Local Scalar Trust Metrics with a Fuzzy Adjustment Method

Yangjin Seo[1] and Sangyong Han[2]
[1] School of Computer Science and Engineering, ChungAng University,Seoul, Korea
[e-mail: yjseo@ec.cse.cau.ac.kr]
[2] School of Computer Science and Engineering, ChungAng University, Seoul, Korea
[e-mail: hansy@cau.ac.kr]
*Corresponding author: Sangyong Han

Received December 20, 2009; revised February 26, 2010; accepted March 7, 2010; published April 29, 2010

Abstract

The interactions between people who do not know each other have been greatly increased with the on-going increase of people's cyberspace activities. In this situation, there exist potential risk factors such as the possibility of fraud, so we need a method to reduce or eliminate those risk factors. Concerning this necessity, rating systems are widely used, and many trust metrics calculated from rate values that people give to each other are proposed to help them make decisions. However, the trust metrics decrease the accuracy, and this is caused by the different rating scales and ranges of each person. So, we propose a fuzzy adjustment method to solve this problem. It is possible to catch the exact meaning of the trust value that each person selects through applying fuzzy sets, which improve the accuracy of the trust metric calculated from the trust values. We have applied our fuzzy adjustment method to the TidalTrust algorithm, a representative algorithm for calculating the local scalar trust metric, and we performed an experimental evaluation with four data sets and three evaluation methods.

Keywords: Trust metric, fuzzy adjustment method, user interaction, decision making

This Research was supported by the Chung-Ang University Research Grants in 2010.

DOI: 10.3837/tiis.2010.04.004

1. Introduction

Interactions that are mediated by the Internet are immensely increasing due to the on-going development of the Internet. Those interactions often take place between people who do not know each other. People tend to be uncomfortable while interacting with strangers and they are interested in finding ways to relieve their discomfort. Therefore, rating systems are widely used in which users choose rate values within a given range to express their opinion of a specific topic. The range given to users can be either continuous or discrete, and a discrete range is more commonly used. Value scales have to be defined for a discrete range, and the simplest one is using binary values and values from two to ten are generally used.

However, the actual value range that each user chooses can vary. For instance, let us suppose that each user shows the reliability of their neighbors by selecting a number from one to ten from a predefined range of 1 to 10 (number ten means fully trusted). **Fig. 1** demonstrates this with a weighted directed graph; the circle and arrow express the person and the direction of trust, respectively. Here, user B and user C equally give trust rates to their three neighbors. User C uses a wide range (rate 3, 6, and 9), while user B uses a narrow range (rate 7, 8, and 9). The rate value that is selected by each user is utterly subjective, so no one can raise an objection about that. However, it becomes a completely different issue when there is a specific method based on these values. For instance, user A needs a control mechanism when he allows user F or user H access to his personal data. If user A adopts an access control method based on trust values, he would consult user B and user C because he does not directly know user F and user H. User A equally gives a trust value of eight to both user B and user C. Therefore, the trust value from user A to user F is seven and the trust value from user A to user H is six. Accordingly, user A concludes that user F is more trusted than user H.

However, this is not fair because it seems that seven means low trust from the viewpoint of user B and six means average trust from the viewpoint of user C.

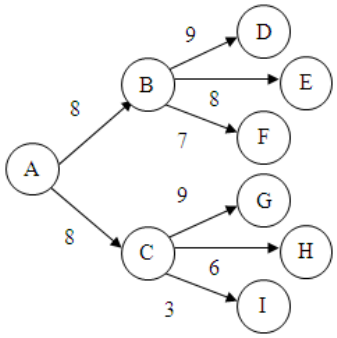

Fig. 1. A different trust value range among users

Because a single trust value can have a different meaning to different persons as we described above, a way to correctly interpret it is needed.

To solve this problem, we propose an adjustment method based on fuzzy sets. Particularly, we will apply our adjustment method to the local scalar trust metrics to improve its precision. It becomes possible to interpret the exact meaning of the values that each person gives to his

neighbors through using the fuzzy membership functions. Based on this interpretation, calculating the trust metrics with adjusted values becomes more accurate.

This paper is constructed as follows. Section 2 introduces the trust metrics and explains the reason why local scalar trust metrics need an adjustment. In section 3, we describe how a fuzzy membership function can be effectively used for an adjustment and we propose a FuzzyTidalTrust algorithm that applies our adjustment method to the TidalTrust, a local scalar trust metric algorithm. In section 4, we show the experimental results that compare the FuzzyTidalTrust to the TidalTrust with four data sets and three evaluation methods. Finally, section 5 concludes the proposals made in this paper and suggests some topics for future studies.

2. Defining Trust Metrics

Trust is used as an important guideline when people make decisions [1]. The trust of sellers or products is used as an important basis in buyer's selections [2][3][4][5][6], and it is used as a criterion to select a data source in P2P file sharing systems [7][8][9] or recommendation systems [10][11]. Meanwhile, as trust is gaining attention as being a useful tool, various methods have emerged for calculating trust metrics. Ziegler [12] classified various trust metrics according to some attributes as in **Fig. 2**. 'Distributed' and 'Centralized' refer to the place where the trust metrics are evaluated and quantified. The global trust metric is calculated from the whole graph information, while the local trust metric is quantified with partial graph information.

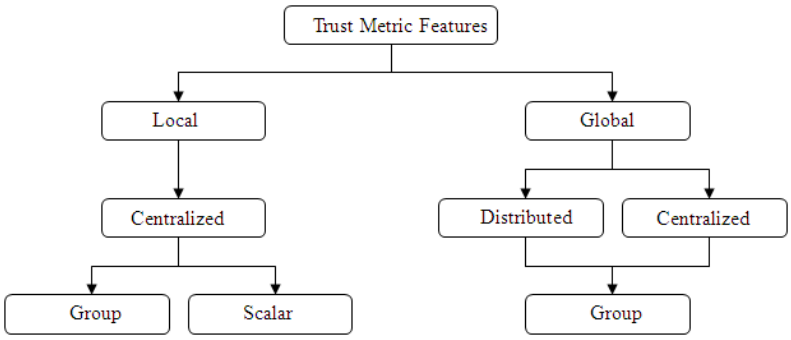

Fig. 2. Trust Metric Classification.

Finally, 'group' and 'scalar' are features of the calculated values. In the case of the scalar trust metric, a single value is produced along the path from source to target, while a trust value is given as a relative rank among persons included in the calculation in the case of the group trust metric. PageRank [13] and Ziegler's Appleseed are representative algorithms that give group trust metrics. These algorithms include the normalization process. Equation (1) is a basic equation of the PageRank algorithm. The reputation value of P_i, denoted $r(P_i)$, is the sum of the reputations of all P_j having an outlink to P_i. $|P_j|$ is the number of outlinks from P_j.

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \tag{1}$$

In other words, each page equally distributes its own reputation to its neighbors; actually, PageRank defines a damping factor, so each page holds some portion of its reputation and

hands out the rest to its neighbors. For instance, suppose that the links between pages are given as in **Fig. 3**. Page A equally gives its reputation to Page B and Page D, i.e., 1/2.

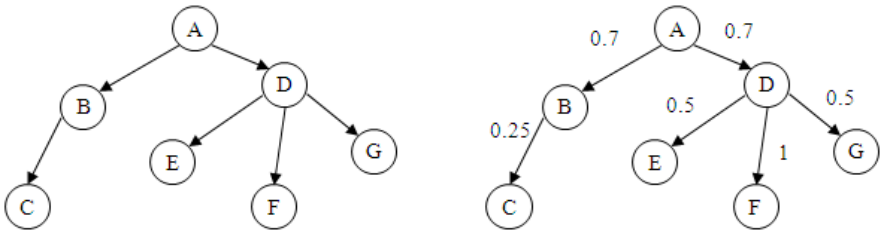

Fig. 3. A graph without weight values and a graph with weight values.

The PageRank algorithm assumes that every link has the same weight value, but the normalization process can be applied to the case where each link has a different weight. Appleseed, a representative local group metric algorithm, defines equation (2) for a graph as shown on the right-hand side of **Fig. 3**. The weight value has a real number between 0 and 1. A bigger number means a higher trust value. In addition, Appleseed defines a spreading factor which carries out the same function as that of the damping of PageRank. 'x' user distributes his reputation to y as much as the weight given to 'y' that's normalized by the sum of weights of all the outgoing edges of 'x'. For example, 'D' distribute r(D) to 'F' as much as '1/(0.5+1+0.5)'. Appleseed is similar the PageRank, except that each individual distributes his own reputation to the adjacent neighbors proportional to the weight value.

$$r(x) \cdot \frac{W(x,y)}{\sum_{(x,s)\in E} W(x,s)} \tag{2}$$

This way, the calculation of group trust metrics lets each person distribute his reputation to his neighbors through normalization. Therefore, if a certain person adds an outlink, the portion given to his existing neighbors is proportionally reduced, as is shown on the right side of **Fig. 3**. If 'D' adds 'H' as his neighbor and D gives H a weight value of 1, then the portion distributed to 'F' falls from '1/(0.5+1+0.5)' to '1/(0.5+1+0.5+1)'.

On the contrary, the calculation of scalar trust metrics does not include the process of normalization, so the trust value that each person gives to his neighbors can be infinitely used during the calculation process. Suppose that we calculate an indirect trust value with using TidalTrust [14], which is a representative algorithm to evaluate local scalar trust metric. (3) is the equation of the TidalTrust. t_{ik} is the trust value from person 'i' to person 'k', and adj(i) is a set of the neighbors of person 'i'. Each person gives a trust value to his adjacent neighbor. A trust value is a number from one to ten (number ten means fully trusted). 'max' is established during the process of the calculation, and only the opinions of the neighbors that are rated greater than max are included in the calculation according to the condition '$t_{ij} \geq$ max'. How to find the 'max' is explained minutely in [14].

$$t_{ik} = \frac{\sum_{j \in adj(i) \mid t_{ij} \geq max} t_{ij} t_{jk}}{\sum_{j \in adj(i) \mid t_{ij} \geq max} t_{ij}} \tag{3}$$

In **Fig. 4**, although person 'A' adds hundreds of new neighbors, the trust value used to calculate the trust of person 'C' is not changed according to the definition of equation (3). However, such calculation may produce incorrect results because it does not reflect the exact meaning of the number that each person gives to his neighbor. For instance, let us directly and indirectly calculate the trust metric of one neighbor: the direct trust from person A to person B is seven, while the indirect trust is five when we calculate the trust towards person C. If we analyze the results with only numbers, then two results seem to be different. However, trust value seven that person A gives to his neighbor might be equal to the trust value five that person C gives to his neighbors when analyzing all numbers that person A and person C give to their neighbors. We propose an adjustment method based on fuzzy sets to solve this problem. By using fuzzy sets, the exact meaning of numbers that each person gives to his neighbors can be captured. As a result, the precision of the local scalar trust metric is improved.

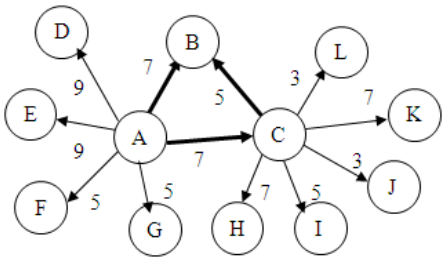

Fig. 4. A graph weighted with the trust label.

3. Adjusting Trust Values Using Fuzzy Sets

In the real world, it is not an easy problem to quantify a person's opinion or evaluation of a specific subject, and the same is true when quantifying the trust of neighbors; if there are only two words for representing trust, i.e., 'reliable' and 'unreliable', then this will be a simple problem. But in practice, 'very reliable' or 'little reliable' are general statements. Therefore, rating systems enable people to select a value from a given range in order to represent their opinions of specific subjects, and this approach makes quantification possible. However, as described above, that is not a simple problem because the same trust number can have different meanings depending on the person.

For instance, **Fig. 5** depicts a trust graph, and the trust values given are built into a matrix as in **Fig. 6**; an item (i, j) is the trust value from person i to person j. We get indirect trust values for this trust graph by applying the TidalTrust algorithm and the results of that are in the matrix in **Fig. 7** (we applied one as a max value of equation (3) and the underline indicates an item of the direct trust value). We can find interesting items in these results. For example, person 14 has seven neighbors and the minimum and maximum trust values he gave to his neighbors are 2 and 7, respectively. However, there are indirect trust values from person 14 that are out of the range from 2 to 7 (the indirect trust values to person 1 and person 18 are 10). For the person 14, the trust value 2 means 'least reliable' and the trust value 7 means 'most reliable'. Therefore, the trust values out of the range from 2 to 7 are values that person 14 cannot figure out. In this way, the fact that a single trust value has different meanings for different persons is the cause of the reduced precision of the trust metrics that will perform a calculation with using this number. Thus, a method to relieve this matter is needed.

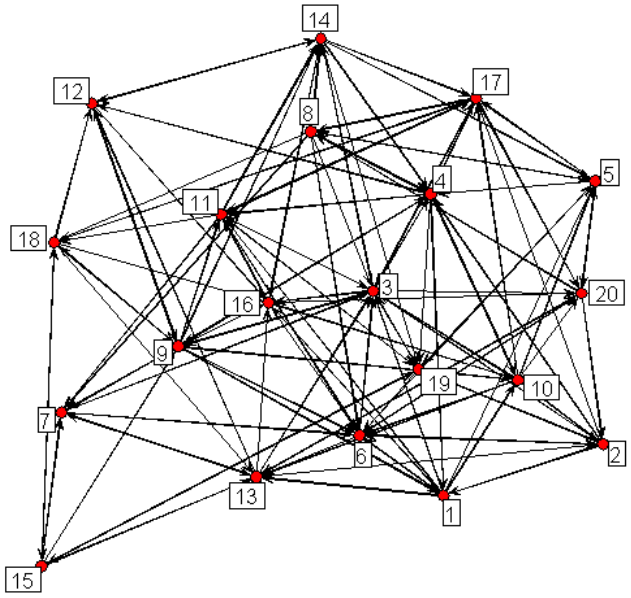

Fig. 5. A trust graph composed of twenty members.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	9	0	7	3	0	0	0	6	6	0	5	0	0	8	0	0	0	0
2	3	0	0	9	0	2	0	0	0	6	0	0	0	0	0	0	6	0	4	0
3	0	3	0	6	0	0	0	0	4	0	4	0	0	3	0	3	10	0	9	4
4	10	0	0	0	0	0	0	2	3	5	5	2	0	0	0	0	0	0	8	0
5	0	0	0	2	0	0	0	7	0	0	0	0	0	0	0	0	3	0	7	5
6	0	1	6	0	7	0	0	0	0	0	9	0	7	0	0	0	0	0	0	4
7	0	0	7	0	0	6	0	0	0	0	0	0	3	0	0	0	0	0	0	0
8	0	0	2	2	7	6	4	0	0	0	0	0	0	8	0	0	0	10	0	0
9	3	0	0	6	0	1	9	0	0	1	3	0	0	2	0	10	0	4	0	0
10	0	0	7	0	4	3	0	0	0	0	0	0	8	0	0	0	7	0	0	5
11	0	0	1	0	0	0	9	0	0	0	0	0	0	9	0	0	6	8	3	0
12	0	0	0	0	0	0	0	0	10	0	0	0	0	4	0	4	0	0	0	0
13	0	3	10	0	0	0	7	0	0	0	0	3	0	0	0	4	0	0	0	9
14	0	0	0	5	2	0	0	6	0	0	0	4	0	0	0	2	3	0	7	0
15	0	0	0	0	0	0	5	0	1	0	0	0	5	0	0	0	0	8	3	0
16	0	0	0	0	0	1	0	0	0	6	9	0	0	0	0	0	0	0	0	1
17	0	0	9	10	0	0	0	8	0	0	1	0	0	0	0	0	0	0	0	6
18	0	0	0	0	0	0	0	0	6	0	0	4	7	0	6	3	6	0	0	0
19	0	0	8	0	9	7	0	0	0	5	10	0	0	0	1	0	0	0	0	0
20	0	6	0	0	10	0	0	3	0	0	0	0	10	0	0	0	0	0	0	0

Fig. 6. A trust values' matrix for the trust graph of **Fig. 5**.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	2.6	9	4.3	7	3	8.1	7	4	6	6	3	5	5.4	1.6	8	6.8	8	6.7	4.4
2	3	0	8	9	6.3	2	7.8	4.4	3	6	5.3	2	7	5.7	1	8	6	7.8	4	5.3
3	6.2	3	0	6	7.9	4	9	5.3	4	4.6	4	2.7	10	3	1	3	10	6	9	4
4	10	2.6	6.6	0	7	4.1	8	2	3	5	5	2	6	6.3	1	7.9	6.5	7.2	8	5
5	10	6	5.7	2	0	6.5	4	7	3	5	6.9	2	10	8	1	4.7	3	10	7	5
6	3	1	6	4.2	7	0	8.1	5.5	4	6	9	3	7	6.6	1.8	3.5	6.1	8	5.8	4
7	6	2.3	7	6	7	6	0	5.3	4	4.7	6.3	3	3	3	1	3.3	10	6.6	9	4.9
8	10	1.5	2	2	7	6	4	0	5.3	5	7.2	3.8	6.2	8	6	2.6	4.6	10	7.3	4.5
9	3	1	6.2	6	5.1	1	9	3	0	1	3	3	4.7	2	6	10	5.5	4	6.5	1.6
10	6.5	3.4	7	6.7	4	3	7	6.2	4	0	3.6	3	8	3	1	3.5	7	7.7	8.3	5
11	7.2	3	1	6.9	3.8	6.3	9	6.8	5.8	5	0	4	4.9	9	4.6	2.5	6	8	3	5.7
12	3	1.4	6	5.7	2	1	9	6	10	2.4	4.7	0	5	4	4.6	4	3	4	7	1
13	3	3	10	6.7	10	3.7	7	3	5.4	6	5.4	3	0	3.2	1	4	9.1	6	7.8	9
14	10	3.5	5.9	5	2	5.8	4	6	6.1	5.1	6.8	4	6.8	0	1	2	3	10	7	4.3
15	3	3	8.4	6	9	5.8	5	4	1	4	8.3	3.6	5	2	0	3.9	6	8	3	9
16	3	3.5	3.6	5.7	5.1	1	9	3	4.4	6	9	3.3	8.1	9	4.6	0	6.4	8	3	1
17	10	4.2	9	10	8.3	6	4.6	8	3.5	5	1	2	10	5.6	1.9	3	0	9.8	8.2	6
18	3	3	9.5	8	6.2	1	7	8	6	2.7	3.4	4	7	2.8	6	3	6	0	3	6.4
19	6.5	2.1	8	3.9	9	7	8.6	7	3.7	5	10	2.8	7.2	6.3	1	3	6.3	8	0	4.5
20	3	6	8.2	4.2	10	3.3	6.3	3	3.7	6	5.8	3	10	8	2.1	4	4.1	10	5.9	0

Fig. 7. An indirect trust values' matrix after applying TidalTrust algorithm.

Fuzzy sets [15] is a useful tool to solve those problems. Fuzzy sets and fuzzy logic are used in various areas [16][17][18], which include the problems of imprecision, uncertainty and partial truth. We used the fuzzy sets for interpreting the meaning of trust values that each person gives to his neighbors and solve the problem described above. First, we defined three membership functions for trust values: mf_{Low}, mf_{Medium}, and mf_{High}. When the minimum, maximum and average of one person are t_{min}, t_{max}, and t_{avg}, respectively, mf_{Low}, mf_{Medium}, and mf_{High} are defined as in equation (4), (5), and (6), respectively.

$$mf_{Low}(t; t_{min}, t_{avg}) = \begin{cases} 1, t \leq t_{min} \\ \frac{t_{avg}-t}{t_{avg}-t_{min}}, t_{min} \leq t \leq t_{avg} \\ 0, t_{avg} \leq t \end{cases} \quad (4)$$

$$mf_{Medium}(t; t_{min}, t_{avg}, t_{max}) = \begin{cases} 0, t \leq t_{min} \\ \frac{t-t_{min}}{t_{avg}-t_{min}}, t_{min} \leq t \leq t_{avg} \\ \frac{t_{max}-t}{t_{max}-t_{avg}}, t_{avg} \leq t \leq t_{max} \\ 0, t_{max} \leq t \end{cases} \quad (5)$$

$$mf_{High}(t; t_{avg}, t_{max}) = \begin{cases} 0, t \leq t_{avg} \\ \frac{t-t_{avg}}{t_{max}-t_{avg}}, t_{avg} \leq t \leq t_{max} \\ 1, t_{max} \leq t \end{cases} \quad (6)$$

For instance, the minimum, average and maximum of all trust values of person 1 are 3, 6.29,

and 9, respectively; three membership functions for person 1 are given in **Fig. 8**.

On the contrary, three membership functions for person 5 are as in **Fig. 9**; the minimum, average and maximum of all trust values are 2, 4.8, and 7, respectively.

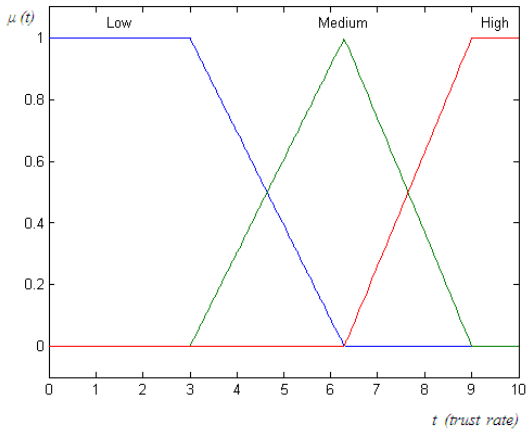

Fig. 8. Three membership functions for person 1.

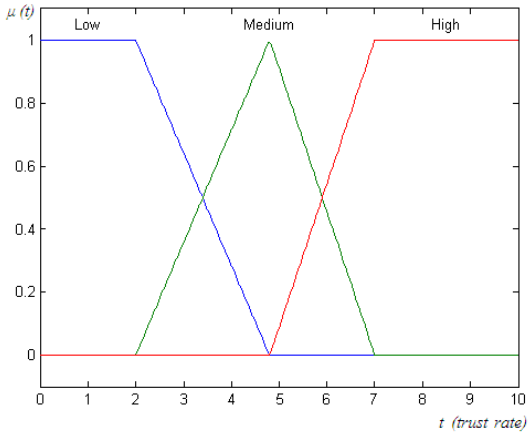

Fig. 9. Three membership functions for person 5.

As shown in **Fig. 8** and **Fig. 9**, the same trust value can have different meanings for different persons, so the calculated value of the trust metric will be imprecise if the raw trust values are used in the calculation. We devised a value adjustment method that uses the fuzzy sets to solve this matter, and we applied it to the TidalTrust algorithm as below:

[Algorithm 1] FuzzyTidalTrust

```
calculateFuzzyTidalTrust( person_i, person_j )
{
  /* Get trust values that person_i gives to his neighbors */
  tempTrust := getTrustRate( person_i );

  /* Define three membership functions using values in tempTrust */
  ( MF_Low, MF_Medium, MF_High ) := buidMF( tempTrust )
```

```
/*  Get the membership functions of all persons except for person_i, and adjust raw trust
    values using the membership functions of person_i */
for( tempPerson in arrayAllPersons )
{
    tempTrust  := getTrustRate( tempPerson );
    ( tempMF_Low, tempMF_Medium, tempMF_High ) := buidMF( tempTrust )
    /* Adjust trust values of each user according to the view of person_i */
    newTrust := adjustTrustRate( MF_Low, MF_Medium, MF_High,
                                 tempMF_Low, tempMF_Medium, tempMF_High )
    /* Apply the adjusted values */
    setTrustRate( tempPerson, newTrust )
}
  /* Calculate the metric of TidalTrust with the adjusted values */
Return calculateTidalTrust( person_i, person_j )
}
```

Suppose that we calculate the indirect trust value from person 1 to person 4 using FuzzyTidalTrust. In the process of FuzzyTidalTrust, the trust value from person 5 to person 4 is adjusted as shown in **Fig. 10**. For person 5, trust value '2', the trust value from person 5 to person 4, is a member of the fuzzy set 'Low' and its membership value is '1', and trust value '3' is a member of fuzzy set 'Low' and its membership value is '1' for person 1. Therefore, the trust value from person 5 to person 4 is adjusted from '2' to '3' for calculating the indirect trust value from person 1 to person 4.

But, the original trust value from person 5 to person 4 is restored after finishing this calculation.

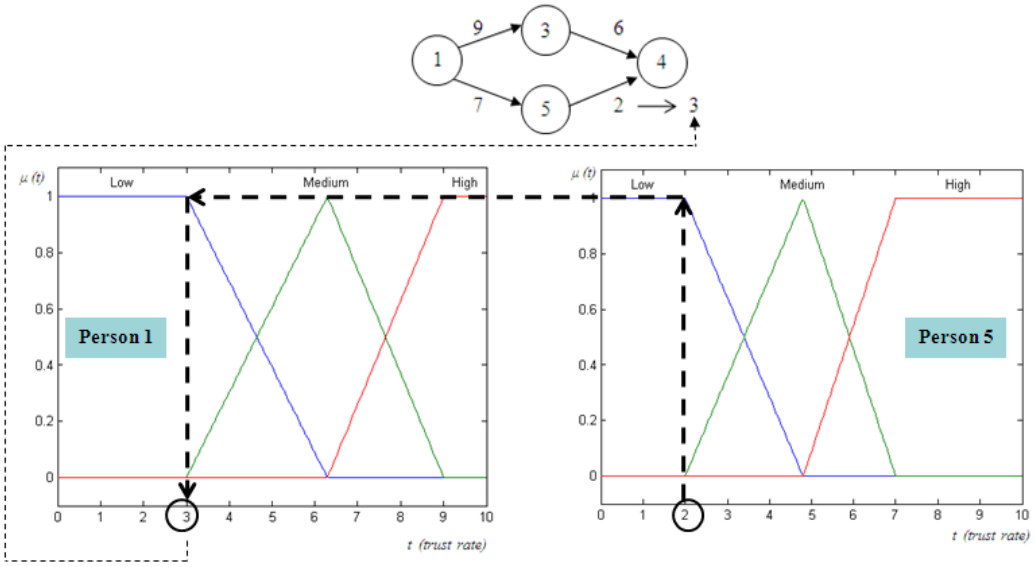

Fig. 10. The trust value adjustment for calculating the indirect trust value from person 1 to person 4.

Fig. 11 shows the results of applying FuzzyTidalTrust to a graph of **Fig. 5**, in a matrix. The results of **Fig. 11** have an average difference of 0.66 compared with the results of applying TidalTrust (**Fig. 7**); the minimum difference is 0.1 and the maximum is 3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x	3	9	5.1	7	3	8.1	9	4.6	6	6	3	5	5.4	3.5	8	7	8.1	7.5	5.3
2	3	x	7.4	9	5.4	2	7.4	3.8	3	6	5.2	2	7.2	5.1	2	7.5	6	7.5	4	4.3
3	6.8	3	x	6	7.9	4.2	9.6	5.4	4	4.9	4	3.7	10	3	3	3	10	6.6	9	4
4	10	2.1	7	x	6.4	3.2	8.5	2	3	5	5	2	5.9	6.5	2	7.7	6.4	7.6	8	4.3
5	7	4	4.3	2	x	5	3.6	7	2.9	4.2	5.4	2	7	6	2	3.5	3	7	7	5
6	2.6	1	6	3.8	7	x	7.9	6.1	3.2	6.5	9	1	7	5.8	1.9	1.8	5.7	7.9	6.2	4
7	5.2	3	7	5.6	6	6	x	4.6	4.1	4.9	5.4	3	3	3	3	3.2	7	5.9	6.7	4.7
8	10	2	2	2	7	6	4	x	6.2	5.6	7.9	4.1	7.6	8	7.3	2	5.4	10	9.7	5
9	3	1	8	6	4	1	9	2.8	x	1	3	2	4.1	2	6.6	10	5.2	4	6.7	1.4
10	6	3.4	7	6.1	4	3	6.3	5.8	4.3	x	4.4	3	8	3	3	3.5	7	7	7.7	5
11	7.4	1	1	7.7	2.8	7	9	7.6	6.8	4.5	x	4.5	4.8	9	5.5	1	6	8	3	5
12	5.2	4.1	7.7	7.2	4	4	9.3	8.6	10	5	6.6	x	7	4	6.6	4	4.9	5.8	10	4
13	4	3	10	7.5	10	5.3	7	3	5.7	7.3	6	3	x	3	3	4	9.3	7.2	8.2	9
14	7	2.6	4.2	5	2	4.1	3.2	6	4.6	4	5.3	4	5.4	x	2	2	3	7	7	3.5
15	3	1	7.5	5.5	6.9	4.9	5	2.8	1	2.8	6.8	1.9	5	2	x	1.9	5.8	8	3	7.1
16	2.1	2.1	3.5	5.2	3.6	1	9	1	3.3	6	9	1.7	8.6	9	4.9	x	5.3	7.4	2.3	1
17	10	2.6	9	10	8.8	7.1	4.9	8	3.3	6.8	1	1	10	4.9	2.3	1	x	9.9	8.7	6
18	4.4	3	6.8	6.4	5.1	3	6	6	6	4	4.4	4	7	3.4	6	3	6	x	4.4	5.3
19	7.6	1	8	4	9	7	9.7	10	3.4	5	10	1.8	8.7	6	1	1	6.8	9	x	5.2
20	4.4	6	8.4	5.2	10	4.5	7.3	3	5.1	7.9	7.2	3	10	8.8	3.9	4.4	5.8	10	8.4	x

Fig. 11. An indirect trust values' matrix of applying the FuzzyTidalTrust algorithm.

4. Experimental Results and Analysis

TidalTrust and FuzzyTidalTrust produce different results from the same input data, so we need a performance evaluation scheme. Intuitively, we can say that FuzzyTidalTrust more accurately reflects each person's intent than TidalTrust in that indirect trust values from FuzzyTidalTrust are in the scope of trust values that each person gives to his neighbors. However, we need to prove that in an objective way.

ASSUMPTION 1. Persons who are trusted highly will tend to agree with the user more about the trustworthiness of others than persons who are less trusted.

The assumption 1 means that both a trustor and his highly rated trustee will give the same trust value to their common neighbors with high probability. In principle, the local scalar trust metrics are based on assumption 1. However, the real world does not comply with that because each person can have a different opinion about a particular person. Moreover, a trust value can be interpreted as having a different meaning depending on the person who uses that in rating systems supporting multi-scale.

To capture this aspect, we have compared the performance of FuzzyTidalTrust to TidalTrust with four trust value sets that have different features and different three evaluation methods. Four trust value sets are defined as below.

- **Set 1**: The trust values that are randomly generated are given to each user.
- **Set 2**: A trustor and his trustees give the same trust values to their common neighbors. For instance, both person A and person C give the same trust value of seven to person B, who is a common neighbor of both person A and person C, as is shown in **Fig. 12**.

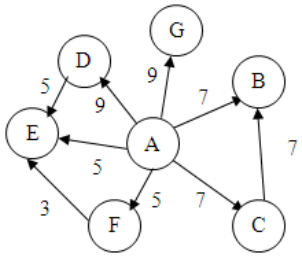

Fig. 12. A graph with trust values.

- **Set 3**: The trust value that a trustee gives to common neighbors is proportional to the trust value that the trustor gives to the trustee. For instance, person D, whom person A gives the trust value of 9, gives the trust value of 5 to person E who is a common neighbor of both person A and person D, while person F, whom person A gives the trust value of 5, gives the trust value of 3 to person E who is a common neighbor of both person A and person F. The algorithm for generating data set 3 is as below:

[Algorithm 2]

```
generateDataset3()
{
    commonNeighbor := getCommonNeighbor( );

    while( commonNeighbor != NULL ) {
      /* Get a trust level from trustor to trustee */
      fuzzyLevel = getFuzzyLevel( trustor, trustee );
      switch( fuzzyLevel ) {
      case 'Low':
        /* Get a random number in the range from 0 to 7 */
        difference  := getRandomNumber( ) % 8;
        break;
      case 'Medium':
        /* Get a random number in the range from 0 to 3 */
        difference  := getRandomNumber( ) % 4;
        break;
      case 'High':
        /* Get a random number in the range from 0 to 1 */
        difference  := getRandomNumber( ) % 2;
        break;
      }

      Operation := getRandomNumber( ) % 2;
      if( Operation == 0 ) {
        trustValue := trustValueArray[trustor][commonNeighbor] – difference;
        if( trustValue < 1 )
            trustValue := 1;
      }
      else {
        trustValue := trustValueArray[trustor][commonNeighbor] + difference;
```

```
            if( trustValue > 10 )
               trustValue := 10;
         }
         trustValueArray[trustee][commonNeighbor] = trustValue;

         commonNeighbor := getNextCommonNeighbor( );
      }
   }
```

Set 1, 2, and 3 are artificially generated data, so we have added 'set 4' for a more objective evaluation.

- **Set 4**: The trust values extracted from Enron email corpus[21] [22] are used; Enron email corpus contains the mail directories of 150 users. The extraction process is as follows:
  1) There is no explicit trust rating in Enron email corpus, so we consider email interactions among persons a metric of trust.
  2) Each person interacted with some other person. All persons whom one person interacted with are ranked according to the interaction count, and then they are divided into ten groups.
  3) Trust value '10' is given to a group with the highest rank, and trust value '1' is given to a group with the least rank.

The 'leave-one-out cross validation' is commonly used for evaluating local scalar trust metrics. In the leave-one-out cross validation, the original value is compared with the predicted value that is calculated excluding the original value. In this context, the indirect trust value is compared with the direct trust value. We have used three evaluation methods as follows:

- **Evaluation method 1**: We calculated the MAE (Mean Absolute Error), one of the measures of using the leave-one-out cross validation. The MAE is give by equation (7); f_i is the prediction and y_i is the true value (the original trust value given by each person).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \quad (7)$$

  For instance, the trust value that person A gives to person B is 7, as is shown in **Fig. 12**, and the indirect trust value from person A to person B is 6, assuming that the edge from person A to person B does not exist. Therefore, e_B is 1.
- **Evaluation method 2**: We evaluated precision using the trust levels that are defined by the membership functions of the fuzzy sets. If the true trust level equals the predicted trust level, then we counted this as a correct instance.
- **Evaluation method 3**: We defined this as a correct instance where the true level equals all trust levels that neighbors whom trustor gives 'High' level give to the same person. For instance, in **Fig. 13**, if the predicted trust level from person A to person Z is the 'Medium' and person D and person G give the 'Medium' level to person Z, then this is a correct instance.

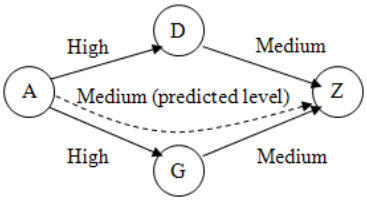

Fig. 13. A graph with trust levels.

Although evaluation method 3 is not commonly used, it is an effective way to evaluate local scalar trust metrics in that the problem of evaluating various metrics is how they can correctly calculate indirect trust values. In short, in rating the systems that support multi-scale, if every person uses trust values with the same meaning and assumption 1 is fully satisfied, then evaluation method 1 and 2 are enough to evaluate the performance of various local scalar trust metrics. Otherwise, evaluation method 3 will be a suitable choice.

Two trust graphs are used in our experiments (graph 1 and 2). 'Graph 1' is a randomly generated graph having 200 members, and 'graph 2' is a graph extracted from Enron email corpus; their densities are approximately 0.2 and 0.1, respectively. We have applied data set 1, 2, and 3 and three evaluation methods to these two graphs. Data set 4 can be applied only to the 'graph 2'.

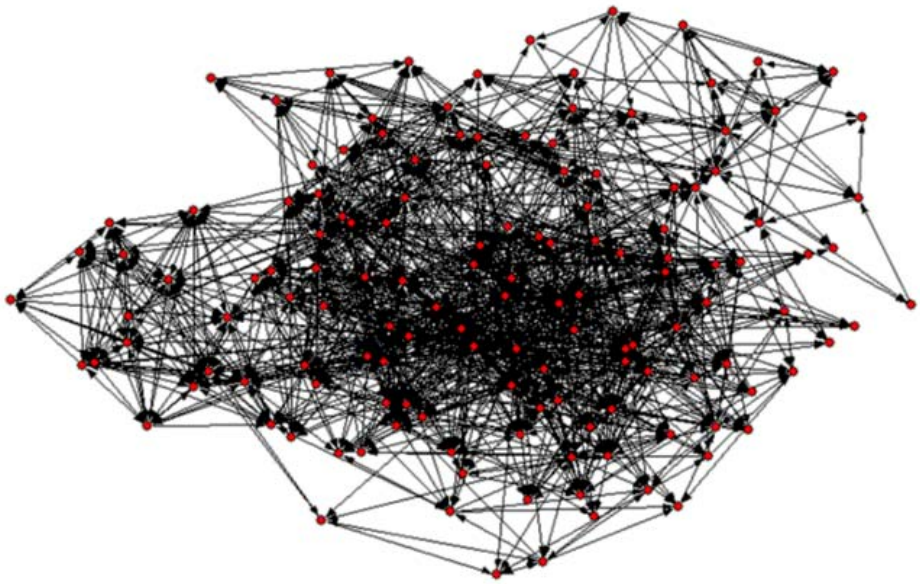

Fig. 14. Trust graph extracted from Enron email corpus (graph 2)

Table 1 shows the results of applying data set 1, 2, and 3 and three evaluation methods to the 'graph 1' and **Table 2** shows the results of applying four data sets and three evaluation methods to the 'graph 2'. The results of evaluation methods 2 and 3 are the percentages that represent precision values. There are some observations on these results. First, the results show much difference according to the feature of data sets. As we mentioned earlier, local scalar trust metrics are based on the assumption 1. However, the results from data set 2 are better than the results from data set 3 for all evaluation methods; data set 3 is in high

compliance with assumption 1, and data set 2 does not comply with assumption 1 because each trustee gives the same trust value to common neighbors all the time.

Moreover, the results from data set 4, a real-world data set, in **Table 2** indicate that the performance of local scalar trust metrics is clearly reduced if the assumption is not satisfied. Second, the results with the 'graph 1' have the same patterns as the results with the 'graph 2'. These results say that graph topology is not a critical factor in evaluating the performance of local scalar trust metrics although it can play a role in emphasizing the results. Third, the results from TidalTrust are better than the results from FuzzyTidalTrust in the case of applying evaluation method 1 and evaluation method 2 to data set 2. These are natural outcomes because data set 2 assumes that a trustor and his trustees give the same trust values to their common neighbors and FuzzyTidalTrust breaks this assumption. One interesting thing is that the results of applying evaluation method 3 are reversed. This is also natural because a trustor consults only neighbors whom he gives 'High' level. Finally, FuzzyTidalTrust has an improved precision when evaluation method 3 is applied. The precision is improved by 2% for all data sets for the 'graph 1', whereas it varies from 9% to 15% for the 'graph 2'. We guess that these remarkable differences between two results arise from the density difference between two graphs. We will do a further research on applying evaluation method 3 to various graphs and data sets.

Table 1. Experimental results with the 'graph 1'

Data set	Evaluation method 1		Evaluation method 2		Evaluation method 3	
	Tidal	FuzzyTidal	Tidal	FuzzyTidal	Tidal	FuzzyTidal
Set 1	2.80	2.76	40	40	83	85
Set 2	1.58	1.62	63	62	91	93
Set 3	2.26	2.24	49	50	80	82

Table 2. Experimental results with the 'graph 2'

Data set	Evaluation method 1		Evaluation method 2		Evaluation method 3	
	Tidal	FuzzyTidal	Tidal	FuzzyTidal	Tidal	FuzzyTidal
Set 1	2.77	2.71	41	41	60	75
Set 2	0.55	0.83	88	79	81	93
Set 3	1.99	1.93	55	56	73	80
Set 4	3.12	2.96	33	34	67	76

5. Conclusions and Future Works

With the increased use of the Internet, the interactions between people who do not know each other have also been exponentially increased. Particularly, there will be tremendous expansion of social relations, beyond our expectations, along with the development of the next generation web represented by online social network services. The uncertainty and risk of the interactions between people who do not know each other are becoming barriers to this expansion.

Besides, for search engines and recommender systems, the core players in a world of information overload, a method that let people evaluate the reliability of others effectively is an essential tool. Reliability has different meanings depending on the context in which it is used, and the metrics used for evaluating it vary. If the factor of personal preference is excluded, then group trust metrics, producing rank among objects, will be the most suitable tool because they hold objectivity.

However, we have to use the local scalar trust metrics to reflect personal preference. In short,

we can use group trust metrics to find out which movie everyone likes, while local scalar trust metrics are suitable tools in the case of finding out which movie a particular person likes. Local scalar trust metrics are founded on the assumption that persons who are trusted highly will tend to agree with the user more about the trustworthiness of others than persons who are less trusted.

However, a trust value that is selected by each user is utterly subjective and can be differently interpreted depending on the person who uses that trust value in rating multi-scale systems, so we need a method to calculate local scalar trust metrics that reflect these points. Accordingly, we designed a fuzzy adjustment method that can capture the real meaning of the trust value that each person uses, and we applied it to calculating the local scalar trust metrics. Through the experiments with three data sets and three evaluation methods, we showed that our FuzzyTidalTrust outperforms TidalTrust, a representative algorithm calculating local scalar trust metric. Some researchers claim that only local trust metrics are true trust metrics [19].

However, in practice, local trust metrics are not widely used, while group trust metrics are used in various fields. One reason for this situation might be the absence of a suitable evaluation scheme for local scalar trust metrics. We have proposed a new evaluation method using fuzzy sets against this problem. As written in [20], evaluating the performance of local scalar metrics is still a challenging problem. In future works, we will find a way to evaluate our FuzzyTidalTrust in a more objective way. For example, applying our algorithm to a recommender system based on online social networks will be a good testbed.

References

[1] A. Josang, R. Ismail, C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support System*, pp.618-644, 2007.

[2] P. Resnick and R. Zeckhauser, "Trust among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System", *Technical report: University of Michigan*, 2001.

[3] S. Grabner-Kraeuter, "The Role of Consumers' Trust in Online-Shopping," *Journal of Business Ethics*, vo.39, no.1-2, pp.43-50, 2002.

[4] S. Grabner-Kräuter and E. A. Kaluscha, "Empirical research in on-line trust: a Review and Critical Assessment," *International Journal of Human-Computer Studies*, vol.58, no.6, pp. 783-812, 2003.

[5] P. A. Pavlou, "Consumer Acceptance of Electronic Commerce: Integrating Trust and Risk with the Technology Acceptance Model," *International Journal of Electronic Commerce*, vol.7, no.3, pp.101-134, 2003.

[6] K. K. Bharadwaj and M. Y. H. Al-Shamri, "Fuzzy Computational Models for Trust and Reputation Systems," *Electronic Commerce Research and Applications*, vol.8, no.1, pp.37-47, 2009.

[7] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-based Trust for Peer-to-peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.7, pp. 843-857, 2004.

[8] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," in *Proc. of 12th Int Conf. on World Wide Web*, pp. 640-651, 2003.

[9] Z. Liang and W. Shi, "PET: A PErsonalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing," in *Proc of. 38th Int. Annual Hawaii Conf. on System Sciences (HICSS'05),* vol.7, pp.201b, 2005.

[10] P. Avesani, P. Mass, and R. Tiella, "A Trust-enhanced Recommender System Application: Moleskiing," in *Proc of. 2005 ACM symposium on Applied computing*, pp.1589-1593, 2005.

[11] F. E. Walter, S. Battiston, and F. Schweitzer, "A Model of a Trust-based Recommendation System on a Social Network," *Autonomous Agents and Multi-Agent Systems*, vol.16, no.1, pp. 57-74, 2008.

[12] C. Ziegler and G. Lausen, "Propagation Models for Trust and Distrust in Social Networks," *Information Systems Frontiers*, vol.7, no.4-5, pp.337-358, 2005.
[13] L. Page, S. Brin, R. Motwani, T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *Technical Report*, Stanford Digital Library Technologies Project, 1998.
[14] J. A. Golbeck, "Computing and Applying Trust in Web-based Social Networks," *Ph.D. Dissertation University of Maryland-College Park*, 2005.
[15] L. A. Zadeh, "Fuzzy Sets, Fuzzy Logic, Fuzzy Systems," *World Scientific Press*, 1996.
[16] R. Falcone, G. Pezzulo, and C. Castelfranchi, "A Fuzzy Approach to a Belief-Based Trust Computation," *Lecture Notes in Computer Science*, vol.2631, pp.55-60, 2003.
[17] S. Song, K. Hwang, and M. Macwa, "Fuzzy Trust Integration for Security Enforcement in Grid Computing," *Lecture Notes in Computer Science*, vol.3222, pp.9-21, 2004.
[18] S. Song, K. Hwang, R. Zhou, and Y. Kwok, "Trusted P2P Transactions with Fuzzy Reputation Aggregation," *IEEE Internet Computing*, vol.9, no.6, pp. 24-34, 2005.
[19] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation," in *Proc of. 35th Int Hawaii Conf. on System Sciences*, pp.188-196, 2002.
[20] J. A. Golbeck, "Computing with Social Trust (Human-Computer Interaction Series)," *Springer*, 2009.
[21] B. Klimt and Y. Yang, "Introducing the Enron Corpus," in *Proc of.* 1st *Conf. on Email and Anti-Spam (CEAS)*, 2004.
[22] Enron Email Dataset, http://www.cs.cmu.edu/~enron/.

Yangjin Seo received B.S. and MS degrees in computer science and engineering from ChungAng University, South Korea, in 1998 and in 2000, respectively. After finishing Ph. D coursework from ChungAng University, he is currently with Softcamp in South Korea. His research interests include computer security, semantic web, information retrieval, machine learning, and online social networks.

Sangyong Han is a professor of the school of computer science and engineering, ChungAng University, Seoul, Korea. He received Bachelor of Engineering in College of Engineering from Seoul National University in 1975, and the Ph.D. degree in 1984. From 1984 to 1995, he worked at Poughkeepsie Lab. and Watson Research Center in IBM, USA. His research interests include next generation web, information retrieval, and optimization.