Trust and Context-based Rating Prediction using Collaborative Filtering: A Hybrid Approach

Vineet Kumar Sejwal
Department of Computer Science
Jamia Millia Islamia, New Delhi, India
vineetsejwal.jmi@gmail.com

Muhammad Abulaish, *SMIEEE*[*]
Department of Computer Science
South Asian University, New Delhi, India
abulaish@ieee.org

ABSTRACT

In order to tackle the problem of *information overload* and effective recommendation based on users' preference, need, and interest a number of research contributions has been made for the development of recommender systems. However, certain challenges, such as *data sparsity*, *profiling attack*, and *black-box recommendation* still exist and hamper their prediction accuracy. In this paper, we propose a hybrid approach to predict user ratings by incorporating both trust and context of the users in traditional recommender systems using collaborative filtering method. The similarity between two users is computed using both trust value and context-based similarity. The trust value is based on three trust statements – *rating deviation*, *emotions*, and *reviews helpfulness*. On the other hand, context-based similarity is based on four contextual features – *companion*, *place*, *day*, and *priority*. The performance of the proposed trust- and context-based hybrid approach is analyzed using *mean absolute error* and *root mean square error* on a real dataset generated from two movie data sources (`IMDB` and `Rotten Tomatoes`), and it performs significantly better in comparison to some of the standard baseline methods. The rating prediction using only trust statements gives better results in comparison to other collaborative filtering approaches, such as user-based and item-based filtering approaches. Similarly, context-based collaborative filtering approach also outperforms standard collaborative filtering approaches. In addition, rating prediction using both trust- and context-based features performs better in comparison to only trust-based or context-based approaches.

CCS CONCEPTS

•Information systems → Recommender systems; •Human centered computing → *Collaborative Filtering;*

KEYWORDS

Recommender system, Collaborative filtering, Context-based recommendation, Trust-based recommendation, Trust network.

ACM Reference format:
Vineet Kumar Sejwal and Muhammad Abulaish, *SMIEEE*. 2019. Trust and Context-based Rating Prediction using Collaborative Filtering: A Hybrid Approach. In *Proceedings of 9th International Conference on Web Intelligence, Mining and Semantics, Seoul, Republic of Korea, June 26–28, 2019 (WIMS2019)*, 10 pages.
DOI: 10.1145/3326467.3326491

1 INTRODUCTION

There is plethora of information on the Web and users spend a good amount of time daily to search and find relevant information. The development of World Wide Web and infiltration of internet technology has attracted billions of new users who are generating huge volume of data, and consequently the issue of information overload is becoming severe problem. In parallel, a large number of e-commerce platforms and online social networks have emerged that too generate huge amount of data, providing newer research dimensions, such as human behavior analysis, big data analysis, recommender systems, and so on.

Recommender systems are software systems developed to provide recommendation to users and to handle the problem of information overload. These systems work as an information filter and provide recommendation to users based on their preferences, need, and interest. They are generally used in e-commerce platforms and different online social networking sites to recommend music, movies, news, books, friends, research articles, and so on. Accurate recommendation reduces users' workload and improves their usage experience. However, due to availability of huge amount of data, recommending relevant products to users is a challenging task. In order to improve the efficiency of the recommender systems, a number of data filtering approaches, such as content-based filtering, collaborative filtering, demographic (location-based) filtering, and hybrid (incorporating content and collaborative filtering) approaches have been proposed [7]. Among these filtering approaches, collaborative filtering (CF) is a successful and important approach used by most of the traditional recommender systems. The working of CF is to filter similar users and items based on their opinion, behavior, characteristics, and profile features, to the target users and items [25]. Nearest neighbor is a very frequently used approach in CF algorithms, where k-similar users (or items) represent the k most relevant users (or items) for a target user (or item). The development of recommender systems has several in-built issues, and most of the existing approaches target a specific issue, such as *cold-start*, *data sparsity*, *black-box recommendation*, rather than solving a set of issues in a single approach. Incorporation of users context and trust in recommendation process seems one of

[*] *Corresponding author*

WIMS2019, Seoul, Republic of Korea
© 2019 ACM. 978-1-4503-6190-3/19/06…$15.00
DOI: 10.1145/3326467.3326491

the promising approaches to tackle most of the issues faced by the traditional recommender systems.

Context-based recommender systems incorporate context as a dimension in existing traditional recommender systems [3]. Contexts are different constraints or conditions defined for a user, item, and user's decision [28]. Abowd et al. [1] defined context as "*any information that can be used to characterize the situation of an entity such as person, place, or object which is relevant in the interaction between the entity and an application, including the user and applications themselves*". According to Lieberman et al. [19] "*context can be considered to be everything that affects computation except the explicit input and output*". Cantador and Castells [8] defined context as "*the background topics under which activities of a user occur within a given unit of time*". Contexts are very helpful for finding exact requirements of a user from a rich amount of data, and thereby improving the prediction accuracy for recommendation. For example, finding travel destinations for a user u in Summer or Winter, watching movies with right companions, listening music during work or study, and visiting restaurants for office lunch or family dinner, all these are different actions performed by user with different entities. We use *place*, *day*, *companion*, and *popularity* as context dimensions. Each of these context dimensions can have different context values such as $\{$*home, theater*$\}$ for *place* context dimension, $\{$*weekend, weekday*$\}$ for *day* context dimension, and $\{$*family, friends*$\}$ for *companion* context dimension.

On the other hand, trust between two individuals refers to a condition in which one individual say u(trustor) has faith in another individual say v (trustee), based-on his/her actions and activities. According to Mayer et al. [23], trust can be defined as "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the truster, irrespective of the ability to monitor or control that other party". The trust between two individuals can be asymmetric, transitive, generic, and distributive. In trust-aware recommender systems (TARS), trust is computed between users using various trust statements and metrics and then incorporated with traditional recommender systems [13] [20] [21] [24]. A user's trust in a given trust network can be categorized into *local trust* and *global trust*. Local trust defines trust score between a user to all other users of the trust network as $T : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, whereas global trust defines the cumulative trust of each user based on all users of the network as $T : \mathcal{U} \rightarrow [0, 1]$. Another important concept in a trust network is *trust propagation*, which is very helpful for handling data sparsity problem. In a trust network, all users are not connected directly to each other, and trust propagation is used to calculate trust score for such users. *Tidal trust* and *mole trust* are two frequently used algorithms for computing trust propagation within a trust network [13] [21].

