstract:** Trust is a popular replacement for the similarity measure among the Collaborative Filtering researchers. However, it is difficult to get the explicit trust relation between users. A common solution is to use of the prediction accuracy as the implicit trust. However, this requires an additional step of calculating the predictions. In this paper, we present, AgreeTrust, a much simpler and efficient method in which trust is directly inferred from the user ratings. We have shown that when combined with similarity, our method outperforms its counterparts.

**Keywords:** Trust; Collaborative Filtering; Recommender System

1. Introduction

Recommendation systems often employ the Collaborative Filtering (CF) method to make predictions. The preferences of similar users are aggregated to predict a personalized recommendation [1]. The intuition is that, users who had similar preferences in the past would have similar preferences in the future. Preference similarities are calculated as correlation between users’ rating profiles. Most common similarity measures include Pearson, Cosine, and Jacquard correlation [2], [3]. Correlation based CF systems suffer from various problems such as cold-start, a situation where finding similar users is difficult due to lack of ratings [4], [5]. In order address the issues with correlation, trust is often used as a replacement for similarity in Recommender Systems (RS). The intuition behind using trust is, that users are more likely to accept a recommendation from a trustworthy partner.

Success of incorporating trust in computer models have been shown in various context, such as reputation systems e.g. Amazon.com [6], dynamic network [7] and mobile environment [8]. For a detailed survey of various computational trust models, interested readers can refer to [9]. Researchers have been incorporating many aspects of social relationship between the users to increase the efficiency of recommender systems. Trust has become one of the key avenue for such exploration. Recently, there has been a growing number of work on trust-based recommender systems. Use of trust in recommender system has, also, been shown to alleviate problems such as cold start and data sparsity [10]. Furthermore, trust has been used to increase the understandability as well as to improve the robustness of CF recommender system [11].

In general, trust computation methods are broadly classified as two types; implicit trust, where trust between two users are inferred from their rating profiles, and explicit trust, where existing social links are used for trust inference. The goal of both methods is to use the underlying trust relationship to aggregate the user preferences in such a way, so that more weight is given to trustworthy partners.

Although, intuitively it is more sensible to use explicit trust for prediction, in many real-world recommender systems, it is difficult to get the social link data among the users. For instance, many online shops do not require users to be registered, in order to buy products. In addition to that, using social network data increases the risk of exposing the users’ privacy [12]. Furthermore, number of ratings available is far greater than that of explicit trust links, which often comes in the form of binary value. It is possible to generate real values for the binary data however it could add noise to the data.

In this paper, we propose a novel approach, for calculating implicit trust in recommender systems, which considers both positive and negative ratings. We have shown that proposed method performs reasonably better than the traditional memory-based methods as well as other implicit trust models. The rest of the paper is organized as follow. The related work section explores the background of previous works incorporating trust in RS. Next, we detail the problem and formalized our model. The experiment section, covers the details of data set and the evaluation metrics. In result section, we compare the performance of the proposed method with the baseline methods. We conclude the paper highlighting the possible future works.

2. Related Work

Similar to traditional brick-and-mortar businesses, trust plays a vital role in the success e-commerce business [7]. In [13], Massa & Avesani showed that incorporating trust increase the efficiency of the recommender system. Several trust inference models have been proposed, to increase the accuracy of the recommender system using explicit trust links, see e.g. [5], [10], [14], [15]. Availability explicit trust information is relative very low compare to that of ratings. We focus of our study on implicit trust inference

O’Donovan [16] proposed a method based on recommender’s contribution to prediction accuracy. The trust is calculated as the ratio of the correct number of recommendation or total number of recommendation. Higher the contribution to the accuracy, trust between two users increases. Similar to our approach, Resnick prediction formula is used to calculate the predictions. The algorithm defines a contribution threshold , to filter the trusted neighbors. The main disadvantage of this method is the time complexity of trust matrix generation. For each user, the algorithm requires to make a prediction for all other users, in order to calculate the absolute difference between the predicted rating and ground truth. Also, the method does not consider the direction of agreement, but simply the accuracy either by sum of squared errors or mean absolute error. In contrast, our proposed trust model takes direction of agreement into consideration

Addressing the issues of using similarity in kNN CF, Lathia et al. [17] proposed a similar trust-based k-nearest neighbor CF method. In this method, the absolute prediction difference, between two users, is subtracted from 1, before being divided by the maximum of the rating scale. The average of the result is taken as trust value between the users. The trust value is ranged from 0 to 1. If the target user does not have co-rated items, then trust between them is 0. The predictions are then made using the trust matrix rather than the similarity. This method assumes a recommender with negative similarity correlation is more trustworthy, to the target user, than those who have not yet rated the item. Similar to [16], this method depends on prediction to be calculated in order to generate the trust matrix

Similarly, Pitsilis & Marshall [18] derived trust by measuring the uncertainty in the similarity values. The users’ inability to make accurate predictions is modelled as *uncertainty*. The similarity matrix is then scaled to according the user’s belief and disbelief on the rating provider (trustee). The sum of belief, disbelief, uncertainty adds up to 1. Although inclusion of belief in this model inclines to subjective probability, the essence of prediction depends on the correlation of users. In this aspect, the model is similar to ours.

