The Method of Trust and Reputation Systems Based on Link Prediction and Clustering

Jiwan Seo, Seungjin Choi, and Sangyong Han*

Chung-ang University, Seoul, Republic of Korea
{jwseo,bethemoney}@ec.cse.cau.ac.kr, hansy@cau.ac.kr

Abstract. Online environments offer a major advantage that data can be accessed freely. At the same time however, they present us with an issue of trust: how much any data from online sites can be trusted. Trust and Reputation Systems (TRS), developed to address this issue of trust on network, quantify reliability in terms of semantics and derive a trust-network from a targeted online data. The performance of TRS is often hindered despite the promises because the number of links formed in the ideal scenario frequently is not reached, suffering from the problems of cold-start and sparsity. In this paper, we propose a method in which Link Prediction(LP) and Clustering are applied to TRS so that these two problems are adequately addressed. We evaluate our proposed method with a recommendation system we constructed. Our experiment results show that our method positively contributes to the performance of a recommendation system and help control the problems of cold-start and sparsity in TRS.

Keywords: Trust and Reputation Systems, Link Prediction, Clustering.

1 Introduction

Spread of the Internet in recent years has made it possible for the general public to search and access data easily. At the same time, a problem has risen whether users could trust such data unlike as in off-line situations [1]. Therefore in most cases, users end up relying on the information provided either by commercial sources with vested interest or any arbitrary third parties. Such inputs can be biased or inaccurate, causing users to make unintelligent decisions [2,3]. Trust and Reputation Systems (TRS) effectively address these two problems. Being based on the Word-of-Mouth (WoM) algorithm, TRS form a virtual network using links people with direct experiences on a given topic. From this virtual network, TRS can extract useful data for various application programs. However, two major problems hinder the performance of TRS when a real-world data set is applied [4]. One is, constructing functioning trust-networks for new users is hard because there is insufficient amount of data related to them.The term the cold-start problem refers to this. The other problem refers to a case when links

* Corresponding author.

C. Fernandez-Gago et al. (Eds.): IFIPTM 2013, IFIP AICT 401, pp. 223–230, 2013.

between nodes are not formed even though they are expected to in theory. This is called the sparsity problem. Both of these problems deteriorate the performance of a system based on a trust-network.

There are two algorithms that can address these two problems: the Link Prediction and Clustering algorithms. The LP algorithm analyzes the structural characteristics such as betweenness, degree, and density of the nodes in a network and determines the strength of the feasibility between any two unconnected nodes [5,6]. Then links with good chances of forming in the future are identified. As the number of feasible links increases through the LP algorithm, sparsity problem is directly controlled. The Clustering algorithm, on the other hand, addresses the sparsity and cold-start problems differently by utilizing the concept of group of nodes. The Clustering algorithm identifies clusters in a network, assigning each node a membership to a cluster or clusters. It is assumed that a node can trust the other nodes within its cluster with more confidence even without direct links established between them, and this distinction helps control the sparsity and cold-start problems. In this paper, we suggest an improved trust-network to which the LP and Clustering algorithms are applied and use this in an application program to evaluate our method. For experiment, we derived a trust-network from a real-world data set of a film-rating website. This data was quantified from various semantic points and the LP algorithm was applied with the structural perspective to improve the trust-network. Then the Clustering algorithm was applied as well, resulting in significant clusters. With the resultant trust-network, we constructed a movie recommendation system, with which we evaluated our method.

The remaining parts of this paper are organized as follows: In Section 2, we discuss works related to TRS, the LP algorithm, the Clustering algorithm. Section 3 details our proposed method. Section 4 shows a comparison of performances between the conventional method and ours. Further explorations are suggested in Section 5.

2 Related Work

2.1 Trust and Reputation Systems

In an online environment, user access data from other users. In this case, it is hard to tell whether the sources of the data have any vested interest in the topic in which users are interested. In the end, they have to resort to relying on the data from dubious sources. Users might end up making purchases online without knowing about the goods and services directly. Taking actions without prior direct experience on a given topic is referred to as the risk of performance [2]. This risk can be addressed with TRS, which are based on the WoM algorithm. The WoM algorithm extracts opinions and data from a virtual network of people with direct experiences on a given topic so as to suggest valuable information for decision making processes. Based on direct experiences of other users, both trust and reputation systems in common estimate credibility. But, the trust system utilizes trust value, which is derived from each user, and the reputation

system uses reputation value, a reliability factor to can derive from groups. On the other hand, the performance of programs based on TRS was decreased by some problems, which were called the cold-start and sparsity problems. So, recently many researches proposed the method based on extending the trust and reputation network for controlling these problems [10,11].

