Unsal, M., Baylar, A., Tugal, M. & Ozkan, F. 2008. Increased aeration efficiency of high-head conduit flow systems, Journal of Hydraulic Research **46**(5):711-714.

Unsal, M., Baylar, A., Tugal, M. & Ozkan, F. 2009. Aeration efficiency of free–surface conduit flow systems, Environmental Technology **30**(14): 1539-1546.

Submitted: 09/10/2012
Revised: 19/03/2013
Accepted: 20/02/2014

Journal of Engg. Research **Vol. 2 - (2) June 2014** pp. 29 - 48

خوارزميات استدلال الثقة لشبكات التواصل الاجتماعي

مها فيصل، أسماء السميط، وزينب الأمير

قسم هندسة الكمبيوتر - جامعة الكويت

خلاصة

اكتسب التواصل الاجتماعي على الانترنت جاذبية كبيرة بين اوساط الشباب وشرائح المجتمع الاخرى من خلال مجموعة من التطبيقات مثل الرسائل الفورية ، المدونات، ومواقع شبكات التواصل الاجتماعي . ولكن يتسم هذا النوع من التواصل بعدم توفر وسائل كافية لتقييم مقدار الثقة التي يجب ان تعطى للمعلومات والاشخاص الذين يتم التعامل معهم من خلال هذه التطبيقات. يمكن لأنظمة المزكي ان تلعب دورا هاما في تقييم مصداقية المستخدم وتقديم توصيات تجاه إقامة صلات بين أعضاء الشبكة الاجتماعية. تهدف هذه الورقة لتحسين أنظمة المزكي القائمة على الثقة من خلال اقتراح خوارزمية جديدة وتعديل بعض خوارزميات القائمة. في هذه الورقة تم تقييم أداء أربع خوارزميات معدلة مع الخوارزمية الجديدة المقترحة لاستدلال درجة الثقة في المستخدمين باستخدام بيانات من موقع التواصل الاجتماعي تويتر. وأظهرت النتائج ان الخوارزمية المقترحة تقوم بتقديم توصيات دقيقة خاصة عندما تكون الشبكة الاجتماعية كثيفة.

Trust inference algorithms for social networks

MAHA FAISAL*, ASMAA ALSUMAIT** AND ZAINAB AL-AMEER***

Computer Engineering Department - College of Computing Sciences & Engineering - Kuwait University. P.O. Box : 5969, Safat 13060, Kuwait

Emails: maha.faisal@ku.edu.kw,** alsumait@eng.kuniv.edu.kw, *** eng_zalameer@yahoo.com*
** Corresponding Author.*

ABSTRACT

The exponential growth of social networks has made establishing a trusted relationship increasingly important. Recommender systems can play an important role in assessing a user's trustworthiness. Such systems are designed to offer recommendations of trustworthiness when establishing connections among social network members, where the system rates members by inferring their degrees of trust. In this work, we developed a recommender system that provides recommendations about trusted social network members. We compared the time complexity and the accuracy of the following four adapted algorithms and a new proposed algorithm: Top Trusted Members, Target's Reputation and Similarity, Depth First Search (DFS) Trust Propagation, Dijkstra's Trust Propagation and Target's Followers (new). An experiment was conducted using a dataset from Twitter. The results show that the Target's Followers algorithm is a promising approach for making accurate recommendations, especially when the network is dense.

Keywords: Recommender system; reputation; simulations; social network; trust factor.

INTRODUCTION

Social Networking Systems (SNS) are online services that focus on facilitating the development of social relationships among people who share interests or real-life connections. SN members are represented by their profiles, which can reveal a lot about them through information they share and the connections and interactions they make online. Currently, SN members are spending more time producing and consuming information; however, only a few online systems explicitly handle trust, especially in the social context.

This paper utilizes Recommender Systems (RS) to help SNS members establish trustworthy relationships. We describe five algorithms that provide an explicit interpretation of trust in social networks and provide information on establishing trusted relationships through recommender systems. These algorithms infer trust between two individuals with no relationship based on reputation, similarity and

relationships with other social members. The contribution of this paper is in designing and analyzing trust models for SNS and addressing the lack of direct trust values between SN members. Additionally, our proposed algorithm addresses cases in which a trusted chain of friends cannot provide recommendations.

We begin by presenting related research on trust and recommender systems. We then define the problem and terms that are addressed in this paper. Next, we describe the five inference algorithms that are used to find trusted SNS members. We then present our developed Recommender System Tool and our experiments. We conclude with a discussion on our results and directions for future research.

RELATED WORK

As SNS become a popular platform for user-generated content, modeling trust has become an active research topic. Therefore, trust is a key component of several SNS (Dellarocas, 2001; Jøsang *et al.*, 2007). There are two ways to estimate trust: global and local trust metrics. Both methods attempt to predict the trustworthiness of a given user. Global trust metrics assign to a given user a unique trust score, which is the same across the system. In contrast, a local trust metric provides a personalized trust score that depends on the point of view of the evaluating user (Massa & Avesani, 2007). In this paper, we assume that SN members should rate each other when establishing relationships and use aggregated indicators to rate a given member and derive a trust score. This approach can help other members decide whether to establish a relationship with that member in the future. As we discuss related work, we summarize the relevant work on recommender systems.

Recommender systems may produce a list of recommendations based on collaborative filtering, content-based filtering or a hybrid. Collaborative filtering builds a model from a user's past behavior, as well as similar decisions made by other users, to rate new users and make recommendations (Herlocker *et al.*, 2004; Walter *et al.*, 2008). Content-based filtering utilizes a series of discrete characteristics of a user to recommend additional similar users or items. Both approaches are often combined to produce Hybrid Recommender Systems.

