ISSN 2229-5518

A Trust Based and Time Influenced Similarity Computation Method for Collaborative Filtering

Falguni Roy[1], Mohammad Nuruzzaman Bhuiyan[2]

Abstract— Recommender systems were designed with the aim of guiding and helping the users to cope up with the information overloading problem. Collaborative filtering based recommender systems are the most useful systems to retrieve important information from the information pool. The key of collaborative filtering is to define similar users and/or items using the user-item rating matrix and help the system to generate recommendations for the users. However, traditional collaborative filtering approaches face many shortcomings like as data sparsity, cold start, symmetric similar assumption and do not have any scope to trace on users' changing interest, causing system's low and inefficient performance. By aware of these shortcomings, we design a new trust based similarity algorithm to address the solution of symmetric similar assumption and ignoring users' changing interest issues. In this paper, we define users' trust relation by using implicit trust information between them. We use users' rating time to make the trust score dynamic while computing trust score. The experiments are done on the MovieLens and Epinions database by splitting 80% - 90% data as training set and rest of the data as test set data. The experimental results demonstrate that the proposed approach performs better than the existing trust-based recommendation algorithms in terms of accuracy and coverage. The proposed method produces approximately 9.50% less prediction error for MovieLens dataset and 0.63% less prediction error for Epinions dataset by dealing with the mentioned shortcomings.

Index Terms— Recommender System (RS), Collaborative Filtering (CF), Trust based Recommender System (Trust based RS), Content based filtering, Demographic based filtering, Cold-start, Data sparsity

---

1 Introduction

For the availability and ease to use, the Internet has become the biggest information pool that has ever existed. Information retrieval is an area of study that helps to design scalable algorithms for storage and retrieval of useful information from that vast information pool and reduces information overload. Recommender system (RS) is the most popular form of web information customization system, which uses some of the classical information retrieval techniques. It suggests items to the user by predicting the ratings that the user would give to that item. The process of judging ratings can be performed by either using heuristics or machine learning or both approaches. RS is mainly used in E-commerce and entertainment based websites with the aim to predict a set of interesting items which are most likely preferable by the users of the system. The most common and widely used examples of recommender system are Amazon.com, Netflix.com, MovieLens and so on. According to the filtering approach, there are five categories of RS and the approaches are: Content based filtering, Demographic based filtering, Collaborative filtering, Knowledge based filtering and Hybrid approach but among of them, collaborative filtering is the most used approach in RS and much efficient [1].

1.1 Collaborative Filtering (CF)

Collaborative filtering needs regular users' participation with an easy way to represent users' choices to the system and an algorithm that is able to compare people with their similar choices. Based on the methodology, CF can be categorized as either memory based or model based CF. Model based CF provides item recommendation by developing a model of users' ratings using different machine learning algorithms such as Bayesian network, neural network etc [4]. Memory based CF uses a rating matrix and provides recommendation for a specific item based on the relationship between the target user and others by applying some statistical techniques on the rating matrix [2]. Based on the process of how similarity between users is determined, memory based CF can be further divided as either user-based or item-based approach. By using trust relationship between users, user-based CF is categorized as trust based RS.

1.2 Trust Based Recommender System

Trust based recommender system is the next generation of memory based CF. It is used to overcome the limitations and uncertainties of the traditional CF, that are arisen for the presence of sparsity in the input data and treating users as equal similar without concerning their interests etc. It utilizes the trust of the ratings to provide recommendations for the users based on those people they trust. According to the sociological definition, trust requires a belief and an oral commitment. As a consequence, it is very difficult to define and model trust between users using a mathematical equation or computationally. In RS, trust is defined on the basis of personal background, history of interaction, similarity, trust statement etc. in the system. Guo et al. defines trust in RS as "Trust is defined as one's belief towards the ability of others in providing valuable ratings" [6]. According to trust theory, trust has four important distinct properties [5]:

- ***Asymmetry***. Trust is personal and varies with different

[1]*Institute of Information Technology; Noakhali Science & Technology University, E-mail: falguniroy.iit@gmail.com*
[2]*Institute of Information Technology; Noakhali Science & Technology University, E-mail: mdnuruzzaman2001@yahoo.com*

people with their different opinions. People might have different faiths on a particular person based on their experiences with that person. So if a person U_a trusts another person U_b, it is not assured that person U_b trusts person U_a to the same extent. Trust is direct and asymmetric.

- ***Transitivity***. Trust is transitive. It defines that if a person U_a trusts person U_b and U_b trusts person U_c then it can be inferred that person U_a could trust person U_c to some extent.
- ***Context Dependence***. Trust strongly depends on the context in which it has formed. It means if a person U_b, who is trustworthy in movies recommendation to the person U_a, may not be trustable in technology for the person U_a.
- ***Dynamicity***. Usually, trust is built in a continuous way which is gradually established and changed as the time going on with more experiences. It can be increased with positive experiences and decreased with negative experiences.

Based on how to filter trust, trust based RS is divided by either explicit or implicit trust filtering approach. The explicit trust filtering approach collects trust values from pre-existing social links between users [8], [9]. The pre-existing social links are defined as either "trust statement" or "web of trust". Many researchers have worked with explicit trust based RS and tried to improve RS's performance with its effectiveness [7], [10]. Although explicit trust based RS performs good and supports transitivity and asymmetric properties of trust but it suffers from few limitations on defining trust score and the limitations are: First, it requires additional manual labor and user effort to receive RS's service. Second, for simplicity and privacy issue, explicit trust scores are defined as binary format which also bound the users to express their degree of trust on others. Third, the amount of trust information is comparatively small and can be easily noisy in terms of users' choice.

On the other hand, implicit trust based RS alleviates these limitations because it extracts trust values between users based on the item ratings [3], [11], [12]. By analyzing rating patterns, rating values and historical behavior of ratings of the users, implicit trust is measured between the users and identifies reliable users whose ratings are useful for recommendation.

