TRUST METRICS IN RECOMMENDER SYSTEMS: A SURVEY

MortezaGhorbani Moghaddam[1],Norwati Mustapha[1], Aida Mustapha[1]NurfadhlinaMohd Sharef[1], Anousheh Elahian[2]

[1]Department of Computer Science, University Putra Malaysia, Selangor, Malaysia
[2]Department of Information Technology,Virtual University of Shiraz, Shiraz, Iran

ABSTRACT

Information overload is a new challenge in e-commerce sites. The problem refers to the fast growing of information that lead following the information flow in real world be impossible. Recommender systems, as the most successful application of information filtering, help users to find items of their interest from huge datasets. Collaborative filtering, as the most successful technique for recommendation, utilises social behaviours of users to detect their interests. Traditional challenges of Collaborative filtering, such as cold start, sparcity problem, accuracy and malicious attacks, derived researchers to use new metadata to improve accuracy of recommenders and solve the traditional problems. Trust based recommender systems focus on trustworthy value on relation among users to make more reliable and accurate recommends. In this paper our focus is on trust based approach and discuss about the process of making recommendation in these method. Furthermore, we review different proposed trust metrics, as the most important step in this process.

KEYWORDS

Recommender Systems, Trust metric, Collaborative Filtering, Information overload, E-Commerce.

[

1. INTRODUCTION

Growth of the Internet has made possible for faster access to information as compared to the past. But the growing rate of information is very high. The digital data created reached 4 zettabytes in 2013 [1] and the future grow is staggering. Intel predicted that by 2015, it is the numbers of networked devices across the world will be two times higher than the global population. At that time, it would take 5 years to view all crossing IP networks at each second [2]. By 2017 it is predicted that mobile traffic will have grown 13 times in compare with 2012. At that time, there will be 3 times more connected devices than people on earth [1]. All of these data present that in the future it is impossible to physically follow the information flow in real time. This is essentially the phenomena of information overload, whereby there exist a huge amount of information over the Internet but finding the exact information of interest would be difficult. While the access to the information may be faster, it is not necessarily easier because finding the information ofinterest in the huge datasets is time consume and difficult.

To solve the problem, information should be filtered. One of the most important techniques of filtering of information is recommender systems (RS). RS have proposed to help and guide users to find items (e.g., sites, books, movies, news, music, etc.) of their interest from a plethora of available choices in huge pool of data such as the Internet [3]. RS are being used widely in modern e-commerce applications to cope with information overload problem, however they suffer from several inherent issues such as data sparsity, cold start users, low accuracy and malicious attacks.

DOI:10.5121/acii.2015.2301

As RS utilize behavior of users to make prediction about their interests, when there is not enough information about previous behavior of users the problem has been called cold start and causes the system not be able to make prediction properly. Sparsity problem refers to huge number of users and items in an e-commerce site. For example in Amazon.com has thousands users who are able to buy millions items. At this case, for each user, the ratio of the system about previous items of interest to whole number of items is very small. Evan for active users, with hundreds previous transactions, this ratio is not comparable. Sparcity problem causes that finding users with common tastes in a big dataset be difficult.

By introducing Web2.0 websites, which allow their user to interact to each other, Social-based recommender systems (SRS) as next generation of recommender systems have been proposed. SRS use relationship between users and social activities metadata to solve challenges of traditional RS and improve accuracy of recommendations[4]. Trust-based RS use trustworthy value in the process of making recommendations to improve accuracy of recommendation and solve problem of malicious attacks. However several proposed trust-based approach have targeted cold start and sparcity problems and achievements have been followed[3], [5]–[7].

Trust in RS is defined as one's belief toward others in providing accurate ratings relative to the preferences of the active user. To measure trust intensity between users, variety trust metric have been proposed.Some approaches have used explicit trust [8], [9], however others inferred implicit trust [10], [11].Furthermore, the calculated trust intensity in trust metrics may beglobal or local.In this paper we describe in detail, the process of making recommendation in Trust-based RS and focus on trust metrics as the most important step in the process. Furthermore, we have reviewed the most important trust-based approaches and have summarized their trust metrics.

The rest of this paper is organized as follows: properties of trust and process of making recommendation in trust-based RS are discussed in section 2. Section 3 describes the most important trust-based approach and review in detail the trust metrics in these approaches. Finally, the last section summarizes the paper.