Li et. al [19] improvised model proposed in [16] by including preference similarity, recommendation trust, and social relations into the recommendation algorithm. In their recommendation trust analysis module, the implicit trust is calculated exactly same as in [16]. Therefore, we argue that replacing their trust module with ours would increase the performance of the recommender system.

Most trust inference methods we discussed earlier requires predictions to be calculated before generating the trust matrix. This involves use of a similarity matric for making initial predictions. Inspired by O’Donovan’s model, we propose AgreeTrust, a much simpler method in which trust is inferred directly from the user preferences and does not require prior predictions. The users’ positive and negative preferences are used to generate trust between them.

3. Preliminaries and Model

Let be users, be the items, the rating represents user ’s evaluation on item , and is the maximum of the rating scale (often 5). The predictions are calculated using the k-nearest neighbor version of Resnick method as in Eq. (1):

(1)

where is the predicted rating of item j for the user u. and are the mean ratings of user u and v respectively. The similarity between u and v is using the Pearson correlation given by:

(2)

3.1. AgreeTrust

Similarly, let and be the set of items rated by user and user respectively, and be the trust between them. The trust , is calculated as the ratio of agreements, i.e. sum of positive agreements and negative agreements, in co-rated items:

(3)

(4)

(5)

where , separates positive and negative ratings. Trust value is in the range of [0,1] where 0 means no trust and 1 is completely trustworthy. The positive agreements do not overlap with negative agreements, thus, guarantees that trust value would not grow beyond 1. Positive agreements are the number of items both users have liked, similarly negative agreements contain items both users disliked. Predictions are then made by replacing the similarity in Eq. (1) with trust:

(6)

where is the predicted rating.

4. Experiment

4.1 Data set

In order to evaluate our method, we used publicly available Movie Lens 100k benchmark data set [20]. The data set contains 100,000 ratings from 943 users on 1682 movies. The ratings are scaled from 1 to 5. The ratings are divided into a training set containing 80% of the rating and a test set containing 20% rating.

Note that the data set is very sparse. Since our model is based on the co-rated items for trust calculation, trust values between users who does not have a common rating will be 0. We compare the results of AgreeTrust with two benchmark methods; O’Donovan’s trust model (denoted as O’Donovan), the trust model proposed by Pitsilis and Marsh (denoted as PitsMarsh), as well as combined trust and similarity (denoted as SimTrust), and with basic kNN with means.

In [16], O’Donovan adopted harmonic mean to combine trust and similarity. However, we found our model gives best accuracy when using simple athematic mean as in Eq. (7):

(7)

where and are the trust matrix and similarity matrix, respectively. is generated using the Eq. (3) and is calculated using the Pearson correlation formula as in Eq. (2).

We used *surprise* framework [21], a well-known python framework for recommendation, to implement our algorithm. We used an Intel Core i5 2.7GHz with 8GB RAM Macbook Pro to run our experiment. We set r = 5 (since the maximum rating scale of ml-100k data set is 5), =40, and trust threshold = 0.2 (for O’Donovan’s) in the experiment unless otherwise stated.

4.2 Evaluation Metrics

Most commonly used accuracy metrics in CF are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Therefore, we adopt these two matrices to measure the prediction accuracy our model. MAE treats all errors equally while RMSE punishes more on higher deviation from ground truth. RMSE is calculated as:

(8)

Mean Absolute Error (MAE) is a calculated as:

(9)

where is the predicted rating, is the actual rating of the *jth* item, and is the predicted rating matrix. MAE and RMSE will be equal if there are no variance in errors.

5. Results and Discussion

5.1 Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **AgreeTrust** | **SimTrust** | **O’Donovan** | **PitsMarsh** | **kNNmeans** |
| RMSE (item) | 0.9355 | **0.9283** | 0.9517 | 0.9353 | 0.9394 |
| MAE (item) | 0.7358 | **0.7293** | 0.7506 | 0.7349 | 0.7364 |
| RMSE (user) | 0.9547 | **0.9409** | 0.9620 | 0.9430 | 0.9438 |
| MAE (user) | 0.7539 | **0.7374** | 0.7571 | 0.7381 | 0.7381 |

Table 1. Accuracy of the prediction algorithms

We can see from the result table 1. that all the RMSE values are higher than MAE. The difference in error metrics shows that variance within set of error values. For all the methods, item-based predictions have better accuracy than user-based prediction. Although not significant, there is a clear improvement in prediction accuracy for most of the trust-based method, except user-based O’Donovan’s. For instance, MAE for PitsMarsh is equal with kNNmeans, however PistMarsh has a lower RMSE, suggesting that PitsMarsh has low variance in errors.

O’Donovan performs worst of all the methods. Similarly, AgreeTrust performs worse than kNNmeans on user-based CF. However, on item-based CF, AgreeTrust results are better than kNNMeans. Although, PistMash method performs better than sole AgreeTrust, we should note that *belief* calculation in their method involves aggregation of similarity.