1.3 Motivation

RS applies some statistical analysis on the user-item rating matrix to define implicit trust. The statistical analysis are done by defining similarity between users, measuring users' rating difference etc. For these statistical analysis, various methods are used like Pearson correlation coefficient, amount of co-rated items etc. Such statistical analysis are done by either measuring the amount of co-rated items between a pair of users or by defining the absolute correlation of users' ratings. These analysis return same assumption for a couple of users. So based on these statistical analysis, both users are treated as same trustable which means the degree of trust between a couple of user is same. But according to trust literature [5], trust is asymmetric; it means that if a user U_a trusts another user U_b then U_b's trust on U_a will be not same in any extent and even it's not obvious that user U_b trusts U_a. But traditional implicit trust based RS treats both users as symmetrically trustable. On the other side, implicit trust based RS defines trust based on either co-rated items or co-relation of the items rating values without concerning the rating time of the items. Let's assume that user U_a rated an item *i* two years ago according to his/her that time's preferences and user U_b rated the same item two days earlier according to his/her current preferences. Now, if the ratings are very close or same, it does not necessarily mean that user U_a can trust on the rating of U_b and vice versa; because, it's natural that preference changes with time. It is formally claimed by trust literature [5] that trust is dynamic and it increases or decreases according to the ongoing experiences of the users. So traditional approaches also violate this common property of trust.

In this work, we propose a framework which considers trust, time and similarity in a single approach and deals with the existing problems of symmetric trustable users and constant trust definition. The proposed framework of this paper will use dynamic implicit trust among users for constructing trust based similarity method, which will improve prediction accuracy and reliability by treating similarity as asymmetric with the involvement of users' changing interest. The proposed method consists of several units where similarity and trust between users are determined; then both are combined to form a more effective and accurate similarity computation metric for finding out most similar neighbors of target user for executing future functions of the system.

1.4 Paper Contribution

The primary goal of the research is to define a single framework to support trust properties by dealing with the several issues which are mentioned in the previous section. In the context of the research, target user denotes a user for whom item rating prediction is being performed, where recommender user denotes the user whose data is being used to perform collaborative filtering. The contribution of the paper is four fold and they are described as follows:

- We have defined a time weighted function and modified Resnick's prediction method based on recommender user's rating time to penalize his/her ratings' if the ratings are too old.
- We have provided a method to learn the constant parameter, which is used in the time weighted function, based on the rating history of each user in the system. The constant is a person-wise constant which means the constant parameter will remain constant for a user for a span but it will vary from user to user.
- We have defined a single trust method to support maximum trust properties and those are asymmetry, transitivity and dynamicity to sort out existing problems, which are mentioned in the previous section.
- We have defined similar minded and trustworthy users by combining the degree of trust and similarity between them.

2 Related Work

User based CF algorithm is firstly proposed as a new format by Resnick which is a part of memory based CF. Resnick *et al*. considered the user based CF as a family of algorithms instead of a single algorithm [13]. They divided their entire approaches into three steps. In the first step, users' similarity is computed by using Pearson Correlation Coefficient (PCC). After that, similar minded neighborhood is selected by applying maximum number of neighbors' algorithm. In the last step, they computed prediction on the basis of weighting the correlation between neighbors. Usually PCC defines similarity based on the ratings value of each pair's co-rated items. So, if a pair of users have small amount of co-rated items with similar rating pattern, then PCC interpret them as highly similar without concerning the amount of individual rated items. To resolve each similarity measured method limitations, many researchers had suggested their methods as an integration of two of any similarity measured methods. Bobadilla *et al.* [14] proposed a new similarity measured method by integrating Jaccard and MSD method naming as JMSD and ensures better performance with higher accuracy in compare with other combined and singular methods like *Jaccard* × *PCC*, *Jaccard* × *COS, Jaccard* × *SRC, PCC* etc. Generally Jaccard does not grant too much credibility to the similarity between users based only on the similitude of a very limited set of common items and MSD defines users similarity based on numerical information like as users' rating difference and it generates very good general results with low average error, high percentage of correct predictions and low percentage of incorrect predictions. Their proposed method, JMSD combines each method's positive side by resolving each method's individual limitations. However, they did not address user's current rating behavior and determined users' similarity as identical.

Various researchers had suggested to integrate trust relationship between users with traditional CF system to minimize existing limitations. Some of them also worked with implicit trust based RS and proposed their trust metric with better performance by addressing existing limitations. For example, Qusai Shambour *et al.* proposed their trust method based on mean squared distance (MSD). Before defining trust between each pair of users, based on MSD, each time they analyzed predicted rating by considering both users as their sole recommendation partner. For predicting ratings, they used simple version of Resnick's prediction formula and then propagated trust, based on the MoleTrust matric [1]. Hwang *et al.* almost did the same thing by using a different approach [11] and addressed two types of trust metric and those are, local trust metric and global trust metric. For constructing local trust metric, they measured the trust score of their method by averaging the prediction error on co-rated items. Global trust metric is constructed by using global trust value which is produced by averaging the local trust values of all the trusted neighbors. Papagelis *et al.* defined their trust metric based on measuring similarity between users by using PCC method [15]. If the computed similarity is above threshold then they considered both users as trustable to each other. O'Donovan *et al.* proposed another implicit trust computation method in [3] and according to their method, a rating provided by a user is correct if the absolute difference between the predicted and the actual rating is smaller than a threshold. However, all these methodologies used users' rating information for trust computation and are transitive as well as asymmetric in some cases. However, these trust computation methodologies are not dynamic and/or context dependent [5], [16]. A user's trust on others could be increased through good experiences and decreased by negative experiences over time. But these methodologies do not pay any concern on the changes of user's trust and most of the cases, treat both users as equal trustable by each other which leads to low performance of trust based RS.

