The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 1, 2018
Pages 10-19
Regular Paper

Improving Recommender Systems Using Context-Dependent Trust Relationships

Faezeh Sadat Gohari Fereidoon Shams Aliee Hassan Haghighi

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

Abstract

Trust-based recommender systems use trust relationships between users to improve the quality of recommendations. One of the most important features of trust is context-dependency. Despite the importance of context-dependency, this feature has been largely neglected in the current literature. In this paper, we propose a new approach that considers the semantic context of items to infer trust relationships between users. In this approach, the level of trust between two users varies depending on different contexts. Therefore, the trustworthy neighbors of an active user will be different for different target items, and these neighbors are determined according to the target context. The focus on context-specific ratings instead of all ratings results in fewer online computations, thus increasing the efficiency of the system as well as the accuracy of recommendations. Experimental results on a real-world data set show the higher accuracy and efficiency of the proposed approach compared to its counterparts.

Keywords: Recommender systems, Trust, Semantic context, Trust-based recommender systems, Collaborative filtering, Context-aware recommender systems

1. Introduction

With the ongoing rapid growth of the Internet, people are faced with more information than they can process and understand. One solution to this information overload problem is the use of Recommender Systems (RSs). These systems aim to provide end users with suggestions about information items, products or services that are likely to be of their interest. In recent years, RSs have become increasingly popular and have been applied to diverse domains [1–4].

Collaborative Filtering (CF) is the most popular recommendation technique [5]. The underlying assumption of CF is that similar users share similar interests. In order to provide relevant recommendations for an active user, CF first identifies a set of users (neighbors) that are similar to the active user. It then uses the opinions of these neighbors to predict the unknown preferences of the active user. Despite the popularity of CF, it suffers from two main problems: *data sparsity* [6–8] and *cold-start* [9–12]. Data sparsity arises when the number of observed ratings in the rating matrix is insufficient to find similar users. The cold-start problem refers to new users who have rated only a few or even no items. A new user cannot receive any personalized recommendations due to the lack of enough initial ratings. To overcome these problems, trust-based RSs have been proposed, which compensate for the lack of rating data with trust information. The intuition behind trust-based systems is that in real life users tend to prefer recommendations from people they trust [13]. Previous researches have shown that the incorporation of trust information into the traditional CF method can resolve the above-mentioned problems and improve the quality of recommendations [3,5–7,9,14–17].

Trust-based recommendation approaches can be divided into explicit and implicit approaches [5]. In explicit approaches, trust is directly specified by users themselves. In implicit approaches, trust is inferred from user behavior (e.g., ratings). Although explicit approaches tend to be more accurate than implicit ones, they require additional user effort. Due to this fact, explicit trust statements may not always be available [18]. Therefore, implicit approaches are more practical than explicit approaches [5]. In this work, we focus on implicit trust.

According to the trust theory, one of the most important features of trust is context-dependency [19]. This means that users who are trustworthy in a specific context are not necessarily trustworthy in another context. For example, a user who provides good recommendations in the movie domain may not be trustworthy in the context of music. The context can refer to the type of rated items or the condition in which ratings are issued, such as location and time. Despite the importance of context-dependency, this feature has been largely neglected in the current literature [19].

In this paper, we propose a new Context-Aware Trust-based Recommendation Approach (CATRA), which considers the semantic context of items to infer trust relationships between users. The rationale behind this idea is that the trust value between two users depends on the semantic content of the existing items. For example, a user who provides valuable recommendations of comedy movies may not have sufficient expertise in recommending horror movies. Thus, depending on the content of an item, recommendations from a neighbor may or may not be trustworthy. In our proposed approach, the level of trust between two users varies depending on different semantic contexts. Therefore, the trustworthy neighbors of an active user will be different for different target items, and these neighbors are determined according to the target context.

CATRA works in two phases: offline and online. In the offline phase, items are clustered based on their semantic similarities. In the online phase, CATRA predicts the unknown rating of an active user for a target item. For this purpose, a context-dependent trust network is generated according to the semantic context of the target item. Then, this network is used to identify the most trustworthy neighbors of the active user in the target context. In the next step, the unknown rating of the target item is predicted by aggregating the ratings of the selected neighbors. Finally, the top-N items with the highest predicted values are recommended to the active user. We conduct several experiments on the MovieLens data set to evaluate the performance of CATRA. The results confirm that the proposed approach achieves higher accuracy and efficiency than its counterparts.

The rest of this paper is organized as follows. Section 2 provides a brief overview of related work. In Section 3, we describe the proposed approach in detail. The experimental results are presented and discussed in Section 4. Finally, Section 5 presents the conclusions and outlines future research directions.

2. Related work

CF is the most widely used approach in the current RSs [20]. CF-based approaches can be classified into memory-based [21,22] and model-based [23–27] approaches. Memory-based approaches use the entire rating matrix to find similar users and generate a prediction. These approaches are simple, robust and easily implementable [14]. However, they have a major drawback of high computational complexity. Therefore, these approaches suffer from serious scalability problems when there are millions of users and items [28].

To avoid the scalability problem of memory-based CF, model-based approaches learn a model from rating data. Once the learning phase is completed, unseen ratings can be quickly predicted using the learned model. Among the widely used models are clustering models [23,29,30], probabilistic models [31], matrix factorization [24] and latent factor models [25,32]. Although model-based approaches are highly scalable, they require an expensive model-building process. In these approaches, new data cannot be immediately incorporated into the model, since this incurs significant update overhead.

