Knowledge-Based Systems 80 (2015) 67–77

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

The two sides of the theory of rough sets

Yiyu Yao

Department of Computer Science, University of Regina, Regina, Saskatchewan S4S 0A2, Canada

ARTICLE INFO

Article history:
Received 15 October 2014
Received in revised form 11 January 2015
Accepted 12 January 2015
Available online 22 January 2015

Keywords:
Computational formulation
Concepts before computation
Conceptual formulation
Reducts
Reduct construction algorithms
Rough sets

ABSTRACT

There exist two formulations of the theory of rough sets. A conceptual formulation emphasizes on the meaning and interpretation of the concepts and notions of the theory, whereas a computational formulation focuses on procedures and algorithms for constructing these notions. Except for a few earlier studies, computational formulations dominate research in rough sets. In this paper, we argue that an oversight of conceptual formulations makes an in-depth understanding of rough set theory very difficult. The conceptual and computational formulations are the two sides of the same coin; it is essential to pay equal, if not more, attention to conceptual formulations. As a demonstration, we examine and compare conceptual and computational formulations of two fundamental concepts of rough sets, namely, approximations and reducts.

1. Introduction

To develop and study a theory related to computation, one must precisely define its basic concepts and notions. A conceptual definition or formulation of a concept focuses on the meaning, interpretation and inherent properties of the concept, whereas a computational definition or formulation focuses on algorithms and methods for testing if something is an instance of the concept or constructing an instance of the concept. A conceptual definition, although essential for understanding, may not directly give a computationally efficient method. On the other hand, a computational definition, although suitable for computation, may push the meaning of a concept to the background and make an understanding difficult. Conceptual and computational definitions are the two sides of the same coin; it is necessary to consider both of them.

An important application of the theory of rough sets is analyzing data and reasoning about data [40], with a very strong computational orientation. It is therefore not surprising that the majority of studies of rough sets concentrate on computational definitions and formulations. A review of literature shows that there are very limited studies on conceptual formulations of rough sets, except for a few earlier studies that motivate the introduction of rough set theory [28–30,36,37,56]. Many theoretical studies of rough sets introduce new concepts and produce new results without a clear interpretation of their meaning.

An overlook of studies on conceptual definitions and formulations may hinder further development of rough set theory. Therefore, we advocate a change of attention to conceptual definitions and formulations of the basic concepts and notions of rough set theory. Our goal is not to introduce new concepts per se, but to recast existing concepts and results in a setting of conceptual formulations. To demonstrate the power and value of conceptual definitions and formulations of rough set theory, we examine and compare both conceptual and computational formulations of rough set approximations and reducts.

In Section 2, we provide a brief discussion on conceptual and computational formulations across many disciplines. Section 3 examines a conceptual definition of rough set approximations in terms of the definability of sets. More specifically, the definability of sets in an information table is a primitive notion, from which rough set approximations are naturally derived. Section 4 reviews computational definitions and shows the equivalence of conceptual definitions and computational definitions. Section 5 gives a conceptual definition of a reduct of a set. Section 6 shows that attribute reducts, relative attribute reducts, attribute–value reducts, and rule reducts [12,40] can be unified under the conceptual definition introduced in Section 5. More importantly, we have a simple explanation of Pawlak rough set analysis [40] as a three-step process [60]: (1) finding an attribute reduct to simplify an information table, (2) finding an attribute–value pair reduct to simplify the left-hand-side of a classification or decision rule, and (3) finding a rule reduct to simplify a set of classification rules. Section 7 looks at the construction of a reduct. We present three classes of algorithms by using deletion, addition-deletion, and addition strategies [61], respectively.

E-mail address: yyao@cs.uregina.ca

http://dx.doi.org/10.1016/j.knosys.2015.01.004
0950-7051/

2. Conceptual versus computational formulations

According to Watt and van den Berg [51], concepts are the building blocks of scientific theories. A scientific concept consists of a label, a theoretical definition and an operational definition. The theoretical definition of a concept concerns the meaning of the concept. An operational definition provides ways so that the concept can be objectively observed or measured. A more detailed discussion on definitions, their purposes, and various kinds of definitions (including theoretical definitions and operational definitions) can be found in the book by Hurley [16]. In this paper, we focus on the conceptual and computational formulations or aspects as suggested by theoretical and operational definitions, respectively.

We first look at an example used by Guttmannova et al. [13] to illustrate the differences between definitional (i.e., conceptual) and computational definitions in statistics. To define the variance of a population, one uses the definitional formula:

$$s^2 = \frac{\sum_i^n (x_i - m)^2}{n}, \tag{1}$$

where x_i's are the raw scores, n is the number of scores, and

$$m = \frac{\sum_i^n x_i}{n}, \tag{2}$$

is the arithmetic mean of all observations. The definitional formula offers a clear interpretation that the "variance of a distribution is a measure of how dispersed the scores are around the mean in the distribution" [13]. On the other hand, a *direct* application or *literal* use of the formula requires two scans of a data table. The first scan computes the arithmetic mean and the second scan computes the sum of squared differences from mean. By combining Eqs. (1) and (2) and performing some equivalent transformations, one arrives at a computational formula:

$$s^2 = \frac{1}{n}\left(\sum_i^n x_i^2 - \frac{\left(\sum_i^n x_i\right)^2}{n}\right). \tag{3}$$

Accordingly, a single scan of data enables us to compute both $\sum_i^n x_i^2$ and $\left(\sum_i^n x_i\right)^2$, resulting in a more efficient computational method. Unfortunately, the meaning of the variance becomes obscure with the computational formula.

Guttmannova et al. [13] conclude, "conceptual understanding is advanced by examining and using the definitional formula, whereas the computational formula is only a computational tool with no interpretative significance." In a study on students' understanding of the mean, Pollatsek et al. [41] similarly conclude, "Learning a computational formula is a poor substitute for gaining an understanding of the basic underlying concept." "Knowledge of a computational rule not only does not imply any real understanding of the basic underlying concept but may actually inhibit the acquisition of more adequate (relational) understanding." Rybolt and Recck [43] discuss a few more examples of conceptual and computational formulas from calculus and statistics and suggest that "students perform better when using the conceptual style."

The basic ideas of the two sides of statistics, namely, the conceptual nature of statistics and the computation of the statistics, are generally applicable across many disciplines, although they are often discussed by using different terminologies. We give a few examples as a demonstration. Interesting enough, they are related to teaching and eduction where conceptual understanding plays a crucial role.

Ma [26] investigates the roles of a conceptual understanding and a procedural understanding of a topic in teaching elementary mathematics. The National Mathematics Advisory Panel's (2008) final report [31] stresses on the following aspects in teaching elementary mathemetics, as well as their mutually reinforcing benefits:

conceptual understanding,	computational/ precedural proficiency;
conceptual development,	computational/ procedural fluency;
conceptual knowledge and skills,	procedural knowledge and skills.

For teaching physics, Ballif and Eibble [3] take a "conceptual rather than mathematical" approach to the principles of physics, in an attempt to "present and explain the fundamental principles of physics in a way that non-scientists will find comprehensible and interesting." Hewett [14] advocates a conceptual approach to teaching physics that "emphasizes comprehension before computation," or concepts before computation. By studying how students solve conceptual and computational problems of chemical equilibrium, Niaz [32] reports that "students who perform better on problems requiring conceptual understanding also perform significantly better on problems requiring manipulation of the data."

A conceptual formulation and development is generally appropriate for understanding, but may not immediately provide a computational method. On the other hand, a computational formulation is good for calculation, but may make an in-depth understanding difficult. They are the two sides of the same coin. It is therefore important to consider both and fully explore their mutual reinforcement.

If we do accept the vast scientific evidence that supports the value of conceptual formulation cross many disciplines [35], we should pay attention to conceptual formulations of rough set theory. This paper is a plea for conceptual formulations of rough sets, in an attempt to complement existing studies dominated by computational formulations.

3. A conceptual understanding of approximations

From a conceptual point of view, rough sets approximations are introduced due to the undefinability of some subsets of objects in an information table. In other words, a basic idea of rough set theory is the approximations of undefinable sets by definable sets.

3.1. Definability of sets in an information table

In the classical view of concepts, a concept has well-defined boundaries and is describable by sets of singly necessary and jointly sufficient conditions [48]. Every concept is represented and understood as two parts, namely, the intension and the extension of the concept [16,33,46–48]. The intension of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities that are instances of the concept.

In this paper, we consider a concept analysis view of rough sets [59], focusing on the analysis of data represented in a tabular form [40]. Formally, an information table T can be defined by a tuple as follows [37,40]:

$$T = (OB, AT, \{V_a \mid a \in AT\}, \{I_a \mid a \in AT\}), \tag{4}$$

where OB is a finite nonempty set of objects called the universe, AT is a finite nonempty set of attributes, V_a is the domain of attribute a, and $I_a : U \longrightarrow V_a$ is an information function. We use $I_a(x)$ to denote the value of object x on attribute a. With reference to the classical view of concepts, a fundamental task of rough set analysis is to explicitly define and interpret the intension and extension of a concept by using an information table.