In this paper, our approach is to define the time weight at the beginning of the implicit trust construction for the memory based CF and ensure the constructed trust supports maximum property of trust; which means the computed trust score of our approach, is *asymmetric*, *transitive* and *dynamic.*

3 Proposed Method

The primary objective of this paper is to design an effective and reliable RS by integrating the dynamic trust computation into traditional CF process which is basically dependent on similarity computation. The proposed system consists of three units and the first unit, named as Similarity Computation (SC) unit, computes the correlation between users. The second unit, which is Trust Computation (TC) Unit, generates the trust score for each pair of users from rating matrix. The last unit, also named as Combined Trust and Similarity Computation (CTSC) unit, integrates the trust and similarity scores to produce actual similar user pairs on the basis of trust and similarity. The system takes user-item rating matrix as input and returns trust based similarity matrix. The rating matrix is $m \times n$ matrix, where m represents the set of users and n denotes the set of items which exists in the system. Each cell in the matrix denotes the rating of an item provided by a user. The trust based similarity matrix is $m \times m$ square matrix, where each cell denotes the amount of trust that exists between a couples of users in the system. Detailed process of trust and similarity computation are included in the following sections.

3.1 Similarity Computation (SC) Unit

In this unit, we extract a neighborhood of similar minded users for the targeted user U_a for whom we want to generate recommendations. For this purpose, we first determine the similarity between U_a and all other users U_b of the system where $U_b \in U = \{U_1, U_2, \ldots, U_n\}$ and $U_b \neq U_a$ and U denotes the set of all user of the system. Similarity is calculated by computing Pearson Correlation Coefficient (PCC) and Jaccard similarity [1] for a pair of users. Usually, PCC is used to measure the numerical similarity between users and it determines how similarly they rate the items. However, the major drawback of PCC is that it only considers the co-rated items' ratings between users but doesn't consider the amount of common and individual rated items by both user U_a and U_b. So, if a pair of users have limited co-rated items with similar rating pattern then PCC treats them as highly similar to each other. To resolve this drawback, Jaccard is introduced to take into account not only co-rated items between users but also the items that they rated individually. The basic idea here is that, two people are more similar if they have rated sufficient number of common items in comparison with their individually rated items.

$$PCC(a,b) = \frac{\sum_{i=1}^{I_a \cap I_b}(r_{a,i}-\bar{r}_a)\times(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i=1}^{I_a \cap I_b}(r_{a,i}-\bar{r}_a)^2}\times\sqrt{\sum_{i=1}^{I_a \cap I_b}(r_{b,i}-\bar{r}_b)^2}} \quad \ldots (1)$$

$$Jaccard\ (a,b) = \frac{I_a \cap I_b}{I_a \cup I_b} \quad \ldots (2)$$

Where $r_{a,i}$ and $r_{b,i} \in [1, 5]$ represent the ratings of target user U_a and recommender user U_b for item i respectively, while $\bar{r_a}$ and $\bar{r_b} \in [1, 5]$ represent the average rating of all items in the system that target user U_a and recommender user U_b rated separately. I_a and I_b represent the number of items that both users U_a and U_b rated individually and $I_a \cap I_b$ determines the number of items that are commonly rated by them. However, Jaccard also suffers a major drawback. Jaccrad treats users' similarity based on the amount of users' common and uncommon rated items without considering their rating pattern. If a user rates an item, denoting as his/her liking and another user rates same item by expressing his/her disliking, Jaccard treats that item as common rated item without paying attention on what they rate that item. After combining PCC and Jaccard method, we get another similarity method denoted as JPCC [1] that resolved each method's drawback. PCC calculates similarity within -1 to 1 and Jaccard defines similarity in 0 to 1 range so the range of JPCC is [-1, 1].

$$JPCC(a,b) = PCC(a,b) \times Jaccard(a,b) \quad \ldots (3)$$

3.2 Trust Computation (TC) Unit

Here, we propose a new method for determining the implicit trust as an integration of Mean Square Difference (MSD) and Confidence between users' ratings. MSD is used to define the degree of similarity between a pair of user U_a and U_b [1] and Confidence determines how much a target user U_a should rely on the ratings of user U_b [19]. This unit is divided into two sub-units, Computation of MSD and Confidence Determination. Trust computation unit takes the user-item rating matrix as input and calculates the direct implicit trust scores of every pair of users.

3.2.1 Computation of MSD

This section defines the trustworthiness of a recommender user U_b by measuring the prediction accuracy of that user in the past, to the target user U_a. For example, if recommender user U_b has delivered high accurate recommendations to target user U_a in the past, then user U_b should acquire high trustworthiness from user U_a. The trustworthiness of a pair of users is defined into two steps. In the first step, we predict target user U_a's ratings based on recommender user U_b's rating pattern by using Resnick's prediction equation [13] and then in the second step, we determine the prediction accuracy by using MSD [1].

3.2.1.1 Prediction Measure

At this step, each time the recommendation process is performed separately by using recommender user U_b as target user U_a's sole recommendation partner. To perform the recommendation process, we use Resnick's prediction method. However, most of the time both user rate same item at different time stamp. In some cases, their rating time difference of the same item is too high. But traditional Resnick's prediction method does not pay any concern on users' ratings time. It treats a pair of user as equally similar if they commonly rate a set of items with similar rating pattern though their rating time difference is high and at the meantime the recommender's preferences got changed. For this reason, at the time of performing prediction of items' rating, we try to penalize recommender's prediction effects on target user according to the time when the recommender rated the predicted item inspired by forgetting curve of the psychology [17], [18]. We use exponential decay function for penalizing prediction effects of a recommender according to his/her rating time. Based on the exponential decay function, if a predicted item is rated too early by the recommender, the recommender's role should be very low at the time of same item's rating prediction. For this reason, we modify Resnick's prediction method by using the decay function to compute the predicted rating and the modified equation is equation **4**. To maintain the calculated trust score's range [-1, 1], we use Max-Min Normalization method [20] to normalize each user's ratings and after that we predict U_a's rating based on U_b's rating.

