A Recommender Model Based on Trust Value and Time Decay

Improve the Quality of Product Rating Score in E-commerce Platforms

Muhittin IŞIK
Institute of Science and Engineering
Kadir Has University
Istanbul, Turkey
muhittin-isik@hotmail.com

Hasan DAĞ
Management Information Systems
Kadir Has University
Istanbul, Turkey
hasan.dag@khas.edu.tr

***Abstract*— Most of the existing products rating score algorithms do not take fake accounts and time decay of users' ratings into account when creating the list of recommendations. The trust values and the time decay of users' ratings to an item may improve the quality of product rating score in e-commerce platforms, especially when it is thought that nowadays the majority of customers read the reviews before making a purchase.**

In this paper, we first introduce the concept *trust value of users* by explaining its mathematical definition and redefine the product rating score based on users' trust relationship. Then we calculate the product rating score based on time decay by making the concept *time decay* clear. After that we execute both algorithms together in order to show their both effects on the quality of product rating score. Finally, we present experimentally effectiveness of three approaches on a large real dataset.

Keywords*— *Recommender Systems, Trust Rank, Time Decay, Rating Score

I. INTRODUCTION

Online consumer reviews bring a number of benefits such as saving time and money, finding experienced products by other consumers but most of the e-commerce platforms do not have any verification or authentication mechanisms on their online users' reviews related to their products even if online ratings and reviews have become quite determinant on customers' purchase decisions. Report [1] indicates that almost 82% of internet customers read reviews before making a purchase. This information shows us that the importance of online reviews/ratings has a great effect on purchasing behavior of customers in e-commerce platforms. Even if 78% of customers think the information found online is vital and more trustful than advertisements, it is confirmed that most of the reviews are fake [2]. Generally, these fake accounts are created by the owner of the companies, e-reputation companies, malicious users or biased bloggers. Sometimes on average 100 fake accounts can easily identify the rating score of a hotel such as on TripAdvisor or on booking.com if that hotel does not have much rating. Finding a common ground between an effective recommender system for gathering user reviews and a strong control mechanism to avoid abuses is the most important duty [2]. To overcome the negative effects of the fake accounts in e-commerce platforms, we try to create a recommender model which finds the consumers who are trustful and have a great effect on other users' opinion.

Time is also an important factor while calculating product rating score especially in some e-commerce platforms such as hotels, restaurants, travel agencies and other service based companies. Most of the existing algorithms calculate rating score of products or services based on ratings but they ignore the time of each user's rating. As we mentioned above, for some e-commerce platforms rating time is a crucial factor since their products or services change over time. That is to say, a review made yesterday and a review made ten years ago should not be considered as having the same value when calculating a hotel's rating score. For this purpose, we propose a simple algorithm considering rating time when calculating of a product or a service rating score.

In this study to overcome the issues mentioned above, we present a product rating score algorithm which calculates the product rating score based on trust values of users and the time decay of users' ratings. We think that the quality of product rating score calculation can be improved by looking at these two factors. For the calculation of product rating score, our algorithm tries to find each user's trust value based on PageRank algorithm by looking at given users' relationships and at the same time it calculates the time decay of users' ratings to that product. Thus, each user affects the rating score of the products with regard to his/her trust value and time decay of his/her rating.

The rest of this paper is organized as follows. After the introduction, Section II contains some of the recent works especially related on social network-based, trust-based and time-based recommender systems. In Section III, we present details about the mathematical definition of finding trust value of each user by given relationship between users and the calculation of product rating score based on time decay of ratings' date. Section V introduces a real-life dataset used in our experiment and reports some experimental outcomes and evaluates the proposed algorithm. Finally, we conclude and discuss our future work in Section VI.

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

II. RELATED WORK

There is a vast literature on recommender systems and in this section; we review some major approaches especially to social network-based recommender systems, trust-based recommender systems and time-based recommender systems.

A. Social Network-Based Recommender Systems

Traditional recommender systems are generally based on users rating history, items description, or user demographic information to provide preferable services, information, items or people. But nowadays, personalized recommender systems have become an essential tool in order to deal with continuously increased items almost in all environments such as in e-commerce platforms, social networks, search engines, service applications, etc.

Social network-based recommenders generally depend on one of the followings; collaborative filtering, content based filtering, demographic information, or influential ranking algorithms. Trujillo et al. [3] carry out a research to work up the performance of the recommender systems based on multi features such as demographic and psychographic information to calculate the similarity between users. The similarity between users is calculated by adding firstly demographic features, secondly interest areas, and lastly downloaded documents, respectively. Ma et al. [4] present a novel approach called SoRec (Social Recommendation) integrating users' social network information with rating matrix. According to the authors, users are affected by their social connections. Recommender systems can deal with missing values using relationships efficiently. Yu [5] proposes a new method called dynamic competitive recommendation to be used in social networks. The method calculates recommendations with several algorithms. The dynamic competitive recommendation algorithm chooses the highest one according to the results of each algorithm. Fijałkowski and Zatoka [6] present a concept to improve the effectiveness of e-commerce recommender systems using social network user profiles obtained via facebook Open API. The authors propose to obtain some objects from user's facebook profile such as user posts, published links, comments along with user's friends' posts, and comments liked by the user. These objects can be used to calculate similarity between users' interests based on keywords list in the context in order to enrich dataset in e-commerce platforms. Carrer-Neto et al. [7] use an application based on movies. According to the experiment results, adding social heritage to the recommender system decreases the quality of the recommender since more contents affect it. Sun et al. [8] suggest an approach that integrates social network graph to improve the prediction accuracy of recommender systems using a biclustering algorithm by finding most suitable group of friends. According to the authors when we do not separate different type of friendships between users, all will be treated equally. But in this way, we can't improve the accuracy of recommendation. Han et al. [9] propose a prediction model in order to find interests of a user who doesn't have enough information to make a recommendation on online social network. For this purpose, the authors utilize social information such as demographic information, social relationships (user friend list, friend similarity), and obtainable users' interests. Yuan et al. [10] show that friends have different influence on users' behavior on social network. Some of them, called *buddy* in the research, have strong influence on user. To find user's closer friends and calculate his/her susceptibility, the authors use rating similarity and edge embeddedness. If the target user and his/her buddy have high taste similarity and common friends, it means that buddy has a great influence on the target user. Chaney et al. [11] present a probabilistic model called Social Poisson Factorization combining user preferences for items with social network influence information by developing a scalable algorithm. According to the authors, when we decide to choose something, our behaviors are affected by our general preferences and influences of our closest friends. Gan [12] proposes a novel method called COUSIN which is a network based regression model correlating object and user similarity profiles. For this purpose, [12] creates two matrices consisting of user and object similarities using historical data. Colace et al. [13] propose a novel collaborative and user centered recommendation approach using some aspects related to the target user such as preferences, opinions, behavior, feedbacks integrated with item features and content information.