In an information table, we can simply use a subset of objects $X \subseteq OB$ to define the extension of a concept. In order to describe formally the intension of a concept, we use a simplified sublanguage of a description language suggested by Pawlak [36,40] and by Marek and Pawlak [29]. A description language DL is recursively defined by using logic conjunction $\wedge$ and disjunction $\vee$, as well as parentheses, as follows:

$$\begin{aligned} &(1) \quad \langle a, v\rangle \in DL, \text{ where } a \in AT, v \in V_a, \\ &(2) \quad \text{if } p, q \in DL, \text{ then } (p) \wedge (q), (p) \vee (q) \in DL. \end{aligned}$$

Formulas defined by (1) are called atomic formulas or attribute–value pairs. Formulas defined by (2) contain extra parentheses to indicate the order of computation. By assuming that $\wedge$ has a higher precedence than $\vee$ and association of operators is from left to right, one may remove unnecessary parentheses in a formula. If one only considers the syntactic forms of formulas without using unnecessary parentheses, one may change (2) into:

$$(2') \quad \text{if } p, q \in DL, \text{ then } (p), p \wedge q, p \vee q \in DL.$$

In this paper, we use (2) and (2′) interchangeably. One can simply remove or add extra parentheses to transform syntactic forms of a formula without changing its meaning.

The satisfiability of a formula p by an object x, written $x \models p$, is defined as follows [40]:

$$\begin{aligned} &(\text{i}) \quad x \models \langle a, v\rangle, \text{ iff } I_a(x) = v, \\ &(\text{ii}) \quad x \models (p) \wedge (q), \text{ iff } x \models p \text{ and } x \models q, \\ &(\text{iii}) \quad x \models (p) \vee (q), \text{ iff } x \models p \text{ or } x \models q. \end{aligned} \tag{5}$$

Based on the notion of satisfiability, we define the meaning of a formula in an information table. For a formula $p \in DL$, the set of objects $\|p\| \subseteq OB$ defined by:

$$\|p\| = \{x \in OB | x \models p\} \tag{6}$$

is called the meaning set of formula p. The meaning set $\|p\|$ consists of all objects that satisfy p. With the introduction of meaning sets, we can establish the following linkage between logic and set operations:

$$\begin{aligned} &(\text{m1}) \quad \|\langle a, v\rangle\| = \{x \in OB | I_a(x) = v\}, \\ &(\text{m2}) \quad \|(p) \wedge (q)\| = \|p\| \cap \|q\|, \\ &(\text{m3}) \quad \|(p) \vee (q)\| = \|p\| \cup \|q\|. \end{aligned} \tag{7}$$

That is, logic conjunction and disjunction are interpreted in terms of set intersection and union, respectively.

If a formula $q \in DL$ only contains logic conjunction, the formula is called a conjunctive formula. In other words, a conjunctive formula is simply a conjunction of a family of atomic formulas or attribute–value pairs. An atomic formula $\langle a, v\rangle$ is a conjunctive formula. Conjunctive formulas play an important role in rough set analysis. Any formula in DL can be re-expressed as a disjunction of a family of conjunctive formulas by repeatedly applying distributive laws $(p \vee q) \wedge (p \vee r) = p \vee (q \wedge r)$ and $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$. In fact, decision or classification rules used in rough set theory are represented by using conjunctive formulas.

The meaning-set function $\|\cdot\|$ maps a formula p to a unique subset $\|p\|$ of OB. However, the reverse process is not so simple. For an arbitrary subset $X \subseteq OB$, we may or may not find a formula $p \in DL$ such that $X = \|p\|$. In the case of $X = \|p\|$, we call p a description of objects in X or an intension of a concept defined by the extension X. However, such a description is not necessarily unique. We may also find another formula $p' \in DL$ such that $X = \|p'\|$. For rough set analysis, one may find either a description or the family of all descriptions of a given set. In this paper, we are interested in the existence of a description for introducing definability of sets or concepts, following the studies by Pawlak [36], Marek and Palwak [29], and Yao [56].

Definition 1. Suppose a set of objects $X \subseteq OB$ is the extension of a concept. We call X a definable concept or a definable set if there exists a formula $p \in DL$ such that

$$X = \|p\|. \tag{8}$$

Otherwise, X is an undefinable set. Furthermore, X is a conjunctively definable set if there exists a conjunctive formula $q \in DL$ such that $X = \|q\|$.

Prior to the introduction of the theory of rough sets, Pawlak [36], Marek and Pawlak [29] had already suggested this notion of a definable set in an information table and called it a describable set. Unfortunately, this semantically superior notion of definability was not explicitly used in subsequent computational formulations of rough sets [38], except for a few studies. Marek and Truszczyński [30] used the notion of definability to explain Pawlak rough set approximations. Yao [56], unaware of the paper by Marek and Truszczyński, argued again for a semantically sound conceptual definition of rough sets by using the notion of definability. There is an important point from this historical remark. Although it was possible to have a semantically sound conceptual definition of rough set approximations based on results of prior studies that motivated the introduction of rough sets, such a definition was overlooked due to an emphasis on computational definitions. For a discussion on early developments of rough set theory, see a recent paper by Marek [28].

The definability of a set is defined with respect to the entire set of attributes AT. One can also talk about the definability of a set with respect to a nonempty subset of attributes $\emptyset \neq A \subseteq AT$. In this case, we simply restrict the language DL into DL_A by using only attributes in A.

Definition 2. Suppose $\emptyset \neq A \subseteq AT$ is a nonempty subset of attributes in an information table. A set of objects $X \subseteq OB$ is an A-definable concept or set if there exists a formula $p \in DL_A$ such that

$$X = \|p\|. \tag{9}$$

Otherwise, X is an A-undefinable set. Furthermore, X is a conjunctively A-definable set if there exists a conjunctive formula $q \in DL_A$ such that $X = \|q\|$.

By definition, a definable set is an AT-definable set. Let $\mathrm{DEF}_A(T)$ denote the family of all A-definable sets in an information table T. It can be verified [29,30,56] that, when an information table consists of a finite set of objects and a finite set of attributes, $\mathrm{DEF}_A(T)$ contains the empty set $\emptyset$, the universe OB, and is closed under set complement, intersection and union. In other words, $\mathrm{DEF}_A(T)$ is a sub-Boolean algebra of the power set $P(OB)$ of OB. Any set in $P(OB) - \mathrm{DEF}_A(T)$ is an A-undefinable set. In the rest of the paper, we also simply refer to an A-definable set as a definable set, an A-undefinable set as an undefinable set, when A is clear from the context.

3.2. A conceptual definition of rough set approximations

For each undefinable set, one cannot find a formula to describe it. In order to make an inference about an undefinable set, one must approximate it by using definable concepts. Thus, the notion of definability leads naturally to another fundamental notion of rough sets, namely, approximations of an undefinable set by a pair of definable sets [30,56].

Definition 3. In an information table T, for a subset $X \subseteq OB$, the lower and upper approximations of X are defined by the following pair of definable sets,

$$\begin{aligned} \underline{apr}_A(X) &= \text{the greatest definable set in } \mathrm{DEF}_A(T) \text{ contained by } X, \\ \overline{apr}_A(X) &= \text{the least definable set in } \mathrm{DEF}_A(T) \text{ containing } X, \end{aligned} \tag{10}$$

where the greatest and the least definable sets are defined with respect to the set-inclusion relation $\subseteq$. We call the pair $(\underline{apr}_A(X), \overline{apr}_A(X))$ the rough set induced by X.

The definability of a set depends on a description language. In the definition, $\mathrm{DEF}_A(T)$ denotes the family of all definable sets with respect to a description language constructed from a subset of attributes $A \subseteq AT$. Definition 3 is a conceptual definition of rough set approximations. It provides a clear interpretation and a conceptual understanding of approximations. By Definition 3, X lies within its lower and upper approximations:

$$\underline{apr}_A(X) \subseteq X \subseteq \overline{apr}_A(X). \tag{11}$$

Furthermore, with respect to the set-inclusion relation $\subseteq$, the lower approximation is the definable set closest to X from below and the upper approximation is the definable set closest to X from above, that is,

$$\begin{aligned} &Y \in \mathrm{DEF}_A(T) \wedge \underline{apr}_A(X) \subseteq Y \subseteq X \Rightarrow Y = \underline{apr}_A(X), \\ &Z \in \mathrm{DEF}_A(T) \wedge X \subseteq Z \subseteq \overline{apr}_A(X) \Rightarrow Z = \overline{apr}_A(X). \end{aligned} \tag{12}$$

In other words, there does not exist another definable set between $\underline{apr}_A(X)$ and X, nor a definable set between X and $\overline{apr}_A(X)$.

One way to justify the meaningfulness of rough set approximations given in Definition 3 is to examine their consistency with the definability of sets. From the definition, one can easily verify the following equivalences:

$$\begin{aligned} X \in \mathrm{DEF}_A(T) &\iff \underline{apr}_A(X) = \overline{apr}_A(X) = X, \\ X \notin \mathrm{DEF}_A(T) &\iff \underline{apr}_A(X) \neq \overline{apr}_A(X) \\ &\iff \underline{apr}_A(X) \neq X \\ &\iff \overline{apr}_A(X) \neq X. \end{aligned} \tag{13}$$

That is, both lower and upper approximations of a definable set are the set itself. The lower and upper approximations of an undefinable set is different and both of them are different from the set. Thus, the approximations are indeed consistent with the definability of sets and, hence, are meaningful.