2. THE PROCESS OF TRUST-BASED RECOMMENDATION

Recommender systems can generally be categorized into four types, which are Collaborative Filtering, content-based, demographic-based, and hybrid approaches. CF techniques are the most successful techniques for recommendation [3], [12]. This technique capitalizes on history of users' activities while disregarding the actual content of items. This technique requires users to indicate preferences in the form of ratings. Based on previous ratings, the system recommends items to user based on its understanding on users' interests and finding users with similar tastes [13].

In content-based methods, the system utilises the previous items of interest to understand the contents that are favourites of the user. These methods extract certain contents from items of interest and build relations between users and the extracted contents. Unlike collaborative filtering, which is not related to items, in content-based methods access to content of items is essential [14].

Demographic methods use similarity between users to make recommendation. In this respect, they are similar to collaborative filtering. The key difference between these methods is similarity metric. Collaborative filtering methods measure similarity between users based on their previous actions, however demographic-based approaches set demographic information of users as metrics to measure similarity.

Usually trust-based recommender systems place in the first category, where the trust intensity is used as an importance factor to measure the most similar users and aggregating their tastes for making prediction or recommendation. Later in this section, we discuss about the process of making recommendation in trust-based RS.

There are four steps in the process of trust-based recommendation. These steps are shown in Figure 1. The first step, that is the most important step and our focus in this paper, is measuring the trust. The output of this step is a $|U| \times |U|$ matrix, where $|U|$ is number of users and content of cell u, v presents the trust intensity between users u and v. Based on characteristics of trust this number may be binary or floating point, positive or negative. In some approaches such as [8], [9], [15]–[17] the trust value between some users are defined explicitly. At this case the system fills corresponding cells with explicit values and calculates trust intensity for other cells. On the other hand, there are implicit trust approaches, which infer value of all cells of the matrix. In these approaches, the trust value between users is not defined explicitly by users and the system infers them based on social behavior of users. Although accuracy of explicit trust is more than implicit one, when explicit trust is not exists, using of implicit trust-based approaches in comparison with traditional RS can improve accuracy of recommendation results[10], [11].

The trust matrix is sparce. It means that many cells in the matrix are empty. To solve this problem usually trust propagation methods are used. By trust propagation methods, the system calculate the trust intensity between users based on non-empty cells of the trust matrix. Be note that there is different between implicit inferring trust and trust propagation. For inferring implicit trust, the system focuses on social behavior of users, however for propagating of trust, it focuses on previous calculated trust values. For this reason usually trust relationship between users present as a graph, called trust graph. Trust graph is a two dimensional graph, that is constructed based on the trust matrix. In this graph users are presented as nodes and trust value between users are presented as edges. Furthermore, the trust intensity between users is used as weight of the edge. If all cells of the trust matrix are non-empty therefore the graph is complete. It means that trust intensity between all users are calculated. But most of time the graph is not complete, even it may not be connected. Trust propagationis one of the most important sub processesthat calculates trust intensity on missed edges.

Neighbors are the most related users to the active users, that the system utilizes their information to make prediction about interests of the active users. Similar to traditional collaborative filtering approach, the second step in trust-based approaches is selecting the neighbors. But there is an important difference here. In traditional approaches neighbors are chosen based on similarity, however in trust-based approaches the trust value is key parameter.

One intrinsic problem of collaborative filtering approaches is malicious attacks. This problem refers to this step because neighbors can affect recommendation results. In collaborative filtering approaches, the system measures similarity between users based on previous common tastes. If an attacker copy activities of the active users, therefore the system choose him/her as the most similar users. It means that he/she is able to drive recommendations by using his next rates. To solve this problem trust-based approaches uses calculated trust value between users to choose more reliable users as neighbors. Previous researches presented that user with high trust intensity

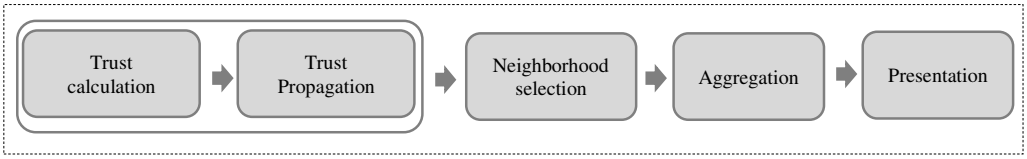

Figure 1. Steps of recommendation in trust-based approaches

may have low similar tastes[3], [18]. Therefore usually combination of similarity and trust is used in selecting neighbors to improve accuracy of recommendation and solve malicious attacks [3], [5], [6].