CF-based approaches suffer from the data sparsity and cold-start problems. These problems have a serious effect on the accuracy of recommendations. To alleviate these issues, various hybrid approaches have been proposed that improve the performance of CF using additional sources of information [3,15,33,34]. Recent studies have shown that social information, (e.g., trust relationship between users) is a rich source of side information [35]. In this regard, trust-based RSs have emerged to provide users with personalized recommendations based on their trust relationships [3,5–7,9,14–17]. Trust information can be explicitly collected from users or implicitly inferred from their ratings on items. Until now, many successful explicit trust-based approaches have been reported in the literature [14,26,35,36]. However, since explicit trust statements are not always available, implicit approaches are more practical [5].

Implicit trust-based recommendation approaches construct an implicit trust network by analyzing rating patterns of users. For example, in [37,38], the trustworthiness of user y is measured as the percentage of correct predictions that y has made in general or with respect to a particular item. In [6,18], trust is calculated using the Pearson correlation coefficient. In [39], trust is defined as the mean absolute difference between the ratings of two users on co-rated items. In [40], the degree of trust between a trustor and a trustee is calculated by averaging the prediction error on co-rated items.

In some other works [5,15,41], trust is defined as a combination of two metrics: (1) Mean Squared Differences (MSD), which is related to the prediction accuracy, and (2) Jaccard metric, which calculates the ratio of the number of items co-rated by two users to the total number of items rated by both users. In [35], the authors adopt and revise a Beta trust model to calculate the trust value between two users. This trust model evaluates whether a trustee shares consistent preferences with a trustor across different items. Depending on whether trustee y 's preference on an item is close to or far away from trustor x's preference, it is considered that y has displayed either a “good behavior” (i.e., consistent preference) or a “bad behavior” (i.e., inconsistent preference). This trust model also incorporates a time factor to capture users’ preference changes.

None of the aforementioned studies take the semantic context of items into account when constructing the trust network. In this paper, we propose a new approach that incorporates semantic information of items into a trust model. The aim of this approach is to improve the effectiveness and efficiency of the neighborhood selection process.

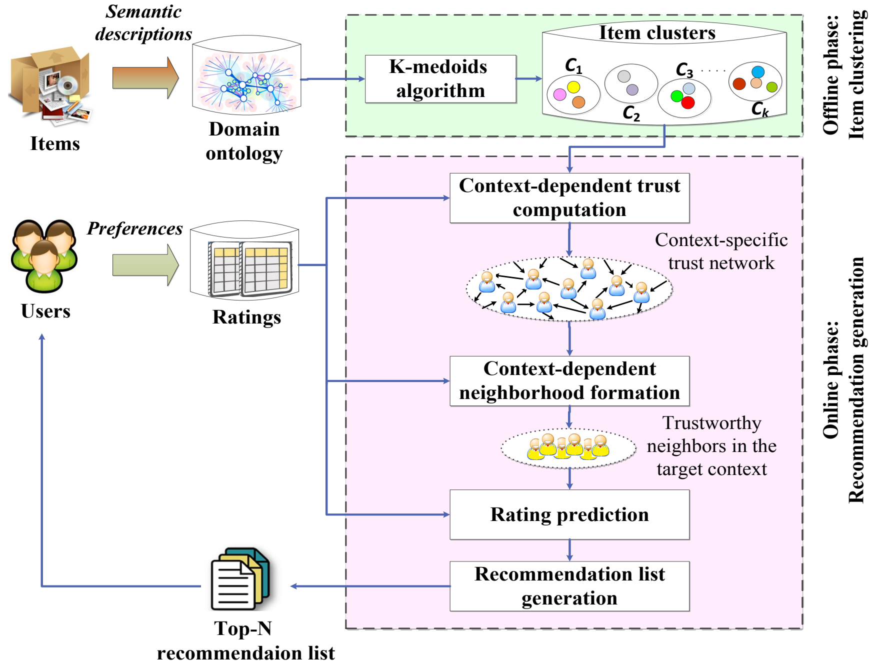


Fig. 1: The framework of the proposed approach

3. Proposed approach

In this section, we propose a new context-aware trust-based recommendation approach, called CATRA. Fig. 1 shows the framework of the proposed approach. As shown in Fig. 1, CATRA consists of an offline and an online phase. The aim of the offline phase is to cluster items based on their semantic similarities. The online phase consists of four steps. First, CATRA derives trust values between users in the target context (i.e., context-dependent trust computation step). In the second step, the most trustworthy neighbors of an active user are determined according to the target context (i.e., context-dependent neighborhood formation step). In the next step, the unknown ratings of the active user are predicted based on the opinions of his neighbors in the target context (i.e., rating prediction step). Finally, a recommendation list is generated for the active user (i.e., recommendation list generation step). The pseudo-code of CATRA is shown in Algorithm 1.

3.1. Offline phase: Item clustering

As mentioned before, in CATRA, items are clustered according to their semantic descriptions. For this purpose, item characteristics should be represented by an ontology. The domain ontology used in this study is Movie Ontology[1] (MO), which has been developed according to the Ontology Web Language (OWL) standard by the University of Zurich. We chose the movie domain because it is a well-known application domain of RSs. Also, there are a large number of concepts and relationships in this domain (such as movies, genres, actors, directors, writers, etc.). However, the proposed approach can potentially be used for other domains with different ontologies.

