How to Calculate Trust between Social Network Users?

Vedran Podobnik, Darko Striga, Ana Jandras, Ignac Lovrek
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
{vedran.podobnik, darko.striga, ana.jandras, ignac.lovrek}@fer.hr

Abstract: **Trust among users in a social network plays an important role when trying to build a social recommender system. To be able to calculate *who-trusts-whom* and implement that knowledge in the social recommender, it is crucial to examine in detail all possible relations between users in the social network and to evaluate those relations properly. We propose a model to transform a single user's social graph from a binary structure to a structure with the concrete weights between the nodes in a graph. This paper first addresses a question of how can a general model for calculating trust be implemented using a specific social network as a medium (i.e., Facebook). Afterwards, the proposed model is verified through a Facebook application named *"Closest Friends"*, which calculates the user's closest friends on the Facebook social network. Finally, our approach has been evaluated with the experiment among the 150 Facebook users who tested *"Closest Friends"* application.**

1. INTRODUCTION

Recommender systems are very important part of electronic transactions' lifecycle [1]. Initially, recommender systems only helped users to determine which specific product/service to buy (i.e., which brand, from which provider, etc.) after users have already identified a need for a certain type of product/service. However, nowadays in the era of impulse purchase of low priced, instant access products/services that can be bought through a various online stores [2], recommender systems more and more become triggers which define a user need for a certain type of product/service. This is especially true for social recommenders, a special type of recommender systems which recommend products/services based on activities of user's friends[1] in a social network.

This paper addresses two questions: i) how does a general model for calculating trust in a social network look like; and ii) how can a general model for calculating trust be implemented using a specific social network as a medium (i.e., Facebook)?

This paper is organized as follows. Section 2 will first make an overview of the state-of-the-art research in three areas relevant for the calculation of trust in social networks: i) trust in distributed systems; ii) social recommender systems; and iii) social networks. Afterwards, Section 3 will present the general model for calculating trust on a Facebook social network, while Section 4 will demonstrate the implementation and functionalities of our proof-of-concept application for calculating trust between Facebook users, the *"Closest Friends"* Facebook application. Section 5 gives an evaluation of the proof-of-concept system, while Section 6 concludes the paper and presents the planned future work.

2. RELATED WORK

2.1 Trust in Distributed Environments

Trust is a research field which has recently been attracting scientists from many fields (e.g., computer science, cognitive sciences, sociology, economics, and psychology) [3]. As a result, there exist a number of different definitions of trust. In the context of this paper, we define trust as the *expectancy of a consumer to be able to rely on recommendations of his/her friends within a social network.*

When studying trust in distributed computing environments, one can identify three main tracks of research: i) modelling trust in electronic markets [4]; ii) modelling trust in P2P (peer-to-peer) systems [5]; and iii) modelling trust in social networks [6][7][8]. If one takes a closer look at state-of-the-art in researching trust in social networks, two main questions arise: i) how to identify your best friends within a social network (i.e., who are user's friends that he/she can trust most); and ii) how to calculate trust between social network users who are not directly connected (e.g., how to measure trust between a user and his/her friends-of-a-friends) [9]. In the context of this paper, we will focus on the former problem – how to measure trust between a user and his/her friends.

2.2 Social recommender systems

Recommender systems are traditionally classified into three categories: *item-based*, *collaborative* and *hybrid recommender systems* [10]. Item-based recommender system [11] will recommend the consumer products/services similar to the ones the consumer preferred in the past. Collaborative recommender system [12] will recommend the consumer products/services that people with similar tastes and preferences liked in the past. Hybrid recommender system [13] combines item-based and collaborative approaches.

[1] *Friends* are directly connected users in a social network.

Podobnik, Vedran; Striga, Darko; Jandras, Ana; Lovrek, Ignac.
How to Calculate Trust between Social Network Users?
Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2012). (2012)

Social recommender [14][15] is a new type among recommender systems, whose emergence was enabled by proliferation of web-based social networking. Social recommender system *will recommend the consumer products/services that consumer's best friends (i.e., those friends in the social network whom consumer trusts most) liked in the past*.

2.3 Social networks

Generally, a network can be defined as a set of nodes interconnected via links. Networks are built with a purpose of exchange and they can have various topologies. A *social network* is the specific implementation of a general network concept – the social network can be defined as a *set of actors interconnected via relationships*. Actors could be various – people, organizations, brands, etc. Relationships could be various as well – acquaintance, familiar bond, dislike, etc. However, a notion of common interest is glue that always connects actors involved in a certain social network. Social networks are based on actor profiles, while the creating principle could be twofold – *explicit* or *implicit*. In explicit social networks all connections between actors are direct result of intentional action of those actors, i.e., every social network user must initiate a connection with another user for them two to connect. Each user of a social network, therefore, consciously connects. Social relationships established by popular Social Networking Services (SNSs), such as Facebook, Twitter (*http://twitter.com*) and Foursquare (*http://foursquare.com*), are all based on ego social networks – every user is building his/her own social network by explicitly defining connections with other people. On the other hand, connections between users in an implicit social networks [16][17] are based on a "third party" reasoning over user profiles, resulting in users with similar profiles being (semi-)autonomously connected.

3. GENERAL MODEL FOR CALCULATING TRUST ON SOCIAL NETWORK

When building a social recommender upon an Internet-based platform which implements one of popular SNSs, one needs to focus on how to transform discrete valued on-off connections among social network members (e.g., on the Facebook two users are either friends or not) into continuous valued weighted connections[2] (Fig. 1). This transformation can be done based on: i) similarity of *ego user's*[3] profile with profiles of his/her friends; and/or ii) social activities of ego user and his/her friends in a social network.

[2] *Continuous valued connection weight* reflects the level of a trust between two users in a process of a social recommendation – the higher weight of the connection represents the greater trust between users.

[3] *Ego user* is a specific individual user who is put in focus.

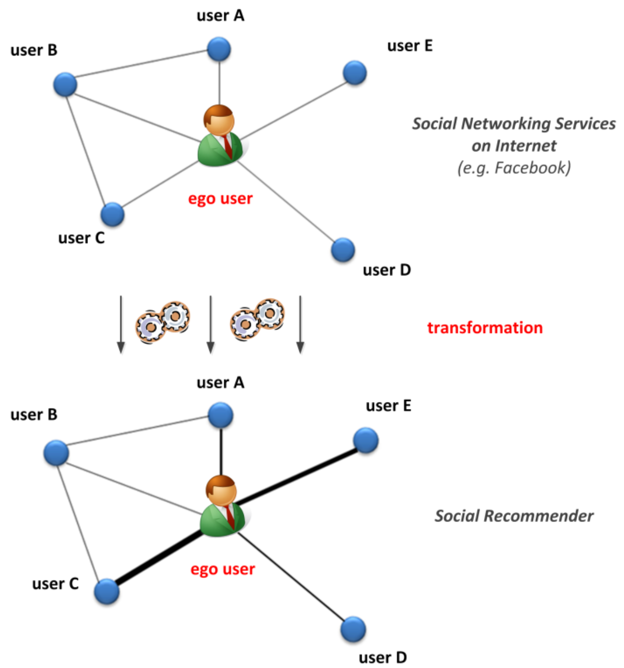

Figure 1. Transformation of discrete into continuous valued weighted connections (*a weight of a connection is labelled with a thickness of an edge connecting two vertices in a social graph*)

In our previous work, in which we modelled a creation of implicit social networks, we focused on the first approach and studied how similarity of user profiles can be used to calculate a "level of connection" between two users in a social network [18][19]. Here we will focus on the second approach and show how social activities of an ego user and his/her friends can be used to calculate a "level of trust" between two friends in a social network.

In the context of this paper, we define a *social activity* as an social interaction between two directly connected social network users (i.e., friends). An example of social activity can be joint tagging of two friends on the same photo. Now, we can define the $\boldsymbol{P}(user_x)$ as a list of friends who are tagged on a same photo with the $user_x$ during a certain period of time. Furthermore, an element in the $\boldsymbol{P}(user_x)$ is a pair:

$$(user_y, number\ of\ interactions) \tag{1}$$

where $user_y$ identifies a social network user which is directly connected to $user_x$ and which had $number\ of\ interactions$ joint photo tags with $user_x$. For example, a list $\boldsymbol{P}(user_x)$ can be defined as:

$$\boldsymbol{P}(user_x) = \{(user_a, 3), (user_b, 2), (user_c, 5)\} \tag{2}$$

what means that $user_x$ is tagged on 3 photos with $user_a$, on 2 photos with $user_b$ and on 5 photos with $user_c$. Moreover, we define the $\boldsymbol{P}(user_x, user_y)$ as a number of joint photo tags of social network friends $user_x$ and $user_y$. In the example presented with Formula 2 the $\boldsymbol{P}(user_x, user_a) = 3$.

Various social networks support different types of social activities amnong their users. To calculate a "level of trust" between two directly connected users in a social network, we have to create as comprehensive description of all social activities between these friends as possible. Therefore, we define descriptions of various social activities of the $user_x$ as the $\boldsymbol{social\ activity\ set}(user_x)$, which is the set of lists where: i) every list describes different social activity; and ii) all lists are defined analogously as the list $\boldsymbol{P}(user_x)$.

To be able to calculate trust from interactions data contained in the $\boldsymbol{social\ activity\ set}(user_x)$, it is necessary to multiply the specific activity data from every list in $\boldsymbol{social\ activity\ set}(user_x)$ with certain weights specific for every interaction type. Formula 3 is used to calculate the *level of trust* (i.e., $trust(ego\ user, friend_x)$) between social network ego user and all of his/her friends (closer friends have the higher level of trust):

$$trust(ego\ user, friend_x) = \\ = \frac{\sum_{N \in \boldsymbol{social\ activity\ set}(ego\ user)} w_N \times \boldsymbol{N}(ego\ user, friend_x)}{\sum_{N \in \boldsymbol{social\ activity\ set}(ego\ user)} w_N}, \quad (3)$$

$$\forall\ friend_x \in \boldsymbol{F}(ego\ user)$$

4. PROOF-OF-CONCEPT: FACEBOOK APPLICATION *"CLOSEST FRIENDS"*

"Closest Friends" is the Facebook application[4] with the functionality of a closest friends calculation for a particular Facebook user (i.e., Facebook ego user). The implemented Facebook application tracks activities (i.e., "liking" and commenting posts, statuses, links and pictures, "writing" on the *Facebook Wall*, etc.) of friends (for every user who installed our application) and calculates a trust level between our application users and all of their friends.

The *"Closest Friends"* Facebook application consists of three modules: i) *Information Sources* module; ii) *Calculation* module; and iii) *User Application* module.

4.1 *"Closest Friends"* Information Sources module

The data collecting process includes collecting personal and behavioural user data from the *Facebook ego user*[5] profile pages. This data includes:

- ego user first name and surname;
- list of ego user's friends;
- list of friends who are tagged on photos with the ego user;
- list of friends who write on the ego user's *Wall*;
- list of friends who leave comments on the ego user's *Wall*;
- list of friends who like posts on the ego user's *Wall*;
- list of friends who write to the ego user's *Facebook inbox*;
- list of friends on whose *Walls* the ego user writes or comments;
- list of friends who like ego user's photos; and
- list of friends who leave comments on ego user's photos.

Namely, in our model we did not focus on ranking ego user's friends but on calculating a set of ego user's 10 closest friends on Facebook (i.e., 10 friends which a particular user trusts the most). To do that, we firstly had to collect and classify all the relevant social activity data from the ego user's Facebook profile (Table 1).

Table 1 – Lists of $user_x$'s Facebook friends classified by different social activities on the Facebook social network

List label	Description of the list	Weight assigned to the list
$\boldsymbol{P}(user_x)$	List of friends who are tagged on a same photo with the Facebook $user_x$	w_P
$\boldsymbol{S}(user_x)$	List of friends who write on the Facebook $user_x$'s *Wall*	w_S
$\boldsymbol{C}(user_x)$	List of friends who leave comments on the Facebook $user_x$'s *Wall*	w_C
$\boldsymbol{L}(user_x)$	List of friends who like posts on the Facebook $user_x$'s *Wall*	w_L
$\boldsymbol{I}(user_x)$	List of friends who write to the Facebook $user_x$'s inbox	w_I
$\boldsymbol{M}(user_x)$	List of friends on whose *Walls* the Facebook $user_x$ writes or comments	w_M
$\boldsymbol{PL}(user_x)$	List of friends who like the Facebook $user_x$'s photos	w_{PL}
$\boldsymbol{PC}(user_x)$	List of friends who leave comments on the Facebook $user_x$'s photos	w_{PC}
$\boldsymbol{F}(user_x)$	List of all friends of the Facebook $user_x$	–

All lists from Table 1 (except list $\boldsymbol{F}(user_x)$) form complete set of social activities between Facebook $user_x$ and all of his/her Facebook friends:

$$\boldsymbol{social\ activity\ set}(user_x) = \\ \{\boldsymbol{P}(user_x), \boldsymbol{S}(user_x), \boldsymbol{C}(user_x), \boldsymbol{L}(user_x), \boldsymbol{I}(user_x), \boldsymbol{M}(user_x), \\ \boldsymbol{PL}(user_x), \boldsymbol{PC}(user_x)\} \quad (4)$$

Every list from the set $\boldsymbol{social\ activities\ set}(user_x)$ contains a different number of pairs defined with Formula 1. It is not possible to predict the number of friends that a certain list will contain, because that number varies

[4] The application can be accessed via link: *http://161.53.19.220/anaapp5*.

[5] Here, *Facebook ego user* is a specific individual Facebook user who installed our *"Closest Friends"* Facebook application.

considerably among different users of a Facebook social network. Namely, every Facebook user has developed his/her pattern of behaviour and interaction on Facebook. Some users use Facebook social network only as a platform to exchange their photos, while other users do not exchange any photos, but rather use Facebook to chat and send private messages via *Facebook inbox* application. There are also users who are extremely passive on Facebook, although they posses a significant number of friends there. Additionally, some users access Facebook only via their smartphones, what allows them a perfect real-time monitoring of *News Feed* Facebook application on their smartphone screen. Those users tend to be more active in terms of commenting and liking their friends' posts and photos.

Consequently, if there are 50 different friends tagged on all ego user's photographs on Facebook, a list $\boldsymbol{P}(ego\ user)$ will consist of 50 pairs defined with Formula 1. Also, if the ego user communicates via *Facebook inbox* with 20 different people, a list $\boldsymbol{I}(ego\ user)$ will consist of 20 pairs. But, if the ego user does not comment on anyone's posts and does not write on anyone's *Facebook Wall*, the list $\boldsymbol{M}(ego\ user)$ will be empty (i.e., $\boldsymbol{M}(ego\ user) = \emptyset$).

4.2 *"Closest Friends"* Calculation module

In the *"Closest Friends"* application, 4 different friend lists are presented to every user, each of them containing a set of the 10 closest Facebook friends for that particular ego user. These 4 friend lists are created by applying the same algorithm defined with Formula 3, but using different weight sets presented in the Table 2.

Table 2 – Weight sets assigned to the trust calculation algorithm implemented in the *"Closest Friends"* application

Weight	Set 1	Set 2	Set 3	Set 4
w_P	6	*random()*	2	5
w_S	4	*random()*	4	2
w_C	3	*random()*	3	2
w_L	2	*random()*	2	1
w_I	1	*random()*	5	1
w_M	5	*random()*	5	1
w_{PL}	1	*random()*	1	1
w_{PC}	1	*random()*	1	1

4.3 *"Closest Friends"* User Application module

The *"Closest Friends"* application has the following life-cycle:

1. Ask the ego user to gain access to his/her private data;
2. Gather data and create all individual lists presented in Table 1 (this procedure can take up to few minutes due to limitations of the Facebook API);
3. Prompt the ego user with the first question in a questionnaire, asking him/her to name the 10 friends he/she trusts most (i.e., the closest friends) on the Facebook social network;
4. Prompt the ego user with the second question in a questionnaire, asking him/her to decide which of the presented 4 friend lists does he/she find the most accurate for himself/herself (i.e., which list does he/she think has calculated his/her closest friends the best);
5. Present to the ego user all the calculated lists of his/her friends that he/she may find interesting (e.g., lists of top-ranked friends by interaction based on *Facebook inbox*, lists of friends who are most often tagged on photos with the ego user, etc.).

5. EVALUATION OF THE PROPOSED MODEL FOR CALCULATING TRUST ON FACEBOOK

For the evaluation of the proposed model, we divided a total of 150 users of the *"Closest Friends"* application in three groups, based on the level of permissions they granted to the Facebook application. It is possible to grant:

- *'user_photos'* permission (which allows the application to access all the ego user's photos on Facebook);
- *'read_stream'* permission (which allows the application to access all data on the ego user's Wall); and
- *'read_mailbox'* permission (which allows the application to access the ego user's inbox).

Each of the permissions can be granted individually. Distinct user groups are presented in the Table 3.

Table 3 – Different groups of *"Closest Friends"* application users based on the granted permissions' level

	'user_photos'	*'read_stream'*	*'read_mailbox'*	User count
Users A	YES	YES	YES	92
Users B	YES	YES	NO	22
Users C	YES	NO	NO	36

We conducted two different types of the evaluation of the proposed trust calculation model:

- *subjective evaluation* – evaluation of the ego user's choice for the best algorithm weight set; and
- *objective evaluation* – mathematical calculation of the precision measure for each algorithm weight set based on widely used metrics for evaluation of recommender systems.

The ego user's choice for the best algorithm weight set (Figure 2) determined the $Algorithm\ weight\ set\ 3$ as the subjectively most accurate (i.e., 43% of users have chosen it as their best choice), while the $Algorithm\ weight\ set$ 1 was only 2% behind (i.e., 41% of users have chosen it as their best choice). The result of only 2% of users who have chosen the $Algorithm\ weight\ set$ 2 (i.e., the "random algorithm") have shown that our proposed model calculates list of closest Facebook friends correctly, while the supremacy of $Algorithm\ weight\ set$ 3 and $Algorithm\ weight\ set$ 1 over $Algorithm\ weight\ set$ 4 shows the high importance of assigning a high weight factor to the $\boldsymbol{M}(ego\ user)$ list (i.e., the list of friends on whose *Walls* the Facebook ego user writes or comments).

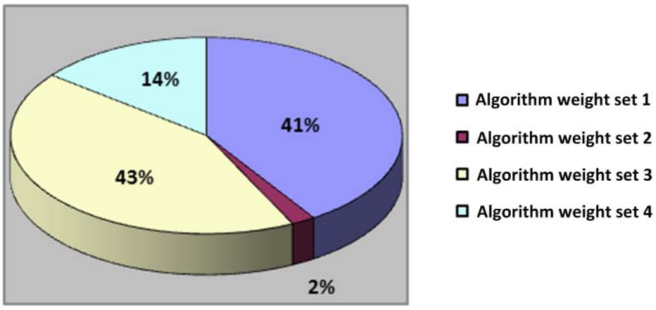

Figure 2. Subjective evaluation: the ego user's choice for the best algorithm weight set

Objective evaluation (Figure 3) based on algorithm precision calculation confirmed conclusions from the subjective evaluation – again the $Algorithm\ weight\ set$ 3 yielded the highest precision, while the $Algorithm\ weight\ set$ 1 was near behind.

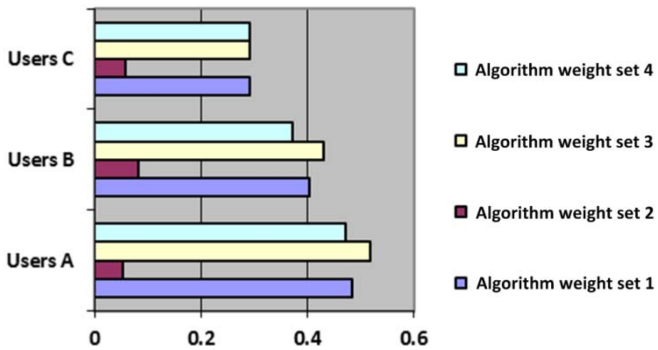

Figure 3. Objective evaluation: the precision of the different algorithm weight sets for the $Users\ A$, $Users\ B$ and $Users\ C$

Here we also analysed how permissions granted by Facebook users influence the algorithm precision. As expected, through comparing the different users groups, it can be noticed that the highest precision occurs within $Users\ A$, who gave the highest set of permissions to the *"Closest Friends"* application. Indifferent to the algorithm weight set used, $Users\ A$ always had mathematically highest precision among all user groups. Analogously, precision for the $Users\ C$ is significantly lower then the precision of any other users' group, indifferent to the algorithm weight set used. The reason for that is an insufficient number of permissions which $Users\ C$ granted to the application.

6. CONCLUSION AND FUTURE WORK

Social recommender systems calculate level of trust among friends in a social network by combining know-how from i) modelling trust systems in distributed environments; ii) building recommender systems; and iii) analysing web-based social networks. This paper presented a general model for the calculation of trust among social network users, as well as described how a general model can be implemented using a specific social network as a medium (i.e., Facebook). Finally, the proposed model is verified through the *"Closest Friends"* Facebook application for finding the user's closest friends on Facebook.

The level of user's satisfaction with a set of identified top-friends within a Facebook social network is evaluated on 150 Facebook users and the conclusion is following: comparing the results, we conclude that the algorithm should assign a high weight factor to the ego user's actions (e.g., writing or commenting) on his/her friends' *Facebook Walls*. Using our algorithm with such weights resulted in the highest precision for every user group (i.e., regardless of the level of permissions granted to our Facebook application). Moreover, algorithm with a high weight factor assigned to the ego user's actions on his/her friends' *Facebook Walls* also got the best feedback from users.

Although ideal algorithm precision result would be around 0.9, the current situation is that the highest precision of our best-performed algorithm weight set is 0.52. Reasons for the decreased precision could be interpreted by the certain specific situations in which users of the *"Closest Friends"* application have found themselves while filling out the application's questionnaire:

- Some users did not understand the first question in the questionnaire correctly – it was the question where they had to name their 10 closest friends of Facebook. Instead of entering the 10 friends with whom they communicate and interact the most on Facebook, some users entered their 10 closest friends from their private life, or even their relatives, with whom they do not communicate on Facebook at all or only sporadically.
- Some users "sabotaged" the work of the *"Closest Friends"* application, by purposely entering the wrong data in the questionnaire. Namely, there were 3 users detected who chose the "random algorithm" to be the best one, even though this algorithm weight set could certainly not be the best one for any user. It is very likely that exactly those 3 users have also entered the completely wrong set of their 10 personally closest friends, so the mathematical precision for those 3 users is

extremely low as well. Unfortunately, precision results for the specific algorithm weight sets are calculated on an average of all 150 users of the *"Closest Friends"* application, which means that some occasionally extreme low precision can significantly lower the average precision results for the each algorithm weight set.

- Some users have an extremely passive usage pattern on Facebook social network, which means that they have lots of friends, but very few social interactions with them. For those users, the calculated mathematical precision will be very low, which will in the end lower the precision results for the each algorithm weight set.

In the future, we plan to upgrade our system with the user feedback option, which will enable personalization and fine-tuning of weight parameters used for calculation of closest Facebook friends, as well as implement a proof-of-concept applications for other major social networking services.

7. ACKNOWLEDGMENTS

The authors acknowledge the support of research project "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia. The authors wish to iSTUDIO emloyees for their valuable feedback on our design and for their assistance with the development of the *"Closest Friends"* application. Special thanks go to Tomislav Grubisic and Daniel Ackermann.

REFERENCES

[1] Guttman, R.H., Moukas, A. G., & Maes, P. Agent-Mediated Electronic Commerce: A Survey. Knowledge Engineering Review 13 (2), 1998, pp. 147-159.

[2] Podobnik, V. & Lovrek, I. Transforming Social Networking from a Service to a Platform: a Case Study of Ad-hoc Social Networking. Proceedings of the 13th International Conference on Electronic Commerce (ICEC'11). Liverpool, UK: ACM, 2011.

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

[4] Kim, D.J., Ferrin, D.L., & Rao, H.R. A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents. Decision Support Systems 44 (2), 2008, pp. 544-564.

[5] Singh A. & Liu, L. TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems. In Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P '03). Washington, USA: IEEE Computer Society, 2003. 142-149.

[6] Golbeck, J. Trust and nuanced profile similarity in online social networks. ACM Transactions on the Web 3 (4), 2009, art.no. 12.

[7] Golbeck, J. Computing with Social Trust. Springer Publishing Company, 2010.

[8] Matsuo, Y. & Yamamoto, H. Community gravity: measuring bidirectional effects by trust and rating on online social networks. In Proceedings of the 18th International Conference on World Wide Web (WWW '09). New York, USA: ACM, 2009. 751-760.

[9] Dubois, T., Golbeck, J., & Srinivasan, A. Predicting Trust and Distrust in Social Networks. In Proceedings of the 3rd IEEE International Conference on Social Computing, Boston, Massachusetts, 2011.

[10] Adomavicius, G. & Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17 (6), 2005, pp. 734-749.

[11] Sambolec, I., Rukavina, I., & Podobnik, V. RecoMMobile: A spatiotemporal recommender system for mobile users. Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2011). Split, Croatia: IEEE, 2011.

[12] Aggarwal, C.C., Wolf, J.L., Wu, K-L., & Yu, P.S. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.

[13] Balabanovic, M. & Shoham, Y. Fab: Content-Based, Collaborative Recommendation. Communication of the ACM 40 (3), 1997, pp. 66-72.

[14] Arazy, O., Kumar, N., & Shapira, B. Improving social recommender systems. IT Professional 11 (4), 2009, pp. 38-44

[15] Victor, P., De Cock, M., & Cornelis, C. Trust and Recommendations. In Recommender Systems Handbook, P.B. Kantor, et al. (eds.), Springer, New York, 2010, pp. 645-676.

[16] Yang, Q., Zhou, Z.-H., Mao, W., Li, W., & Liu, N.N. Social Learning. IEEE Intelligent Systems 25 (4), 2010, pp. 9-11.

[17] Podobnik, V., & Lovrek, I. Telco Agent: Enabler of Paradigm Shift towards Customer-Managed Relationship. Lecture Notes in Computer Science 6276, 2010, pp. 251-260.

[18] Podobnik, V., & Lovrek, I. An Agent-based Platform for Ad-hoc Social Networking. Lecture Notes in Computer Science 6682, 2011, pp. 74-83.

[19] Podobnik, V., & Lovrek, I. Telco Agent: Enabler of Paradigm Shift towards Customer-Managed Relationship. Lecture Notes in Computer Science 6276, 2010, pp. 251-260.