

An algebraic approach to defining rough set approximations and generating logic rules

D. Sitnikov¹ & O. Ryabov²

¹*Kharkov State Academy of Culture, Ukraine*

²*National Institute of Advanced Industrial Science and Technology, Japan*

Abstract

The rough set concept is a relatively new mathematical approach to vagueness and uncertainty in data. The rough set theory is a well-understood formal framework for building data mining models in the form of logic rules, on the basis of which it is possible to issue predictions that allow classifying new cases. The indiscernibility relation and approximations based on this relation form the mathematical basis of the rough set theory. The classical topological definitions of rough approximations are based on the indiscernibility relation. Unlike the classical approaches, in this paper we define rough approximations in an algebraic way. We use a set of predicates and predicate operations, which we call the *approximation language*. We introduce the terms “*exact upper approximation*” and “*exact lower approximation*” to stress the fact that there can exist a variety of approximations but it is always possible to select the approximations that *cannot be improved* in the terms of the approximation language. These new definitions are compared to the classical ones (which use an equivalence relation) and are shown to be more general in the sense that the classical definitions can be deduced from them if we put some restrictions on our model. The process of generating logic rules based on the exact approximations has also been considered. Logic rules are naturally obtained from predicate formulae for the exact approximations. The introduced approach allows generating logic rules quickly and efficiently since only Boolean operations with binary strings are used to produce logic formulae.



1 Introduction

Data mining has to deal with making decisions under uncertain conditions. It often happens that many rows in a database can be interpreted as the indication of the fact that a new case (for example, whether or not a patient has a disease) should be classified as positive (the patient does have the disease). Nevertheless, many other rows may state that patients with the same symptoms do not have the disease. Which decision should be taken when the database contains contradictory information? In order to enable the analyst to make decisions in situations where no conclusion can be drawn with a probability of 100%, various mathematical theories have been developed.

The rough set concept is a relatively new mathematical approach to vagueness and uncertainty in data [1, 2]. The indiscernibility relation and approximations based on this relation form the mathematical basis of the rough set theory. Classical definitions of lower and upper approximations were originally introduced to describe some topological properties of rough sets [3]. These definitions were proposed with reference to an indiscernibility relation, which was assumed to be an equivalence relation. Various generalized definitions of rough approximations have been developed, the majority of them dealing with more general types of relations [4]. In particular, some authors considered a tolerance relation (reflexive and symmetric) as a basis for approximations, which allowed them to express weaker forms of indiscernibility [5]. In some papers researchers proposed interesting definitions of rough approximations based on a similarity relation [6].

The authors of the above papers follow the classical topological way of defining ambiguity concepts. They start from introducing an indiscernibility relation, define some special properties for it and then lower and upper approximations appear in a natural way. In this paper we go in a different direction. We do not use any indiscernibility relation but only predicates in terms of which an arbitrary set of objects can be described. We define lower and upper approximations for sets of objects in an algebraic way using Boolean operations. These new definitions are compared to the classical ones (which use an equivalence relation) and are shown to be more general in the sense that the classical definitions can be deduced from them if we put some restrictions on our model. Using binary codes allows us to quickly calculate the approximations of a set in accordance with the new definitions and generate logic rules based on the approximations.

1.1 An algebraic definition of rough approximations

The classical rough sets theory deals with the “indiscernibility” concept. This concept considers some elements of a set, which cannot be “discerned” in terms of some relation (it normally can be an equivalence or tolerance relation). We would like to consider the ambiguity concept in a different way. We suppose that the available information on the elements of a set is represented with the help of their “properties”. Such an approach allows us to consider *any* relations defined on the set as *properties* of elements, pairs of elements, ordered sets of



elements etc. One can apply various operations to these relations and, as a result, obtain new relations. In fact, any information on the elements of the set can be represented in the form of a relation. The main problem is how to describe any relation between elements of the set in terms of the available relations, i.e. how to represent new information in terms of the available information. Sometimes one can precisely express a given relation in terms of the available relations but often it is not possible to do it and one has to consider approximations to the relation being described.

