Augmenting Trust Networks for Improved Recommendation Generation: A T-index Approach

Soude Fazeli[1], Alireza Zarghami[2], Nima Dokoohaki[1], and Mihhail Matskin[3]

[1] Royal Institute of Technology (KTH)
[2] University of Twente (UT)
[3] Norwegian University of Science and Technology (NTNU)
{soude,nimad}@kth.se[1] a.zarghami@utwente.nl[2] misha@imit.kth.se[3]

Abstract. Social Networks have dominated growth and popularity of the Web to an extent which has never been witnessed before. Such popularity puts forward issue of trust to the participants of Social Networks. Collaborative Filtering Recommenders have been among many systems which have begun taking full advantage of Social Trust phenomena for generating more accurate predictions. For analyzing the evolution of constructed networks of trust, we utilize Collaborative Filtering enhanced with T-index as an estimate of a user's trustworthiness to identify and select neighbors in an effective manner. Our empirical evaluation demonstrates how T-index improves the Trust Network structure by generating connections to more trustworthy users. We also show that exploiting T-index results in better prediction accuracy and coverage of recommendations collected along few edges that connect users on a network.

Keywords: Social Networks, Social Trust, Recommendation, Collaborative Filtering, Trust networks, Ontological modeling, Performance

1 Introduction

Semantic Web vision noted trust as one of the most crucial technologies enabling a future Web of openness and collaboration, collectively referred to as "'Web of Trust"' [1]. Emergence of Social Networks and most importantly Web-Based Social Networks (WBSN)[2] from one side, and research into trust from the other side, combined with Semantic Web technologies created an exclusive opportunity to merge existing efforts and create means for Social Network Analysis (SNA) at the top of Semantic Web [3]. Among many systems which have realized the impact of so called "Social Trust", Recommender Systems have been the most influential ones. Recommenders are software systems which retrieve data items on users behalf, by taking into account similarity between users interests (social or collaborative based), or just by considering similarity between items (content-based), or by considering both item and user similarity (hybrid). Social(also known as Collaborative Filtering) Recommenders can be enhanced to take into account the trust relation in-between users, in order to provide users with better suggestions.

In this paper, we propose a mechanism augmenting a Social Recommender. We refer to this approach as T-index which digests trust values between users on the Trust Networks in order to provide more improved recommendations to users. To do so, we introduce a T-index measure inspired by H-index[4] to discover the users within our trust network who provide trust values higher than or equal to *T*, for number of users larger than *T*. Therefore, more users, from divergent areas of users' preferences, might be accessible within few edges of the path that connects users on the network. We demonstrate the applicability of this approach in the context of a movie recommender. Initial results with a large extent of users have proved our hypothesis. The rest of the work is documented as follows: Section 2 provides the background and related works. Section 3 describes our approach and then, Section 4 shows our experimental results and discussions. Finally, we conclude and present an overview of the future work in Section 5 .

2 Background

Users are able to express information about their relationships such as how much they can rely on people On a Social Network[5]. This phenomena leads us to the social notion of trust which helps recommendations to be generated from trustworthy partners [6]. FOAF (Friend-of-a-Friend) vocabulary [7] describes users' information and their social connections through concepts and properties in the form of an ontology using Semantic Web technologies [8], [5]. Golbeck[5] introduces an ontology that extends FOAF vocabulary for modeling trust relationship between users. Although Golbeck's ontology provides an efficient structure, every relationship describes only one subject. Dokoohaki et al. [9] present an ontology for modeling structure of trust relations between users that is more efficient in terms of the size of the generated networks using ontology. We extend this ontology to model trust between users with an extra element for measuring T-index-based trustworthiness of a user.