In this paper, we propose a hybrid approach incorporating both trust and context of users for rating prediction and recommendation. Incorporating trust into recommender systems can handle issues such as *data sparsity* and *profiling attack*, whereas context can be used to overcome *black-box* recommendation problem. Both trust value and context-based similarity are considered to computer similarity between the users. The trust value is calculated using three trust parameters – *rating deviation*, *emotion*, and *review helpfulness*, whereas context-based similarity is calculated using four contextual features – *companion*, *place*, *day*, and *priority*. The performance of the proposed hybrid approach is analyzed using *mean absolute error* and *root mean square error* on a real dataset generated from two movie data sources (`IMDB` and `Rotten Tomatoes`). It is also compared with some of the standard baseline methods and performs significantly better.

The rest of the paper is organized as follows. Section 2 presents a brief review of the state-of-the-art methods for trust-aware and context-based recommender systems. Section 3 a brief overview of the preliminary concepts. Section 4 presents details of the proposed trust- and context-based recommendation approach, including trust statements generation and context-based semantic similarity calculation. Section 5 presents our experimental setup and evaluation results. Finally, section 6 concludes the paper.

2 LITERATURE REVIEW

Golbeck [13] introduced the concept of trust relationship in a trust network by introducing FilmTrust, a website for movie recommendation. They proposed *Tidal Trust* algorithm for computing trust between the users who are not directly connected to each other in the trust network. In [20], authors proposed a trust-aware recommender system using collaborative filtering for improving the rating prediction accuracy. Their proposed work is a two-step process – first, filtering users based on their reputation score in the trust network using trust propagation, and Second, using trust metrics to improve quality assessment, which is one of the main issues in computing users' similarity. In continuation to this work, Massa and Avesani [21] proposed an approach to handle data sparsity problem in collaborative filtering-based recommender systems. An algorithm using trust metrics and *MoleTrust* was used to compute trust in the trust network. O'Donovan and Smyth [24] proposed trustworthiness as an important consideration for recommender systems. They defined item-level and profile-level trusts to handle recommendation errors. The item-level trust computes the correct predictions in terms of specific items, whereas profile-level trust provides correct predictions for user profile. In [17], *local trust* and *global trust* are computed for users' in the trust network, and experimental results show significant improvement in recommendation accuracy and coverage.

The recommendation approaches discussed so far are based on implicit trust inferred from user ratings of items. Guo et al. [15] analyzed five different existing trust metrics based on implicit trust and then proposed two metrics for trust ranking. These trust metrics are also capable for distinguishing explicit trust from implicit trust.

Context on the other hand, defines an entity situation when same action is repeated again and again [9]. When same action is performed multiple times, context dimensions may change with each action. A traditional recommender system is defined as $Traditional{-}RS : users \times items \rightarrow ratings$. Incorporation of context in an existing traditional recommender systems defines a context-aware recommender system as $Contextual{-}RS : users{\times}items{\times}context \rightarrow ratings$ [3]. A context-aware recommender systems (CARS) can use the algorithms like *contextual-prefiltering*, *contextual-postfiltering*, and *contextual modeling* [2] for filtering purpose. In *contextual-prefiltering* paradigm, first relevant data is selected based on contextual features and then rating is predicted based on the selected data. In *contextual-postfiltering* paradigm, first rating is predicted on available dataset and then filtering is performed using context of users. Finally, *Contextual modeling* paradigm directly uses contextual information in modeling for rating estimation. User splitting, item splitting, user-item splitting, and reduction-based approaches are some examples of contextual-prefiltering. Tensor factorization, and deviation and similarity-based sparse linear method are examples of contextual modeling. The hardest part in CARS is to find contextual features. Yao et al.[28] categorized context as *user context*, *item context*, and *decision context*. This gives an idea about different categories of contexts used in a CARS. In [16] Hariri et al. proposed a recommender system which incorporate user context mined from user reviews. The extracted context is then combined with user preferred ratings to define a recommendation function.

Though a good amount of work has been proposed for trust and context-based recommendation separately, to the best of our knowledge, no one has incorporated both trust and context together into a traditional recommender system. In this work, we proposed a hybrid approach for providing recommendation using both users contextual situations and their trust in a unified manner.

3 PRELIMINARIES

This section presents a brief overview of some of the basic concepts related to both trust and context. Starting with the discussion of trust, trust metrics (local and global), and trust propagation it discusses context and its various categories in the following subsections.

3.1 Trust

In social context, trust score between two users is a condition in which a user u (trustor) has faith in another user v (trustee), based on the actions and activities of v. Trust can be computed between individuals, families, communities, and organizations using various trust statements identified from user and item attributes. It can be *asymmetric*, *transitive*, *generic*, and *distributive* that are briefly described in the following paragraphs [6][14].

1. **Asymmetric**: Two users u and v hold asymmetric trust relation if first user u trusts on second user v, but converse is not true. i.e., $(uTv) \neq (vTu)$.
2. **Transitivity**: Trust can be transitive by nature, although it is not always true. On the basis of users relations and trust network, transitive trust between a pair of users can be inferred as $(uTv) \wedge (vTw) \implies (uTw)$.
3. **Generic**: Trust is not generic by nature in the sense that if users u and v trust each other for book recommendation, then it is not necessary that u will v or vice-versa on other domains like movie, hotel, and restaurant. In other words, trust is context dependent.
4. **Distributive**: Trust is not distributive by nature. In a trust network, if a user u trusts users v and w together, then this does not imply that u trusts v and w individually.

3.2 Local and Global Trust

In this section, we discuss two trust metrics namely *local trust* and *global trust* that can be used to determine trust scores among the users of a trust network.

3.2.1 *Local Trust.*

In a trust network, *local trust* is the trust score between an active user and all other users in the network. The trust value shows the level of trust other users have in the active user. *Local trust* is peer to peer in a network, and it is calculated for each user of the network with every other users of the network. It is local in the sense that it is calculated between every pair of users, rather than based on all users of the network. In a network of n users, local trust for each user u_i is calculated with all remaining $n-1$ users of the network. The *local trust* metric represents the trust score between a pair of users as $T : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, where $\mathcal{U}$ is the set of users in the trust network, and trust score between a pair of users is in the range of $[0, 1]$.

3.2.2 *Global Trust.*

In a *global trust* metric, every user receives a single trust score in the trust network and represents its trustworthiness score in that network. It is an aggregation of opinions in the form of trust from all other users to the target user. *Global trust* metric represents the trust score of a user in trust network as $T : \mathcal{U} \rightarrow [0, 1]$. For computing *global trust* in terms of *profile-level trust*, each user is assigned a single trust score, whereas global trust for *item-level trust* is the user's trust score assigned with respect to the items.