The third step is aggregating tastes of neighbors. At this time the system predicts interest of the active user *u* about non-rates item *i*. for this reason it chooses the neighbors who have rated the items *i* previously and aggregates their rates:

$$\hat{r}(u,i) = Aggr_{\ v \in Neighbors}(r(v,i)) \tag{1}$$

Where $\hat{r}$ and $r$are predicted and actual rates respectively. One of the most common aggregation formula, is Resnick[19] that is formulate as equation 2.

$$\hat{r}_{Resnick}(u,i) = \bar{r_u} + \frac{\sum(r(v,i) - \bar{r_v})W(u,v)}{\sum|W(u,v)|} \tag{2}$$

Where $\bar{r}$is average previous rates of user *v* and $w(u,v)$ is weight that presents importance of *v* on *u*'s point of view. As discussed before, in trust-based approach w is combination of similarity and trust intensity.

The final step in making recommendation is showing the results. This step create output results that may be: prediction, recommendation or ranked list. In prediction, the input is the active user and the active item. At this case the system should predicts interest of the user about the item. But in recommendation, the input is only the active user. At this case the output is a list of items that are more related to the user. The number of items in the list may be define as an input threshold. In other hand, the input threshold is minimum interest value. At this case the system predicts interest of the user to all items and shows only the items that there $\hat{r}$ is greater than the input threshold. If output result of recommendation is a sorted list of items based on predicted rates, the output is a ranked list of items.

3. TRUST METRICS

As discussed in previous section, the method of calculating and propagating trust is very important and affects result of recommendation. In this section we review in detail the different trust metrics in trust-based approaches. Also main contribution of discussed approaches are summarize in Table 1 at the end of this section.

Advogato[20]has been proposed by Levin in order to discover the most trustworthy users. The main goal of Advogato's has been attack resistance. It does not make prediction or recommendation and just find the most trustworthy users however there is no distinction between the trusted users in this approach. Therefore it is not appropriate for recommendation. Advogato transforms the trust network into a flow network. This transformation is the essence of the algorithm. The computation of the trust metric is performed relative to a seeds of trustworthy users, a limited count of users who explicitly have defined as base users. Advogato uses a breath-first search algorithm from the seed to assign capacities to nodes. For this reason it uses the distance from the seeds (D_u) and the out-degree of nodes with a smaller distance (if any).

$$Trust^{Advogato}(u) = D_u \tag{3}$$

Ziegler proposed Appleseed [21] in his Ph.D. thesis. His approach is similar to Advogato. But here, edges in trust graph are assigned continuous weights. In contrast to Advogato, Appleseed uses is suitable for distributed implementation [22]. It models trust as energy and propagates it

using spreading activation mechanism. The output of the algorithm is the top-N nearest trust neighbors for every user. The novelty of this approach lies in the propagation step, whereby a source node u is activated through an injection of energy. The energy then propagates to other nodes through edges of the graph. The ratio of energy propagates on each edge is related to the weight of the edge. Therefore, the more direct trusted users in the network, receive higher level of energy. AppleSeed supposed that trust is additive. Therefore, the amount of energy received by a user is the sum of total energy that travels through variety of paths. If there are many weakly trusted paths between two users, this pair of users will obtain a high trust value. Equation 4 presents trust metric in Appleseed. Where $in(u)$ is set of nodes who there is an input link from them to u. $eng_s(v,u)$ presents the propagated energy from v to u when the source energy injected in node s.

$$Trust^{AppleSeed}(s,u) = \sum_{v \in in(u)} eng_s(v,u) \tag{4}$$

Golbeck proposed TidalTrust[8] as a recommender system that used explicit trust to make move prediction. Like Appleseed, main contribution of TidalTrust is related to trust propagation, where it only considers the paths of the shortest length, disregarding others. Among the shortest paths, TidalTrust calculates the maximum of the minimum edge weights on a path, Smaxmin. The minimum of the edge weights on a path is therefore considered as a measure of weight of this path, as in Advogato. After the path selection process, TidalTrust again uses multiplication to calculate the weight of a path from its edge weights, as in Appleseed. Smaxminis used as a threshold value to select the best paths. The paths that will be considered in the final trust computation are only the shortest paths from with a minimum of the edge weights that is not less than Smaxmin. Let T^S be set of trustworthy users of s whom the trust value is greater than or equal to the Smaxmin, and $Trust^{Explicit}(s,v)$ be explicit trust between s and v. Therefore Golbeck used below recursive formula for trust propagation:

$$Trust^{TidalTrust}(s,u) = \frac{\sum_{v \epsilon T^S}(Trust^{Explicit}(s,v) \times Trust^{TidalTrust}(v,u))}{\sum_{v \epsilon T^S}(Trust^{Explicit}(s,v))} \tag{5}$$