$$p_{a,i} = \bar{r}_a + (r_{b,i} - \bar{r}_b)e^{-T\lambda} \quad \ldots (4)$$

In equation **4**, $r_{b,i} \in [1, 5]$ denotes the normalize rating of item i by the recommender user U_b, and $\bar{r_a}$ and $\bar{r_b} \in [1, 5]$ denotes average rating of both user U_a and U_b, respectively. $p_{a,i}$ denotes the predicted rating of item i for target user U_a and predicted by recommender user U_b. T denotes the time interval, $T = T_r - T_i$ where, T_r is the most recent rating time of recommender user U_b which means the last time that U_b assigned rating of an item and T_i defined the exact time when U_b rates the item i. λ is a person-wise constant parameter which defines the decay rate to control the decreasing rate of person's previous ratings.

3.2.1.2 Method for Learning Constant (λ)

Interest varies from user to user and specially depends on time. Even the same user's interest span could vary according to the type of things he or she likes. The span, for which a user's preference for a specific item lasts, is generally determined by his/her present choices; so the value of old ratings is questionable. On the other side, the some old ratings have different impact on different users. Thus, the importance of the old ratings is distinct for different users and depends on the nature of the interest's span of a user. The decay rate, is a constant rate, controls the effect of the old rated data based on the span of user preferences for items. It also varies from user to user according to individual span of interests. If a target user's preference lasts longer for the items, then the decay rate will be lower. For the lower decay rate, the importance of the old rated data will decrease slowly at the future taste determination time.

In this paper, λ is a personalized parameter denoting the decay rate which means it is constant for all previous rated items of a user for a span but it would vary from user to user. In this paper, we include λ as one of the parameter of our model and our contribution lies in learning λ. In order to learn λ, we take all the previous rated items of a user into account. Furthermore, we compute time interval T for each rated item as $T = T_r - T_i$, where T_r is the time of the most recent rating made by the user and T_i is the specific time when the same user rates item i. Then, we determine T_{median} from all the time intervals of the rated items for a specific user. Usually a regular user of the system rates a lot of items from the beginning to the present and the time interval of each item differs with other items. As a result, time intervals for all the items create a skew distribution and median value is better suited for skewed distributions to derive at central tendency. After determining T_{median} for each user, we define λ for each user as,

$$\lambda = \frac{1}{T_{median}} \quad \ldots (5)$$

For the lower value of T_{median}, the value of λ will be higher and for the higher value of λ, the old rated data's effect at the prediction time decay faster by treating the old rated data as less important information at the prediction time compared to recent rated items. On the other hand, for the higher value of T_{median}, the value of λ will be lower and for this reason the old rated data's effect at the prediction time decay slower by treating as the important information for the prediction. The higher limit of λ is "1" and the lower limit is "0". The step by step process of λ determination is shown in the following section using Algorithm **1**.

3.2.1.3 MSD: Define Prediction Accuracy

After calculating all predicted ratings of the set of co-rated items for a pair of user U_a and U_b, we use MSD to define the degree of similarity between them from the prediction error of co-rated items. To do this, first, we measure the total difference between each predicted and actual rating of the target user U_a for the co-rated set of items. Then, the degree of similarity is defined between the pair by subtracting total ratings difference from "1". If the predicted and actual ratings differences are low then it is assumed that user U_b has provided accurate prediction in the past for user U_a. So the degree of similarity between them will be high otherwise they treat as less similar or dissimilar users.

$$MSD(a,b) = 1 - \frac{\sum_{i=1}^{I_a \cap I_b} (p_{a,i} - r_{a,i})^2}{I_a \cap I_b}$$

$$= \frac{(I_a \cap I_b) - \sum_{i=1}^{I_a \cap I_b} ((\bar{r}_a + (r_{b,i} - \bar{r}_b) e^{-T\lambda}) - r_{a,i})^2}{I_a \cap I_b} \quad \ldots (6)$$

Here, $p_{a,i}$ specifies to the predicted rating of item i for target user U_a. $r_{a,i}$ and $r_{b,i}$ refer to the actual rating that user U_a and user U_b were given to the item i. $\bar{r_a}$ and $\bar{r_b}$ represent the average rating of all items in the system that individually rated. $I_a \cap I_b$ denotes the common items which are rated by both user. T denotes the time interval of user U_b for item i and λ is the constant parameter for user U_b.

Algorithm 1 : Leaning Constant Parameter (λ)

Input: A user-item rating matrix including users' rating time, $R_{m \times n}$
Output: A list of λ for the users, T_{user1} , T_{user2}, …, $T_{userN} \in T$

```
Begin
  U ← set of m users, I ← set of n users
  Tr ← 0, Ia ← 0
  list of T ← 0
  for each user Ua ∈ U do
      Tr = user Ua's recent rating time
      for each item i ∈ I do
          Ia = set of rated items of Ua
          if ( item i ∈ Ia ) then
              determine item i's rating time Ti
              calculate time interval, T = Tr - Ti
          end if
      end for
      define Tmedian = median value(list of time intervals of Ua)
      calculate λ for each user using Equation 5
  end for
End
```

3.2.2 Confidence Determination

Usually, the similarity based trust has some major drawbacks, which have been described in previous research works [11], [12]. There are cases like if both user rate a small amount of common items then according to MSD, they can even appear as highly trustworthy to each other. To resolve all of those drawbacks, we incorporate the confidence value in our method. According to [15], *Confidence expresses the reliability of the affiliation between the users based on the number of co-rated items and influenced when the amount of co-rated items are changed.* Higher confidence of user U_a on user U_b specifies that user U_b is highly reliable to user U_a in a sense that their co-rated items are a significant percent of the number of items rate by user U_b. The confidence of target user U_a for whom we will generate recommendation with respect to recommender user U_b is calculated using equation **7**. As it can be seen from the equation **7** that the confidence of U_a on U_b will not be same as the confidence of U_b has on U_a. *Confindence(b,a)* would be a com-

ISSN 2229-5518

pletely different value than *Confindence(a,b)*.

$$Confidence(a,b) = \frac{I_a \cap I_b}{I_b} \qquad \dots(7)$$

Where, $I_a \cap I_b$ determines the number of items that are commonly rated by both user and I_b denotes the total amount of items that recommender user U_b rates in the system.