MO provides a controlled vocabulary to semantically describe concepts and their associated individuals. The main class of this ontology is “Movie”. All the movies are instances of this class. To instantiate MO, we use the Internet Movie Database[2] and gather required data by a web crawler. The semantic similarity between two items *A* and *B* based on their semantic descriptions is defined as [42]:

$$SemSim(A,B)=\sum_{i=1}^{|\mathrm{P}|}\left(\frac{\mathrm{common}(A,B,P[i])}{\max(\deg(A,P[i]),\deg(B,P[i]))}\right)\times \mathrm{Weight}(P[i]) \qquad (1)$$

where P is a vector that contains a set of properties of the Movie class; $\deg(A,p)$ represents the number of instances associated with A by property p; $\mathrm{common}(A,B,p)$ denotes the number of common instances associated with A and B by property p; and $\mathrm{Weight}(p)$ indicates the importance of property p. The weight of a property is subjectively determined according to the given domain. For example, in the movie domain, the genre of a movie is more important than its filming locations.

We now give an example to illustrate Eq. (1). Suppose that the property set is as follows:

P = [belongsToGenre, hasFilmLocation]

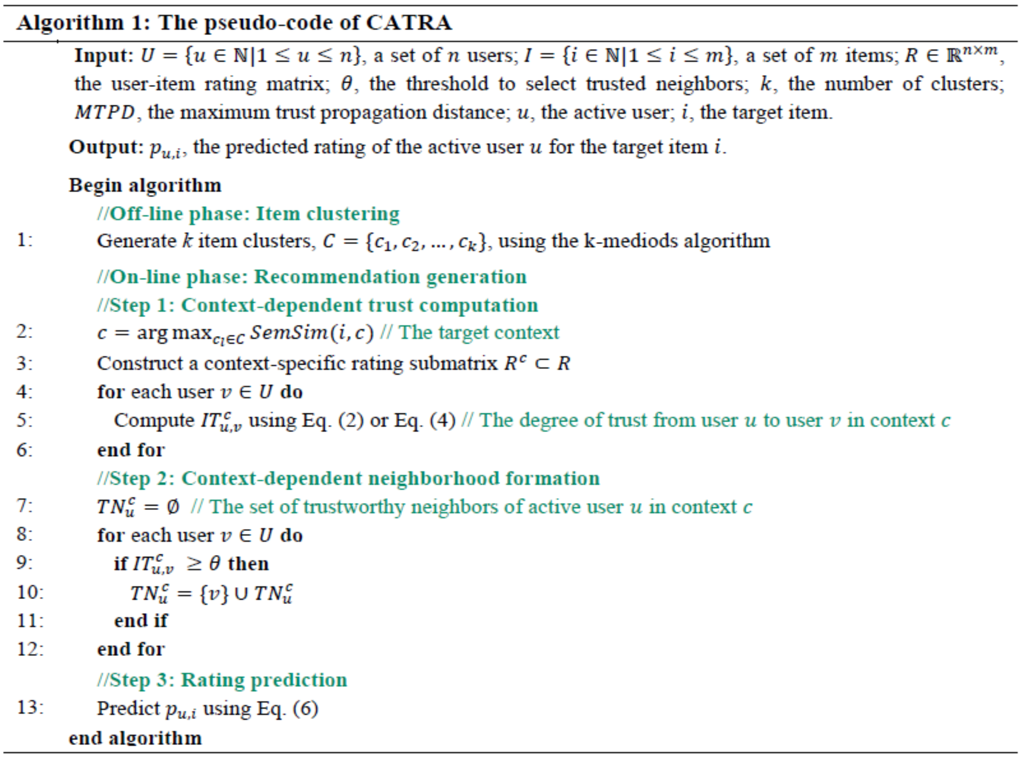
where *belongsToGenre* represents the genre(s) of a movie and *hasFilmLocation* denotes its filming locations. Associated instances with *A* and *B* are:

---

[1] http://www.movieontology.org/

[2] http://www.imdb.com/

A.belongsToGenre = [Drama, Horror]
A.hasFilmLocation = [USA]



Algorithm 1: The pseudo-code of CATRA

```
Input: $U = \{u \in \mathbb{N} | 1 \le u \le n\}$, a set of $n$ users; $I = \{i \in \mathbb{N} | 1 \le i \le m\}$, a set of $m$ items; $R \in \mathbb{R}^{n \times m}$, the user-item rating matrix; $\theta$, the threshold to select trusted neighbors; $k$, the number of clusters; $MTPD$, the maximum trust propagation distance; $u$, the active user; $i$, the target item.
Output: $p_{u,i}$, the predicted rating of the active user $u$ for the target item $i$.
Begin algorithm
    //Off-line phase: Item clustering
    Generate $k$ item clusters, $C = \{c_1, c_2, \dots, c_k\}$, using the k-medoids algorithm
    //On-line phase: Recommendation generation
    //Step 1: Context-dependent trust computation
    $c = \arg\max_{c_i \in C} SemSim(i, c)$ // The target context
    Construct a context-specific rating submatrix $R^c \subset R$
    for each user $v \in U$ do
        Compute $IT_{u,v}^c$ using Eq. (2) or Eq. (4) // The degree of trust from user $u$ to user $v$ in context $c$
    end for
    //Step 2: Context-dependent neighborhood formation
    $TN_u^c = \emptyset$ // The set of trustworthy neighbors of active user $u$ in context $c$
    for each user $v \in U$ do
        if $IT_{u,v}^c \ge \theta$ then
            $TN_u^c = \{v\} \cup TN_u^c$
        end if
    end for
    //Step 3: Rating prediction
    Predict $p_{u,i}$ using Eq. (6)
end algorithm
```

B.belongsToGenre = [Romance, Drama, Sci-fi]
B.hasFilmLocation = [Italy]

Therefore:

deg(A, belongsToGenre) = 2
deg(A, hasFilmLocation) = 1
deg(B, belongsToGenre) = 3
deg(B, hasFilmLocation) = 1
common(A, B, belongsToGenre) = 1
common(A, B, hasFilmLocation) = 0

Suppose that the weights of *belongsToGenre* and *hasFilmLocation* are 0.8 and 0.2, respectively. Then:

SemSim(A,B) = (1/3) × 0.8 + (0/1) × 0.2 = 0.266

After calculating the semantic similarity between each pair of movies, the clustering process is performed. For this purpose, we adopt the k-medoids algorithm [43] due to its simplicity and high accuracy. K-medoids is a partition based clustering algorithm similar to k-means [44], but it selects objects as centers (medoids) instead of taking the mean value of the object. K-medoids is more robust to the noise and outliers than k-means, because it minimizes the sum of pairwise distances in a cluster. Also, it is not generally influenced by the order of presentation of the objects.