Each type of system has its own strengths and weaknesses. Collaborative filtering has the data sparsity problem and the cold-start problem. In contrast to the very large number of items in recommender systems, each user typically only rates a few items. Therefore, the user rating matrix is typically very sparse. It is difficult for recommender systems to accurately measure user similarities from a limited number of ratings (Lee *et al.*, 2004). A related problem is the cold-start problem, which appears when a user first registers with a system and has no ratings on record. Thus, no personalized predictions can be provided (Schein *et al.*, 2002). While content-based systems require very little information to begin, they are far more limited in scope. In such systems,

the member is limited to the recommendations of other members that are similar to previously rated items. Even a 'perfect' content-based technique would never find anything surprising, limiting the range of applications for which it would be useful. This shortcoming is called the serendipity problem (Weng *et al.*, 2006).

Recent research has demonstrated that a hybrid approach that combines collaborative filtering and content-based filtering could be more effective in some cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them, by adding content-based capabilities to a collaborative approach, or by unifying the approaches into one model (Adomavicius & Tuzhilin, 2005). Some recommender systems incorporate trust information into the recommendation process (O'Doherty *et al.*, 2012). These trust-based recommender systems have been shown to perform significantly better than traditional recommendation techniques in terms of overall accuracy and coverage of rating predictions (Jøsang *et al.*, 2007; Victor *et al.*, 2011; O'Doherty *et al.*, 2012).

Collaborative filtering

Golbeck (2005) introduced the TidalTrust algorithm, which estimates the trust value between two members in a social network. This algorithm calculates the trust values using the average weights of all of the member's neighbors. Later, Kuter & Golbeck (2007) proposed a more accurate trust estimate algorithm called SUNNY, which uses a probabilistic sampling technique to estimate the confidence in the trust information from some designated sources. Massa & Avesani (2007) proposed a very similar approach called the MoleTrust algorithm. The MoleTrust algorithm performs a depth-first search to propagate and infer trust in the trust network. Al-Oufi *et al.* (2012) proposed a propagation mechanism that broadcasts node capacity through social connections in a personal network to identify and rank local trusted users.

O'Donovan & Smith (2005) investigated how trust can be used to decrease recommendation errors. They developed a method for automatically measuring trust between users based on the ratings history of the two users. They distinguish two levels of trust to generate a reliable score, namely, the item-level and profile-level. Victor *et al.* (2011) compared the performance of several well-known trust algorithms incorporated into collaborative filtering techniques. However, this comparison was performed uniquely on the Epinions.com dataset.

Content-based

Ali *et al.* (2007) proposed an access control scheme called Personal Data Access Control, inspired by multi-level security, to allow users to share data among their friends using a trust computation. The user classifies his friends into one of three

protection zones that determine whether that friend can obtain access to the user's data.

Sarda *et al.* (2008) proposed a model that combines two trust forms: friendship trust and domain expertise, according to trained levels. Friendship is related to trust in a friend's recommendations, where domain expertise arises from others' knowledge in certain fields.

Maheswari & Karpagam (2010) evaluated trust using a reputation-based approach. Three properties were used in their approach: transitivity, composability, and asymmetric explicitly. Transitivity means that for two nodes A and B, A should trust B to make recommendations to A about others. Composability describes the situation when a number of recommendations about a node's trustworthiness are received; the trust values in those recommendations should then be composable into a single belief about the node's trustworthiness.

Wang *et al.* (2011) developed a method for measuring trust between users based on their common tastes. The items are clustered into different groups, and a personalized taste set is built for the user and used to infer trust based on the common taste sets between users.

Hybrid

Walter *et al.* (2008) proposed a model that uses social network information in recommendation systems. The users use their social network to obtain information and their trust relationships to filter it. The authors also investigated how the dynamics of trust among agents/users affect the performance of the system by comparing their system to a frequency-based recommendation system.

Pitsilis & Marshall (2006) developed a model that constructs trust relationships between entities based on their common choices. Trust propagation was used to extend trust relationships beyond the direct neighbors.

Mori (2008) proposed trust-based recommendation strategies that show how neighbors are selected and weighted during the recommendation procedure. This trust model incorporates trust and reputation into recommender systems.

Chen & Fong (2010) proposed a framework that extends the Kazienko & Musiał (2006) recommendation framework. They measured trust factors, with the challenge of determining their scaled weights, and they generated recommendations by filtering information from similar users. The activities are represented as dynamic data that are monitored and gathered by the system.

Jiang *et al.* (2012) proposed the SWTrust framework to generate small trusted graphs for trust evaluation in large online social networks. Users' neighbors are

classified by their social distance, and the neighbors' priorities are calculated based on their topic-related degree and target-related degree.

Huang *et al.* (2010) used a modified Dijkstra's algorithm to propagate trust and the PageRank algorithm to calculate the popularity of a user in relation to certain keywords. As a result, they generated a ranked list of trusted friends that have common interests.

All of these methods generate trust estimates between users, and use these trust relationships in various techniques to generate personalized predictions for items for users. Each study has shown that incorporating trust into recommendation techniques improves the accuracy of recommender systems in comparison to the basic collaborative filtering or content-based systems.

TERMS AND PROBLEM DEFINITION

We have developed a recommender system that provides recommendations for establishing relationships with trustworthy Twitter members. Currently, Twitter does not provide trust values for its members, which makes it difficult to apply most of the current SN trust models that rely on an existing trust score between its members. Our system rates members by estimating their degrees of trust, which are augmented to every member profile. Then, member a can decide whether to follow member c according to the calculated degree of trust. This approach could be adapted by other SNS that lack trust values among their members. Next, we define trust, trust score, trust propagation, similarity and reputation.

Relationship establishment

When establishing relationships, people usually analyze the costs and benefits of creating a relationship (Berger, 1986). Usually, a relationship starts as a weak tie, especially if it is established with an unknown person. Based on a cost-benefit analysis, the relationship will become stronger (higher trust), remain weak or become disconnected (Nisan *et al.*, 2007). In our work, we recommend relationships that would be considered weak ties. Thus, we start by recommending weak ties; maintaining such a relationship will move the tie from weak to strong. Other indicators of trust are explained in the following section and are used to suggest higher trustworthiness and to classify a relationship as having a greater strength.

Defining trust

Trust between SN members could be demonstrated through individual social behavior and social structure (Brass *et al.*, 1998). Trust is a perceived quality, as defined by Tseng & Fogg (1999), who stated that "trust indicates a positive belief about the

perceived reliability of, dependability of, and confidence in a person, object or process." In addition to dependability, reputation and homophily (similarity), there are other indicators of trust (Bonaccio & Dalal, 2006). In this paper, we define trust in the following manner: the image presented through indicators that suggests that the member is a reliable source of advice that would aid in decision making. The related indicators for building a member's image are shown in Figure 1. We are deriving our indicators from Twitter; however, this approach could be applied to other SNS.

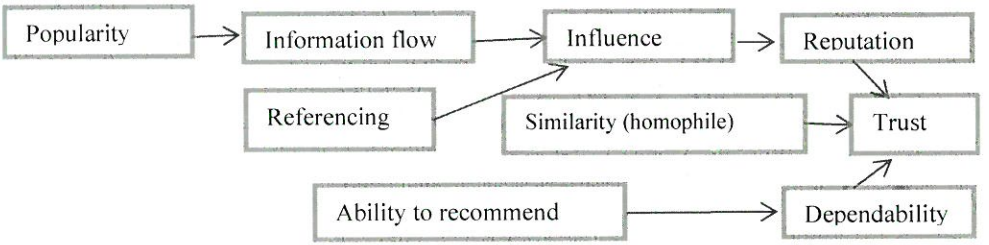

Fig. 1. Building trust from social indicators

People trust information or advice if the source of information has knowledge and has a good reputation based on his personal interactions or the opinion of other members in the community. Additionally, a recommendation is considered more trustworthy, if the recommender shares the same interests and is part of the same community. Therefore, the social factors or indicators that influence trust are similarity, reputation and dependability. Table 1 presents the relationships of different social factors related to trust and their indicators in Twitter.

Table 1. Social factor indicators in Twitter

Social Factor	Indicator in Twitter
Information flow	Number of re-tweets, Number of followers
Referencing	Number of mentions
Influence	Number of mentions, Number of re-tweets, Number of tweets
Popularity	Number of followers
Similarity	Overlap between members' profile fields and followers
Ability to recommend	Number of followers

Trust score

The Trust score (T_{ac}) represents the degree of trust that an opinion or a recommendation provided by member c to member a will help a make appropriate decisions. On Twitter, a trust score is assigned for each pair of SN members a and c whenever a "follows"

c. However, trust scores are not part of the Twitter user profile. Jøsang (1999) showed that uncertainty could be used as a measure of trust. In addition, information theory states that entropy (H) is a natural measure for uncertainty (Sun *et al.*, 2006). Let $P(c)$ denote the probability that c can provide a recommendation; then, the Trust score T_{ac} is computed as follows.

$$T_{ac} = \begin{cases} H(P(c)) - 1 & if \quad 0 \leq P(c) < 0.5 \\ 1 - H(P(c)) & if \quad 0.5 \leq P(c) \leq 1 \end{cases} \tag{1}$$

wehre $P(c) = \frac{k+1}{n+2}$

$$H(P(c) = -(P(c))log_2(1 - P(c))$$

Where k is the number of friends of member c that are part of relationship requests (some members want to follow them), and n is the total number of relationship requests. T_{ac} is a real number in the range [-1,1]. When $P(c) = 1$, a will have the highest trust in c ($T_{ac} = 1$). While $T_{ac} = -1$, when $P(c) = 0$, a will have the lowest trust in c because c cannot provide helpful recommendations. Overall, the trust score will be negative for $0 <= P(c) <= 0.5$, which indicates distrust, and positive for $0.5 <= P(c) <= 1$, which indicates trust.

Trust propagation

The goal of trust score propagation is to provide the trust scores of users when a SN member cannot receive a recommendation from direct friends (Weng *et al.*, 2006). The requesting member a will request a trust score from his direct friend b, who will forward this request to their most trustworthy friends and continue asking until they reach a friend who can provide a rating for member c. The trust score T_{ac} is calculated using the following formula:

$$T_{ac} = T_{ab} * T_{ab} + (1 - T_{ab}) * (1 - T_{ab}) \tag{2}$$

Reputation

Reputation is holding an opinion about a person within a community; a reputation system computes reputation scores for online community members based on a collection of opinions provided by other members in the community and supports decision making when building online communities and establishing relations in SNS (Lazzari, 2010; Jøsang *et al.*, 2007). Highly influential members usually have a good reputation in their community (a specific domain or field). The influence of a Twitter member is recognized through the member's ability to spread information (re-tweeting) and having a large number of referrers. Let M be the total number of

members in the social network, and let *R(i)* be the rank given to member *i* based on the number of re-tweets and mentions (Cha *et al.*, 2010). The reputation factor *Rep (i)* is computed as follows:

$$Rep(i) = (M - R(i))/M \tag{3}$$

Similarity

People do not establish social relationships randomly; they tend to connect with other members who have similar traits and tastes. In this paper, the similarity between SN members: the similarity factor (Sim), is based on the similarity in their profiles (location, language) and social interests and interaction (members they are following). We use the overlap coefficient as a similarity measure by using the following equation (Markines, *et al.*, 2009):

$$Sim(a, c) = \frac{|a.profile \cap c.profile|}{\min(a.profile, c.profile)} + \frac{|a.following \cap following|}{\min(a.flollowing, c.following)} \tag{4}$$

TRUST INFERENCE ALGORITHMS

Brass *et al.* (1998) has shown that both social network structure and individual factors have a large influence on ethical human behavior. Additionally, recommender systems that use hybrid methods of collaborative filtering (relying on the network structure) and content-based recommendations (relying on the user profile) have shown promising performance (Resnick & Varian, 1997). In this paper, we are adapting similar strategies to infer trust in SNS.

Next, we describe three adapted collaborative filtering methods, one adapted content-based method, and a new proposed hybrid trust inference algorithm that is used in our newly implemented tool.

Top trusted members algorithm (Top)

This algorithm is based on calculating the trust core using collaborative filtering. Asking trusted direct neighbors about a target member is used to obtain trust scores. Previous research by Ziegler & Golbeck (2007) indicates that the most accurate information will come from highly trusted neighbors. The proposed algorithm, shown in Figure 2, predicts the trust score (ř) for a new friend using a modification of the Resnick formula (Resnick *et al.*, 1994; Resnick & Varian, 1997). The Resnick formula is the foundation of an automatic collaborative filtering algorithm based on a k-Nearest Neighbors (kNN) algorithm among the members. The proposed formula will predict how SN member *a* will rate (assign a score to) a member to be followed, *c*, if their relationship is established. The predicted trust score will be generated by asking the

trusted direct neighbors of requester a, shown in Equation 5, or by asking the most influential members, shown in Equation 6. The most influential members are the members who have the highest reputation; they are identified and ranked as described earlier. This algorithm sets a minimum trust threshold (τ_{trust}) and only considers direct neighbors that have trust ratings at or above the threshold. In this manner, only the highest trust ratings possible are included (ignoring direct neighbors that have low trust score values) without limiting the number of inferences that can be made.

$$\check{r}_{a,c} = r_a + \frac{\sum_{bi \in B(a)} T_{ab}(T_{b_i c} - r_b)}{\sum_{bi \in B(a)} T_{ab}} \tag{5}$$

Where r_a is the average score that member a assigns to his friends, b_i is one of a's direct friends, and T_{ab} is the trust score that member a assigns to user b. $B_{(a)}$ refers to the set of all of a's direct friends that have a rating for member i.

$$\check{r}_{a,i} = r_a + \frac{\sum_{fi \in F(a)} T_{af}(T_{f_i c} - r_f)}{\sum_{f \in F(a)} T_{af}} \tag{6}$$

$F(a)$ refers to the set of the most influential members, and f refers to a top influential member. When the explicit relationship between the members is not available, asking "the most influential members about member c" is an alternative method of determining the trustworthiness of member c.

1. Read requests (a, c).
2. Find $\check{r}_{a,c} = r_a + \frac{\sum_{bi \in B(a)} T_{ab}(T_{b_i c} - r_b)}{\sum_{bi \in B(a)} T_{ab}}$ for all $bi \in B(a)$: $T_{b_i c} > \tau_{trust}$
3. if ($\check{r}_{a,c} > \tau_{trust}$), establish a connection.
4. Else find $\check{r}_{a,c} = r_a + \frac{\sum_{fi \in F(a)} T_{af}(T_{f_i c} - r_f)}{\sum_{f \in F(a)} T_{af}}$ for all top influential members $fi \in F(a)$: $T_{f_i c} > \tau_{trust}$

   4.1. if ($\check{r}_{a,c} > \tau_{trust}$), establish a connection.

Fig. 2. Inference algorithm using the top trusted members

This algorithm bases its prediction on values provided by members of the community; having a small number of values affects the ability of the system to infer trust. Typically, the number of influential users in a SN is much smaller than the set of direct friends ($|\mathbf{F(a)}| << |B(a)|$), so this algorithm has a reasonable time complexity of $O(|B(a)|)$ assuming that influential users are already identified and ranked.

Target's Reputation and Similarity Algorithm (TRS)

This algorithm is based on calculating the trust core by using a content-based strategy. Maheswari & Karpagam (2010) showed that trust could be calculated using

a reputation-based approach. In contrast, Wang *et al.*(2011) developed a method for generating trust between users based on their common interests. This proposed algorithm adds similarity to the reputation-based trust-inference approach because, in reality, people would obtain recommendations from reputable members of the community or from others who have similar interests, as in Figure 3. Most of the algorithms recommend items, whereas our reputation model recommends members based on the influence of the member on the SNS. Equation 7 shows the formula that is used to infer trust.

$$\check{r}_{ac} = ((\omega_{rep} * Rep(c)) + (\omega_{sim} * Sim(a,c)/2)) * Z_{\max} \quad (7)$$

Where Z_{max} is the maximum trust score, and ω_{rep} and ω_{sim} are weights to convey the relative priority of the reputation and the similarity with other SN members. However, this algorithm cannot infer trust when the target member has very low influence and low similarity to the requester. The weights used by the algorithm could be adjusted by the user or automatically learned. Currently, a default value is used; automatic adjustment is outside our current scope.

1. Read requests (a, c).
2. $\check{r}_{ac}=((\omega_{rep}*Rep(c))+(\omega_{sim}*Sim(a,c)/2))* Z_{max}$
   a. if ($\check{r}_{a,c} > \tau_{trust}$) , establish a connection.

Fig. 3. Inference algorithm using target's reputation and similarity

This algorithm is suitable for cases when no trusted chain could be formed to the target through direct friends or influential users. This method relies only on the target's reputation and similarity; it scales well if the rank of every SN member is already identified. The time complexity will be *O(T(Sim))*, where T is time complexity. *T(sim)* = O(|*max(a.followers, c. followers)*|) if a hash table is used. Thus, in the worst case, the time complexity is *O(n)*.

DFS Trust Propagation Algorithm (DFSProp)

This algorithm is based on calculating the trust core using collaborative filtering. We adopted the Capra (2004) concept of a web of trust to expand trust across SNS; this concept is based on a weighted transitivity of trust. A certain degree of trust is obtained based on the length of a trust chain. For example, when the length of the trust chain is 4, e.g., if *a* trusts *b*, *b* trusts *c*, *c* trusts *d*, and *d* trusts *e*, then *a* can trust *e*. However, the longer the trust chain is, the greater the decay in the degree of trust. When a SN member wants to establish a relationship for which no trust score could be generated from direct friends, the Depth First Search (DFS) algorithm is used. DFS is typically

used to identify if a certain node in a graph is reachable from a given source, which applies to any graph traversal algorithm. DFS differs from other search algorithms by going deeper in the graph. If the target is deep within the graph, DFS usually reaches it faster. In our work, friends with the highest trust scores are traversed first to generate a path to the target member. As depicted in Figure 4, trust scores are propagated through a path in which the requesting member starts by asking his direct friends, forwarding the request to their most trustworthy friends, and continuing to ask until he reaches a friend who can provide a rating for the target member.

The propagated trust score will decrease as the number of hops in the path increases. A limit on the number of hops in the path should lead to more accurate results. Previous work by Golbeck (2005) has shown that the average error, which is measured as the absolute difference between the computed rating and the user's rating, increases as the number of hops increases. Therefore, the accuracy decreases as the path length increases, and thus, shorter paths are more desirable.

We assumed that the optimal length of a trust chain (τ_{path}) should not exceed 5 to generate the most accurate trust score without revealing risk, which is based on a tradeoff between the trust availability and the path reliability across a SN. On average, approximately 50% of the people on Twitter are only four steps away from each other, while nearly everyone is five steps or less away (Backstrom *et al.*, 2012). Thus, *DFSProp* has good scalability *O(n)*; however, it may fail to reach the target in some cases.

1. Read request (a, c).
2. While T_{ac}= 0 and Path Length <= τ_{path}
   - *2.1. b= getMaxTrusted(a.directfriends)*
   - *2.2. if c ∈ b.freinds*
     - *2.2.1.* $T_{ac} = T_{ab} * T_{bc} + (1-T_{ab}) * (1-T_{bc})$
   - 2.3. else
     - 2.3.1. *x =getMaxTrusted(b.friends)*
     - *2.3.2. if* ($T_{ax} = 0$)
       - *2.3.2.1.* $T_{ax} = T_{ab} * T_{bx} + (1-T_{ab}) * (1-T_{bx})$

Fig. 4. Inference algorithm using DFS trust score

Dijkstra's Trust Propagation Algorithm (DProp)

We used Dijkstra's algorithm to determine the maximum trust score. This algorithm calculates the trust score using collaborative filtering. Trust is propagated through a

path generated by Dijkstra's algorithm. Dijkstra's algorithm provides a solution for the single source shortest path problem in graphs (Dijkstra, 1959). In our case, we are finding single source highest trust. Because the graph in the *DFSProp* algorithm is pruned (does not exceed 5 hops), some targets may not be reached. Additionally, because *DFSProp* is making a greedy choice in the traversal, its path to the target may not be optimal, while Dijkstra is guaranteed to generate the optimal solution (the path with the highest trust). In the adopted algorithm, a heap was employed to make an ordering of the neighbors of a according to their trust scores. A trust score was then calculated using the maximum trust path as it can be efficiently computed with Dijkstra's shortest path algorithm. Figure 5 explains the algorithm that we followed in detail. This algorithm has a poor time complexity of $O(n^2)$.

Trust[a] is an array that includes the trust scores T_{ab} for every b ∈ SN

1. for each member b in SN excluding a
   - 1.1. if b is a direct friend of a
     - 1.1.1. Add the trust score T_{ab} to matrix Trust[a]
   - 1.2. else
     - 1.2.1. Let T_{ab}=0 and add Trust[a]

2. Create a maxHeap H from Trust[a]
   - 2.1. while H is not empty
     - 2.1.1. Remove u (a's neighbor with maximum trust) from H
     - 2.1.2. For each neighbor v of u: // *where v has not yet been removed from H*
       - 2.1.2.1. $T_{av} = max\ (T_{av}, T_{au} * T_{uv} + (1-T_{au}) * (1-T_{uv}))$
       - 2.1.2.2. Update Trust [a]
       - 2.1.2.3. If v is not visited, then update the Trust score T_{av} in the heap and heapify.

     end for

   end while

Use the trust score T_{ac} in Trust[***a***]

Fig. 5. Trust propagation using Dijkstra's algorithm

Target Follower's Algorithm (TFollower)

This newly proposed algorithm calculates the trust score using a hybrid strategy. It is based on a common principle: if you do not know target member c directly, then one way to obtain information about him is to ask someone who is related to him, i.e., the target's followers. In this manner, if member a does not know member c, then member a checks c's direct friends to determine whether they have a high reputation and similarity with him. The average of their trust scores will be used by a. Member a, the requester, checks the reputation and similarity of the target's friends to infer the level of trust; see Figure 6.

This algorithm resolves two main issues: lack of a trusted chain of friends to propagate trust, which is a limitation of the trust propagation algorithms (*DFSProp*

and *DProp*), and target members with very low influence and similarity with the requester, which limits the *TRS's* ability to infer trust.

1. Read requests (a, c).
2. Count = 0
3. Trust scores = 0
1. For all $bi \in B(c)$: $T_{bic} > \tau_{trust}$

   3.1. $\check{r}_{abi}=((\omega_{rep}*Rep(b_i))+(\omega_{sim}*Sim(a,b_i)/2)* Z_{max}$

   3.2. if ($\check{r}_{abi} > \tau_{trust}$)

      3.2.1.1. trust scores = trust scores + $\check{r}_{abi}$

      3.2.1.2. increment count
4. T_{ac} = trust scores / count.

Fig. 6. Inference algorithm using the target's followers

In addition, this algorithm relies on values derived from the target's followers. Thus, it will have better scalability because only a subset of the network (trusted target followers) is involved in the computation. Assuming that every SN member's rank is already identified, the time complexity will be $O(|B(target)| * T(Sim))$.