Recommender Systems which are extended with trust phenomena have proven to provide users with more reliable recommendations. Massa and Avesani[10] introduce an architecture for a trust-aware recommender capable of trust aggregation for all of the users on a network. Andersen et al.[11] propose an axiomatic approach in which the users' opinions are aggregated in trust networks to generate personalized recommendations. The methods mentioned above are all limited to some explicit trust rating to infer other trust relations. Some efforts have been made to formalize the trust where it can not be explicitly expressed by users. O'Donovan and Smyth [12] represent computational models of trust in which a recommender's rating is correct if the difference between its rating and the target user's rating is less than a predefined value. Lathia et al.,[13] propose a trust-learning method that is similar to the models presented by O'Donovan and Smyth in [12]. The main idea is that the recommenders, who provide useful information, should be rewarded and those who have no information available, should be downgraded. The trust-based collaborative filtering algorithm used in their method requires a centralized user-item matrix which might lead to scalability problem as the number of users increases. Weng et al.[14] assume each user as a peer connected to other users in a decentralized trust network of users. The trust between two users is computed based on the "Goodman-Kruskal measures of association of cross classifications"[15]. In this paper, we adapt the formalization proposed by Lathia et al.[13] to derive the trust value between users. We introduce an agent-setting in which every user is considered to be an agent connected to other users to form a trust network. Such a setting should provide better scalability since the distributed allocation of trust-related data is supported.

3 A Semantic Trust-ware Recommendation Framework

Our goal is to create trust relationships among all types of users with respect to different types of items, accessible through unique URI across heterogeneous networks and environments. To achieve this, we have developed an ontological framework, shown in Fig. 1, composed of three main modules: Semantic Profile Manager, Trust Engine and Recommendation System.

Upon rating an item by a user, the Semantic Profile Manager module either creates or updates an ontology-based profile for both user and item.

The Trust Engine module generates a so-called *trust network* of users based on the profile information of users and items in a distributed manner. To do so, a user profile extends the trust ontology to keep top-n neighbors and its mutual trust values with them. Note that there is no global view of a trust network for users and they are only provided with information regarding their neighbors and rating history. Therefore, it is possible to maintain users in different groups on several servers to achieve better scalability. To cope with privacy requirements, these servers can be located in different organizations while profiles of users and items are accessible only through their URI.

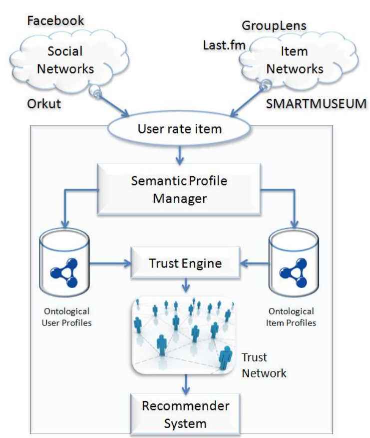


Fig. 1. Ontological Framework

The Recommendation System module enables traversals through the trust network to collect recommendations for a target user and finally makes a predicted rating for the user.

The whole model is built on top of a knowledge acquisition system to improve manipulation of ontological data. The presented ontological framework provides us with high interoperability and openness to deal with heterogeneous networks.

3.1 TopTrustee and T-index

In order to build trust relationships among users, we enhance Collaborative Filtering with two novel concepts: T-index and TopTrustee.

T-index The H-index [4] was defined by Jorge E. Hirsch, a physicist, "'as the number of papers with a citation number higher or equal to *H*, as a useful index to characterize the scientific output of a researcher"'. Extending this idea, we propose an estimate of a user's trustworthiness called T-index, similar to the H-index in showing the number of trust relationships between a user and its trusters with trust value higher than or equal to *T*. T-index can be introduced as Indegree of nodes in a trust network which provides not only number of incoming edges as a regular Indegree, but it also considers the weight of incoming trust relationships. For a node on a network, Indegree represents the number of head endpoints adjacent to a node while Outdegree is the number of tail endpoints.

Algorithm 1 Computing T-index

procedure ComputeT-index $\langle user, TrusterList \rangle$
    $TrusterValueList \leftarrow TrusterList$.sort($trustValue, descending$)
    for all $trustValue$ **in** $TrusterValueList$ **do**
        $trustValue \leftarrow$ multiply($trustValue, Max_{T-index}$).rounded
    end for
    $Counter \leftarrow 1$
    for all $trustValue$ **in** $TrusterValueList$ **do**
        if $Counter < trustValue$ **then**
            $Counter \leftarrow Counter + 1$
        else
            break
        end if
    end for
    $T\text{-}index \leftarrow Counter - 1$
    return $T\text{-}index$
end procedure

The algorithm 1 describes how T-index is computed for a user. First, we introduce the maximum value of T-index as a global variable which defines the precision of T-index computation. Thus, we multiply all trust values (shown as label of arrows in Fig. 2) by this maximum value. In our work, trust value is in the range of 0 to 1. Then, we start to count the number of trusters until the counter becomes greater than the trust values.