Kuter and Golbeck proposed SUNNY [16] as a trust-based recommender system. The process of propagation trust in SUNNY is similar to TidalTrust. The key difference lies in the selection of paths that contain specially selected neighbours of destination users. In SUNNY each neighbour of destination node B has one of three states: “unknown”, “include” or “exclude” from the set of paths used for the final trust calculation. In the beginning, the Bayesian network is used to calculate bounds on the confidence of S in B’s recommendation when all neighbours of B have an “unknown” state. After this initial step, Sunny iterates through all neighbours of B and tries to change their state to “include” and again calculates confidence bounds for the belief of S in a recommendation from B. If the resulting confidence bounds are too different from the original confidence bounds, then the state of this neighbour of B is changed to “exclude”. Finally, only the neighbors of B that do not change the original confidence bounds too much are selected.

For this reason, SUNNY calculates differences between users at three levels: Overall difference, Difference on extremes and Maximum difference. It also calculates confidence between users based on Pearson Correlation Coefficient (PCC) similarity metric. To estimate trustworthy from source node S to destination node B, it performs a variant of backward breadth-first search over the nodes of the trust network towards the source node S and constructed a Bayesian Network by using the trust network, S and B. SUNNY computes conditional probabilities for every edge in the Bayesian Network based on computed confidences and levels of differences between users.

Massa and Avesani proposed MoleTrust[23] as a trust metric that is similar to TidalTrust in the sense that it also uses explicit trust. The major difference lies in the propagation method. While TidalTrust used depth-first search for finding the highest trust users, MoleTrust used breadth-first search. In this approach, a maximum-depth is specified as an input and all raters will be considered up to the maximum-depth.

MoleTrust includes two steps: Purpose of the first step is to destroy cycles in the graph. Second step consists of a graph walk starting from source node with the goal of computing the trust value of visited nodes. For this reason the initial trust value of the source user is set to 1 and to compute the trust value between u and v, MoleTrust performs a backward exploration, whereby the trust value from u to v is the aggregation of trust values between u and users directly trusting v is weighted by the direct trust values.

$$Trust^{MoleTrust}(s,u) = \frac{\sum_{v \epsilon predecessors}(Trust^{Explicit}(v,u) \times Trust^{MoleTrust}(s,v))}{\sum_{v \epsilon predecessors}(Trust^{MoleTrust}(s,v))} \quad (6)$$

Here, intensity of trust is depend on predicted trust for users in previous level ($predecessors$) and trust statement ($Trust^{Explicit}$) between them to the node u. For predicting the trust value of a user, MoleTrust analyses all the incoming trust edges and accepts only the ones coming from users with a predicted trust score greater or equal than a certain threshold.

A Trust-aware Recommender System (TaRS) is also proposed by authors of ModelTrust in [17]. The trust metric in TaRS is same as MoleTrust and difference lie in aggregation step, where TaRS uses combination of trust and similarity as weight.

In [11], trust is defined and used differently whereby the trust value is represented in terms of percentage of correct predictions. This research proposed a new metric for inferring global implicit trust by distinguishing two types of trust for each user. The first type of trust is item-level trust, which is measured by perceived trustworthiness of the user to each item. Item-level trust is calculated based on ratio of correct predictions of the user to all of his/her predictions about the item. Suppose that $Predicts(u,i)$ is set of all prediction of user u about i. some of these prediction have been correct and other have been incorrect. If $Predicts^{Correct}(u,i)$ be set of correct predictions, at this case item-level trust is formulated as below equation:

$$Trust^{Item}(u,i) = \frac{|Predicts^{Correct}(u,i)|}{|Predicts(u,i)|} \quad (7)$$

O'Donovan and Smyth proposed also profile-level trust similar to item-level trust: For each user, ratio of correct predictions to all predictions, independent of items, introduced as profile-level trust.

$$Trust^{profile}(u,i) = \frac{|Predicts^{Correct}(u)|}{|Predicts(u)|} \quad (8)$$

Experiments in [11] presented that accuracy of item-level is 22% more than collaborative filtering. However this value for profile-level trust is around 15%. Furthermore experiments in [11] showed that using of combination of trust and similarity in aggregation step, improves accuracy of results.