3.2.3 New Implicit Trust Measuring Method

After determining MSD and Confidence between users, we combine these to measure implicit trust between them as stated in equation **8** which create a new trust metric. The range of the calculated implicit trust is [-1, 1]. The positive trust value of each pair determines their degree of trust and the negative value of trust defines their level of not trust each other.

$$Trust(a,b) = MSD \times Confidence$$

$$= \frac{(I_a \cap I_b) - \sum_{i=1}^{I_a \cap I_b}\left(\left(\bar{r}_a + (r_{b,i} - \bar{r}_b)e^{-T\lambda}\right) - r_{a,i}\right)^2}{I_a \cap I_b} \times \frac{I_a \cap I_b}{I_b}$$

$$= \frac{(I_a \cap I_b) - \sum_{i=1}^{I_a \cap I_b}((\bar{r}_a + (r_{b,i} - \bar{r}_b)e^{-T\lambda}) - r_{a,i})^2}{I_b} \qquad \dots(8)$$

The computational process of new implicit trust definition is described in Algorithm **2** and the computed implicit trust, generated from equation **8** supports the following properties:

- ***Transitivity***: Transitive property says that if user U_a trusts recommender user U_b and U_b trusts another user U_c in the system then it can be inferred that user U_a also could trust on U_c to some extent. The computed trust value of this method, is transitive because we could build indirect trust connection between users with this trust value by using any trust propagation algorithm such as *MoleTrust* [9] or *TidalTrust* [21].
- ***Asymmetry***: In reality, trust is a personal and subjective issue. So, two user who are involved in a trust relationship, might not trust each other to the same extent and it is a common phenomenon. In the proposed trust method, we use Confidence and MSD. Confidence is calculated by the amount of co-rated items between users divided by the amount of recommender's total rated items. As the amount of individual rated items is varied from user to user, so the Confidence between users will not be same. On the other hand, we calculate MSD by penalizing recommender's rating effects based on recommender's items' ratings time. To do so, we first predict item's rating for the target user U_a based on the recommender user U_b's penalize rating effects which are measured by considering recommender's items' rating time. After that by using that predicted rating, we calculate MSD. So the calculated trust score of the proposed method will not be the same for both users as the amount of individual rated items and the ratings time are differed from user to user. Although at the worst case, if the amount of individual rated items are same, still our method will support asymmetry property of trust on the basis of users' rating time. So for all cases, the calculated trust score of the proposed method is asymmetry.
- ***Dynamicity***: Usually trust is built in a gradual way and changed by the time going on with good or bad experience with the trusted user. Trust can be increased with good experience and decreased with bad experience. In our proposed method, we use recommender's items rating time at equation **8** for predicting items' ratings for a target user and these time effects do not make the calculated trust as constant for both users. So, the trust score will be changed as the time going on with the increasing experiences and the continuous participation of the users at the RS.

Algorithm 2 : Computation of Trust

Input: A user-item rating matrix including users' rating time, $R_{m\times n}$
Output: A user-user trust matrix, $T_{m\times m}$

Begin
$U \leftarrow$ set of m users, $I \leftarrow$ set of n users
$I_a \leftarrow 0$, $I_b \leftarrow 0$, $I_c \leftarrow 0$
for each item $i \in I$ **do**
  normalize each rating into [-1,1] range applying Max-Min Normalization
end for
for each user $U_a \in U$ **do**
  for each user $U_b \in U$ **do**
    if (U_a != U_b) **then**
      for each item $i \in I$ **do**
        predict user's U_a rating on the basis of user's U_b rating by using Equation ***4***
        I_a = *all items rated by* U_a
        I_b = *all items rated by* U_b
        $I_c = I_a \cap I_b$
      end for
      for each item $i \in I_c$ **do**
        sum the square difference of predicted and actual rating of user U_a
        $Trust_{a,b}$ = calculate trust score of U_a and U_b by using Equation ***8***
        $T[a][b] = Trust_{a,b}$
      end for
    end if
  end for
end for
End

3.3 Combined Trust and Similarity Computation (CTSC) Unit

In this unit, we integrate similarity and trust scores, to define similar minded trusted users, which are generated from previous two units, SC and TC units, between recommender user U_b and target user U_a. We use average of trust and similarity scores between a pair of user as we are defining similar minded and trusted users so both scores should equally effect at the definition of similar minded trusted users and the integrated method denoted as *TJPCC(a,b)* and it's range belongs to [-1, 1].

$$TJPCC(a,b) = \frac{JPCC(a,b)+Trust(a,b)}{2}$$
…(9)

We explore each component that are *JPCC(a,b)* and *Trust(a,b)* of *TJPCC(a,b)* by determining the prediction accuracy in the following section. We calculate each component's mean absolute error *(MAE)* and compare the *MAE* values with the *TJPCC(a,b)*'s *MAE* and reveal the performance accuracy of the proposed method.

4 Results and Discussion

4.1 Dataset Description

To evaluate the performance of our proposed framework, we used two most popular datasets, one of them is MovieLens (*http://grouplens.org/datasets/movielens*), which is developed by GroupLens and Internet Movie Database (IMDB) and another is Epinions which is collected from epinions.com (*http://www.epinions.com*).

