### AgreeTrust, a simple implicit trust inference method for memory-based CF recommenders

## Introduction

Recommendation systems often employ Collaborative Filtering (CF) method to make predictions. The preferences of the similar users are aggregated to predict a personalized recommendation [huang]. 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. Correlation matrices suffer various problems []. In order address the issues with correlation, trust is often used as replacement for similarity in recommenders. The intuition behind using trust is that users are more likely to accept a recommendation from trustworthy partner.

Application incorporating trust in computer model have been shown successful in various context, such as reputation systems [epinion, amazon], dynamic network [19 of f]and mobile environment [20 of F]. For a detailed survey of various computational trust models, interested readers can refer to [josang]. In order to increase the efficiency of recommender system, researchers have been incorporating many aspects of social relationship between the users. Recently, there has been a growing number of work on trust-based recommender systems. Use of trust in recommenders has also been shown to alleviate problems such as cold start and data sparsity [5 in A]. Furthermore, trust has been used for increase the explainablity as well as to improve the robustness of CF-based recommenders [para 1, chapter2, F].

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 [Stanford] [ include film in C]. Furthermore, number of ratings available is far greater than number of explicit trust link, 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 preferences.

In this paper, we propose a novel approach for calculating implicit trust in recommenders. 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 incorporating trust in recommenders found in the literature. Next, we detail the problem and formalized our model. In the experiment and result section, we compare the performance of the proposed with the baseline method. We have chosen, O’dnonvan’s item-trust-profile as baseline method, since we draw motivation for the proposed model from their model.

Related Work

Similar to traditional brick-and-mortar businesses, trust plays a vital role in the success e-commerce business [Trust worthiness in ecommerce 4, 5, 13]. Mase & Avane showed that incorporating trust increase the efficiency of the recommender system. Several trust inference models have been proposed, by various researches in the field, to increase the accuracy of the recommender system, see e.g. […add any number of references here]. The focus of our study is explicitly on implicit trust inference.

Odnovan [] 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 formula is used to calculate the predictions. The algorithm defines a contribution threshold , to filter the trusted neighbours. 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.[Lathia] proposed a similar trust-based k-nearest neighbour CF method. The trust between two users is calculated as the average of 1 minus absolute rating difference over the maximum of rating scale for all the co-rated items. Thus, 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 [odnovan’s method], this method depends on prediction to be calculated in order to generate the trust matrix.

Simialrly Pitsilis & Marshall 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 our proposed model.

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

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. The users’ positive and negative preferences are used to generate trust between them.

Preliminaries and Model

Let be users, be the items, and the rating represents user ’s evaluation on item . The predictions are calculated using the Resnik method as in Eq.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:

AgreeTrust

Similarly, let and be the set of items rated by user and user respectively, and be the trust between them.

The trust ,can be calculated as the ratio of agreements, i.e. sum of positive agreements and negative agreements, in co-rated items as:

where , and is the maximum of the rating scale (often 5). 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 equation 1 with trust k-nearest neighbours:

where is the predicted rating.

Trust matrix calculation algorithm 1

Get number the co-rated item positively co-rated items and negatively co-rated items

For each user get the rated items

For all other users get the co-rated item

Positives 🡪 length(common > )

Negtivae 🡪 length(common < )

Agreement 🡪 negative + postive

Trust 🡪 agreement/length(co-rated item)

Experiment and Result

Dataset

To evaluate our method, we use publicly available Movielense 100k benchmark data set. The data set contains 100,000 ratings from 943 users on 1682 movies. The ratings are scaled from 1 to 5. In order to reproduce results of O’Donovan’s model, 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 Pits and Marsh (denoted as PitsMarsh), and with baseline kNN. We used *surprise* framework, a well-known python framework for recommendation, to implement our algorithm.

We set r = 5 (since the maximum rating scale of ml-100k data set is 5), =40, and trust threshold = 0.2 (for O’Dnovan’s) in our experiments unless otherwise stated.

We have also explored the effect of combining trust and similarity. The Sim\_Trust column of the result table 1 shows the accuracy of the predictions when trust is combined with similarity using AgreeTrust model. In [] 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.322

where and are the Trust matrix and Similarity matrix, respectively. is generated using the equation 3 and is calculated using the Pearson correlation formula as in equation 2.