The k-medoids algorithm works as follows: First, k items are randomly selected as the initial medoids. Then, the algorithm proceeds by alternating between two steps. In the first step, each item is assigned to the cluster with the nearest medoid. In this study, the semantic similarity is used as the distance metric to measure the closeness of two items. The distance between two items A and B is calculated as $1 - Semsim(A, B)$. In the second step, for each cluster, k-medoids swaps each non-medoid item with the medoid of that cluster. If the sum of within-cluster distances decreases, that item is chosen as a new medoid. These steps are repeated until the medoids become fixed. Therefore, a set of k item clusters, $C = \{c_1, c_2, \dots, c_k\}$, is generated.

3.2. Online phase: Recommendation generation

The online phase aims to generate a list of recommendations for an active user. For this purpose, CATRA predicts ratings of unseen items for the active user. According to the predicted ratings, the items are ranked, and those with the highest ranking are recommended to the user. To predict the rating of user u on target item i, the following steps are taken:

3.2.1. Context-dependent trust computation

In our proposed approach, trust relationships depend on the context. In particular, CATRA considers the semantic context of items to infer trust relationships between users. Therefore, the level of trust between two users varies depending on different contexts. For example, suppose that the trust value from user u to user v is 0.9 in context c_1, while this value is 0.1 in context c_2. This indicates that v is a trustworthy neighbor of u in context c_1 but is not trustworthy in context c_2. Context-specific trust values are calculated as follows:

Let U be a set of n users and I be a set of m items. The rating data are stored in an $n \times m$ user-item rating matrix, denoted by R. To predict the rating of active user u on target item i, CATRA first determines the semantic context of the target item (line 2 of Algorithm 1). For this purpose, the semantic similarity between the target item and each cluster center is calculated. The most similar cluster is considered as the target context and is denoted by c.

According to the target context c, a context-specific rating submatrix, denoted by R^c, is constructed (line 3). This $n \times |c|$ submatrix contains user ratings for items in cluster c. Using R^c, the implicit trust values between the active user u and other users are calculated in the target context c (lines 4-5). In this study, we adopt the implicit trust metric proposed in [26] and modify it to incorporate contextual information. Let I_u^c be the set of items in context c rated by user u. According to R^c, if two users u and v have rated at least one common item in context c, the implicit trust value is directly computed as follows:

$$IT_{u,v}^{c} = \left(1 - \frac{\sum_{i \in I_u^c \cap I_v^c} (p_{u,i}^{v} - r_{u,i})^2}{\left|I_u^c \cap I_v^c\right|}\right) \times \frac{\left|I_u^c \cap I_v^c\right|}{\left|I_u^c\right| + \left|I_v^c\right| - \left|I_u^c \cap I_v^c\right|} \tag{2}$$

where $IT_{u,v}^{c}$ is the implicit trust of user u to user v in context c; $I_u^c \cap I_v^c$ is the set of items in context c that both users have rated; $r_{u,i}$ represents the rating of user u on item i; and $p_{u,i}^{v}$ is the predicted rating of item i for user u based on the opinion of v:

$$p_{u,i}^{v} = \bar{r}_u + (r_{v,i} - \bar{r}_v) \tag{3}$$

where $\bar{r}_u$ is the mean rating of user u in the matrix R^c.

If there is no co-rated item between u and v in context c, trust propagation is used to establish an indirect trust relationship between them [26]:

$$IT_{u,v}^{c} = 1 - \frac{\sum_{s \in DN(u)} IT_{u,s}^{c} \times (IT_{s,v}^{c} \times \beta)}{\sum_{s \in DN(u)} IT_{u,s}^{c}} \tag{4}$$

where $DN(u)$ denotes the set of direct neighbors of user u; and β is a parameter to ensure that trust decreases along the propagation direction:

$$\beta = (MTPD - d + 1) / MTPD \tag{5}$$

where $MTPD$ is a tunable parameter that controls the maximum trust propagation distance; and d is the distance between the trustor and the trustee. With increasing d, the parameter β decreases, resulting in a lower trust value.

3.2.2. Context-dependent neighborhood formation

In this step, CATRA selects the most trustworthy neighbors of the active user u. The decision on whether to select a user as a neighbor of u depends on the target context. Thus, the trustworthy neighbors of the active user are different for different contexts. Let TN_u^c be the set of trustworthy neighbors of user u in context c (line 7 of Algorithm 1). As shown in Algorithm 1, user v is a trustworthy neighbor of u in context c if $IT_{u,v}^{c}$ is greater than or equal to a specified threshold θ (lines 8–12).