B. Trust-Based Recommender Systems

Because of providing remarkable improvements, many researchers have investigated trust-based recommender systems on social networks in recent years. The basic idea is that our preferences are not completely independent from our relationships. Namely, when we think of purchasing or choosing a product, our friends' opinions will have a significant influence on our decision. In this section, we review several major approaches for trust-based recommendations in general.

In place of using similarity between ratings of users' profiles in collaborative filtering to calculate the ratings of predicted item, O'Donovan and Smyth [14] propose a new approach based on trustworthiness of users on a specific rating prediction. Their algorithm calculates the trust by looking at the percentage of correct recommendations comparing with predicted rating between target user and any other user. Bedi et al. [15] present a recommendation method using knowledge domain for generating a recommendation. According to the selected domain by the user, the algorithm brings the product based on trust calculating by that chosen product experiences. When a user selects a product, the related domain of the target user is changed by the system. So, the trust means, in the research, how many times the users selected that product, in other word it is based on experiences. Jamali and Ester [16] propose a random walk recommender model, called TrustWalker, using both trust-based and item-based collaborative filtering recommendation. To provide a recommendation to a user, the authors apply random walks on the trust network. To construct efficient and effective recommender system, Ma et al. [17] present a novel approach called Social Trust Ensemble combining preferences of the user's with trusted friends'. To improve the prediction accuracy of recommender systems, Lathia et al. [18] propose a variation of k-nearest neighbor algorithm to reveal how much a user is close to the target user in recommender system. The trust value increases when the distance between two users' ratings decreases. Hang and Singh [19] propose a link structure and

trust network. To calculate similarity between graphs, the authors use a convergent iterative process. The method is applied on a vertex similarity measurement between graphs by calculating the similarity between trust network and a structure graph. Li et al. [20] use Multi-Criteria Decision Making method that consists of preference similarity, recommendation trust, and social relation. To predict a recommendation, the algorithm calculates all three categories. Chen et al. [21] propose a novel approach based on social trust relationships called RSTR that explicitly and implicitly uses social trust relationships simultaneously. Actually, this study is a combination of SoRec proposed by Ma et al. [4] and RSTE proposed by Ma et al. [17]. Yang et al. [22] propose an approach by the way of compounding ratings and social trust network. With this object in mind, the authors use truster-specific feature vector and trustee-specific feature vector. The main mentality of this trust based study is one can be affected by other users and one can also affect other's opinion. Deng, Huang, and Xu [23] present an approach to service recommendation using trust relationship between social network users called as RelevantTrustWalker (RTW). To measure the trustworthiness degree between users, the authors use a matrix factorization. They get recommendation results by the use of RTW which is a random walk algorithm based on the trust relevancy among users. Zhong et al. [24] propose to use directed trust graph. For this purpose, firstly the authors reveal directed and indirected relationship between users. Thus, when calculating the predicted rating, the algorithm considers if the target user has a direct or indirect relation to the other users. Guo et al. [25] carry out an empirical study to compare five different trust algorithms on two different datasets. According to the empirical study results, there is no single trust algorithm superior than others when the data set is changed. To personalize recommendations, Alahmadi and Zeng [26] present a new approach using trust relationships and comments of users' friends crawled from Twitter. The authors calculate the trust value between target user and his friend by normalized average "RT" (re-tweet action) and "L" value which indicates percentage of followers to overall number of followers and followings. Deng et al [27] present an approach called Trust-based Service Recommendation using preferences of users and trust relationships among users by looking at invoked services by each user. Keikha et al. [28] propose a method called TB-CA (trust-based context-aware) using information of user trust network to recommend items which are matched user preferences. Actually, this research is based on Jamali and Ester [16] and Shin et al [29]. To improve the quality of item recommendations in social networks, Wu et al. [30] set up a new algorithmic framework called collaborative topic regression (CTR) by social trust ensemble exploiting user-item feedback, item content and social network.

C. Time-Based Recommender systems

The latest researches show that the time factor significantly increases the quality of recommender systems, especially on e-commerce platforms. According to the observations, more recent reviews and ratings better reflect the quality of products and services. In this section, we review several major approaches for the time-based recommendations in general.