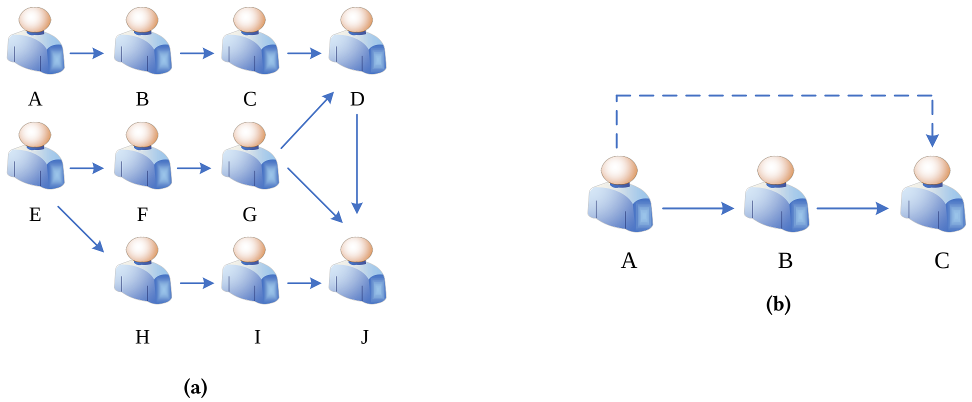
Local trust metric requires more computing as compared to *global trust* metric because in *local trust* each pair of users is evaluated to assign a trust score between them. However, *local trust* is more robust against the fake profiling attack and controversial topics. Additionally, *local trust* represents user's interest more accurately. However, consideration of local or global trust metric depends on the domain, topics, and trust network.

3.3 Trust Propagation

Trust propagation algorithms calculate trust scores for those users where there is no direct path to connect them with other users in the trust network. As defined earlier, trust network exhibits transitive relation between users, and thereby this type of trust is also known as atomic direct propagation. However, trust propagation between users based on transitive relation is not always true; it is not necessary that if (aTb) and (bTc) then (aTc). Therefore, it can be realized that trust is subjective by nature [18]. In large trust network, there could be more than one path to connect any two random users. In such cases, only propagation is not sufficient for trust computation between users, rather we need to aggregate different propagating trust values for such user pairs. Therefore, depending on trust network and requirement, both propagation and aggregation trust metrics are used for trust distribution. Figures 1(a) and 1(b) show trust aggregation and propagation mechanisms in an exemplar network. Figure 1(b) shows single path from user A to C via B. Therefore, we calculate trust between A and C using trust propagation. However, in order to calculate trust between users E and J in figure 1(a), all three different paths are used for trust computation, and trust scores are aggregated to determine the final trust value between E and J.

3.3.1 *Mole Trust.*

Mole trust [22] is a *local trust* propagation metric, which is used to find trust score between user pairs. It works as a backward exploration. For a user pair, a walk is established between the users through the trust edges connecting them, and



(a)

(b)

Figure 1: (a) Trust aggregation (b) Trust propagation

equation (1) is used to find *mole trust* between the user pair in a trust network. In this equation, $t(k)$ is the trust score for user k and $edge(k, u)$ is the trust edge. All incoming trust edges towards the target user is analyzed and only those edges having trust score greater than a threshold value are considered. For example, figure 2 presents a small trust network to estimate trust value for a user pair i and u, based on the threshold value 0.20. Using equation (1), the mole trust between i and u can be calculated as $\frac{0.75*0.62+0.82*0.45}{0.75+0.82} = 0.5312$. It may be noted than the path between i and u via user E is not considered because trust score between user i and user E is below the threshold value.

$$t_u = \frac{\sum_{i\in predec}(t(k) * edge(k, u))}{\sum_{k\in predec} t(k)} \tag{1}$$

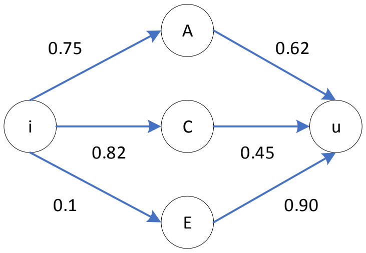


Figure 2: Propagation of mole trust in a trust network

3.3.2 Tidal Trust. Golbeck [13] proposed *tidal trust* model for movie recommendation. In comparison to *mole trust* where trust representation is binary, *tidal trust* represents gradual trust among users. In order to compute trust using *tidal trust*, first all possible paths from active user to other users are explored; then the shortest path among all paths is selected. Each trust path follows two properties – length and strength. Equation (2) defines the tidal trust computation in a trust network formally.

$$t_{i,s} = \frac{\sum_{k\in adjacent(l)|t_{l,k}\geq max} t_{l,k}t_{k,s}}{\sum_{k\in adjacent(l)|t_{l,k}\geq max} t_{l,k}} \tag{2}$$

3.4 Context in Recommender System

As explained in [10], context in a recommender system can be of two types – representational view context and interactional view context. In representational view context, set of observable attributes that define contextual features are known. These attributes are static in nature and their structure do not change with time. On the other hand, interactional views are dynamic in nature and contextual attributes are not known. The incorporation of context in an existing two dimensional recommender system can be achieved using the context-based filtering algorithms defined in [2]. In *contextual prefiltering*, data is first filtered using the context values, and then it is used for rating prediction using the classical recommendation technique. In *contextual postfiltering*, first rating prediction is completed and then contextual information is used for filtering. Finally, in *contextual modeling*, contextual values are directly used in modeling techniques for rating prediction.

Table 1: A user-item interaction based on ratings and contextual features

User	Item	Rating	Time	Location	Companion
u_1	i_1	9	weekday	home	kids
u_2	i_2	8	weekend	home	alone
u_3	i_1	8	weekday	cinema	family

Table 1 presents a snippet of a movie recommendation with three users $u1$, $u2$, $u3$, two movie items $i1$, $i2$, and three contextual dimensions – *time*, *location*, and *companion*. Each contextual dimension can have multiple contextual conditions, which are the possible values for each contextual dimension. For example, possible contextual values for context dimension "time" are "weekend" and "weekday".

3.5 Types of Context

This section presents a brief description of different types of context that are generally used in recommender systems. As mentioned in [29], context can be categorized as *user context*, *item context*, and *decision context*. A brief details about these contexts is presented in the following sub-sections.

3.5.1 *User Context.*

User context represents the contextual features, such as *age*, *gender*, *friends*, etc. of the users for their characterization. The context of a user u is denoted by U_c and represented as the set of feature-value pairs.
For example, $U_c = < Gender : Male >, < Friends : Bob, Alice >$ provides contextual information about user u whose gender is male and have Bob and Alice as friends.

3.5.2 *Item Context.*

Item context represents the contextual features of items for their characterization. For example, *genre*, *subgenre*, and *cast performance* can be considered as contextual features of movie items. The context of an item I is denoted by I_c and represented as the set of feature-value pairs.
For example, $I_c =< Genre : Sci - Fi >, < Director : RidleyScott >$ provides contextual information about item I, which has Sci-Fi genre and Ridley Scott as the director.