From Eqs. (11) and (13) and Definition 3, one can verify that rough set approximations have the following additional properties [38,40]:

$$\begin{aligned} \text{(i)}\ & \underline{apr}_A(X) = (\overline{apr}_A(X^c))^c, \\ & \overline{apr}_A(X) = (\underline{apr}_A(X^c))^c, \\ \text{(ii)}\ & X \subseteq Y \Rightarrow \underline{apr}_A(X) \subseteq \underline{apr}_A(Y), \\ & X \subseteq Y \Rightarrow \overline{apr}_A(X) \subseteq \overline{apr}_A(Y), \\ \text{(iii)}\ & \underline{apr}_A(X \cap Y) = \underline{apr}_A(X) \cap \underline{apr}_A(Y), \\ & \overline{apr}_A(X \cup Y) = \overline{apr}v(X) \cup \overline{apr}_A(Y), \\ \text{(iv)}\ & \underline{apr}_A(\underline{apr}_A(X)) = \underline{apr}_A(X), \\ & \underline{apr}_A(\overline{apr}_A(X)) = \overline{apr}_A(X), \\ & \overline{apr}_A(\underline{apr}_A(X)) = \underline{apr}_A(X), \\ & \overline{apr}_A(\overline{apr}_A(X)) = \overline{apr}_A(X), \end{aligned}$$

where $(\cdot)^c$ denotes the complement of a set. According to properties in (i), the two approximations are dual to each other with respect to set complement. It is sufficient to define only one of them. Properties in (ii) state that both approximations are monotonic with respect to the set-inclusion relation. Properties in (iii) imply the corresponding properties in (ii). It is interesting to note that these two properties provide rules for computing the lower approximation of the intersection two sets and the upper approximation of the union of two sets from their approximations, respectively. In this case, the conceptual formulation does provide a way for computing approximations for some sets. However, one cannot compute the upper approximation of the intersection of two sets nor the lower approximation of the union of two sets. Properties in (iv) follow from the conceptual definition of approximations. By definition, both $\underline{apr}_A(X)$ and $\overline{apr}_A(X)$ are definable sets. According to Eq. (13), lower and upper approximations of a definable set are the set itself. Thus, all properties in (iv) hold. The conceptual formulation not only captures the meaning of rough set approximations but also makes it easier to prove some basic properties of rough set approximations.

For a finite universe of objects, the greatest definable set contained in X and the least definable set containing X are given by the union of all definable sets contained in X and the intersection of all definable sets containing X, respectively. Thus, we have the following equivalent definition suggested by Bonikowski [4].

Definition 4 (*Subsystem-based definition*). For a subset $X \subseteq OB$, the lower and upper approximations of X are defined by:

$$\begin{aligned} \underline{apr}_A(X) &= \bigcup \{Y \in \mathrm{DEF}_A(T) | Y \subseteq X\}, \\ \overline{apr}_A(X) &= \bigcap \{Z \in \mathrm{DEF}_A(T) | X \subseteq Z\}. \end{aligned} \tag{14}$$

By observing that the family of definable sets $\mathrm{DEF}(T)$ is a subsystem of $P(OB)$, Yao [55] calls Eq. (14) a subsystem-based definition of rough sets. This definition explicitly gives another method to construct rough set approximations. It provides a slightly different angle for interpreting and understanding Definition 3.

An alternative approach to define and interpret rough set approximations is based on three regions. Given a subset $X \subseteq OB$, we can divide the universe U into three pair-wise disjoint regions, namely, the positive, negative and boundary regions, as follows:

$$\begin{aligned} \mathrm{POS}_A(X) &= \text{the greatest definable set in } \mathrm{DEF}_A(T) \text{ contained by } X, \\ \mathrm{NEG}_A(X) &= \text{the greatest definable set in } \mathrm{DEF}_A(T) \text{ contained by } X^c, \\ \mathrm{BND}_A(X) &= (\mathrm{POS}_A(X) \cup \mathrm{NEG}_A(X))^c. \end{aligned} \tag{15}$$

All three regions are definable sets. In this way, a set is approximated by three pair-wise disjoint definable sets; some of them may be the empty set $\emptyset$.

The pair of lower and upper approximations and the three regions are two different, but mathematically equivalent, forms of rough set approximations. They determine each other as follows:

$$\begin{aligned} \mathrm{POS}_A(X) &= \underline{apr}_A(X), \\ \mathrm{NEG}_A(X) &= \underline{apr}_A(X^c) = (\overline{apr}_A(X))^c, \\ \mathrm{BND}_A(X) &= \overline{apr}_A(X) - \underline{apr}_A(X), \end{aligned} \tag{16}$$

and

$$\begin{aligned} \underline{apr}_A(X) &= \mathrm{POS}_A(X), \\ \overline{apr}_A(X) &= \mathrm{POS}_A(X) \cup \mathrm{BND}_A(X). \end{aligned} \tag{17}$$

One may build a theory of rough sets based on either the pair of approximations or three pair-wise disjoint regions.

Each formulation has its advantages. The pair of approximations explicitly gives the range within which the set lies. The formulation based on three regions has led to the introduction of a theory of three-way decisions [57,58]. However, it may be emphasized that the theory of three-way decisions goes beyond rough sets [15,52] and has applications in decision-theoretic rough sets [17,18,20,21,24,27,57], logic [6,7,44], user preference modeling [10], decision-making [1,2,19,42], subjective decision support systems [9], classification and cluster analysis [23,25,50,62,64], approximation of fuzzy sets [8], text processing [66] and many others [63,65].

In contrast to many existing studies, definability is treated as a primitive notion and approximations are a derived notion in our conceptual formulation. We can clearly see the motivations and reasons for introducing rough set approximations. The meaningfulness of rough set approximations is explicitly stated. Consequently, one can have an insightful and deep conceptual understanding of rough set theory.

4. Computational definitions of approximations

The conceptual Definition 3 provides a semantically sound interpretation of rough set approximations. However, it is computationally difficult to construct the approximations of a set by using the definition directly. In this section, we present a computational formulation for efficient construction of approximations and show its equivalence to the conceptual formulation.

4.1. The family of definable sets and the Boolean algebra induced by an equivalence relation

To design a computationally efficient method for constructing rough set approximations, we need to search for computational definitions of definable sets and the family of definable sets. This is accomplished by using an equivalence relation defined by a subset of attributes and the induced partition of the universe.

A binary relation on OB is called an equivalence relation if it is reflexive, symmetric and transitive. In an information table, one can construct an equivalence relation by using a subset of attributes [40]. For a subset of attributes $A \subseteq AT$, we define an equivalence relation E_A as follows:

$$xE_Ay \iff \forall a \in A(I_a(x) = I_a(y)). \tag{18}$$

The equivalence class containing x is given by:

$$[x]_A = \{y \in OB | xE_Ay\}. \tag{19}$$

By definition, it follows that, for $A \subseteq AT$,

$$E_A = \bigcap_{a \in A} E_{\{a\}},$$
$$[x]_A = \bigcap_{a \in A} [x]_{\{a\}}. \tag{20}$$

That is, we can construct the equivalence relation induced by a set of attributes A by using the family of equivalence relations induced by individual attributes in A.

For a set of attributes $A \subseteq AT$, let OB/E_A be the partition of OB consisting of all equivalence classes of E_A, namely, a family of nonempty and pairwise disjoint subsets whose union is the universe. One can construct an atomic sub-Boolean algebra $B(OB/E_A)$ of $P(OB)$ with OB/E_A as the set of atoms:

$$B(OB/E_A) = \{\bigcup F | F \subseteq OB/E_A\}. \tag{21}$$

Each element in $B(OB/E_A)$ is the union of a family of equivalence classes. The Boolean algebra $B(OB/E_A)$ contains the empty set $\emptyset$, the whole set OB, and is closed under set complement, intersection, and union. The three notions of an equivalence relation E_A, the partition OB/E_A and the atomic Boolean algebra $B(OB/E_A)$ uniquely determine each other. We can therefore use them interchangeably.

The equivalence of the two structures $B(OB/E_A)$ and $\mathrm{DEF}_A(T)$ of an information table T can be demonstrated as follows.

First, we show that any set in the atomic Boolean algebra $B(OB/E_A)$ is definable. Consider a subset of attributes $A \subseteq AT$, an object x can be described by a logic formula:

$$\bigwedge_{a \in A} \langle a, I_a(x) \rangle, \tag{22}$$

where $I_a(x) \in V_a$ is the value of x on attribute a and the atomic formula $\langle a, I_a(x) \rangle$ indicates that the value of an object on attribute a is $I_a(x)$. By the definition of the equivalence relation E_A in Eq. (18), we know that all and only those objects in $[x]_{E_A}$ have the same description as x with respect to the subset of attribute $A \subseteq AT$. In other words, $[x]_{E_A}$ is the set of objects satisfying the formula $\bigwedge_{a \in A} \langle a, I_a(x) \rangle$:

$$\| \bigwedge_{a \in A} \langle a, I_a(x) \rangle \| = [x]_{E_A}. \tag{23}$$

That is, $[x]_{E_A}$ is a definable set. Furthermore, any proper subset of $[x]_{E_A}$ is undefinable with respect to DL_A and, hence, $[x]_{E_A}$ is a minimal nonempty definable set. According to the relationship between logic disjunction $\vee$ and set union $\cup$, i.e., (m3) in Eq. (7), we can conclude that the union of a family of equivalence classes is a definable set. Thus, any set in the atomic Boolean algebra $B(OB/E_A)$ is a definable set.