MovieLens is also referred to as ML-1M and it includes 6,040 users and 3,952 movies with a total of 1,000,209 ratings. The range of ratings is [1, 5]. In ML-1M dataset, each person has rated at least 20 movies and each movie belongs to one of the 19 types. The density of the user-item matrix is 4.10%.

Epinions is a product and shop review site, where users can review items like as movies, books, and software and users can also assign items numeric ratings in the range [1, 5]. It consists of 22,166 users who have rated a total of 2, 96,277 items belonging to any combination of 27 categories. The total number of ratings is 9, 22,267 and the amount of cold-start items is more than 1, 00,000. The sparseness of the dataset is hence more than 99.99%.

4.2 Evaluation Metrics

To measure the accuracy of the recommendations by using our proposed method, we used two most popular evaluation metrics: Mean Absolute Error (MAE) and Coverage.

The MAE is the most widely used method to determine the accuracy of recommendations [13] and defined as the average of absolute deviations between the system's predicted rating against the actual rating assigned by the user for a set of items. A lower MAE value represents a higher recommendation accuracy. Given the set of actual/predicted pairs ($r_{a,i}$, $r_{p,i}$) for all the items (n) rated by users, the MAE is computed as:

$$MAE = \frac{\sum_{i=1}^{n_a} | r_{a,i} - r_{p,i} |}{n}$$
… (10)

The coverage evaluates the capability of a RS to provide recommendations. The coverage is calculated by the total predicted items (I_p) using the proposed methodology divided by the total number of items (n) which are available in the system. Coverage defines the percentage of items that the proposed methodology would able to predict in the system [22]. The equation of coverage calculation is:

$$Coverage = \frac{I_p}{n}$$
…(11)

4.3 Performance Evaluation

In order to test our RS and evaluate the performance of our method we conducted a series of experiments on the datasets. We used a different set of benchmark methods for evaluating our method's performance on MovieLens and Epinions dataset. In the following sections, we have discussed our method's performance evaluation.

4.3.1 Performance Analysis on MovieLens Dataset

For the experiment, we have divided the dataset with an approximate split ratio of 80:20. Based on the amount of users' rated items, we have considered 80% data as train dataset and rest of the 20% data are considered as test dataset. To compare our method's recommendation quality, we choose different traditional trust measured methods like TFS [1], O'Donovan-Trust [3], JMSD [14] and Resnick-UCF [13] as our benchmark methods.

Figure **1** shows the comparison of prediction accuracy of the proposed method with benchmark methods and demonstrates that the TJPCC recommendation approach achieves the better recommendation accuracy in all neighborhood sizes in the dataset, compared to other benchmark recommendation approaches.

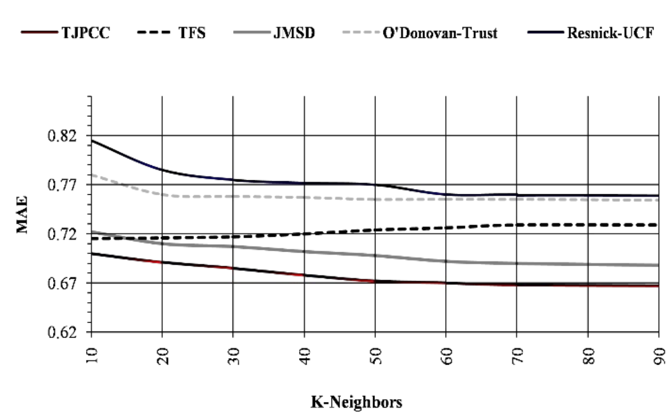

Figure 1: Comparison of TJPCC and other existing methods in terms of MAE on MovieLens dataset.

We also compare our proposed method's performance with the performance of each component of the method. According to the experiment, proposed method TJPCC preforms better than each of the individual components, which are aggregated in TJPCC using equation **9**. Figure **2** shows the MAE comparison between the proposed method with each component, JPCC and Trust that are used for TJPCC construction.

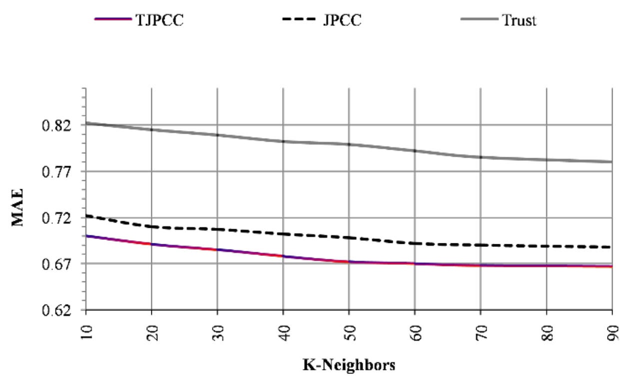

Figure 2: Comparison of TJPCC and its individual components (JPCC and Trust) in terms of MAE on MovieLens dataset.

Figure **3** shows the coverage comparison between the proposed and benchmark methods and the experiment demonstrates that the proposed method can improve the coverage for any neighborhood size in relation to other benchmark methods.

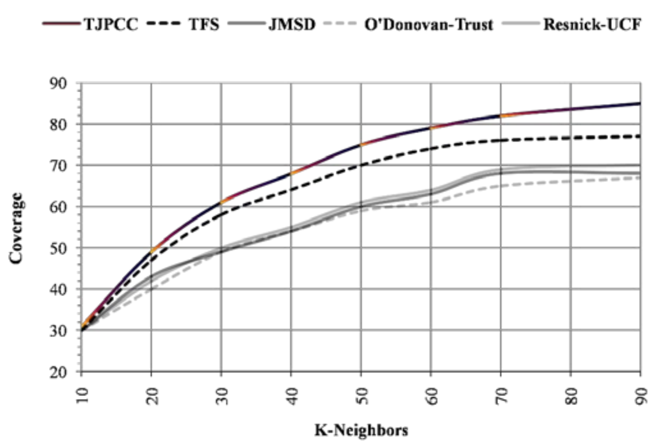

Figure 3: Comparison of TJPCC and other existing methods in terms of Coverage on MovieLens dataset.

