International Journal of Physics and Mathematical Sciences ISSN: 2277-2111 (Online)
An Open Access, Online International Journal Available at http://www.cibtech.org/jpms.htm
2015 Vol. 5 (3) July- September, pp. 7-13/Mao et al.
Research Article

APPROXIMATION DISTANCE CALCULATION ON CONCEPT LATTICE

Hua Mao, *Ran Kang, Ming Shi and Yichao Liu
Department of Mathematics and Information Science, Hebei University, Baoding 071002, China
**Author for Correspondence*

ABSTRACT
The heart of the extended rough set theory is the generalization of the notion of the standard set inclusion degree. Considering the equivalent relationship between equivalence class in rough set theory and concept in concept lattice, we will study a classification problem of undefinable object set (undefinable object set is the object set that is not an extension) in concept lattice. First, with the definition of inclusion degree, we propose a definition of approximation distance. Second, we introduce an algorithm to vectorize a set. Third, in a context, with the assistance of the algorithm above, we find out an algorithm to calculate a closest extension to an undefinable object set. We also use an example to illustrate the feasibility of this algorithm. Finally, we conclude this paper and point out the future works.

Keywords: *Inclusion Degree, Concept Lattice, Approximation Distance, Undefinable Object Set*

INTRODUCTION
Rough set theory is proposed by Pawlaw (1982) based on equivalence relations for handling the inexact, incomplete and uncertainty data. It is widely used for knowledge classification and rule learning (Dai, 2012; Miao, 2011). Owing to the restrictions of equivalence relations, classical rough set has large limitations in its application. On the other hand, in classical rough set model, the classification is fully correct and certain, and all conclusions are only applied to the set of objects. Therefore, as a generalization, Ziarko (1993) provided variable precision rough set model by introducing the measure of the relative degree of misclassification in classical rough set model, aimed at the classification problems involving uncertain and imprecise information. Moreover, some researchers have more development based on variable precision rough set model.
Xu and Zhang, (2013) combined the algorithm β-upper and β-lower distribution reduction in variable precision rough sets with the characteristics of context in concept lattice, and proposed an algorithm of concept lattice reduction based on variable precision rough set. Mao and Kang (2015) presented a definition of lower and upper approximations in concept lattice and generated the lower and upper approximations concept of concept lattice. Zhu and Zhu (2014) proposed a variable precision covering-based rough set model based on functions by introducing misclassification rate functions and presented the concepts of the β-lower and β-upper approximations.
Concept lattice theory is proposed by Wille (1982). Utilized the mapping relationship between objects and attributes from concepts, all the concepts constitute a concept lattice. In recent years, as an effective tool for knowledge classification and learning, concept similarity is introduced. Some scholars have carried on the thorough research on concept similarity measure. Huang *et al*., (2015) have introduced a bounded transitive similarity graph based on FCA concept similarity computing method in order to reduce the size of computing similarity by means of similarity graph. Clobanu and Vaideanu (2014) put forward new types of similarity relations between objects and attributes in fuzzy attribute-oriented concept lattices and analyzed the properties. Huang *et al*., (2015) proposed a semantic information content based FCA concept similarity computation method.
Inclusion degree takes advantage of the relationship between sets. It focuses on set-inclusion problems, but its application scope is limited. In context, concept similarity can only compute the similarity between concepts, and does not capture the similarity between concept and undefinable object. In this paper, we combine the inclusion degree in variable precision rough set and the equivalent relationship between equivalence class and extension, and put forward an approximation distance operator. We calculate the

degree of approximation between undefinable object and an extension in concept lattice. Aimed at solving the classification problem of undefiable object set in concept lattice, two algorithms are proposed

The content is arranged as follows: Section 2 reviews basic definitions of variable precision rough set and concept lattice. In Section 3, we first propose a definition of approximation distance. Afterwards, vectorization process is introduced. After that, we provide an algorithm to calculate the closest extension to an undefinable object set. And then, an example is given to show the effectiveness of this algorithm. The paper is concluded in Section 4.