Conversely, we can show that any definable set in $\mathrm{DEF}_A(T)$ is in the atomic Boolean algebra $B(OB/E_A)$. With respect to a finite set of objects and a finite set of attributes, for each attribute $a \in AT$ we have:

$$\| \bigvee_{v \in V_a} \langle a, v \rangle \| = OB. \tag{24}$$

For a definable set in $\mathrm{DEF}_A(T)$ with $X = \|p\|, p \in DL_A$, it follows that,

$$\|(p) \wedge (\bigvee_{v \in V_a} \langle a, v \rangle)\| = \|p\| \cap \|(\bigvee_{v \in V_a} \langle a, v \rangle)\| = \|p\|. \tag{25}$$

Thus, we can introduce an attribute $a \in A$ into p if p does not use a. After introducing all attributes in A but not in p, we transform the formula into the disjunction-of-conjunction form. We then remove all those conjunctions whose meaning sets are the empty set. The resulting formula has the same meaning set as p. Furthermore, each conjunction in the formula defines an equivalence class of E_A. Therefore, the definable set X can be expressed as the union of a family of equivalence classes and is a member of $B(OB/E_A)$. Thus, any definable set is in the atomic Boolean algebra $B(OB/E_A)$.

From the discussion, we arrive at the following fundamental theorem that connects conceptual and computational definitions of rough set approximations [28,56].

Theorem 1. *In an information table T, for a nonempty set of attributes $\emptyset \neq A \subseteq AT$, $\mathrm{DEF}_A(T) = B(OB/E_A)$.*

An equivalence class $[x]_A$ is a minimal nonempty set of definable set. It follows that, for a definable set $Y \in \mathrm{DEF}_A(T)$, we have:

$$[x]_A \subseteq Y \iff [x]_A \cap Y \neq \emptyset. \tag{26}$$

This immediately leads to the next corollary.

Corollary 1. *A definable set $Y \in \mathrm{DEF}_A(T)$ can be expressed as the union of a family of equivalence classes:*

$$Y = \bigcup\{[x]_A \in OB/E_A | [x]_A \subseteq Y\} = \bigcup\{[x]_A \in OB/E_A | [x]_A \cap Y \neq \emptyset\}.$$

Although the description language DL does not include logic negation, it is sufficient for the study of the two structures used in Theorem 1, regarding an information table with a finite set of objects and a finite set of attributes. In fact, one may define the negation of an atomic formula as $\neg\langle a, v \rangle = \bigvee\{\langle a, v' \rangle | v' \in V_a, v' \neq v\}$ and interpret it as $\|\neg\langle a, v \rangle\| = (\|\langle a, v \rangle\|)^c$. One can explain the negation of any formula in terms of the negation of atomic formulas by using De Morgan's laws.

4.2. Computational definitions of approximations

By exploring the equivalence of $\mathrm{DEF}_A(T)$ and $B(OB/E_A)$, we can derive computational definitions of rough set approximations by using the equivalence classes in OB/E_A.

According to Definition 3, $\underline{apr}_A(X)$ is the greatest definable set contained in X, that is, $\underline{apr}_A(X) \subseteq X$. It follows that, for any equivalence class $[x]_A$,

$$[x]_A \subseteq \underline{apr}_A(X) \Rightarrow [x]_A \subseteq X. \tag{27}$$

From the the fact that $[x]_A$ is a definable set and $\underline{apr}_A(X)$ is the greatest definable set contained in X, we can also conclude that,

$$[x]_A \subseteq X \Rightarrow [x]_A \subseteq \underline{apr}_A(X). \tag{28}$$

Thus,

$$[x]_A \subseteq X \iff [x]_A \subseteq \underline{apr}_A(X). \tag{29}$$

By Corollary 1, we can express the lower approximation by a union of a family of equivalence classes:

$$\underline{apr}_A(X) = \bigcup\{[x]_A \in OB/E_A | [x]_A \subseteq \underline{apr}_A(X)\}. \tag{30}$$

By combining Eqs. (29) and (30), we have another expression of the lower approximation:

$$\underline{apr}_A(X) = \bigcup\{[x]_A \in OB/E_A | [x]_A \subseteq X\}. \tag{31}$$

Similarly, by the definition that the upper approximation $\overline{apr}_A(X)$ is the least definable set containing X and the fact that an equivalence class is a minimal nonempty definable set, we have:

$$[x]_A \cap X \neq \emptyset \iff [x]_A \cap \overline{apr}_A(X) \neq \emptyset. \tag{32}$$

According to Corollary 1, we have:

$$\overline{apr}_A(X) = \bigcup\{[x]_A \in OB/E_A | [x]_A \cap \overline{apr}_A(X) \neq \emptyset\} = \bigcup\{[x]_A \in OB/E_A | [x]_A \cap X \neq \emptyset\}. \tag{33}$$

From the discussions, we derive a computational definition of rough set approximations [40].

Definition 5 (*Granule-based definition*). For a subset $X \subseteq OB$, the lower and upper approximations of X are defined by:

$$\begin{aligned} \underline{apr}_A(X) &= \bigcup\{[x]_A \in OB/E_A | [x]_A \subseteq X\}, \\ \overline{apr}_A(X) &= \bigcup\{[x]_A \in OB/E_A | [x]_A \cap X \neq \emptyset\}. \end{aligned} \tag{34}$$

By interpreting an equivalence class $[x]_A$ as a granule, Yao [55] calls this definition a granule-based definition. The definition offers a computationally efficient way to construct rough set approximations. If we focus on objects in an equivalence class, we have:

$$\begin{aligned} x \in \underline{apr}_A(X) &\iff [x]_A \subseteq X, \\ x \in \overline{apr}_A(X) &\iff [x]_A \cap X \neq \emptyset. \end{aligned} \tag{35}$$

This produces an element-based definition [40,55].

Definition 6 (*Element-based definition*). For a subset $X \subseteq OB$, the lower and upper approximations of X are defined by:

$$\begin{aligned} \underline{apr}_A(X) &= \{x \in OB | [x]_A \subseteq X\}, \\ \overline{apr}_A(X) &= \{x \in OB | [x]_A \cap X \neq \emptyset\}. \end{aligned} \tag{36}$$

Although the subsystem-based, granule-based, and element-based definitions are mathematically equivalent, they offer different hints when we generalize Pawlak rough sets. The element-based definition enables us to establish a connection between rough sets and modal logics, offering a direction for generalizing rough sets by using non-equivalence relations [53]. The granule-based definition connects rough sets and granular computing [54], offering another direction for generalizing rough sets by using coverings [53]. The subsystem-based definition can be used to generalize rough sets with other mathematical structures such as Boolean algebra, lattices, topological spaces, closure systems, and posets [55].

In the literature of rough sets, there are two more methods to connect the notions of definability and approximations. One method assumes that, given an equivalence relation E, the union of a family of equivalence classes is definable, or more precisely, E-definable [11,34,39], which leads to the atomic Boolean algebra $B(OB/E)$. Approximations are then defined by using $B(OB/E)$ to replace $\mathrm{DEF}_A(T)$ in Definition 3. As we have demonstrated earlier, if we use the same subset of attributes to construct an equivalence relation and a description language, respectively, we produce the same family of definable sets from the equivalence relation based definition and the description language based definition of definability. However, it is important to note that we do not have to construct an equivalence relation in order to explain the notion of definability. In addition, the notion of definability defined by using an equivalence relation is also explained in terms of a description language. Another method first defines rough set approximations by using an equivalence relation. A set is said to be definable or describable if its lower and upper approximations are the same [38]. In such a formulation, approximations are used as a primitive notion and definability is a derived notion. The motivations for introducing approximations are not entirely clear. On the other hand, formulations with definability as a primitive notion enable us to see clearly the motivations for introducing rough set approximations.

5. A conceptual understanding of reducts

We give a general definition of reducts of a set [60] and examine three types of reducts in rough set theory.

5.1. A conceptual definition of reducts

The notion of reducts plays an essential role in rough set analysis. To have an in-depth conceptual understanding of reducts, we may search for an explanation and interpretation in a wider context. Given a set or a space, a fundamental question is whether there exists a subset or a subspace that serves the same purpose, has the same power, or gives the same performance, as that of the entire set or space. Such a subset or subspace may be considered as a reduct of the original set or space. A conceptual definition of a reduct of a set is developed based on this intuitive understanding [60].

Suppose S is a finite set and $P(S)$ is the power set of S. Let $\mathbb{P}$ denote a unary predicate on subsets of S, that is, for $A \subseteq S, \mathbb{P}(A)$ stands for the statement that "subset A has property $\mathbb{P}$." The values of $\mathbb{P}$ are computed by an evaluation e with reference to certain available data or information. For example, an evaluation may be defined by using an information table. For a subset $A \subseteq S, \mathbb{P}_e(A)$ is true if A has property $\mathbb{P}$, otherwise, it is false. A conceptual definition of a reduct is defined based on an evaluation.

Definition 7 (*Subset-based definition*). Given an evaluation e of $\mathbb{P}$, if a subset $R \subseteq S$ satisfies the following conditions:

$$\begin{aligned} &\text{existence}: \quad \mathbb{P}_e(S), \\ &\text{sufficiency}: \quad \mathbb{P}_e(R), \\ &\text{minimization}: \quad \forall B \subsetneq R(\neg \mathbb{P}_e(B)), \end{aligned}$$

we call R a reduct of S.

