International Journal of Database Theory and Application
Vol.8, No.2 (2015), pp.171-184
http://dx.doi.org/10.14257/ijdta.2015.8.2.16

A Method of Description on the Data Association Based on Granulation Trees

Yan Shuo and Yan Lin

School of Computer and Information Technology, Beijing Jiao Tong University, Beijing, China
College of Computer and Information Engineering, Henan Normal University, Xinxiang, China
1hnsdyl@163.com, 211112097@bjtu.edu.cn

Abstract

To investigate the association of data with other data in reality, the research begins with data sets which are divided into different partitions. Because each partition consists of granules and owns a level, all the partitions constitute a granulation set whose elements are the granules. As a hierarchy system, the granulation set together with the inclusion relation gives rise to a structure called a granulation tree. The research on the data association establishes a method to describe the associations of the data in a granulation tree with the data in another granulation tree. The method involves a necessary and sufficient condition used to check the data associations. Because the necessary and sufficient condition is bound up with the upper approximation, the study also develops a way of investigation into rough sets. As an example, a practical problem is modeled by granulation trees, and the associations of the data in a granulation tree with the data in another granulation tree can be examined by use of the necessary and sufficient condition. Meanwhile, because the study is closely linked to granules and alterations of granularity, the process can be viewed as an approach to research on granular computing.

Keywords: *Data set, Granulation tree, Data association, Upper approximation, Granule, level*

1. Introduction

The development of information science and technology accompanies various researches into data processing. In the meanwhile, the researches cover further investigations on new topics. This vigorously promotes the progress of data research, and gives rise to different research directions, such as data mining [1-4], data reasoning [5-8], data reduction [9-11], data warehousing [12, 13], as well as big data, and so on which are all subjects focused on by researchers and have become academic branches. Many topics involved by the directions always inspire the interest of investigators, and are often taken as focus issues to be studied by experts. So research achievements are obtained, which also bring about new topics and further arouse widespread concern. At the same time, the discussion of integrating different topics enhances the research level, and pushes forward the development of the academic research. However, in terms of data processing, there are still problems which remain to be explored. For example, data association is a problem that is rarely concerned by researchers. Here the data association means the situation that a common data associates data with other data. So the data association mentioned here refers to the association of some data with others, which is done by taking a common data as a bridge to link the data.

ISSN: 2005-4270 IJDTA

Actually, if we pay attention to observation of real life, we can be conscious of this data association that often exists around us. Consider the following examples:

a) A student leads to the association of his hometown people with his classmates in a university.

b) Joe Smith suggests the association of his family members with his company staff.

c) Discrete mathematics establishes the association of the computer courses with the mathematics courses.

d) An airliner from airport A to airport B sets up the association of the airliners of airport A with the airliners of airport B.

e) A newlywed brings about the association of one family with another family.

f) An amphibian builds the association of land animals with marine animals.

g) A spy causes the association of one side with opposing side.

These associations have some things in common reflected by the fact that a common object associates the data in a class with the data in another class. If we refer to each of the associations as the data association, how we can describe the data association will be related to the establishment of a mathematical model that is the basis of algorithm design and computer programming. Moreover, the automation management and automatic information inquiry for the data association also depend on the model. Because the data association often exists around us, it deserves to be investigated carefully. What we do in the following will focus on it.

These examples shows that each of them involves a common object. If each common object is called an association data, then the student, Joe Smith, discrete mathematics, airliner, newlywed, amphibian and spy are association data. Their double role makes it possible to associate one data class with another class, thereby generates the data associations.

Because the previous work rarely or never involves the research on the data association, the discussion on it will be significant in theory and practice. What we do in the following will demonstrate our work that will include the theoretical exploration, as well as the description on an actual problem.

It follows that an association data plays a role in associating a data with another data. In the meanwhile, the close degree of the data association is relevant to the classification of data sets. The coarse or fine classes that the data sets are divided into are linked with the closeness of data association. For instance, in example *a*), a student can leads to the association of a township people with his classmates in a faculty. If the township is divided into villages, and the faculty is divided into grades, then the student also associates a village with a grade. Obviously, the association of a village with a grade is closer than a township with a faculty. Of course, a township and a village, or a faculty and a grade are subclasses of the hometown people set, or the university student set that are taken as data sets. So the coarse or fine classification of data sets correlates with the close degree of the data association. This is linked to some concepts in granular computing [14-16], such as granules and alterations of granularity. Generally, a granule is regarded as a clump of data drawn from a data set by a property. Any change of the property will determine the amount of data in the granule, also affects the data's indistinguishability. This is usually viewed as the alteration of granularity.

Therefore, in order to describe the data association, we are going to create a method that is consistent with the data processing mode proposed in granular computing. To do this, we need to construct a structure called a granulation tree which will be closely linked to granules and their granularity. So our study can also be viewed as a method of research on granular computing. For this purpose, we start from a data set.

2. Fundamentals

Let K be a set whose elements are referred to as data. So K is called a data set. In order to study the data association, it needs to consider the classification of a data set.

Definition 1[14] Given a data set K, let $E=\{E1, E2, \cdots, Ek\}(k\geq 1)$, where $Ei\subseteq K(i=1, 2, \cdots, k)$. E is called a ***partition*** of K, if it satisfies the following conditions:

(1) $Ei\neq\emptyset(i=1, 2, \cdots, k)$.

(2) $Ei\cap Ej=\emptyset(i\neq j$ and $1\leq i, j\leq k)$.

(3) $E1\cup E2\cup\cdots\cup Ek = K$.

In this case, each Ei ($\in E$) is called a ***granule*** of K.

A partition $E=\{E1, E2, \cdots, Ek\}$ is a set whose elements are subsets of K, each of which is called a granule. From definition 1 we know that for $Ei\in E$ and $Ej\in E$, if $Ei\cap Ej\neq\emptyset$, then $Ei = Ej$; also if $u\in K$, then $u\in Ei$ for a granule $Ei\in E$. These will be used in the following discussion.

Now consider the granules of E. Each granule consists of the data which satisfy the same property, or have the common characteristic. Sometimes, the granules of E need to be divided into smaller granules. So consider the following definition:

Definition 2[14] Let $E=\{E1, E2, \cdots, Ek\}$ and $F=\{F1, F2, \cdots, Fr\}$ be two partitions of the data set K. For any $Fj\in F$, if there exists a granule $Ei\in E$ such that $Fj\subseteq Ei$, then F is called a ***sub-partition*** of E.