3.2.3. Rating prediction

In the last step, the unknown rating of the active user for the target item is predicted (line 13). Let $p_{u,i}$ be the predicted rating of active user u on target item i. The prediction $p_{u,i}$ is calculated as the weighted average of the ratings given by neighbors to item i:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in TN_u^c} (r_{v,i} - \bar{r}_v) \times IT_{u,v}^{c}}{\sum_{v \in TN_u^c} IT_{u,v}^{c}} \tag{6}$$

3.3. Computational complexity analysis

In this section, we analyze the computational complexity of the offline and online phases of CATRA. In the offline phase, $O(k(m-k)^2)$ is required for clustering m items using the k-medoids algorithm. In general, offline computations have no effect on the online performance of an RS. In other words, the efficiency of an RS depends on the amount of online computations. In the online phase, CATRA requires $O((n \times m)^{MTPD})$ time to construct the implicit trust network. Then, $O(n)$ is required for the neighborhood formation process. Finally, it takes $O(m)$ time to predict the unknown ratings of the active user. Therefore, the online time complexity of CATRA is of $O((n \times m)^{MTPD}) + O(n) + O(m) \approx O((n \times m)^{MTPD})$.

The network construction process is the most time-consuming step in a trust-based RS. Due to this fact, in some systems, this process is performed offline to improve the efficiency of the system. However, the offline computation of trust has a negative effect on the accuracy of recommendations, because new ratings cannot be immediately incorporated into the recommendation process. To avoid this problem, CATRA performs the network construction process online. In our proposed approach, the use of context-specific ratings instead of all ratings speeds up the computations. In other words, due to a significant decrease in m, CATRA is more efficient than those that take all items into account when constructing the trust network.

4. Experiments

In this section, we conduct a number of experiments to evaluate the performance of the proposed approach and compare it with other competing approaches. We compare CATRA with the following approaches:

- TACF [40]: The basic trust-aware CF approach, which calculates implicit trust values based on the prediction error of co-rated items.
- TARS [45]: A trust-based ant recommender system that uses the concept of dynamic trust to provide recommendations. It continuously updates the implicit trust network and selects the best neighborhood using ant colony optimization.
- TSF [26]: A trust–semantic fusion-based approach that combines a user-based trust-enhanced CF method with an item-based semantic-enhanced CF method. TSF obtains trust values from an implicit trust network.
- HUIT [5]: A hybrid user-item trust-based recommendation approach that combines the users' and items' implicit trust information to extend the neighborhood of a user or an item.
- MVCCF [46]: A model-based CF approach that uses a multi-view clustering method to cluster users and items from multiple modalities. In this approach, users (items) belonging to the same cluster are considered as neighbors and involved in the prediction process.
- UPUC-CF [47]: A CF approach based on user preference clustering that aims to reduce the impact of data sparsity. In this approach, users are clustered into different user groups (i.e., optimistic, pessimistic and neutral) to distinguish users with different preferences.
- TBRSK [48]: It is a trust-based RS using k-means clustering. TBRSK uses the k-means algorithm to cluster users based on their implicit trust relationships. This approach uses the ant colony to find trustworthy neighbors of a user.
- TMFSF [38]: A CF approach that is based on an implicit trust model with fused similarity factor. TMFSF uses a new factor to revise the similarity. The adjusted similarity is combined with the trust model to derive harmonic weights.
- TTLCF [35]: A dynamic implicit trust-based approach that uses a novel two-layer neighbor selection scheme. This scheme selects the most capable and trustworthy neighbors of a user. It also considers time factors to address users' preference changes. For the sake of fairness, we do not consider time factors in our experiments.

4.1. Experimental design

We conduct experiments on the MovieLens 1M data set (http://grouplens.org/datasets/movielens/1m/). This data set consists of 1,000,209 ratings from 6040 users on 3952 movies. Ratings are discrete values from 1 to 5. We use 5-fold-cross-validation to evaluate the performance of an RS over multiple runs. We randomly split the data set into a training set (80% of users) and a test set (20% of users). For each test user, *n* (i.e., 5, 10, 15 and 25) ratings are randomly selected and put into the training set. The remaining ratings are used to test the system.

The performance is measured in terms of predictive accuracy. For this purpose, we use two popular metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Both metrics measure the closeness of predicted ratings to the true ratings, with the difference that RMSE gives more weight to large errors:

$$\mathrm{MAE} = \frac{1}{\#\mathrm{TS}} \sum_{(u,i)\in \mathrm{TS}} \left| r_{u,i} - p_{u,i} \right| \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\#\mathrm{TS}} \sum_{(u,i)\in \mathrm{TS}} \left(r_{u,i} - p_{u,i} \right)^2} \tag{8}$$

where TS denotes the test set and #TS is the cardinality of TS.

All approaches are implemented in Matlab R2016b. The parameters of each approach are tuned to obtain the best results. To interact with the ontology, we use OWL API [49], a high-level Application Programming Interface (API) for working with OWL ontologies. OWL API has been implemented in Java. It is available as open source under an LGPL license.

The experiments are performed on a 2.53 GHz Intel Core i5 processor, with 4.00 GB RAM, and Windows 7 operating system.

4.2. Parameter setup

To achieve the best performance, it is important to fine-tune the input parameters of CATRA. These parameters include the number of clusters (k), maximum trust propagation distance (*MTPD*) and the neighbor selection threshold (θ). The optimal values of these parameters are selected based on a sensitivity analysis. We examine the parameters individually. For a given parameter x, we change its value, while keeping the other parameters constant. The appropriate value of x is determined. This optimal value is assigned to x and the procedure is repeated for the next parameter. Table 1 shows the parameter setting of the proposed approach.