3.5.3 *Decision Context.*

Decision context represents the contextual features, such as *location*, *companion*, and *time* that are used by the users for consumption of items. Based on user and item context values, decision context dimensions also change. The context of a decision D is denoted by D_c and represented as the set of feature-value pairs.
For example, $D_c = < Location : Home >, < Companion : Kids >$ represents a user who watched a movie item at home with his/her kids.

4 PROPOSED APPROACH

This section presents a detailed description of our proposed approach to incorporate user trust and context for rating prediction. Figure 3 shows the work-flow of our proposed approach, which starts with the construction of trust network to extract contextual features. Thereafter, similarity is calculated between every pair of users based on trust and context-based features. Finally, rating is predicted using users' similarity values. A detailed description of these steps is presented in the following sub-sections.

4.1 Trust Network Construction and Trust-Based User Similarity Calculation

This section presents the construction of users' trust network using three trust statements derived from user ratings, user emotions, and review helpfulness, which are further described in the following paragraphs.

Rating-Based Deviation. The first *trust statement* for finding trust between a user pair a and u, where a is an active user, is based on their rating deviations. The idea behind this trust statement is to infer trust between a user pair based on their rating behavior. To this end, a user-item rating matrix is constructed, in which each rating value represents user interest in the respective item. Equations (3) and (4) compute the trust scores of a and u, respectively, where $\mathcal{R}$ represents the average rating of an item i, r_{ai} is the rating given by user a on item i, and $\mathcal{N}$ is the rating scale. The deviation between $\mathcal{R}$ and u represents trust value of u on item i. Since user pair a and u gives rating r_{ai} and r_{ui} on multiple items, their ratings information can be used to determine the trust score between them.

$$\mathcal{T}^{\delta}_{ai} = \begin{cases} \frac{-1}{\mathcal{N}}|\mathcal{R} - r_{ai}| + 1, & if\, r_{ai} \neq 0 \\ 0, & otherwise \end{cases} \tag{3}$$

$$\mathcal{T}^{\delta}_{ui} = \begin{cases} \frac{-1}{\mathcal{N}}|\mathcal{R} - r_{ui}| + 1, & if\, r_{ui} \neq 0 \\ 0, & otherwise \end{cases} \tag{4}$$

The difference in the rating deviation of a and u represents the trust score between them, as given in equation (5). With increasing rating deviation between a user pair, their trust score decreases. It can be observed from this equation that trust score will be high for users who will give nearly similar ratings for identical items. Equation (5) defines the trust score between a and u based on n items rated by them.

$$\mathcal{T}^{\delta}_{a,u,n} = \frac{\sum_{j=1}^{n}(1 - |\mathcal{T}^{\delta}_{aj} - \mathcal{T}^{\delta}_{uj}|)}{n} \tag{5}$$

Emotions-Based Information. The emotion-based information, such as *anger*, *joy*, *happiness*, and *sadness* can be used to predict the trust/distrust relations between an active user a and another user u. Many sociologists and psychologists in their research work have shown that emotions are strong indicator of trust/distrust between users [11] [5]. textitJoy, *happiness*, and *satisfaction* are used as positive emotions representing trust, whereas *fear*, *anger*, and *sadness* are used as negative emotions representing distrust [4]. The emotions have the tendency to reduce data sparsity effects in trust/distrust prediction [4][5]. On e-commerce platforms, users rate and review various items, and these ratings along with the emotions extracted from reviews are highly correlated. In case of high ratings, positive emotion values are high, whereas in case of low ratings, negative emotion values are generally high, representing the extreme values of rating. As a result, rating deviation in both of the cases is high.

Let $\mathcal{U} = \{u_1, u_2, u_3, \ldots, u_n\}$ be the set of n users who have given ratings $\mathcal{R} = \{r_1, r_2, r_3, \ldots, r_k\}$ to the items $\mathcal{I} = \{i_1, i_2, i_3, \ldots, i_m\}$ such that $\mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$.

We define an emotion vector $\varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5]$, where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$ represent *anger*, *joy*, *sadness*, *fear*, and *disgust*. Equation (6) can be used to compute trust between the users pair a and u based on emotions, where $\mathcal{T}^{\varepsilon}_{a,u,n}$ represents the emotion-based trust score, $\mathcal{D}(a_j(\varepsilon), u_j(\varepsilon))$ represents the deviation between a and u based on their emotion vector ε on n items.

$$\mathcal{T}^{\varepsilon}_{a,u,n} = \frac{\sum_{j=1}^{n}(1 - \mathcal{D}(a_j(\varepsilon), u_j(\varepsilon))}{n} \tag{6}$$

Review Helpfulness. Review helpfulness can be used to find users reputation, which later can be used to infer trust. In e-commerce platforms and movie review websites, users express opinions on various aspects of the items in the form of reviews, which other users may like or dislike. Thus, users opinion can be evaluated by other users by voting them as *helpful* or *not helpful.* In addition, *review helpfulness* also provides an idea about the quality of the review. Many websites, such as *Amazon*, *IMDB*, *eBay*, *Rotten Tomatoes*, and *Flipkart* provide a summary of users' votes on a review in the form, e.g., "45 out of 56 people find this review helpful". In addition, *review helpfulness* can be used to compute the *reliability*, *honesty*, and *trust* of a user. In [12], authors proved that reviewers'