In this work, we define cluster as a group of users who all trust a common user, called Centric User as the most trustworthy one within the cluster. Fig. 2 shows u_a and u_f as centric users of two clusters.

Item's TopTrustee An item's *TopTrustee* is a user who has already rated the item and can join item's *TopTrustee* list if its T-index value is higher than a certain threshold. In fact, *TopTrustee* list introduces trustworthy users to the user who has just rated the item. The users in *TopTrustee* list may have no trust relationship with the user yet as they can not be reached through the maximum path length of *L*. However, They might be a source of useful information for the item's rater. We form *TopTrustee* lists by exploiting T-index.

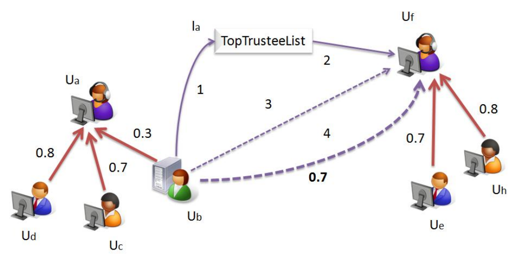


Fig. 2. A scenario of utilizing TopTrustee List

As shown in Fig. 2, when u_b rates item i_a, its mutual trust values with all users in two sets are computed and updated. The first set is its *top-n neighbors* as the first *n* users who are not only directly connected to the user but also provide the highest mutual trust values with the user. The other set is the item's *TopTrustee* list. The arrows between the users and the *TopTrustee* list show that the users rated i_a. u_f has rated i_a and is already located in i_a's *TopTrustee* list. After computing the trust value between u_b and u_f based on the trust formula presented by [13], u_b finds u_f more trustworthy than u_a as one of its current top-n neighbors even though u_f is not accessible to u_b within path length of *L*. Eventually, u_b adds u_f to its top-n neighbors. As a result, u_b can be provided by u_f with more reliable recommendations in comparison with u_a's recommendations.

3.2 Semantic Profiling Manager

Semantic Profile Manager module is responsible for creating and updating ontology-based profiles for both user and item.

Ontological User Profile We take advantage of the trust model presented by Dokoohaki et al. [9] to define the trust between users who are expressed using the *FOAF Agent* concept. Dokoohaki's trust ontology has three concepts. *Relationship* is the main element which expresses the trust relations on top of the Social Network of FOAF user profiles.

MainProperties and *AuxiliaryProperties* are the other main components of aforementioned ontology, which respectively define essential and optional attributes for relations which exist in between users on the network. Two associations connect both *MainProperties* and *AuxiliaryProperties* to the *Relationship* concept. *Relationship* always has a sink and a source, which is described by a *Truster* and a *Trustee*. Reader is refered to [9] for more information about the complete structure of trust ontology. In our model, a trust value is computed based on users' ratings to different items, possibly in different contexts. To compute the trust value between users, we follow the approach proposed in [13] based on the difference of a user's rating and its recommender's rating to their common item(s). As a result, as the distance between their rating values increases, trust decreases linearly.