2.2 The Link Prediction and Clustering Algorithm

The LP algorithm [5] evaluates the relation between any of the two unconnected nodes within a network based on the structural characteristics of the pair in relation to the network. With this evaluation, this algorithm then produces a set of links that are likely to form in due course. Many variants of the LP algorithm have been developed in recent years. One of them, the Common Neighbor algorithm, computes the possibility of a link formation between any two unconnected nodes by assigning scores. This is done by comparing the neighbor group of an arbitrary node x and the neighbor group of another arbitrary node y. Another variant, the Adamic-Adar Index algorithm, extends the Common Neighbor algorithm with the additional step of discounting the scores of higher-degree nodes found among the common neighbor nodes.

The Clustering algorithm [8] exploits the characteristics of the definition of a community within a network. A community, which is a group of nodes by definition, has higher density of links inside itself. By identifying a community that a user belongs to, application programs may consider only a restricted range of nodes rather than all of the nodes within the network because noisy links can be ignored, hence more useful results can be obtained. There are a handful variants of Clustering algorithms with different methods of constructing a cluster. A cluster consists of a head node and member nodes which surround the head. The various kinds of Clustering algorithm essentially differ in selection of head nodes and their surrounding member nodes.

3 Proposed Method

3.1 Establishing the Trust-Network

A trust-network can be constructed by quantifying trust, the numbers of which are calculated with statistical and probabilistic analyses. A link might be suggested between users A and B after these analyses. Higher frequency of interactions between two nodes signifies a stronger feasibility of a link between them. We built a trust-network for a recommendation system in our experiment using data obtained from a movie ratings site. For this, we chose the users' profiles and movie ratings that ranged between 1 and 5.

We assumed that the tastes of two users were similar when both of them gave the same ratings to a movie. The number of matches in their ratings were normalized to yield a matrix that contained a value within the range of 0 and 1. When this was rounded with a threshold that we manipulated, a weighted undirected

sparse trust-network was obtained. User profile was one major component for building our trust-network. Our user-profile-based trust-network took the gender, age, profession and area of residence of each user as input. Our assumption was that each of these properties had an influence on users movie preferences. We defined and assigned weight to each property based on the following equation. Affinity is an attempt to gauge how much each of the properties influences a users taste for movies. The following equation was used to calculate affinity:

$$Affinity = \frac{NumberofLink_{occurrence}}{NumberofLink_{possible}} \tag{1}$$

Numeric values of profile-based user affinity were normalized by dividing each affinity value with the total sum of the profile affinity values. Common attributes of profiles from a pair of users were translated to numeric affinity values between the pair, and the values were added up to be assigned as the strength of links. Going through this step for every pair of users yielded a user-profile-based trust-network.

3.2 Link Prediction for Enhancing Trust-Network

We chose the Common Neighbor algorithm among the many LP algorithms available because of its simplicity in implementation and effectiveness in finding significant results. The Common Neighbor algorithm takes the number of common neighbor nodes that a given nodes of x and y have in order to calculate the score of each edge. A higher score between two nodes signifies that both of the nodes are likely to form a new link judged from the weights of their neighboring links. Whereas it is an unweighted undirected network that the Common Neighbor algorithm takes when considering the number of common neighbors, our method used a weighted undirected trust-network. This makes it necessary for our method to use the following equation to calculate the score of each predicted links in order to reflect the weight of each neighbor node [9]:

$$score(x, y) = \sum_{a \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, a) + w(y, a)}{2} \tag{2}$$

3.3 Grouping with the Clustering Algorithm

Using the Clustering algorithm allowed us to identify a cluster or clusters that an arbitrarily chosen user belongs to. clusters were identified from either the structural or the semantic viewpoint of the network. This guaranteed improved services for users as well as program scalability. We used the K-means Clustering algorithm because of its ease of implementation [7]. In this algorithm, K seeds are selected, each of which is the header for each cluster; then members are added to clusters based on their relation with the seeds. Once all of the nodes in the network have obtained membership to any one of the clusters, the first step was finished and evaluation on this clustering began. If the evaluation result turns

out to be unsatisfactory, another new set of headers are selected and clustering is repeated. In constructing our movie recommendation system, we decided that K users with higher degrees were the influential users in the trust-network, and chose them as seeds for the Clustering algorithm.