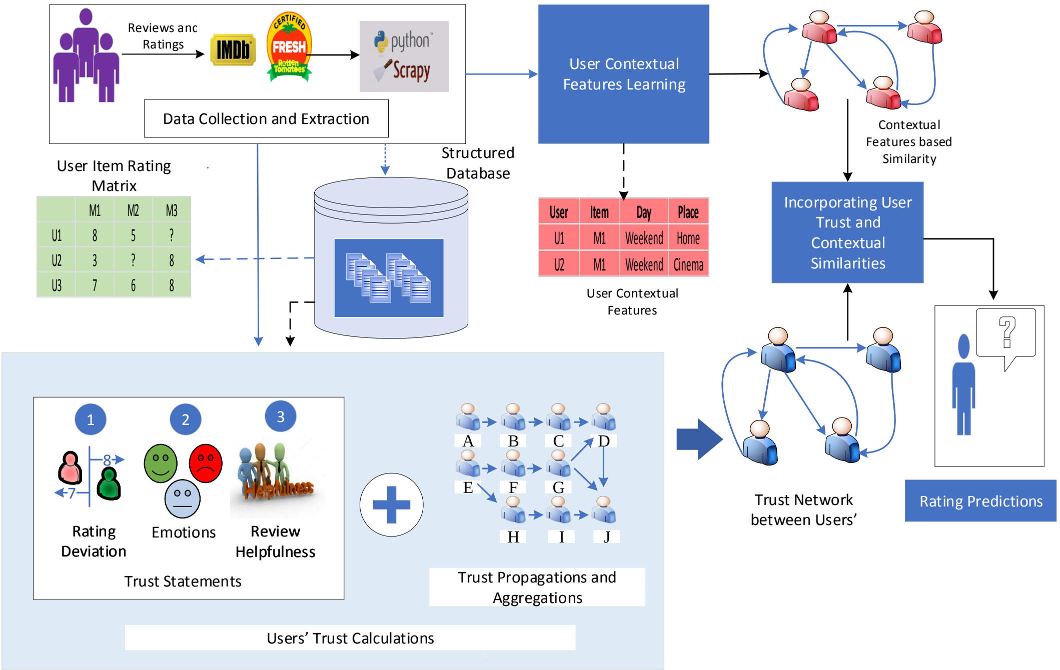


Figure 3: Work-flow of the proposed hybrid approach for rating prediction using context and trust

reputation and *review helpfulness* are correlated and they determine the *reliability*, *honesty*, and *trust* of the reviewers. Therefore, we used *review helpfulness* as the third trust statement to find *trust* among users. We used a variant of the trust function defined by Wang et al. in [27] to find the *trust* score between users based on *review helpfulness* votes. It is defined in equation 7, where $\mathcal{T}_u^{\mathcal{H}}$ represents *trust* score of user u calculated from review helpfulness $\mathcal{H}$, K is the upper bound of *trust* score (here we used $K = 1$) and V is the *vote skewness* score. The *vote skewness* score is calculated using equation 8, where f_{ip} and f_{in} are the positive (helpful) and negative (not helpful) votes given by the users on i^{th} review of user u, and N is the total number of reviews given by u. For example, a user u with voting summary 3/10 and 97/130 for two reviews has received 3 and 97 positive votes, and 7 and 33 negative votes. Therefore, the voting skewness of u will be 100 - 40 = 60. Skewness can be positive or negative. The *trust* score between an active user a and another user u based on *review helpfulness* can be calculated using equation (9). Finally, the aggregate *trust* score between the active user a and another user u based on n items and using *rating deviation*, *emotions*, and *review helpfulness* can be computed using equation (10).

$$\mathcal{T}_u^{\mathcal{H}} = \frac{K}{1 + e^{-KV_u}} \tag{7}$$

$$V_u = \frac{\sum_{i=1}^{m} f_{ip} - \sum_{i=1}^{m} f_{in}}{N} \tag{8}$$

$$\mathcal{T}_{a,u,n}^{\mathcal{H}} = \frac{1 - |\mathcal{T}_a^{\mathcal{H}} - \mathcal{T}_u^{\mathcal{H}}|}{n} \tag{9}$$

$$\mathcal{T}_{a,u,n} = \frac{\mathcal{T}_{a,u,n}^{\delta} + \mathcal{T}_{a,u,n}^{\varepsilon} + \mathcal{T}_{a,u,n}^{\mathcal{H}}}{3} \tag{10}$$

4.2 Context-Based User Similarity Calculation

This section presents details about context-based user similarity, which is derived from users' ratings and different contextual features. We have used reviewers' timestamps and reviews for contextual features. The contextual features (aka *relational context*) are extracted from users reviews and associated meta-data. Contexts are various possible conditions that describe user requirements towards the consumption of items. The incorporation of context in an existing recommender system can be useful in terms of accuracy, prediction, and recommendation. Context-aware recommender systems (CARS) can have multiple contextual dimensions, thereby making it a multidimensional system, where each dimension (aka contextual variable) can have one or more contextual conditions.

Table 2: A sampler user-item interaction based on ratings and contextual features

User	Item	Rating	Time	Location	Companion
u_1	i_1	9	weekend	home	kids
u_2	i_1	8	weekend	cinema	friends
u_3	i_2	8	weekday	cinema	family
u_3	i_2	?	weekend	cinema	alone
u_2	i_1	3	weekday	cinema	kids

Table 2 presents an exemplar user-item interaction based on contextual dimensions and values. It contains three users u_1, u_2, and u_3, two items i_1 and i_2, and three contextual dimensions – *time* (weekday or weekend), *location* (cinema or home), and *companion* (alone, kids, family). It can be observed from this table 2 that rating of each item by same users changes according to the contextual dimensions and values. For example, user u_2 consumed items i_1 and i_2 at different *time* with different *companion* and gives rating 8 and 3. It can be assumed that user u_2 likes item i_1 *alone*, but not with *kids*. The contextual situation for two users, say, u_1 and u_2 can be represented by sets of different contextual conditions c_k and c_m. Further, l represents the l^{th} contextual conditions for set c_k and c_m as $c_{k,l}$ and $c_{m,l}$. For example, in case of random contextual values from table 2, c_k is {weekday, home, kids}, whereas c_k at $l = 3$ is $c_{k,3}$ = "kids".

In order to find similarity between a user pair based on their contextual conditions, it is necessary to compute similarity on same contextual dimensions. For example, similarity should not be measured on "time = weekday" and "location = home", as both contextual conditions belong to different contextual dimensions. Zheng et al. [31] proposed a method to compute context-based user similarity, which is defined in equation (11). However, their method has certain limitations that were later overcomed by *Latent Context Similarity* (LCS) and *Multidimensional Context Similarity* (MCS) methods. However, these methods too suffer from *data sparsity* problem [30]. For example, based on the data given in table 2, if we want to predict rating of user u_3 on item i_2 with respect to the contextual conditions set $\{weekend, cinema, alone\}$, we need to find users that have rated item i_2 exactly with same contextual values. Hence, such strict conditions generate *data sparsity* problem because no such user exists in table 2.

$$sim(c_k, c_m) = \prod_{l=1}^{L} similarity(c_{k,l}, c_{m,l}) \tag{11}$$

In order to handle such type of strict situations in CARS, Zheng et al. [30] proposed Differential Contextual Weighting (DCW) scheme where weights are assigned to the matched context values in c_k and c_m so that finding exactly similar context values is no longer required. The DCW weight assignment controls the contribution of each context variable towards ratings prediction. Therefore, more similar the context values are in c_k and c_m, more helpful it would be to predict the accurate ratings. Equation (12) can be used to assign weight to the matched context values in c_k and c_m.

$$\mathcal{W}(c_k, c_m, w) = \frac{\sum_{j \in c_k \cap c_m} w_j}{\sum_{j \in c_k \cup c_m} w_j} \tag{12}$$