Lee et al. [31] present a novel based approach to build a recommender system based on implicit feedback. According to the empirical results, using two kinds of temporal information such as user rating time and product launching time improves either recommendation accuracy or performance. Jamali and Ester [32] investigate whether a user rates after being exposed an item rated by the target user's neighbors in a certain time. According to the experiment results, the influence of direct neighbors or rating items in the social network is higher than in the similarity network on datasets when the user are exposed to an item at a time. Zheng and Li [33] propose a new computational approach using tag and time information. For this purpose, the authors use three strategies which are "tag weigth", "time weigth" and "tag and time". They use "tag and time" strategy for calculating the target user's rating values with the combination of tag and time information. Raju et al. [34] propose an approach using a graph based structure which uses the relationships between *customers*, *products*, *customers and products*. The authors utilize a matrix which consists of visiting area information, visiting date, visiting time, need type and satisfaction level for recommendations. They apply Collaborative Filtering to find the most similar users based on filtered items and other user's additional information including day, time and need type etc. Ullah, Sarwar, and Lee [35] offer an interesting study that is the use of the recommender systems in a different area. The authors propose a smart device which recommends TV programs according to the user preferences and the user social networks data. To calculate the rating value, they divide the time which the target user spends on the program during the broadcast, with the total time of the program. Celdrán et al. [36] present a hybrid recommender and to compute the users' tracking, the authors calculate the number of the times of visits and direction of the user to recommend items on that location and lastly the date when the target user visited the related item last time. Yang et al. [37] present a hybrid recommender model which considers time of the target user's interest. According to the authors the performance of the recommender system is changed as regards the time of the user selecting an item. Namely, the current interest of a user is more effective on the performance of the recommender system. Zhang et al. [38] present time series analysis for dynamic-aware recommendation to overcome data insufficiency. The developed algorithm called FARIMA deals with the year-long period of purchasing data to provide daily-aware predictions. Xia et al. [39] propose an algorithm which uses time decay to provide dynamic item-based top-*N* recommendations. To show effects of time decay on recommendations the authors use three patterns of time decay which are *concave time decay function*, *convex time decay function* and *linear time decay function*. According to the result, algorithm with time decay provides better recommendations if the value of the time decay coefficient is chosen properly. Biancalana et al. [40] propose a hybrid recommender system which calculates contextual factors related to time to improve the quality of collaborative filtering approaches. According to the result, new items (which are movies in the study) can have higher potential of being interesting than old ones.

III. Prepare Background and Context

As online shopping constantly increases over the e-commerce platforms, the need for the quality of product rating score becomes larger. Because increasing online shopping also increases the e-reputation companies, malicious users, scammers or biased bloggers who want to affect results of product rating score algorithms in order to increase or decrease rating score of products by creating fake accounts. There are lots of surveys which show the impact of recommender systems on purchasing behaviors. The survey [41] shows that 93% of worldwide travelers say their booking decisions are impacted by online reviews and the survey [42] also shows that ratings/scores of accommodations on review site is the most important decision-making factor when booking accommodation after the price of accommodation. In this study, in order to overcome negative effects of fake accounts, we analyze the relationship among users to find trust values of users and calculate the rating score of products according to those trust values. For this purpose, we first explain the basic mathematical background of the PageRank algorithm which is used in our research to reveal relationship between users. This will let us know trust values of users.

A. PageRank Algorithm

PageRank algorithm was the backbone of the Google search engine in 2000s and developed by Larry Page and Sergey Brin at Stanford. As the well-known, the basic logic of PageRank algorithm is simply the number and importance of inbound links to a page. Namely, PageRank algorithm considers links as votes when a page gives a link to another. But one page can distribute its vote among others if it thinks more than one page represent its page. Because of this reason when your page gets a link from another page, it doesn't mean you get all its points, it means your page gets some part of its points. One more thing, because of getting more inlinks, like *cnn.com* or *bbc.com* websites can affect your page score more than hundreds of any ordinary pages if you get a link from them. Of course, it is a dynamic structure and score of each page is continuously changing on the web graph. As a result, everyone's vote is not equal as in general election on the web graph. Let's look at the mathematics of PageRank algorithm.

Brin et al. used very simple algorithm in order to calculate PageRank [43].

$$r(P_i) = \sum_{P_i \in B_{(P_i)}} \frac{r(P_j)}{|P_j|} \quad (1)$$

Equation (1) is the Simple PageRank Formula and in order to calculate the PageRank of i^{th} page ($r(P_i)$). it sums up all PageRanks ($r(P_j)$) coming from other pages and then divide by $|P_j|$ which indicates outlinks from the page P_j. $B_{(Pi)}$ indicates inlinks to the page P_i. But how can $r(P_j)$ be calculated at first? To handle this challenge, Page and Brin used an iterative formula and gave to each page an equal PageRank score when they execute the formula at the first iterate.

$$r_{k+1}(P_i) = \sum_{P_i \in B_{(P_i)}} \frac{r_k(P_j)}{|P_j|} \quad (2)$$

Equation (2) is the Iterative PageRank Formula; the PageRank value of each P_i is calculated by getting one before value of the P_j. So, in order to get $r_{k+1}(P_i)$ of page P_i at iteration k+1, we use the adjacent formula. This process is started for all page in the graph with $r_0(P_i) = 1/n$, where n is the number of page in the given related graph. To illustrate this calculation, let us apply on a simple graph.

In the paper, to calculate the trust value of each user in our dataset via PageRank algorithm, and to understand concept easily, we pretend to each user as a node on a graph.

1) Transforming Users Graph to a Matrix

In order to understand how to calculate trust value of each user easier, we transform our graph to a matrix. To illustrate how to realize all the process, we will show it on a simple graph. To transform our graph into a matrix, we use the following rule [44].

$$H_{ij} := \begin{cases} \frac{1}{N} & \text{if there is a link from } P_i \text{ to } P_j \\ 0 & \text{otherwise} \end{cases}$$

According to the rule *H*, each i^{th} user in the graph has 1 trust value at the beginning of the process and we share its trust value equally among the other users which trusted by the i^{th} user.