4.3. Experimental results and discussions

In the first experiment, we compare the predictive performance of CATRA with the other approaches. The MAE and RMSE results for different values of *n* (i.e., the number of observed ratings for a test user) are shown in Figs. 2 and 3, respectively. As expected, with increasing *n*, the prediction accuracy tends to improve. According to the results, the trust-based approaches often perform better than MVCCF and UPUC-CF. The reason is that MVCCF and UPUC-CF consider only the rating information in the recommendation process. The use of trust as an additional source of information alleviates the sparsity and cold-start issues, resulting in higher accuracy for trust-based approaches. As it can be seen in Figs. 2 and 3, CATRA outperforms all the other approaches in terms of both MAE and RMSE.

Table 1: CATRA parameter settings

Parameter	k	θ	*MTPD*
Value	15	0.75	3

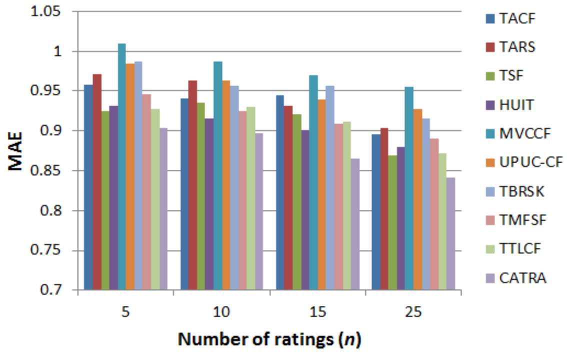


Fig. 2: Comparison of different approaches in terms of MAE

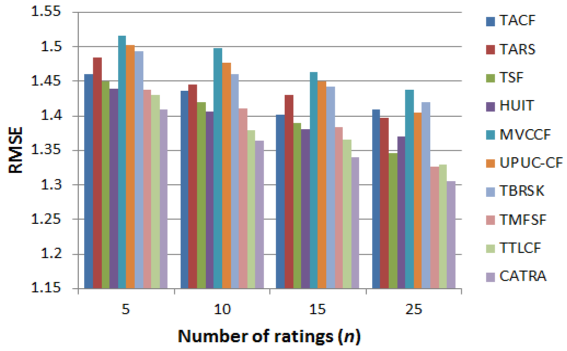


Fig. 3: Comparison of different approaches in terms of RMSE

In our proposed approach, the incorporation of contextual information into the trust computation improves the accuracy of trust values, which in turn increases the quality of the user neighborhood. None of the examined approaches (i.e., TACF, TARS, TSF, HUIT, TBRSK, TMFSF and TTLCF) consider the semantic context of items for computing implicit trust values. Thus, the neighborhood set of a user is the same across different items, regardless of their semantic content. This has a negative effect on the accuracy of recommendations. This experiment proves that the incorporation of semantic information into the trust model is effective in improving the predictive performance of trust-based RSs.

In the second experiment, we compare the efficiency of CATRA with the other approaches. For this purpose, we measure the amount of time required for the online phase of each approach. For the trust-based approaches, the maximum trust propagation distance (*MTPD*) is set to 2. Fig. 4 shows the execution time (in milliseconds) of each approach.

As shown in Fig. 4, MVCCF, UPUC-CF and TBRSK are more efficient than the other approaches. The reason is that, in these model-based approaches, users (or items) are clustered based on their ratings. Thus, the time-consuming process of calculating similarity (or trust) values is performed offline. This will reduce the time required for the online phase. However, as shown in the previous experiment, these approaches do not perform well in terms of prediction accuracy. Moreover, they reduce the ability of the system to adapt to changes in users' preferences. In real-world applications, new ratings arrive continuously at a high rate. However, these new data cannot be immediately incorporated into the clustering model, since this incurs significant update overhead. Therefore, due to the dynamic nature of the rating matrix, the use of rating information for clustering purposes reduces the adaptability of the system.

The results show that CATRA has a shorter execution time than TACF, TARS, TSF, HUIT, TMFSF and TTLCF. To better understand the efficiency of the proposed approach, we also measure the execution time for different values of *MTPD*. Fig. 5 shows the results for different approaches. As it can be seen from Fig. 5, CATRA has the best performance at different propagation distances. The superiority of CATRA becomes more evident as *MTPD* increases. As mentioned in Section 3.3, the most time-consuming part of a trust-based RS is the network construction process. The more ratings users provide, the slower the network construction process. In TACF, TARS, TSF, HUIT, TMFSF and TTLCF, the trust computation is performed based on all available ratings. Thus, they require a long execution time, which increases with increasing *MTPD*. In contrast, CATRA focuses on context-specific ratings instead of all ratings, which results in fewer online computations and thus higher efficiency.

According to the results of these experiments, the incorporation of contextual information into the trust computation improves not only the prediction accuracy but also the efficiency of the system.

5. Conclusions and future work

In this paper, we proposed a new trust-based recommendation approach, called CATRA, which considers the semantic context of items to infer trust relationships between users. It consists of an offline and an online phase. The offline phase aims to cluster items based their semantic content. The online phase is responsible for generating recommendations. For this purpose, it first constructs a context-dependent trust network. Then, the most trustworthy neighbors of a user are identified in the target context. Finally, a list of recommendations is generated for the user based on the preferences of selected neighbors. The experimental results on the MovieLens data set showed that CATRA outperforms the benchmark methods in terms of both prediction accuracy and efficiency. For future research, we plan to further improve the accuracy of CATRA by incorporating different types of contextual information such as the user's current mood, time or location.