3.4 Collaborative Filtering for Recommendation System

We enhanced the trust-network and selected significant groups following the aforementioned steps. After enhancement, we constructed a recommendation system based on this trust-network and the widely used Collabrative Filtering(CF) algorithm was used for this. This algorithm, based on the K-nn algorithm, estimates weights by considering similarities between users, from which recommendations are made. How distance is defined and calculated yields different results, but the trust-network itself can provide the distance data for the purpose of our experiment. Therefore, the CF algorithm can be implemented using our trust-network and Euclidean Distance. In addition, the group selected via the Clustering algorithm becomes the target to apply the CF algorithm.

4 Experiment

4.1 Experiment Method

To test our proposed method, we used the actual data set from the website MovieLens, a movie ratings site. We considered movie ratings data made by 1,000 arbitrarily chosen users, and picked a timestamp that divided the ratings data chronologically: this timestamp was set to a point when 80% of the ratings were done. The data before this time dividing timestamp were used as training data and the rest for evaluation. To evaluate our method, we built three different trust-networks - the first one was based on user tastes, the second on user profiles, and the third integrated the first and the second. The additional test cases were where only the LP algorithm was applied and where the Clustering algorithm was applied.

We used Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) indices, which are widely used for the purpose of evaluating the performance of recommendation systems. MAE index signifies average of differences between actual values and estimated values. RMSE is the standard deviation of the differences between actual values and estimated values. While both of them evaluates the recommendation performance of a recommendation system, RMSE especially, can measure the stability of performance of a recommendation system. By using these two indices as measuring tools, we demonstrate that the problems of cold start and sparsity can be controlled by comparing the number of links before and after our method was applied. The Table 1 shows the result of our experiment.

Table 1. Results of experiments

	Tn			LP-Tn			LPCl-Tn		
	taste	profile	hybrid	taste	profile	hybrid	taste	profile	hybrid
MAE	0.765	0.739	0.765	0.701	0.797	0.701	0.641	0.775	0.641
RMSE	1.013	0.927	1.012	0.897	1.004	0.897	0.822	0.972	0.823

4.2 Experiment Analysis

There are nine columns for data in the table as three different trust-networks were implemented, each of which was tested with or without the LP and the Clustering algorithms applied. As the rating scores range between 1 and 5, the values of RMSE and MAE are less than 4; the closer the value approaches 0, the more similar the predicted value is to the actual values.

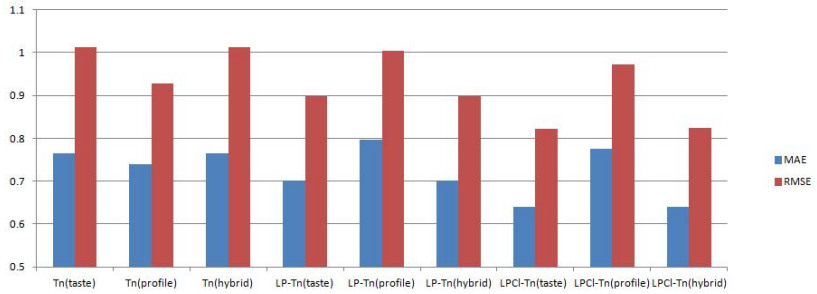

Fig. 1. Graph of results

Tn(taste), which was the trust-network of taste, performed worse than the profile-only trust-network. Because you can see in the table 2, the taste-based trust network consists of a sparse matrix than the profile-based. It is hard to derive the rating for recommendation. The LP and Cl algorithms improved the result to a great degree over when Tn(taste) was used. We observed that the number of links in the network increased with the application of the LP algorithm. This implies that users can take advantage of trust of more users. Such increase of feasible links contributes to controlling of the cold-start and sparsity problems. Also, the Clustering algorithm identifying significant clusters to which weights were assigned enhanced performance. Use of the Clustering algorithm made the recommendation system more receptive to the opinions from more significant users than opinions from less significant users. This reduced the influence of noisy data, resulting in enhanced performance. However, if trust-network isn't sparse, as like a Tn(profile), LP and Cl make the noisy data and cloud systems judgment. Two trust-networks merged together can complement each other. One of our trust-networks, Tn(hybrid), was constructed this way. In short, the LP algorithm enlarged the scope of significant links so that there

were more opinions to be considered. The Clustering algorithm increased receptivity towards opinions of significant users. Both, when combined together, improved the recommendation performance of the trust-network-based recommendation system.