When $F=\{F1, F2, \cdots, Fr\}$ is a sub-partition of $E=\{E1, E2, \cdots, Ek\}$, it follows form definitions 1 and 2 that if $Ei\in E$, then $Ei=Fj_1\cup Fj_2\cup\cdots\cup Fj_s$ for some granules Fj_1, $Fj_2, \cdots, Fj_s\in F$. This means that the granule Ei is divided into the smaller granules Fj_1, $Fj_2, \cdots$, and Fj_s.

Let $E=\{E1, E2, \cdots, Ek\}$ and $F=\{F1, F2, \cdots, Fr\}$ be two partitions of the data set K. If F is a sub-partition of E, then we have the conclusion:

Conclusion 1 For $Ei\in E$ and $Fj\in F$, if there is a data $x\in K$ such that $x\in Ei$ and $x\in Fj$, then $Fj\subseteq Ei$.

Also, when $G=\{G1, G2, \cdots, Gt\}$, $E=\{E1, E2, \cdots, Ek\}$ and $F=\{F1, F2, \cdots, Fr\}$ are partitions of the data set K, the following conclusion holds:

Conclusion 2 If F is a sub-partition of E, and E is a sub-partition of G, then F is a sub-partition of G.

Given a partition $E=\{E1, E2, \cdots, Ek\}$ of K, let $R=\{<x, y> \mid x, y\in K$ and there is a granule $Ei\in E$ such that $x, y\in Ei\}$. Then R is an equivalence relation on K, and is referred to as the equivalence relation on K relative to E. Meanwhile R corresponds to the set $K/R=\{ [x] \mid x\in K\}$, where $[x]$ is the R-equivalence class about x, i.e. $[x]=\{ y \mid y\in K$ and $<x, y>\in R\}$. In this case, K/R is a partition of K and $K/R= E$ [14]. In fact, K/R and R are determined by each other. So if $R1$ and $R2$ are equivalence relations on K relative to E and F respectively, where E and F are partitions of K, then $R1= R2$ if and only if $K/R1=K/R2$, if and only if $E=F$. This means that an equivalence relation on K uniquely corresponds to a partition of K, and vice versa [14].

Usually, an equivalence relation or a partition is bound up with a property that can determine the equivalence relation or the partition. For example, let K be a data set consisting of the students in a university. A relation on K, denoted by R, is defined as follows:

For $x, y\in K$, $<x, y>\in R$ if and only if x and y belong to the same faculty.

Then R is determined by the property "the students belong to the same faculty". Obviously, R is reflective, symmetric and transitive. Hence R is an equivalence relation on K, of course, R corresponds to the partition K/R that is also determined by the property.

Thus when we say that E is a partition of K relative to a property, we mean that the property determines an equivalence relation R and $K/R=E$.

What we discussed in this section can be found in [14]. We review them is aimed at the holistic consideration on our discussion.

3. Granulation Tree Based on Data Set

Given a data set K, if $E=\{K\}$, it is trivial that E is a partition of K. In addition to this, being relative to different properties, different partitions of K can be obtained. If one of them is a sub-partition of another, each partition will correspond to a level. All the partitions can constitute a hierarchy set consisting of granules, each of which belongs to one of the partitions. In order to demonstrate the hierarchy characteristic, we now set up an algorithm to show the process of how the hierarchy set is generated.

Hierarchy Algorithm:

Step (1) For a data set K, let $U=\{K\}$, $k=0$ and $Sk=\{K\}$. Enter an integer n, proceed to

Step (2) Given a property, seek a partition E of K relative to the property such that E is a sub-partition of Sk, proceed to

Step (3) Let $k=k+1$, $Sk=E$ and $U=U\cup Sk$, proceed to

Step (4) If $k=n$, output the set U, the algorithm terminates. If $k<n$, repeat step (2).

The purpose of entering the integer n in step (1) is to seek n partitions $S1, S2,\cdots$, and Sn of K. This algorithm is aimed at getting the set $U=S0\cup S1\cup S2\cup\cdots\cup Sn(S0=\{K\})$ by use of the recursion formula $U=U\cup Sk$. Obviously, U is a granule set because the partition $Sk(k=0, 1,\cdots, n)$ consists of granules. The subscript k occurring in $Sk(k=0, 1,\cdots, n)$ will defined as the level of Sk. Thus U is constituted by the different level partitions, and is a hierarchy set. The property that the partition E is relative to in step (2) means $E=K/R$ and R is the equivalence relation determined the property. Of course, n partitions involve n different properties, each of which is given when the algorithm loops to step (2). Since $Sk+1(k=0, 1,\cdots, n-1)$ is a sub-partition of Sk, it follows from conclusion 2 in section 2 that Sk is a sub-partition of Sr if $k>r$.

We introduce this algorithm is to show the different levels of the partitions $S0, S1,\cdots$, and Sn. The algorithm can help us understand the hierarchy characteristic of the set U. Our intention is to show the generation process of U, rather than to formulate a program according to the steps in the algorithm. Actually, if we formulated a computer program, we would consider how to introduce the property in step (2) that is used to seek the partition $Sk(=E)$.

So if we say that U is obtained by use of the Hierarchy Algorithm on K, we mean that $U=S0\cup S1\cup S2\cup\cdots\cup Sn(S0=\{K\})$, where $Sk(k=0, 1, 2,\cdots, n)$ is a partition of K, and Sk is a sub-partition of $Sr(k>r)$. In this case, U consists of the granules in $S0\cup S1\cup S2\cup\cdots\cup Sn$. For $Ei\in U$, there is a partition Sk $(0\le k\le n)$ such that $Ei\in Sk$ and $Ei\subseteq K$. Thus when $Ei\in U$ and $Ej\in U$, it is possible to have $Ei\subseteq Ej$ or $Ej\subseteq Ei$. The inclusion relation $\subseteq$ is actually a partial order relation on $U(=S0\cup S1\cup S2\cup\cdots\cup Sn)$. By use of the set U together with the inclusion relation $\subseteq$, we can get a structure denoted by $T(K)=(U, \subseteq)$.

