Computational Model of Association Activity Measure and Its Algorithmic Implementation

Xinquan Chen[1,2]
[1]School of Computer Science & Engineering, Chongqing Three Gorges University, Chongqing, China
[2]Web Sciences Center, University of Electronic Science & Technologhy of China, Chengdu, China
Email: chenxqscut@126.com

***Abstract*—To characterize one kind of association relation of different objects in a complex system, a computational model of association activity measure among different objects is presented after defined some concepts. To implement the computational model, a simple computational algorithm of association activity measure is introduced. After designed some appropriate data structures, an improved computational algorithm is presented by using the strategy of spending more space to decrease time cost. Through the simulations of one kind of artificial hypergraph with several pairs of parameters, we can verify that the two algorithms are equivalent and the improved algorithm holds some validity with less time cost. In the end, a research expectation is given to disinter and popularize the computational model and its algorithms.**

***Index Terms*—hypergraph, association relation, association activity measure, computational model**

I. Introduction

Recently, social networks [1] bring more challenges and opportunities to some researchers. Social network analysis [2], [3], [4], [5], [6] views the associated relationship of network elements by using some advanced techniques, such as data mining, graph theory, hypergraph theory, and so on. Social networks are often depicted in a social network diagram, where nodes are represented as points and ties are represented as lines.

Facebook [7], Quazza.com, Myspace, Orkut, and Twitter are several famous social network websites. Many researchers try to mining some valuable information from the large number of user generated content [8], [9]. For example, now the interpersonal tightness of social networks is becoming a research hotspot.

Due to the further expansion of social networks, exploring valuable information with multidimensional and multiscale relationship from massive object libraries in social networks attracts the attention of some researchers. In exploring the different relationship and interesting characters of complex social networks, hypergraph model obtains more research and application. Today, hypergraph [10], [11], [12], which is an extension of the traditional graph theory and can describe some complex systems more clearly, has obtained more attention in the fields of complex social network analysis, searching in Internet, searching in Internet of Things, and so on.

Hypergraph model can describe and organize objects of the real world very well by using a framework with some abstract data types. Hypergraph model gets a good development in Europe, especially in France. United States, Canada, Japan, and China, also have some reports about the hypergraph model theory and its applied research. Research and application confirm that hypergraph theory has strict data elements and flexible data structures.

The remainder of this paper is organized as follows. Section 2 gives some definitions and the computational model of association activity measure among different objects. Section 3 presents two examples of the computational model. Section 4 introduces a simple computational algorithm of the association activity measure and its improved version. Section 5 compares and validates the equivalence of the two algorithms and the effect of the improved algorithm by some simulation experiments. Conclusions and future works are presented in Section 6.

II. Some Definition and The Computational Model of Association Activity Measure

A. Description of The Problem and Some Related Definitions

Suppose a hypergraph H is described by a set of objects, $OS = \{o_1, o_2, \ldots, o_n\}$, a set of weight of objects, $OW = \{ow_1, ow_2, \ldots, ow_n\}$, a set of hyperedges, $ES = \{e_1, e_2, \ldots, e_m\}$, which describes the group relationship in the set OS, and a set of weight of hyperedges, $EW = \{ew_1, ew_2, \ldots, ew_m\}$. Obviously, this kind of hypergraph model can be used to describe some complex systems, such as the group system of Tencent's QQ and a schoolfellows' relationship network. In the group system of Tencent's QQ, the set of objects is composed of all QQ numbers, and the set of hyperedges is composed of all QQ group numbers. Tencent's QQ system can be described by a usual graph model and a hypergraph model together. And it can also be described by a hypergraph model. In the

Manuscript received June 23, 2013; revised September 27, 2013; accepted October 1, 2013. Corresponding author: Xinquan Chen (chenxqscut@126.com).

This work was supported by a project (grant No. 12RC01) from Chongqing Three Gorges University, China.

doi:10.4304/jsw.9.5.1135-1140

schoolfellows' relationship network, the set of objects is composed of ID numbers of all registered students, and the set of hyperedges is composed of ID numbers of all registered schools. If using a usual graph model to describe the schoolfellows' relationship network, a very complex graph is needed. If using a hypergraph model to describe the schoolfellows' relationship network, a very simple hypergraph is enough.

Definition 1. The associated matrix [10] $B = (b_{ik})_{n*m}$ between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$, can be defined as:

$$b_{ik} \equiv \begin{cases} 1 & o_i \in e_k \\ 0 & o_i \notin e_k \end{cases}, \quad i = 1,2,\ldots,n;\ k = 1,2,\ldots,m. \quad (1)$$

Definition 2. The set of associated hyperedges of object o_i ($i = 1, 2, \ldots, n$) can be defined as:

$$AES(o_i) \equiv \{e_k \mid o_i \in e_k, k \in \{1,2,\ldots,m\}\} \quad (2)$$

Usually, the number of hyperedges of object o_i is regarded as its degree of associated hyperedges, denoted by $|AES(o_i)|$. Obviously, $|AES(o_i)|$ is equal to the sum of the i-th row elements in the associated matrix B.

Definition 3. The set of associated objects of object o_i ($i = 1, 2, \ldots, n$) can be defined as:

$$AOS(o_i) \equiv \{o_j \mid o_i \in e_k, o_j \in e_k, j \neq i, j \in \{1,2,\ldots,n\}; k \in \{1,2,\ldots,m\}\} \quad (3)$$

Usually, the number of objects sharing common hyperedges with object o_i is regarded as its degree of straight-connected points, denoted by $|AOS(o_i)|$.

Definition 4. The set of objects of hyperedge e_k ($k = 1, 2, \ldots, m$) can be defined as:

$$EOS(e_k) \equiv \{o_i \mid o_i \in e_k, i \in \{1,2,\ldots,n\}\} \quad (4)$$

Usually, the number of objects contained in hyperedge e_k is regarded as its degree of associated points, denoted by $|EOS(e_k)|$. Obviously, $|EOS(e_k)|$ is equal to the sum of the k-th column elements in the associated matrix B.

Definition 5. Referencing associated matrix [10] between objects and hyperedges, the matrix of activity (or participation) measure, $C = (c_{ik})_{n*m}$, between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$, can be defined as:

$$c_{ik} \equiv \begin{cases} u(i,k) & o_i \in e_k \\ 0 & o_i \notin e_k \end{cases}, \quad i = 1,2,\ldots,n;\ k = 1,2,\ldots,m. \quad (5)$$

If object o_i belongs to hyperedge e_k, c_{ik} takes a measure $u(i, k)$ that reflects the activity (or participation) degree of object o_i in hyperedge e_k. The measure $u(i, k)$ is a positive number, an interval, or other measures suited to concrete problems.

Obviously, these definitions and concepts can also be applied to usual simple graphs. For example, when $|EOS(e_k)| = 2$ is satisfied for all $k = 1, 2, \ldots, m$, this kind of hypergraph will be a usual simple graph. In usual simple graphs, two and only two elements in each column of associated matrix B take 1, $|AOS(o_i)|$ is the degree of vertex o_i, $EOS(e_k)$ is the set of two vertexes of edge e_k, and $AES(o_i)$ is the set of edges containing vertex o_i. Generally, for most sparse graphs, associated matrix can save more storage space than adjacency matrix.

B. A General Computational Model of Associated Activity Measure

According to the matrix of activity (or participation) measure, $C = (c_{ik})_{n*m}$, between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$, and the associated similarity $sim(c_{ik}, c_{jk})$ between two different active elements c_{ik} and c_{jk} in the same hyperedge e_k, a general computational model of associated activity measure between two different objects o_i and o_j can be defined as:

$$ass(o_i, o_j) \equiv \sum_{k \in \{r \mid \{o_i, o_j\} \subseteq e_r, e_r \in ES\}} \left(sim(c_{ik}, c_{jk}) \cdot ew_k \right), \quad i \neq j, i, j \in \{1,2,\ldots,n\} \quad (6)$$

By using the associated matrix B between objects and hyperedges, equation (6) can also be expressed as:

$$ass(o_i, o_j) \equiv \sum_{k=1}^{m} \left(b_{ik} \cdot b_{jk} \cdot sim(c_{ik}, c_{jk}) \cdot ew_k \right), \quad i \neq j, i, j \in \{1,2,\ldots,n\} \quad (7)$$

Where the associated similarity $sim(c_{ik}, c_{jk})$ between two different active elements c_{ik} and c_{jk} in the same hyperedge e_k is often defined according to the concrete application problems. For example, two kind of definitions of associated similarity are presented in the next two examples.

C. A Concrete Computational Model of Associated Activity Measure

According to some concrete application problems, after given the matrix of activity (or participation) measure, $C = (c_{ik})_{n*m}$, between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$, a concrete computational model of associated activity measure between two different objects o_i and o_j can be defined as:

$$assc(o_i, o_j) \equiv \sum_{k \in \{r \mid \{o_i, o_j\} \subseteq e_r, e_r \in ES\}} \left(ew_k \cdot \frac{c_{ik} + c_{jk}}{\sum_{t \in \{r \mid o_r \in e_k\}} c_{tk}} \right), \quad i \neq j, i, j \in \{1,2,\ldots,n\} \quad (8)$$

Equation (8) means that the associated activity measure between two different objects o_i and o_j is their scale summation of activity (or participation) measure in their common hyperedges. By using the associated matrix B between objects and hyperedges, equation (8) can also be expressed as:

$$assc(o_i, o_j) \equiv \sum_{k=1}^{m} \left(\frac{b_{ik} \cdot b_{jk} \cdot (c_{ik} + c_{jk})}{\sum_{t=1}^{n} c_{tk}} \cdot ew_k \right), \quad i \neq j, i, j \in \{1,2,\ldots,n\} \quad (9)$$

Where the associated similarity $sim(c_{ik}, c_{jk})$ between two different active elements c_{ik} and c_{jk} in the same hyperedge e_k is often defined according to the concrete application problems. Obviously, after set $sim(c_{ik}, c_{jk}) =$

$(c_{ik}+c_{jk})/\sum_{t=1}^{n} c_{tk}$, equation (6) becomes the special computing equation (8).

III. Two Application Examples of The Computational Model

A. The Computational Problem of Schoolship Measure in Schoolfellow Network

Suppose there are n registered students who have registered their study experience and m registered schools which have been registered by those registered students in a schoolfellow database. This problem will become more simple and convenient if using a hypergraph model.

In this problem, it is better that the activity (or participation) measure $u(i, k)$ of registered student o_i in his (or her) school e_k takes one (or more) studying time section. If let $|c_{ik}|$ record the cumulative length of section $u(i, k)$ and let $samelength(c_{ik}, c_{jk})$ record the common studying time in school e_k of registered students o_i and o_j, then a more reasonable associated similarity of two schoolfellows o_i and o_j in school e_k can be defined as

$$sim(c_{ik},c_{jk}) = (\left|c_{ik}\right| / \sum_{t=1}^{n}\left|c_{it}\right| + \left|c_{jk}\right| / \sum_{t=1}^{n}\left|c_{jt}\right|)\cdot(1+samelength(c_{ik},c_{jk})) \quad (10)$$

Where $\sum_{t=1}^{n}\left|c_{it}\right|$ is the sum of all studying time of student o_i.

If all schools are equally important, then we can set $ew_k = 1$ for $k = 1, 2, \ldots, m$. According to the registered studying information of all students, it is easy to compute the associated similarity of any two schoolfellows by using equation (10). After that, the associated activity measure in schoolfellow network of any two students can be computed by using equation (6).

B. The Computational Problem of Intimacy Measure of Different Users in Tencent's QQ System

Suppose there are n registered QQ IDs and m registered QQ group IDs in Tencent's QQ database. Each QQ ID stores all QQ IDs of its friends and its QQ group IDs. This problem will become more simple and convenient if using a graph model and a hypergraph model.

In the hypergraph model, it is better that the activity (or participation) measure $u(i, k)$ of QQ user o_i in his QQ group e_k takes one (or more) speaking time section. If let $|c_{ik}|$ record the cumulative length of section $u(i, k)$, which is the sum of speaking time, and let $samelength(c_{ik}, c_{jk})$ record the common speaking time in group e_k of users o_i and o_j, which can be computed from the chatting log of group e_k. Then a more reasonable associated similarity of two users o_i and o_j in group e_k can be defined by using equation (10). Usually, we can regard the sum of online time of group e_k as ew_k for $k = 1, 2, \ldots, m$.

According to the chatting logs of all users, it is easy to compute the QQ group associated similarity of any two users by using equation (10). After that, the QQ group associated activity measure of any two users can be computed by using equation (6).

In the graph model, the sum of the chatting time or the inputing words time is the most appropriate activity (or participation) measure $t(o_i, o_j)$ of QQ users o_i and o_j. ow_i represents the total time of user o_i chatting with or inputing words to its all QQ friends, so as ow_j. Then a more reasonable QQ friend associated measure between two QQ users o_i and o_j can be defined as:

$$ars(o_i,o_j) = t(o_i,o_j)\cdot\left(1/ow_i + 1/ow_j\right) \quad (11)$$

According to the chatting information of all QQ users, after computed the QQ group associated activity measure of two QQ users o_i and o_j by using equation (6) and the QQ friend associated measure of two QQ users o_i and o_j by using equation (11), the intimacy measure of two QQ users o_i and o_j can be defined as a weighted equation (see equation (12)) of the QQ group associated activity measure and the QQ friend associated measure.

$$app(o_i,o_j) \equiv ars(o_i,o_j) + \lambda\cdot ass(o_i,o_j) \quad (12)$$

Where parameter λ is a weighted coefficient.

IV. The Computational Algorithms of Associated Activity Measure Among Objects

A. A Simple Computational Algorithm of Associated Activity Measure Among Objects

According to the above concrete computational model of associated activity measure, it is easy to design a simple computational algorithm.

Algorithm Name: a simple computational algorithm of associated activity measure among objects (Algorithm 1).

Input: a set of weight of hyperedges, $EW = \{ew_1, ew_2, \ldots, ew_m\}$; the matrix of activity measure, $C = (c_{ik})_{n*m}$, between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$.

Output: associated activity measure among different objects.

Procedure:

Step1. According to the matrix C, compute the sum of activity measure of objects of every hyperedge in the set of hyperedges, ES, by using equation (8). The concrete implementation of this step is as follows:

```
for (k = 1; k ≤ m; k++)
{
        SumowInek[k] ← 0;        /* The array element SumowInek [k] stores the sum of activity measure of all objects contained in hyperedge e_k. */
        for (j = 1; j ≤ n; j++)
                if (c_jk > 0)        /* Object o_j belongs to hyperedge e_k. */
                        SumowInek[k] ← SumowInek[k] + c_jk;
}
```

Step2. According to the matrix C and the set EW, compute the activity measure of two different objects o_i and o_j ($i < j$) by using equation (8). The concrete implementation of this step is as follows:

```
for (i = 1; i < n; i ++)
        for (j = i + 1; j ≤ n; j++)
        {
                a_ij ← 0;
```

for ($k = 1$; $k \le m$; k++)
if (($c_{ik} > 0$) **and** ($c_{jk} > 0$))
/* Objects o_i and o_j belong to hyperedge e_k at the same time. */
$a_{ij} \leftarrow a_{ij} + (c_{ik} + c_{jk}) * ew_k$ / SumowInek[k];
}

Step3. a_{ij} ($i < j$) is just the associated activity measure between objects o_i and o_j.

B. An Improved Computational Algorithm of Associated Activity Measure Among Objects

Algorithm Name: an improved computational algorithm of associated activity measure among objects (Algorithm 2).

Input: a set of weight of hyperedges, $EW = \{ew_1, ew_2, \ldots, ew_m\}$; the matrix of activity measure, $C = (c_{ik})_{n*m}$, between the set of objects, $\{o_1, o_2, \ldots, o_n\}$, and the set of hyperedges, $\{e_1, e_2, \ldots, e_m\}$.

Output: associated activity measure among different objects.

Procedure:

Step1. According to the matrix C, definition 2, and definition 3, construct the set of associated hyperedges, $AES(o_i)$, and the set of associated objects, $AOS(o_i)$, for every object o_i ($i = 1, 2, \ldots, n$). According to the matrix C and definition 4, construct the set of objects, $EOS(e_k)$, for every hyperedge e_k ($k = 1, 2, \ldots, m$).

The set $AES(o_i)$ of object o_i and the set $EOS(e_k)$ of hyperedge e_k can be constructed easily from the matrix C. The set $AOS(o_i)$ of object o_i is constructed from the set $AES(o_i)$ and the set $EOS(e_k)$. Its concrete construction process is described as follows:

/* Constructing the set $AES(o_i)$ of object o_i ($i = 1, 2, \ldots, n$). */
for ($i = 1$; $i \le n$; i ++)
{
The set $AES(o_i)$ is set as the empty set;
for ($k = 1$; $k \le m$; k++)
if ($c_{ik} > 0$) /* Object o_i belongs to hyperedge e_k. */
$AES(o_i) \leftarrow AES(o_i) \cup \{e_k\}$;
/* Hyperedge e_k is inserted into the set $AES(o_i)$. */
}

/* Constructing the set $EOS(e_k)$ of hyperedge e_k ($k = 1, 2, \ldots, m$). */
for ($k = 1$; $k \le m$; k++)
{
The set $EOS(e_k)$ is set as the empty set;
for ($i = 1$; $i \le n$; i ++)
if ($c_{ik} > 0$) /* Object o_i belongs to hyperedge e_k. */
$EOS(e_k) \leftarrow EOS(e_k) \cup \{o_i\}$;
/* Objects o_i is inserted into the set $EOS(e_k)$. */
}

/* Constructing the set $AOS(o_i)$ of object o_i ($i = 1, 2, \ldots, n$). */
for ($i = 1$; $i \le n$; i ++)
{
The set $AOS(o_i)$ is set as the empty set;
for (every element e_k in the set of $AES(o_i)$)
/* Hyperedge e_k belongs to the set $AES(o_i)$ of object o_i. */
$AOS(o_i) \leftarrow AOS(o_i) \cup (EOS(e_k) - \{o_i\})$;
/* The other elements in $EOS(e_k)$ is inserted into the set $AOS(o_i)$. */
}

Step2. According to the matrix C and the set $EOS(e_k)$ of hyperedge e_k ($k = 1, 2, \ldots, m$), compute the sum of activity measure of all objects contained in hyperedge e_k by using equation (8). The concrete implementation of this step is as follows:

for ($k = 1$; $k \le m$; k++)
{
SumowInek[k] $\leftarrow$ 0; /* The array element SumowInek [k] stores the sum of activity measure of objects contained in hyperedge e_k. */
for ($j = 1$; $j \le n$; j++)
if ($c_{jk} > 0$) /* Object o_j belongs to the hyperedge e_k. */
SumowInek[k] $\leftarrow$ SumowInek[k] + c_{jk};
}

Step3. According to the matrix C, the set EW, and two sets $AES(o_i)$ and $AOS(o_i)$ of object o_i ($i = 1, 2, \ldots, n$), compute the activity measure of two different objects o_i and o_j ($i < j$) by using equation (8). The concrete implementation of this step is as follows:

for ($i = 1$; $i < n$; i ++)
{
for (every object o_j which is in the set $AOS(o_i)$ subject to $j > i$)
{
$a_{ij} \leftarrow 0$;
Constructing the common associated edges between $AES(o_i)$ and $AES(o_j)$;
for (every hyperedge e_k in the common associated edges between $AES(o_i)$ and $AES(o_j)$)
$a_{ij} \leftarrow a_{ij} + (c_{ik} + c_{jk}) * ew_k$ / SumowInek[k]; /* Objects o_i and o_j belong to hyperedge e_k at the same time. */
}
for (every object o_j which is not in the set $AOS(o_i)$ subject to $j > i$)
$a_{ij} \leftarrow 0$;
}

Step4. a_{ij} ($i < j$) is the associated activity measure between objects o_i and o_j.

C. Some Notes of Two Algorithms

Some notes of algorithm 1:

(1) Time and space complexity analysis of algorithm 1:
Step1 needs Time = $O(n \cdot m)$ and Space = $O(n \cdot m)$.
Step2 needs Time = $O(n^2 \cdot m)$ and Space = $O(n \cdot m)$.

(2) Algorithm 1 is a simple computational method based on equation (8). It does not use any features of

concrete application. If matrix C is very sparse, an improved algorithm can be designed by constructing the set of associated hyperedges for every object, the set of associated objects for every object, and the set of objects for every hyperedge in advance.

Some notes of algorithm 2:

(1) In Step 1, if the set of associated hyperedges of every object, the set of associated objects of every object, and the set of objects of every hyperedge are stored in ordered adjacency list structure [13], [14] (sorted in ascending order according to the label), performance can be improved by speeding up the retrieval speed and reduce the storage space. The ordered adjacency list structure of hypergraph storing the set of associated objects for every object is similar to the adjacency table structure of graph.

(2) Time and space complexity analysis of algorithm 2:

Step1 needs Time = $O(n \cdot m + \sum_{i=1}^{n} \sum_{e_k \in AES(O_i)} (|EOS(e_k)| \cdot |AOS(O_i)|))$ and Space = $O(n \cdot m + \sum_{i=1}^{n} |AES(o_i)| + \sum_{i=1}^{n} |AOS(o_i)| + \sum_{k=1}^{m} |EOS(e_k)|)$.

Step2 needs Time = $O(\sum_{k=1}^{m} |EOS(e_k)|)$ and Space = $O(n + m + \sum_{k=1}^{m} |EOS(e_k)|)$.

Step3 needs Time = $O(\sum_{i=1}^{n} \left((|AOS(o_i)| + \sum_{o_j \in AOS(o_i)} (|AES(o_i)| + |AES(o_j)|)) + (n - |AOS(o_i)|) \right))$ and Space = $O(n + m + \sum_{i=1}^{n} |AES(o_i)| + \sum_{i=1}^{n} |AOS(o_i)|)$.

(3) Algorithm 2 is an improved algorithm based on equation (8) by constructing the set of associated hyperedges of every object, the set of associated objects of every object, and the set of objects of every hyperedge. In fact, in Step3, there is no need to compute the association activity measure between objects o_i and o_j which is not in the set $AOS(o_i)$ subject to $j > i$. In some applications, we can directly get the set $AES(o_i)$, the set $AOS(o_i)$, and the set $EOS(e_k)$. In this case, the time and space cost of algorithm 2 can be accepted by some current computer systems.

V. Simulation Experiments of Two Computational Algorithms

A. Experimental Design

Our simulation experiments are finished in a personal computer (Basic configuration: Intel(R) Pentium(R) Dual CPU T4500 2.3GHz, 2G Memory). Experimental environment is Visual C++6.0 under Windows XP.

To verify the correctness and the validity of two algorithms, there will be some experiments of one kind of artificial hypergraph in the next subsection.

The artificial hypergraph has n objects and m hyperedges , where parameters n and m are a pair of intergers. The matrix $C = (c_{ik})_{n*m}$ are designed by producing ($n*m$) random real numbers in the region [0, 1] such that the ($i*k$)-th random real number is assigned to the elment c_{ik} if it is larger than threshold parameter Δ. Otherwise 0 is assigned to the elment $c_{ik.}$ The set $EW = \{ew_1, ew_2, \ldots, ew_m\}$ are designed by producing m random positive real numbers.

In the next part, algorithm 2 will be compared with algorithm 1 by using several hypergraphs with different values of parameters n and m.

Performance of algorithms is measured by time cost (label: ST, unit: second). Effect of algorithms is measured by the percentage between the number of different pairs of objects whose association activity measure is larger than 0.00001 and the number of all different pairs of objects (label: Per).

B. Experimental Results

Table 1 and Table 2 list some comparative experimental results of algorithm 1 and algorithm 2 by processing the artificial hypergraph with several pairs of parameters (n, m).

TABLE I.
COMPARATIVE EXPERIMENTAL RESULTS OF TWO ALGORITHMS
(Threshold parameter Δ = 0.95 in the artificial hypergraph)

Pairs of parameters (n, m)		Algorithm 1		Algorithm 2	
n	***m***	***ST***	***Per***	***ST***	***Per***
10000	5	2	1.27%	5	1.27%
10000	10	2	2.47%	13	2.47%
10000	20	-	-	-	-
2000	5	0	1.20%	0	1.20%
2000	10	0	2.54%	0	2.54%
2000	20	0	5.04%	0	5.04%
2000	30	0	7.41%	1	7.41%
2000	40	1	9.62%	1	9.62%
2000	50	1	11.77%	1	11.77%
2000	60	1	13.65%	1	13.65%
2000	70	-	-	-	-

TABLE II.
COMPARATIVE EXPERIMENTAL RESULTS OF TWO ALGORITHMS
(Threshold parameter Δ = 0.99 in the artificial hypergraph)

Pairs of parameters (n, m)		Algorithm 1		Algorithm 2	
n	***m***	***ST***	***Per***	***ST***	***Per***
10000	5	2	0.05%	0	0.05%
10000	10	2	0.11%	0	0.11%
10000	20	-	-	-	-
2000	50	0	0.53%	0	0.53%
2000	80	0	0.80%	-	-
2000	100	0	1.02%	-	-
2000	120	1	1.22%	-	-
2000	130	-	-	-	-

C. Analysis and Conclusions of Experimental Results

From Table 1, Table 2, and other experimental results, we observe that algorithm 1 and algorithm 2 have the same experimental results except the running time and space cost. This shows that algorithm 2 is equivalent to algorithm 1 in essence. Comparing Table 1 with Table 2, we find that the larger the threshold parameter Δ is, the greater improvement algorithm 2 gets in time performance.

Although parameters *n* and *m* in the experiments cannot be set too large due to memory constraints, the simulation results can still reflect the performance improvement of algorithm 2.

VI. Conclusions

In the environment of complex social networks, there is a wide variety of diverse relations among massive objects. In order to describe these relations more easily, this paper presents a computational model of associated activity measure among different objects and gives two concrete algorithms to achieve this computational model. Through designing appropriate data structures, an improved algorithm can be constructed based on the simple algorithm. Simulations of artificial hypergraph verify that the two algorithms are equivalent and the time performance of the simple computational algorithm can be improved. We can foresee that this kind of computational model has some theoretical value and application prospects.

The next work is to do more experimental comparison and to study the relationship between the value of parameters and the experimental results.

References

[1] Boyd D M, Ellison N B. Social network sites: Definition,history, and scholarship. *Journal of Computer-Mediated Communication*, vol. 13 no. 1, pp. 252-279, 2007.

[2] Pinheiro, Carlos A.R. *Social Network Analysis in Telecommunications*. John Wiley & Sons, 2011.

[3] Jie Tang, Jimeng Sun, Chi Wang, Zi Yang. "Social Influence Analysis in Large-scale Networks," *In Proceedings of the 15th ACM SIGKDD*, 2009, pp. 807-816.

[4] Gyeong-Mi Park, Sung-Hwan Kim, Hwan-Gue Cho. "Intuitive Network Applications: Learning for Personalized Converged Services Involving Social Networks," *Journal of Computer*, vol. 2 no. 6, pp. 72-84, 2007.

[5] Min Peng, ZhengQuan Xu, ShaoMing Pan, Rui Li, Tengyue Mao. "AgentTMS: A MAS Trust Model based on Agent Social Relationship," *Journal of Computer*, vol. 7 no. 6, pp. 1535-1542, 2012.

[6] Gyeong-Mi Park, Sung-Hwan Kim, Hwan-Gue Cho. "Structural Analysis on Social Network Constructed from Characters in Literature Texts," *Journal of Computer*, vol. 8 no. 9, pp. 2442-2447, 2013.

[7] Atif Nazir, Saqib Raza, Chen-Nee Chuah. "Unveiling Facebook: A Measurement Study of Social Network Based Applications," *In Proceedings of Internet Measurement Conference*, New York, ACM Press, 2008, pp. 43-56.

[8] Clauset A, Newman M E J, Moore C. "Finding community structure in very large networks," *Physical Review E*, vol. 70 no. 6, 066111, 2004.

[9] Haewoon Kwak, Yoonchan Choi, Young-Ho Eom. "Mining Communities in Networks: A Solution for Consistency and Its Evaluation," *In Proceedings of the 9th ACM Internet Measurement Conference*, New York, ACM Press, 2009, pp. 302-314.

[10] Berge C. *Hypergraphs*, North—Holland, 1989.

[11] Zhiping Wang, Zhongtuo Wang. *Supernetwork theory and its application*, The Science Press, China, 2008.

[12] Vitaly I. Voloshin. *Introduction to Graph and Hypergraph Theory*, Nova Science Publishers, Inc., 2009.

[13] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, 3rd ed., The MIT Press, USA, 2009.

[14] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*, 8th ed., Morgan Kaufmann, USA, 2006.

Xinquan Chen received the PhD degree from South China University of Technology in China. He is now an associate professor of Chongqing Three Gorges University in China and a postdoctor of University of Electronic Science & Technologhy of China. His research interests include data mining, machine learning, clustering algorithms, and optimization methods.