Consider an example where all relations are unary, i.e. all relations are sets of elements. Suppose we consider the elements $a_1, a_2, a_3, a_4, a_5, a_6$, we have two sets $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{a_2, a_4, a_5\}$ and a set $X = \{a_2, a_4\}$. The question is what we can say about the set X in terms of the sets A and B . In this case we can state that an element belongs to X if and only if it belongs to both A and B . In other words X can be obtained as the intersection of A and B . Suppose now that $X = \{a_1, a_5\}$. In this case we cannot formulate any criterion for an element belonging to X . The set X cannot be obtained from the sets A and B with the help of any operations of intersection, union or negation, therefore we have to consider sets that are *most close* to X . For example, the intersection of A and B is the set $\{a_2, a_4\}$, which is a subset of X . In this case we can say that if an element belongs to the intersection of A and B , it necessarily belongs to X . Since $\{a_2, a_4\}$ is a subset of X , we say that $\{a_2, a_4\}$ is a *lower approximation* for X . Let us now consider quite a general case, where a set of elements should be described in terms of other sets.

Suppose we are given a finite nonempty set of objects $U = \{a_1, a_2, \dots, a_n\}$, called *universe*. Consider also a set of unary predicates (functions that take on their values from the set $\{0,1\}$) defined on U :

$$P_1(t), P_2(t), \dots, P_k(t),$$

which we will call *coordinates*.

The predicates P_1, P_2, \dots, P_k can be interpreted as characteristic functions for some properties of objects of the universe. In this case an object a_i has the property P_j if and only if $P_j(a_i) = 1$. Following the basic concepts of the rough set theory we should describe an arbitrary set $X \subseteq U$ in terms of the coordinates. Since there exists a one-to-one correspondence between all the predicates defined on U and all the subsets of U , instead of a set $X \subseteq U$ we can consider a predicate $X(t)$ that equals 1 if and only if $t \in X$. Thus we should give a description of an arbitrary predicate $X(t)$ in terms of the predicates P_1, P_2, \dots, P_k . In this connection it seems natural to consider some logic or algebraic language that would allow us to discover links between the predicates $X(t)$ and P_1, P_2, \dots, P_k . In this paper we suppose that only Boolean operations can be applied to the predicates. We will say that the *approximation language* consists of the unary predicates P_1, P_2, \dots, P_k and the Boolean operations. It is necessary to stress that in the general case the approximation language can include other types of predicates and operations.

Consider the set Φ of all possible formulae constructed with the help of the Boolean operations *conjunction* ($\&$), *disjunction* (\vee) and *negation* (\neg) applied to



the predicates P_1, P_2, \dots, P_k . For example: $(P_1 \& P_2 \& \neg P_3) \vee P_4, (P_1 \vee (P_2 \& \neg P_3)) \& P_4, \neg (P_1 \& P_2 \& \dots \& P_4)$ etc. On calculating all the formulae belonging to Φ , we will obviously obtain a set of predicates, which we will denote Λ . We must note here that different formulae (not necessarily equivalent ones) can correspond to the same predicate. Let us consider a simple example. Suppose $U = \{a_1, a_2\}$; $P_1(a_1)=1, P_1(a_2)=1$; $P_2(a_1)=1, P_2(a_2)=0$. On calculating the formulae $\varphi = P_1 \& P_2$ and $\pi = P_1$ we get the same predicate P_1 although these formulae are not logically equivalent (neither of them can be obtained from the other by equivalent transformations), nevertheless in this particular case their values are equal.

If the predicate $X(t)$ belongs to Λ , it means that $X(t)$ can be expressed in terms of the coordinates and the set X corresponding to the predicate $X(t)$ can be called *crisp* with respect to the coordinates. If the predicate $X(t)$ does not belong to Λ , this predicate cannot be expressed in terms of the coordinates and we should describe it *approximately*. The following definitions allow us to do it.

Definition 1. If $A(t) \in \Lambda$ and $\forall t \in U A(t) \rightarrow X(t)$ then we say that $A(t)$ is a *lower approximation* for $X(t)$.

Definition 2. If $B(t) \in \Lambda$ and $\forall t \in U X(t) \rightarrow B(t)$ then we say that $B(t)$ is an *upper approximation* for $X(t)$.

Definition 3. If a predicate $I_*(t)$ is a lower approximation for the predicate $X(t)$ and for any lower approximation $A(t)$ of this predicate $\forall t \in U A(t) \rightarrow I_*(t)$ then we say that $I_*(t)$ is an *exact lower approximation* for $X(t)$.

Definition 4. If a predicate $I^*(t)$ is an upper approximation for the predicate $X(t)$ and for any upper approximation $B(t)$ of this predicate $\forall t \in U I^*(t) \rightarrow B(t)$ then we say that $I^*(t)$ is an *exact upper approximation* for $X(t)$.

1.2 Properties of the approximations

It can be easily shown that:

1. The set of lower approximations is not empty for any $X(t)$. This follows from the fact that the predicate $0 = P_1 \& \neg P_1$ is always a lower approximation for $X(t)$.
2. The set of upper approximations is not empty for any $X(t)$. This statement is true since the predicate $1 = P_1 \vee \neg P_1$ is always an upper approximation for $X(t)$.
3. For any predicate $X(t)$ there cannot exist more than one exact lower approximation. Suppose that we have two exact lower approximations $I_1^*(t)$ and $I_2^*(t)$ of $X(t)$. Consider the predicate $I^*(t) = I_1^*(t) \vee I_2^*(t)$. It is obviously an exact lower approximation because $I_1^*(t) \rightarrow I^*(t)$ and $I_2^*(t) \rightarrow I^*(t) \forall t \in U$. On the other hand, since $I_1^*(t)$ and $I_2^*(t)$ are exact lower approximations for $X(t)$, $\forall t \in U I^*(t) \rightarrow I_1^*(t)$ and $I^*(t) \rightarrow I_2^*(t)$. Thus $I^*(t) = I_1^*(t) = I_2^*(t)$.
4. For any predicate $X(t)$ there exists at least one exact lower approximation. Consider the predicate $I^*(t)$ that is the disjunction of all the lower approximations for $X(t)$. It is obvious that for any lower approximation $A(t)$ the following property is true: $\forall t \in U A(t) \rightarrow I^*(t)$. Thus $I^*(t)$ is an exact lower approximation.



5. For any predicate $X(t)$ there exists the only exact lower approximation. It follows from properties 3 and 4.
6. For any predicate $X(t)$ there cannot exist more than one exact upper approximation. Suppose that we have two exact upper approximations $I1^*(t)$ and $I2^*(t)$ of $X(t)$. Consider the predicate $I^*(t) = I1^*(t) \& I2^*(t)$. It is obviously an exact upper approximation because $I^*(t) \rightarrow I1^*(t)$ and $I^*(t) \rightarrow I2^*(t) \quad \forall t \in U$. On the other hand, since $I1^*(t)$ and $I2^*(t)$ are exact upper approximations for $X(t)$, $\forall t \in U I1^*(t) \rightarrow I^*(t)$ and $I2^*(t) \rightarrow I^*(t)$. Thus $I^*(t) = I1^*(t) = I2^*(t)$.
7. For any predicate $X(t)$ there exists at least one exact upper approximation. Consider the predicate $I^*(t)$ that is the conjunction of all the upper approximations for $X(t)$. It is obvious that for any upper approximation $A(t)$ the following property is true: $\forall t \in U I^*(t) \rightarrow A(t)$. Thus $I^*(t)$ is an exact lower approximation.
8. For any predicate $X(t)$ there exists the only exact upper approximation. It follows from properties 6 and 7.

1.3 Comparison with the classical definitions

Let us compare the classical definitions of lower and upper approximations to the definitions of exact approximations introduced in this paper. In the classical rough set theory an indiscernibility equivalence relation is the mathematical basis for rough approximations. Suppose we are given an equivalence relation I , which is used for constructing approximations of a set X [2]. The relation I defines a set of equivalence classes, where each class corresponds to a predicate taking on a value of 1 for the elements belonging to the class and a value of 0 for the other elements of the universe. Thus we have a set of predicates $P_1(t), P_2(t), \dots, P_k(t)$ that satisfy the following conditions corresponding to the well-known properties of equivalence classes:

$$\exists t P_i(t), i = 1, 2, \dots, k, \quad (1)$$

$$\forall t \in U P_1(t) \vee P_2(t) \vee \dots \vee P_k(t), \quad (2)$$

$$\forall t \in U \neg(P_i(t) \& P_j(t)); i, j = 1, 2, \dots, k; i \neq j. \quad (3)$$

It can be deduced from the classical definitions of rough approximations that the lower approximation for a set X is the union of all the equivalence classes that are subsets of X , the upper approximation for X is the union of all the equivalence classes that have a non empty intersection with X .

Using logic terms we can re-formulate these definitions as follows:

Definition 5. The lower approximation for a predicate $X(t)$ is the disjunction of all the predicates $P_i(t)$ for which $\forall t \in U P_i(t) \rightarrow X(t)$.

Definition 6. The upper approximation for a predicate $X(t)$ is the disjunction of all the predicates $P_i(t)$ for which $\exists t \in U P_i(t) \& X(t)$.

Let us show that definitions 5 and 6 are equivalent to definitions 3 and 4 accordingly (exact upper and lower approximations in definitions 3 and 4 correspond to the upper and lower approximations in definitions 5 and 6) given conditions (2) and (3) are satisfied. Obviously it is sufficient to prove that for an



arbitrary set X the approximations obtained according to these definitions are the same predicates.

Consider an arbitrary predicate $X(t)$, $t \in U$. On calculating any Boolean formula with the predicates $P_1(t)$, $P_2(t)$, ..., $P_k(t)$ we obtain the disjunction of some of these predicates, since the conjunction of two different predicates $P_i(t)$ and $P_j(t)$ equals 0 for any $t \in U$ and the negation operation applied to the disjunction of some predicates $P_i(t)$ gives the disjunction of the rest of the predicates (see conditions 2 and 3). Since

- any lower approximation is the disjunction of some of the predicates $P_i(t)$ for which $\forall t \in U P_i(t) \rightarrow X(t)$,
 - the exact lower approximation for $X(t)$ is the disjunction of all the lower approximations obtained according to definition 1 (property 4),
- we can state that the exact lower approximation obtained according to definition 3 is equal to the lower approximation obtained according to definition 5. We have shown that definitions 3 and 5 are equivalent under the assumptions made. Since

- the disjunction D of all the predicates $P_i(t)$ for which $\exists t \in U P_i(t) \& X(t)$ is an upper approximation in terms of definition 2,
- any other upper approximation F is the disjunction of some predicates $P_i(t)$, which necessarily includes all the predicates from D

we can state that the exact lower approximation obtained according to definition 4 is equal to the lower approximation obtained according to definition 6. We have demonstrated the fact that definitions 4 and 6 are equivalent under the assumptions made.

Note that the new definitions are equivalent to the classical ones only if the predicates $P_i(t)$ satisfy conditions (2) and (3) (i.e. in the case where the predicates $P_i(t)$ describe equivalence classes). If it is not so, nevertheless, definitions (3) and (4) still can be used to approximately describe rough sets *in terms of the coordinates*.

1.4 Calculating the exact approximations for a predicate

Verifying all possible Boolean formulae to calculate exact approximations is a time consuming procedure. Nevertheless, there exists a way of obtaining the approximations for a predicate that allows us to quickly write down necessary formulae. Consider the following table:

	a_1	a_2	...	a_n
P_1	δ_{11}	δ_{12}	...	δ_{1n}
P_2	δ_{21}	δ_{22}	...	δ_{2n}
...
P_k	δ_{k1}	δ_{k2}	...	δ_{kn}
X	λ_1	λ_2	...	λ_n

where $\delta_{ij}, \lambda_j \in \{0,1\}$, if $\delta_{ij} = 1$ then $P_i(a_j) = 1$, if $\delta_{ij} = 0$ then $P_i(a_j) = 0$, if $\lambda_j = 1$ then $X(a_j) = 1$, if $\lambda_j = 0$ then $X(a_j) = 0$.

Suppose that the predicate X should be described in terms of the coordinates P_1, P_2, \dots, P_k . Let us find the exact upper approximation for X . For this purpose consider the columns of the table that contain 1 for the predicate X and write down the corresponding disjunctive normal form. A simple example is given below:

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	0	1	0
P_2	0	0	1	1	1
P_3	0	1	0	0	0
X	0	1	0	1	1

For this example we will get the following formula:

$$I^* = (\neg P_1 \ \& \ \neg P_2 \ \& \ P_3) \vee (P_1 \ \& \ P_2 \ \& \ \neg P_3) \vee (\neg P_1 \ \& \ P_2 \ \& \ \neg P_3) \quad (4)$$

Consider now the columns that contain 0 for the predicate X and write down the corresponding conjunctive normal form. For this example:

$$I_* = (\neg P_1 \vee P_2 \vee P_3) \ \& \ (P_1 \vee \neg P_2 \vee P_3) \quad (5)$$

Formulae (4) and (5) produce the following results:

	a_1	a_2	a_3	a_4	a_5
P_1	1	0	0	1	0
P_2	0	0	1	1	1
P_3	0	1	0	0	0
X	0	1	0	1	1
I^*	0	1	1	1	1
I_*	0	1	0	1	0

In the general case the predicates I^* and I_* can be represented as follows:

$$I^* = (\lambda_1 \ \& \ P_1 * \delta_{11} \ \& \ P_2 * \delta_{21} \ \& \ \dots \ \& \ P_k * \delta_{k1}) \vee (\lambda_2 \ \& \ P_1 * \delta_{12} \ \& \ P_2 * \delta_{22} \ \& \ \dots \ \& \ P_k * \delta_{k2}) \vee \dots \vee (\lambda_n \ \& \ P_1 * \delta_{1n} \ \& \ P_2 * \delta_{2n} \ \& \ \dots \ \& \ P_k * \delta_{kn}), \quad (6)$$



$$I^* = (\lambda_1 \vee P_1^*(1-\delta_{11}) \vee P_2^*(1-\delta_{21}) \vee \dots \vee P_k^*(1-\delta_{k1})) \& (\lambda_2 \vee P_1^*(1-\delta_{12}) \vee P_2^*(1-\delta_{22}) \vee \dots \vee P_k^*(1-\delta_{k2})) \& \dots \& (\lambda_n \vee P_1^*(1-\delta_{1n}) \vee P_2^*(1-\delta_{2n}) \vee \dots \vee P_k^*(1-\delta_{kn})), \quad (7)$$

where $P^*\delta = P$ if $\delta = 1$ and $P^*\delta = \neg P$ if $\delta = 0$ for any predicate P .

Let us show that the predicates I^* and I_* are the exact upper and lower approximations. It is obvious that the predicate I^* is an upper approximation for the predicate X in terms of definition 2. If one removes from formula (6) any conjunction where $\lambda_i = 1$, the resulting formula will not be an upper approximation, as the predicate I^* will have 0 in a column where X has 1. (We suppose here that in case there are several conjunctions identical to the one removed all of them should be removed). For example if one removes the conjunction $(\neg P_1 \& P_2 \& \neg P_3)$ from formula (4), then $I^*(a_5)$ becomes 0 whereas $X(a_5) = 1$. It means that the approximation I^* cannot be improved and, therefore, I^* is the exact upper approximation. The predicate I_* is obviously a lower approximation for X in terms of definition 1. If one removes from formula (7) any disjunction where $\lambda_i = 0$, the resulting formula will not be a lower approximation as the predicate I_* will have 1 in a column where X has 0. (We suppose that if there are several disjunctions identical to the one removed all of them should be removed). For example if one removes the disjunction $(P_1 \vee \neg P_2 \vee P_3)$ from formula (5), then $I^*(a_3) = 1$ whereas $X(a_3) = 0$. It means that the approximation I_* cannot be improved and, therefore, I_* is the exact lower approximation.

1.5 Approximation-based logic rules

Consider an example of generating logic rules with the help of formulae (4) and (5). Following traditional rough set concepts we can say that rules based on the exact upper approximation *may* exist in the data set, and rules based on the exact lower approximation *must* exist in the data. Transform the expression on the right side of formula (5) to get:

$$I_* = P_3 \vee (\neg P_1 \vee P_2) \& (P_1 \vee \neg P_2) = P_3 \vee (P_1 \& P_2) \vee (\neg P_1 \& \neg P_2)$$

We can now formulate the following *exact* rules:

- A. An element *belongs* to the set X if
 - a) property P_3 is true for this element
- OR
- b) properties P_1 and P_2 are true for this element
- OR
- c) neither property P_1 nor P_2 are true for this element

This rule says that if one of the conditions a), b) or c) holds, an element belongs to the set X .

Let us simplify the expression on the right side of formula (4) to get:

$$\Gamma^* = (\neg P_1 \ \&\neg P_2 \ \&P_3) \vee (P_1 \vee \neg P_1) \ \& (P_2 \ \&\neg P_3) = (\neg P_1 \ \&\neg P_2 \ \&P_3) \vee (P_2 \ \&\neg P_3)$$

This formula allows us to formulate the following *approximate* rules:

- B. An element *may belong* to the set X *if* neither property P_1 nor P_2 are true *AND* property P_3 is true for this element
- C. An element *may belong* to the set X *if* property P_2 is true *AND* property P_3 is not true for this element.

Note that the rule B can be removed from the set of the rules since, according to the rule A, if property P_3 is true then an element belongs to X.

We have shown that logic rules that allow answering questions on whether or not an element belongs to a given set can be deduced from the available information on properties of elements by using Boolean operations. Since Boolean calculations are very quick, the resulting rules can be efficiently obtained even for great numbers of elements and predicates.

2 Conclusion

In this paper a new approach to defining rough set approximations has been considered. The main idea of this approach is to describe a concept in terms of other concepts in an *algebraic way*, i.e. to find formulae that represent the most exact approximations of the concept under analysis. Although in this paper mainly Boolean functions have been investigated, this approach can be applied to other logic and algebraic structures that may be used to describe rough concepts. For example, if we remove the negation operation from the set of the operations allowed to describe predicates or add other operations, we will obtain new interesting properties of the approximations. The terms “exact upper approximation” and “exact lower approximation” have been introduced to stress the fact that there can exist a variety of upper and lower approximations, but it is possible to select approximations that *cannot be improved* in the terms of the approximation language. Such approximations describe a rough set in the most precise way. Logic rules generated with the help of the exact approximations can be used to classify new elements, i.e. to define whether or not an element belongs to a set, which is a classical problem of Data Mining. The proposed approach considers only operations with Boolean strings, which makes the process of extracting logic rules very quick from the computational point of view.

Logic rules obtained as a result of the proposed approach are represented in the perfect disjunctive (conjunctive) normal form, which makes it difficult to distinguish between salient and non-salient features of the set elements since any term of the resulting formula contains all the coordinates of the approximation language. The authors are looking for efficient ways of determining which properties are more (less) important to classify elements of a set in terms of the available features.



References

- [1] Pawlak Z.; (1982), *Rough sets*, International Journal of Computer and Information Sciences, 11, pp. 341-356.
- [2] Pawlak, Z.; (1995) *Vagueness and uncertainty: a rough set perspective*, Computational Intelligence, vol. 11 (issue 2), pp. 227-232.
- [3] Pawlak, Z.; (1996), *Rough sets, rough relations and rough functions*, Fundamenta Informaticae, vol. 27, no. 2/3, pp. 103-108.
- [4] Yao, Y.; Wong, S.; (1995), *Generalization of rough sets using relationships between attribute values*, in Proceedings of the 2nd Annual Joint Conference on Information Sciences, pp. 30-33.
- [5] Polkowski, L.; Skowron, A.; Zytkow, J.; (1995), *Tolerance based rough sets*, in Lin, T; Wildberger, A.; (eds.), *Soft computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management*, Simulation councils, Inc., San Diego, pp. 55-58.
- [6] Slowinski R.; Vanderpooten D.; (1997), *Similarity relation as a bases for rough approximations*, in Wong, P.P.; (ed.), *Advances in Machine Intelligence and Soft-Computing*, vol. IV, Bookwrights, Raleigh, NC, 17-33.