Zhang and Xu proposed Topic-level trust in [10]. This method is similar to item-level trust [11] in the sense that it also uses correct prediction to measure implicit and global trust. But it uses topics in taxonomy to calculate trustworthiness.

The trust metric in [10] is formulated as below equation, where $Trust^{Topic}(u,t)$ denotes trust intensity of user u about topic t, $Trust^{item}(u,i)$ denotes items-level trust and $Items(t)$ is set of items that are related to topic t.

$$Trust^{Topic}(u,t) = \frac{\sum_{i \in Items(t)} Trust^{Item}(u,i)}{|Items(t)|} \quad (9)$$

Zhang and Xu also proposed new method to calculate item-level trust. by. They used ratio of sum of trust value to number of raters (Equation 10):

$$Trust^{Item}(u,i) = \frac{\sum_{v \in Raters(i)} (1 - \frac{|\hat{r}(u,i,v) - r(u,i)|}{S})}{|Raters(i)|} \quad (10)$$

Where $\hat{r}(u,i,v)$ is the predicted rate by v about interest of u to i, S represents the rating scale, and $Raters(i)$ denotes set of users, except u, who have rated item i.Experiments in [10] showed that topic-level trust produces lower error (around 13%) than traditional RS.

Jamali and Ester proposed TrustWalker[6] as a random walk model that combines trust-based and item-based CF approaches to further improve prediction accuracy and solve cold-start as well as sparsity problems. This method uses explicit friendship to calculate local trust.Like Appleseed, the trust metric in Trustwalker is related to out-degree of nodes, but here all edges have equal weight. Suppose that there is an edge between u and v, thus TrustWalker calculates trust value between the nodes by using below equation:

$$Trust^{TrustWalker}(u,v) = 1/|out(u)| \quad (11)$$

Where $out(u)$ is set direct friends of u. The novelty of TidalTrustis on the proposed random walk model.In this approach, to predict the rate of user u about item i, the system uses direct friends. If they have rated item i before, the system returns the rate. Otherwise, the system returns one of previous rated items based on probabilistic calculation or asks their direct friends. The proposed probabilistic model in [6] is related to maximum similarity of items that are being rated by current user.

Given that size of the set of common users is also important Jamali and Ester proposed a new similarity metric to affect$|CU_{i,j}|$, as number of common users who have rated to both items, in computing similarity between users. They used a sigmoid function based on Pearson Correlation Coefficient (SPCC) to avoid favoring the size of common users too much:

$$sim(i,j)^{SPCC} = sim(i,j)^{PCC} . \frac{1}{1 + \exp(-\frac{|CU_{i,j}|}{2})} \quad (12)$$

If the size of the set of common users is big enough, then the second part of the equation would converge to 1, but for small sets of common users, the factor would be 0.6. The number 2 in the denominator of the exponent is because they wanted to have a factor of greater than .9 if the size is greater than 5. Based on experiments in [6], Jamali and Ester claimed that considering the size of the set of common users in the item similarity metric reduces the error.

The experiments on Epinions data set showed especially for cold-start users, who have only a few ratings, TrustWalker achieved better results in comparison with other methods. For example in comparison with traditional collaborative filtering, it improved accuracy of recommendation around 20%, while it increased the coverage more than three times. Jamali and Ester compared their proposed approach with other trust-based approaches, too. The result presented that its improvement on accuracy was around 2% and 17% for TidalTrust and MoleTrust respectively, while the improvement on coverage was more than 20% for both compared approaches.

A modified version of TrustWalker, called MWalker[5], was proposed by Jin and Chen. MWalker used multiplication of binary friendship, as explicit trust, and similarity between users to calculate trust value.

$$Trust^{MWalker}(u,v) = Trust^{Explicit}(u,v) \times Sim^{Tag}(u,v) \quad (13)$$

MWalker used a breadth-first search method to make recommendation. For this reason it starts from source node u and sorts his/her friends according to the trust value between them. Top-K most trustworthy users compose the candidate set. The algorithm visits each user in the candidate set sequentially. If the user rated to the target item i, then returns it directly. Otherwise the user returns his/her rate about item j that is the most similar item to i. The described algorithm continues for all candidate users till maximum defined depth.