The similarity score between every user pair can be calculated using Pearson Correlation Coefficient (PCC), as given in equation (13). However, in this paper, we have modified equation (13) to incorporate weight-based context similarity, and the same is presented in equations (14) and (15). Finally, contextual features-based similarity for each user pair is calculated using equation (15).

$$sim(a,u) = \frac{\sum_{i \in I_{a,u}} (r_i(a) - \overline{r}(a)(r_i(u) - \overline{r}(u)}{\sqrt{\sum_{i \in I_{a,u}} (r_i(a) - \overline{r}(a))^2} \sqrt{\sum_{i \in I_{a,u}} (r_i(u) - \overline{r}(u))^2}} \tag{13}$$

$$sim_{a,u}^{context} = \frac{\sum_{i \in I_{a,u}} (r_i(a) - \overline{r}(a)(r_i(u) - \overline{r}(u) \mathcal{W}(c_k, c_m, w)}{\sqrt{\sum_{i \in I_{a,u}} \mathcal{W}(c_k, c_m, w)(r_i(a) - \overline{r}(a))^2 \sum_{i \in I_{a,u}} \mathcal{W}(c_k, c_m, w)(r_i(u) - \overline{r}(u))^2}} \tag{14}$$

$$sim_{a,u}^{context} = \frac{\sum_{i \in I_{a,u}} (r_i(a) - \overline{r}(a)(r_i(u) - \overline{r}(u) \mathcal{W}(c_k, c_m, w)}{\sqrt{\sum_{i \in I_{a,u}} (r_i(a) - \overline{r}(a))^2 \sum_{i \in I_{a,u}} (r_i(u) - \overline{r}(u))^2 \sum_{i \in I_{a,u}} \mathcal{W}(c_k, c_m, w)^2}} \tag{15}$$

4.3 Rating Prediction using Trust and Context

This section presents the rating prediction of an item for a particular user, incorporating both trust and context and using user-based collaborative filtering. To this end, we have used harmonic mean of the incorporated *context-based similarity* and *trust score* between users a (active user) and u, as given in equation (16). Harmonic mean shows robustness in comparison to other similarity measures for extreme input values. As a result, both trust score and context-based similarity score should be high to achieve high similarity weighting.

$$w(sim_{a,u}^{context}, \mathcal{T}_{a,u,n}) = \begin{cases} \frac{2 \times sim_{a,u}^{context} \times \mathcal{T}_{a,u,n}}{sim_{a,u}^{context} + \mathcal{T}_{a,u,n}} & if, \mathcal{T}_{a,u,n} + sim_{a,u}^{context} \neq 0 \\ sim_{a,u}^{context} & elseif\, sim_{a,u}^{context} \neq 0 \; and \; \mathcal{T}_{a,u,n} = 0 \\ \mathcal{T}_{a,u,n} & elseif\, sim_{a,u}^{context} = 0 \; and \; \mathcal{T}_{a,u,n} \neq 0 \\ 0 & else \end{cases} \tag{16}$$

In order to predict the rating of an item (say i) by an active user (say a), top-k users similar to a are identified using equation (16). Thereafter, for each top-k users, the similarity values are combined with i's ratings (if available) using equation (17) given by [26] to predict the rating of a on i ($\hat{r}_{ui}$). In this equation, U_u^n represents the set of n most similar users to u.

$$\hat{r}_{ui} = \frac{\sum_{j \in U_u^n} w_j(sim_{a,u}^{context}, \mathcal{T}_{a,u,n}) r_{uj}}{\sum_{j \in U_u^n} |w_j(sim_{a,u}^{context}, \mathcal{T}_{a,u,n})|} \tag{17}$$

It is possible that some users give high or low rating on certain items due to biasness or critical nature, which may affect the rating prediction. In order to overcome such biasness, the authors in [26] proposed the incorporation of a first order approximation, as given in equation (18), where $b_{ui} = \mu + b_u + b_i$, μ represents the overall average rating of items in the dataset, b_i and b_u are deviation of user and item ratings with respect to average rating μ, and r_{uj} is the rating value given by user u for item j .

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in U_u^n} w_j(sim_{a,u}^{context}, \mathcal{T}_{a,u,n}) r_{uj}}{\sum_{j \in U_u^n} |w_j(sim_{a,u}^{context}, \mathcal{T}_{a,u,n})|} \tag{18}$$

Table 3: Dataset description

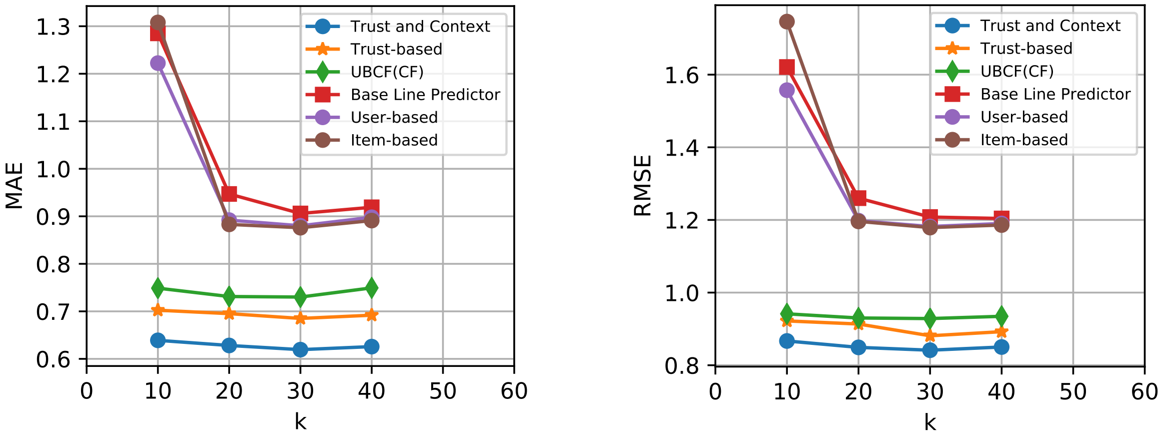
Category	Freq. count	Source
Movies	1102	https://www.rottentomatoes.com
Reviewers	78356	http://www.imdb.com
Reviews	174576	http://www.imdb.com
Critics	5650	http://www.rottentomatoes.com
Ratings	160606	http://www.imdb.com

Table 4: *MAE* and *RMSE* values for the proposed trust and context-based rating prediction and recommendation method

	k=10		k=20		k=30		k=40	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Trust and Context-based	0.6390	0.8668	0.6282	0.8492	0.6194	0.8414	0.6258	0.8502