Preliminaries

To facilitate our discussion, this section reviews some notions related to variable precision rough set and concept lattice. More details for variable precision rough set, please see (Pawlak, 1982), and for concept lattice, please refer to (Davey and Priestley, 2002).

Variable Precision Rough Set

This subsection introduces some contents for variable precision rough set. In this paper the parameter β ($0\leq\beta<0.5$) denotes the degree of misclassification.

Definition 1 (Chmielewski and Grzymala-Busse, 1996) Let U be a finite set. Let $X,Y\subseteq U$, and $X,Y\notin\{\phi\}$. Then we define inclusion degree as follows:

$\mu(X,Y)= card(X\cap Y)/card(X)$, (1)

where $card(Z)$ denotes the set cardinality of Z .

Remark 1 The inclusion degree μ of two sets is defined as the degree of one set contained in another set. In general, the following conditions are satisfied (Zhang and Leung, 1996):

(1) $0 \leq\mu(X,Y)\leq 1$;

(2) $X\subset Y$ if and only if $\mu(X,Y)=1$;

(3) If $X\subset Y\subset Z$, then $\mu(X,Z)\leq\mu(X,Y)$.

Based on the inclusion degree, the measure $c(X$, Y) of the relative degree of misclassification of a set X with respect to a set Y is defined.

Definition 2 (Ziarko, 1993) We defined $c(X,Y)$ as follows:

$c(X,Y)=1-card(X\cap Y)/card(X)$ if $card(X)>0$, (2)

$c(X,Y)=0$ if $card(X)=0$, (3)

where $card(Z)$ denotes the set cardinality of Z .

The specified majority requirement the admissible level of classification error β must be within the range $0\leq\beta<0.5$. In variable precision rough set the notion of β-lower and β-upper approximation as follows.

Definition 3 (Ziarko, 1993) Let $X\subseteq U$, its generalized notion of β-lower approximation is defined by:

$\underline{R}_\beta X = \cup\{E\in R^*|\ c(E,X)\leq\beta\}$; (4)

the β-upper approximation of a set $X\subseteq U$ is defined as:

$\overline{R}_\beta X = \cup\{E\in R^*|\ c(E,X)<1-\beta\}$, (5)

where E is an equivalence class, and $R^*=\{E_1, E_2, \ldots, E_n\}$ is an set of equivalence class on U.

Remark 2 In fact, the β-lower approximation of a set X can be interpreted as the collection of all those elements of U which can be classified into X with the classification error not greater than β. Similarly, the β-upper approximation of the set X is the collection of all those elements of U which can be classified into X with the classification error less than $1-\beta$.

Concept Lattice

Concept lattice deals with visual presentation and analysis of data (Yao and Chen, 2006; Ganter and Wille, 1999; Xu *et al*., 2009) and focuses on the definability of a set of objects based on a set of attributes, and vice versa.

Definition 4 (Davey and Priestley, 2002) A context is a triple (G,M,I) where G and M are sets and $I\subseteq G\times M$. The elements of G and M are called objects and attributes respectively. As usual, instead of writing $(g,m)\in I$, we write gIm and say 'the object g has the attribute m'.

For $A\subseteq G$ and $B\subseteq M$, we define

$A'=\{m\in M \mid (\forall g\in A)gIm\}$; (6)
$B'=\{g\in G \mid (\forall m\in B)gIm\}$. (7)

From Definition 4, Davey (Davey and Priestley, 2002) points that A' is the set of attributes common to all the objects in A, and B' is the set of objects possessing the attributes in B.

Definition 5 (Davey and Priestley, 2002) For $A\subseteq G$ and $B\subseteq M$, the pair (A,B) is called a concept of (G,M,I) if $A=B'$ and $B=A'$, and A is the extension of the concept, B is the intension of the concept.