Evaluation metrics

Most commonly used accuracy matrices 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:

Mean Absolute Error (MAE) is a calculated as:

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.

Accuracy comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | AgreeTrust | Sim\_Trust | O’Donovan | PitstMarsh | kNNwithMeans |
| RMSE (item) | 0.9355 | 0.9283 |  | 0.9353 | 0.9394 |
| MAE (item) | 0.7358 | 0.7293 |  | 0.7349 | 0.7364 |
| RMSE (user) | 0.9547 | 0.9409 |  | 0.9430 | 0.9438 |
| MAE (user) | 0.7539 | 0.7374 |  | 0.7381 | 0.7381 |

As we can see from the result table 1. all the RMSE values are higher than MAE. The difference in error matrices shows that variance within set of error values. For all the methods, item-based predictions have better accuracy than user-based prediction. There is a clear improvement in prediction accuracy for all the trust-based method, although the improvement is not significant. For instance, MAE for PitsMarsh is equal with kNNwithMean, however PistMarsh has a lower RMSE, suggesting that PitsMarsh has low variance in errors. The results, further confirm finding of previous researches that claim inclusion of trust in the CF based recommender increases the accuracy.

User-based AgreeTrust performs worse than all the other methods. However, item-based AgreeTrust results are better than kNNwithMeans. The best result is achieved when similarity is combined with AgreeTrust.

AgreeTrust outperforms its counter parts as well as basic kNN, with an average increased accuracy of 9.3%. Odnovan’s and PitsMarsh methods have a relatively similar accuracy. To our surprised, it performs worse than regular kNN. Among the evaluated trust methods, PitsMarsh performs the worst. As in their work, we have only used belief to derive trust, for details, see [].

Although, there is a clear improvement in accuracy for both item-based and user-based predictions using the combined Sim\_Trust, results are not as good as lone AgreeTrust.

Basic kNN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | AgreeTrust | Sim\_Trust | O’Donovan | PitstMarsh | kNN |
| RMSE (item) | 1.0041 | 0.9991 | 1.0546 | 1.0355 | 1.0428 |
| MAE (item) | 0.7891 | 0.7922 | 0.8421 | 0.8289 | 0.8341 |
| RMSE (user) | 1.0036 | 0.9936 |  | 1.0040 | 1.0041 |
| MAE (user) | 0.7947 | 0.7883 |  | 0.7991 | 0.7967 |

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

Complexity analysis

The pairwise trust calculation requires O(N2) time as model loops through each user for rating comparison with other users. In contrast to our AgreeTrust model, Odnovan model is prohibitively expensive since it requires additional O(kN2) time as there is a prediction step for each user. Thus, all the models, that requires prediction during trust matrix generation, requires O(kN2 + N2) 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 matrices. On the other hand, AgreeTrust takes far less time to generate the trust matrix.

Conclusion

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 significantly. 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 counts 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 who do not have common item ratings. We plan to explore ways to address the sparsity of trust matrix as future work.

Modern recommenders 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 used users’ direct 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.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 91118002)

Reference:

Huang, Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering

Quasi, An effective recommender system by unifying user and item trust information for B2B applications

Yung-Ming Li ⁎, Chun-Te Wu, Cheng-Yang Lai , A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship

F = [https://pdfs.semanticscholar.org/d550/f9437c22d72be8fcadd3ad0fd77c66752a65.pdf

The Role of Trust in Collaborative Filtering

A = <http://www.ntu.edu.sg/home/ZhangJ/paper/sac14-guibing.pdf>

C = trust <file:///Users/xahiru/Downloads/sac14-guibing.pdf>

[Yuan et. al]W. Yuan, L. Shu, H. Chao, D. Guan, Y. Lee, and S. Lee. itars: trust-aware recommender system using implicit trust networks. Communications, IET, 4(14):1709–1721, 2010.

[Lathia et al. ]file:///Users/xahiru/Documents/trust/CFTrustmodel/sessionTrust/087a8aadfb2eddd82220bf393433a4a329bd.pdf

[Stanford ]http://theory.stanford.edu/~korolova/link\_privacy\_CIKM08.pdf