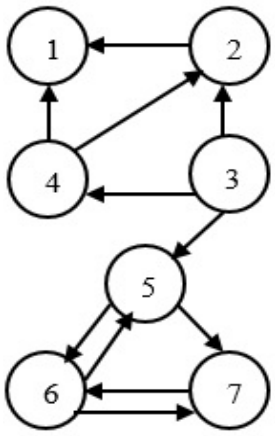

Figure 1: A graph with seven nodes

As seen on the Fig. 1, there are seven users in our graph which symbolizes a small version of our real dataset. Each arrow represents the trust from i^{th} user to j^{th} user and it means i^{th} user trusts to j^{th} user. For instance, the user 3 trusts to the user 2, 4 and 5. The user 1 trusts to no one and the user 2 trusts only the user 1. So, we can interpret the graph as follows: the user 5 distributes its trust value between the user 6 and 7 while the user 7 distributes its trust value to the only user 6. After distributing the trust value of each user, we can arrange our matrix as follow:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

As it is seen in the H matrix, if there is no trust link from the user P_i to another user, we put a 0 to that place. Namely, we do not distribute P_i's trust value to anyone. For instance, because the user 1 does not trust to anyone, we put 0 to each box of the user 1's row in the H_{ij} matrix. Accordingly, H_{ij} indicates a directed trust link from i^{th} user to j^{th} user. Likewise, N_i indicates the total number of outlink trusts from i^{th} user. Thus, each row represents the outlink trusts from i^{th} user while each column represents the inlink trusts to i^{th} user. Now, we can calculate the trust value of each user according to the obtained values by the iterative formula:

$$r(P)^T_{k+1} = r(P)^T_k * H \quad , \quad k = 0, 1, 2, \ldots \tag{3}$$

We will denote the trustworthiness value with π in the following sections. Thus,

$$\pi^T_{k+1} = \pi^T_k * H \quad , \quad k = 0, 1, 2, \ldots \tag{4}$$

2) Random Walker

In this part, we will explain a simple random walk to comprehend some problems when we calculate users' trust values on our H matrix. A random walk can be defined as a random process in which Random Walker moves randomly among users. Random Walker starts to move by choosing a user in a graph and move on to one another by randomly selecting a trust link on that user. This movement is repeated for each occurrence of a new user. But Random Walker can pass i^{th} user to j^{th} user if there is a trust link from i^{th} user to j^{th} user. So, if the j^{th} user has too much inlink trusts, the probability of the Random Walker chooses the j^{th} user will be more than other users. And since a random walk is an example of a Markov chain, choosing a user in each step is random and it's completely independent from all past history [45]. It means, the next movement of Random Walker on j^{th} user passed on i^{th} user is not affected by the i^{th} user. Random Walker goes on its way by choosing a trust link on the j^{th} user.

3) Dangling Users Problem

As we mentioned above, Random Walker passes one user to another using a trust link but sometimes it can't. For instance, when the Random Walker comes on user 1 in our graph as seen on Fig.1, it can't pass to any other user since user 1 who called as a *dangling user* trusts to no one. In this case, Random walker stops the process and starts again but this situation reduces the performance of the Random Walker. To overcome this challenge, Brin and Page applied to the following method [43].

$$d_i := \begin{cases} 1 & \text{if page "i" is a dangling node} \\ & \quad i=1,2,\ldots,n \\ 0 & \text{otherwise} \end{cases}$$

Thus, when a user trusts to no one, the algorithm distributes his/her trust value to all other users in equally. For n dimensional matrices, all the entries of the row consisted by zeros will be replaced by 1/n. So, when we execute this method on our H matrix, our new matrix formulation will be as follow:

$$S = H + (\frac{1}{n}) d e^T \tag{5}$$

Where in (5), e represents the column vector of all 1s and d indicates a vector which each entry equal to 1 if it is a dangling node or 0 otherwise as it is stated as below. If we apply this formula on H matrix, the d_1 column vector which is first row consisted of zeros will get 1 value.

$$S = H + (\frac{1}{7}) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Thus, our new matrix will be as follow:

$$S = \begin{bmatrix} 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Here, S matrix is a row stochastic matrix which sum of all the entries is equal to 1.

4) Rank Sink Sub Graphs Problem

There is another movement problem of Random walker. Sometimes some graphs can create a loop for movement of Random Walker. As seen on Fig. 1, we have two sub graphs and the first one is consisted of users 1, 2, 3 and 4 and the second one is consisted of users 5, 6 and 7. When the Random Walker passes from user 3 to user 5, it will drop a loop because it moves just among users in the second graph and it can not pass the first graph since there is no trust link from second graph to the first graph. This means that the users in the second graph get all trust values whenever Random Walker drops in the second graph. We call this problem as a *rank sink sub graphs* problem. To overcome this challenge, we use *teleportation* method providing PageRank Algorithm turn into an irreducible status. In this way, Random Walker can

make a transition between all users or sub graphs. Let's see the formula:

$$G = \alpha S + (1-\alpha)\left(\frac{1}{n}\right)ee^T \quad (6)$$

As seen in the 6, e represents the column vector of all 1s, α is the damping factor of rank sink of sub graphs (teleportation probability factor) which is between 0 and 1 (generally it is equal to 0.85). Let's apply this formula on our matrix.

$$G = 0.85\begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} + (1-0.85)(1/7)\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Thus, our G matrix will be as follow:

$$G = \begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{61}{70} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{3}{140} & \frac{32}{105} & \frac{3}{140} & \frac{32}{105} & \frac{32}{105} & \frac{3}{140} & \frac{3}{140} \\ \frac{25}{56} & \frac{25}{56} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{25}{56} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{3}{140} & \frac{25}{56} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{61}{70} & \frac{3}{140} \end{bmatrix}$$