Definition 6 (Davey and Priestley, 2002) (1) The set of all concepts from a context (G,M,I) is called a concept lattice and is denoted by:

$$\boldsymbol{B}(G,M,I)=\{(A,B) \mid A\in G, B\in M \text{ and } A=B',\ B=A'\}. \tag{8}$$

(2) For concepts (A_1,B_1) and (A_2,B_2) in $\boldsymbol{B}(G,M,I)$, we write $(A_1,B_1)\le(A_2,B_2)$, if $A_1\subseteq A_2$. Also $A_1\subseteq A_2$ implies that $A_1'\supseteq A_2'$, and the reverse implication is also valid, because $A_1 = A_1''$ and $A_2''=A_2$. We therefore have

$$(A_1,B_1)\le(A_2,B_2)\Leftrightarrow A_1\subseteq A_2 \Leftrightarrow B_1\supseteq B_2. \tag{9}$$

Lemma 1 (Davey and Priestley, 2002) The relation $\le$ is an order on $\boldsymbol{B}(G,M,I)$. We still call $\langle \boldsymbol{B}(G,M,I),\le\rangle$ is a concept lattice.

Definition 7(Zhi *et al*., 2008) In a domain ontology and a number of contexts (G_i, M_i, I_i), i=1… k, the similarity (*Sim*) of two concepts (A_1,B_1) and (A_2,B_2) are defined as follows:

$$Sim((A_1,B_1),(A_2,B_2))=\left(\frac{|A_1\cap A_2|}{m}*a+\frac{|B_1\cap B_2|}{n}*b\right)*(1+c)^{(l_1+l_2)}, \tag{10}$$

where $a+b=1$, $c>0$, $m=\max(|A_1|,|A_2|)$, $n=\max(|B_1|,|B_2|)$. l_1, l_2 are the value of depth in concept lattice, c=0.01 is a numerical value to reflect the depth of similarity.

Relationship between Equivalence Class and Concept

Xu (Xu *et al*., 2009) proposed that a concept in a concept lattice is corresponding to the equivalence class in rough set. Let x be a set and $[x]$ be an equivalence classes for an equivalence relation defined on x.

Lemma 2 (Xu *et al*., 2009) Let (G,M,I) be a context, $P\subseteq M$ and $P\neq\phi$. The following statements hold:

(1) Let$(A,B)\in\boldsymbol{B}(G,M,I)$. Then $A=[x]_B$ is correct for any $x\in A$.

(2) $([x]_P, [x]_{P'})\in\boldsymbol{B}(G,M,I)$ is correct for any $x\in G$.

Lemma 2 shows that any extension of a concept for a given context must be an equivalence class of a rough set. Conversely, any equivalence class in a rough set is an extension in concept lattice.

Algorithm of Classification

In this section, we solve classification problem of an undefinable object set (undefinable object set is the object set that is not an extension). Based on inclusion degree and similarity concept, we first propose approximate distance in concept lattice. It can effectively reflect the degree of similarity between an undefinable object set and an extension in concept lattice. To realize this idea completely, a vectorization process and classification algorithm is introduced. At last, an example is given to show the feasibility of this algorithm.

Definition of Approximation Distance

The paper (Mao and Kang, 2015) proposed the notions of the β-upper and β-lower approximation operators in concept lattice. In this paper, based on the β-upper and β-lower approximation, we will propose a method to classifing an undefinable object set, and find out attribute set which are shared by this object. Next, we will present a new algorithm to seek out the closest extension to an undefinable object set. Before introducing the algorithm, we first give the definition of approximation distance.

Definition 8 Let $E=(e_1,e_2,\ldots,e_n)$, $X=(x_1,x_2,\ldots,x_n)$be two n-dimensional vectors. The approximation distance ($Adis(E,X)$) of X relative to E is defined as:

$$Adis(E,X)=\frac{\sqrt{(e_1-x_1)^2+(e_2-x_2)^2+\ldots+(e_n-x_n)^2}}{\sqrt{e_1^2+e_2^2+\ldots+e_n^2}}. \tag{11}$$

Remark 3 Definition 8 uses second norm to calculate the distance between two vectors in order to divide the length of the vector E and calculate the approximation distance.

The following example will illustrate Definition 8.

Example 1 Let $E_1=(0.5, 0)$, $E_2=(1, 1)$ and $X=(1, 1.25)$ be three vectors in a two-dimensional vector space. We can use the approximation distance to finding out which one is closest to X between E_1 and E_2.