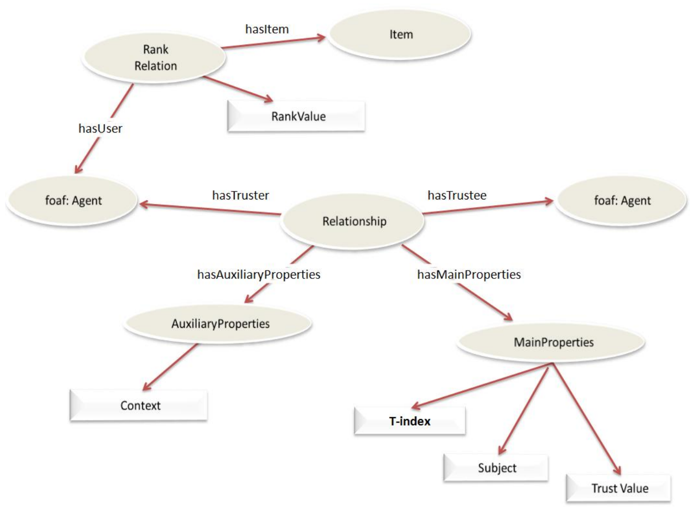


Fig. 3. User Ontology Model

As shown in Fig. 3, we create an instance of *Relationship* concept between two users for whom a trust value is computed. The users are specified as *Truster* and *Trustee* and their trust value and subject is assigned as *MainProperties* [9] to the instance defined earlier. In addition, we assign T-index as a *MainProperty* of the *Relationship* instance. We also define the *RankRelation* concept for associating a user to an item by a rank value. This concept is used to keep track of rated items by a user that we refer to as *user profile*.

Ontological Item Profile We have developed an ontology for item's knowledge domain which can be extended by all other ontologies in the same domain. We introduce a new concept called *TopTrustee*, which is derived from the notion of item's *TopTrustee* described in section 3.1, and we assign it to an individual item to create a list of users who rate the item. The list of raters is ordered by their T-index. In a real world scenario, these *TopTrustee* lists can be implemented by Distributed Hash Tables (DHT) [16] with unique URI as their keys.

3.3 Trust Engine

We adapt the formalization of trust presented by Lathia et al.[13] based on difference of a user's rating and its recommender's rating to their common item(s). As the difference of their rating values decreases, trust value between them increases linearly. Suppose we have two users u_a and u_b. Trust between them is formalized as follows [13]:

$$T(u_a, u_b) = 1 - \frac{\sum_{i=1}^{n}(r_{u_a,i_i} - r_{u_b,i_i})}{r_{max} * n} \quad (1)$$

This formula computes the total differences between a user's rating values and its recommender's rating values over *n* historical ratings of u_a multiplied by the maximum value in each rating scale (i.e., 5). This trust value is used to update the trust between the user and its respective recommenders.

3.4 Trust Network

We gradually build up the trust relationships between users based on the rating information of user profile and item profile to generate a so-called *trust network* of users.

As mentioned, we keep top-n neighbors of a user in an ontological structure based on their mutual trust values. The list is updated on "'rating a new item"' event. If the event leads up to some modifications in top-n neighbors of a user, then T-index value is recalculated and updated in all *TopTrustee* lists which contain the user. The scenario is described as follows: when a user rates a new item, we compute its trust with all item's *TopTrustee*s who do not exist in its current top-n neighbors but might be potentially trustworthy users. We also update trust values between the user and its top-n neighbors. Eventually, we form a new top-n neighbors by selecting the most trustworthy users from the union of its preceding neighbors and the potential trustees.

3.5 Recommendation System

There is no central view of similar users' ratings in distributed recommender systems. Thus, in order to generate a recommendation, we need to find a solution for gathering neighbors' opinions. Traversals through neighbors would be an appropriate solution for collecting an item's ratings. In addition, length of connected edges between users through the trust network should be limited to an upper bound (*L*). However, defining a suitable value for *L* is challenging as it leads to a trade off between accuracy and performance. Therefore, as the number of parallel traversals and *L* increase, we can achieve better prediction accuracy and coverage for recommendations, while we require more resources of bandwidth and computations. On the other hand, a user is allowed to traverse through its either direct or indirect neighbors as long as its mutual trust value does not fall down a predefined minimum threshold (*v*).

After collecting all the information from a user's neighborhood by traversals, we aim to minimize the risk of recommending irrelevant items to a user [13]. Therefore, predicted rating value provides us with the fact that whether the user is interested in an item or not. Prediction value is taken as a weighted average of user *a*'s neighbors ratings[17]. Reader is advised to refer to Zarghami et al.[18] for more information regarding collecting the recommendations and making predictions.