5) Computation of PageRank Vector

All the entries in G matrix are bigger than 0 now, it means that our Random Walker will not get stuck on any users or in any sub graphs. We call this type of graphs as a *strongly connected graph*. So now we can start to calculate trust value of each user. For that we will distribute our trust value which is equal to 1 among our users equally. Because of having seven users, at the beginning of execution of the formula, each user gets 1/n trust value which is 1/7 for our graph. As we mentioned before, (2) is an iterative formula. Namely, at each iteration, users get a changing trust value according to their trust rate until certain iteration. After the certain iteration, even if we execute the formula on the G, the result will not be changed, it will reach a balance. It means each user gets the same trust value as one before iteration. Let's apply the formula on G.

$$\pi^T_{(k+1)} = \pi^T_{(k)} G \quad (7)$$

According to the (7), our first couple of iterations will be as follows:

1.step → $\pi^T_{(k+1)} = \pi^T_{(k)} G$,

2.step → $\pi^T_{(k+1)} = (\pi^T_{(k)} G)G$,

3.step → $\pi^T_{(k+1)} = ((\pi^T_{(k)} G)G)G$, and so on

Before executing the formula on our G matrix, each user has;

$$\pi^T = \begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{bmatrix}$$

If we go on to compute the trustworthiness vector, we can get ultimate trustworthiness vector (π^T*). The trustworthiness vector of each iteration is shown in the table I and ultimate trustworthiness vector (π^T*) too.

TABLE I. TRUSTWORTHINESS VECTOR

Users	Trustworthiness value of each user after each iteration				
	πT1	***πT2***	***πT…***	***πT44***	***πT****
User 1	0,2209	0,2009	…	0,1065	0,1065
User 2	0,1399	0,0929	…	0,0628	0,0628
User 3	0,0387	0,0482	…	0,0343	0,0343
User 4	0,0792	0,05924	…	0,0440	0,0440
User 5	0,1399	0,1531	…	0,1811	0,1811
User 6	0,2209	0,2439	…	0,3225	0,3225
User 7	0,1602	0,2016	…	0,2484	0,2484

As seen on Table I, The trustworthiness vector is not changed after the 44th iterations which called threshold of the iterations. Consequently, the order of the user importance is as 6 > 7 > 5 > 1 > 2 > 4 > 3 in this tiny graph. According to the result, the most important person is the user 6 and the least important person is the user 3. If we interpret the result, the Random Walker visits the user 3, 4.40% and the user 6, 32.25% of the time.

B. Time Decay of Users' Ratings

Another important deficiency in calculating rating score of a product is time factor. Most of the existing rating score algorithms do not take into account the time of users' ratings. But as we mentioned before, time is an important factor to some e-commerce platforms and it is believed that more recently reviewed products better appeal to customers' needs. For these types of reasons, in this section, we first introduce the concept of time decay by explaining its mathematical definition then we calculate a product rating score based on time decay.

In the method to be explained, we assume that a record is a quartet in which user *u* rate item *i* with *r* on the t[th] day as shown in Table 1.

1) Time Decay of a Rating

The more current rate on a product by a user, the more current information for us about the quality of that product. To calculate currency of a rate, we use the formula of the motion at constant or uniform acceleration. As you know that acceleration is the rate of change of velocity of an object. It is so common in Physics and daily life that some basic equations are derived to work out the situations in which acceleration is constant. As it is known the position equation for the constant acceleration is as follow:

$$d = \frac{1}{2}\alpha t^2 \tag{8}$$

Where d indicates the position, α is the acceleration and t is the time. In the following sections, we indicate currency of a rate as a position. Thus, our equation will be as follow:

$$c = \frac{1}{2}\alpha t^2 \tag{9}$$

According to the (9), currency or we can say that importance of a rate increases as the time increases. If the currency-time data for such a product were graphed, then the resulting graph would look like the graph at the below:

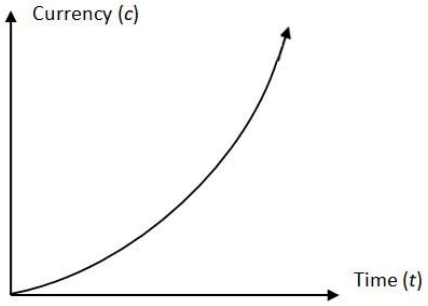

Figure 2: Currency-Time graph

According to the graph, as the date of a rate gets closer to the current day, the importance of the rate will increase. Let's show it with a real example on our dataset.

2) Rating Score of a Product based on Time Decay

Let's assume that we get the product 2 from our real dataset.

TABLE II. INFORMATION ABOUT PRODUCT 2

Itemid	Userid	Rating	Day_Distance	Date
2	2244178	4	969	2002-02-27
2	66286	4	931	2002-01-20
2	1	5	3358	2008-09-12

Where *Day_Distance* indicates the distance of the date of the rate to the date of the item 2 added to the database (which is 1999-07-04). Normal average rating score of the item 2 is 4.333 but we want to calculate based on time decay. For this purpose, we first calculate the acceleration of all ratings' date of the item according to the currency value which is set to over 100 in this study. The equation of the acceleration is as follow:

$$\alpha = 2c \div (t_1 - t_2)^2 \tag{10}$$

Currency of any rating is from 1 to 100 and time is the difference between the date of the item added to the database (t_2)and the date of the today (which is the last day of our dataset obtained, t_1). Thus, our α value will be as follow:

$$\alpha = 2 * 100 \div (2011\text{-}06\text{-}16 - 1999\text{-}07\text{-}04)^2 = 1.049$$