4.3.1.1 Discussion of Prediction Accuracy in Term of MAE for MovieLens dataset

For the maximum number of neighborhoods, our proposed method ensures lower prediction error in compare with benchmark algorithms. Usually, prediction error is measured by the deviation of predicted and actual ratings of the user. Here, we shown that the percentage of MAE reduction level of the proposed method in compare with the benchmark algorithms in Table **1**. For percentage calculation, we used the following equation **12**:

$$IP = \frac{MAE_{BA} - MAE_{TJPCC}}{MAE_{BA}} \times 100 \quad \ldots$$

(12)

Here, *IP* denotes improved performance and *BA* determines benchmark algorithm.

Table 1. Comparison of Prediction Accuracy in term of MAE using maximum neighborhoods for MovieLens dataset.

Selecting All Users as Neighbors				
TJPCC (0.667)	TFS	JMSD	O'Donovan-Trust	Resnick-UCF
	0.729	0.688	0.754	0.759
	8%	3%	13%	14%

4.3.2 Performance Analysis on Epinions Dataset

For experiment, we have divided the dataset with an approximate split ratio of 90:10 where 90% data are considered as train dataset and rest of the 10% data as test dataset and to compare recommendation quality of our method's, we select different set of trust measured methods like TFS [1], TPCC [15] and Hwang-Trust [11] as benchmark methods for Epinions dataset.

Figure **4** shows the comparison of prediction accuracy of the proposed method with benchmark methods in terms of MAE. It exhibits that the TJPCC recommendation approach achieves the better recommendation accuracy in the Epinions dataset, compared to other benchmark approaches.

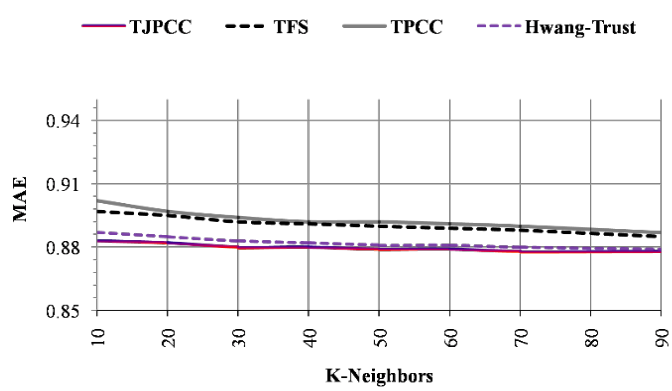

Figure 4: Comparison of TJPCC and other existing methods in terms of MAE on Epinions dataset.

For Epinions dataset, again we compare our proposed method's performance with each of the component of our method. According to the experiment, proposed method TJPCC preforms better in comparison with each of the single component (JPCC and/or Trust) of TJPCC. Figure **5** demostrates the MAE comparison between the proposed method with each component, JPCC and Trust that are used for TJPCC construction.

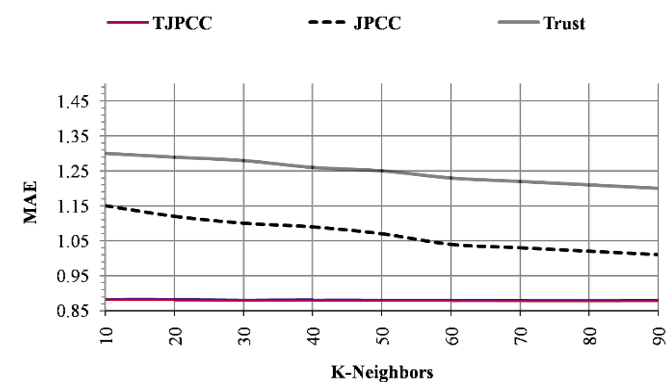

Figure 5: Comparison of TJPCC and JPCC and Trust in terms of MAE on Epinions dataset.

Figure **6** is showing the coverage comparison between the proposed and benchmark methods. Based on the experiment, the proposed method can improve the coverage for any neighborhood size in relation to other benchmark methods.

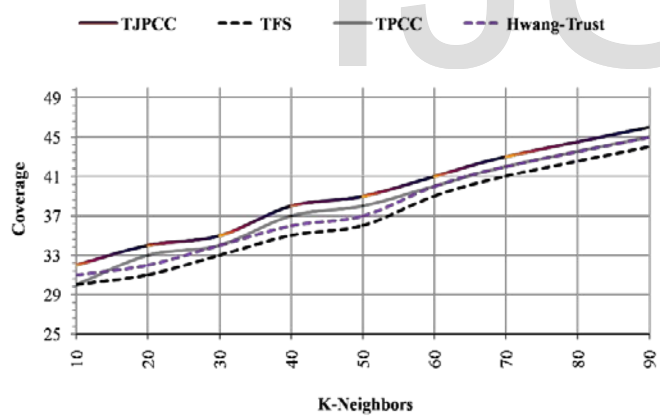

Figure 6: Comparison of TJPCC and benchmark methods in terms of Coverage on Epinions dataset.

4.3.2.1 Discussion of Prediction Accuracy in Term of MAE for Epinions dataset

Like as MovieLens dataset, we also shown the percentage of MAE reduction level of the proposed method in compare with the benchmark algorithms for Epinions dataset in Table **2**. For percentage calculation, we use same equation **12**:

Table 2. Comparison of Prediction Accuracy in term of MAE using maximum neighborhoods for Epinions dataset.