Table 2. Comparing original network and prediction network

	Number of links	The percentage of links
Tn(taste)	5,742	1.149%
Tn(profile)	350,541	70.18%
LP-Tn(taste)	37,566	7.520%
LPCL-Tn(taste)	47,586	9.527%

The cold start and sparsity problems are caused because the number of links in a trust-network is small, both of which deteriorate the performance of application programs based on a trust-network. Using the proposed method in our paper contributes to controlling of the two problems because the LP algorithm increases the number of links as shown in the table. The Clustering algorithm contributes to controlling of the two problems as it uses the data from members of focused clusters.

In the processes of constructing a trust-network and going through the LP algorithm, promising links are rounded; the threshold for rounding fundamentally affects the link density of the trust-network. In our experiment, the performance of our application program was negatively influenced when the number of links was either too small or too large.

Table 3. The MAE in accordance with change of round

	No round	Average round	Round(0.2)
MAE	0.767	0.751	0.765

5 Conclusion and Suggestion for Further Explorations

In this paper, we proposed the method based on LP and Clustering algorithms for solving problems which were named cold-start and sparsity problems. We constructed a trust-network from a semantics viewpoint to improve reliability. The LP algorithm and the Clustering algorithm were applied for improving the reliability from the structural viewpoint. Based on our proposed trust-network, we implemented a recommendation system operating on a real-world data set to test whether an improvement in performance can be achieved with our proposed method. We could observe that when the LP algorithm is applied to a trust-network, the density of the network changes depending on how rounding is performed in forming links, which affects the performance of the recommendation system. We also observed that applying the Clustering algorithm to a

trust-network improves the trust-network by finding significant links. Our proposed method, by expanding links and finding significant links, could successfully control the problems of cold-start and sparsity that frequently take place in a trust-network and improve the application program. It should be noted, however, that the algorithms used in this paper are comparatively easy to implement, a reason for variable results depending on how the algorithms were set and combined. Securing improved performance thus requires flexible and intelligent combining of algorithms depending on the characteristics of the system being developed. Further explorations may investigate how trust-networks are constructed from various viewpoints as well as how different algorithms are combined intelligently to provide improved services to users.

Acknowledgments. This research was supported by the MKE(The Ministry of Knowledge Economy), Korea and "LG Electronics". under IT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency) (H0504-23-1022) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012009387).

References

1. Seo, J., Kim, M., Han, S.: The method based on professionalism of evaluator for reliability reputation model. In: Park, J.J(J.H.), Jeong, Y.-S., Park, S.O., Chen, H.-C. (eds.) EMC Technology and Service. LNEE, vol. 181, pp. 205–211. Springer, Heidelberg (2012)
2. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43, 618–644 (2007)
3. Yang, Y., Sun, Y.L., Kay, S., Yang, Q.: Defending Online Reputation Systems against Collaborative Unfair Raters through Signal Modeling and Trust. In: ACM symposium on Applied Computing, pp. 1308–1315 (2009)
4. Kim, M., Seo, J., Noh, S., Han, S.: Identity management-based social trust model for mediating information sharing and privacy enhancement. Security Comm. Networks (2011)
5. Getoor, L., Diehl, C.P.: Link mining: a Survey. ACM SIGKDD Explorations Newsletter (2005)
6. Butts, C.T.: Network inference, error, and information (in)accuracy: a Bayesian approach. Soc. Networks 25, 103 (2003)
7. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. In: JSTOR (1979)
8. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3) (2005)
9. Murata, T., Moriyasu, S.: Link Prediction of Social Network based on weighted proximity measures. IEEE (2007)
10. Chen, C., Wan, Y., Chung, M., Sun, Y.: An effective recommendation method for cold start new users using trust and distrust networks. Information Science (2013)
11. Pitsilis, G., Knapskog, S.J.: Social Trust as a solution to address sparsity-inherent problems of Recommender systems. ACM RecSys (2012)