Partitions, Coverings, Reducts and Rule Learning in Rough Set Theory

Yiyu Yao and Rong Fu

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{yyao,fu207}@cs.uregina.ca

Abstract. When applying rough set theory to rule learning, one commonly associates equivalence relations or partitions to a complete information table and tolerance relations or coverings to an incomplete table. Such associations are sometimes misleading. We argue that Pawlak three-step approach for data analysis indeed uses both partitions and coverings for a complete information table. A slightly different formulation of Pawlak approach is given based on the notions of attribute reducts of a classification table, attribute reducts of objects and rule reducts. Variations of Pawlak approach are examined.

Keywords: Attribute reduction, coverings, partitions, rule learning.

1 Introduction

In his book, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Pawlak [7] introduces a three-step method for analyzing data in an information table. The method can be applied to decision table simplification and rule learning. The main objective of this paper is to review Pawlak approach and to examine several of its variations, with a slightly different formulation based on a generic notion of a reduct of a set and an explicit expression of a concept by a pair of intension and extension. It is shown that the three steps use three types of reducts, namely, attribute reducts of the table with respect to decision attributes, attribute reducts of an object with respect to decision attributes, and rule reducts. The first step is related to partitions of a universe of objects [6–8], and the last two steps are related to coverings [2, 4, 5].

The reformulation of Pawlak three-step approach provides several insights. The generic notion of reducts unifies the three steps and demonstrates the simplicity and reflexibility of Pawlak approach. The explicit expression of concepts, in terms of logic formulas as intensions and subsets of objects as extensions, offers new understanding of reduct constructions through the notions of partitions and coverings. The results enable us to relate Pawlak approach to other standard rule learning algorithms, such as partition-based decision-three methods [9, 14] and covering-based sequential covering methods [1–4]. While other methods focus mainly on rule learning algorithms, Pawlak approach emphasizes on a study of intrinsic properties of rules independent of a particular rule learning algorithm.

J.T. Yao et al. (Eds): RSKT 2011, LNCS 6954, pp. 101–109, 2011.

More importantly, it is possible to clarify a misconception that coverings of a universe are commonly associated with incomplete information tables. In other words, one can use either partitions or coverings in a complete information table for rule learning. The latter has been done by Grzymala-Busse [4] in LERS.

2 Representations of Concepts and Classification

In rough set theory, a classification of a universe of objects is a partition of the universe, i.e., a family of pair-wise disjoint subsets whose union is the universe. Each block of the partition is considered to be instances of a class or a concept. Classification rule learning is to find rules that define classes of a classification based on some well defined concepts. A basic task of classification rule mining is to precisely construct and represent concepts used in classification [11, 12].

There are many views for interpreting the notion of concepts [10]. We adopt the classical view that interprets and represents a concept jointly by a pair of intension and extension. The intension is intrinsic properties of a concept based on which one can determine if an object is an instance of the concept; the extension is the set of instances of the concepts. In rough set theory, intension and extension of a concept can be precisely defined through the notion of an information table, in which a finite set of objects is described by a finite set of attributes. The next few definitions are adopted from Pawlak [7].

Definition 1. *An information table is the following tuple:*

$$S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, V_a is a nonempty set of values for $a \in At$, and $I_a : U \longrightarrow V_a$ is a complete information function that maps an object of U to exactly one value in V_a.

When an information function is a partial function, that is, it is not defined for some objects in U, the information table is called an incomplete table. In this paper, we only consider complete tables.

Definition 2. *For a subset of attributes $P \subseteq At$, an equivalence relation on U is defined by:*

$$xE_P y \Leftrightarrow \forall a \in P(I_a(x) = I_a(y)). \tag{1}$$

That is, x and y are equivalent if and only if they have the same values on all attributes in P. The induced partition of the universe is denoted by $U/E_P = \{[x]_P \mid x \in U\}$, where $[x]_P = [x]_{E_P} = \{y \mid xE_p y\}$ is the equivalence class containing x.

An equivalence relation on U induces a partition of U; conversely, a partition of U uniquely defines an equivalence relation on U. There is a one-to-one correspondence between all equivalence relations on U and all partitions of U.