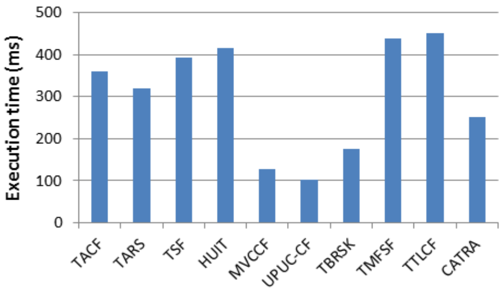


Fig. 4: Comparison of execution times

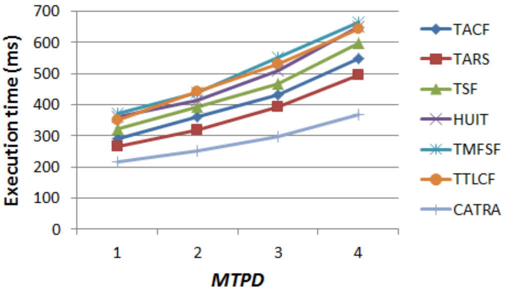


Fig. 5: Comparison of execution times for different propagation distances

References

[1] S. Deng, D. Wang, X. Li, G. Xu, "Exploring user emotion in microblogs for music recommendation," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9284–9293, 2015.

[2] M. Capelle, F. Hogenboom, A. Hogenboom, F. Frasincar, "Semantic News Recommendation Using Wordnet and Bing Similarities," Proc, ACM 28th Annual Symposium on Applied Computing, pp. 296–302, 2013.

[3] F.S. Gohari, F.S. Aliee, H. Haghighi, "A New Confidence-Based Recommendation Approach: Combining Trust and Certainty," *Information Sciences*, vol. 422, pp. 21–50, 2018.

[4] D. Bianchini, V. De Antonellis, N. De Franceschi, M. Melchiori, "PREFer: A prescription-based food recommender system," *Computer Standards & Interfaces*, vol. 54, pp. 64–75, 2017.

[5] Q. Shambour, J. Lu, "An effective recommender system by unifying user and item trust information for B2B applications," *Journal of Computer and System Sciences*, vol. 81, no. 7, pp. 1110–1126, 2015.

[6] M. Papagelis, D. Plexousakis, T. Kutsuras, "Alleviating the sparsity problem of collaborative filtering using trust inferences," Trust Management, Springer, pp. 224–239, 2005.

[7] G. Guo, J. Zhang, D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems*, vol. 57, pp. 57–68, 2014.

[8] Y. Moshfeghi, B. Piwowarski, J.M. Jose, "Handling Data Sparsity in Collaborative Filtering Using Emotion and Semantic Based Features," Proc, ACM 34th International SIGIR Conference on Research and Development in Information Retrieval, pp. 625–634, 2011.

[9] C.C. Chen, Y.-H. Wan, M.-C. Chung, Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2013.

[10] A. Hernando, J. Bobadilla, F. Ortega, A. Gutiérrez, "A probabilistic model for recommending to new cold-start non-registered users," *Information Sciences*, vol. 376, pp. 216–232, 2017.

[11] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[12] L. Li, X. Tang, "A Solution to the Cold-Start Problem in Recommender Systems Based on Social Choice Theory," Intelligent and Evolutionary Systems, Springer, pp. 267–279, 2016.

[13] R.R. Sinha, K. Swearingen, "Comparing recommendations made by online systems and friends," DELOS Workshop on Personalisation and Recommender Systems in Digital Libraries, vol. 106, 2001.

[14] M. Ghavipour, M.R. Meybodi, "Stochastic trust network enriched by similarity relations to enhance trust-aware recommendations," *Appl Intell*, pp. 1–14, 2018.

[15] Q. Shambour, J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decision Support Systems*, vol. 54, no. 1, pp. 768–780, 2012.

[16] Z. Zhang, Y. Liu, Z. Jin, R. Zhang, "A dynamic trust based two-layer neighbor selection scheme towards online recommender systems," *Neurocomputing*, vol. 285, pp. 94–103, 2018.

[17] F.S. Gohari, H. Haghighi, F.S. Aliee, "A semantic-enhanced trust based recommender system using ant colony optimization," *Appl Intell*, vol. 46, no. 2, pp. 328–364, 2017.

[18] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, S.J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 232–238, 2010.

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

[20] F.S. Gohari, M.J. Tarokh, "Classification and Comparison of the Hybrid Collaborative Filtering Systems," *International Journal of Research in Industrial Engineering*, vol. 4, no. 2, pp. 129–148, 2017.

[21] P. Resnick, H.R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[22] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proc, ACM 10th International Conference on World Wide Web, pp. 285–295, 2001.

[23] N. Mirbakhsh, C.X. Ling, "Leveraging clustering to improve collaborative filtering," *Information Systems Frontiers*, vol. 20, no. 1, pp. 111–124, 2018.

[24] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, L. Sun, "Dual-Regularized Matrix Factorization with Deep Neural Networks for Recommender Systems," *Knowledge-Based Systems*, vol. 145, pp. 46–58, 2018.

[25] G.-N. HU, X.-Y. Dai, F.-Y. Qiu, R. Xia, T. Li, S.-J. Huang, J.-J. Chen, "Collaborative Filtering with Topic and Social Latent Factors Incorporating Implicit Feedback," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 12, no. 2, p. 23, 2018.

[26] G. Guo, J. Zhang, F. Zhu, X. Wang, “Factored similarity models with social trust for top-N item recommendation,” *Knowledge-Based Systems*, vol. 122, pp. 17–25, 2017.

[27] F.S. Gohari, F.S. Aliee, H. Haghighi, “A trust-aware group recommender system using particle swarm optimization,” International Symposium on Computer Science and Software Engineering Conference (CSSE), pp. 80–85, 2017.

[28] A. Hernando, J. Bobadilla, F. Ortega, “A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model,” *Knowledge-Based Systems*, vol. 97, pp. 188–202, 2016.