Table 5: A comparative analysis of the proposed trust and context-based rating prediction and recommendation method with existing baseline methods

	k=10		k=20		k=30		k=40	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Baseline Predictor	1.285	1.621	0.947	1.260	0.906	1.208	0.919	1.204
Item-based Collaborative Filtering	1.308	1.746	0.883	1.196	0.876	1.179	0.891	1.186
User-based Collaborative Filtering	1.224	1.5572	0.892	1.198	0.8801	1.182	0.898	1.1902
UBCF$_{CF}$	0.7488	0.9415	0.7312	0.9301	0.7301	0.9285	0.7494	0.9347
Trust-based	0.7024	0.9220	0.6952	0.9135	0.6851	0.8811	0.6919	0.8924
Trust and Context-based	0.6390	0.8668	0.6282	0.8492	0.6194	0.8414	0.6258	0.8502



(a)

(b)

Figure 4: (a) MAE values for the proposed and baseline methods at different k values (b) RMSE values for the proposed and baseline methods at different k values

5 EXPERIMENTAL SETUP AND RESULTS

This section describes the experimental setup and results to establish the efficacy of the proposed trust and context-based rating prediction and recommendation approach. As mentioned earlier, we used data extracted from two well-known data sources in movie domain – Rotten Tomatoes and IMDB. To measure the performance of the proposed trust and context-based hybrid approach, we used MAE and RMSE as the evaluation metrics.

5.1 Dataset

For experimental evaluation and results, we implemented a crawler in Python to retrieve user data such as, name, reviews, review timestamps, user ratings, and review helpfulness, and movie data such as, genre, certification, theater release date, dvd release date, and cast members from two movie data sources Rotten Tomatoes and IMDB. Table 3 presents a brief description of our movie dataset.

As shown in table 3, a total number of 1102 movies with user reviews, critic reviews, movie features, and user ratings were retrieved. At IMDB, each movie is rated by the users on a 10-point scale, with 1 representing the lowest rating and 10 representing the highest rating. User reviews along with user ratings and review timestamps were retrieved and stored in a structured format. The extracted user and movie features are used for constructing a trust and context-based network where users are connected to each other with some weights calculated as a function of their trust score and context similarity.

5.2 Evaluation Results

This section presents the evaluation results of the proposed trust and context-based rating prediction and recommendation approach using standard metrics MAE and RMSE. The MAE defined in equation (19) is the absolute sum of the deviation between the users actual and predicted ratings. On the other hand, RMSE defined in equation (20) is the standard deviation of the prediction errors. RMSE basically determines the intensity of data to a line of best fit and penalizes large errors. In these equations, $\mathcal{T}$ is the test dataset having movies with their ratings provided by different users, and r_{ui} and $\hat{r}_{ui}$ are the actual and predicted ratings for user u on item i, respectively.

$$MAE = \frac{\sum_{(ui)\in\mathcal{T}} |\hat{r}_{ui} - r_{ui}|}{|\mathcal{T}|} \tag{19}$$

$$RMSE = \sqrt{\frac{\sum_{(ui)\in\mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}{|\mathcal{T}|}} \tag{20}$$

5.3 Comparative Analysis

In this section, we present the comparative analysis of our proposed trust and context-based rating prediction and recommendation approach with some baseline methods described in the following paragraphs.

- `Baseline Predictor`: The baseline prediction method is used to handle conditions when both trust and context scores between a user pair is zero, as shown in equation (16). Baseline predictor is defined as the sum of overall average ratings (μ), user deviation (b_u), and item deviation (b_i), such as ($\mu + b_u + b_i$). If both trust and context score for a user pair is zero then rating prediction $\hat{r}_{ui} = b_{ui}$, as given in equation (18), where b_{ui} represents the baseline predictor.

- `Item-Based Collaborative Filtering (IBCF)`: IBCF is used to predict ratings for users based on similarity between target items to other $k-$most similar items.

- `User-Based Collaborative Filtering (UBCF)`: UBCF is used to predict ratings for users based on similarity between $k-$most similar users.

- `Trust-Based Rating Prediction`: In trust-based rating prediction, we only consider trust score computed using equation (10) for multiple trust statements defined using equations (5), (6), and (9).

- `Context-Based Rating Prediction`: In context-based rating prediction, we used only contextual features for finding similarity between a user pair. We used $UBCF_{CF}$ as a baseline method for computing the similarity between $k-$most similar users using their contextual features.

Table 5 and figure 4(a),4(b) presents the comparison results in terms of MAE and RMSE for different top-k values for various baseline and proposed method. It can be observed from this table that MAE and RMSE values for our proposed method is lowest in comparison to other the baseline methods. The values of MAE and RMSE varied in the range of 0.6194 to 0.6390 and 0.8414 to 0.8668, respectively for proposed method, and both are minimum for $k = 30$, which is significantly better than the existing methods. On analysis, it is found that baseline prediction method has maximum MAE and RMSE values in comparison to other methods. Also, User-Based Collaborative Filtering using Contextual Features ($UBCF_{CF}$) outperformed `UBCF` (excluding contextual features) in terms of MAE and RMSE. Our trust and context-based hybrid approach performed 25.6% and 26.07% better in terms of MAE and 33.76% and 34.06% better in terms of RMSE in comparison to the `IBCF` and `UBCF` approaches.

Further, to check the efficacy of our proposed hybrid approach, we compared our proposed approach with two others baseline approaches, first baseline contains only the trust statements, whereas second baseline contains only contextual features. On analysis, we found that proposed approach performed 11.07% and 6.57% better in terms of MAE, and 8.71% and 3.97% better in terms of RMSE in comparison to $UBCF_{CF}$ and trust-based approaches.

6 CONCLUSION

In this paper, we have proposed a trust- and context-based hybrid approach for traditional recommender systems to predict user ratings based on user-based collaborative filtering. The novelty of the proposed approach lies in handling various open issues, such as *data sparsity*, *profiling attack*, and *black-box* recommendation that are generally associated with the traditional recommender systems. In order to compute *trust score* between a pair of users three new trust statements namely *rating-based deviation*, *emotional information*, and *review helpfulness* are defined. On the other hand, context is computed using weighted Jaccard similarity and weighted Pearson's correlation coefficient. The proposed approach have used collaborative filtering, which mainly facilitate the rating prediction for new items.