Now we can calculate rating score of item 2 based on weighted average with currency of the rates. The equation will be as follow:

$$\check{r} = \frac{\sum_{u=1}^{n} cr}{\sum_{u=1}^{n} c} \tag{11}$$

Where ř indicates product rating score, c is currency, and r is the rating of each user to the related product. Thus, we can calculate the weighted average of the ratings according to the currency of each rating. Currency of each user's rating is as follow:

TABLE III. CURRENCY OF EACH RATING

Userid	Rating	Day Distance ($t_1 - t_2$)	Currency ((1/2) *α*t^2)
2244178	4	969	= (1/2)*1.049*$(969)^2$=4,92
66286	4	931	= (1/2)* 1.049*$(931)^2$=4,54
1	5	3358	= (1/2)* 1.049*$(3358)^2$=59,14

Thus, weighted average of the ratings based on time decay is as follow:

$$\check{r} = \frac{(4{,}92 * 4) + (4{,}54 * 4) + (59{,}14 * 5)}{4{,}92 + 4{,}54 + 59{,}14} = 4.86$$

As you can see from the result, the rating score of the product has increased. The reason is that the user 1 gives 5 rate to the item and the date of the rate is more current than others. Other two users gave rates in 2002 but user 1's is in 2008. The difference of the date of the rates is about 6 years. This time difference is really important to us when we think about some e-commerce platforms such as hotels, restaurants, travel agencies and other service based companies. There can be many reasons for the increase of rating score of product 2. If we assume that the product 2 as a hotel, the owner of the hotel may have changed or the hotel could be renovated or the hotel's service policy may have changed, etc.

IV. Experimental Testing Results

In this section, we carry out several experiments in order to verify the quality of the proposed recommender model based on trust value and time decay. For this purpose, we perform 3 different methods, i.e., product rating score based on trust values, time decay and both, and then we compare with normal average score.

A. Dataset Description

In this paper, the dataset we use to evaluate the proposed algorithm is a real e-commerce dataset extracted from Epinions in June 2011. It is available at http://liris.cnrs.fr/red/. It is worked on two datasets. First one contains reviews from users on items and second one contains trust relationship between users.

TABLE IV. Apperance of the Rating dataset

Review id	User id	Rating	Item id	Date
51902	182	4	43286	2001-06-21

The Ratings dataset contains review id, user id, item id, user's rating, between 1 and 5, and the date of the review. There are 1127673 reviews which 113629 users have at last one rating. Table IV shows that the user 182 has a review with id 51902 and gives 4 points for the item 43286 on the date 2001-06-21. It is a .csv file and 62.0 Mb.

TABLE V. Apperance of the Trust network dataset

Trustor id	Trustee id	Value
22	434	1

The Trust Network dataset shows which user trust to whom, only positive values appear in the dataset and there are 538392 trust values which 47522 users have at last one trust relation. Table V shows the user 22 trusts the user 434. The value of 1 indicates that the user trusts to other one. It is a .csv file and 10.2 Mb.

Our dataset contains 131228 users, 317775 items and 1127673 reviews, namely our dataset has 0.003% sparsity. Our algorithms are executed on Jupyter Notebook with python version 2.7.11.

B. Findings

1) Rating Score Based on Trust Values

TABLE VI. Trust value of some users

User id	Trust Values
1	14.380358249
2	10.348435726
3	3.243809321

As we explained before, after applying the formula in (7) on our Trust Network dataset we get the results as it is shown on the Table VI. Approximately, *user 1* has 14, *user 2* has 10, user 3 has 3 trust value and so on respectively. According to the result, we can say that user 1 is the most trustful user among the three users. As it is known, most of the existing recommender algorithms calculate the product rating score by calculating the average of all ratings of the users who rated to related item. But now we can calculate a weighted average based on trust value of each user who rated the product. That is to say, user 1 will affect the rating score of a product more than user 2 if they both rated to the product [46].

$$\check{r} = \frac{\sum_{u=1}^{n} tr}{\sum_{u=1}^{n} t} \tag{12}$$

Where ř indicates product rating score, t is trust value of user, and r is the rating of each user to the related product. Thus, we can calculate the weighted average of the ratings according to the trust value of each user. Results will be as follows:

TABLE VII. Average Ratings based on Trust value

item id	Number of users	Average of ratings	Average of ratings based on trust
1	2	4.0	4.0
2	3	4.333	4.965
3	9	4.555	4.117

According the Table VII., rating score of item 1 does not change because of given the same rate by the rated users. But for item 2, there is a great difference between average rating (AR) and weighted average rating based on trust values (WARTV). Most probably one of the users who have a more trust value gives a rate to the item more than average. The same for the item 3 but this time one of the user who has a more trust value gives a rate to the item less than average. Let's examine one of these items in details.

TABLE VIII. Trust Values of each User Who Rated Item 2

User id	User's rating for item 2	Trust values of Users
1	5	14.380358249
66286	4	0.325909815
244178	4	0.201582285

As seen on the table VIII, user 1 has a great trust value more than other users. Because of this reason, even other two users give 4 rates for the item 2; item 2's rating score is almost 5.

Another important output of the research is that the average difference between AR and WARTV decreases as the number of users increases. That is mean that items which rated by too less people are affected easier by the fake accounts but according to our dataset, rating score of items rated by more than 100 people have almost the same rating score based on trust values. Let us see the average difference between AR and WARTV when the number of rated users increases on 1.000 items in our dataset.

TABLE IX. AVERAGE DIFFERENCE BETWEEN AR AND WARTV

Number of users	Number of items	Average Difference
2-10	339	0,32080
11-50	223	0,27434
>50	195	0,19845

As seen on the Table IX, as the number of users who rated the products increases, the average difference between AR and WARTV decreases. Also, the number of items rated by more than 50 users decreases when we execute the algorithm on more than 1000 items. Of course these results are just for 1000 items and when we increase the items, the results emerge clearer. Actually, when we continue to calculate rating score of the products rated by more users, we see that the average difference between AR and WARTV come closer, almost 0.