From Definition 8, we can obtain:

$$Adis(E_1, X) = \frac{\sqrt{(0.5-1)^2 + (0-1.25)^2}}{\sqrt{(0.5)^2}} \approx 2.693,$$

$$Adis(E_2, X) = \frac{\sqrt{(1-1)^2 + (1-1.25)^2}}{\sqrt{(1)^2 + (1)^2}} \approx 0.177.$$

Comparing the above calculations, we know that X is closest to E_2.

Algorithm Process

For a given context (G,M,I), we can use NextClsoure Algorithm (Thomas *et al*., 2005) to obtain all the extensions $\{E_j, (j=1,2,\ldots,m)\}$. We know that the complexity of NextClsoure Algorithm is delayed with linear time. Hence, $|\{E_j, (j=1,2,\ldots,m)\}|=m$ is unchangeable for (G,M,I). For a set $X\subseteq G$, though we can use Definition 8 to compute the value of X relative to E_j, Definition 8 asks both of X and E_j to appear as vectorized forms. Thus, we need to introduce an algorithm to realize vectorization process for a set.

Let (G,M,I) be a context, where $G=\{a_1, a_2,\ldots, a_n\}$, $n=|G|$ and $a_i\neq a_j$, $i\neq j$; $i,j=1,2,\ldots,n$. For $Y=\{g_1, g_2,\ldots, g_m\}\subseteq G$, $m\leq n$. Then we will vectorize Y by Algorithm 1:

Algorithm 1 Vectorization of a subset of G.

Input $G=\{a_1, a_2,\ldots, a_n\}$ and $Y=\{g_1, g_2,\ldots, g_m\}\subseteq G$.

Output The vectoring expression of Y in n-dimensional vector space.

Step 1. $Y=(0,\ldots, 0)$, that is, Y is the zero vector in n-dimensional vector space. $i:=1$, $j:=1$.

Step 2. If $i<n+1$, $j<m+1$, then go to Step 3.

Otherwise, go to Step 5.

Step 3. If $a_i=g_j$, then $Y:=Y+(0,\ldots,0, i, 0, \ldots, 0)$, and $i:=i+1$, $j:=j+1$, go to Step 2,

where $(0,\ldots,0, i, 0, \ldots, 0)$ is the vector in n-dimensional vector space such that every component is 0 except the i-th is i.

Otherwise, go to Step 4.

Step 4. If $a_i\neq g_j$, then $Y:=Y$, and $i:=i+1$, $j:=j$, go to Step 2.

Step 5. Stop.

Remark 4 (1) In Algorithm 1, according to $Y\subseteq G$, we confirm $m\leq n$. This follows that in Algorithm 1, i can not be larger than n, and j is not larger than m. Hence, after finite steps, Algorithm 1 will stop. In other words, we can obtain the vectoring representation of a subset Y in G.

(2) The algorithm complexity analysis is as follows: both i and j are less than or equal to n. When repeating i and j, we only need to compare the size between i and $n+1$, and the size between j and $m+1$. Therefore, the algorithm complexity is $O(n)$.

We will present an example to illustrate Algorithm 1.

Example 2 Let (G,M,I) be a content in which $G=\{a_1, a_2, a_3, a_4, a_5\}$ satisfies $a_i\neq a_j$, $i\neq j$; $i, j=1,2,3,4,5$. Let $Y=\{g_1,g_2\}\subseteq G$, where $g_1=a_2$, $g_2=a_4$. Since $|G|=5$, we say $n=5$. We will establish the expression of Y on the 5-dimensional vector space by Algorithm 1.