The three conditions reflect the fundamental characteristics of a reduct. The condition of existence requires that the whole set S must have the property $\mathbb{P}$. It ensures that a reduct of S exists, as S itself is a candidate of a reduct. In many studies, this condition is implicitly assumed or embedded in $\mathbb{P}$. We explicitly state the condition of existence, in order to better capture the meaning of a reduct. The condition of sufficiency requires that a reduct R of S is sufficient for preserving property $\mathbb{P}$ of S. The condition of minimization requires that a reduct is a minimal subset of S having property $\mathbb{P}$ in the sense that none of the proper subsets of R has the property. Since we need to check all subsets of R, we call this definition a subset-based definition. In general, there may exist more than one reduct.

According to Definition 7, one must check all proper subsets of R in order to verify whether R is a reduct. For an arbitrary property, this is unavoidable. Fortunately, for certain classes of properties one does not have to verify all proper subsets. One of such classes is examined in the next subsection.

5.2. Reducts under properties with monotonicity

In rough set analysis, one often considers a class of properties satisfying a condition of monotonicity with respect to set inclusion.

Definition 8. A predicate $\mathbb{P}$ is monotonic with respect to set inclusion if it satisfies the following property:

$$\forall A, B \subseteq S (A \subseteq B \Rightarrow (\mathbb{P}(A) \Rightarrow \mathbb{P}(B))). \tag{37}$$

That is, if a subset has a property, then a superset of it also has the property.

It is important to point out that the monotonicity of $\mathbb{P}$ is defined based on all possible evaluations. That is, the monotonicity must hold for all possible evaluations. For example, in rough set analysis, an information table determines an evaluation and all possible tables determine all possible evaluations. The monotonicity of a predicate must hold for all possible information tables.

The monotonicity can be equivalently re-expressed as:

$$\forall A, B \subseteq S (A \subseteq B \Rightarrow (\neg\mathbb{P}(B) \Rightarrow \neg\mathbb{P}(A))). \tag{38}$$

That is, if a set does not have a property, then none of its subsets has the property. Once we know that a set does not have the property, we can conclude that all its subsets do not have the property, without the need to further check the subsets. By monotonicity, the condition of minimization in Definition 7 can be equivalently expressed as:

$$\forall B \subsetneq R(\neg\mathbb{P}_e(B)) \iff \forall a \in R(\neg\mathbb{P}_e(R - \{a\})). \tag{39}$$

Thus, instead of checking all proper subsets of R, we only need to check all maximal proper subsets of R obtained by removing a single element from R. Since there is a one-to-one correspondence between all maximal proper subsets of R and all elements of R, we can also focus on elements of R. For a subset $A \subseteq S$, an element $a \in A$ is redundant in A if $\mathbb{P}_e(A - \{a\})$ and is necessary if $\neg\mathbb{P}_e(A - \{a\})$. The condition $\forall a \in R(\neg\mathbb{P}_e(R - \{a\}))$, in fact, states that every element of a reduct R is necessary. This leads to an element-based definition of reducts.

Definition 9 (*Element-based definition*). Suppose $\mathbb{P}$ satisfies monotonicity. Given an evaluation e of $\mathbb{P}$, if a subset $R \subseteq S$ satisfies the following conditions:

$$\begin{aligned}
&\text{existence}: \quad \mathbb{P}_e(S),\\
&\text{joint sufficiency}: \quad \mathbb{P}_e(R),\\
&\text{individual necessity}: \quad \forall a \in R(\neg\mathbb{P}_e(R - \{a\})),
\end{aligned}$$

we call R is a reduct of S.

By focusing on elements of R, the second condition states that elements of R are jointly sufficient for preserving $\mathbb{P}$ and the third condition states that elements of R are individually necessary. One can neither add another element from S to R nor delete an element from R. This interpretation, in terms of adding and deleting elements, brings us closer to a computationally efficient method for constructing a reduct. It also provides a plausible explanation for the common use of the element-based definition in the literature of rough sets. However, the element-based definition is only a special case of the subset-based definition, as the former requires the assumption of the monotonicity of $\mathbb{P}$. In many existing studies, the assumption of monotonicity is not made explicitly. By using two definitions, we make explicit the underlying assumption of the element-based definition.

6. Reducts and Pawlak rough set analysis

Yao and Fu [60] re-interpret Pawlak rough set analysis in terms of constructing three types of reducts in three steps. As another demonstration of the power of conceptual definitions and formulations, we review the main results in this section.

6.1. Attribute reducts

In an information table, a subset of attributes induces a family of definable sets. The family is in fact an atomic Boolean algebra generated from the partition of the equivalence relation defined by the same subset of attributes. An attribute reduct is a minimal subset of attributes having the same expressive power as the entire set of attributes. To apply Definition 7, we have the following setting:

$$\begin{aligned}
&S: \quad AT,\\
&\mathbb{P}_e: \quad \mathrm{DEF}_A(T) = \mathrm{DEF}_{AT}(T), A \subseteq AT, \text{or equivalently},\\
&E_A = E_{AT}, A \subseteq AT.
\end{aligned} \tag{40}$$

We immediately have a conceptual definition of an attribute reduct.

Definition 10. In an information table T, if a subset of attributes $R \subseteq AT$ satisfies the following conditions:

$$\begin{aligned}
&\text{existence}: \quad \mathrm{DEF}_{AT}(T) = \mathrm{DEF}_{AT}(T),\\
&\text{sufficiency}: \quad \mathrm{DEF}_R(T) = \mathrm{DEF}_{AT}(T),\\
&\text{minimization}: \quad \forall R' \subsetneq R(\neg(\mathrm{DEF}_{R'}(T) = \mathrm{DEF}_{AT}(T))),
\end{aligned}$$

we call R an attribute reduct of AT.

The condition of existence trivially holds and is not explicitly used in many existing studies. All three conditions can also be expressed in terms of equivalence relations. For example, the condition of minimization can also be expressed in commonly used equivalent conditions [40]:

$$\begin{aligned}
&\forall R' \subsetneq R(E_{R'} \neq E_{AT}),\\
&\forall a \in R(E_{R-\{a\}} \neq E_{AT}).
\end{aligned} \tag{41}$$

The second condition is a result of the monotonicity of $\mathbb{P}$.

A special class of information tables is called classification or decision tables [40]. A decision table consists of a set of condition attributes C and another set of decision attributes D, that is, $AT = C \cup D$ and $C \cap D = \emptyset$. For a decision table T, decision classes of D, namely, equivalence classes induced by the equivalence relation E_D, are a family of concepts to be approximated by definable concepts in $\mathrm{DEF}_C(T)$. The set of objects with a deterministic classification is the union of the lower approximations of decision classes of D:

$$\mathrm{POS}_C(OB/E_D) = \bigcup\{\underline{apr}_C(K) | K \in OB/E_D\}, \tag{42}$$

where $\underline{apr}_C(K)$ is the lower approximation of K by using the set of condition attributes C, that is,

$$\underline{apr}_C(K) = \{x \in OB | [x]_C \subseteq K\}. \tag{43}$$

A relative attribute reduct of C with respect to D is a minimal subset of condition attributes that produces the same set of objects with deterministic classification as the entire set of condition attributes. We have the following setting:

$$\begin{aligned} S :\quad & C, \\ \mathbb{P}_e :\quad & \mathrm{POS}_A(OB/E_D) = \mathrm{POS}_C(OB/E_D), \quad A \subseteq C. \end{aligned} \tag{44}$$

By Definition 7, we immediately have a conceptual definition of a relative attribute reduct.

Definition 11. In a decision table T with $AT = C \cup D$, if a subset of condition attributes $R \subseteq C$ satisfies the following conditions:

$$\begin{aligned} &\text{existence}: \quad \mathrm{POS}_C(OB/E_D) = \mathrm{POS}_C(OB/E_D) \\ &\text{sufficiency}: \quad \mathrm{POS}_R(OB/E_D) = \mathrm{POS}_C(OB/E_D) \\ &\text{minimization}: \quad \forall R' \subsetneq R(\neg(\mathrm{POS}_{R'}(OB/E_D) = \mathrm{POS}_C(OB/E_D))), \end{aligned}$$

we call R a relative attribute reduct of C.

Again, the condition of existence trivially holds. By the monotonicity of $\mathbb{P}$, we can write the condition of minimization as $\forall a \in R(\mathrm{POS}_{R-\{a\}}(OB/E_D) \neq \mathrm{POS}_C(OB/E_D))$. In rough set analysis, there exist several other definitions of a relative attribute reduct. All of them can be similarly expressed in the general form given by Definition 7.

6.2. Attribute–value pair reducts

In rough set analysis, a classification rule in a decision table T with $AT = C \cup D$ can be expressed in the following form:

$$\bigwedge_{\langle a, v_a\rangle \in L} \langle a, v_a\rangle \rightarrow q, \tag{45}$$

where L is a set of attribute–value pairs or atomic formulas with each attribute appears at most once. That is, the left-hand-side of the rule is a conjunctive formula of the language DL_C and q is a conjunctive formula of DL_D of the form $\bigwedge_{d \in D}\langle d, v_d\rangle$. For the task of rule simplification, we only need to consider the structure of the left-hand-side of a rule and simply use q for representing the right-hand-side without considering its structure.