SimTrust[24] has been proposed by Bhuiyan et al. as a recommender system. In SimTrust, trust has been defined and used in a different way. In this method, the implicit local trust is inferred using tags based on the assumption that a trustworthy text description about each item exists. Therefore, to solve ambiguity problem of tagging, the system is able to extract the semantic meaning of a tag based on the description of the items. Bhuiyan et al. used tf-idf as a text mining approach to extract frequent keywords from description of items. Suppose that Wut is set of frequent keywords about tag t that is used by user u. Actually Wut represents the semantic meaning of the tag t as point of view of user u. SimTrust measured trust value between users by using following equations:

$$Trust^{SimTrust}(u,v) = \left(\sum_{k \in W} \frac{n_{uv}^{k}}{n_{v}^{k}}\right) \Big/ |W| \quad (14)$$

Where n_{uv}^{k} denotes number of tags in intersection of Wu and Wv that contain keyword k. W is set of all keywords in Wu or Wv.

Bhuiyan et al. evaluated their approach by using Amazon book data set for different neighborhood size. Result showed increasing the neighborhood size, improve the recommendation results. They also compared SimTrust with TidalTrust and traditional collaborative filtering. The result presented that SimTrust performed better that compared approaches. It is concluded that the method is helpful in dealing with data sparsity problem.

Guo et al. proposed Merge [3] as a trust-based method to solve cold-start and sparsity problems. Instead of using direct rated items for prediction, Merge uses extended rated items. Extended items are the items being rated by the user explicitly or by at least one of his/her direct friends. Suppose that item i is a member of the extended set for active user u. If the user u rated directly the item i, therefore his/her previous rate will be saved. Otherwise, aggregation of rates of direct friends will be used as the predicted rate for the item in extended list. We call these type of rates, which are aggregated based on rate of direct friends, merged-rates.

The trust metric in Merge is same as MoleTrust. But the contribution of Merge is in choosing aggregation section, where it uses a linear combination of trust, rating similarity and social similarity as importance weight:

$$W(u,v) = \left(\alpha \times Sim^{Rating}(u,v)\right) + \left(\beta \times Trust^{MoleTrust}(u,v)\right) + ((1-\alpha-\beta) \times Sim^{Social}(u,v)) \quad (15)$$

The experimental results on three real-world datasets, FilmTrust, Flixer and Epinion, showed a significant improvement against other methods in terms of both accuracy and coverage, as well as the overall performance. In addition, this approach has performed well in cold-start problem.

Ray and Mahanti focused on trust propagation. They analyzed Epinion dataset and observed that majority of the trust statements passed between users cannot be seen as a reflection of similarity between the two users. They proposed RN [18] as a trust-based approach, to reconstruct the trust networks in order to improve the prediction accuracy. This is done by using combination of trust and similarity between users as the weight of trust in trustworthy graph. The approach then pruned the graph based on a threshold correlation value. They removed all edges in the original trustworthy graph is the correlation value between two users is less than the defined threshold. This approach constructs the network at different propagation levels. For this reason it uses a linear decay approach for propagating trust. The formula used for the trust metric is $(d - n + 1)/d$ where d is the maximum propagation distance and n the distance of the target user from the source user.

Experiment presented that reconstructing the trust network improve accuracy of recommendation in comparison with complete trust network. However RN achieved poor coverage and it failed to function in cold conditions where user similarity may not be computable.

A Trust-based Ant Recommender System (TARS) that uses dynamic trust for prediction has been proposed in [25] by Bedi and Sharma. TARS focuses on sparsity and cold-start problems and uses trust graph as well as biological metaphor of ant colonies to choose the best neighbors. In order to measure initial trust value ($Trust_0$) between users, TARS uses a combination of similarity and confident as below equation:

$$Trust_0(u,v) = \begin{cases} \dfrac{2 \times sim(u,v) \times conf(v|u)}{sim(u,v) + conf(v|u)} & if\ sim(u,v) \neq 0 and\ conf(v|u) \neq 0 \\ k \times conf(v|u) & if\ sim(u,v) = 0 and\ conf(v|u) \neq 0 \\ 0 & if\ sim(u,v) = 0 and\ conf(v|u) = 0 \end{cases} \quad (16)$$

Where $conf(v|u)$is amount of confidence a user u should have on user v, which is calculated as ratio of number of co-rated items.