According to $i=1$, $j=1$, we put $Y=(0,\ldots,0)$. Since $a_1= a_1$ but $g_1=a_2$, we obtain $a_1\neq g_1$. Then $Y:=Y$. Hence we set $i:=i+1=1+1=2$, $j:=j=1$. Owing to $2<5+1$, $1< 2+1$ and Step 2, we continue the vectorizing process;

According to $i=2$, $j=1$, $Y=(0,\ldots,0)$, we may easily see $a_2=a_2$ and $g_1=a_2$. This follows $a_i=g_j=a_2$. In light of Step 3 in Algorithm 1, we obtain $Y:=Y+(0,2,0,0,0)=(0,2,0,0,0)$. Hence we set $i:=i+1=2+1=3$, $j:=j+1=1+1=2$. Owing to $3<5+1$, $2<2+1$, and Step 2, we continue the process;

According to $i=3$, $j=2$, $Y=(0,2,0,0,0)$, we know $a_3=a_3$ but $g_2=a_4$. So, we obtain $a_3\neq g_2$. By Step 4 in Algorithm 1, put $Y:=Y$. Then $i:=i+1=3+1=4$, $j:=j=2$. Because of $4<5+1$, $2<2+1$ and Step 2, we can go on;

According to i=4, $j=2$, Y=(0,2,0,0,0), since a_4=a_4 and g_2=a_4, this follows a_i=g_j=a_4. Then we use Step 3 and receive Y:=(0,2,0,0,0)+(0,0,0,4,0)=(0,2,0,4,0). Hence we set i:=i+1=4+1=5, j:=j+1=2+1=3. Owing to 4<5+1, 3≥2+1 and Step 2, stop this process. We obtain Y=(0,2,0,4,0).

Let E_j be an extension (j=1,2,…,m). We can use Algorithm 1 to vectorize E_j as $\left(e_{j1}, e_{j2}, \ldots, e_{jn}\right)$. Let $X \subseteq G$. We can also use Algorithm 1 to vectorize X as $\left(x_1, x_2, \ldots, x_n\right)$. Next, based on Definition 8, we will present an algorithm to calculate which E_j is closest to X (j=1,2,…,m).

Algorithm 2 To calculate which extension is closest to a subset X, where $X \subseteq G$.

Input: n-dimensional vector $E_j = \left(e_{j1}, e_{j2}, \ldots, e_{jn}\right)$ (j=1,2,…,m), $X = \left(x_1, x_2, \ldots, x_n\right)$ and r_1=$Adis(E_1,X)$.

Output: The closest vector E to X.

Step 1. i=1, E:=E_1, r:=r_1.

Step 2. If i<m+1, then go to Step 3.
Otherwise, go to Step 5.

Step 3. If $r_i \leq r$, then r:=min{ r_i , r}, E:=E_r, and i:=i+1, go to Step 2.
Otherwise, go to Step 4.

Step 4. If $r< r_i$, then r:=r, E:=E_r, and i:=i+1, go to Step 2.

Step 5. Stop.

Remark 5 (1) In Algorithm 2, according to $|G|<\infty$, we confirm $|m|<\infty$. This follows that i in Algorithm 2 can not be larger than m. Hence, after finite steps, Algorithm 2 will stop. In other words, we can calculate the closest vector E to X.

(2) The complexity of Algorithm 2 is analyzed as follows: i is less than or equal to m, and when repeating i, we just need to compare the size between i and m+1; besides, $m \leq |G|$; so the algorithm complexity is $O(m)$.

(3) The result of Algorithm 2 is a vector E_r which is closest to X. Actually, we can apply the following way to express E_r as set language.

Let E_r=($e_{r1}, e_{r2}, \ldots, e_{rn}$). If e_{rj} =0, delete it; if $e_{rj} \neq 0$, remain it. Then we can express E_r as { $e_{t1}, e_{t2}, \ldots, e_{tk}$ }, where $e_{tj} \in$ { $e_{r1}, e_{r2}, \ldots, e_{rn}$ } and $e_{tj} \neq 0$, (j=1,2,...,k; t_1<t_2<...<t_k).

Therefore, the corresponding extension{E_r} is found out.

Theorem 1 When Algorithm 2 stops, the output E must be the closest vector to X.

Proof For a context (G,M,I), and {E_j: j=1,2,…,m} be all extensions of (G,M,I). The repetition i in Algorithm 2 satisfies $i \leq m$, therefore this algorithm can be stop after finite steps.

For $X \subseteq G$, from the Definition 8, we know that the smaller value of r_i=$Adis(E_i, X)$, the accurate of the result. In the processing, each repetition is compared the values of r and r_i, and set r:=min{ r_i , r}, output E:=E_r. Thus the output E must be the closest vector to X.