Definition 3 The structure $T(K)=(U, \subseteq)$ is called an ***n-hierarchy granulation tree*** induced by K, or ***granulation tree*** for short. It is also linked to the following concepts:

(1) $U(=S0\cup S1\cup S2\cup\cdots\cup Sn$ $(S0=\{K\}))$ is called an ***n-hierarchy granulation set*** based on K, or ***granulation set*** for short. The partition Sk $(k=0, 1,\cdots, n)$ is called the ***kth-level partition*** in $T(K)$, and the subscript k is referred to as the ***level*** of Sk.

(2) If $Ei\in U$, Ei is called a ***granule*** of K. Moreover, when $Ei\in Sk$ $(0\le k\le n)$, Ei is also called a ***granule of the kth-level*** in $T(K)$, and the subscript k in Sk is referred to as the ***level*** of the granule Ei.

(3) For $a, b\in K$ and $Ei\in U$, a and b are ***Ei-identical***, or ***Ei-identical*** in $T(K)$ if $a, b\in Ei$, which is also referred to as the ***(Ei, a, b)-identity***, or the ***(Ei, a, b)-identity*** in $T(K)$.

Moreover, when $Ej \in U$, the (Ej, a, b)-*identity* is ***more proximate*** than the (Ei, a, b)-*identity* if $Ej \subseteq Ei$.

The (Ei, a, b)-identity illustrates that a and b all belong to the granule Ei, i.e. $a, b \in Ei$. In this case, we think that Ei does not distinguish a from b, in other words, a and b are considered as the same. Moreover, when $a, b \in Ej$, it follows that a and b are also Ej-identical. If $Ej \subseteq Ei$ and $Ej \neq Ei$, then $Ej \in Sk$ and $Ei \in Sr$ such that $k>r$. The level of Sk is greater than the level of Sr, or Sk is a sub-partition of Sr. The partition that has a greater level must be relative to a stronger property. Thus, when the (Ej, a, b)-identity is more proximate than the (Ei, a, b)-identity, the (Ej, a, b)-identity means a and b are gathered together by a stronger property.

This definition involves the concept "granule" which originates in granular computing, a current research focus. If making an intuitive explanation, researchers generally regard a granule as a part of a data set, or a clump of data drawn from the data set. When $Ei \in U(=S0 \cup S1 \cup S2 \cup \cdots \cup Sn)$, we have $Ei \subseteq K$, i.e. Ei is a subset of K. Of course Ei is a part of K, or a clump of data drawn from K. This is reason why we define Ei as a granule in definition 1 and in definition 3(2). This is consistent with the intuitive understanding of a granule.

Since U is obtained by use of the Hierarchy Algorithm on K, the granulation tree $T(K)=(U, \subseteq)$ also relies on the data set K. The notation $T(K)$ that K occurs in it has shown the dependence of $T(K)$ on K. Obviously, $T(K)=(U, \subseteq)$ is closely linked to the partitions $S1, S2, \cdots$, and Sn, which have different levels.

When $T(K)$ is drawn on the plane, it will be displayed as a diagram called a tree. This is the reason why we call $T(K)$ a granulation tree. For instance, consider the data set $K=\{a1, a2, a3, a4, a5, a6, a7, a8, a9\}$. Let $U=S0 \cup S1 \cup S2 \cup S3(S0=\{K\})$ be a 3-hierarchy granulation set based on K, where $S1= \{\{a1, a2, a3, a4, a5\}, \{a6, a7, a8, a9\}\}$, $S2 = \{\{a1, a2, a3\}, \{a4, a5\}, \{a6, a7, a8, a9\}\}$, $S3 = \{\{a1, a2\}, \{a3\}, \{a4, a5\}, \{a6, a7\}, \{a8, a9\}\}$. Then $T(K)=(U, \subseteq)$ is a 3-hierarchy granulation tree induced by K, shown in Figure 1.

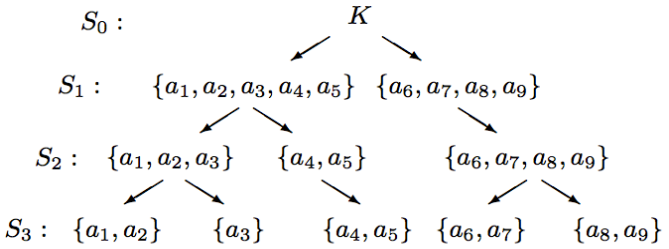


Figure 1. Granulation Tree *T* (*K*)

Figure 1 shows a structure that is a tree in which there is an arrow from Ei to Ej if and only if $Ei \in Sk$, $Ej \in Sk+1(k=0, 1, 2)$ and $Ej \subseteq Ei$. The hierarchy characteristic of the granulation tree is reflected by the different levels of $S0$, $S1$, $S2$ and $S3$. The data set K is taken as the ***root***. The kth-level partition $Sk(k=0, 1, 2, 3)$ in $T(K)$ constitutes the kth-level of the tree. The granules $\{a1, a2\}$, $\{a3\}$, $\{a4, a5\}$, $\{a6, a7\}$ and $\{a8, a9\}$ located in the lowest level are ***leaves***.

Remark: a granule is allowed to occur in different levels. For instance, the granule $\{a4, a5\}$ belongs to both $S2$ and $S3$ in Figure 1.

4. Data Association Based on Granulation Trees

From the examples in section 1, we can see that the associations of data with other data are always related to two data classes. So we consider two data sets $K1$ and $K2$ in this

section. According to the steps in the Hierarchy Algorithm, we can obtain $T(K1)=(U1, \subseteq)$ and $T(K2)= (U2, \subseteq)$, an m-hierarchy granulation tree, and an n-hierarchy granulation tree induced by $K1$ and $K2$ respectively, where $U1=S0 \cup S1 \cup S2 \cup \cdots \cup Sm(S0=\{K1\})$ and $U2=T0 \cup T1 \cup T2 \cup \cdots \cup Tn(T0 =\{K2\})$. The symbol Tj (j=1, 2,⋯, n) used by us is to distinguish it from Sk (k=1, 2,⋯, m). Here, we allow $m=n$, also allow $m \neq n$.

4. 1. Data Association

In this section, we always assume $K1 \cap K2 \neq \varnothing$. Thus there is a data a such that $a \in K1 \cap K2$.

Definition 4 For the data sets $K1$ and $K2$, if $a \in K1 \cap K2$, a is called an ***association data*** of $K1$ and $K2$.

