Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.3, 655–674

CONCEPT APPROXIMATIONS BASED ON ROUGH SETS AND SIMILARITY MEASURES†

JAMIL SAQUER*, JITENDER S. DEOGUN**

The formal concept analysis gives a mathematical definition of a formal concept. However, in many real-life applications, the problem under investigation cannot be described by formal concepts. Such concepts are called the non-definable concepts (Saquer and Deogun, 2000a). The process of finding formal concepts that best describe non-definable concepts is called the concept approximation. In this paper, we present two different approaches to the concept approximation. The first approach is based on rough set theory while the other is based on a similarity measure. We present algorithms for the two approaches.

Keywords: formal concept analysis, similarity measures, rough sets, concept approximation

1. Introduction

The formal concept analysis (FCA) is a mathematical framework, developed by Wille and his colleagues at Darmstadt University, which is useful for representation and an analysis of data (Wille, 1992; 1982; 1989). The pair (A, B) of a set of objects and a set of features where B is the maximal set of features common to the objects in A and A is the maximal set of objects that possess all the features in B is called the formal concept. Using the framework of FCA, formal concepts are structured in the form of a lattice called the concept lattice. The concept lattice is a useful tool for knowledge representation, and knowledge discovery (Godin and Missaoui, 1994; Saquer and Deogun, 2000b). The formal concept analysis has also been applied in the area of conceptual modeling that deals with the acquisition, representation, and organization of knowledge (Kangassalo, 1992). Several concept learning methods have been implemented in (Carpineto and Romano, 1996; Godin and Missaoui, 1994; Ho, 1995) using ideas from the formal concept analysis.

Not every pair of a set of objects and a set of features defines a concept (Wille, 1982). Furthermore, we might be faced with a situation where we only have a single set of features (or a single set of objects) and need to find the best concept that approximates these features (or objects). For example, when a physician diagnoses

† This research was supported in part by the NSF EPSCoR under Grant No. EPS-0091900, and NSF Digital Government Grant No. EIA–0091530.

* Computer Science Department, Southwest Missouri State University, Springfield, MO 65804, USA, e-mail: jms481f@smsu.edu

** Computer Science and Engineering Department, University of Nebraska, Lincoln, NE 68588, USA, e-mail: deogun@cse.unl.edu

a patient, he finds a disease whose symptoms are the closest to the symptoms that the patient has. In this case we can think of the symptoms as features and the diseases as objects. Another example is in the area of information retrieval, where a user's query can be understood as a set of features and the answer to the query may be understood as the set of objects that possess these features. The answer to the query may contain documents that do not comprise all the terms in the query. This means that the pair of documents and terms (or objects and features) here cannot be described as a formal concept. However, in an information retrieval system we retrieve information which is usually presented as documents that contain part of the answer and show the documents in order of their relevance to the user's query. It is therefore of fundamental importance to be able to find concept approximations regardless of how little information is available.

The notion of concept approximation was first introduced in (Kent, 1994; 1996) and further investigated in (Saquer and Deogun, 1999; 2000a). All these approaches use rough sets as the underlying approximation model. In this paper we present two different approaches to concept approximation. In the first approach, which constitutes an extension of our work (Saquer and Deogun, 1999) and is based on rough set theory, we approximate a non-definable concept by two formal concepts that represent its lower and upper approximations. The second approach is based on a similarity between non-definable and formal concepts. We define a similarity measure to find a formal concept that is the closest to a non-definable concept. For each approach we show how a set of objects, a set of features, and the pair of a set of objects and a set of features can be approximated by formal concepts.

The organization of this paper is as follows. In Section 2, we give an overview of FCA results that are needed for this paper. In Section 3, we present a new rough set theory-based approach to concept approximation. In Section 4, we present a similarity-based approach to concept approximation. Finally, a conclusion is drawn in Section 5.

2. Background

Relationships between objects and features in the FCA are given in a *context* which is defined as the triple (G, M, I), where G and M are sets of objects and features (or attributes), respectively, and $I \subseteq G \times M$. An example of a context is given in Table 1, where "$\times$" is placed in the i-th row and the j-th column to indicate that the object in row i possesses the feature in column j. If object g possesses feature m, then we write $(g, m) \in I$ or gIm. The set of all the features common to a set of objects A is denoted by $\beta(A)$ and is defined as $\{m \in M \mid gIm \ \forall g \in A\}$. Similarly, the set of objects possessing all the features in a set $B \subseteq M$ is denoted by $\alpha(B)$ and given by $\{g \in G \mid gIm \ \forall m \in B\}$. The operators α and β satisfy the assertions given in the following lemma.

Lemma 1. (Wille, 1982) *Let* (G, M, I) *be a context. Then the following assertions hold:*

1. $A_1 \subseteq A_2$ *implies* $\beta(A_1) \supseteq \beta(A_2)$ *for every* $A_1, A_2 \subseteq G$, *and* $B_1 \subseteq B_2$ *implies* $\alpha(B_1) \supseteq \alpha(B_2)$ *for every* $B_1, B_2 \subseteq M$.

2. $A \subseteq \alpha(\beta(A))$ *and* $\beta(A) = \beta(\alpha(\beta(A)))$ *for all* $A \subseteq G$*, and*
$B \subseteq \beta(\alpha(B))$ *and* $\alpha(B) = \alpha(\beta(\alpha(B)))$ *for all* $B \in M$*.*

Table 1. Example of a context.

	a	b	c	d	e	f	h	i	j	k	l	x
1				×								
2				×		×	×					
3						×	×					
4	×	×	×	×	×	×	×				×	
5			×	×	×	×	×	×				×
6					×	×	×	×	×	×	×	×
7						×	×	×	×			×
8			×	×	×	×	×					
9			×	×		×					×	
10			×	×	×	×					×	
11						×	×					
12			×	×	×	×	×	×				×
13			×	×	×	×	×	×				×

A *formal concept* in the context (G, M, I) is defined as a pair (A, B) where $A \subseteq G$, $B \subseteq M$, $\beta(A) = B$ and $\alpha(B) = A$. A is called the *extent* of the formal concept and B is called its *intent*. For example, the pair (A, B) where $A = \{4, 9, 10\}$ and $B = \{c, d, f, l\}$ is a formal concept. On the other hand, the pair (A, B) where $A = \{2, 3, 4\}$ and $B = \{f, h\}$ is not a formal concept because $\alpha(B) \neq A$. A pair (A, B) where $A \subseteq G$ and $B \subseteq M$ which is not a formal concept is called a *non-definable concept* (Saquer and Deogun, 2000a).

Given a context (G, M, I), neither is every subset $A \subseteq G$ an extent nor every subset $B \subseteq M$ an intent. From Lemma 1, it follows that $A \subseteq \alpha(\beta(A))$ for any $A \subseteq G$, and $B \subseteq \beta(\alpha(B))$ for any $B \subseteq M$. A set of objects A is called *feasible* if $A = \alpha(\beta(A))$. Similarly, a set of features B is feasible if $B = \beta(\alpha(B))$ (Saquer and Deogun, 1999). A set of objects or features which is not feasible is called *non-feasible*. If A is feasible, then clearly $(A, \beta(A))$ is a concept. Similarly, if B is feasible, then $(\alpha(B), B)$ is a concept.

The Fundamental Theorem of FCA states that the set of all the formal concepts on a given context with the ordering $(A_1, B_1) \leq (A_2, B_2)$ iff $A_1 \subseteq A_2$ is a complete lattice called the *concept lattice*, in which the infima and suprema are given by (Ganter and Wille, 1999; Wille, 1982):

$$\bigwedge_{j \in J} (A_j, B_j) = \Big(\bigcap_{j \in J} A_j, \beta\big(\alpha(\bigcup_{j \in J} B_j)\big) \Big) = \Big(\bigcap_{j \in J} A_j, \beta\Big(\bigcap_{j \in J} A_j \Big) \Big),$$

$$\bigvee_{j \in J} (A_j, B_j) = \Big(\alpha\Big(\beta(\bigcup_{j \in J} A_j)\Big), \bigcap_{j \in J} B_j \Big) = \Big(\alpha\Big(\bigcap_{j \in J} B_j \Big), \bigcap_{j \in J} B_j \Big).$$

The concept lattice of the context given in Table 1 is shown in Fig. 1, where the concepts are labeled using *reduced labeling* (Ganter and Wille, 1999). The extent of the concept C in Fig. 1 consists of the objects at C and the objects at the concepts that can be reached from C going downward and following the descending paths towards the bottom concept C_1. Similarly, the intent of C consists of the features at C and the features at the concepts that can be reached from C going upwards and following the ascending paths to the top concept C_{23}. The extent and intent of each concept in Fig. 1 are also given in Table 4.

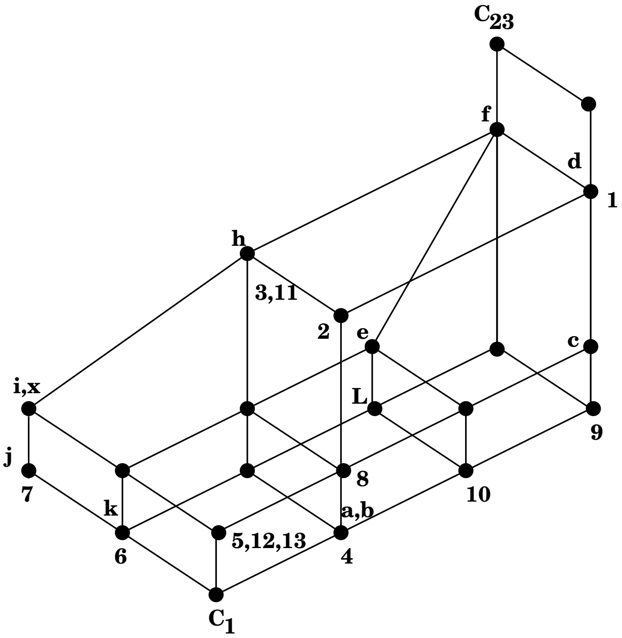

Fig. 1. Concept lattice for the context given in Table 1 with reduced labeling.

3. Rough Set Theory Approach to Concept Approximation

In this section we present a rough set theory approach to approximating concepts. This approach has many novel features and represents a significant improvement over the existing approach given in (Kent, 1994; 1996) for many reasons. First, the relations we use in the approximation are defined in a way that assures that the same answer is always given, while in (Kent, 1994; 1996) the answer depends on the equivalence relation used which is provided by an expert. Second, the equivalence relations we use are defined automatically by the system with no intervention from the user, which makes our approximation method completely automatic and user-independent. Third, we use both the set of objects and the set of features for approximating non-definable concepts. However, in (Kent, 1994; 1996) only the set of features is used, which results in the fact that all the pairs with same set of objects have the same approximation regardless of the equivalence relation used. Fourth, we use the context directly for concept approximation, while in (Kent, 1994; 1996) the context is first approximated by lower and upper approximations, which are then used for the concept approximation. Fifth, our approach is general enough to find formal concepts that approximate a single set of objects or a single set of features.

In the remainder of this paper, when we use the term "approximating a set of objects (features)," we really mean finding formal concepts whose extent (intent) approximates the given set of objects (features). We also use the term "approximating a pair or a non-definable concept (A, B)" while we really mean finding formal concepts whose extent approximates A and whose intent approximates B.

3.1. Approximating a Set of Objects

Given a set of objects $A \subseteq G$, we are interested in finding a formal concept whose extent approximates A. We say that such a concept approximates A. We have the following cases:

Case 1: A is feasible. Clearly, $(A, \beta(A))$ is a formal concept. Therefore, $(A, \beta(A))$ is the best approximation.

Case 2: A is not feasible. It is not easy to find formal concepts that approximate a non-feasible set A. Our approach is to think of A as a rough set. We first find a pair of definable sets $\underline{A}$ and $\overline{A}$ that represent the lower and upper approximations of A, respectively. We then use $\underline{A}$ and $\overline{A}$ for finding two formal concepts that approximate A.

Let $gI = \{m \in M \mid gIm\}$ denote the set of all the features that are possessed by the object g. Define the relation R on G as follows:

$$g_1 R g_2 \ \text{ iff } \ g_1 I = g_2 I, \quad \text{where } \ g_1, g_2 \in G,$$

i.e. two objects are related if and only if they possess the same set of features. Clearly, R is an equivalence relation on G. Therefore, it induces a partition on G. Let G/R be the set of all equivalence classes induced by R on G (also known as the quotient set). The equivalence classes of G/R are called the *elementary sets*. Any finite union of elementary sets is called a *definable set* (Pawlak, 1982). The next proposition shows that two objects belong to the same equivalence class if and only if they have the same object concept where the object concept of an object g is the smallest formal concept containing g and is given by $(\alpha(\beta(g)), \beta(g))$ (Ganter and Wille, 1999).

Proposition 1. *For any two objects g_1 and g_2, $g_1 R g_2$ if and only if the object concept of g_1 equals the object concept of g_2.*

Proof.

($\Longrightarrow$)

$$\begin{aligned} g_1 R g_2 &\Longrightarrow g_1 I = g_2 I \\ &\Longrightarrow \beta(g_1) = \beta(g_2) \\ &\Longrightarrow \alpha(\beta(g_1)) = \alpha(\beta(g_2)). \end{aligned}$$

Therefore $(\alpha(\beta(g_1)), \beta(g_1)) = (\alpha(\beta(g_2)), \beta(g_2))$.

($\Longleftarrow$)

$$\begin{aligned} \alpha(\beta(g_1)) = \alpha(\beta(g_2)) &\Longrightarrow \beta(\alpha(\beta(g_1))) = \beta(\alpha(\beta(g_2))) \\ &\Longrightarrow \beta(g_1) = \beta(g_2) \quad \text{(by Lemma 1)} \\ &\Longrightarrow g_1 I = g_2 I \\ &\Longrightarrow g_1 R g_2 \end{aligned}$$

■

Example 1. Consider the context given in Table 1 and its concept lattice given in Fig. 1. To find the equivalence classes induced by R on G, we need to find the features that are possessed by each object $g \in G$, which is given in Table 2.

Table 2. gI for every $g \in G$.

Object g	Set of features possessed by g
$1I$	$\{d\}$
$2I$	$\{d, f, h\}$
$3I$	$\{f, h\}$
$4I$	$\{a, b, c, d, e, f, h, l\}$
$5I$	$\{c, d, e, f, h, i, x\}$
$6I$	$\{e, f, h, i, j, k, l, x\}$
$7I$	$\{f, h, i, j, x\}$
$8I$	$\{c, d, e, f, h\}$
$9I$	$\{c, d, f, l\}$
$10I$	$\{c, d, e, f, l\}$
$11I$	$\{f, h\}$
$12I$	$\{c, d, e, f, h, i, x\}$
$13I$	$\{c, d, e, f, h, i, x\}$

We notice that $3I = 11I$ and $5I = 12I = 13I$. Therefore, $G/R = \{[1], [2], [3, 11], [4], [5, 12, 13], [6], [7], [8], [9], [10]\}$. Table 3 lists the object concepts to which the members of the equivalence classes of G/R belong. We notice that Objects 3 and 11 have the identical object concept and the same is true for Objects 5, 12 and 13.

Table 3. Object concepts.

Object	Concept
1	$(\{1,2,4,5,8,9,10,12,13\},\{d\})$
2	$(\{2,4,5,8,12,13\},\{d,f,h\})$
3	$(\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$
4	$(\{4\},\{a,b,c,d,e,f,h,l\})$
5	$(\{5,12,13\},\{c,d,e,f,h,i,x\})$
6	$(\{6\},\{e,f,h,i,j,k,l,x\})$
7	$(\{6,7\},\{f,h,i,j,x\})$
8	$(\{4,5,8,12,13\},\{c,d,e,f,h\})$
9	$(\{4,9,10\},\{c,d,f,l\})$
10	$(\{4,10\},\{c,d,e,f,l\})$
11	$(\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$
12	$(\{5,12,13\},\{c,d,e,f,h,i,x\})$
13	$(\{5,12,13\},\{c,d,e,f,h,i,x\})$

Define the lower and upper approximations of $A \subseteq G$ with respect to R as follows:

$$\underline{A} = \{g \in G \mid [g] \subseteq A\} = \bigcup \{X \in G/R \mid X \subseteq A\},$$

$$\overline{A} = \{g \in G \mid [g] \cap A \neq \emptyset\} = \bigcup \{X \in G/R \mid X \cap A \neq \emptyset\}.$$

Now, we can find two formal concepts that approximate A. The lower and upper concept approximations are respectively given by $(\alpha(\beta(\underline{A})), \beta(\underline{A}))$ and $(\alpha(\beta(\overline{A})), \beta(\overline{A}))$. Notice that, if A is definable, we have $\underline{A} = A = \overline{A}$. In this case the lower and upper approximations of A coincide. Furthermore, since $\underline{A} = \overline{A} = A \subseteq \alpha(\beta(A)) = \alpha(\beta(\underline{A})) = \alpha(\beta(\overline{A}))$, the approximation we get is an upper approximation. The problem of finding a lower approximation for a definable set will be investigated in a forthcoming paper. ♦

The pseudo-code for approximating a set of objects is given in Algorithm 1. The input is a set A of objects and the quotient set G/R. The output is the concept approximation for A. First, in Line 1 a test is made to determine if the set A is feasible, in which case the formal concept $((A, \beta(A))$ is returned in Line 2. The evaluation of $\beta(A) = \{m \in M \mid gIm \forall g \in A\}$ requires time of the order $O(|M||A|)$. Similarly, evaluating $\alpha(*)$ requires time $O(|G||*|)$. Therefore, executing Lines 1 and 2 and thus approximating a feasible set of objects requires time $O(|G||M||A||\beta(A)|)$. If A is non-feasible, the lower and upper approximating sets of A are found in Lines 4 and 5, respectively, which requires time $O(|G/R|)$. A test is made in Line 6 to determine if A is definable, in which case an upper approximation is returned in Line 7. If A is not definable, the lower and upper concept approximations of A are evaluated in Lines 9 and 10, and returned in Line 11. The time complexities of Lines 9 and 10 are $(|G||M||\underline{A}||\beta(\underline{A})|$ and $(|G||M||\overline{A}||\beta(\overline{A})|$, respectively. Therefore, approximating a non-feasible set of objects requires time $O(|G/R| + |G||M|(|\underline{A}||\beta(\underline{A})| + |\overline{A}||\beta(\overline{A})|)) = O(|G||M|(|\underline{A}||\beta(\underline{A})| + |\overline{A}||\beta(\overline{A})|))$ because $|G/R|$ is bounded from above by $|G|$. We also notice that the size of any set of objects is also bounded by $|G|$, and that of any set of features is bounded by $|B|$. Therefore, the time complexity of Algorithm 1 is $O\left(|G|^2|M|^2\right)$.

Algorithm 1: `Approximate a set` A `of objects`

`if` $(\alpha(\beta(A)) == A)$ `//i.e.` A `is feasible`
    `Answer` $\leftarrow (A, \beta(A))$
`else`
    $\underline{A} = \bigcup\{X \in G/R \mid X \subseteq A\}$
    $\overline{A} = \bigcup\{X \in G/R \mid X \cap A \neq \emptyset\}$
    `if` $(\underline{A} == \overline{A})$ `//i.e.` A `is definable`
        `Answer` $\leftarrow \left(\alpha(\beta(\overline{A})), \beta(\overline{A})\right)$
    `else` $\{$
        `Lower` $\leftarrow (\alpha(\beta(\underline{A})), \beta(\underline{A}))$
        `Upper` $\leftarrow \left(\alpha(\beta(\overline{A})), \beta(\overline{A})\right)$
        `Answer` $\leftarrow$ `Lower and Upper` $\}$
`end if`

Example 2. In this example we find formal concepts that approximate the sets $A_1 = \{3,4,5\}$, $A_2 = \{2,3\}$, $A_3 = \{4,9\}$, $A_4 = \{4,5,6\}$ and $A_5 = \{5,6,8\}$. $\underline{A_1} = \{4\}$, $\overline{A_1} = \{3,4,5,11,12,13\}$, $\beta(\underline{A_1}) = \{a,b,c,d,e,f,h,l\}$, $\beta(\overline{A_1}) = \{f,h\}$, $\alpha(\beta(\underline{A_1})) = \{4\}$, and $\alpha(\beta(\overline{A_1})) = \{2,3,4,5,6,7,8,11,12,13\}$. Therefore, the lower and upper approximations for A_1 are $(\{4\}, \{a,b,c,d,e,f,h,l\})$ and $(\{2,3,4,5,6,7,8,11,12,13\}, \{f,h\})$, respectively. $\underline{A_2} = \{2\}$, $\overline{A_2} = \{2,3,11\}$, $\beta(\underline{A_2}) = \{d,f,h\}$, $\beta(\overline{A_2}) = \{f,h\}$, $\alpha(\beta(\underline{A_2})) = \{2,4,5,8,12,13\}$, and $\alpha(\beta(\overline{A_2})) = \{2,3,4,5,6,7,8,11,12,13\}$. Therefore, the lower and upper approximations for A_2 are given by $(\{2,4,5,8,12,13\}, \{d,f,h\})$ and $(\{2,3,4,5,6,7,8,11,12,13\}, \{f,h\})$. $\underline{A_3} = \overline{A_3} = A = \{4,9\}$. $\beta(A_3) = \{c,d,f,l\}$, and $\alpha(\beta(A_3)) = \{4,9,10\}$. So, the upper approximation of A_3 is given by $(\{4,9,10\}, \{c,d,f,l\})$. $\underline{A_4} = \{4,6\}$, $\overline{A_4} = \{4,5,6,12,13\}$, $\beta(\underline{A_4}) = \{e,f,h,l\}$, $\beta(\overline{A_4}) = \{e,f,h\}$, $\alpha(\beta(\underline{A_4})) = \{4,6\}$, and $\alpha(\beta(\overline{A_4})) = \{4,5,6,8,12,13\}$. Therefore, the lower and upper approximations for A_4 are given by the formal concepts $(\{4,6\}, \{e,f,h,l\})$ and $(\{4,5,6,8,12,13\}, \{e,f,h\})$, respectively. $\underline{A_5} = \{6,8\}$, $\overline{A_5} = \{5,6,8,12,13\}$, $\beta(\underline{A_4}) = \{e,f,h\}$, $\beta(\overline{A_4}) = \{e,f,h\}$, $\alpha(\beta(\underline{A_4})) = \{4,5,6,8,12,13\}$, and $\alpha(\beta(\overline{A_4})) = \{4,5,6,8,12,13\}$. Therefore, the lower and upper approximations for A_5 are equal and are given by $(\{4,5,6,8,12,13\}, \{e,f,h\})$. ♦

3.2. Approximating a Set of Features

Approximating a set of features $B \subseteq M$ works analogously to approximating a set of objects, and therefore we omit many details in the following discussion.

Case 1: *B* is feasible. $(\alpha(B), B)$ is a formal concept. Therefore $(\alpha(B), B)$ is the best approximation.

Case 2: *B* is not feasible. Let $Im = \{g \in G \mid gIm\}$ be the set of all objects that possess the attribute m. Define a relation R' on M as follows:

$$m_1 R' m_2 \text{ iff } Im_1 = Im_2 \text{ where } m_1, m_2 \in M,$$

i.e. two features are related if and only if they are possessed by the same set of objects. The relation R' is an equivalence relation on M. Let M/R' denote the set of all the equivalence classes induced by R' on M. The equivalence classes of M/R' are called the *elementary sets* and any finite union of elementary sets is called a *definable set*. The feature concept of $m \in M$ is the largest concept containing the feature m and is given by $(\alpha(m), \beta(\alpha(m)))$ (Ganter and Wille, 1999). The following proposition gives a relationship between R' and features concepts. The proof is similar to that of Proposition 1.

Proposition 2. *For any two features m_1 and m_2, $m_1 R' m_2$ if and only if the feature concept of m_1 equals the feature concept of m_2.*

The lower and upper approximations of $B \subseteq M$ are respectively defined as follows:

$$\underline{B} = \{m \in M \mid [m] \subseteq B\} = \bigcup \{Y \in M/R' \mid Y \subseteq B\},$$

$$\overline{B} = \{m \in M \mid [m] \cap B \neq \emptyset\} = \bigcup \{Y \in M/R' \mid Y \cap B \neq \emptyset\}.$$

The lower and upper concept approximations of a non-feasible set of features $B \subseteq M$ are given by the formal concepts $(\alpha(\overline{B}), \beta(\alpha(\overline{B})))$ and $(\alpha(\underline{B}), \beta(\alpha(\underline{B})))$, respectively. Notice that $(\alpha(\overline{B}), \beta(\alpha(\overline{B})))$ is a subconcept of $(\alpha(\underline{B}), \beta(\alpha(\underline{B})))$. To prove this, we notice that $\underline{B} \subseteq B \subseteq \overline{B}$, which implies $\alpha(\overline{B}) \subseteq \alpha(B) \subseteq \alpha(\underline{B})$. Therefore, the extent of the lower approximating concept of B is a subset of the extent of the upper approximating concept of B. However, when B is definable, we have $\underline{B} = B = \overline{B}$. Furthermore, $B \subseteq \beta(\alpha(B)) = \beta(\alpha(\underline{B})) = \beta(\alpha(\overline{B}))$, so $B \subseteq \beta(\alpha(\underline{B}))$. This means that both the approximation formulas give a lower approximation for B.

The pseudo-code for approximating a set of features is given in Algorithm 2. It takes as input a set of features B and the quotient set M/R', and it outputs the concept approximation for B. Algorithm 2 is similar to Algorithm 1. The time complexity for Algorithm 2 is $O(|G||M||B||\alpha(B)|)$ when B is feasible, and $O(|G||M|(|\underline{B}||\alpha(\underline{B})| + |\overline{B}||\alpha(\overline{B})|))$ when B is non-feasible. But since the size of any set of objects is bounded by $|G|$ and that of any set of features by $|B|$, the time complexity for Algorithm 2 is $O(|G|^2|M|^2)$.

Algorithm 2: `Approximate a set` B `of features`

`if` $(\beta(\alpha(B)) == B)$ `//i.e.` B `is feasible`
    `Answer` $\leftarrow$ $(\alpha(B), B)$
`else`
    $\underline{B} = \bigcup\{Y \in M/R' \mid Y \subseteq B\}$
    $\overline{B} = \bigcup\{Y \in M/R' \mid Y \cap B \neq \emptyset\}$
    `if` $(\underline{B} == \overline{B})$ `//i.e.` B `is definable`
        `Answer` $\leftarrow$ $\big(\alpha(\overline{B}), \beta(\alpha(\overline{B}))\big)$
    `else` $\{$
        `Lower` $\leftarrow$ $\big(\alpha(\overline{B}), \beta(\alpha(\overline{B}))\big)$
        `Upper` $\leftarrow$ $(\alpha(\underline{B}), \beta(\alpha(\underline{B})))$
        `Answer` $\leftarrow$ `Lower and Upper` $\}$
`end if`

Example 3. Consider the context given in Table 1 and set $B = \{f, h, i\}$. We want to find formal concepts that approximate B. We have $M/R' = \{\{a, b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{h\}, \{i, x\}, \{k\}, \{l\}\}$, $\underline{B} = \{f, h\}$, $\overline{B} = \{f, h, i, x\}$, $\alpha(\underline{B}) = \{2, 3, 4, 5, 6, 7, 8, 11, 12, 13\}$, $\alpha(\overline{B}) = \{5, 6, 7, 12, 13\}$, $\beta(\alpha(\underline{B})) = \{f, h\}$, and $\beta(\alpha(\overline{B})) = \{f, h, i, x\}$. Therefore the lower and upper approximations of $\{f, h, i\}$ are given by the formal concepts $(\{5, 6, 7, 12, 13\}, \{f, h, i, x\})$ and $(\{2, 3, 4, 5, 6, 7, 8, 11, 12, 13\}, \{f, h\})$, respectively. ♦

3.3. Approximating a Non-Definable Concept

Given a pair (A, B) where $A \subseteq G$ and $B \subseteq M$, we want to find a formal concept(s) whose extent approximates A and whose intent approximates B. Such concepts are said to approximate the pair (A, B). The following four cases need to be considered:

1. Both A and B are feasible.
2. A is feasible and B is not.
3. B is feasible and A is not.
4. Both A and B are non-feasible.

Case 1: Both A and B are feasible. We have the following two subcases.

Case 1.1: $\beta(A) = B$. If $\beta(A) = B$, then $\alpha(B)$ must equal A because both A and B are feasible. Proposition 3 below establishes a similar result. Thus, the pair (A, B) is a formal concept and no approximation is needed.

It may be noted that Proposition 3 results in a more efficient way for testing if both A and B are feasible, $\beta(A) = B$, and $\alpha(B) = A$. This proposition is used in developing an approach for approximating non-definable concepts, which is employed in Algorithm 3.

Proposition 3. *Let* $A \subseteq G$ *and* $B \subseteq M$. *If* A *is feasible and* $\beta(A) = B$, *then* B *is feasible and* $\alpha(B) = A$.

Proof.

$$\begin{aligned} A \text{ feasible} &\implies \alpha(\beta(A)) = A \\ &\implies \alpha(B) = A \quad (\text{using } \beta(A) = B \text{ which is given}) \\ &\implies \beta(\alpha(B)) = \beta(A) \quad (\text{applying } \beta \text{ to both sides}) \\ &\implies \beta(\alpha(B)) = B \quad (\text{using } \beta(A) = B) \\ &\implies B \text{ is feasible.} \end{aligned}$$

Therefore, B is feasible and $\alpha(B) = A$. ■

Case 1.2: $\beta(A) \neq B$. If $\beta(A) \neq B$ (and thus $\alpha(B) \neq A$), let $\beta(A) = A'$ and $\alpha(B) = B'$. Since both A and B are feasible, we see that both (A, A') and (B', B) are formal concepts in (G, M, I). Lower and upper approximations for (A, B) are found by using the Fundamental Theorem of FCA and finding the infima and suprema for (A, A') and (B', B). This results in the following formulas for lower and upper approximations:

$$\underline{(A,B)} = (A \cap B', \beta(A \cap B')), \quad \overline{(A,B)} = (\alpha(A' \cap B), A' \cap B).$$

Example 4. Consider the feasible sets $A = \{4, 5, 6, 8, 12, 13\}$ and $B = \{c, d, e, f\}$. We notice that $\beta(A) = A' = \{e, f, h\} \neq B$ and $\alpha(B) = B' = \{4, 5, 8, 10, 12, 13\} \neq A$. The lower and upper approximations of (A, B) are equal to $\underline{(A,B)} = (A \cap B', \beta(A \cap B')) = (\{4, 5, 8, 12, 13\}, \{c, d, e, f, h\})$, and $\overline{(A,B)} = (\alpha(A' \cap B), A' \cap B) = (\{4, 5, 6, 8, 10, 12, 13\}, \{e, f\})$, respectively. ♦

Case 2: A is feasible and B is not. Since A is feasible, it can be approximated by the formal concept $C_1 = (A, \beta(A))$ as was done in Section 3.1.

Since B is not feasible, we treat it as a rough set and find formal concepts approximating B as was done in Section 3.2. We also assume that B is not definable. Let C_2 and C_3 denote the formal concepts that represent the lower and upper approximations of B, respectively. From Section 3.2, we find that

$$C_2 = \big(\alpha(\overline{B}), \beta(\alpha(\overline{B}))\big), \quad C_3 = (\alpha(\underline{B}), \beta(\alpha(\underline{B}))).$$

The lower and upper approximations for the pair (A, B) are found by the following formulas:

$$\underline{(A,B)} = \inf\{C_1, C_2\} = \big(A \cap \alpha(\overline{B}), \beta(A \cap \alpha(\overline{B})\big),$$

$$\overline{(A,B)} = \sup\{C_1, C_3\} \, (\alpha(\beta(A) \cap \beta(\alpha(\underline{B}))), \beta(A) \cap \beta(\alpha(\underline{B}))),$$

where the infimum and supremum were found using the Fundamental Theorem of formal concept analysis (Ganter and Wille, 1999; Wille, 1982).

Example 5. Let $A = \{4,5,8,12,13\}$ and $B = \{f,h,i\}$. The set A is feasible because $\alpha(\beta(A)) = A$, while the set B is not feasible because $\beta(\alpha(B)) = \{f,h,i,x\} \neq B$. We have $C_1 = (A,\beta(A)) = (\{4,5,8,12,13\},\{c,d,e,f,h\})$. $C_2 = (\{5,6,7,12,13\},\{f,h,i,x\})$ and $C_3 = (\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$. The details of finding C_2 and C_3 are given in Example 3. Therefore, $\underline{(A,B)} = \inf\{C_1,C_2\} = (\{5,12,13\},\{c,d,e,f,h,i,x\})$ and $\overline{(A,B)} = \sup\{C_1,C_3\} = (\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$. ♦

Case 3: B is feasible and A is not. The derivations for this case are analogous to those in the previous case, and therefore many details are omitted in the following discussion.

The set B is approximated by the concept $C_1 = (\alpha(B),B)$, and the set A is approximated by the lower and upper approximation concepts $C_2 = (\alpha(\beta(\underline{A})),\beta(\underline{A})$ and $C_3 = (\alpha(\beta(\overline{A})),\beta(\overline{A}))$, respectively.

The lower and upper approximations are given by

$$\underline{(A,B)} = \inf\{C_1,C_2\} = (\alpha(\beta(\underline{A})) \cap \alpha(B), \beta(\alpha(\beta(\underline{A})) \cap \alpha(B)))\,,$$

$$\overline{(A,B)} = \sup\{C_1,C_3\} = \big(\alpha(\beta(\overline{A}) \cap B), \beta(\overline{A}) \cap B\big)\,.$$

Case 4: Both A and B are not feasible. In this case we treat both A and B as rough sets, and find formal concepts that represent the lower and upper concept approximations of A and B.

Let C_1 and C_2 denote the lower and upper concept approximations of A, and let C_3 and C_4 denote the lower and upper concept approximations of B, respectively. From Sections 3.1 and 3.2, we find that $C_1 = (\alpha(\beta(\underline{A})),\beta(\underline{A}))$, $C_2 = (\alpha(\beta(\overline{A})),\beta(\overline{A}))$, $C_3 = (\alpha(\overline{B}),\beta(\alpha(\overline{B})))$, and $C_4 = (\alpha(\underline{B}),\beta(\alpha(\underline{B})))$. The lower approximation of (A,B) is given by the formal concept

$$\underline{(A,B)} = \inf\{C_1,C_3\} = \big(\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}), \beta(\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}))\big)\,,$$

and the upper approximation of (A,B) is given by the formal concept

$$\overline{(A,B)} = \sup\{C_2,C_4\} = \big(\alpha(\beta(\overline{A}) \cap \beta(\alpha(\underline{B}))), \beta(\overline{A}) \cap \beta(\alpha(\underline{B}))\big)\,.$$

Example 6. Let $A = \{2,3\}$ and $B = \{f,h,i\}$. From Sections 3.1 and 3.2, we find $C_1 = (\{2,4,5,8,12,13\},\{d,f,h\})$, $C_2 = (\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$, $C_3 = (\{5,6,7,12,13\},\{f,h,i,x\})$, and $C_4 = (\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$. Therefore, $\underline{(A,B)} = \inf\{C_1,C_3\} = (\{5,12,13\},\{c,d,e,f,h,i,x\})$ and $\overline{(A,B)} = \sup\{C_2,C_4\} = (\{2,3,4,5,6,7,8,11,12,13\},\{f,h\})$. ♦

The pseudo-code for approximating a pair (A, B) is given in Algorithm 3. The input is a set A of objects, a set B of features, the quotient set G/R, and the quotient set M/R'. The output is the concept approximation for (A, B). In Line 1, the algorithm tests whether the set A is feasible. If this condition is true, we use Proposition 3 in Line 2 to further test if B is also feasible, $\beta(A) = B$ and $\alpha(B) = B$. If this is also true, the algorithm determines that the pair (A, B) is a formal concept and returns (A, B) for the answer. If the condition in Line 2 fails, the algorithm tests in Line 4 if Case 1.2 holds by testing if B is feasible. If the condition in Line 4 fails and B is found to be non-feasible, then Case 2 holds because the set A was tested to be feasible in Line 1, and Lines 9 through 13 will be executed. Line 15 tests if Case 3 holds and the set B is feasible while A is not. If Case 3 is true, then the concept approximation for (A, B) is found in Lines 16 through 20. Finally, if the condition in Line 15 fails, then both A and B are non-feasible, and Lines 22 through 28 will be executed.

The best case time complexity for Algorithm 3 is attained when the pair (A, B) constitutes a formal concept and is given by $O(|G||M||\alpha(A)|)$. The worst case time complexity happens when both A and B are non-feasible. In this case the $O(|G/R|)$ time is needed to find $|\underline{A}|$ and $|\overline{A}|$ in Lines 22 and 23. Similarly, the $O(|M/R'|)$ time is needed to find $|\underline{B}|$ and $|\overline{B}|$ in Lines 24 and 25. Line 26 requires $O\left(|G||M|(|\underline{A}||\beta(\underline{A})| + |\overline{B}||\alpha(\overline{B})|)|\right)$ to find the lower approximation for (A, B). Similarly, $O\left(|G||M|(|\overline{A}||\beta(\overline{A})| + |\underline{B}||\alpha(\underline{B})|)|\right)$ is needed in Line 27 to find the upper approximation of (A, B). Therefore, the worst case time complexity for Algorithm 3 is $O(|G/R| + |M/R'| + |G||M|(|\underline{A}||\beta(\underline{A})| + |\overline{A}||\beta(\overline{A})| + |\underline{B}||\alpha(\underline{B})| + |\overline{B}||\alpha(\overline{B})|)) = O(|G||M|(|\underline{A}||\beta(\underline{A})| + |\overline{A}||\beta(\overline{A})| + |\underline{B}||\alpha(\underline{B})| + |\overline{B}||\alpha(\overline{B})|)$ because $|G/R|$ is at most equal to $|G|$ and $|M/R'|$ is at most equal to $|B|$. Moreover, because the size of any set of objects is at most $|G|$ and the size of any set of features is at most $|B|$, we conclude that the time complexity for Algorithm 3 is $O(|G|^2|M|^2)$.

Algorithm 3: `Approximate a non-definable concept` (A, B)

1. `if` $(\alpha(\beta(A)) == A)$ `//i.e.` A `is feasible`
2.   `if` $(\beta(A) == B)$ `//Case 1.1`
3.   `Answer` $\leftarrow (A, B)$
4.   `else if` $(\beta(\alpha(B)) == B)$ `//Case 1.2`
5.   `Lower` $\leftarrow (A \cap \alpha(B), \beta(A) \cup B)$
6.   `Upper` $\leftarrow (A \cup \alpha(B), \beta(A) \cap B)$
7.   `Answer` $\leftarrow$ `Lower and Upper`
8.   `else //Case 2: A is feasible but B is not`
9. $\underline{B} \leftarrow \bigcup\{Y \in M/R' \mid Y \subseteq B\}$
10. $\overline{B} \leftarrow \bigcup\{Y \in M/R' \mid Y \cap B \neq \emptyset\}$
11.   `Lower` $\leftarrow \left(A \cap \alpha(\overline{B}), \beta(A) \cup \beta(\alpha(\overline{B}))\right)$
12.   `Upper` $\leftarrow (A \cup \alpha(\underline{B}), \beta(A) \cap \beta(\alpha(\underline{B})))$

        Answer $\leftarrow$ Lower and Upper
    end if
else if $(\beta(\alpha(B)) == B)$ //Case 3: B is feasible, A is not
        $\underline{A} \leftarrow \bigcup\{X \in G/R \mid X \subseteq A\}$
        $\overline{A} \leftarrow \bigcup\{X \in G/R \mid X \cap A \neq \emptyset\}$
        Lower $\leftarrow (\alpha(\beta(\underline{A})) \cap \alpha(B), \beta(\underline{A}) \cup B)$
        Upper $\leftarrow \big(\alpha(\beta(\overline{A})) \cup \alpha(B), \beta(\overline{A}) \cap B\big)$
        Answer $\leftarrow$ Lower and Upper
else //Case 4: Both A and B are not feasible
        $\underline{A} \leftarrow \bigcup\{X \in G/R \mid X \subseteq A\}$
        $\overline{A} \leftarrow \bigcup\{X \in G/R \mid X \cap A \neq \emptyset\}$
        $\underline{B} \leftarrow \bigcup\{Y \in M/R' \mid Y \subseteq B\}$
        $\overline{B} \leftarrow \bigcup\{Y \in M/R' \mid Y \cap B \neq \emptyset\}$
        Lower $\leftarrow \big(\alpha(\beta(\underline{A})) \cap \alpha(\overline{B}), \beta(\underline{A}) \cup \beta(\alpha(\overline{B}))\big)$
        Upper $\leftarrow \big(\alpha(\beta(\overline{A})) \cup \alpha(\underline{B}), \beta(\overline{A}) \cap \beta(\alpha(\underline{B}))\big)$
        Answer $\leftarrow$ Lower and Upper
end if

4. Similarity-Based Approach to Concept Approximation

In this section, we present a similarity-based approach to concept approximation. This approach relies on a similarity between a set or a pair of sets to be approximated and the formal concepts on a given context. We define a similarity measure $f_C(*)$ which we use to find a formal concept C that is most similar to the set or the pair of sets to be approximated. The similarity-based approach to concept approximation has the following advantages: First, the approximation is always presented in terms of one formal concept while in the rough set based approaches the approximation may be described in terms of two formal concepts (Kent, 1994; 1996; Saquer and Deogun, 1999). Second, the similarity-based approach is much simpler than the rough set-based approaches. Third, the algorithms that we present for the similarity-based approach are more efficient than the algorithms based on the rough set approach when the size of the concept lattice is not large. The only disadvantage of the similarity-based approach is that when the size of the concept lattice is large, the algorithms become inefficient. In this case the rough set-based algorithms have better time complexity.

4.1. Approximating a Set of Objects and a Set of Features

Because of the duality between objects and features, approximating a set of objects proceeds in much the same way as approximating a set of features. Therefore, we only show how to approximate sets of features. Let $B \subseteq M$ be a set of features. Our goal is to find a formal concept C whose intent is as similar to B as possible. Define a similarity measure $f_C(B)$ that indicates how similar the extent of C is to $\alpha(B)$ and how similar the intent of C is to B.

$$f_C(B) = \frac{1}{2}\left(\frac{|B \cap \mathrm{Intent}(C)|}{|B \cup \mathrm{Intent}(C)|} + \frac{|\alpha(B) \cap \mathrm{Extent(C)}|}{|\alpha(B) \cup \mathrm{Extent(C)}|}\right).$$

The range of $f_C(B)$ is the interval $[0, 1]$. Here $f_C(B) = 0$ when B and $\alpha(B)$ are disjoint from the intent and extent of C, respectively. Moreover, $f_C(B) = 1$ when $B = \mathrm{Intent}(C)$, and therefore $\alpha(B) = \mathrm{Extent}(C)$. In general, the closer the value of $f_C(B)$ is to 1, the greater the similarity between B and the intent of C. Conversely, the closer the value of $f_C(B)$ is to 0, the smaller the similarity between B and the intent of C. The similarity measure $f_C(*)$ is similar to the membership function used in fuzzy set theory (Zadeh, 1965).

To approximate a set of features B, we find a formal concept C that maximizes the value of $f_C(B)$. In case more than one formal concept are found to approximate B with same value of f, we say that these concepts *equally approximate* B. Ties may be broken arbitrarily depending on the application. However, in applications like medical diagnosis, we may need to present the user with all the concepts that equally approximate B and let the user make the final judgment.

The pseudo-code for approximating a set of features is given in Algorithm 4. The input to this algorithm is the set of all the formal concepts on a given context (G, M, I), which we denote by L, and a set of features $B \subseteq M$.[1] L_i denotes the i-th concept in L. The output is a formal concept C that approximates B and the value of $f_C(B)$. The idea of the algorithm is similar to that of finding a maximal element in a set. Finding the value of $f_C(B)$ requires evaluating $\alpha(B)$ which requires time of the order $|B||G|$. The value of $\alpha(B)$ is assigned to the variable Obj outside the `for` loop to improve the efficiency of the algorithm. The running time complexity of Algorithm 4 is therefore $O(|L| + |B||G|)$.

When the size of the concept lattice, $|L|$, is less than $|G|^2|M|^2$, Algorithm 4 has better performance than Algorithm 2. On the other hand, when $|L|$ is larger than $|G|^2|M|^2$, the time complexity, and therefore the performance, of Algorithm 2 is better than that of Algorithm 4.

Algorithm 4: `Approximate a set` B `of features`

```
C ← L_1
Obj ← α(B)
// Assign f_C(B) to maxvalue
maxvalue ← Evaluate-Membership(Obj,B,C)
n ← |L|
for (i ← 2; i ≤ n; i++)
    if ( Evaluate-Membership(Obj,B,L_i) > maxvalue ) then
```

[1] The most efficient algorithm for finding all the formal concepts of a context is called the Next Algorithm (Ganter, 1984). This algorithm is also described in (Ganter and Wille, 1999).

```
        C ← L_i
        maxvalue ← Evaluate-Membership(Obj,B,C)
      end if
  end for
  Answer ← C and maxvalue
```

The function *Evaluate-Membership* is used to compute the function $f_C(*)$. It takes as arguments a set of objects A, a set of features B, and a formal concept C. It returns the degree of similarity between the set of features B and the concept C when called with arguments $\alpha(B)$, B and C as is done in Algorithm 4. Similarly, *Evaluate-Membership* returns the degree of similarity between the set of objects A and the concept C when called with arguments A, $\beta(A)$ and C. It also returns the degree of similarity between the pair (A, B) and the concept C when called with arguments A, B and C.

Algorithm 5: *Evaluate-Membership(A,B,C)*

$$\textit{Return} \quad \frac{1}{2}\left(\frac{|A \cap \textit{Extent(C)}|}{|A \cup \textit{Extent(C)}|} + \frac{|B \cap \textit{Intent(C)}|}{|B \cup \textit{Intent(C)}|}\right).$$

4.2. Approximating a Non-Definable Concept

Suppose that we are given a set of objects A and a set of features B. The objective is to find a formal concept C such that the extent of C is as similar to A as possible and the intent of C is as similar to B as possible. The formal concept C is then said to approximate the non-definable concept described by the pair (A, B). Define a similarity measure $f_C(A, B)$ that indicates how well the formal concept C approximates the pair (A, B) as follows:

$$f_C(A,B) = \frac{1}{2}\left(\frac{|\mathrm{A} \cap \mathrm{Extent(C)}|}{|\mathrm{A} \cup \mathrm{Extent(C)}|} + \frac{|\mathrm{B} \cap \mathrm{Intent(C)}|}{|\mathrm{B} \cup \mathrm{Intent(C)}|}\right).$$

The expression $|A \cap \mathrm{Extent}(C)|/|A \cup \mathrm{Extent}(C)|$ indicates how similar A is to $\mathrm{Extent}(C)$, and the expression $|B \cap \mathrm{Intent}(C)|/|B \cup \mathrm{Intent}(C)|$ indicates how similar B is to $\mathrm{Intent}(C)$. It is also easy to see that the range of $f_C(A, B)$ is the interval $[0, 1]$. We get $f_C(A, B) = 0$ when C and (A, B) do not have any element in common, and $f_C(A, B) = 1$ when (A, B) is equal to the formal concept C. The closer the value of $f_C(A, B)$ is to 1, the greater the similarity between the pair (A, B) and the formal concept C. Conversely, the closer the value of $f_C(A, B)$ is to 0, the smaller the similarity between (A, B) and C. If there are two different formal concepts C_1 and C_2 where the extent of C_1 is more similar to A than the extent of C_2, the intent of C_2 is more similar to B than the intent of C_1, and $f_{C_1}(A, B) = f_{C_2}(A, B)$, then we make no preference between C_1 and C_2. In this case both C_1 and C_2 equally approximate the pair (A, B). However, when the pair (A, B) is very close to being a formal concept where, for example, there is not much difference between A and $\alpha(\beta(A))$, then it is very rare to find two different formal concepts C_1 and C_2 that satisfy the above-mentioned case. This is so because of the relationships between the extent and intent of a formal concept.

Algorithm 6 gives the pseudo-code for approximating a pair (A, B). The input to Algorithm 6 is the set L of all formal concepts on a given context (G, M, I), a set of objects A, and a set of features B. The output is a concept C that approximates (A, B) and the value of $f_C(A, B)$ which is used as an indication of how well C approximates (A, B). The idea of Algorithm 6 is similar to that of Algorithm 4. The running time complexity of Algorithm 6 is $O(|L|)$. Algorithm 6 is more efficient than Algorithm 3 when $|L|$ is less than $|G|^2|M|^2$, and it is less efficient otherwise.

Algorithm 6: `Approximate a non-definable concept` (A, B)

$C \leftarrow L_1$

`// Assign` $f_C(A, B)$ `to maxvalue`

`maxvalue` $\leftarrow Evaluate - Membership(A, B, C)$

$n \leftarrow |L|$

`for` $(i \leftarrow 2; i \leq n; i{+}{+})$

    `if (`$Evaluate - Membership(A, B, L_i)$ $>$ `maxvalue ) then`

        $C \leftarrow L_i$

        `maxvalue` $\leftarrow$ $Evaluate - Membership(A, B, C)$

    `end if`

`end for`

`Answer` $\leftarrow$ C `and maxvalue`

4.3. Numerical Example

In this section we give a numerical example of the approximation ideas discussed in Sections 4.1 and 4.2.

Consider the context (G, M, I), given in Table 1, which provides information about 13 objects and 12 features that the objects can have. This context has 23 formal concepts which were generated using Algorithm Next described in (Ganter and Wille, 1999). Table 4 gives details about executing Algorithm 4 on the set of features $B = \{f, h, i\}$, and about executing Algorithm 6 on the pair $(A_1, B_1) = (\{4, 6, 9\}, \{e, f, h\})$ which is a non-definable concept because $\alpha(B_1) \neq A_1$. The first column in each row is a label of the concept under consideration. The second and third columns give the extent and intent of the concept. The fourth column contains the value of $f_C(\{f, h, i\})$ which is the degree of similarity between $\{f, h, i\}$ and C. Finally, the fifth column gives the value of $f_C(A_1, B_1)$ which is the degree of similarity between (A_1, B_1) and C.

Considering the values of $f_C(\{f, h, i\})$ in the fourth column, Algorithm 4 returns $(\{5, 6, 7, 12, 13\}, \{f, h, i, x\})$ as the formal concept approximating the set of features $\{f, h, i\}$ with a similarity value of 0.8750. Similary, considering the values in the fifth column, Algorithm 6 returns the formal concept $(\{4, 6\}, \{e, f, h, l\})$ as a result of approximating the non-definable concept $(\{4, 6, 9\}, \{e, f, h\})$ with a similarity value of 0.6333.

Table 4. Results of executing Algorithm 4 on $\{f, h, i\}$ and Algorithm 6 on $(\{4, 6, 9\}, \{e, f, h\})$.

Concept	Extent	Intent	$f_C(\{f,h,i\})$	$f_C(A_1, B_1)$
C_1	{}	$\{a,b,c,d,e,f,h,$ $i,j,k,l,x\}$	0.1250	0.1667
C_2	{6}	$\{e,f,h,i,j,k,l,x\}$	0.2875	0.4167
C_3	{6,7}	$\{f,h,i,j,x\}$	0.5000	0.3750
C_4	{5,12,13}	$\{c,d,e,f,h,i,x\}$	0.5143	0.2857
C_5	{5,6,12,13}	$\{e,f,h,i,x\}$	0.7000	0.4833
C_6	{5,6,7,12,13}	$\{f,h,i,x\}$	0.8750	0.3714
C_7	{4}	$\{a,b,c,d,e,f,h,l\}$	0.1111	0.3333
C_8	{4,10}	$\{c,d,e,f,l\}$	0.0714	0.2679
C_9	{4,9,10}	$\{c,d,f,l\}$	0.0833	0.3214
C_{10}	{4,6}	$\{e,f,h,l\}$	0.2833	0.6333
C_{11}	{4,6,10}	$\{e,f,l\}$	0.1714	0.4500
C_{12}	{4,6,9,10}	$\{f,l\}$	0.1875	0.4750
C_{13}	{4,5,8,12,13}	$\{c,d,e,f,h\}$	0.3810	0.3214
C_{14}	{4,5,8,10,12,13}	$\{c,d,e,f\}$	0.2708	0.2292
C_{15}	{4,5,8,9,10,12,13}	$\{c,d,f\}$	0.2667	0.2083
C_{16}	{4,5,6,8,12,13}	$\{e,f,h\}$	0.5357	0.5179
C_{17}	{4,5,6,8,10,12,13}	$\{e,f\}$	0.3750	0.3750
C_{18}	{2,4,5,8,12,13}	$\{d,f,h\}$	0.4375	0.2625
C_{19}	{2,4,5,8,9,10,12, 13}	$\{d,f\}$	0.2750	0.2111
C_{20}	{2,3,4,5,6,7,8,11, 12,13}	$\{f,h\}$	0.5833	0.3409
C_{21}	{2,3,4,5,6,7,8,9,10, 11,12,13}	$\{f\}$	0.3750	0.2500
C_{22}	{1,2,4,5,8,9,10, 12,13}	$\{d\}$	0.1364	0.1000
C_{23}	{1,2,3,4,5,6,7,8,9, 10,11,12,13}	{}	0.1923	0.1154

5. Conclusion

This paper presents two approaches to approximating concepts in the framework of the formal concept analysis. The first approach is based on rough set theory while the other is based on the use of a similarity measure.

Our approaches compare favorably with the one presented in (Kent, 1994; 1996). First, they are fully automatic and user-independent. Second, they always produce the same answer. Third, we use both the set of objects and the set of features for approximating non-definable concepts, which results in the fact that non-definable concepts with the same set of objects have different and more accurate concept approximations. Fourth, our approaches are general enough to find formal concepts that approximate a single set of objects or a single set of features in addition to approximating non-definable concepts.

Aport from all that, our similarity-based approach to concept approximation has the following advantages: First, the approximation is always presented in terms of one formal concept. Second, the similarity-based approach is much simpler. Third, the algorithms that employ the similarity-based approach are more efficient when the size of the concept lattice is not large.

For both the approaches we show how to approximate single sets of objects, single sets of features, and non-definable concepts. The similarity-based algorithms are more efficient than the rough set-based algorithms when the size of the concept lattice is not large, that is, when $|L|$ is less than $|G|^2|M|^2$, and they are not as efficient otherwise.

References

Carpineto C. and Romano G. (1996): *A lattice conceptual clustering system and its application to browsing retrieval.* — Mach. Learn., Vol.24, No.2, pp.95–122.

Ganter B. (1984): *Two basic algorithms in concept analysis.* — FB4-Preprint No. 831, TH Darmstadt.

Ganter B. and Wille R. (1999): *Formal Concept Analysis: Mathematical Foundations.* — Berlin: Springer.

Godin R. and Missaoui R. (1994): *An incremental concept formation for learning from databases.* — Theoret. Comp. Sci., Vol.133, No.2, pp.387–419.

Ho T.B. (1995): *An approach to concept formation based on Formal Concept Analysis.* — IEICE Trans. Inform. Syst., Vol.78, No.5, pp.553–559.

Kangassalo H. (1992): *On the concept of concept for conceptual modeling and concept deduction*, In: Information Modeling and Knowledge Bases III (S. Ohsuga, H. Kangassalo and H. Jaakkola, Eds.). — Amsterdam: IOS Press, pp.17–58.

Kent R. (1994): *Rough concept analysis.* — Proc. Int. Workshop *Rough Sets and Knowledge Discovery*, Banff, Canada, pp.245–253.

Kent R. (1996): *Rough concept analysis: A synthesis of rough sets and formal concept analysis.* — Fund. Inform., Vol.27, No.2, pp.169–181.

Pawlak Z. (1982): *Rough sets.* — Int. J. Inf. Comp. Sci., Vol.11, No.5, pp.341–356.

Saquer J. and Deogun J. (1999): *Formal rough concept analysis*, In: New Directions in Rough Sets, Data Mining, and Granular-Soft Computing (N. Zhong, A. Skowron and S. Ohsuga, Eds.). — Yamaguchi, Japan: Springer, pp.91–99.

Saquer J. and Deogun J. (2000a): *Concept approximations for formal concept analysis.* — Proc. 8-th Int. Conf. *Conceptual Structures (ICCS'2000)*, Vol.II, *Working with Conceptual Structures*, Darmstadt, Germany, pp.73–83.

Saquer J. and Deogun J. (2000b): *Using closed itemsets for discovering representative association rules*, In: Proc. 12-th Int. Symp. *Methodologies for Intelligent Systems (ISMIS'2000)* (Z. Ras and S. Ohsuga, Eds.). — Charlotte, NC: Springer, pp.495–504.

Wille R. (1992): *Concept lattices and conceptual knowledge systems.* — Comp. Math. Appl., Vol.23., No.5, pp.493–515.

Wille R. (1982): *Restructuring lattice theory: An approach based on hierarchies on concepts*, In: Ordered Sets (I. Rival, Ed.). — Dordrecht: Reidel, pp.445–470.

Wille R. (1989): *Knowledge acquisition by methods of formal concept analysis*, In: Data Analysis, Learning Symbolic and Numeric Knowledge (E. Diday, Ed.). — New York: Nova Science, pp.365–380.

Zadeh L. (1965): *Fuzzy sets.* — Inform. Contr., Vol.8, pp.338–353.