[29] L. Sheugh, S.H. Alizadeh, “A novel 2D-Graph clustering method based on trust and similarity measures to enhance accuracy and coverage in recommender systems,” *Information Sciences*, vol. 432, pp. 210–230, 2018.

[30] S. Kant, T. Mahara, V. Kumar Jain, D. Kumar Jain, A. Kumar Sangaiah, “LeaderRank based k-means clustering initialization method for collaborative filtering,” *Computers & Electrical Engineering*, vol. 69, pp. 598–609, 2017.

[31] S. Qiao, N. Han, J. Zhou, R.-H. Li, C. Jin, L.A. Gutierrez, “SocialMix: A familiarity-based and preference-aware location suggestion approach,” *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 192–204, 2018.

[32] Y. Yuan, X. Luo, M.-S. Shang, “Effects of preprocessing and training biases in latent factor models for recommender systems,” *Neurocomputing*, vol. 275, pp. 2019–2030, 2018.

[33] N. Ranjbar Kermany, S.H. Alizadeh, “A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques,” *Electronic Commerce Research and Applications*, vol. 21, pp. 50–64, 2017.

[34] S. Wei, X. Zheng, D. Chen, C. Chen, “A hybrid approach for movie recommendation via tags and ratings,” *Electronic Commerce Research and Applications*, vol. 18, pp. 83–94, 2016.

[35] R. Forsati, M. Mahdavi, M. Shamsfard, M. Sarwat, “Matrix Factorization with Explicit Trust and Distrust Side Information for Improved Social Recommendation,” ACM Trans. Information Systems (TOIS), vol. 32, no. 4, p. 17, 2014.

[36] S. Ahmadian, M. Meghdadi, M. Afsharchi, “A social recommendation method based on an adaptive neighbor selection mechanism,” *Information Processing & Management*, vol. 54, no. 4, pp. 707–725, 2017.

[37] J. O’Donovan, B. Smyth, “Trust in recommender systems,” Proc, ACM 10th International Conference on Intelligent User Interfaces, pp. 167–174, 2005.

[38] L. Ye, C. Wu, M. Li, “Collaborative Filtering Recommendation Based on Trust Model with Fused Similar Factor,” MATEC Web of Conferences, EDP Sciences, p. 00010, 2017.

[39] N. Lathia, S. Hailes, L. Capra, “Trust-based collaborative filtering,” Trust Management II, Springer, pp. 119–134, 2008.

[40] C.-S. Hwang, Y.-P. Chen, “Using trust in collaborative filtering recommendation,” New Trends in Applied Artificial Intelligence, Springer, pp. 1052–1060, 2007.

[41] Q. Shambour, J. Lu, “A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services,” *International Journal of Intelligent Systems*, vol. 26, no. 9, pp. 814–843, 2011.

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

[43] L. Kaufman, P.J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, John Wiley & Sons, 2009.

[44] J. MacQueen, others, “Some methods for classification and analysis of multivariate observations,” Proc, Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297, 1967.

[45] P. Bedi, R. Sharma, “Trust based recommender system using ant colony for trust computation,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 1183–1190, 2012.

[46] L. Pradhan, C. Zhang, P. Chitrakar, “Multi-view Clustering in Collaborative Filtering Based Rating Prediction,” IEEE Tenth International Conference On Semantic Computing (ICSC), pp. 250–253, 2016.

[47] J. Zhang, Y. Lin, M. Lin, J. Liu, “An effective collaborative filtering algorithm based on user preference clustering,” *Applied Intelligence*, vol. 45, no. 2, pp. 1–11, 2016.

[48] N.S. Sani, F.N. Tabriz, “A New Strategy in Trust-Based Recommender System using K-Means Clustering,” *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 8, no. 9, pp. 152–156, 2017.

[49] M. Horridge, S. Bechhofer, “The OWL API: A Java API for OWL ontologies.,” *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

Faezeh Sadat Gohari received her B.S. degree in Software Engineering from University of Science and Culture, Iran, in 2011. She received her M.S. from K.N. Toosi University of Technology, Iran, in 2013. She is currently a Ph.D. candidate in Software Engineering at Shahid Beheshti University, Iran. Her research interests include recommender systems, semantic web, soft computing and data mining.

Email: f_gohari@sbu.ac.ir

Fereidoon Shams Aliee has received his PhD in Software Engineering from the Department of Computer Science, Manchester University, UK, in 1996 and his M.S. from Sharif University of Technology, Tehran, Iran, in 1990. His major interests are Software Architecture, Enterprise Architecture, Service Oriented Architecture, Agile Methodologies, Ultra-Large-Scale (ULS) Systems and Ontological Engineering. He is currently an Associate Professor at the Faculty of Computer Science and Engineering, Shahid Beheshti University, Iran. Also, he is heading two research groups, namely ASER (Automated Software Engineering Research) (aser.sbu.ac.ir) and ISA (Information Systems Architecture) (isa.sbu.ac.ir) at Shahid Beheshti University.

Email: f_shams@sbu.ac.ir

Hassan Haghighi is an Associate Professor at the Faculty of Computer Science and Engineering, Shahid Beheshti University, Iran. He received his Ph.D. in Software Engineering from Sharif University of Technology, Iran, in 2009. He is the head of the Software Testing and Verification Laboratory (ticksoft.sbu.ac.ir) at Shahid Beheshti University. His main research interests are software testing and formal methods.

Email: h_haghighi@sbu.ac.ir

Paper Handling Data:

Submitted: 8.07.2018
Received in revised form: 07.10.2018
Accepted: 15.10.2018
Corresponding author: Dr. Fereidoon Shams Alie
Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran