

A Social Trust Based Friend Recommender for Online Communities “Invited Paper”

Surya Nepal, Cecile Paris, Payam
Aghaei Pour,
CSIRO Computational Informatics
Sydney
{Fname.Lname}@csiro.au

Sanat Kumar Bista
CSIRO Computational Informatics
Canberra
{Fname.Lname}@csiro.au

Jill Freyne
CSIRO Computational Informatics
Sydney
{Fname.Lname}@csiro.au

Abstract—Recommendations to connect like-minded people can result in increased engagement amongst members of online communities, thus playing an important role in their sustainability. We have developed a suite of algorithms for friend recommendations using a social trust model called STrust. In STrust, the social trust of individual members is derived from their behaviours in the community. The unique features of our friend recommendation algorithms are that they capture different behaviours by (a) distinguishing between passive and active behaviours, (b) classifying behaviours as contributing to users’ popularity or engagement and (c) considering different member activities in a variety of contexts. In this paper, we present our social trust based recommendation algorithms and evaluate them against algorithms based on the social graph (such as Friends-Of-A-Friend). We use data collected from the online CSIRO Total Wellbeing Diet portal which has been trialled by over 5,000 Australians over a 12 week period. Our results show that social trust based recommendation algorithms outperform social graph based algorithms.

Keywords—Social Networks, Online Communities, Social Trust, Recommender System, Friend Recommender.

I. INTRODUCTION

Recommendation systems have become an essential part of information systems in today’s interconnected world, where the quantity of information is soaring every day. The information deluge makes it impossible for users to sift through the vast amount of available data and find what they want. Recommender systems have been successfully used in different application domains (e.g., social, health, entertainment, commercial, etc.) for a variety of purposes, ranging from finding the right partners on dating websites to finding good places for holidays.

We classify recommender systems into two generic categories based on the recommended objects: “things” or “people”. Recommender systems in the first category include products [14], movies [13] or music recommendations [27], whereas the latter category includes potential partners [19], friends [5] or experts (e.g., doctors) recommendations [21], etc. Our focus here is on the latter category, more specifically recommending friends.

Recently, online communities have been used in a variety of applications ranging from tourism and health to government. Friend recommendations can play an important role in these communities to foster engagement.

There are two major approaches to recommending friends. The first, the *content based approach*, is based on the assumption that people with similar profiles (e.g., socio-economic background, stated interest, shared groups, etc.) can make for good friends. Recommendations are thus made by matching members’ profiles [5] [19]. The second is based on what is termed the *social graph* and exploits features such as mutual connections [22]. One of the most popular methods in this category recommends people based on Friends-Of-A-Friend (FOAF) connections.

Approaches based on FOAF work well for connecting existing friends, e.g., in environments where there exists friendship relations in the real world, and the network aims to connect existing friends and establish new friendships (such as Facebook¹ and LinkedIn²). However, these methods do not fare well in online communities where members are anonymous, or in transient communities built for specific purposes. Furthermore, these approaches tend to have a cold start problem due to the absence of a social graph or connections to start a recommender system at the beginning of the community.

To address this problem, we propose a *social trust* based recommender system, where the social trust among members is derived from their *behaviour* or interactions with the system. Studies in social and behaviour sciences show that similar behaviours and interests are the driving force in building friendships [28]. This motivated us to use a holistic behaviour based approach to recommend friends in transient anonymous online communities, where the behaviour of an individual member is measured using the density of interactions.

There are other recommender systems based on trust. They, however, typically use *explicit trusts* specified by users in a social graph [12] [15] [1]. Such approaches do not work well in a transient anonymous online community, where users do not yet have enough knowledge to rate others based on trust. In contrast, our social trust based model captures the implicit trust members have towards each other as exhibited through their behaviours in the community. Furthermore, current recommenders that make use of interactions capture only users’ active behavior (e.g., writing a post) and ignore passive behaviours (e.g., reading

¹ www.facebook.com

² www.linkedin.com

a post). Our social trust model captures both active and passive behaviours.

The remainder of the paper is organised as follows. Section II presents related work. Section III describes our social trust based algorithms and the social graph based algorithms we use in our evaluation. In section IV, we explain our experimental setting, the data set and our evaluation metrics. Section V presents the evaluation results and a discussion of the results. We conclude and outline potential future work in section VI.

II. RELATED WORK

Recommender systems have been used in social networking systems to recommend people and content of interest and to sustain participation. We briefly present here relevant work in this area.

A. Friend Recommendation

The role of friends in a social network is to generate and share interesting content for others to browse, to provide feedback to others on their content and to generally contribute to a vibrant community space. Much work in recent years has focused on identifying and recommending people to friend in social networks [5] [7] [11] [19]. In some cases, recommenders assist people in completing their virtual friendship circle. In others, recommenders identify influential people, experts or potential partners to provide a valuable service to network users.

Chen *et al.* explored the use and performance of several recommendation algorithms in an enterprise social network to assist existing users in the completion and expansion of their network [5]. They found that social algorithms allowed people to find those that they already knew in real life, and similarity algorithms assisted in discovering new people. Guy *et al.* employed content based algorithms, exploiting information contained on a company intranet to identify “interesting strangers” for users to assist them in broadening their friendship circle [11]. Friendsense is a mobile friend recommender which looked beyond the data contained in a social network and used physical proximity of people to generate recommendations of friends and connections for a target user [20].

Recommender systems have also been employed to address the issue of attrition in social networks, a challenge particularly relevant when the focus for the networks is less on entertainment and information and more on achieving change, as is the case of diet and lifestyle networks such as that used in this paper. Freyne *et al.* employed a content based recommender algorithm to recommend friends and suggest content to contribute to new members of a social network [7]. They found that the effect of the recommender was to increase engagement with the network over a 6 month period and reduce attrition rates.

B. Trust Based Recommendation

Trust models for social networks can be classified into three groups: graph based trust models, interaction based trust models and hybrid trust models. Graph based trust

models exploit the inherent structural properties of social graphs. For example, Golbeck *et al.* proposed a method for creating a trust network on the Semantic Web by extending the FOAF schema to allow users to indicate a level of trust for people they know [9]. Interaction based trust models exploit the interaction data, such as engagement and popularity in our STrust model [16] [17].

Gilbert and Karaholios [8] carried out a Facebook study with over 1000 participants, observed their behaviour online and surveyed users on their actual relationships with friends listed online. Their derived model, based on online interactions, identifies strong and weak ties in social networks with an accuracy of 85%. Similarly, Wu *et al.* [26] developed a model for computing professional, personal and general closeness of people on an enterprise social networking system, Beehive. Both studies stopped short of applying their models to recommendations. Some work has used the knowledge acquired from models of interactions and their relationship to tie strength to generate personalised news or activity feeds. Paek *et al.* [18] used classifiers to identify the most predictive features of relationship strength to create tailored news feeds. Guy *et al.* [10] considered the content of the activity feeds for profiling users of an enterprise social network. Freyne *et al.* [6] and Berkovsky *et al.* [2] looked beyond the interactions of people on social networking sites and included interaction patterns with types of content to inform their models and generate recommendations.

Interaction based social trust models consider interactions in the community to compute trust but ignore the network structure characteristics. Hybrid models exploit the benefits of both interactions and social graphs. Trifunovic *et al.* proposed such a model for applications such as content distribution and micro-blogs [25]. The model leveraged explicit social trust, based on consciously established social ties/graphs, and implicit trust, based on frequency and duration of contact between two users.

These models have been exploited to generate personalised recommendations by aggregating the opinions of other users in the trust network [12] [15] [1]. For example, Hang *et al.* used a social graph approach to recommend a node in a social network using a similarity measure [12]. Massa *et al.* proposed a trust-based recommendation system to search for trustable users by exploiting trust propagation [15], whereas Andersen *et al.* proposed several recommendation models to provide factual information [1]. For further details and other examples, we refer to [24].

In this work, we focus on the use of interaction data to estimate user trust, which is similar to tie strength, to inform a people recommender system for social networks. We hypothesise that, by monitoring user interactions and behaviours, we can identify relevant individuals to friend and thus encourage increased friending levels in online social networks, in a similar effort to [4].

III. RECOMMENDATION ALGORITHMS

We define behaviour based friend recommendation algorithms for online communities using two principles: Social Trust (STrust) and Social Graph (FOAF).

A. STrust Model

Our social trust model, STrust [16] [17], captures two types of behaviour amongst community members: trusted and trusting. Trusted people are the ones towards whom others in the community have shown trust, e.g., by providing positive ratings to their posts and comments in forums. Similarly, a member can exhibit trusting behaviour towards others by giving them positive rating or by frequently viewing their forum or blog posts.

Social Trust is composed of two types of trust: popularity trust (*PopTrust*) and engagement trust (*EngTrust*). In brief, the popularity trust of a member measures the trusting behaviour of other members in the community towards the member, whereas the engagement trust refers to the behaviour of a member towards other members. Trust relationships can be represented as a graph: an arrow from A to B contributes to the engagement trust of A and the popularity trust of B. Trust relationships are not symmetrical (i.e., A trusting B does not necessarily mean B trusting A). Trusting behaviour can be active (e.g., rating a post) or passive (e.g., reading a post). We now briefly describe the STrust model, followed by the corresponding friend recommendation algorithms.

Let M be the number of members in an online community, and m_i and m_j represent two members. If m_i has a positive/negative interaction with m_j , it is represented as “+”/“-”. m_j 's popularity trust (*PopTrust*) is defined as:

$$PopTrust(m_j) = \frac{\sum_{i=1, i \neq j}^M \frac{|PT_{ij}^{kd+}|_{+1}}{|PT_{ij}^{kd+}|_{+1} + |PT_{ij}^{kd-}|_{+2}}}{M-1} \quad (1)$$

where PT_{ij}^{kd+} represents the positive interactions that m_i has had with m_j with respect to any activity d in context k . A member in the community may interact with other members through a number of activities related to a single context. For example, a member may comment, rate or view a post in the forum. Here, the forum represents the *context*, and commenting, rating and viewing are considered as *activities*. It is possible for each activity and context to have different weights. Let w_d represent the weight for activity d , with $\sum w_d = 1$. Let w_k represent the weight for context k , with $\sum w_k = 1$. Thus, considering all these, PT_{ij}^{kd+} and PT_{ij}^{kd-} are defined as follows:

$$|PT_{ij}^{kd+}| = \sum_{k=1}^K w_k (\sum_{d=1}^D w_d (\sum_{x=1}^X +1)) \quad (2)$$

$$|PT_{ij}^{kd-}| = \sum_{k=1}^K w_k (\sum_{d=1}^D w_d (\sum_{x=1}^X -1)) \quad (3)$$

where K represents the number of *contexts*, D the number of *activities* in each context, and X the number of *interactions* related to the activity and context. Similarly, the engagement trust (*EngTrust*) of m_j is defined as:

$$EngTrust(m_j) = \frac{\sum_{i=1, i \neq j}^M \frac{|ET_{ji}^{kd+}|_{+1}}{|PT_{ji}^{kd+}|_{+1} + |PT_{ji}^{kd-}|_{+2}}}{M-1} \quad (4)$$

We then define the social trust of a member m_j as follows:

$$SocialTrust(m_j) = \alpha \cdot PopTrust(m_j) + (1 - \alpha) \cdot EngTrust(m_j) \quad (5)$$

where α represents the value of a weight in the range of 0 to 1.

B. Social Trust Based Algorithms

We define five algorithms based on the social trust model just described. The first three exploit the overall trusting behaviour of a member in the community, whereas the next two use the trusting interactions between two members. Their pseudo code is shown in Algorithm 1. It has three input parameters: (a) the member who will receive the recommendations, (b) the number of recommendations to be made, and (c) the specific algorithm to be used. The five algorithms are described in brief below.

MostPopular: This algorithm assumes that the most popular member, followed or watched by many, has a high probability of receiving invitations. A high popularity also indicates a propensity to accept friend invitations. In this algorithm, we sort all members based on their popularity trust and recommend the top N popular members not already in their friend list (Algorithm 1: lines 19 to 21).

MostEngaged: In this algorithm, we assume that the person who is most engaged in the community (e.g., through posting, rating) is likely to receive friendship invitations from other members because of their visibility in the community. Active engagement also means that one is likely to accept invitations. In this algorithm, we sort all members based on their engagement trust and recommend the top K engaged members not already in one's friend list (Algorithm 1: lines 22 to 24).

MostSocial: This algorithm is based on the overall social trust in the community. This is a combination of the earlier two algorithms: *MostPopular* and *MostEngaged*. This algorithm recommends either the most popular or most engaged members. In this algorithm, we give equal weights to popularity trust and engagement trust (i.e., $\alpha = 0.5$) (Algorithm 1: lines 25-27).

We have so far discussed the algorithms based on the members' overall behaviours in the community. We believe that a friend recommendation algorithm would be more targeted if we considered the behavioural relationships between two members. We now describe two such algorithms. A behaviour relationship exists between two members when a member performs an activity on another member's activity in a certain context. For example, member A rates a comment by member B on a forum post. This can be stated as: if m_i has had a positive activity with his or her interactions with m_j in a specific context (e.g.,

forum), the social trust between m_i and m_j for a context k and an activity d is defined as follows:

$$SocialTrust(m_i, m_j, k, d) = \alpha \cdot PopTrust(m_i, m_j, k, d) + (1 - \alpha) \cdot EngTrust(m_i, m_j, k, d) \quad (6)$$

Here, $PopTrust(m_i, m_j, k, d)$ and $EngTrust(m_i, m_j, k, d)$ represent the popularity and engagement trust between members j and i in context k with regards to activity d . This is calculated using equations 1- 4. Now, we exploit these relationships in two recommender systems.

Engagement: We first consider recommending two members to be friends with each other if they have highly positive engagements. Engagement between two members is a good indication of potential friendship between them. We use a majority rule in this case. The recommender system recommends a member m_i to invite another member m_j to be a friend for a certain context k under the following conditions (Algorithm 1: lines 28-36):

1. The engagement trust between m_i and m_j is greater than the overall engagement trust of m_i (considering only interacting members).

$$EngTrust(m_i, m_j, k, d) = OverallEngTrust(m_i)$$

$OverallEngTrust(m_i)$

$$= \frac{\sum_{j=1, j \neq i, j \approx i, j \equiv i}^M \frac{|ET_{ij}^{kd+}| + 1}{|ET_{ij}^{kd+}| + |ET_{ij}^{kd-}| + 2}}{\sum_{j=1, j \neq i, j \approx i, j \equiv i}^M + 1}$$

Where $j \approx i$ means no friendship exists between member m_i and m_j ; $j \equiv i$ means there is at least one interaction between member m_i and m_j .

2. The number of positive interactions of m_i with m_j is greater than average number of positive interactions m_i has with other members.

$$|ET_{ij}^{kd+}| \geq \sum_{x=1, x \neq j, j \approx x, j \equiv x}^M \frac{|ET_{jx}^{kd+}|}{\sum_{y=1, y \neq w, y \approx w, y \equiv w}^M + 1}$$

3. The number of positive interactions of m_i is greater than certain threshold (essentially meaning that m_i has some amount of engagement in the community).

$$|ET_{ij}^{kd+}| \geq th$$

Popularity: The assumption here is that if a member reads, comments and rates other member's comments or posts, they have high propensity to be friends. The algorithm includes three conditions akin to those in the previous algorithm (Algorithm 1: lines 37-46).

C. Social Graph

One of the most common social graph is a 'friends' graph. The friend relationship is mutual, i.e., if A is a friend of B,

Algorithm 1: SocialBehaviourRecommendation (mi, N, t)

```

1. Input: member mi, number of recommendation N, type of algorithm t
2. Output: list of recommendation R;
3. M ← all members; D ← all activities; K ← all contexts;
4. foreach  $m_x \in M$  do
5.   foreach  $d \in A, k \in C$  do
6.      $PT_{ix}^{kd} \leftarrow ComputePopInteractions(m_i, m_x, k, d)$ ;
7.      $ET_{ix}^{kd} \leftarrow ComputeEngInteractions(m_i, m_x, k, d)$ ;
8.      $PopTrust_{ix} \leftarrow ComputeMutualPopTrust(PT_{ix}^{kd})$ ;
9.      $EngTrust_{ix} \leftarrow ComputeMutualEngTrust(ET_{ix}^{kd})$ ;
10.  endfor
11.   $PopTrust_x \leftarrow ComputeIndividualPopTrust()$ ;
12.   $EngTrust_x \leftarrow ComputeIndividualEngTrust()$ ;
13.   $SocialTrust_x \leftarrow ComputeIndividualSocialTrust()$ ;
14.   $AvgPopTrust_i \leftarrow ComputeAvgPopularity(PopTrust_{ix})$ ;
15.   $AvgEngTrust_i \leftarrow ComputeAvgEngagement(EngTrust_{ix})$ ;
16.   $AvgEngInteraction_i \leftarrow ComputeAvgEngInteractions(ET_{ix}^{kd})$ ;
17.   $AvgPopInteraction_i \leftarrow ComputeAvgPopInteractions(PT_{ix}^{kd})$ ;
18. endfor
19. if (t == 'MostPopular'){
20.    $R \leftarrow \{m_1, m_2 \dots m_x: PopTrust_x \geq PopTrust_{x+1} \forall x \ \& \ \text{not}(\text{Friend}(m_i, m_x))\}$ 
21.    $R \leftarrow R \setminus \{m_x: x > N\}$ 
21. }else if (t == 'MostEngaged'){
22.    $R \leftarrow \{m_1, m_2 \dots m_x: EngTrust_x \geq EngTrust_{x+1} \forall x \ \& \ \text{not}(\text{Friend}(m_i, m_x))\}$ 
23.    $R \leftarrow R \setminus \{m_x: x > N\}$ 
24. }else if (t == 'MostSocial'){
25.    $R \leftarrow \{m_1, m_2 \dots m_x: SocialTrust_x \geq SocialTrust_{x+1} \forall x \ \& \ \text{not}(\text{Friend}(m_i, m_x))\}$ 
26.    $R \leftarrow R \setminus \{m_x: x > N\}$ 
27. }else if (t == 'Engagement'){
28.   foreach  $m_x \in M$  do
29.     if ( $EngTrust_{ix} \geq AvgEngTrust_i$  &  $ET_{ix}^{kd} \geq AvgEngInteraction_i$  &  $ET_{ix}^{kd} \geq th$  &  $\text{not}(\text{Friend}(m_i, m_x))$ ){
30.        $R \leftarrow m_x$ 
31.     }
32.   endfor
33.    $R \leftarrow \{m_1, m_2 \dots m_x: EngTrust_{ix} \geq EngTrust_{i(x+1)} \forall x\}$ 
34.    $R \leftarrow R \setminus \{m_x: x > N\}$ 
35. }else if (t == 'Popular'){
36.   foreach  $m_x \in M$  do
37.     if ( $PopTrust_{ix} \geq AvgPopTrust_i$  &  $PT_{ix}^{kd} \geq AvgPopInteraction_i$  &  $PT_{ix}^{kd} \geq th$  &  $\text{not}(\text{Friend}(m_i, m_x))$ ){
38.        $R \leftarrow m_x$ 
39.     }
40.   endfor
41.    $R \leftarrow \{m_1, m_2 \dots m_x: PopTrust_{ix} \geq PopTrust_{i(x+1)} \forall x\}$ 
42.    $R \leftarrow R \setminus \{m_x: x > N\}$ 
43. }
44. }
45. return (R);

```

then B is a friend of A. These relationships can thus be represented as a bidirectional graph. Let G_f represents the friendship social graph in the community, where nodes $n \in N$ represent members of the community, and the link $l \in L$ represents the friendship between members.

Algorithm 2: SocialGraphRecommendation (mi, N, t)

```

1. Input: member mi, number of recommendation N, type of algorithm
   t
2. Output: list of recommendation R;
3.  $M \leftarrow$  all members
4.  $F \leftarrow \{(m_x, m_y), \dots : Friend(m_x, m_y)\}$ 
5. if ( $t ==$  'FOAF'){
6.   foreach  $m_x, m_y \in M$  do
7.      $Friends_x \leftarrow m_y : (m_x, m_y) \in F;$ 
8.   Endfor
9.   foreach  $m_x \in Friends_x$  do
10.    foreach  $m_y \in Friends_x$  &  $m_y \neq m_x$  do
11.       $FriendsOfaFriend_i \leftarrow m_y;$ 
12.    Endfor
13.  endfor
14.   $R \leftarrow \{(m_x, P) : P = \text{count}(m_x \in FriendsOfaFriend_i)\}$ 
15.   $R \leftarrow R \setminus \{m_1, m_2 \dots m_x : P_x \geq P_{x+1}\}$ 
16.   $R \leftarrow R \setminus \{m_x : x > N\}$ 
17. }else if ( $t ==$  'MostFriends'){
18.   foreach  $m_x, m_y \in M$  do
19.     if ( $m_x \neq m_i$  &  $m_y \neq m_i$ ){
20.        $Friends_x \leftarrow m_y : (m_x, m_y) \in F;$ 
21.     }
22.   endfor
23.    $R \leftarrow \{(m_x, P) : P = \text{count}(m_x \in Friends_x)\}$ 
24.    $R \leftarrow R \setminus \{m_1, m_2 \dots m_x : P_x \geq P_{x+1}\}$ 
25.    $R \leftarrow R \setminus \{m_x : x > N\}$ 
26. }
27. return (R);

```

Let $n_1 \leftrightarrow n_2$ represents the links between nodes n_1 and n_2 in G_f , then the friends in the community is represented as:

$$Friend(m_i, m_j) = \{\exists n_i \leftrightarrow n_j \in G_f \wedge \exists m_i, m_j \in M \wedge n_i = m_i \wedge n_j = m_j\} \quad (7)$$

Similarly, the set of a pair of friends in the community is represented as:

$$F = \{(m_i, m_j) : Friend(m_i, m_j)\} \quad (8)$$

The set of friends of a member in the community is represented as: $F_i = \{m_j : (m_i, m_j) \in F\}$.

One of the important relationships in a social graph is a FOAF. It is defined as follows:

$$FOAF = \{(m_i, m_k) : \forall m_j \in F_i \wedge (m_j, m_k) \in F \wedge m_k \neq m_i\}$$

$$FOAF_i = \{m_j : (m_i, m_j) \in FOAF\} \quad (9)$$

D. Social Graph Based Algorithms

We now define two algorithms based on social graphs. Their pseudo code is shown in Algorithm 2. It has three input parameters similar to that of Algorithm 1. The first algorithm, FOAF, recommends the friends of a friend (lines 5-16), as defined earlier, with the assumption that someone is likely to befriend the friends of their friends. In this algorithm, members are sorted based on the number of common friends (lines 14-15). If the number is greater than the requested number of recommendations, the ranked list with the requested number of members is returned (line 16).

The second algorithm is MostFriends (lines 17-25). In this algorithm, members are ranked based on the number of friends they have. Unlike FOAF, this algorithm does not care about the connections with the member to whom the recommendation is generated. The underlying assumption

here is that people who already have a large number of friends are likely to have yet more friends, i.e., a member is likely to want to be friend with the member who already has a large number of friends, and that member is likely to accept friendship requests.

IV. EXPERIMENTAL SETTINGS

We hypothesise that, when articulated friendship networks are small and data on users scarce, our algorithms will outperform generic and FOAF algorithms. Our evaluations are done on a set of interaction logs gathered from the Online Total Wellbeing Diet (TWD) Portal, a social networking platform for adults embarking on an online diet [3].

A. Online Portal and Data Set

The TWD portal was typical of online social networks such as Facebook and MySpace³ where each user is represented by a profile page containing a profile image, some personal information, attributes related to food, exercise and interests, a photo gallery, a friends list, a wall (message board) and a personal blog. Only friends could read or write on a person's wall. Friendship requests were possible by clicking on an "Add Friend" button on the profile pages. Requests could be accepted, rejected or ignored. A discussion forum on the portal's Home page supported group discussions and question answering. All interactions with the site by participants were recorded.

The TWD online portal was trialled for 12 weeks with 8,112 registered participants, of which 5,279 became active members (i.e., members who have logged in at least once). We only considered member activities from forum and blog data to generate our social trust scores, as these represent members' behaviours toward each other. They are shown in Table 1. There were 1,810 unique members with at least one activity in the forum or a blog. Among them, 1,770 had at least one engagement type activity and 543 at least one popularity type activity. The data set includes 96,827 activities performed by the 1,770 members, 86% of which are passive. Active activities include 1,881 rating actions and 11,489 comments. The blog attracted more activities than the forum.

Table 1. Members' activity counts in our data set (excluding friendship requests – see Table 2)

Context →	Forum			Blog			Total Interactions
	Comment	Post	Total	Comment	Post	Total	
Ratings	1013	122	1135	0	746	746	1881
Commenting	1933	904	2837	1446	7206	8652	11489
Viewing	0	19366	19366	0	64091	64091	83457
Total	2946	20392	23338	1446	72043	73489	96827

We extracted relevant data from the portal for the purpose of our experiments in two steps as follows.

Step 1: Constructing Two-Mode Behaviour Network: We first extracted the behaviour for all members who participated in the blogs and forum. An individual

³ www.myspace.com

member’s behaviour provides all the activities performed by the member in different contexts. This means it provides us a two mode network.

Step 2: Constructing One-Mode Interaction Network: As the two-mode network obtained in step 1 does not provide all interactions between the members that are essential for our friend recommendation algorithms, we converted it to a one-mode network. If two users have viewed the same post, we have considered it as a viewing interaction between them.

The interaction logs had some limitations that impacted on our analysis. Viewings comments of forum and blog were not distinguishable from viewing the forum and blog posts themselves, in part due to the interface that listed comments under the posts. Rating of the blog comments was not permissible. Finally, the trial site experienced some technical difficulties in its first few days and became unreliable and unavailable for a time. However, friendships established in this time were included in our data. Thus, we include friendship activity for the entire duration, but only viewing and commenting data from the stable period.

The friendships data used for our evaluation is shown in Table 2: 396 members sent 2,608 requests to 572 members. 357 members accepted at least one request, thereby creating a total of 2,194 friendships in the community. 21 members who rejected 29 requests, and 261 members never responded to at least one request, resulting in 270 pending requests. There were 115 unknown requests, as some members never came back to the community after receiving the requests.

Table 2. Basic Statistics of Friendship Requests

# of unique members who sent friendship requests	396
# of unique members who received friendship requests	572
# of unique members who accepted friendship request	357
# of unique members who rejected friendship requests	21
# of unique members who did not respond to requests	261
# of unique members who made friends	500
# of friendship requests	2608
# of accepted requests	2194

B. Evaluation Metrics

It is difficult to evaluate the people recommendation algorithms, as “success” can be defined in a variety of ways. For example, the successful outcome of a recommendation algorithm for an online dating community could be (a) the viewing of the profile of a recommended member, (b) the exchange of information between them, (c) a successful follow up date, and (d) a successful wedding. Therefore, it is imperative that we first define success in our context. We consider two measurements of success, one stricter than the other. The aim of the recommender system is to identify potential friends for others in an the network. Part of this is the identification of people that a requester would like to be friends with, but the second component is whether the recommended person would like to be friends with the requester. Thus, our first measure of

success is whether the algorithms can identify people that are sent friendship requests. The second measure looks at whether the algorithm can identify people that are likely to respond positively to that request, i.e., identify people that accepted the requests.

A number of measures can be used to evaluate success, such as precision, recall, accuracy, success, f-measure, mean average precision, failure, etc. The types of measures to be used in evaluating recommendation algorithms depend on the aim of the algorithms and the properties to be measured [23]. We measure performance using the traditional Precision@N and Recall@N. We define them using the confusion matrix shown in Table 3.

$$\text{Precision@N} = \frac{a}{a + b} \quad \text{Recall@N} = \frac{a}{a + c}$$

In addition to these, we use coverage [18], which measures the domain of items over which the recommendation algorithms can make recommendations. Coverage is measured using two different methods: request space coverage and member space coverage. The first is to identify the percentage of requests for which the algorithm is able to generate a recommendation list. The second is to calculate the percentage of all available users that are ever recommended to a request. In our evaluation, we use the former method, i.e., request space coverage. Thus, coverage in our context is defined as the percentage of all the actual requests for which the algorithm could generate a recommendation list of size > 0.

Table 3. Confusion matrix for Top N recommendations for each friendship request

Top N	Relevant (Requested or accepted)	Not Relevant (Not Requested or Not Accepted)
Recommended	a	b
Not Recommended	c	d

V. EXPERIMENTAL RESULTS AND ANALYSIS

Data analysis: The aim of our analysis is to answer the following questions: (a) whether our social trust based algorithms can identify appropriate friend candidates for the users of the Online TWD Portal as well as how many of those requests end up being friends, as suggested by [11]; (b) whether our social trust based algorithms are equipped to deal with the problem of cold start at the beginning of the community, when there are not enough interactions data and the social graph has not even started to build up; (c) whether the algorithms based on mutual interactions are better in predicting the accurate friendship requests in comparison to the algorithms based on overall interactions in the community; (d) whether the passive interactions have an effect on the overall performance of the algorithms. To this end, we ran an offline analysis, using articulated friendships as the ground truth.

We tested the performance of the seven algorithms described in section III B, using the metrics defined in section IV B. Five of those algorithms were based on the social trust, and two based on the social graph. The social trust based algorithms included two algorithms based on

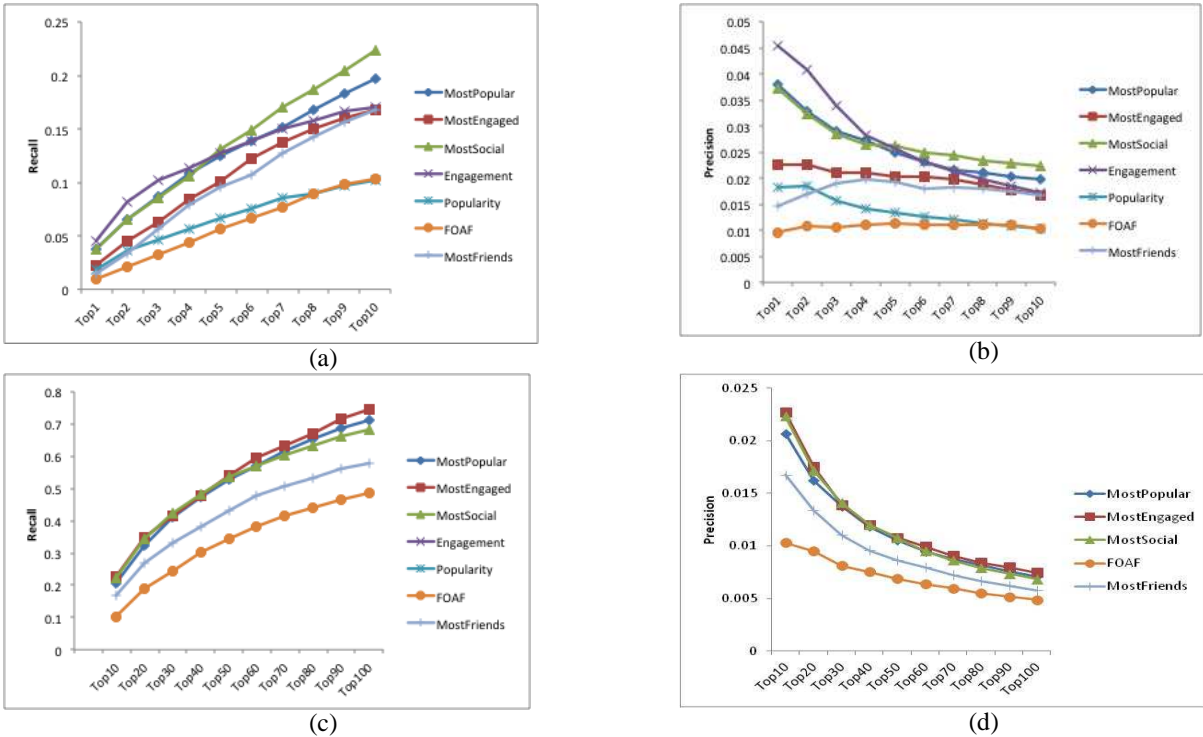


Figure 1. Recall@N and Precision@N, when the success is the identification of a request recipient.

individual interactions (Engagement, Popularity) and three community based algorithms (MostPopular, MostEngaged, and MostSocial). The graph based algorithms are baseline algorithms typically seen in friend recommendation applications (FOAF and MostFriends). They exploit only the articulated social graph.

Methodology: 2,608 friendship requests were logged by the system. For each of the friendship requests, we noted the requester, the recipient and the request time. Each algorithm generates a ranked list of friend recommendations for the requester, given the social network structure (or social graph) and interactions logged at request time. Recommendation lists containing 10 and 100 are discussed in our evaluation to show the performance of algorithms when small and large numbers of recommendations are required. The requirements of a friend recommender system could vary based on type of network in which it is deployed. We examined the small recommendation lists by varying N from 1-10 in single increments, and lists containing 100 recommendations, by varying N in intervals of 10.

A. Algorithm Coverage

We begin by discussing the algorithm coverage using the metric defined earlier, i.e., the percentage of the 2608 friendship requests for which each algorithm was able to generate a recommendation list containing at least one member of the community. Four algorithms achieve 100% coverage: MostPopular, MostEngaged, MostSocial and MostFriends. The Engagement algorithm cannot make

recommendations in 4.6% of cases, because of insufficient activity on the site to generate a recommendation. Similarly, the Popularity algorithm can only generate recommendations for 74% of the considered instances. This is a result of passive users on the network. If users have not sufficiently interacted directly with others on the network, in the form of activities such as comments and ratings, then these algorithms cannot make recommendations. FOAF achieves 88% coverage; this reduction in coverage is due to the algorithm's reliance on the target user having at least one friend, or for their friends having at minimum of one other friend.

B. Algorithm Accuracy

As mentioned, we have two measures of success; the identification of recipients, and the identification of recipients who accept the friendship requests. We start in Figure 1 by examining the performance of the algorithms when success is identifying people who received friendship requests. Figure 1 (a) shows the average Recall@N for the algorithms when lists of 1 to 10 recommendations are considered.

We note the poor performance of the FOAF and Popularity algorithms. Both of these algorithms are likely to suffer heavily due to the cold start problem. With few articulated friendships and few direct communications going on in the initial days, these algorithms struggle to generate long lists of recommendations which in turn impacts heavily on their ability to identify potential friends. We note that this poor performance continues even when lists of 100 are generated, as shown in Figure 1(c). Most

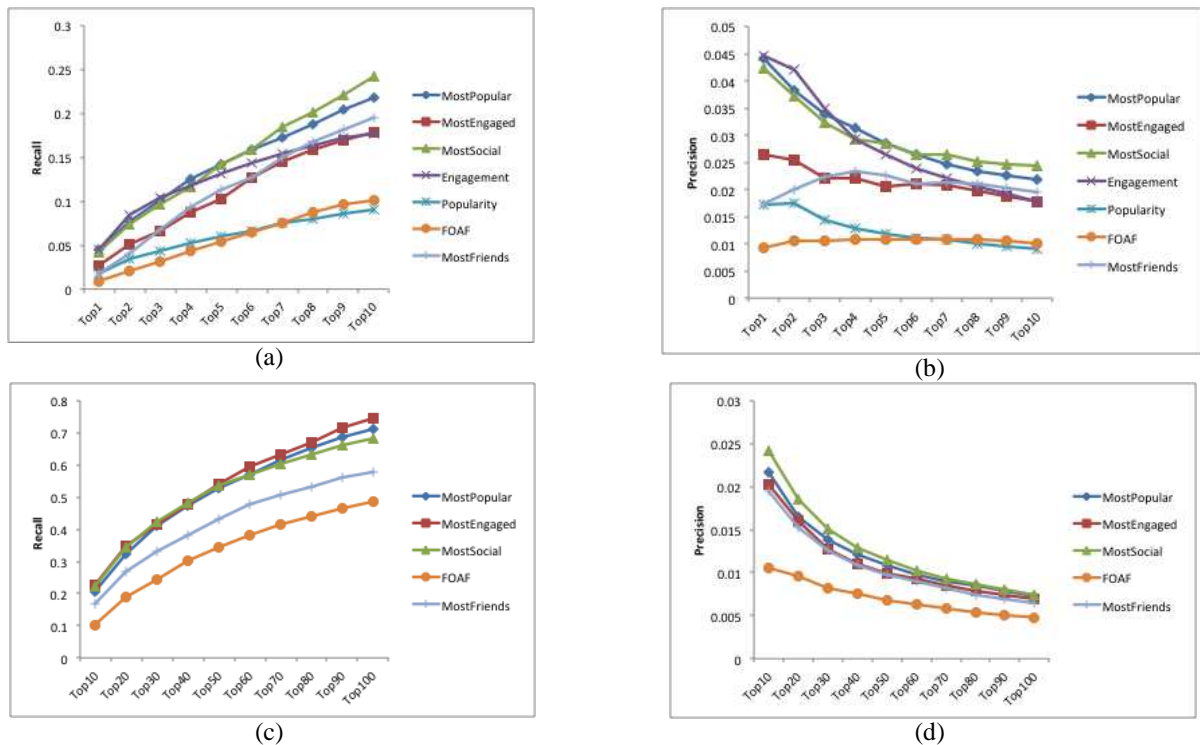


Figure 2. Recall@N and Precision@N, when the success is the correct identification of a recipient who accepted the request.

algorithms show steady increases in recall, with recall increasing as recommendation lists increase in size. The Engagement algorithm, however, is different. Its recall for $N < 3$ is greater than all others, but then its performance slows, and it is outperformed by the MostPopular and MostSocial algorithms when $N > 5$. The Engagement algorithm might be an appropriate algorithm to use in scenarios where a few recommendations are being shown to a user. This outperformance of the Engagement algorithm over the other algorithms is clearly seen in Figure 1(b), which shows the precision for each algorithm. The Engagement algorithm achieves double the precision when $N=1$ in comparison to all algorithms except the MostSocial and MostPopular, over which it has a relative increase of 30%. Once again we see the success of the algorithm being short lived, and the precision when $N \geq 5$ is more comparable to that of the other algorithms. We note the flatness of the FOAF and MostEngaged algorithms, which do not reflect traditional precision curves.

Although it is not practical to recommend more than a few members for friends, we have extended our evaluation up to top 100 in order to understand the trend in Figure 1(c) and (d). One characteristic of the Engagement and Popularity algorithms is that they produce short recommendation lists. Thus, we did not calculate precision and recall for $N=10-100$. We see a clear benefit across all N for the social trust based algorithms in comparison to the FOAF algorithm. We

note that the MostFriends algorithm does comparatively well. However, deploying the MostFriends algorithm in the real world is likely to be problematic. Recommending popular people to everyone on a network is not sustainable and would result in a small portion of the community (the popular people) receiving lots of requests and large portions of the community never being recommended to others, which is not the type of network structure that we would like to encourage.

Until now we have focused on predicting friendship requests, and we have seen that our Engagement algorithm performs best for small recommendation lists, with all social trust algorithms outperforming the social graph algorithms. We move now to our second, stricter measure of success, predicting friendships, i.e., requests that are accepted. 16% (414) of all friendship requests in the Online TWD site were rejected or ignored. Ignoring a request resulted in a pending request which could be pending and seen, or pending but unseen, as was the case if the recipient never logged into the system to see the request. Email notifications were not generated by the Online TWD system. Pending and rejected requests are, for the purpose of this analysis, considered to be the same. Figure 2 shows the performance of the algorithms in identifying the accepted friendship requests, using Recall@N and Precision@N and results lists of size 1-10 and 10-100. Once again we see the highest precision and recall for low values of N achieved by the Engagement algorithm, in Figure 2 (a) and (b), but this time followed

closely by the MostSocial and MostPopular algorithms, which overtake the Engagement algorithm when $N \geq 5$. The FOAF algorithm is clearly outperformed by the social trust based approaches. The MostPopular and MostSocial approaches outperform all others when $N > 10$.

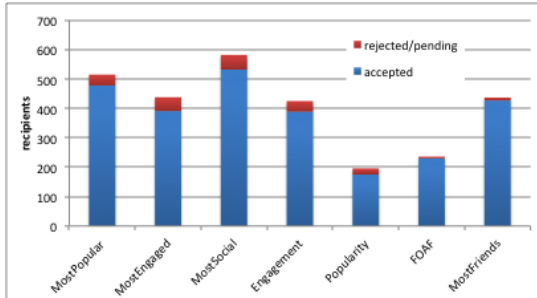


Figure 3. Ratio of accepted versus rejected or pending requests when $N=10$.

Focusing on the proportion of requests converted to friendships, Figure 3 shows the success of each algorithm with the accepted friendships in blue and the rejected and pending in red for recommendations lists of size 10. We observe that the FOAF algorithm has a low success rate, as indicated by the previous recall and precision diagrams, but we note here that the portion of its accepted hits (97%), exceeds that of social graph algorithms. Similarly, MostPopular has a 98% acceptance ratio. The ratio of the social behaviour algorithms ranges from 89-92%, but this is offset by a high precision and recall values as discussed earlier.

Table 4 shows the ratio of accepted to rejected for $N=1-10$. We note that the FOAF and MostFriends algorithms maintain a steady ratio in excess of 95% accepted friendships across all N . For the most part the Engagement algorithm is stable at 90% but we see changing ratios, however, for the other social graph algorithms with the ratio of accepted request declining as more recommendations are considered.

Table 4 Percentage Accepted requests over all N

Algorithms	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
MostPopular	98	98.2	97.8	96.5	96	96.1	95.7	94.1	93.7	92.8
MostEngaged	98.3	94.9	89	87.7	85.6	87.4	88.6	88.5	89	89.3
MostSocial	95.9	96.4	95.5	93.1	91.2	89.7	90.8	90.6	91	91.4
Engagement	86.7	90.6	90.9	91.1	91.2	91.6	91.2	91.3	91.8	91.5
Popularity	94.3	94.4	92.2	90.9	89.1	89	89	89	88.7	89.3
FOAF	95.5	98	98.6	98	96.9	97.4	97.2	97.5	97.3	97.5
MostFriends	100	98.9	99.3	99	99.2	98.9	98.8	98.7	97.6	97.7

We believe that this is due to the visibility of a FOAF network in the profile page in TWD, and the fact that a member can only accept a request through the sender's profile page. When a receiver sees some of his/her friends are also the friends of the sender, the receiver may have a higher chance of accepting the request. It is also plausible that people who have at least one friend in the network are more likely to accept and additional friend request.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a set of algorithms for friend recommendations in online communities and evaluated

their performances on a data set obtained from the CSIRO Total Wellbeing Diet portal. Overall, the results indicate that social behaviour based algorithms outperform social graph based algorithms. We make the following observations from our analysis. An individual's social behaviour in the community is a good indicator for being invited to be friends by other members in the community. On balance, social trust algorithms are good during the bootstrapping stage of the community, when there are not yet many mutual interactions in the community, and no or only few nodes in social graphs. The Engagement algorithm performs better than other algorithms if we prefer to recommend a few friends, which we believe is more realistic in online communities.

In online communities where the number of members are moderate, we could use community based algorithms, such as MostSocial, MostPopular and MostEngaged proposed in the paper. These algorithms have a propensity for recommending the same members, however.

In our future work, we will look to expand on the current analysis, focusing on two distinct areas, widening our criteria for success and examining in more detail the temporal context of the recommendations. As we mentioned, there are several parameters that could indicate a successful friendship. We used two straightforward measures here, identifying relevant people for friendship requests, and relevant people who will accept such a friendship request. In the future, we will look beyond these to the friendships that were established, and look at active and passive interactions to judge the value of the friendship and its role in sustaining participation with the community. We plan to carry out further analysis into the performance of our algorithms at various time points, to understand how the algorithms are impacted upon by the size and density of the community and the levels of interactions. In addition, we plan to look at other recommendation applications, beyond friend recommendations which could leverage our social trust based models to recommend blogs and forum posts as well as content and updates.

REFERENCES

- [1] Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., and Tennenholtz, M., 2008. Trust-based recommendation systems: an axiomatic approach. In *Proceedings of the 17th international conference on World Wide Web* (Beijing, China), ACM, 199-208.
- [2] Berkovsky, S., Freyne, J., and Smith, G., 2012. Personalized network updates: increasing social interactions and contributions in social networks. In *Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization* (Montreal, Canada), Springer-Verlag, 1-13.
- [3] Brindal, E., Freyne, J., Saunders, I., Berkovsky, S., Smith, G., and Noakes, M., 2012. Features Predicting Weight Loss in Overweight or Obese Participants in a Web-Based Intervention: Randomized Trial. *Journal of medical Internet research* 14, 6.

- [4] Chelmiss, C. and Prasanna, V.K., 2012. Predicting Communication Intention in Social Networks. In *ASE/IEEE Fourth International Conference on Social Computing* ASE/IEEE, Amsterdam, Netherlands.
- [5] Chen, J., Geyer, W., Dugan, C., Muller, M., and Guy, I., 2009. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA), ACM, 201-210.
- [6] Freyne, J., Berkovsky, S., Daly, E.M., and Geyer, W., 2010. Social networking feeds: recommending items of interest. In *Proceedings of the 4th ACM conference on Recommender systems* (Barcelona, Spain), ACM, 277-280.
- [7] Freyne, J., Jacovi, M., Guy, I., and Geyer, W., 2009. Increasing engagement through early recommender intervention. In *Proceedings of the 3rd ACM conference on Recommender systems* (New York, NY, USA), ACM, 85-92.
- [8] Gilbert, E. and Karahalios, K., 2009. Predicting tie strength with social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA), ACM, 211-220.
- [9] Golbeck, J., Parsia, B., and Hendler, J., 2003. Trust Networks on the Semantic Web, 238-249.
- [10] Guy, I., Ronen, I., and Raviv, A., 2011. Personalized activity streams: sifting through the "river of news". In *Proceedings of the 5th ACM conference on Recommender systems* (Chicago, IL, USA), ACM, 181-188.
- [11] Guy, I., Ur, S., Ronen, I., Perer, A., and Jacovi, M., 2011. Do you want to know?: recommending strangers in the enterprise. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work* (Hangzhou, China), ACM, 285-294.
- [12] Hang, C.-W. and Singh, M., 2010. Trust-Based Recommendation Based on Graph Similarity. In *Proceedings of the AAMAS Workshop on Trust in Agent Societies* (Toronto, Canada).
- [13] Herlocker, J.L., Konstan, J.A., and Riedl, J., 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (Philadelphia, Pennsylvania, USA), ACM, 241-250.
- [14] Linden, G., Smith, B., and York, J., 2003. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE* 7, 1, 76-80.
- [15] Massa, P. and Avesani, P., 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems* (Minneapolis, MN, USA), ACM, 17-24.
- [16] Nepal, S., Paris, C., Bista, S.K., and Sherchan, W., 2013. A trust model-based analysis of social networks. *International Journal of Trust Management in Computing and Communications* 1, 1, 3-22.
- [17] Nepal, S., Sherchan, W., and Paris, C., 2012. Building trust communities using social trust. In *Proceedings of the 9th international conference on Advances in User Modeling* (Girona, Spain), Springer-Verlag, 243-255.
- [18] Paek, T., Gamon, M., Counts, S., Chickering, D.M., and Dhesi, A., 2010. Predicting the Importance of Newsfeed Posts and Social Network Friends. In *AAAI 2010*, vol. 10, pp. 1419-1424.
- [19] Pizzato, L.A., Rej, T., Yacef, K., Koprinska, I., and Kay, J., 2011. Finding someone you will like and who won't reject you. In *Proceedings of the 19th international conference on User modeling, adaptation, and personalization* (Girona, Spain), Springer-Verlag, 2021879, 269-280.
- [20] Quercia, D. and Capra, L., 2009. FriendSensing: recommending friends using mobile phones. In *Proceedings of the 3rd ACM conference on Recommender systems* (New York, NY, USA), ACM, 273-276.
- [21] Richards, D., Taylor, M., and Busch, P., 2008. Expertise Recommendation: A Two-Way Knowledge Communication Channel. In *Proceedings of ICAS 2008, the Fourth International Conference on Autonomic and Autonomous Systems*, 35-40.
- [22] Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y., and Merom, R., 2010. Suggesting friends using the implicit social graph. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (Washington, DC, USA), ACM, 233-242.
- [23] Shani, G. and Gunawardana, A., 2009. Evaluating Recommender Systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira and P.B. Kantor Eds. Springer, New York.
- [24] Sherchan, W., Nepal, S., and Paris, C., 2013. A survey of trust in social networks. *ACM Computing Surveys (CSUR)* 45, 4, 47.
- [25] Trifunovic, S., Legendre, F., and Anastasiades, C., 2010. Social Trust in Opportunistic Networks. In *Proceedings of INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, 1-6.
- [26] Wu, A., Dimicco, J.M., and Millen, D.R., 2010. Detecting professional versus personal closeness using an enterprise social network site. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA), ACM, 1753622, 1955-1964.
- [27] Yoshii, K., Goto, M., Komatani, K., Ogata, T., and Okuno, H.G., 2008. An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16, 2, 435-447.
- [28] Ziegler, C.-N. and Golbeck, J., 2007. Investigating interactions of trust and interest similarity. *Decision Support Systems* 43, 2, 460-475.