When $Ei \in U1$ and $Fj \in U2$, we have $Ei \in Sk$ and $Fj \in Tr$, where $Sk(0 \leq k \leq m)$ is the *kth*-level partition in $T(K1)$ and $Tr(0 \leq r \leq n)$ is the *rth*-level partition in $T(K2)$. According to definition 3(2), Ei is a granule of the *kth*-level in $T(K1)$, and Fj is a granule of the *rth*-level in $T(K2)$.

Now consider the granulation trees $T(K1)=(U1, \subseteq)$ and $T(K2)=(U2, \subseteq)$, where $U1=S0 \cup S1 \cup S2 \cup \cdots \cup Sm(S0=\{K1\})$ and $U2=T0 \cup T1 \cup T2 \cup \cdots \cup Tn(T0=\{K2\})$. Let Ei be a granule of the *kth*-level in $T(K1)$, i.e. $Ei \in Sk$, and Fj be a granule of the *rth*-level in $T(K2)$, i.e. $Fj \in Tr$. For $a \in K1 \cap K2$, $b \in K1$ and $c \in K2$, it is possible to have $a, b \in Ei$ and $a, c \in Fj$. In situations like this, the data a and b are Ei-identical in $T(K1)$, meanwhile, the data a and c are Fj-identical in $T(K2)$(see definition 3(3)).

Definition 5 Let Ei be a granule of the *kth*-level in $T(K1)$, i.e. $Ei \in Sk$, and Fj be a granule of the *rth*-level in $T(K2)$, i.e. $Fj \in Tr$. For $a \in K1 \cap K2$, $b \in K1$ and $c \in K2$, the data b and c are **(*a, k, r*)-*associated*** if a and b are Ei-identical in $T(K1)$, meanwhile a and c are Fj-identical in $T(K2)$. In this case, it is also referred to as the **(*a, k, r*)- *association* of *b* with *c***.

The (a, k, r)-association of b with c establishes links between the data in $T(K1)$ and the data in $T(K2)$. The triple (a, k, r) not only specifies the association data a that associates b with c, but also uses the numbers k and r to represent the levels of the granules Ei and Fj, where Ei is a granule of the *kth*-level in $T(K1)$ and $a, b \in Ei$; Fj is a granule of the *rth*-level in $T(K2)$ and $a, c \in Fj$. Any change of k or r will lead to the change of the level of the granule Ei or Fj. This may also change the amount of the data in Ei or in Fj. So, the number k or r is connected with the property that determines the *kth*-level partition in $T(K1)$, or the *rth*-level partition in $T(K2)$. The data belonging to a granule of the *kth*-level or the *rth*-level must satisfy the property. Thus the numbers k and r are numerical representation of the data information.

Sometimes, we only use "***data association***" to express the (a, k, r)-association of b with c, if we do not have to specify the data a, b and c.

4. 2. Research on Data Association

How can we conclude that b and c are (a, k, r)-associated is what we are going to study. To do this, we will use the upper approximation to examine the data association. From rough set theory [17-18] we know that the upper approximation and lower approximation form an approach to approximate description of knowledge. In related researches, upper and lower approximations are always combined together to carry out approximate data processing. They depend on each other, and are considered as dual operators.

In the following, we will use the upper approximation to examine the data association, which has nothing to do with the lower approximation. The upper approximation will be

taken as operator to check the (a, k, r)-association of b with c, rather than to describe knowledge. Therefore the discussion will embody our own ideas.

Generally, the upper approximation can be defined by use of an equivalence relation. Since an equivalence relation is uniquely corresponds to a partition, the upper approximation can be also produced by a partition. For the sake of argument, we review its definition:

Let K be a data set, and $E=\{E1, E2, \cdots, Ek\}$ be a partition of K. For a subset $X \subseteq K$, if $E^*(X)$ denotes the E-upper approximation about X, then $E^*(X)$ is defined by the following expression:

$$E^*(X)= \cup \{ Et \mid Et \in E \text{ and } Et \cap X \neq \varnothing\}[17]$$

Thus the E-upper approximation about X is equal to the union of the granules such that the intersection of each of the granules and X is not empty. By this definition, it is easy to know that $X \subseteq E^*(X)$ [17]. Sometimes, we refer to $E^*(X)$, the E-upper approximation about X, as the upper approximation if we do not emphasize the partition E or the subset X. As an operator, the upper approximation can be viewed as a form of granular computing [14]. Therefore our discussion will involve an approach to granular computing. The approach will connect the upper approximation with the data association.

For $x \in K$, we have $\{x\} \subseteq K$. So if $E=\{E1, E2, \cdots, Ek\}$ is a partition of K, then the E-upper approximation about $\{x\}$, i. e. $E^*(\{x\})$, exists. We now have a conclusion about $E^*(\{x\})$:

Lemma 1 Let $E=\{E1, E2, \cdots, Ek\}$ be a partition of K, $x \in K$ and $Ei \in E$. Then $E^*(\{x\})=Ei$ if and only if $x \in Ei$.

Proof Suppose that $E^*(\{x\})=Ei$. Since $\{x\} \subseteq E^*(\{x\})$, we have $\{x\} \subseteq Ei$. This means $x \in Ei$.

Conversely, suppose that $x \in Ei$. Then $\{x\} \subseteq Ei$ which derives $\{x\} \cap Ei \neq \varnothing$. In this case, for any $Ej \in E$, when $Ej \neq Ei$, $\{x\} \cap Ej=\varnothing$. Hence $E^*(\{x\})= \cup \{ Et \mid Et \in E$ and $Et \cap \{x\} \neq \varnothing\}=Ei$.

Let $T(K)=(U, \subseteq)$ be an m-hierarchy granulation tree induced by K, where $U=S0 \cup S1 \cup S2 \cup \cdots \cup Sm(S0=\{K\})$. Consider $Sk(1 \leq k \leq m)$ and $Sr(1 \leq r \leq m)$ which are the kth-level and rth-level partitions in $T(K)$ respectively. If $x \in K$, we can get the upper approximations $Sk^*(\{x\})$ and $Sr^*(\{x\})$. The next lemma is linked with them.

Lemma 2 if $k \geq r$, then $Sk^*(\{x\}) \subseteq Sr^*(\{x\})$.

Proof If $k=r$, then $Sk=Sr$. It is obvious that $Sk^*(\{x\})=Sr^*(\{x\})$. Naturally $Sk^*(\{x\}) \subseteq Sr^*(\{x\})$.