The notion of an attribute–value reduct of a classification rule concerns a minimal subset of attribute–value pairs from the left-hand-side of a rule that preserves the correctness of the original rule. With reference to Definition 7, we have the following setting:

$$\begin{aligned} S :\quad & L, \\ \mathbb{P}_e :\quad & \|\bigwedge_{\langle a, v_a\rangle \in A} \langle a, v_a\rangle\| \subseteq \|q\|, \quad A \subseteq L. \end{aligned} \tag{46}$$

We thus have a conceptual definition of an attribute–value pair reduct for a classification rule.

Definition 12. In a classification rule $\bigwedge_{\langle a, v_a\rangle \in L}\langle a, v_a\rangle \rightarrow q$, if a subset of attribute–value pairs $R \subseteq L$ satisfies the following conditions:

$$\begin{aligned} &\text{existence}: \quad \|\bigwedge_{\langle a, v_a\rangle \in L} \langle a, v_a\rangle\| \subseteq \|q\|, \\ &\text{sufficiency}: \quad \|\bigwedge_{\langle a, v_a\rangle \in R} \langle a, v_a\rangle\| \subseteq \|q\|, \\ &\text{minimization}: \quad \forall R' \subsetneq R(\neg(\|\bigwedge_{\langle a, v_a\rangle \in R'} \langle a, v_a\rangle\| \subseteq \|q\|)), \end{aligned}$$

we call R an attribute–value pair reduct of L for the classification rule.

In this case, the condition of existence is necessary, which requires that the rule itself is correct. That is, an attribute–value pair reduct can preserve the property of correctness only if the original rule is correct. By the monotonicity of $\mathbb{P}$, we can derive other definitions of an attribute–value reduct.

An attribute–value pair reduct is in fact a minimal complex used by Grzymala-Busse [12]. For simplifying the left-hand-side of a rule, instead of using a set of attribute–value pairs, Pawlak [40] uses the family of their meaning sets:

$$ms(L) = \{\|\langle a, v_a\rangle\| | \langle a, v_a\rangle \in L\}. \tag{47}$$

For a subset $A \subseteq L$, we can construct a family of subsets of objects

$$ms(A) = \{\|\langle a, v_a\rangle\| | \langle a, v_a\rangle \in A\}. \tag{48}$$

By using the family $ms(L)$, we can transform logic conjunction in Definition 12 into set intersection. For example, the condition of sufficiency is expressed as:

$$\bigcap_{\|\langle a, v_a\rangle\| \in ms(R)} \|\langle a, v_a\rangle\| \subseteq \|q\| \tag{49}$$

The result is an $\cap$-reduct of the family $ms(L)$ introduced by Pawlak [40]. In this way, an attribute–value pair reduct of L corresponds to an $\cap$-reduct of the the family $ms(L)$ of subsets of objects [60].

6.3. Rule reducts

Let Φ denote a set of classification rules of the form $p \rightarrow q$ with $\|p\| \subseteq \|q\|$. Each rule covers a subset of objects $\|p\| \subseteq OB$. The family of subsets of objects covered by all rules in Φ is given by:

$$ms(\Phi) = \{\|p\| | p \rightarrow q \in \Phi\}. \tag{50}$$

Intuitively speaking, a rule reduct is a minimal subset of rules that covers the same set of objects as Φ. For defining a rule reduct, we have the following setting:

$$\begin{aligned} S :\quad & \Phi, \\ \mathbb{P}_e :\quad & \bigcup ms(A) = \bigcup ms(\Phi), A \subseteq \Phi, \end{aligned} \tag{51}$$

where $ms(A) = \{\|p\| | p \rightarrow q \in A\}$. According to Definition 7, we have a conceptual definition of a rule reduct.

Definition 13. Suppose Φ is a set of classification rules. If a subset of rules $R \subseteq \Phi$ satisfies the following conditions:

$$\begin{aligned} &\text{existence}: \quad \bigcup ms(\Phi) = \bigcup ms(\Phi), \\ &\text{sufficiency}: \quad \bigcup ms(R) = \bigcup ms(\Phi), \\ &\text{minimization}: \quad \forall R' \subsetneq R(\neg(\bigcup ms(R') = \bigcup ms(\Phi))), \end{aligned}$$

we call R a rule reduct of Φ.

The condition of existence trivially holds. From the monotonicity of $\mathbb{P}$, one can obtain other definitions of a rule reduct by using different versions of the condition of minimization. A rule reduct of Φ corresponds to an $\cup$-reduct of the family $ms(\Phi)$ of subsets of objects introduced by Pawlak [40,60].

6.4. Pawlak three-step analysis of a decision table

The discussions so far have demonstrated that a conceptual definition of reducts offers a deep understanding of the notion of reducts. As a consequence, we are able to unify three types of reducts within a common framework, instead of three different models in existing studies. This unification further leads to a simple explanation of Pawlak rough set analysis as a three-step process, with each step focusing on a specific type of reducts.

Step 1: Construction of a relative attribute reduct $R_{at} \subseteq C$ of a decision table T with $AT = C \cup D$. Based on R_{at}, one can construct a family of classification rules: for each $x \in \mathrm{POS}_{R_{at}}(OB/E_D)$,

$$\bigwedge\{\langle a, I_a(x)\rangle | a \in R_{at}\} \rightarrow \bigwedge\{\langle d, I_d(x)\rangle | d \in D\}. \quad (52)$$

By the definition of a relative attribute reduct, it follows that R_{at} produces the same deterministic classification as the entire set of condition attributes and none of the attributes in R_{at} can be removed.

Step 2: Construction of an attribute–value pair reduct for each classification rule obtained in Step 1. For a rule induced by an object x, the set of attribute–value pairs is given by $L(x) = \{\langle a, I_a(x)\rangle | a \in R_{at}\}$. Let $R_{av}(x) \subseteq L(x)$ denote an attribute–value pair reduct of $L(x)$. We have a simplified/reduced classification rule:

$$\bigwedge\{\langle a, I_a(x)\rangle | \langle a, I_a(x)\rangle \in R_{av}(x)\} \rightarrow \bigwedge\{\langle d, I_d(x)\rangle | d \in D\rangle\}. \quad (53)$$

By collecting simplified rules for all $x \in \mathrm{POS}_{R_{at}}(OB/E_D)$, we obtain a set of rules Φ. Each simplified rule may cover more objects than the original rule. The set of simplified rules covers objects that are covered by the original set of rules.

Step 3: Construction of a rule reduct. From the set Φ of simplified rules, we can construct a rule reduct $R_r \subseteq \Phi$. By the definition of a rule reduct, R_r covers the same set of objects as Φ and, furthermore, no rule in R_r can be removed.

After the three steps, we have a minimal set of rules in the sense that no rules in the set can be removed in order to cover the same set of objects. In addition, the left-hand-side of each rule is a minimal set of attribute–value pairs in the sense that no attribute–value pair can be removed in order to keep the correctness of the rule. Finally, it should be commented that our explanation of Pawlak three-step process is a conceptual one, focusing on the meaning of various notions. Algorithms for constructing a reduct are discussed in the next section.

7. Reduct construction algorithms

If a property $\mathbb{P}$ is not monotonic with respect to set inclusion, it is impossible to have a computationally efficient method. In this section, we will consider the class of properties with monotonicity. A conceptual understanding of a reduct offers a unified framework for defining and interpreting a reduct. To further demonstrate the value of the unified framework, we describe generic algorithms for reduct construction based on three search strategies [61].

7.1. Algorithms based on a deletion strategy

Suppose a property $\mathbb{P}$ defined on the power set of a finite set S is monotonic with respect to set inclusion. Given an evaluation e, we assume that the entire set S satisfies $\mathbb{P}$, that is, $\mathbb{P}_e(S)$ is true. However, some elements of S may not be necessary. Reduct construction algorithms based on a deletion strategy search for a reduct by successively deleting redundant elements.

Consider an element $a \in S$. We may check whether the element a can be deleted from S according to conditions in Definition 9. If $\neg\mathbb{P}_e(S - \{a\})$, we cannot delete a from S when searching for a reduct of S, as the remaining set of elements, $S - \{a\}$, can no longer preserve property $\mathbb{P}$. On the other hand, if $\mathbb{P}_e(S - \{a\})$, we can delete a from S. The new set $\mathbb{P}_e(S - \{a\})$ may still contain redundant elements and they must be further removed. Starting from S, we can obtain a reduct by successively verifying every element and deleting redundant elements. Algorithm 1 constructs a reduct based on a deletion search strategy. In the algorithm, CD stands for the set of candidate elements for deletion, which is set to be the entire set S at the beginning and is updated once an element is verified. The algorithm sets R to S and then successively verifies and deletes elements from R, if needed, until $CD = \emptyset$, that is, all elements of S are verified.

Algorithm 1. An algorithm based on a deletion strategy

input: A finite set S and an evaluation e of $\mathbb{P}$.
output: A reduct $R \subseteq S$.

Deletion:
(1) **let** $R = S, CD = S$;
(2) **while** $CD \neq \emptyset$ **do**
  (2.1) **select** an element a from CD, **let** $CD = CD - \{a\}$;
  (2.2) **if** $\mathbb{P}_e(R - \{a\})$ **then let** $R = R - \{a\}$;
(3) Output R.