Therefore, we can use equivalence relations and partitions interchangeably. The standard set inclusion of equivalence relations defines a partial order on partitions as follows:

$$U/E \preceq U/E' \Longleftrightarrow E \subseteq E' \tag{2}$$

If $U/E \preceq U/E'$, each block of U/E' must be a union of some blocks of U/E. Thus, U/E is called a refinement of U/E' and U/E' a coarsening of U/E. With respect to different subsets of attributes, we can establish the following relationship, for $P, P' \subseteq At, x \in U$,

$$\begin{aligned}
&\text{(i).} && E_P = \bigcap_{a \in P} E_{\{a\}}, \\
& && E_{P \cup P'} = E_P \cap E_{P'}, && (3)\\
&\text{(ii).} && [x]_P = \bigcap_{a \in P} [x]_{\{a\}}, \\
& && [x]_{P \cup P'} = [x]_P \cap [x]_{P'}, && (4)\\
&\text{(iii).} && P \subseteq P' \Rightarrow E_{P'} \subseteq E_P, \\
& && P \subseteq P' \Rightarrow U/E_{p'} \preceq U/E_P. && (5)
\end{aligned}$$

Properties (i) and (ii) show that the equivalence relation, or the partition, defined by a subset of attributes can be constructed from the individual equivalence relations or partitions defined by singleton subsets of attribute. Property (iii) states that the refinement-coarsening relation $\preceq$ is monotonic with respect to set inclusion of sets of attributes.

Each equivalence class $[x]_P$ may be viewed as the extension of a concept. We now turn our attention to define the intension of a concept based on a logic language in an information table.

Definition 3. *In an information table, a decision logic language $\mathcal{L}$ is recursively defined as follows: an atomic formula is given by $a = v$,where $a \in At$ and $v \in V_a$. If ϕ and ψ are formulas, then $\phi \wedge \psi$ is a formula.*

The decision language is a sub-language of the decision logic language used by Pawlak [7]. By restricting to only the logic connective $\wedge$, we only consider a concept that can be defined by the conjunction of a family of atomic formulas. The meaning of a formula is defined by the satisfiability of the formula by objects.

Definition 4. *Let $x \vDash \phi$ denote that x satisfies a formula ϕ. It can be formally defined as follows:*

$$x \vDash (a = v) \text{ iff } I_a(x) = v$$
$$x \vDash \phi \wedge \psi \text{ iff } x \vDash \phi \textit{ and } x \vDash \psi$$

Object x satisfies a formula ϕ if and only if it satisfies all atomic formulas in ϕ.

Definition 5. *For a formula ϕ, the set $m(\phi) \subseteq U$ defined by*

$$m(\phi) = \{x \in U \mid x \models \phi\}, \tag{6}$$

is called the meaning of the formula ϕ.

By definition, the meaning of formulas satisfies the following two properties:

$$m(a = v) = \{x \in U \mid I_a(x) = v\}, \tag{7}$$
$$m(\phi \wedge \psi) = m(\phi) \cap m(\psi). \tag{8}$$

With the introduced logic language, we have a precise representation of a concept. A concept is a pair $(\phi, m(\phi))$, where ϕ is the intension of the concept given by a logic formula of $\mathcal{L}$ and $m(\phi)$ is the extension of the concept given by a subset of objects satisfying the formula ϕ.

For a given formula, one can obtain a unique subset of U as its meaning set. In contrast, for an arbitrary subset of U, one may not find a formula that produces the subset. A subset $X \subseteq U$ is called a conjunctively definable set if there exists a formula ϕ such that $m(\phi) = X$; otherwise, X is undefinable. It may also happen that one may find more than one formula for a subset of U. In other words, some subset of U may have multiple representations in terms of formulas in $\mathcal{L}$.

We can establish a connection between conjunctively definable sets and equivalence classes of equivalence relations defined by subsets of attributes. More specifically, we have: for $x \in U, a \in At$, and $P \subseteq At$,

(1) $m(a = I_a(x)) = [x]_{\{a\}}$

(2) $m(\bigwedge_{a \in P} a = I_a(x)) = [x]_P = \bigcap_{a \in P} [x]_{\{a\}}$

In other words, we can construct concepts $(a = I_a(x), [x]_{\{a\}})$ and $(\bigwedge_{a \in P} a = I_a(x), [x]_P)$ by using the decision logic language. The family of such concepts forms a basis for classification rule learning.