Now assume $k>r$. Since Sk and Sr are partitions of K, for $x \in K$, there are granules $Ej \in Sk$ and $Ei \in Sr$ such that $x \in Ej$ and $x \in Ei$. The fact $k>r$ means Sk is a sub-partition of Sr. Since $x \in Ej$ and $x \in Ei$, it follows from conclusion 1 in section 2 that $Ej \subseteq Ei$. Also lemma 1 indicates that $Sk^*(\{x\})=Ej$ and $Sr^*(\{x\})=Ei$. Thus $Sk^*(\{x\}) \subseteq Sr^*(\{x\})$.

As the key component of the granulation tree $T(K)=(U, \subseteq)$, the granulation set U is a hierarchy set that is linked to the partitions $S0, S1, S2, \cdots$, and Sm. Because the partition $Sk(k=1, \cdots, m)$ can produce upper approximation, this has provided the condition for using the upper approximation to examine the (a, k, r)-association of b with c.

Now consider two data sets $K1$ and $K2$. Then $T(K1)=(U1, \subseteq)$ and $T(K2)=(U2, \subseteq)$ can be obtained, where $U1=S0 \cup S1 \cup S2 \cup \cdots \cup Sm(S0=\{K1\})$ and $U2=T0 \cup T1 \cup T2 \cup \cdots \cup Tn(T0=\{K2\})$. They are an m-hierarchy granulation tree and an n-hierarchy granulation tree induced by $K1$ and $K2$, respectively. For a partition $Sk(1 \leq k \leq m)$ in $T(K1)$ and a partition $Tr(1 \leq r \leq n)$ in $T(K2)$, if $b \in K1$ and $c \in K2$, the upper approximations $Sk^*(\{b\})$

and $Tr^*(\{c\})$ exist. When $K1 \cap K2 \neq \varnothing$, for $a \in K1 \cap K2$, we are able to use the upper approximations $Sk^*(\{b\})$ and $Tr^*(\{c\})$ to check whether b and c are (a, k, r)-associated.

Theorem 1 Let $a \in K1 \cap K2$, $b \in K1$ and $c \in K2$. Then b and c are (a, k, r)-associated if and only if $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$, where Sk is the kth-level partition in $T(K1)$, and Tr is the rth-level partition in $T(K2)$.

Proof Suppose that b and c are (a, k, r)-associated. Then a and b are Ei-identical in $T(K1)$, meanwhile a and c are Fj-identical in $T(K2)$, where $Ei \in Sk$ and $Fj \in Tr$, i.e. Ei is a granule of the kth-level in $T(K1)$, and Fj is a granule of the rth-level in $T(K2)$. It follows from definition 3(3) that $a, b \in Ei$ and $a, c \in Fj$. Thus $b \in Ei$, $c \in Fj$ and $a \in Ei \cap Fj$. By lemma 1 we have $Sk^*(\{b\}) = Ei$ and $Tr^*(\{c\}) = Fj$. So $Sk^*(\{b\}) \cap Tr^*(\{c\}) = Ei \cap Fj$. From $a \in Ei \cap Fj$, we know $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$.

Conversely, suppose that $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$. Then $a \in Sk^*(\{b\})$ and $a \in Tr^*(\{c\})$. Since Sk is the kth-level partition in $T(K1)$ and $b \in K1$, as well as Tr is the rth-level partition in $T(K2)$ and $c \in K2$, there exist $Ei \in Sk$ and $Fj \in Tr$ such that $b \in Ei$ and $c \in Fj$. It follows from lemma 1 that $Sk^*(\{b\}) = Ei$ and $Tr^*(\{c\}) = Fj$. This concludes $a \in Ei$ and $a \in Fj$ from $a \in Sk^*(\{b\})$ and $a \in Tr^*(\{c\})$. It follows that $a, b \in Ei$ and $a, c \in Fj$, i.e. a and b are Ei-identical in $T(K1)$, meanwhile a and c are Fj-identical in $T(K2)$, Notice that Ei is a granule of the kth-level, and Fj is a granule of the rth-level in $T(K1)$ and $T(K2)$ respectively. Thus b and c are (a, k, r)-associated.

Theorem 1 offers a necessary and sufficient condition for checking the (a, k, r)-association of b with c. It is bound up with the upper approximation, but not relevant to the lower approximation. Also, the condition only takes the upper approximation as an operator, rather than use it to describe knowledge. So our discussion differs from the previous researches which always combine the upper approximation with the lower approximation to make the approximate description about knowledge.

The subscripts k and r in $Sk^*(\{b\})$ and $Tr^*(\{c\})$ are the levels of the partitions Sk and Tr in $T(K1)$ and $T(K2)$ respectively. They represent the properties that determine the partitions Sk and Tr, and can be thought of as the numerical representation of the data information.

4. 3. Comparison between Data Associations

When b and c are (a, k, r)-associated, what will it be if the association data a, or the numbers k and r change? We now proceed to this problem.

Consider the granulation trees $T(K1) = (U1, \subseteq)$ and $T(K2) = (U2, \subseteq)$, where $U1 = S0 \cup S1 \cup S2 \cup \cdots \cup Sm (S0 = \{K1\})$ and $U2 = T0 \cup T1 \cup T2 \cup \cdots \cup Tn (T0 = \{K2\})$.

First, given two association data $a, a' \in K1 \cap K2$, it is possible to have $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$ and $a' \in Sk^*(\{b\}) \cap Tr^*(\{c\})$, where $b \in K1$ and $c \in K2$. If it is so, it follows from theorem 1 that b and c are both (a, k, r)-associated and (a', k, r)-associated. We now make a definition show the relationship between the data associations.

Definition 6 The (a, k, r)-association of b with c is ***identical to*** the (a', k, r)-association of b with c if b and c are not only (a, k, r)-associated, but also (a', k, r)-associated.

So when the (a, k, r)-association of b with c is identical to the (a', k, r)-association of b with c, the triple (a, k, r) contains the same numbers k and r as those in (a', k, r), although a and a' may be different.

Theorem 2 If the (a, k, r)-association of b with c is identical to the (a', k, r)-association of b with c, then a and a' are Ei-identical in $T(K1)$, meanwhile a and a' are Fj-identical in $T(K2)$, where $Ei \in Sk$ and $Fj \in Tr$.

Proof When the (a, k, r)-association of b with c is identical to the (a', k, r)-association of b with c, it follows from definition 6 that b and c are both (a, k, r)-associated and (a', k, r)-associated. So a, $b \in Ei$ and a, $c \in Fj$, as well as a', $b \in Es$ and a', $c \in Ft$, where Ei and Es are granules of the kth-level in $T(K1)$; Fj and Ft are granules of the rth-level in $T(K2)$. Thus Ei, $Es \in Sk$ and Fj, $Ft \in Tr$. Notice $b \in Ei \cap Es$, i.e. $Ei \cap Es \neq \varnothing$. We have $Ei = Es$ because Ei, $Es \in Sk$ and Sk is a partition. Similarly, $Fj = Ft$ because $c \in Fj \cap Ft$ and Fj, $Ft \in Tr$. These illustrate that a, $a' \in Ei$ ($=Es$), and a, $a' \in Fj$ ($=Ft$). Hence a and a' are Ei-identical in $T(K1)$, meanwhile, a and a' are Fj-identical in $T(K2)$.

Therefore, the (Ei, a, a')-identity in $T(K1)$, as well as the (Fj, a, a')-identity in $T(K2)$ determines the result that the (a, k, r)-association of b with c is identical to the (a', k, r)-association of b with c. The identity between the data associations is based on the data identity in $T(K1)$ and in $T(K2)$.

Second, given data $a \in K1 \cap K2$, $b \in K1$ and $c \in K2$, if b and c are both (a, k, r)-associated and (a, s, t)-associated, then by comparing k with s, or r with t, we will be able to discuss the relationship between the (a, k, r)-association and the (a, s, t)-association of b with c. To this end, we introduce the notations:

$(s, t) > (k, r)$ if and only if $s > k$ and $t \geq r$, or $s \geq k$ and $t > r$, where s, t, k and r are natural numbers.

$(s, t) \geq (k, r)$ stands for $(s, t) > (k, r)$ or $(s, t) = (k, r)$(i.e. $s = k$ and $t = r$).

Definition 7 (1) Let b and c be both (a, s, t)-associated and (a, k, r)-associated. The (a, s, t)-association of b with c is ***closer*** than the (a, k, r)-association of b with c if $(s, t) > (k, r)$.

(2) Let b and c be (a, k, r)-associated. The (a, k, r)-association of b with c is ***maximal*** if b and c are not (a, s, t)-associated for $(s, t) > (k, r)$.

The (a, s, t)-association of b with c means a, $b \in Ei'$ and $Ei' \in Ss$, as well as a, $c \in Fj'$ and $Fj' \in Tt$. Also the (a, k, r)-association of b with c means a, $b \in Ei$ and $Ei \in Sk$, as well as a, $c \in Fj$ and $Fj \in Tr$. When the (a, s, t)-association of b with c is closer than the (a, k, r)-association of b with c, we have $(s, t) > (k, r)$ which means $s > k$ and $t \geq r$, or $s \geq k$ and $t > r$. Thus Ss is a sub-partition of Sk, or Tt is a sub-partition of Tr. By conclusion 1 in section 1 we have $Ei' \subseteq Ei$ or $Fj' \subseteq Fj$. Since a, $b \in Ei'$ and a, $b \in Ei$, as well as a, $c \in Fj'$ and a, $c \in Fj$, we also know that the (Ei', a, b)-identity is more proximate than the (Ei, a, b)-identity in $T(K1)$, or the (Fj', a, c)-identity is more proximate than the (Fj, a, c)-identity in $T(K2)$(see definition 3(3)). Hence, the closer data association relies on the data identity that is more proximate, and can be examined by comparing (k, r) with (s, t).

Theorem 3 For $a \in K1 \cap K2$, $b \in K1$ and $c \in K2$, the following conclusions hold:

(1) When $(s, t) > (k, r)$, if b and c are (a, s, t)-associated, then b and c must be (a, k, r)-associated.

(2) If b and c are (a, k, r)-associated, then there exists a (s, t) such that $(s, t) \geq (k, r)$, and the (a, s, t)-association of b with c is maximal.

Proof (1) Suppose that b and c are (a, s, t)-associated. It follows from theorem 1 that $a \in Ss^*(\{b\}) \cap Tt^*(\{c\})$. When $(s, t) > (k, r)$, we have $s > k$ and $t \geq r$, or $s \geq k$ and $t > r$. It follows from lemma 2 that $Ss^*(\{b\}) \subseteq Sk^*(\{b\})$ and $Tt^*(\{c\}) \subseteq Tr^*(\{c\})$. This implies $Ss^*(\{b\}) \cap Tt^*(\{c\}) \subseteq Sk^*(\{b\}) \cap Tr^*(\{c\})$. Thus $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$. From theorem 1 again, we conclude that b and c are (a, k, r)-associated.

(2) Suppose that b and c are (a, k, r)-associated. It follows from theorem 1 that $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$. Let s and t be the largest subscripts in $Ss^*(\{b\})$ and in $Tt^*(\{c\})$ respectively, satisfying $a \in Ss^*(\{b\}) \cap Tt^*(\{c\})$. It is obvious that $(s, t) \geq (k, r)$, meanwhile theorem 1 indicates that b and c are (a, s, t)-associated. From the values of s and t, we conclude that the (a, s, t)-association of b with c is maximal.

If $(s, t) > (k, r)$, the above proof shows that $Ss^*(\{b\}) \cap Tt^*(\{c\}) \subseteq Sk^*(\{b\}) \cap Tr^*(\{c\})$. Assume $Ss^*(\{b\}) \cap Tt^*(\{c\}) \neq \varnothing$. Then $Sk^*(\{b\}) \cap Tr^*(\{c\}) \neq \varnothing$. Let $a' \in Ss^*(\{b\}) \cap Tt^*(\{c\})$ and $a \in Sk^*(\{b\}) \cap Tr^*(\{c\})$. By theorem 1, b and c are both (a', s, t)-associated and (a, k, r)-associated. Although $(s, t) > (k, r)$, we are not sure that the (a', s, t)-association of b with c is closer than the (a, k, r)-association of b with c, because definition 7(1) requires that a' and a are the same data. However, since $Ss^*(\{b\}) \cap Tt^*(\{c\}) \subseteq Sk^*(\{b\}) \cap Tr^*(\{c\})$ and $a' \in Ss^*(\{b\}) \cap Tt^*(\{c\})$, we have $a' \in Sk^*(\{b\}) \cap Tr^*(\{c\})$. So b and c are (a', k, r)-associated. Because the (a', k, r)-association of b with c is identical to the (a, k, r)-association of b with c(see definition 6), and when $(s, t) > (k, r)$, the (a', s, t)-association of b with c is closer than the (a', k, r)-association of b with c, we may think that the (a', s, t)-association of b with c is closer than the (a, k, r)-association of b with c.