We have several comments about the algorithm. First, the algorithm is generic in the sense that it works for any type of reducts. By applying different settings of S and $\mathbb{P}$, we can compute different types of reducts, including attribute reducts, attribute–value pair reducts, and rule reducts [12,40]. Second, the algorithm only describes the control structure during a search for a reduct. Step (2.1) does not specify how to select an element from CD. Typically, one may design different heuristics [22,45] for selecting a suitable element for deletion. Since one has to verify every element in S to find a reduct, the use of heuristics is only meaningful when one attempts to find an approximate reduct instead of a reduct. Third, the algorithm requires the examination of all elements of S. This may be computationally expensive, particular in the case when the size of a reduct is much smaller than the size of S.

7.2. Algorithms based on an addition-deletion strategy

To resolve the problem of verifying all elements of S in a deletion based algorithm, one may consider a reverse process. By starting with the empty set $\emptyset$, we can successively add elements until we have a set that is jointly sufficient for preserving $\mathbb{P}$. Since such a set may contain redundant elements, we need to re-apply the deletion procedure of a deletion based algorithm. Algorithm 2 constructs a reduct based on an addition-deletion strategy. In the algorithm, CA denotes the set of candidate elements for addition, which is set to be S at the beginning.

Algorithm 2. An algorithm based on an addition-deletion strategy

input: A finite set S and an evaluation e of $\mathbb{P}$.
output: A reduct $R \subseteq S$.

Addition:
(1) **let** $R = \emptyset, CA = S$;
(2) **while** $\neg\mathbb{P}(R)$ **do**
  (2.1) **select** an element a from CA, **let** $CA = CA - \{a\}$;
  (2.2) **let** $R = R \cup \{a\}$;

Deletion:
(3) **let** $CD = R$;
(4) **while** $CD \neq \emptyset$ **do**
  (4.1) **select** an element a from CD, **let** $CD = CD - \{a\}$;
  (4.2) **if** $\mathbb{P}_e(R - \{a\})$ **then let** $R = R - \{a\}$;
(5) Output R.

The Deletion part of Algorithm 2 is essentially the Algorithm 1, except that we start with a smaller set obtained from the Addition part. The Addition part of Algorithm 2 produces a set that contains a reduct. When the size of a reduct is much smaller than the size of S, we expect that the set produced in the Addition part is also much smaller than S. In such cases, Algorithm 2 works efficiently. Algorithm 1 may be viewed as a special case of Algorithm 2 that uses the largest set S as a result of the Addition part. When the size of a reduct is more than half the size of S, Algorithm 2 will perform poorly, as it requires to scan attributes in a reduct twice.

Again, one may choose different heuristics to select most suitable elements in Steps (2.1) and (4.1). In practical situations, one may omit the deletion procedure to obtain an approximate reduct. If one chooses a good heuristic for selection in Step (2.1), the resulting approximate reduct may be close enough to a reduct.

7.3. Algorithms based on an addition strategy

For an addition-deletion based algorithm, if we can ensure that every element added will be necessary, we can avoid the deletion procedure. Such an algorithm is described as follows. Consider a subset $Q \subseteq S$. If Q is a subset of a reduct of S, we call Q a partial reduct. It is possible that Q may be a subset of more than one reduct. Algorithm 3 constructs a reduct based on an addition strategy.

Algorithm 3. Algorithms based on an addition strategy

input: A set S.
output: A reduct $R \subseteq S$.

Addition:
(1) **let** $R = \emptyset, CA = S$;
(2) **while** $\neg\mathbb{P}(R)$ **do**
  (2.1) **select** an element a form CA, **let** $CA = CA - \{a\}$;
  (2.2) **if** $R \cup \{a\}$ is a partial reduct **then let** $R = R \cup \{a\}$;
(3) Output R.

Algorithm 3 is essentially the Addition part of Algorithm 2, except that every element added in Step (2.2) is necessary. In its present form, an addition based algorithm is only a conceptual algorithm. As an almost perfect mirror image of a deletion based algorithm, it helps us to have an insightful understanding of reduct construction processes. A crucial issue remaining is how to test if a subset of S is a partial reduct, which is not a trivial task. Wang and Wang [49] give an addition based algorithm for constructing an attribute reduct by using the discernibility matrix of a decision table.

7.4. The use of core elements

The notion of core elements plays an important role in rough set analysis and particularly in designing reduct construction algorithms [40].

Definition 14. Suppose that a property $\mathbb{P}$ on the power set $P(S)$ of a set S is monotonic with respect to set inclusion. An element $s \in S$ is called a core element under an evaluation e with $\mathbb{P}_e(S)$ if $\neg(\mathbb{P}_e(S - \{s\}))$.

By definition, property $\mathbb{P}$ does not hold if one removes s. This conceptual definition also provides a method to verify if an element is a core element. The notion of core elements are conceptually very useful for understanding reducts. A core element is essential for preserving property $\mathbb{P}$. Every reduct must contain all core elements and the set of all core elements are the intersection of all reducts [40].

Many studies on reduct construction algorithms in fact use core elements for computational purposes [40]. This typically involves the computation of all core elements, which needs a scan of all elements in S by verifying $\neg(\mathbb{P}_e(S - \{s\}))$. Let $CORE \subseteq S$ be the set of all core elements. Due to the cost of constructing $CORE$, the computational advantages of using core elements need a close examination.

In Algorithm 1, for Step (1), we can set $CD = S - CORE$. On surface, we have a less number of elements to be verified in the **while** loop. However, to find the set of all core elements $CORE$ is computationally more expensive than the saving for finding *one* reduct, as one has to scan every element in S when constructing $CORE$. Thus, for deletion based algorithms, the use of core elements is computationally inferior for finding one reduct. It is only computationally advantageous when we are interested in finding many reducts.

In Algorithms 2 and 3, at Step (1), we may start with $R = CORE$ and $CA = S - CORE$. By using core elements, we ensure that essential elements are included and we may find a set that contains a reduct before adding all elements from $S - CORE$. In addition, we eliminate the need to compare core elements with other elements when selecting an element for addition in Step (2.1). Thus, the use of core elements may be computationally superior for addition-deletion based and addition based algorithms.

8. Conclusion

When studying and applying rough set theory, it is important to realize that there exist two sides of the theory. One is a sound conceptual understanding and the other is an efficient computation. With limited studies on conceptual understanding, we are struggling with a search for a simple and elegant formulation and explanation of rough set theory. We are confused by different expressions of the same notions. It may be necessary to revisit some of the earlier studies that motivated the introduction of rough sets in the first place.

In this paper, we advocate the essentiality of a conceptual formulation and interpretation of rough set theory. To demonstrate the power and value of a conceptual understanding, we examine two fundamental notions of rough sets, namely, approximations and reducts. We give a conceptual definition of rough set approximations, which is a natural consequence of the definability and undefinability of sets. That is, we treat definability as a primitive notion and rough set approximations as a derived notion from the definability. This settles a semantics difficulty of the commonly used computational definitions that treat approximations as a primitive notion and definability as a derived notion.

We give a conceptual definition of reducts that unifies attribute reducts, relative attribute reducts, attribute–value pair reducts, and rule reducts. Consequently, Pawlak rough set analysis can be simply and uniformly explained as a three-step process. In each step, we construct a reduct of a specific type. Furthermore, a generic reduct construction algorithm can be designed, independent of any particular types of reducts.

This paper mainly reviews and summarizes results from existing studies, but recasts them in the light of a conceptual understanding, rather than computational methods. A lot of research work is still needed to realize the full potential of the conceptual side of rough set theory. Conceptual formulations may be useful in forming a solid foundation of rough set theory [5]. Considering the importance of an in-depth understanding from conceptual formulations, further research efforts are worthwhile.

Acknowledgements

This work was supported in part by a Discovery Grant from NSERC, Canada. The author thanks Professor Hamido Fujita for encouragement and constructive discussions. The author is very grateful to reviewers for their valuable and insightful comments.