Selecting All Users as Neighbors			
TJPCC (0.878)	TFS	TPCC	Hwang-Trust
	0.885	0.887	0.879
	0.8%	1.0%	0.1%

4.4 Discussion

The proposed method deals with the following weakness of the existing trust based CF algorithms:

i. The computed trust score of our method supports asymmetric property of trust which means that the degree of trust between two users will not be same. As a consequence, the proposed method provides two different similarities for a pair of users based on the trust value that they have on other users.
ii. The proposed method uses the recommender's items' rating time at TC unit to pay concern on the recommender's changing interests and it gives more importance to the recommender's recent preferences compared to his/her old preferences at the time of trust computation, which will effect on similar users definition.

5 Conclusions

In this paper, we present a new user similarity computation method using dynamic implicit trust relationship. The performance of traditional implicit trust method suffers from two major issues. Firstly, it treats users as symmetrically trusted by each other and secondly, does not pay any concern on users' current preferences. In order to overcome these two issues, we propose a new trust based similarity method in our research work. We compare our method with some traditional algorithms and experimentally proved that our method ensure better prediction accuracy in term of mean absolute error (MAE) and coverage.

The main idea of this research work is to define precisely similar users by constructing dynamic trust relation. The items which were rated recently by a recommender should have a bigger impact on the contraction of the trust relationship with other users than the impact of an item that was rated long time ago. In order to apply users' item rated time at the trust construction, a time weight function is introduced in our method and the rated items' effect is controlled by a constant parameter. To learn the constant parameter, we observe user's rating behavior and identify appropriate personalized parameter for each user. We performed a series of experiments using data from the MovieLens and Epinions database. By comparing with different benchmark algorithms, we evinced that our method is able to improve performance approximately 8% - 14% for MovieLens dataset and 0.8% - 1% for Epinions dataset. However, we also investigated the efficiency of our metric TJPCC in another way. We have integrated similarity metric JPCC and Trust based metric and averaged them. We have shown that proposed TJPCC performs better than JPCC or the trust based metric individually. Hence we can conclude that aggregation of the two metrics was a fruitful approach.

REFERENCES

[1] Shambour, Qusai, and Jie Lu. "A trust-semantic fusion-based recommendation approach for e-business applications." *Decision Support Systems* 54, no. 1 (2012): 768-780.

[2] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. ACM, 2001.

[3] O'Donovan, John, and Barry Smyth. "Trust in recommender systems." In Proceedings of the 10th international conference on Intelligent user interfaces, pp. 167-174. ACM, 2005.

[4] Mannan, Noman Bin, Sheikh Muhammad Sarwar, and Najeeb Elahi. "A new user similarity computation method for collaborative filtering using artificial neural network." In International Conference on Engineering Applications of Neural Networks, pp. 145-154. Springer, Cham, 2014.

[5] Fazeli, Soude, Babak Loni, Alejandro Bellogin, Hendrik Drachsler, and Peter Sloep. "Implicit vs. explicit trust in social matrix factorization." In *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 317-320. ACM, 2014.

[6] Guo, Guibing. "Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems." In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 451-454. ACM, 2013.

[7] Massa, Paolo, and Bobby Bhattacharjee. "Using trust in recommender systems: an experimental analysis." In *International Conference on Trust Management*, pp. 221-235. Springer, Berlin, Heidelberg, 2004.

[8] Golbeck, Jennifer. "Generating predictive movie recommendations from trust in social networks." In *International Conference on Trust Management*, pp. 93-104. Springer, Berlin, Heidelberg, 2006.

[9] Massa, Paolo, and Paolo Avesani. "Trust-aware recommender systems." In *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17-24. ACM, 2007.

[10] Avesani, Paolo, Paolo Massa, and Roberto Tiella. "Moleskiing. it: a trust-aware recommender system for ski mountaineering." *International Journal for Infonomics* 20, no. 35 (2005): 1-10.

[11] Hwang, Chein-Shung, and Yu-Pin Chen. "Using trust in collaborative filtering recommendation." In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 1052-1060. Springer, Berlin, Heidelberg, 2007.

[12] Yuan, Weiwei, Lei Shu, H-C. Chao, Donghai Guan, Y-K. Lee, and Sungyoung Lee. "ITARS: trust-aware recommender system using implicit trust networks." *IET communications* 4, no. 14 (2010): 1709-1721.

[13] Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. "GroupLens: an open architecture for collaborative filtering of netnews." In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175-186. ACM, 1994.

[14] Bobadilla, Jesús, Francisco Serradilla, and Jesus Bernal. "A new collaborative filtering metric that improves the behavior of recommender systems." *Knowledge-Based Systems* 23, no. 6 (2010): 520-528.

[15] Papagelis, Manos, Dimitris Plexousakis, and Themistoklis Kutsuras. "Alleviating the sparsity problem of collaborative filtering using trust inferences." In *Trust management*, pp. 224-239. Springer, Berlin, Heidelberg, 2005.

[16] Guo, Guibing, Jie Zhang, Daniel Thalmann, Anirban Basu, and Neil Yorke-Smith. "From ratings to trust: an empirical study of implicit trust in recommender systems." In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 248-253. ACM, 2014.

[17] Ding, Yi, and Xue Li. "Time weight collaborative filtering." In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 485-492. ACM, 2005.

[18] Wang, Qian, Min Sun, and Cong Xu. "An improved user-model-based collaborative filtering algorithm." *J. Inf. Comput. Sci* 8, no. 10 (2011): 1837-1846.

[19] Kaleroun, Abhishek, and Shalini Batra. "Hybrid bee colony trust mechanism in recommender system." PhD diss., 2014.

[20] Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining: concepts and techniques, (the morgan kaufmann series in data management systems)." pp-230-240 (2006).

[21] Golbeck, Jennifer Ann. "Computing and applying trust in web-based social networks." PhD diss., 2005.

[22] Herlocker, Jon, Joseph A. Konstan, and John Riedl. "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms." *Information retrieval* 5, no. 4 (2002): 287-310.