The (a, k, r)-association of b with c not only bases the data association on the association data a, but also relies on the (Ei, a, b)-identity and the (Fj, a, c)-identity, where Ei is a granule of the kth-level in T ($K1$), and Fj is a granule of the rth-level in $T(K2)$. Especially, the way of using the upper approximation to check the (a, k, r)-association of b with c shows an application of rough set theory. The application is different from the previous researches in which the upper approximation always combines with the lower approximation to make the approximate description about knowledge. So our discussion gives a different way of research on rough sets.

5. Description of an Actual Problem

What we have done can be taken as a mathematical model to deal with problems. It is also the basis for algorithm design and computer programming. We now use it to describe an actual problem.

Example 1 Let $K1$ and $K2$ be two data sets which are defined as follows:

$K1$ consists of the people born in a county. The people live in the county, or leave the county for college.

$K2$ is a set of students studying at several universities, or graduated from the universities in recent 5 years.

Consider the data set $K1$, the people set of the county. In China a county is constituted by townships. A township contains a number of villages. Each village is generally divided into teams. A team consists of men or women. According to these divisions, we can get a 4-hierarchy granulation tree induced by $K1$, denoted by $T(K1)=(U1, \subseteq)$ in which $U1=S0 \cup S1 \cup S2 \cup S3 \cup S4(S0=\{K1\})$. Where Si $(i=1, 2, 3, 4)$ is a partition of $K1$ relative to a property, and Sk is a sub-partition of $Sr(k>r)$. Specifically, if $K1$ is classified into subsets, each of which is the set of a township people, then we get the partition $S1$ in which a granule is one of the subsets, corresponding to a township. $S2$ is sub-partition of $S1$, a granule of $S2$ consists of the people of a village, in this case, a township, a granule of $S1$, is divided into villages. $S3$ is a sub-partition of $S2$ such that each granule in $S2$ is divided into teams which are granules of $S3$. When a team in $S3$ is classified into the man set and woman set, we can get the partition $S4$ that is sub-partition of $S3$. These partitions constitute $U1(=S0 \cup S1 \cup S2 \cup S3 \cup S4)$ that is the major component of the granulation tree $T(K1)=(U1, \subseteq)$.

Also, consider the data set $K2$ consisting of the students studying at the universities or graduated from the universities in recent 5 years. The students who get into or graduated from the universities in 5 years will involve ten grades. Based on $K2$, a 6-hierarchy granulation set can be obtained, denoted by $U2=T0 \cup T1 \cup T2 \cup T3 \cup T4 \cup T5 \cup T6(T0=\{K2\})$. Also, $U2$ together with $\subseteq$ constitutes $T(K2)=(U2, \subseteq)$, a 6-hierarchy granulation tree induced by $K2$. As a partition of $K2$ relative to a property, $Ti(i=1, 2, 3, 4,$

5, 6) is a set of granules, and Tt is sub-partition of $Ts(t>s)$. Specifically, $T1$ is a partition of $K2$, a granule of $T1$ is the student set of a university. $T2$ is sub-partition of $T1$ such that a university is divided into faculties, and the students in a faculty constitute a granule of $T2$. By dividing a faculty into departments, we get the partition $T3$ in which a granule corresponds to a department, clearly, $T3$ is a sub-partition of $T2$. Because a department involves different grades, a grade taken as a granule leads to the partition $T4$ that is a sub-partition of $T3$. A grade is constituted by a number of classes. So we can get the partition $T5$ that takes each class as a granule, and forms a sub-partition of $T4$. Since a class contains male students and female students, this leads to $T6$ that is a sub-partition of $T5$, and a granule of $T6$ is a male student set or female student set. The partitions $T1$, $T2$, $T3$, $T4$, $T5$ and $T6$ constitute the 6-hierarchy granulation set $U2$.

On such occasions, we get the partitions of $K1$, as well as the partitions of $K2$, each of which is relative to a property. These give rise to the granulation trees $T(K1)=(U1, \subseteq)$ and $T(K2)= (U2, \subseteq)$, taking $U1=S0\cup S1\cup S2\cup S3\cup S4(S0= \{K1\})$ and $U2=T0\cup T1\cup T2\cup T3\cup T4\cup T5\cup T6(T0 =\{K2\})$ as their main parts respectively.

For $b\in K1$ and $c\in K2$, examine $Sk^*(\{b\})\cap Tr^*(\{c\})$, where $1\le k\le 4$ and $1\le r\le 6$. When $Sk^*(\{b\})\cap Tr^*(\{c\})\neq\varnothing$, there must be an association data a, i.e. $a\in K1\cap K2$, such that $a\in Sk^*(\{b\})\cap Tr^*(\{c\})$. It follows form theorem 1 that b and c are (a, k, r)-associated. So $a, b\in Ei$ and $a, c\in Fj$, where Ei is a granule of the kth-level in $T(K1)$, and Fj is a granule of the rth-level in $T(K2)$. The association data a associates the data of Ei with the data of Fj. The numbers k and r representing the levels of Ei and Fj respectively are intimately bound up with the close degree of the (a, k, r)-association of b with c.

We now focus on a specific situation such that $k=2$ and $r=5$. In situation like this, the data b and c are $(a, 2, 5)$-associated. For the granules Ei and Fj, we have $Ei\in S2$ and $Fj\in T5$. So Ei represents a village, and Fj corresponds to a class because $S2$ takes villages as granules, and a granule of $T5$ consists of the students of a class. The association data a in $(a, 2, 5)$ can be viewed as a bridge connecting a village with a class. The numbers 2 and 5 are the numerical representation of the data information.

In addition, when $(s, t) > (2, 5)$ or $(s, t)<(2, 5)$, by use of theorems 1, 2 or 3, it is entirely possible to determine whether data b and c are (a, s, t)-associated. If $(s, t) > (2, 5)$, the (a, s, t)-association of b with c will be closer than the $(a, 2, 5)$-association. In this situation, the numbers s and t will imply much more information.