In the second phase, TARS uses the user's constructed trust network with several ants that are placed in root of the network at initialization time. The ants move around the network and find the most trustworthy users. To selecting neighbors, they use trust valueson edges, as pheromone, and level of connecters from source node to transfer.

After aggregating, which is calculated by Resnick's formula, updating the pheromone values is final step. In TARS the trust value at time t, is calculated as below equation:

$$Trust_t(u,v) = (1-\rho) Trust_{t-1}(u,v) + \frac{T\_traced_v \times \prod_{i=1}^{d_{sv}} Trust_{t-1}(i,v)}{T\_unrated_v \times d_{sv}} \quad (17)$$

Where ρ is evaporation rate, d_{sv} is the level of connectedness from the source node S to v,T_tracedv denotes the number of items traced by user v which are unrated by S, andT_unrateds denotes the total number of items unrated by S.

Authors of TARS, experimented their proposed approach by using two real datasets, Jester and MoveLenz. As it was expected, the result presented that increasing the number of neighbors, decreases the average quality of the recommendations. Also comparison between TARS and traditional collaborative filtering showed, TARS overcomes traditional CF especially over time, when it updates trust value. Furthermore, TARS identified popular users with high DTP as default trusted friends. It claimed thatto solve cold-start problem, the list of default trusted friends can be captured and automatically added to the new user's list at initial stage.

AgeTrust[26]focused on aggregation method and proposed a temporal weighting method. This research claimed that older friends are more reliable than new one. Therefore they used age of friendship to rank trusted users. It means that in AgeTrust, the importance weight of older trusted usersaremore than newer ones. Suppose that user u has n friends and position of user v in the rank list, based on time of friendship, is p. At this case the importance weigh is calculated as bellow equation:

$$W(u,v) = Sim(u,v) \times 1 - \left(p * \left(\frac{1-\alpha}{n-1} \right) \right) \times Trust(u,v) \quad (18)$$

Where $\alpha \epsilon [0,1]$is a minimum trust value which is assigned to newest friend, n denotes number of direct friends of user u. To validate the proposed approach, [26] used Delicious dataset and calculated preference of user to items based on number of assigned tags. Comparison against the traditional collaborative and pure friendship-based approaches showed that the prediction accuracy of AgeTrust, is more than other approaches. It means there is a positive relation between the age of trust and the trust value.

Main contributions of discussed approaches in this section are summarize in Table 1.

Table 1. Main contribution of trust-based approaches

Trust-based approaches	Trust measurement		Neighborhood selection	Aggregation
	Trust calculation	Trust propagation		
Advogato		✓		
AppleSeed		✓		
Item-level trust	✓			
Profile-level trust	✓			
TidalTrust		✓		
SUNNY		✓		
MoleTrust		✓		
TaRS				✓
Topic-level trust	✓			
TrustWalker			✓	✓
SimTrust	✓			
RN		✓		
TARS			✓	
MWalker				✓
Merge			✓	
AgeTrust				✓

4. SUMMARY

Traditional recommender systems suffer from several inherent issues such as data sparsity, cold start users, low accuracy and malicious attacks. Social recommender system use metadata such as relation among users and social behaviour of users to solve these problem. Trust-based recommender systems are the most successful approaches in this generation. They utilizes trust to create more accruable and reliable predictions. Trust in recommender systems is defined as one's belief toward others in providing accurate ratings relative to the preferences of the active user.

In this paper we discussed on process of making recommendation in trust-based approaches. We categorized this process to four steps: Trust measurement, selecting the neighbours, aggregation and presentation. Furthermore, we reviewed in detail variety trust-based approaches and argued about different steps in these methods. Also, the main characteristics of trust, such as globality and visibility have been discussed in different approaches. Aware of this review, the most approaches have used explicit trust rather than implicit one. It may be due to accuracy of explicit trust that is more than implicit one. However to solve sparcity problem, propagating of trust is effective and imperative. Therefore different approaches used various propagation methods.

Based on reviewed methods, accuracy and coverage of trust-based approaches is better than traditional recommender system. Furthermore, these methods achieved significant improvements about cold start data, when there is not enough information about behaviour of users. It seems that focus of first trust-based approaches were more on trust metrics, however recent approaches are more related to selection of neighbours and aggregation.

REFERENCES

[1] "Internet-minute-infographic. (2014) What Happens in an Internet Minute?. Retrievef from August 3, 2014, http://www.intel.com/content/www/us/en/communications/internet-minute-infographic.html." .