REFERENCES

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*. Springer, Berlin, Heidelberg, 304–307. https://doi.org/10.1007/3-540-48157-5_29

[2] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. 2011. Context-Aware Recommender Systems. *Association for the Advancement of Artificial Intelligence,* 32, 3 (2011), 217–253. https://doi.org/10.1609/aimag.v32i3.2364

[3] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *Transactions on Information Systems (TOIS)* 23, 1 (2005), 103–145. https://doi.org/10.1145/1055709.1055714

[4] Ghazaleh Beigi, Jiliang Tang, Suhang Wang, and Huan Liu. 2016. Exploiting Emotional Information for Trust/Distrust Prediction. In *Proceedings of the 16th SIAM International Conference on Data Mining (SDM' 16)*. SIAM, Miami, USA, 81–89. https://doi.org/10.1137/1.9781611974348.10

[5] Glenn R. Bewsell. 2012. Distrust, Fear and Emotional Learning: An Online Auction Perspective. *Journal of Theoretical and Applied Electronic Commerce Research* 7, 2 (2012), 1–12. https://doi.org/10.4067/S0718-18762012000200002

[6] Touhid Bhuiyan, Audun Josang, and Yue Xu. 2010. Managing trust in online social networks. In *Handbook of Social Network Technologies and Applications*. Springer, Boston, MA, 471–496. https://doi.org/10.1007/978-1-4419-7142-5_22

[7] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender Systems Survey. *Knowledge-Based Systems* 46 (2013), 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

[8] Iván Cantador and Pablo Castells. 2009. Semantic Contextualisation in a News Recommender System.. In *Proceedings of 1st Workshop on Context-Aware Recommender System (CARS-RecSys'09)*. ACM Press, New York, NY, USA.

[9] Anind K. Dey. 2001. Understanding and using Context. *Personal and Ubiquitous Computing* 5, 1 (2001), 4–7. https://doi.org/10.1007/s007790170019

[10] Paul Dourish. 2004. What We Talk About When We Talk About Context. *Personal and Ubiquitous Computing* 8, 1 (2004), 19–30. https://doi.org/10.1007/s00779-003-0253-8

[11] Jennifer R Dunn and Maurice E Schweitzer. 2005. Feeling and Believing: The Influence of Emotion on Trust. *Journal of Personality and Social Psychology* 88, 1 (2005), 736–748. https://doi.org/10.5465/ambpp.2003.13793140

[12] Anindya Ghose and Panagiotis G Ipeirotis. 2011. Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics. *IEEE Transactions on Knowledge and Data Engineering* 23, 10 (2011), 1498–1512. https://doi.org/10.1109/TKDE.2010.188

[13] Jennifer Golbeck. 2006. Generating Predictive Movie Recommendations from Trust in Social Networks. In *Proceedings of the 4th International Conference on Trust Management (i-Trust' 06)*. Springer, Berlin, Heidelberg, Berlin, Heidelberg, 93–104. https://doi.org/10.1007/11755593_8

[14] Jennifer A. Golbeck. 2005. *Computing and Applying Trust in Web-based Social Networks*. Ph.D. Dissertation.

[15] Guibing Guo, Jie Zhang, Daniel Thalmann, Anirban Basu, and Neil Yorke-Smith. 2014. From Ratings to Trust: An Empirical Study of Implicit Trust in Recommender Systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC' 14)*. ACM Press, Gyeongju, Korea, 248–253. https://doi.org/10.1145/2554850.2554878

[16] Negar Hariri, Bamshad Mobasher, Robin Burke, and Yong Zheng. 2011. Context-Aware Recommendation based on Review Mining. In *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP–IJCAI'11)*. AAAI Press, Barcelona, Spain, 30–36.

[17] Chein-Shung Hwang and Yu-Pin Chen. 2007. Using Trust in Collaborative Filtering Recommendation. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE' 07)*. Springer, Berlin, Heidelberg, Kyoto, Japan, 1052–1060. https://doi.org/10.1007/978-3-540-73325-6_105

[18] Audun Jøsang, Ross Hayward, and Simon Pope. 2006. Trust Network Analysis with Subjective Logic. In *Proceedings of the 29th Australasian Computer Science Conference (ACSC ' 06)*, Vol. 48. ACM Press, Hobart, Australia, 85–94.

[19] Henry Lieberman and Ted Selker. 2000. Out of context: Computer Systems that Adapt to, and Learn from, Context. *IBM Systems Journal* 39, 3.4 (2000), 617–632. https://doi.org/10.1147/sj.393.0617

[20] Paolo Massa and Paolo Avesani. 2004. Trust-aware Collaborative Filtering for Recommender Systems. In *Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems" (OTM' 04)*. Springer, Berlin, Heidelberg, Napa, Cyprus, 492–508. https://doi.org/10.1007/978-3-540-30468-5_31

[21] Paolo Massa and Paolo Avesani. 2007. Trust-aware Recommender Systems. In *Proceedings of the 7th International Conference on Recommender Systems (RecSys' 07)*. ACM Press, Minneapolis, USA, 17–24. https://doi.org/10.1145/1297231.1297235

[22] Paolo Massa and Paolo Avesani. 2007. Trust Metrics on Controversial Users: Balancing Between Tyranny of the Majority. *International Journal on Semantic Web and Information Systems (IJSWIS)* 3, 1 (2007), 39–64.

[23] Roger C. Mayer, James H. Davis, and David F. Schoorman. 1995. An Integrative Model of Organizational Trust. *Academy of Management Review* 20, 3 (1995), 709–734. https://doi.org/10.5465/amr.1995.9508080335

[24] John O'Donovan and Barry Smyth. 2005. Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI ' 05)*. ACM Press, San Diego, California, USA, 167–174. https://doi.org/10.1145/1040830.1040870

[25] Ben J. Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web*. Springer, Berlin, Heidelberg, 291–324.

[26] Ben J. Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. *The Adaptive Web* 4321 (2007), 291–324. https://doi.org/10.1007/978-3-540-72079-9_9

[27] Guan Wang, Sihong Xie, Bing Liu, and S. Yu Philip. 2011. Review Graph Based Online Store Review Spammer Detection. In *Proceedings of the 11th International Conference on Data Mining (ICDM' 11)*. ACM Press, Vancouver, Canada, 1242–1247. https://doi.org/10.1109/ICDM.2011.124

[28] Weilong Yao, Jing He, Guangyan Huang, Jie Cao, and Yanchun Zhang. 2015. A graph-based Model for Context-aware Recommendation Using Implicit Feedback Data. *World Wide Web* 18, 5 (2015), 1351–1371.

[29] Weilong Yao, Jing He, Guangyan Huang, Jie Cao, and Yanchun Zhang. 2015. A Graph-based Model for Context-Aware Recommendation Using Implicit Feedback Data. *World Wide Web* 18, 5 (2015), 1351–1371. https://doi.org/10.1007/s11280-014-0307-z

[30] Yong Zheng, Robin Burke, and Bamshad Mobasher. 2013. Recommendation with Differential Context Weighting. In *Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization*. Springer, Berlin, Heidelberg, 152–164.

[31] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. Similarity-based Context-aware Recommendation. In *Proceedings of the 16th International Conference on Web Information Systems Engineering (WISE' 16)*. Springer, Miami, USA, 431–447.