RECOMMENDER SYSTEM EVALUATION

Experimental design

The goal of this experiment is to compare the performance of the proposed algorithms in terms of their efficiency and accuracy when inferring trust. Moreover, the effect of network density on the trust inference is investigated.

In our experiments, NodeXL (Smith *et al.*, 2010) was used to import data from the Computer Engineering Department's (CpE) Twitter SN, which had 300 members and 80730 following relationships. Three datasets were constructed from the imported data; the first data set included the entire population and all relationships, the second dataset included 24570 following relationships, and the third dataset included 1071 following relationships. An evaluation experiment was conducted for each of the three datasets. The three datasets had the following network densities: Loose (L) with $O(n)$ following relationship, Average (A) with $O(n^2/2)$ following relationship and Dense (D) with $O(n^2)$ following relationship, respectively. The hold-out evaluation method was repeated 6 times to produce different sub-samples. In each simulation,

we calculated the average relationship recovery for 10 iterations; *recovery rate* refers to the percentage of the number of re-established relationship requests that were recommended as trustworthy to the total number of relationship establishment requests. For every simulation, 500 following relationships were randomly selected from the dataset to construct the test set, called *relationship requests*. We performed a simulation using the three network densities with the trust inference algorithms described in the previous section.

We designed three experiments to measure the average recovery rate. The first experiment investigated three inference algorithms from three different strategies for computing trust, *Top*, *TRS* and *DFSProp*. Simulations were conducted to investigate the effect of the network density on every trust inference algorithm and to identify the average recovery rate for each. The second experiment compared different methods of trust propagation (*DFSProp* and *DProp*) to investigate possible improvements in the average recovery rate. In the third experiment, we tested the *TFollower* algorithm. Experimental parameters were set as shown in Table 2.

Table 2. Experimental Parameters

Parameters	Value
τ_{trust}	2.5
Z_{max}	5
τ_{path}	5
ω_{rep}	0.5
ω_{sim}	0.5

Metrics

We measure the RS's ability to recover removed relationships and provide a correct recommendation. Recommending a recovered relationship will suggest that at least there was a weak tie between the two SN members because we always assume that an existing relationship is either weak or strong.

The average recovery rate was used to measure the system's performance, as shown in Equation 8 (Massa & Avesani, 2007). A higher rate indicates that the RS will help members establish trusted relationships. A low value indicates that the RS will not be able to help members establish trusted relationships. Other performance measures were not used (for example, precision and recall) because our system provides recommendations for establishing relationships and does not provide a trust value (nominal variable).

$$Recovery - rate = \frac{1}{|relationship\ Requests|} \sum_{x \in relationship\ Requests} recover(x); \quad (8)$$

$$recover(x) = \begin{cases} 1, & if\ relationship\ established \\ 0, & other \end{cases}$$

Results and Discussion

In the first set of experiments, we considered obtaining a trust score rating directly from the trusted members. The *Top* and *TRS* algorithms were compared. Notably, regardless of the network density, using the *Top* algorithm to obtain scores to infer trust had the worst average recovery rate of the algorithms, as shown in Table 3. This algorithm requires a substantial number of trusted connections to produce an accurate inference. Moreover, the *TRS* algorithm had the best average recovery rate of the three algorithms, especially in sparse networks; it was able to recover 59.5% more relationships than *Top*. Additionally, as the network became denser (with more relationships), both of the *TRS* algorithms had very high recovery averages, and the accuracy of using *Top* improved. In a loose network, it is difficult to establish a relationship due to the sparsity of the data and the cold start problem. *TRS* performs well in different network densities because it depends only on measuring the reputation of the target member and its similarity to the requestor.

In the second experiment, the average recovery rate of the two propagation algorithms, the *DFSProp* and *DProp* algorithms, were compared. By propagating trust, it is possible to reach more SN members, which allows members to be trusted by trusted members (indirect neighbors), to be considered as possible neighbors and hence to compute the trust score. Both algorithms had similar recovery rates for loose and average network densities, while *DProp* had a better recovery rate when the network became dense. This is due to the fact that *DFSProp* uses pruning, while when propagating trust through the SN, some targets may not be reached if the network is dense. However, *DProp* is slow in computing trust scores, $O(n^2)$. *DFSProp* is useful for cold starts because it performs adequately in loose networks.

The third experiment examined the case in which a requestor cannot reach a target member directly from trusted members or through propagation or in which the target has a low reputation and similarity score to the requestor. If the target's follower has tastes that are very similar to those of the requestor, then it is possible to estimate the requestor's trust score for the target. This experiment showed that the *TFollower* recovery rate is similar to those of *TRS*, *DFSProp* and *DProp* while having

the advantage of resolving the issues faced by the other algorithms, as discussed. The following table provides a comparison of the five different algorithms.

A summary of the conducted experiments is shown in Table 3.

Table 3. Experimental Results. The naming convention used is algorithmName-network density (e.g., for Top-D, the algorithm is top trusted and the network density is dense).