References

[1] N. Azam, J.T. Yao, Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets, Int. J. Approx. Reason. 55 (2014) 142–155.
[2] N. Azam, J.T. Yao, Game-theoretic rough sets for recommender systems, Knowl.-Based Syst. 72 (2014) 96–107.
[3] J.R. Billif, W.E. Eibble, Conceptual Physics: Matter in Motion, John Wiley & Sons, New York, 1969.
[4] Z. Bonikowski, A certain conception of the calculus of rough sets, Notre Dame J. Formal Logic 33 (1992) 412–421.
[5] M.K. Chakraborty, M. Banerjee, Rough sets: some foundational issues, Fundam. Inform. 127 (2013) 1–15.
[6] D. Ciucci, Orthopairs in the 1960s: historical remarks and new ideas, in: Proceedings of RSCTC 2014, LNCS(LNAI) 8536, 2014, pp. 1–12.
[7] D. Ciucci, D. Dubois, A map of dependencies among three-valued logics, Inform. Sci. 250 (2013) 162–177.
[8] X.F. Deng, Y.Y. Yao, Decision-theoretic three-way approximations of fuzzy sets, Inform. Sci. 279 (2014) 702–715.
[9] H. Fujita, Research directions in subjective decision support system: case study on medical diagnosis, in: Proceedings of RSKT 2014, LNCS(LNAI) 8818, 2014, pp. XI–XIII.
[10] H. Fujita, Y.C. Ko, Evidential probabilities for rough set in a case of competitiveness, in: V.H. Nguyen, A.C. Le, V.N. Huynh (Eds.), Knowledge Systems Engineering, Springer International Publishing, Switzerland, 2015, pp. 3–13.
[11] J.W. Grzymala-Busse, Knowledge acquisition under uncertainty – a rough set approach, J. Intell. Robot. Syst. 1 (1988) 3–16.
[12] J.W. Grzymala-Busse, A new version of the rule induction system LERS, Fundam. Inform. 31 (1997) 27–39.
[13] K. Guttmannova, A.L. Shields, J.C. Caruso, Promoting conceptual understanding of statistics: definitional versus computational formulas, Teaching Psychol. 32 (2005) 251–253.
[14] P.G. Hewett, Conceptual Physics, eighth ed., Addison-Wesley, Reading, Massachusetts, 1998.
[15] B.Q. Hu, Three-way decisions space and three-way decisions, Inform. Sci. 281 (2014) 21–52.
[16] P.J. Hurley, A Concise Introduction to Logic, seventh ed., Wadsworth Publishing, Belmont, California, 2000.
[17] X.Y. Jia, W.H. Liao, Z.M. Tang, L. Shang, Minimum cost attribute reduction in decision-theoretic rough set models, Inform. Sci. 219 (2013) 151–167.
[18] X.Y. Jia, Z.M. Tang, W.H. Liao, L. Shang, On an optimization representation of decision-theoretic rough set model, Int. J. Approx. Reason. 55 (2014) 156–166.
[19] H.X. Li, X.Z. Zhou, B. Huang, D. Liu, Cost-sensitive three-way decision: a sequential strategy, in: Proceedings of RSKT 2013, LNCS(LNAI) 8171, 2013, pp. 325–337.
[20] P. Li, L. Shang, H.X. Li, A method to reduce boundary regions in three-way decision theory, in: Proceedings of RSKT 2014, LNCS(LNAI) 8818, 2014, pp. 834–843.
[21] D.C. Liang, D. Liu, Systematic studies on three-way decisions with interval-valued decision-theoretic rough sets, Inform. Sci. 276 (2014) 186–203.
[22] J.Y. Liang, Uncertainty and feature selection in rough set theory, in: Proceedings of RSKT 2011, LNCS (LNAI) 6954, 2011, pp. 8–15.
[23] D. Liu, T.R. Li, D.C. Liang, Incorporating logistic regression to decision-theoretic rough sets for classifications, Int. J. Approx. Reason. 55 (2014) 197–210.
[24] D. Liu, T.R. Li, D. Ruan, Probabilistic model criteria with decision-theoretic rough sets, Inform. Sci. 181 (2011) 3709–3722.
[25] C. Luo, T.R. Li, H.M. Chen, Dynamic maintenance of three-way decision rule, in: Proceedings of RSKT 2014, LNCS(LNAI) 8811, 2014, pp. 801–811.
[26] L.P. Ma, Knowing and Teaching Elementary Mathematics: Teachers' Understanding of Fundamental Mathematics in China and the United States, Lawrence Erlbaum Associates, Mahwah, New Jersey, 1999.
[27] X.A. Ma, G.Y. Wang, H. Yu, T.R. Li, Decision region distribution preservation reduction in decision-theoretic rough set model, Inform. Sci. 278 (2014) 614–640.
[28] V.W. Marek, Zdzisław Pawlak, databases and rough sets, in: A. Skowron, Z. Suraj (Eds.), Rough Sets and Intelligent Systems, Springer, Berlin, 2013, pp. 175–184.
[29] V.W. Marek, Z. Pawlak, Information storage and retrieval systems: mathematical foundations, Theor. Comput. Sci. 1 (1976) 331–354.
[30] V.W. Marek, M. Truszczyński, Contributions to the theory of rough sets, Fundam. Inform. 39 (1999) 389–409.
[31] National Mathematics Advisory Panel, Foundations for Success: The Final Report of the National Mathematics Advisory Panel, 2008, <http://www2.ed.gov/about/bdscomm/list/mathpanel/report/final-report.pdf> (accessed 03.10.14).
[32] M. Niaz, Relationship between student performance on conceptual and computational problems of chemical equilibrium, Int. J. Sci. Education 17 (1995) 343–355.
[33] C.K. Ogden, I.A. Richards, The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, eighth ed., Harcourt Brace, New York, 1946.
[34] E. Orłowska, Z. Pawlak, Expressive power of knowledge representation systems, Int. J. Man–Machine Stud. 20 (1984) 485–500.
[35] F. Ornek, I.M. Saleh (Eds.), Contemporary Science Teaching Approaches, Promoting Conceptual Understanding in Science, Information Age Publishing, Inc., Charlotte, North Carolina, 2012.
[36] Z. Pawlak, Mathematical Foundations of Information Retrieval, Research Report CC PAS Report 101, Computation Center, Polish Academy of Sciences, 1973.
[37] Z. Pawlak, Information systems, theoretical foundations, Inform. Syst. 6 (1981) 205–218.
[38] Z. Pawlak, Rough sets, Int. J. Comput. Inform. Sci. 11 (1982) 341–356.
[39] Z. Pawlak, Rough classification, Int. J. Man–Machine Stud. 20 (1984) 469–483.
[40] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Boston, 1991.
[41] A. Pollatsek, S. Lima, A.D. Well, Concept or computation: students' understanding of the mean, Educational Stud. Math. 12 (1981) 191–204.
[42] Y.H. Qian, H. Zhang, Y.L. Sang, J.Y. Liang, Multigranulation decision-theoretic rough sets, Int. J. Approx. Reason. 55 (2014) 225–237.
[43] W. Rybolt, G. Recck, Conceptual versus computational formulae in calculus and statistics courses, Int. J. Technol. Knowl. Soc. 8 (2012) 1–6.
[44] Y.H. She, On determination of thresholds in three-way approximation of many-valued NM-logic, in: Proceedings of RSCTC 2014, LNCS(LNAI) 8536, 2014, pp. 136–143.
[45] D. Ślęzak, Association reducts: complexity and heuristics, in: Proceedings of RSCTC 2006, LNCS(LNAI) 4259, 2006, pp. 157–164.
[46] E.E. Smith, Concepts and induction, in: M.I. Posner (Ed.), Foundations of Cognitive Science, The MIT Press, Cambridge, Massachusetts, 1989, pp. 501–526.
[47] J.F. Sowa, Conceptual Structures, Information Processing in Mind and Machine, Addison-Wesley, Reading, Massachusetts, 1984.
[48] I. van Mechelen, J. Hampton, R.S. Michalski, P. Theuns (Eds.), Categories and Concepts, Theoretical Views and Inductive Data Analysis, Academic Press, New York, 1993.
[49] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix: the ordered attributes method, J. Comput. Sci. Technol. 16 (2001) 489–504.
[50] B.L. Wang, J.Y. Liang, A novel intelligent multi-attribute three-way group sorting method based on Dempster-Shafer theory, in: Proceedings of RSKT 2014, LNCS(LNAI) 8811, 2014, pp. 789–800.
[51] J.H. Watt, S. van den Berg, Research Methods for Communication Science, Allyn and Bacon, Boston, 2002.
[52] J.T. Yao, H.X. Li, G. Peters, Decision-theoretic rough sets and beyond, Int. J. Approx. Reason. 55 (2014) 99–100.
[53] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Inform. Sci. 101 (1998) 239–259.
[54] Y.Y. Yao, Information granulation and rough set approximation, Int. J. Intell. Syst. 16 (2001) 87–104.
[55] Y.Y. Yao, On generalizing rough set theory, in: Proceedings of RSFDGrC 2003, LNCS(LNAI) 2639, 2003, pp. 44–51.
[56] Y.Y. Yao, A note on definability and approximations, in: LNCS Transactions on Rough Sets, VII, LNCS 4400, 2007, pp. 274–282.
[57] Y.Y. Yao, Three-way decisions with probabilistic rough sets, Inform. Sci. 180 (2010) 341–353.
[58] Y.Y. Yao, An outline of a theory of three-way decisions, in: Proceedings of RSKT 2012, LNCS(LNAI) 7413, 2012, pp. 1–17.
[59] Y.Y. Yao, Rough set approximations: a concept analysis point of view, manuscript, 2014.
[60] Y.Y. Yao, R. Fu, The concept of reducts in Pawlak three-step rough set analysis, in: LNCS Transactions on Rough Sets XVI, LNCS 7736, 2013, pp. 53–72.
[61] Y.Y. Yao, Y. Zhao, J. Wang, On reduct construction algorithms, in: LNCS Transactions on Computational Science II, LNCS 5150, 2008, pp. 100–117.
[62] H. Yu, Z.G. Liu, G.Y. Wang, An automatic method to determine the number of clusters using decision-theoretic rough set, Int. J. Approx. Reason. 55 (2014) 101–115.
[63] X.Y. Zhang, D.Q. Miao, Reduction target structure-based hierarchical attribute reduction for two-category decision-theoretic rough sets, Inform. Sci. 277 (2014) 755–776.
[64] Z.H. Zhang, Y. Li, W.B. Chen, F. Min, A three-way decision approach to incremental frequent itemsets mining, J. Inform. Comput. Sci. 11 (2014) 3399–3410.
[65] B. Zhou, Multi-class decision-theoretic rough sets, Int. J. Approx. Reason. 55 (2014) 211–224.
[66] Y.H. Zhu, H.L. Tian, J. Ma, J. Liu, T. Liang, An integrated method for micro-blog subjective sentence identification based on three-way decisions and naive Bayes, in: Proceedings of RSKT 2014, LNCS(LNAI) 8811, 2014, pp. 844–855.