The best result is achieved when similarity is combined with AgreeTrust (SimTrust), outperforms all the methods with an average increased accuracy of 0.99%. Especially, item-based CF with SimTrust is significantly better then rest of the contenders.

5.2 Complexity analysis

The pairwise trust calculation requires time as the model loops through each user for rating comparison with other users. In contrast to our AgreeTrust model, O’Donovan’s model is prohibitively time expensive since it requires additional time as there is a prediction step for each user. Thus, all the models, that requires prediction during trust matrix generation, requires ) time. Due to the expensive time requirement of the O’Donovan trust matrix generation, the only practical way to implement the method in a real-world system is to pre-calculate the trust matrix. On the other hand, AgreeTrust takes far less time to generate the trust matrix.

6. Conclusion and Future Works

In this paper, we have proposed AgreeTrust, an implicit inference model for Collaborative Filtering system. The basic idea is that trust relationship between trustor and trustee relies on the ratio of agreed number of rating (considering both positive and negatives) and total number of co-rated items. We have shown, by experimental evaluations on real benchmark data set, that AgreeTrust improves prediction accuracy. In contrast to most implicit trust inference methods that uses rating as primary source trust, our model takes into account the direction of rating agreement between users. Unfortunately, our trust inference method does not count the non-co-rated items. If any two users have no common item, then the trust between them is 0, and it is very common to find user pairs without any common ratings. As future work, we plan to explore ways to address the sparsity of trust matrix.

Modern recommender systems are heavily relying on deep learning models. However, not much work has been done to incorporate trust in deep learning models. Therefore, we plan to investigate the effect of incorporating trust in deep learning models. Furthermore, most trust implicit models use same prediction method for both trust generation and model evaluation. In this work, we have used users rating for trust inference. Alternatively, we could have used predicted ratings. Another direction for future work is to explore the efficiency of trust models under different prediction methods for trust generation.

**Supplementary Materials:** For the purpose of reproducibility, we have published the code for our experiment at http://github.com/xahiru/agreerecom.

**Author Contributions:** Conceptualization of the original idea, conducting the experiment as well as writing—original draft preparation was done by Ahmed Zahir. Formal analysis, review and editing is done by Krishna Moniz. Yuan Yuyu supervised the project.

**Acknowledgments:** This work is supported by Natural Science Foundation of China (Grant No. 91118002).

**Conflicts of Interest:** The authors declare no conflict of interest.

References

[1] Z. Huang, H. Chen, and D. Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, 2004.

[2] F. Ricci, L. Rokach, and B. Shapira, “Recommender systems: introduction and challenges,” in *Recommender systems handbook*, Springer, 2015, pp. 1–34.

[3] G. Guo, J. Zhang, and N. Yorke-Smith, “A Novel Bayesian Similarity Measure for Recommender Systems.,” in *IJCAI*, 2013, pp. 2619–2625.

[4] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 253–260.

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

[6] Amazon.com Inc., “Amazon,” *www.amazon.com*, 2018. .

[7] M. Carbone, M. Nielsen, and V. Sassone, “A formal model for trust in dynamic networks,” in *Software Engineering and Formal Methods, 2003. Proceedings. First International Conference on*, 2003, pp. 54–61.

[8] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss, “A Specification of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies,” 2004.

[9] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[10] G. Guo, J. Zhang, and D. Thalmann, “A simple but effective method to incorporate trusted neighbors in recommender systems,” in *International Conference on User Modeling, Adaptation, and Personalization*, 2012, pp. 114–125.

[11] P. Pu and L. Chen, “Trust building with explanation interfaces,” in *Proceedings of the 11th international conference on Intelligent user interfaces*, 2006, pp. 93–100.

[12] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu, “Link privacy in social networks,” in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 289–298.

[13] P. Massa and P. Avesani, “Trust metrics in recommender systems,” in *Computing with social trust*, Springer, 2009, pp. 259–285.

[14] J. Golbeck, J. Hendler, and others, “Filmtrust: Movie recommendations using trust in web-based social networks,” in *Proceedings of the IEEE Consumer communications and networking conference*, 2006, vol. 96, no. 1, pp. 282–286.

[15] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 135–142.

[16] J. O’Donovan and B. Smyth, “Trust in recommender systems,” in *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005, pp. 167–174.

[17] N. Lathia, S. Hailes, and L. Capra, “Trust-based collaborative filtering,” in *IFIP international conference on trust management*, 2008, pp. 119–134.

[18] G. Pitsilis and L. F. Marshall, *A model of trust derivation from evidence for use in recommendation systems*. University of Newcastle upon Tyne, Computing Science, 2004.

[19] Y.-M. Li, C.-T. Wu, and C.-Y. Lai, “A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship,” *Decis. Support Syst.*, vol. 55, no. 3, pp. 740–752, 2013.

[20] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *Acm Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, 2015.

[21] N. Hug, “{S}urprise, a {P}ython library for recommender systems.” 2017.

H:\documents\layout\new template June 2014\figures\CC-BY logo original v1.wmf© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).