Strategy	Simulations						Ave. recovery rate
	1	2	3	4	5	6	
Top-D	71.6	70.1	72.6	72.2	70.8	73.6	**71.8**
Top-A	44	43.8	39.2	46.2	43.6	36.2	**42.2**
Top-L	20.2	19.8	20.4	20.2	17.5	18.4	**19.4**
TRS-D	91.2	91.8	89.6	93.8	92.2	91.4	**91.7**
TRS-A	85.3	86.8	86.8	83.2	86.6	87.6	**86.1**
TRS-L	80.4	77.8	77	79.6	77.8	80.6	**78.9**
DFSProp-D	85.6	86	88	88	86.8	86	**86.7**
DFSProp-A	79.6	79.6	78.8	80.8	81.3	81.4	**80.3**
DFSProp-L	66.4	65.4	64.6	66.6	63.4	65.2	**65.3**
DProp-D	89.4	91.6	91.4	90.6	91	90	**90.7**
DProp-A	80	81	81	80.8	80.2	81.39	**80.7**
DProp-L	63.8	64.2	66.4	66.6	66	66.8	**65.6**
TFollower-D	90.2	88.8	88.2	90	89.4	92.6	**89.9**
TFollower-A	82.8	82.3	82.3	81.3	81.2	82.1	**82**
TFollower-L	68.2	67.4	67.6	68	66.8	68.8	**67.8**

CONCLUSIONS

With the continuous, rapid growth of SNS, trust-based recommender systems will become more and more popular and important. In this paper, we have continued the investigation of inference algorithms in social networks that began in 2011. We discussed how establishing social network relationships can benefit from recommender systems. A recommender system tool for social networks was developed, and we compared five different algorithms that provide recommendations about trusted users on social network sites. We also tested the effect of the network density on our recommender system; three network densities were used, which varied from loose to dense. Comparing the performance in terms of the average relationship recovery rate for the five algorithms in the loose, average, and dense networks, we demonstrated that calculating the trust by *TRS* had the highest average recovery rates, which means that it provides the best recommendations. The recovery rates of the proposed hybrid

algorithm *TFollower* is also promising. Due to its excellent performance in the present study, in future work, we will investigate the performance of *TRS* when the source and the target are in different communities. Moreover, Dijkstra had a high time complexity, which makes it useless in the real world.

Much remains to be accomplished. As in any online community, a number of fake and cloned profiles exist, and these profiles might affect the prediction of the trust scores. In future work, we aim to identify such fake or cloned accounts and Sybil attacks to then investigate the influence of malicious behavior on our algorithms. Suspicious profiles can be discovered by analyzing the similarity of the profile attributes and the friends' networks. We would like to investigate the effect of incorporating distrust between social network members and whether distrust can play a beneficial role in recommender systems. In such SNS, trust can represent the perceived risk of choosing a recommendation from the corresponding member, while distrust is a protective measure indicating the level of doubt about the level of trust that is assigned to the same member. We also plan to incorporate time and topic or domain information into the trust model to handle establishing relationships among members of different communities.

ACKNOWLEDGEMENT

The authors would like to acknowledge the support of Kuwait University under research grant no. EO03/11.

REFERENCES

Adomavicius, G. & Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, **17**(6), 734-749.

Ali, B., Villegas, W. & Maheswaran, M. 2007. A trust based approach for protecting user data in social networks. In Proceedings of the 2007 conference of the center for advanced studies on Collaborative research (pp. 288-293). IBM Corp..

Al-Oufi, S., Kim, H. N. & El Saddik, A. 2012. A group trust metric for identifying people of trust in online social networks. Expert Systems with Applications, **39**(18), 13173-13181.

Backstrom, L., Boldi, P., Rosa, M., Ugander, J. & Vigna, S. 2012. Four degrees of separation. In Proceedings of the 3rd Annual ACM Web Science Conference (pp. 33-42). ACM.

Berger, C. R. 1986. Uncertain outcome values in predicted relationships uncertainty reduction theory then and now. Human Communication Research, **13**(1), 34-38.

Bonaccio, S. & Dalal, R. S. 2006. Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences. Organizational Behavior and Human Decision Processes, **101**(2), 127-151.

Brass, D. J., Butterfield, K. D. & Skaggs, B. C. 1998. Relationships and unethical behavior: A social network perspective. Academy of Management Review, **23**(1), 14-31.

Capra L.2004. Toward a human trust model for mobile ad-hoc networks. In: Proceedings of second UK UbiNet workshop.

Cha, M., Haddadi, H., Benevenuto, F. & Gummadi, P. K. 2010. Measuring user influence in twitter: The Million Follower Fallacy. ICWSM, 10, 10-17.

Chen, W. & Fong, S. 2010. Social network collaborative filtering framework and online trust factors: a case study on Facebook. In Digital Information Management (ICDIM), 2010 Fifth International Conference on (pp. 266-273). IEEE.

Dellarocas, C. 2001. Analyzing the economic efficiency of eBay-like online reputation reporting mechanisms. In Proceedings of the 3rd ACM Conference on Electronic Commerce (pp. 171-179). ACM.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. Numerische mathematik, **1**(1), 269-271.

Golbeck, J. A. 2005. Computing and applying trust in web-based social networks.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. & Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), **22**(1), 5-53.

Huang, C., Chen, Y., Wang, W., Cui, Y., Wang, H. & Du, N. 2010. A novel social search model based on trust and popularity. In Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on (pp. 1030-1034). IEEE.

Jiang, W., Wang, G. & Wu, J. 2012. Generating trusted graphs for trust evaluation in online social networks. Future Generation Computer Systems.

Jøsang, A., Ismail, R. & Boyd, C. 2007. A survey of trust and reputation systems for online service provision. Decision support systems, **43**(2), 618-644.

Jøsang, A. 1999. An Algebra for Assessing Trust in Certification Chains. In NDSS (Vol. 99, p. 6th).

Kazienko, P. & Musial, K. 2006. Recommendation framework for online social networks. In Advances in Web Intelligence and Data Mining (pp. 111-120). Springer Berlin Heidelberg.

Kuter, U. & Golbeck, J. 2007. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In AAAI (Vol. 7, pp. 1377-1382).

Lazzari, M. 2010. An experiment on the weakness of reputation algorithms used in professional social networks: the case of Naymz. In Proceedings of the IADIS International Conference e-Society (pp. 18-21).

Lee, S., Yang, J. & Park, S. Y. 2004. Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. In Discovery Science (pp. 396-402). Springer Berlin Heidelberg.