4 Evaluation

4.1 Setup

We evaluate above presented method based on *MovieLens*[4] dataset which consists of 943 user profiles. Ratings are based on five point scale. The profiles are divided into training and test sets including 80% and 20% of ratings, respectively. To design ontological profiles for user and item, we use Protégé[19]. We take advantage of Protégé API in Java for implementing the recommendation system. First, we build up trust-aware social networks as described before, based on the training data and we visualize the constructed networks by Welkin[20] to study effect of T-index on structure of the networks. Then, we use a traversal mechanism for collecting recommendations through the trust networks. In fact, evaluating the trust computation is not our concern. As we explained in Section **??**, we have adapted a light-weight trust formalization to conduct our experiments for investigating the impact of T-index on the performance of our recommendation system.

In this work, we aim to show how the network structure based on trust relationships in a social setting, can be affected by T-index. To do so, we first compare Indegree distribution of top-10 trustworthy users with different values for T-index. Then, we build trust networks with and without T-index to observe the difference. The differences includes both inferred and trimmed edges made when T-index is employed. We study the effect of T-index variation on the prediction coverage and accuracy of recommendations collected based on rating values of neighbors who provide mutual trust value higher than the minimum threshold(v) as 0.1 and can be reached within the upper bound for path length of traversals (L) as 3.

We run our experiment in different settings for various sizes of top-n neighbors for each user as n and *TopTrustee* list for each item as m. Although utilizing T-index we achieved more improved results, we have gained the most significant improvement when experimenting with $m = 5$ and $n = 5$ in previous work. Therefore, we choose the values of both n and m to be 5 for studying the Indegree distribution and trust networks structure in an effective manner. We also consider different values for T-index which range from 0 meaning no T-index is used to other values $25, 50, 100, 200, 500, 1000$. To study coverage and accuracy, the values of n are tuned to be $\in 2, 3, 5, 10, 20, 50$ while m stays the same as 5.

4.2 Results and Discussions

In the first step, we study the Indegree distribution of the top-10 trustworthy users for various values of T-index while n and m are both equal to 5. As mentioned earlier,

[4] http://www.cs.umn.edu/research/GroupLens/data/

Indegree represents incoming edges to a node as a user who is trusted by others. As shown in Fig. 4, when T-index is employed (T-index$<>$ 0), the top-10 trustworthy users' weights in terms of incoming trust relationships are more balanced. This means that users have on average more opportunities to find the most similar centric nodes as their main clusters. As a result, the load of incoming trust relationships imposed on the most trustworthy user, is distributed among other trustworthy users which makes our recommendation system more resistant against node failures or bottlenecks on the trust networks. Thus, the results significantly change when T-index is used, regardless of its non-zero values $(25, 50, 100, 200, 500, 1000)$.

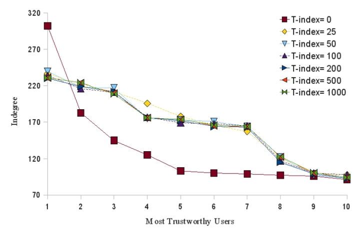
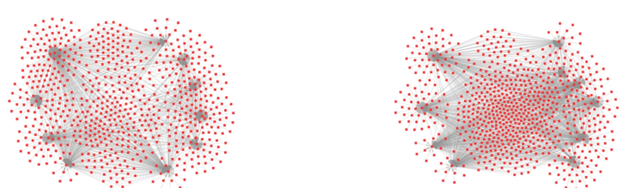


Fig. 4. The Top-10 trustworthy users Indegree

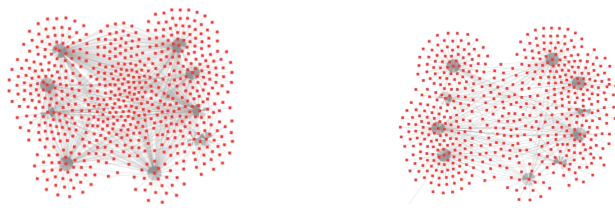
To study the effect of T-index on trust networks structure, we generate two trust networks with and without T-index while *n* and *m* are the same as 5. Fig. 4 shows that the Indegree distribution dramatically declines for the first top-5 trustworthy users without using T-index and the first top-10 trustworthy users with applying T-index. However, for the most trustworthy users placed after the first ten, the Indegree distribution has a steady decrease continuously. For the sake of simplicity, we only study the trust networks structure of the users who are directly connected to at least one of the top-10 trustworthy users.