Based on the granulation trees $T(K1)=(U1, \subseteq)$ and $T(K2)=(U2, \subseteq)$, we make an approach to associating a local government with the university talents. What we have developed can be taken as a mathematical model to describe associations of data with data. The model is the basis of the algorithm design. This may make it possible to realize computerized management of data associations. The research is the fundamental work of computer programming.

The association of a local government with the universities is relevant to the government's economic development, talent introduction, technical progress, enterprise transformation, etc. Meanwhile it is also linked to knowledge transformation of the universities, including talent employment, training base, places for practice, technology application, and so on. The discussion of the association of data with data offers the mathematical basis for algorithm design and computer programming. It is significant in theory and practice.

6. Conclusion

The discussion of the (a, k, r)-association of b with c only relates to the data association occurring in two granulation trees. When $K1$, $K2,\cdots$, and Kn are n data sets, we can obtained n granulation trees $T(K1)=(U1, \subseteq)$, $T(K2) =(U2, \subseteq),\cdots$, and $T(Kn)=(Un, \subseteq)$ induced by the data sets. In this case, the data association connecting

with the *n* granulation trees will become a path, which will bring up a subject of research on association paths. An association path will involve multiple data, and lead to multiple data associations. This can be taken as a research topic to be investigated in the future.

If we look at the above discussion, we can see that the work we do is closely connected with granules and alterations of granularity. For instance, the (*a*, *k*, *r*)-association of *b* with *c* is close linked to the granules that determine the identity between *a* and *b* in $T(K1)$, as well as the identity between *a* and *c* in $T(K2)$. Also, the close degree of the data association is bound up with alterations of granularity. All of these are based on the granulation trees which have close links with granules. Thus, what we have done in this paper may suggest a method of research on granular computing.

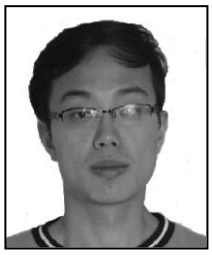
Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 11371003, as well as by the Natural Science Foundation of Guangxi under Grant No. 2011GXNSFA018154 and No. 2012GXNSFGA060003.

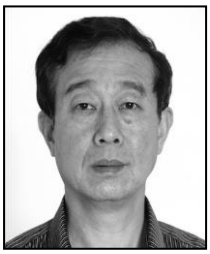
References

[1] J. B. Zhang, T. R. Li and H. M. Chen, "Composite rough sets for dynamic data mining," Information Sciences, vol. 257, **(2014)**, pp. 81-100.
[2] J. B. Zhang, T. R. Li and D. Ruan, "Neighborhood rough sets for dynamic data mining," International Journal of Intelligent Systems, vol. 27, no. 4, **(2012)**, pp. 317-342.
[3] P. Honko, "Association discovery from relational data via granular computing," Information Sciences, vol. 234, **(2013)**, pp. 136-149.
[4] J. M. Merigo, "The probabilistic weighted average and its application in multiperson decision making," International Journal of Intelligent Systems, vol. 27, no. 5, **(2012)**, pp. 457-476.
[5] Y. H. She, "On the rough consistency measures of logic theories and approximate reasoning in rough logic", International Journal of Approximate Reasoning, vol. 55, no. 1, **(2014)**, pp. 486-499.
[6] L. Yan and S. Yan, "Granular reasoning and decision system's decomposition", Journal of Software, vol. 7, no. 3, **(2012)**, pp. 683-690.
[7] L. Yan and Q. Liu, "Researches on granular reasoning based on granular space", Proceedings of 2008 IEEE International Conference on Granular Computing, Hangzhou, China, **(2008)**, pp. 706- 711.
[8] L. Yan and Q. Liu, "Granular resolution and granular reasoning", Proceedings of 2009 IEEE International Conference on Granular Computing, Nanchang, China, **(2009)**, pp. 668-671.
[9] J. H. Li, C. L. Mei and Y. J. Lv, "Incomplete decision contexts: approximate concept construction, rule acquisition and knowledge reduction", International Journal of Approximate Reasoning, vol. 54, no. 1, **(2013)**, pp. 149-165.
[10] X. Y. Jia, W. H. Liao and Z. M. Tang, "Minimum cost attribute reduction in decision-theoretic rough set models," Information Sciences, vol. 219, **(2013)**, pp. 151-167.
[11] R. A. McAllister, R. A. Angryk, "Abstracting for dimensionality reduction in text classification," International Journal of Intelligent Systems, vol. 28, no. 2, **(2013)**, pp. 115-138.
[12] A. Tagarelli, "Exploring dictionary-based semantic relatedness in labeled tree data," Information Sciences, vol. 220, **(2013)**, pp. 244-268.
[13] F. G. Cozman, "Independence for full conditional probabilities: structure, factorization, non-uniqueness, and bayesian networks," International Journal of Approximate Reasoning, vol. 54, no. 9, **(2013)**, pp. 1261- 1278.
[14] L. Yan, "Fundamentals of Mathematical Logic and Granular Computing," Science Press, Beijing, China **(2007)**, (in Chinese).
[15] W. Pedrycz, "Granular Computing: Analysis and Design of Intelligent Systems," CRC Press/Francis Taylor, Boca Raton, FL, USA, **(2013)**.
[16] A. Skowron, J. Stepaniuk and R. Swiniarski, "Modeling rough granular computing based on approximation spaces", Information Sciences, vol. 184, **(2012)**, pp. 20-43.
[17] Z. Pawlak, "Rough Set—Theoretical Aspects of Reasoning about Data," Kluwer Academic Publishers, Dordrecht, Holland, **(1992)**.
[18] T. F. Fan, "Rough set analysis of relational structures," Information Sciences, vol. 221, **(2013)**, pp. 230-244.

Authors



Yan Shuo, Computer Science M. Sc., graduated from Beijing Jiaotong University. Now he is a Ph. D. candidate of Beijing Jiaotong University. His research interests include mathematical logic, decision logic and computer algebra.



Yan Lin, Computer Science M. Sc., graduated from Institute of Software, Chinese Academy of Science. His research interests include mathematical logic, non-classical logic, rough set theory, granular computing and rough logic.

He is a professor of College of Computer and Information Engineering, Henan Normal University.