2) Rating Score Based on Time Decay

TABLE X. AVERAGE RATINGS BASED ON TIME DECAY

item id	Number of users	Average of ratings	Weighted Average of ratings by TD
1	2	4.0	4.0
2	3	4.333	4.862
9	31	4.161	3.140
17	2	2.5	3.936

As seen on the Table X, after applying (11) on our dataset, rating score of each item changes according to the ratings' date of users. Some rating scores of the items increase while some decrease. That is to say, there is no regular structure.

3) Rating Score Based on Trust Values and Time Decay

In order to find rating score of products based on TV and TD, we apply the (13) on our dataset.

$$\check{r} = \left\{ \left(\frac{1}{2} \frac{\sum_{u=1}^{n} tr}{\sum_{u=1}^{n} t} \right) + \left(\frac{1}{2} \frac{\sum_{u=1}^{n} cr}{\sum_{u=1}^{n} c} \right) \right\} \qquad (13)$$

Dividing one-half means that each method will affect the results equally.

TABLE XI. AVERAGE RATINGS BASED ON TRUSTWORTHINESS VALUE AND TIME DECAY

item id	Number of users	Average ratings	With TV	With TD	With TV and TD
1	2	4.0	4.0	4.0	4.0
2	3	4.333	4.965	4.862	4.913
9	31	4.161	4.388	3.140	3.764
17	2	2.5	3.707	3.936	3.821

As seen on the Table XI, weighted average rating score of products based on both methods is a balance between two other methods. This may be because of some users' having a high trust values, but their comments are too old or vice versa.

V. CONCLUSION AND FUTURE WORK

In this paper, we tried to improve the quality of product rating score based on trust values of users and time decay of the date of users' ratings. First, we introduce the concept of PageRank by giving its mathematical definition and redefine for revealing the relationship between users. In this way, we tried to reduce the effects of fake accounts on the rating score of products by using trust value of each user. As seen on the Table IX, the normal average rating score of products comes close to the trust values based average as the number of users who rated the related products increases. This result shows us that our algorithm is on the right track. Secondly, we introduce the concept of time decay by giving its mathematical definition and redefine for reducing the effects of old ratings when determining the rating score of products. Lastly, we apply both methods on the dataset. In this way, we break down the power of one method on the results since a product can be rated by trustful users but their ratings' date may be too old or vice versa. But if a product is rated by trustful users and their ratings' date is up-to-date enough, this indicates that the rating score of that product is actually correct. In the future work, we are considering adding the general popularity score to this calculation. Namely, if a product has received more comments in the recent times than others, it indicates that the value of that product has increased.

REFERENCES

[1] "Trust marks report 2013," Linnea Persson, European Consumer Centre Sweden, 2013.

[2] T. Nwaogu, V. Simittchieva, M. Whittle and M. Richardson, "Study on online consumer reviews in the hotel sector," European Commision / Risk & Policy Analysts (RPA) Ltd, CSES and EPRD, 2014.

[3] M. Trujillo, M. Millan and E. Ortiz, "A recommender system based on multi-features," *Proceedings of the 2007 international conference on Computational science and Its applications,* pp. 370-382, 2007.

[4] H. Ma, H. Yang, M. R. Lyu and I. King, "SoRec: social recommendation using probabilistic matrix factorization," *Proceedings of the 17th ACM conference on Information and knowledge management,* pp. 931-940, 2008.

[5] S. J. Yu, "The dynamic competitive recommendation algorithm in social network services," *Information Sciences: an International Journal,* pp. 1-14, March 2012.

[6] D. Fijałkowski and R. Zatoka, "An architecture of a web recommender system using social network user profiles for e-commerce," *Proceedings of the Federated Conference on Computer Science and Information Systems,* p. 287–290, 2011.

[7] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain," *Expert Systems with Applications, vol. 39, no. 12,* pp. 10990-11000, September 2012.

[8] Z. Sun, L. Han, W. Huanga, X. Wang, X. Zeng, M. Wang and H. Yan, "Recommender systems based on social networks," *Journal of Systems and Software, vol. 99,* pp. 109-119, January 2015.

[9] X. Han, L. Wang, N. Crespi, S. Park and Á. Cuevas, "Alike people, alike interests? Inferring interest similarity in online social networks," *Decision Support Systems,* pp. 92-106, 2015.

[10] T. Yuan, J. Cheng, X. Zhang, Q. Liu and H. Lu, "How friends affect user behaviors? An exploration of social relation analysis for

recommendation," *Knowledge-Based Systems,* p. 70–84, 2015.

[11] A. J. Chaney, D. M. Blei and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," *Proceedings of the 9th ACM Conference on Recommender Systems,* pp. 43-50, September 2015.

[12] M. Gan, "COUSIN: A network-based regression model for personalized recommendations," *Decision Support Systems, vol. 82, no. C,* pp. 58-68, February 2016.

[13] F. Colace, M. De Santo, L. Greco, V. Moscato and A. Picariello, "A collaborative user-centered framework for recommending items in online social networks," *Computers in Human Behavior, vol. 51, no. B,* pp. 694-704, October 2015.

[14] J. O'Donovan and B. Smyth, "Trust in recommender systems," *Proceedings of the 10th international conference on Intelligent user interfaces,* pp. 167-174, January 2005.

[15] P. Bedi, H. Kaur and S. Marwaha, "Trust based recommender system for the semantic web," *Proceedings of the 20th international joint conference on Artifical intelligence,* pp. 2677-2682, January 2007.

[16] 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,* p. 2009, June 28- July 1 2009.

[17] H. Ma, I. King and M. R. Lyu, "Learning to recommend with social trust ensemble," *Proceedings of the 32nd international ACM SIGIR conference,* pp. 203-210, July 2009.