(a) Without T-index (b) With T-index= 100

Fig. 5. Generated Trust Networks for Top-10 Trustworthy Users (n= 5, m= 5)

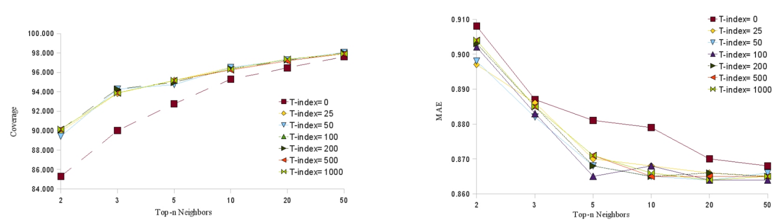
Fig. 5(a) and Fig. 5(b) depict the trust networks structure with and without T-index, (T-index=100) and (T-index=0), respectively. Figs. 5(a) and 5(b) show the trust networks' structure with and without T-index, for T-index=100 and T-index=0, respectively. For the sake of simplicity, we display only users(displayed as nodes) and their connections (trust relationships) to top-10 trustworthy users. As mentioned, each cluster is described as a group of like-minded users in terms of trust. It is shown that the number of common users between clusters increases which enables users of different clusters to find each other easier. In our case, more users form divergent areas of users' interests, presented as clusters, can be accessible.



(a) Inferred Edges (b) Trimmed Edges

Fig. 6. Alignment of Trust Networks for Top-10 Trustworthy Users (n= 5, m= 5)

To justify the results, we compare the formed trust networks with and without T-index to show the inferred and trimmed edges individually. Fig. 6(a) indicates that inferred edges are mostly located between centric nodes. Therefore, the number of users which belong to different clusters, grows in the centric area of the figure. In contrast, 6(b) reveals that most of the trimmed edges are located in just one cluster.



(a) Coverage (b) MAE

Fig. 7. Comparing the results based on different T-index values

Finally, we study coverage and MAE of the generated recommendations for several n with different T-index values while the value for m is the same and equal to 5. As shown in Fig. 7(a), the minimum coverage for n= 2 without T-index is more than 85% which is improved in camparison with the result of similar work[14] with coverage< 60% at the same path length (L= 3) and even for larger sizes of n. Fig. 7(a) shows that coverage has improved at all values of n when T-index is employed. We also demonstrate that the coverage improvement is almost the same for all non-zero values of T-index. Nevertheless, we achieve better results for coverage as the size of neighbors list (n) decreases. As shown in Fig. 7(b), the maximum MAE value for n= 2 without

T-index is less than 0.91 which outperforms a similar work[14] with MAE> 0.96 considering the same threshold for path length of traversals (L= 3). It shows that including items' *TopTrusteeList* in "top-n neighbors" can improve the results. On the other hand, it reveals that utilizing T-index achieves better results. As with coverage, we observe in Fig. 7(b) that T-index improves MAE for all values of n. However, the extent of improvement of MAE changes with a constant value of T-index and different values of n. For instance, although MAE has the most effective result with T-index$= 100$ and $n= 5$, it has its worst value with the same T-index when $n= 10$. Despite coverage, T-index does not always make MAE better as the size of neighborhood list decreases. Fig. 7(b) shows that MAE is improved significantly with T-index when $n= 5$ and 10 whereas MAE result is trivial when $n= 3$ and 50. In conclusion, while using T-index results in better prediction accuracy and coverage of recommendations, accuracy is more affected by different values of T-index and the size of neighborhood list (n).