[2] "Internet-minute-infographic. (2011) What Happens in an Internet Minute?. Retrievef from January 23, 2012, http://www.intel.com/content/www/us/en/communications/internet-minute-infographic.html." .

[3] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," Knowledge-Based Syst, Dec. 2013.

[4] N. Ghorbani Moghaddam, Morteza Mustapha, A. Mustapha, N. Mohd Sharef, and A. Elahian, "A Review on Trust-based Recommender Systems."

[5] J. Jin and Q. Chen, "A trust-based Top-K recommender system using social tagging network," in Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), 2012, no. Fskd, pp. 1270–1274.

[6] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 397–406.

[7] G. Guo, J. Zhang, and D. Thalmann, "A Simple but Effective Method to Incorporate Trusted Neighbors in Recommender Systems," in 20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2012), 2012, no. Imi, pp. 114–125.

[8] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in proceedings of the 4th international conference on Trust Managementi (Trust'06), 2006, pp. 93–104.

[9] P. Massa and P. Avesani, "Trust metrics on controversial users: Balancing between tyranny of the majority," Int J Semant Web Inf Syst, vol. 3, no. 1, pp. 1–21, 2007.

[10] Z. Fu-guo and X. Sheng-hua, "Topic-level Trust in Recommender Systems," in 2007 International Conference on Management Science and Engineering, 2007, pp. 156–161.

[11] J. O'Donovan and B. Smyth, "Trust in recommender systems," in Proceedings of the 10th international conference on Intelligent user interfaces - IUI '05, 2005, p. 167.

[12] C. Li, L. Ma, and K. Dong, "Collaborative Filtering Cold-Start Recommendation Based on Dynamic Browsing Tree Model in E-commerce," in 2009 International Conference on Web Information Systems and Mining Collaborative, 2009, no. 2, pp. 620–624.

[13] M. Ghorbani Moghaddam, N. Mustapha, A. Mustapha, N. Mohd Sharef, and A. Elahian, "A Review on Similarity Measurement Methods in Trust-based Recommender Systems," in 9th International Conference on e-Commerce with focus on E-Trust (ECDC 2014), 2014.
[14] M. Qin, Q. Yang, and F. Fu, "A context-aware media content personalized recommendation for community networks," 2013 Int Conf Wirel Commun Signal Process, no. 2, pp. 1–6, Oct. 2013.
[15] J. Golbeck, "Personalizing applications through integration of inferred trust values in semantic web-based social networks," in Proceedings of Semantic Network Analysis Workshop, 2005, pp. 15–28.
[16] U. Kuter and J. Golbeck, "Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models," in AAAI'07 Proceedings of the 22nd national conference on Artificial intelligence, 2007, vol. 2, pp. 1377–1382.
[17] P. Massa and P. Avesani, "Trust-aware recommender systems," in Proceedings of the 2007 ACM conference on Recommender systems, 2007, pp. 17–24.
[18] S. Ray and A. Mahanti, "Improving Prediction Accuracy in Trust-Aware Recommender Systems," in the 43rd Hawaii International Conference on System Sciences, 2010, pp. 1–9.
[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94, 1994, pp. 175–186.
[20] R. Levien, "Attack resistant trust metrics," UC Berkeley, 2004.
[21] C. Ziegler, "Towards decentralized recommender systems.," 2005.
[22] A. Wierzbicki, Trust and Fairness in Open, Distributed Systems. 2010, p. 244.
[23] P. Massa and P. Avesani, "Trust metrics on controversial users: Balancing between tyranny of the majority," Int J Semant Web Inf Syst, vol. 3, no. 1, pp. 1–21, 2007.
[24] T. Bhuiyan, Y. Xu, A. Jøsang, H. Liang, and C. Cox, "Developing trust networks based on user tagging information for recommendation making," in Web Information Systems Engineering – WISE 2010 LNCS, 2010, pp. 357–364.
[25] P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation," Expert Syst Appl, vol. 39, no. 1, pp. 1183–1190, Jan. 2012.
[26] M. G. Moghaddam, N. Mustapha, A. Mustapha, N. Mohd Sharef, and A. Elahian, "AgeTrust : A new temporal trust-based collaborative filtering approach- In press," in the proceeding of 5th International Conference on Information Science & Applications (ICISA' 14), 2014, pp. 1–4.