We will present an example to illustrate Algorithm 2.

Example 3 Let (G,M,I) as Table 1, where the meaning of each attribute is different document and each object is a keyword. Let X={3,5,6}$\subseteq G$. Then, we can use Algorithm 1 and Algorithm 2 to find out the closest extension to X.

Table 1: A context

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
1.d_1	×	×					×		
2.d_2	×	×					×	×	
3.d_3	×	×	×				×	×	
4.d_4	×		×				×	×	×
5.d_5	×	×		×		×			
6.d_6	×	×	×	×		×			
7.d_7	×		×	×	×				
8.d_8	×		×	×		×			

For this context, we can use the NextClsoure Algorithm to obtain the extensions of (G,M,I):

$$E_1=\{1,2,3,4,5,6,7,8\},\ E_2=\{1,2,3,4\},\ E_3=\{1,2,3,5,6\},\ E_4=\{3,4,6,7,8\},\ E_5=\{5,6,7,8\},\ E_6=\{1,2,3\},\ E_7=\{2,3,4\},\ E_8=\{5,6,8\},\ E_9=\{6,7,8\},\ E_{10}=\{2,3\},\ E_{11}=\{3,4\},\ E_{12}=\{5,6\},\ E_{13}=\{3,6\},\ E_{14}=\{6,8\},\ E_{15}=\{4\},\ E_{16}=\{3\},\ E_{17}=\{6\},\ E_{18}=\{7\},\ E_{19}=\{\phi\}.$$

Through Algorithm 1, we obtain:

$$E_1=(1,2,3,4,5,6,7,8),\quad E_2=(1,2,3,4,0,0,0,0),\quad E_3=(1,2,3,0,5,6,0,0),\quad E_4=(0,0,3,4,0,6,7,8),\ E_5=(0,0,0,0,5,6,7,8),\ E_6=(1,2,3,0,0,0,0,0),\ E_7=(0,2,3,4,0,0,0,0),\ E_8=(0,0,0,0,5,6,0,8),\ E_9=(0,0,0,0,0,6,7,8),\ E_{10}=(0,2,3,0,0,0,0,0),\ E_{11}=(0,0,3,4,0,0,0,0),\ E_{12}=(0,0,0,0,5,6,0,0),\ E_{13}=(0,0,3,0,0,6,0,0),\ E_{14}=(0,0,0,0,0,6,0,8),\ E_{15}=(0,0,0,4,0,0,0,0),\ E_{16}=(0,0,3,0,0,0,0,0),\ E_{17}=(0,0,0,0,0,6,0,0),\ E_{18}=(0,0,0,0,0,0,7,0),\ E_{19}=\phi.$$

Let $X=\{3,5,6\}\subseteq G$. We use Algorithm 1 to obtain $X=(0,0,3,0,5,6,0,0)$.
Next, we use Algorithm 2 to compute the closest extension E to X:
Since we get all the extensions above, we can easily know m=19.
According to i=1, $E:=E_1$=(1,2,3,4,5,6,7,8), $r:=r_1=Adis(E_1,X)\approx0.811$. We set $i:=i+1=1+1=2$. Then from Definition 8, $r_2=Adis(E_2,X)\approx1.652$. Since $r<r_2$, in light of Step 4 in Algorithm 2, and then $r:=r$, $E:=E_1$=(1,2,3,4,5,6,7,8). We set $i:=i+1=2+1=3$. Because of 3<19+1 and Step 2, we continue the process;
According to i=3, $E:=E_1$=(1,2,3,4,5,6,7,8), from Definition 8, we know $r_3=Adis(E_3,X)\approx0.258$. Since $r_3\leq r$, by Step 3 we put $r:=\min(r_3, r)= r_3$, $E:=E_3$=(1,2,3,0,5,6,0,0). Let $i:=i+1=3+1=4$. Owing to 4<19+1 and Step 2, we go on;
According to i=4, $E:=E_3$=(1,2,3,0,5,6,0,0), $r_4=Adis(E_4,X)\approx0.947$, we easily obtain $r<r_4$. By Step 4, we know $r:=r$, $E:=E_3$=(1,2,3,0,5,6,0,0). Let $i:=i+1=4+1=5$. For 5<19+1, we can continue.
Repeating the above process, we finally get that E_3=(1,2,3,0,5,6,0,0) is the result. And by Remark 5 (3), we know the corresponding extension E_3={1,2,3,5,6} is the needed result. Therefore the extension E_3 is closest to X.