5 Conclusion and Further Work

In this work, we have developed an ontological framework to create trust relationships among all types of users with respect to different types of items, accessed by unique URI across heterogeneous networks. we have built up a trust network of users to collect recommendations for a target user by a walking algorithm. We have introduced an item's *TopTrustee* list which include users who might not be reachable through a predefined maximum path length of traversals. Thus, when a user rates a new item, its neighborhood potentially consist of its own top-n neighbors plus m users of the item's *TopTrustee* list. We have also proposed a measure called T-index to prioritize the users of *TopTrustee* lists based on their trustworthiness. Therefore, an item's *TopTrustee* list keeps the top-m trustworthy users who rate the item. In order to justify the results, we have analyzed and visualized the effect of T-index on the structure of constructed trust network based on experimental data. Our empirical evaluation indicates that T-index increases the number of common users between different clusters and enables their users to access each others more convenient. Therefore, it improves prediction coverage and accuracy of recommendations collected within few edges that connect users. We show that the amount of improvement for accuracy, despite coverage, heavily depends on T-index value as well as on the size of neighborhood list (n). We intend to employ T-index as a coefficient to trust formalization in order to contribute trustworthy users more effectively.

Acknowledgment

This work has been done within the FP7-216923 EU IST funded SMARTMUSEUM project and is part of the IOP GenCom U-Care project sponsored by the Dutch government under contract IGC0816.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. The Semantic Web, Scientific American (May 2001)

2. Golbeck, J.: Computing and applying trust in web-based social networks. (2005)
3. Golbeck, J., Hendler, J.: Accuracy of metrics for inferring trust and reputation in semantic web-based social network. Lecture Notes in Computer ScienceSpringer (2004) 116–131
4. Hirsch, J.E.: An index to quantify an individual's scientific research output. PNAS **102**(46) (November 2005) 16569–16572
5. Golbeck, J.A.: Computing and applying trust in web-based social networks. PhD thesis, University of Maryland at College Park, College Park, MD, USA (2005) Chair-Hendler, James.
6. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "'word of mouth"'. In: ACM Conference on Human Factors in Computing Systems, ACM Press (1995) 210–217
7. FOAF: Last modified Feb-2009, Available: http://www.foaf-project.org/.
8. Dokoohaki, N., Matskin, M.: Structural determination of ontology-driven trust networks in semantic social institutions and ecosystems. In: International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '07) and the International Conference on Advances in Semantic Processing (SEMAPRO 2007), IEEE Computer Society (2007) 263–268
9. Dokoohaki, N., Matskin, M.: Effective design of trust ontologies for improvement in the structure of socio-semantic trust networks. International Journal On Advances in Intelligent Systems **1st**(1942-2679) (2008) 23–42
10. Massa, P., Avesani, P.: Trust-aware collaborative filtering for recommender systems. Lecture Notes in Computer Science **3290** (2004) 492–508
11. Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., Tennenholtz, M.: Trust-based recommendation systems: an axiomatic approach. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 199–208
12. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces, New York, NY, USA, ACM (2005) 167–174
13. Lathia, N., Hailes, S., Capra, L.: Trust-based collaborative filtering. In: IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust management and Security, Department of Computer Science, University College London, London, UK (2008) 14
14. Weng, J., Miao, C., Goh, A.: Improving collaborative filtering with trust-based metrics. In: SAC '06: Proceedings of the 2006 ACM symposium on Applied computing, New York, NY, USA, ACM (2006) 1860–1864
15. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classifications. Journal of the American Statistical Association **49(268)** (1954) 732–764
16. DHT: Last modified March-2009, Available: http://en.wikipedia.org/wiki/Distributed_hash_table.
17. Bell, R., Koren, Y.: Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: In IEEE International Conference on Data Mining (ICDM'07), IEEE (2007) 175–186
18. Zarghami, A., Fazeli, S., Dokoohaki, N., Matskin, M.: Social trust-aware recommendation system: A t-index approach. In: Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT 2009). (2009)
19. Protégé: Copyright 2009 Stanford Center for Biomedical Informatics Research, Available: http://protege.stanford.edu/.
20. Welkin: Copyright 2004-2008 Massachusetts Institute of Technology, Available: http://simile.mit.edu/welkin/.