ICIC Express Letters
Part B: Applications
Volume **7**, Number **2**, February **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 407–413

FINDING USERS ATTRIBUTES FROM USER COMMON INTEREST ON SOCIAL NETWORKS

Rung-Ching Chen* and Hendry

Department of Information Management
Chaoyang University of Technology
No. 168, Jifong East Road, Wufong Dist., Taichung County 41349, Taiwan
*Corresponding author: crching@cyut.edu.tw; hendry.honk@gmail.com

Received July 2015; accepted September 2015

Abstract. *With the recent growth of activities on social media like Facebook, Twitter, or Youtube, discovering social common interests becomes more important due to its usefulness to help people connect each other with the same interests, and encourage people to contribute and share more contents. This paper proposes similarity calculation to discover social common interest, especially on Facebook. With similarity calculation, the similarities between users based on their favorite music, movies, and books could be found. Each user will generate a perfect tree which is the base for similarities calculation. Each node of perfect tree has a weight. Its value is determined by the numbers of the users. Music is significant attributes from user's common interest because the similarity value between users is the best.*
Keywords: Social common interest, Weighted tree similarity, Similarity measure

1. **Introduction.** In the present modern life, social media has become a trend in the society. Almost everything has to do with social media. Social media is an Internet-based application built on Web2.0 technology, so it could encourage users to participate, share and create contents on the website [1]. Web2.0 changes the paradigm in the Internet. Social media is one of the examples of Web2.0 and it consists of user interest fields [3-7], such as music [8], video [9,10], product and user's interests [11]. Circle of social media also has a factor and attributes about what information and recommendation that user will choose, because this factor influences the user similarity preference based on their friend's interest [2].

In a social media site like Facebook, each user is able to provide personal information such as name, date of birth, address, hobby, favorite music, film, and books to others. Users might have the same or similar interest on a certain subject; this is known as social common interests. On the other hand, a different condition occurs when users do not know each other and do not join a certain group but have similar interest. In this situation, the search of social common interest becomes important. The finding of social common interest in a social network helps to connect users with similar interest to find out what current trending topic is in the social media, and even recommend content that fits to the user's interest [12]. For business owners, social common interest is useful to identify the type of product or service to be introduced to the market.

Boley et al. proposed Tag-based Social Interest Discovery [13], to explain the discovery of social common interest by developing a system named Internet Social Interest Discovery System or ISID. ISID enables user to tag interesting materials especially in social media del.icio.us. The system uses these tags to group users according to the specific interest. Qian et al. proposed an inference influences between users to look for the common interest between users on yelp database [2]. All the previous research looks for the common interests between users according to the tag to the trending topic. Our research tries to look for the factor that affects the common interest between the users than calculate its similarity value.

2. **Background.** In this section, we will briefly explain about the algorithm and technology that had been used to discover social common interests. There are FacebookAPI, and weighted tree similarity. FacebookAPI is used to collect Facebook user's personal information. Weighted tree similarity algorithm is used to calculate the similarity value between users.

2.1. **FacebookAPI.** FacebookAPI is a platform for building applications. They are available to the social network of Facebook [14]. With FacebookAPI, third party developers could build their own applications and services to access Facebook's data. In Figure 1, we could find how FacebookAPI works. First, whenever you request FacebookAPI and get the needed data from Facebook, the system will request those data through FacebookAPI. FacebookAPI then will pass the request to Facebook server. If the request is valid and matched with application's current permission, Facebook will give the right response to client through FacebookAPI. There are several products of FacebookAPI. In this paper, we used Graph API for PHP and combined with Facebook Query Language (FQL).

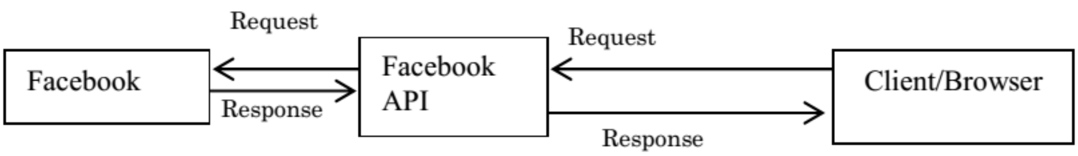

Figure 1. Process on FacebookAPI

2.2. **Weighted tree similarity.** In this paper, we have two types of weighted tree, one is perfect tree and the other is individual tree. Perfect tree is a tree which represents all interests from all users that have been in the system. On the other hand, individual tree is a tree which represents personal interests from each user. Both trees are assumed that they have normalized form. The weight for every node is up to 1. According to [15], a tree is defined as follows.

Define a single tree has label node. A tree $T = (V, E, L)$ is a simple single tree consisting of the minimum V, E and L, where E connects V in directed graph as follows.

1. V is nodes elements.
2. E is edge that connects to V, where E is a directed graph from V to L.
3. Every L could contain node V or Value, and each of L could be connected by E.
4. L is a label, and this L could be concepts V or a Value (LV).

In similarity tree measure, there are some issues to tackle regarding the general shape of the tree, their recursive nature, and their arbitrary size. Weighted tree similarity is an algorithm which could solve all the issues, so it will be used to compute the similarity between two trees [16]. In this algorithm, there are 3 functions, treesim, treemap and treeplicity. Its main function is treesim that will call the "workhorse" treemap which co-recursively calls treesim and treeplicity.

Treesim[N,A](t, t').

Treesim is a function recursively to compare the two sub-trees T and T'. They return the value 1.0 if the nodes are identical, otherwise the return value will be 0. If one of the nodes compared to a node does not have value, this situation is considered as non equal value that will return 0 and checked function will work if the shape is the same between the trees. This step will be conducted for every leaves with the same level, and will sum up to the root with the same shape and root.

The similarity of two sub-trees, T and T', including leaves with no equal node label is defined by 0.0 and it will be considered vice versa. If the root, leaves and node label are equal, it will be defined by 1.0. In this paper, we assumed the value of N is consistently to be 1 and 0.

Treemap[N,A](l, l').

Treemap is the function to recursively calculate the weight of the two trees node l and l'. Every weight in each level of the sub-trees will be calculated as weighted arcs W_i and W_i', and would have the value between 0 and 1. The weight is calculated using arithmetic mean, and it will be divided with the number of the nodes that consists in the tree for each level. The process will be summed up of all the value in the same level, to be used in the upper level calculation. This formula could be described as follows.

$$Sim = \frac{\sum_{i=1}^{n} F(n) * (W_i * LVi)/N}{\sum_{i=1}^{n} (W_i * LVi)} \tag{1}$$

In this paper, every node has weights up to 1, so Formula (1) could be simplified into Formula (2).

$$Sim = \frac{1}{N} \sum_{i=1}^{n} F(n) * (W_i * LVi) \tag{2}$$

where N is total node of the tree; LVi is node that contains the value; W_i is arc weight of the i^{th} arc below the root node of tree T. In addition, the function of treeplicity is shown in Figure 2.

Input: The depth degradation index i. A single tree T.
Output: The simplicity value of T.
Initialization: $treeplideg = 0.5$
treeplicity (i, T)
Begin
    If T only contains a single node **return** i;
    endif
    else
        sum = 0;
        for (j *is every node in* T);
            sum+=(W_i)****treeplicity*** (i**treeplideg*, T_j);
        endfor
        Sim = (1/total node of tree T)* sum;
        return Sim;
    endelse
End.

FIGURE 2. Weighted simplicity tree algorithm

The treeplideg value is less than 0.5 to make sure that summed up value is 1 because we already normalized the weight and the node value between $[0, 1]$. This will keep the value always smaller than 1. For every leaf nodes in the tree T, the current i value is the simplicity of the tree. The simplicity function is represented as a recursive formula in

Formula (3).

$$S(T) = \begin{cases} D_I.(D_F)^d & \text{If } T \text{ is leaf node,} \\ \frac{1}{N}\sum_{i=1}^{n} S(T') * (W_i * LVi) & \text{else} \end{cases} \quad (3)$$

where

$S(T)$: the simplicity value of a single tree T

D_I and D_F: depth degradation index and depth degradation factor

d: depth of a leaf node

N: total node of the tree

LVi: node that contains the value

W_i: arc weight of the i^{th} arc below the root node of tree T

T': sub-trees of the tree T

3. **Implementation.** In this section, the process of the discovery of common interest, such as how to build a similarity tree, calculation of similarity value and the result of similarity search is presented and discussed.

3.1. **Tree representation.** To begin with, the finding of social common interests is started with developing a tree that has data on favorite music, films and books from a number of Facebook's users. A tree has four levels resembling a person as a root. A root has two children. They are interest and name. Interest node has three children. They are music, movie, and book. Music has genre, artist, and record label as sub-child. Movie has movie genre, director, and movie cast, and last a book has author and title. Beneath each of the sub-child, there are other nodes, such as music genre having genre pop, rock, and jazz. The shape of a tree is presented in Figure 3.

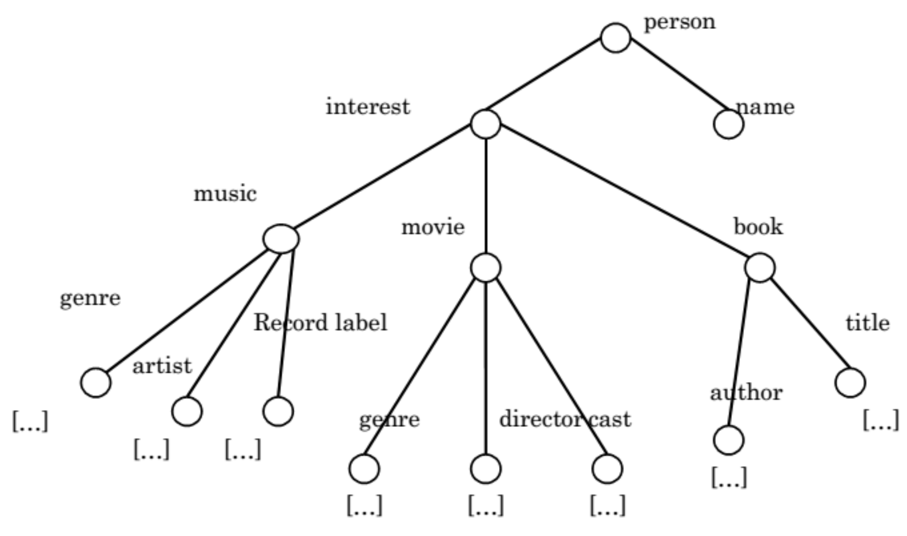

FIGURE 3. Representation of tree

3.2. **Weight calculation on tree.** Besides having its own tree, each user contributes in giving a weight to the perfect tree. In this research, a perfect tree is a tree which represents all interests from all users which have been introduced into the system. Initially, the weight of a perfect tree is assumed at one point which is $w = 1/n$. A new user enters the system, and the weight on the perfect tree changes. The weight changes follow the rule of Popularity (POP), where a node on a perfect tree is scored 1 (one) if there is a similarity to the user and 0 (zero) in the absence of similarity [19]. The scoring formula is presented in Formula (4).

$$W_i = \Sigma\, rating(U[n])/n \quad (4)$$

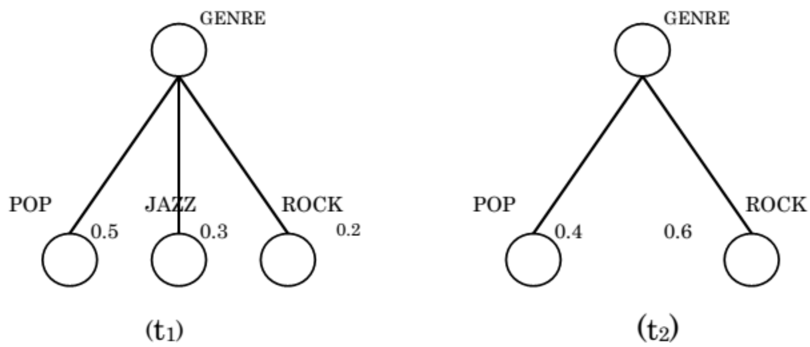

Figure 4. Example of weighted tree similarity

The calculation for similarity between buyer through weighted tree similarity algorithm is explained previously. An example of similarity tree calculation is presented in Figure 4.

If tree t_1 is a user's tree and t_2 is another user's tree, tree t_1 has three children which are Pop, Jazz, and Rock and each weights 0.5, 0.3, and 0.2 respectively. In the meantime, tree t_2 only has two children, Pop and Rock, weight is 0.4 and 0.6. Trees t_1 and t_2 have the same root, the genre, while t_2 does not have sub-tree jazz. This condition initiates treeplicity function when node jazz is added with 0 (zero) weight and the value of $F(n)$ for node jazz is 0.5. Hence, the similarity value for node jazz is $0.5 * (0.3 + 0)/2 = 0.075$. With the same condition for the other 2 nodes, the $F(n)$ value for both nodes is one. The similarity value for node pop is $1 * (0.5 + 0.6)/2 = 0.55$, and for node rock is $1 * (0.2 + 0.4)/2 = 0.3$. That makes the similarity value between the two agents be $Sim(t_1, t_2) = 0.075 + 0.55 + 0.3 = 0.925$. The similarity calculation process is repeated continuously between trees. All similarity results between trees are stored in the database.

4. **Results.** This part will discuss on the results of similarity search towards 30 samples of Facebook users. The all thirty samples were chosen randomly. The results of the trials are divided by the location of similarity, and it can be seen in Table 1 and Table 2.

Table 1. Results from users similarity on movie

User 1	**User 2**	**Similarity value**	**User 1**	**User 2**	**Similarity value**
R Brian Hendrasta	Benedictus Ridho	0.5169	Inna Septiana	Yulita Ambarsari	0.7593
Franklien Phoanda	Benedictus Ridho	0.4173	Hafidz Habibie	Yoga Sanjaya	0.5115
Putri Kurniati	Yehuda Aribowo	0.4482	Magentha Dea	Shandra Setya	0.3084
Putri Kurniati	Yulita Ambarsari	0.4842	Rifki Muhammad	Maulida Retna	0.4506
Yehuda Aribowo	Yulita Ambarsari	0.5376	Nia Erista	Handri Huang	0.3336
Dwinda Sekar	Inna Septiana	0.6417	Maulida Retna	Inna Septiana	0.5883
Inna Septiana	Yoga Sanjaya	0.6213			

Table 2. Results from users similarity on music

User 1	User 2	Similarity value	User 1	User 2	Similarity value
R Brian Hendrasta	Franklien Phoanda	0.393	Shabrina Mei	Chorintan Prabelia	0.5274
Vera Nica	Chorintan Prabelia	0.4128	Chorintan Prabelia	Yulita Ambarsari	0.528
Vera Nica	Fadhil Pasau	0.3495	Stevano Andreas	Chorintan Prabelia	0.3855
Dwinda Sekar	Fadhil Pasau	0.4992	Stevano Andreas	Yulita Ambarsari	0.612
Yehuda Aribowo	Yulita Ambarsari	0.5376	Magentha Dea	Rifki Muhammad	0.3561
Norman Wisnu	Yoga Sanjaya	0.4344	Rifki Muhammad	Shandra Setya	0.4058
Norman Wisnu	Chorintan Prabelia	0.4068	Nia Erista	Zara Bunga	0.2421
Antonius Jerry	Chorintan Prabelia	0.4467	Maulida Retna	Eunike Gloria	0.3726
Antonius Jerry	Stevano Andreas	0.3372	Eunike Gloria	Nalurita Absari	0.3864
Hafidz Habibie	Hanung Nugroho	0.4803	Widya Katika	Katarina Thatya	0.615
Shabrina Mei	Yulita Ambarsari	0.5916	Katarina Thatya	Eunike Gloria	0.4509

Table 1 is a set of users where the most similarity located on their favorite movies. There are 13 combinations from 20 users. The highest similarity value is 0.7593, and the lowest one is 0.3084.

Table 2 is a set of users where the most similarity located on their favorite is music. There are 22 combinations from 26 users and the highest similarity value is 0.615 and the lowest one is 0.2421.

Based on Table 1 and Table 2, there are 30 samples with total 34 combinations between them. After the trial was done, it can be seen that music is the most significant indicator in the match making process. It is proof with 22 user combinations that have the most similarity value on music. On the counterpart, book is the most irrelevant indicator of match making. There are no samples of users that have the book for the most similarity part.

5. **Conclusions and Future Work.** Based on the research, it can be concluded as follows. (1) A system is proposed to read Facebook data through FacebookAPI and built the perfect tree of interest. This tree will be compared into user tree to calculate the similarity as an indicator of social common interest. (2) We found that music is the best significant value for discovering social common interest better than other factors. (3) If two users have a high similarity, and the second user also has a similarity with the third user, the first user could have a high similarity with the third user as well. In the future, we will develop this application to let social common interests could be found not only in Facebook, but also from other social media at the same time. In other side, we will prove the effectiveness of the proposed method in discovering social common interests. In addition, the system could also give recommendations about something that some users might be interested in.

Acknowledgment. This work is supported by Ministry of Science and Technology, Taiwan, with number: MOS 103-2221-E-324-02 and MOS 103-2632-E-324-001-MY3.

REFERENCES

[1] A. M. Kaplan and M. Haenlein, Users of the world, unite! The challenges and opportunities of social media, *Business Horizons*, vol.53, no.1, pp.59-68, 2010.

[2] X. Qian, H. Feng, G. Zhao and T. Mei, Personalized recommendation combining user interest and social circle, *IEEE Trans. Knowledge and Data Engineering*, vol.26, no.7, pp.1763-1777, 2014.

[3] M. Ou et al., Comparing apples to oranges: A scalable solution with heterogeneous hasing, *Proc. of the 19th ACM SIGKDD*, Chicago, IL, USA, pp.230-238, 2013.

[4] P. Cui, F. Wang, S. Liu, M. Ou and S. Yang, Who should share what? Item-level social influence prediction for users and posts ranking, *Proc. of Int. ACM SIGIR Conf.*, Beijing, China, 2011.

[5] P. Cui, F. Wang and S. Yang, Item-level social influence prediction with probabilistic hybrid factor matrix factorization, *Proc. of the 25th AAAI Conf. Artificial Intelligence*, 2011.

[6] M. Jiang et al., Social recommendation across multiple relational domains, *Proc. of the 21st ACM Int. CIKM*, Maui, HI, USA, 2012.

[7] H. Feng and X. Qian, Recommendation via user's personality and social contextual, *Proc. of the 22nd ACM CIKM*, New York, NY, USA, 2013.

[8] J. Shen, H. Pang, M. Wang and S. Yan, Modeling concept dynamics for large scale music search, *Proc. of the 35th Int. ACM SIGIR Conf. Research Development Information Retrieval*, New York, NY, USA, pp.455-464, 2012.

[9] J. Shen and Z. Cheng, Personalized video similarity measure, *Multimedia System*, vol.17, no.5, pp.421-433, 2011.

[10] J. Shen, D. Tao and X. Li, Modality mixture projections for semantic video event detection, *IEEE Trans. Circuits System Video Technology*, vol.18, no.11, pp.1587-1596, 2008.

[11] X. Qian, X. Liu, C. Zheng, Y. Du and X. Hou, Tagging photos using users vocabularies, *Neurocomputing*, vol.111, pp.144-153, 2013.

[12] X. Li, L. Guo and Y. E. Zhao, Tag-based social interest discovery, *Proc. of the 17th International Conference on World Wide Web ACM*, pp.675-684, 2008.

[13] H. Boley, V. C. Bhavsar, D. Hirtle, A. Singh, Z. Sun and L. Yang, Agent matcher search in weighted, tree structured learning object metadata, *Learning Objects Summit*, pp.29-30, 2004.

[14] *Facebook API Documentations*, https://developers.facebook.com/docs/, 2014.

[15] L. Yang and H. Boley, A weighted-tree simplicity algorithm for similarity matching of partial product descriptions, *Proc. of the 14th International Conference on Intelligent and Adaptive Systems and Software Engineering*, Toronto, 2005.

[16] W. W. Cohen and W. Fan, Web-collaborative filtering: Recommending music by crawling the web, *Computer Networks*, vol.33, no.1, pp.685-698, 2000.