[18] N. Lathia, S. Hailes and L. Capra, "Trust-based collaborative filtering," *International Conference on Trust Management,* pp. 119-134, 2008.

[19] C.-W. Hang and M. P. Singh, "Trust-based recommendation based on graph similarity," *Proceedings of the 13th AAMAS Workshop on Trust in Agent Societies,* 2010.

[20] Y.-M. Li, C.-T. Wu and C.-Y. Lai, "A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship," *Decision Support Systems,* pp. 740-752, June 2013.

[21] C. Chen, J. Zeng, X. Zheng and D. Chen, "Recommender system based on social trust relationships," *2013 IEEE 10th International Conference on e-Business Engineering,* September 2013.

[22] B. Yang, Y. Lei, D. Liu and J. Liu, "Social collaborative filtering by trust," *Proceedings of the 23th international joint conference on Artificial Intelligence,* pp. 2747-2753 , August 2013.

[23] S. Deng, L. Huang and G. Xu, "Social network-based service recommandation with trust enhancement," *Expert Systems with Applications,* p. 8075–8084, 2014.

[24] H. Zhong, S. Zhang, Y. Wang and Y. Shu, "Study on directed trust graph based recommendation for e-commerce system," *International Journal of Computers Communication & Control,* pp. 510-523, August 2014.

[25] G. Guo, J. Zhang, D. Thalmann, A. Basu and N. Yorke-Smith, "From ratings to trust: an empirical study of implicit trust in recommender systems," *Proceedings of the 29th Annual ACM Symposium on Applied Computing,* pp. 248-253, March 2014.

[26] D. H. Alahmadi and X.-J. Zeng, "ISTS:Implicit social trust and sentiment based approach to recommender systems," *Expert Systems With Applications,* p. 8840–8849, 2015.

[27] S. Deng, L. Huang, Y. Yin and W. Tang, "Trust-based service recommendation in social network," *Applied Mathematics & Information Sciences,* pp. 1567-1574, 2015.

[28] F. Keikha, M. Fathian and M. R. Gholamian, "TB-CA: A hybrid method based on trust and context-aware for recommender system in social networks," *Management Science Letters,* p. 471–480, 2015.

[29] D. Shin, J.-w. Lee, J. Yeon and S.-g. Lee, "Context-aware recommendation by aggregating user context," *IEEE Conference on Commerce and Enterprise Computing,* July 2009.

[30] H. Wu, K. Yue, Y. Pei, B. Li, Y. Zhao and F. Dong, "Collaborative topic regression with social trust ensemble for recommendation in social media systems," *Knowledge-BasedSystems,* p. 1–12, 2016.

[31] Q. T. Lee, Y. Park and Y.-T. Park, "A time-based approach to effective recommender systems using implicit feedback," *Expert Systems with Applications,* pp. 3055-3062, 2008.

[32] M. Jamali and M. Ester, "Modeling and comparing the influence of neighbors on the behavior of users in social and similarity networks," *IEEE International Conference on Data Mining Workshops,* 2010.

[33] N. Zheng and Q. Li, "A recommender system based on tag and time information for social tagging systems," *Expert Systems with Applications,* p. 4575–4587, 2011.

[34] R. Raju, I. Pradeep, I.Bhagyasri, P. Praneetha and P. Teja, "Recommender systems for e-commerce: novel parameters and issues," *International Journal of Advanced Research in Computer Science and Software Engineering,* pp. 429-253, 2013.

[35] F. Ullah, G. Sarwar and S. Lee, "Social network and device aware personalized content recommendation," *Conference on Electronics, Telecommunications and Computers– CETC 2013,* p. 528 – 533, 2014.

[36] A. H. Celdrán, M. G. Péreza, F. J. G. Clemente and G. M. Pérez, "Design of a recommender system based on users' behavior and collaborative location and tracking," *Journal of Computational Science,* p. 83–94, 2016.

[37] C. Yang, Y. Zhou and D. M. Chiu, "Who are like-minded: mining user interest similarity in online social networks," *Proceeding of the AAAI Conference on Web and Social Media,* pp. 731-734, 2016.

[38] Y. Zhang, M. Zhang, Y. Zhang, G. Lai, Y. Liu, H. Zhang and S. Ma, "Daily-aware personalized recommendation based on feature-level time series analysis," *Proceedings of the 24th International Conference on World Wide Web,* pp. 1373-1383, May 2015.

[39] X. Jiang and Y. Zhang, "Dynamic item-based recommendation algorithm with time decay," *2010 Sixth International Conference on Natural Computation ,* pp. 241-247, 2010.

[40] C. Biancalana, F. Gasparetti, A. Micarelli, A. Miola and G. Sansonetti, "Context-aware movie recommendation based on signal processing and machine learning," *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation,* pp. 5-10, October 2011.

[41] "Insights to global consumers travel interests in 2014," TripBarometer by TripAdvisor, 2014.

[42] "Tripbarometer traveler respondents," TripBarometer by TripAdvisor, 2015.

[43] S. Brin, L. Page, R. Motwami and T. Winogard, "The PageRank citation ranking: Bringing order to the web," Stanford University, Computer Science Department, 1999.

[44] A. N. Langville and C. D. Meyer, Google's PageRank and beyond: The science of search engine rankings, Princeton, New Jersey: Princeton University Press, 2006.

[45] R. S. Wills, *When rank trumps precision: Using the power method to compute Google's PageRank,* Raleigh: Nort Carolina State University, Dept. of Mathematics, 2007.

[46] M. Işık and H. Dağ, "An effective recommender moder for e-commerce platforms," *IMISC'17 Management Information Systems Conference,* October 2017, unpublished.