Maheswari, S. & Karpagam, G. 2010. Empirical Evaluation of Reputation Based Trust In Semantic Web. International Journal of Engineering Science and Technology, **2**(10), 5672-5678.

Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A. & Stumme, G. 2009. Evaluating similarity measures for emergent semantics of social tagging. In Proceedings of the 18th international conference on World wide web (pp. 641-650). ACM.

Massa, P. & Avesani, P. 2007. Trust metrics on controversial users: Balancing between Tyranny of the majority. International Journal on Semantic Web and Information Systems (IJSWIS), **3**(1), 39-64.

Mori, Kristen. 2008.Trust-Networks in Recommender Systems. Master's Projects. Dept. Computer Science, San Jose State University.

Nisan &Noam 2007. Algorithmic game theory. Cambridge University Press.

O'Doherty, D., J,ouili, S. & Van Roy, P. 2012. Trust-based recommendation: an empirical analysis. In Submitted to: Proceedings of the Sixth ACM SIGKDD Workshop on Social Network Mining and Analysis SNA-KDD, Beijing, China, ACM.

O'Donovan, J. & Smyth, B. 2005. Trust in recommender systems. In Proceedings of the 10th international conference on Intelligent user interfaces (pp. 167-174). ACM.

Pitsilis, G. & Marshall, L. 2006. A trust-enabled P2P recommender system. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on (pp. 59-64). IEEE.

Resnick, P. & Varian, H. R. 1997. Recommender systems. Communications of the ACM, 40(3), 56-58.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.

Sarda, K., Gupta, P., Mukherjee, D., Padhy, S. & Saran, H. 2008. A distributed trust-based recommendation system on social networks. In 2nd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb 2008). IEEE (December 2008).

Schein, A. I., Popescul, A., Ungar, L. H. & Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 253-260). ACM.

Smith, M., Milic-Frayling, N., Shneiderman, B., Mendes Rodrigues, E., Leskovec, J. & Dunne, C.2010. NodeXL: a free and open network overview, discovery and exploration add-in for Excel 2007/2010, http://nodexl.codeplex.com/ from the Social Media Research Foundation, http://www.smrfoundation.org

Sun, Y. L., Yu, W., Han, Z. & Liu, K. R. 2006. Information theoretic framework of trust modeling and evaluation for ad hoc networks. Selected Areas in Communications, IEEE Journal on, **24**(2), 305-317.

Tseng, S. & Fogg, B. J. 1999. Credibility and computing technology. Communications of the ACM, **42**(5), 39-44.

Victor, P., Cornelis, C., De Cock, M. & Teredesai, A. 2011. Trust-and distrust-based recommendations for controversial reviews. IEEE Intelligent Systems, **26**(1), 48-55.

Walter, F. E., Battiston, S. & Schweitzer, F. 2008. A model of a trust-based recommendation system on a social network. Autonomous Agents and Multi-Agent Systems, **16**(1), 57-74.

Wang, J., Yin, J., Liu, Y. & Huang, C. 2011. Trust-based collaborative filtering. In Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on (Vol. 4, pp. 2650-2654). IEEE.

Weng, J., Miao, C. & Goh, A. 2006. Improving collaborative filtering with trust-based metrics. In Proceedings of the 2006 ACM symposium on Applied computing (pp. 1860-1864). ACM.

Ziegler, C. N. & Golbeck, J. 2007. Investigating interactions of trust and interest similarity. Decision Support Systems, **43**(2), 460-475.

Submitted: 20/03/2013
Revised: 11/02/2014
Accepted: 12/02/2014

Journal of Engg. Research Vol. 2 - (2) June 2014 pp. 49 - 61

تحليل الضغط لمجال المويجات للفتحة الاصطناعية للتداخل السونارية التفاضلية للصور الرادارية (Dinsar)

*راشد حسين و **عبدالرحمن ميمن

* أستاذ مساعد ونائب مدير كلية هندسة العلوم والتكنولوجيا - جامعة همدرد كراتشي
rashid.hussain@handard.edu.pk - باكستان - 74600

** عميد كلية هندسة العلوم والتكنولوجيا - جامعة همدرد كراتشي 74600 - باكستان

الخلاصة

يشرح هذا البحث عن أساليب ضغط الحزمة على أساس العويجات للفتحة الاصطناعية للتداخل السونارية التفاضلية للصور الرادارية (DInSAR). الفتحة الاصطناعية الرادارية (SAR) هي تكنولوجيا تصوير الرادار القمر الصناعي، والتي تستخدم لالتقاط البيانات للفترات المختلفة مثل الليل أو النهار ولمختلف الطقوس، مثل العديد من أوهام الأقمار الصناعية، يمكن ضغط الصور الرادارية (DInSAR) أثناء الاسترجاع، النقل والتخزين، تفتيت الضغط الأمثل مطلوبة للحفاظ على المعلومات المحتوية في الصورة عالية الطيف.

توضح هذه الدراسة تحليل الضغط ما بعد المعاجلة للصور (DInSAR) باستخدام حزم المويجات مع التركيز على انتقاء مناسب للأ... المويجية والمستوى العتبة للأساليب التجريبية.

السلوكيات المختلفة للضغط تسمح لنا بالتصميم واختيار الأم المويجية/ الأساليب العتبة لتحقيق الأداء الأمثل. ولوحظ أن وظائف المويجات سيمليت تملك الأداء المتسق من حيث معنى الخطأي لمربع (MSE) وإشارة الذروة إلى قيم نسبة الضوضاء (PSNR). الأم المويجية bior 3، 7 أظهرت أسواء الأداء. نجح البحث التحقيقي في تقديم الأداء المحسن للضغط من الأم المويجية للصور الرادارية (DInSAR).

الكلمات الرئيسية: الفتحة الصناعية الرادارية، المستوى العتبة للأساليب التجريرية، صورة مضغوطة، صور رادار القمر الصناعي والاستشعار عن بعد.