CONCLUSION

Nowadays, the application of classification rules has been widely increasing. The thought of classification has been extensively used in medical diagnosis, weather forecasting, market analysis, books retrieval and so on. In this paper, we have presented a definition of approximation distance based on inclusion degree and similarity theory. In addition, using the equivalent relationship between concepts in the concept lattice and the equivalence classes in rough set, we have proposed an algorithm to classify the undefinable object set to a closest extension.
We may indicate that approximate distance operator is not mature, since this operator can only calculate the quantized sets. Hence, we need to research on the method of approximate distance to calculate the non-quantization sets in the further. How to simplify the calculation is also the goal of future research.

ACKNOWLEDGMENTS

This paper is granted by NSF of China (61073121) and NSF of Hebei province (A2013201119, A2014201033).

REFERENCES

Chmielewski MR and Grzymala-Busse JW (1996). Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning* **15** 319-331.
Clobanu G and Vaideanu C (2014). Similarity relation in fuzzy attribute-oriented concept lattices. *Fuzzy Set and Systems* **6716** 1-21.
Dai J, Wang W, Xu Q and Tian H (2012). Uncertainty measurement for interval-valued decision systems based on extended conditional entropy. *Knowledge-Based Systems* **27** 443–450.

Davey BA and Priestley HA (2002). *Introduction to Lattices and Order*, Second edition (Cambridge: Cambridge University Press) 66-69.
Ganter B and Wille R (1999). *Formal Concept Analysis: Mathematical Foundations* (New York: Springer).
Hu XG, Xue F, Zhang YH and Zhang J (2009). Attribute reduction methods of decision table based on concept lattice. *Pattern Recognition and Artificial Intelligence* **22**(4) 624-629.
Huang HT, Cheng QJ and Wan QS (2015). FCA concept similarity computation based on semantic information content. *Application Research of Computers* **32**(3) 731-735.
Huang HT, Wu ZL and Wan QS (2015). FCA concept similarity computation based on bounded transitive similarity graph. *Computer Science* **42**(1) 285-289.
Mao H and Kang R (2015). Variable precision rough set approximations in concept lattice. *Journal of Progressive Research in Mathematics* **2**(1) 47-56.
Miao D, Gao C, Zhang N and Zhang Z (2011). Diverse reduce subspaces based co-training for partially labeled data. *International Journal of Approximate Reasoning* **52**(8) 1103-1117.
Pawlak Z (1982). Rough sets. *International Journal of Computer and Information Sciences* **11** 341-356.
Thomas T, Richard C, Peter B and Peter E (2005). A Survey of Formal Concept Analysis Support for Software Engineering Activities. In: *Formal Concept Analysis*, edited by Ganter B, Stumme G and Wille R (New York: Springer) 250-271.
Wille R (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. In: *Ordered Sets*, edited by Rivaled I (Dordrecht Boston) 445-470.
Xu HS and Zhang RL (2013). Application of improved variable precision rough set in concept lattice construction. *Computer Science* **40**(3) 271-275.
Yao YY and Chen YH (2006). Rough set approximations in formal concept analysis. In: *Transactions on Rough Sets V*, edited by Peters JF and Skowion A (New York: Springer) 285-305.
Zhang W and Leung Y (1996). *The Uncertainty Reasoning Principless* (Xian: Xian Jiaotong University Press).
Zhi HL, Zhi DJ and Liu ZT (2008). Concept similarity based on concept lattice. *Computer Science* **35**(9) 156-157.
Zhu YQ and Zhu W (2014). A variable precision covering-based rough set model based on functions. *The Scientific World Journal* **210129** 1-5.
Ziarko W (1993). Variable precision rough set model. *Journal of Computer and System Sciences* **46** 39-59