Definition 6. *A classification table or a decision table is an information table $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, where C is the set of condition attributes and D is the set of classification or decision attributes. It is a consistent classification table if $U/E_C \preceq U/E_D$.*

In a consistent classification table, objects with the same values on condition attributes would have the same values on classification attributes. That is, $\forall x, y \in U (I_C(x) = I_C(y) \Longrightarrow I_D(x) = I_D(y))$, where $I_C(x)$ and $I_D(x)$ denote the value of object x on the sets of attributes C and D, respectively. In this paper, we only consider consistent classification tables.

One can construct concepts based on condition attributes and use them to define classes given by classification attributes. Suppose that $[x]_P$ and $[x]_D$ are equivalence classes of E_P and E_D, respectively, where $P \subseteq C$. We have two concepts $(\bigwedge_{a \in P} a = I_a(x), [x]_P)$ and $(\bigwedge_{d \in D} d = I_d(x), [x]_D)$. A classification rule can be inferred based on both extensions and intensions. Specifically, if their extensions are related by,

$$[x]_P \subseteq [x]_D, \tag{9}$$

we can formulate a classification rule based on their intensions:

$$\bigwedge_{a\in P} a = I_a(x) \rightarrow \bigwedge_{d\in D} d = I_d(x). \tag{10}$$

That is, a rule is a plausible relationship between the intensions of concepts inferred from their extensions in an information table.

Suppose $RS = \{l_i \rightarrow r_i \mid 1 \leq i \leq k\}$ is a set of k rules of a classification table. We can easily express requirements on RS based on the notions of partitions and coverings. As a minimum requirement, one expects that the rule set covers all instances. That is, $\{m(l_i) \mid 1 \leq i \leq k\}$ is a covering of U. It may also be required that rules in RS are pair-wise disjoints. That is, $\{m(l_i) \mid 1 \leq i \leq k\}$ is a partition of U. Two additional properties of a rule set may be stated. Two rules $l_i \rightarrow r_i$ and $l_j \rightarrow r_j$ are consistent if $x \in m(l_i \wedge l_j) \Longrightarrow x \in m(r_i \wedge r_j)$. That is, overlapping rules with $m(l_i) \cap m(l_j) \neq \emptyset$ will provide the same classification of objects in $m(l_i) \cap m(l_j)$. For a consistent classification table, rules obtained based on equations (9) and (10) are pair-wise consistent. Suppose RS is a set of pair-wise consistent rules. A rule $l \rightarrow r$ is a redundant rule if there exists a family of rules $RS' \subseteq RS - \{l \rightarrow r\}$ such that $m(l) \subseteq \bigcup\{m(l_i) \mid l_i \rightarrow r_i \in RS'\}$. That is, all objects covered by rule $l \rightarrow r$ are covered by other rules in RS. While there is no redundant rule for a partition, there are may be redundant rules for a covering. Thus, for a covering, one may further impose a non-redundancy conditions. All those issues will be discussed in the next section.

3 Pawlak Three-Step Approach to Rule Learning

A new formulation of Pawlak three-step approach [7] to data analysis is presented in this section based on a more general and generic notion of reducts.

3.1 Step 1: Construction of an Attribute Reduct of a Table with Respect to Classification Attributes

The first step of Pawlak approach for data analysis is to find a minimal set of attributes that has the same power as the entire set of condition attributes for defining classification E_D. Such a subset of C is called an attribute reduct of a table relative to D.

Definition 7. *Given a consistent classification table $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, a subset $R \subseteq C$ is called a reduct of C relative to D, if R satisfies the two conditions:*

$$\text{(s1).} \quad U/E_R \preceq U/E_D;$$
$$\text{(n1).} \quad \textit{for any } a \in R, \neg(U/E_{R-\{a\}} \preceq U/E_D);$$

where $\preceq$ stands for the refinement-coarsening relation between partitions.

Condition (s1) states that attributes in R are jointly sufficient for classifying objects. For $P \subseteq At$, an attribute satisfying the condition $U/E_{P-\{a\}} \preceq U/E_D$ is called a redundant attribute. Condition (n1) states that attributes in R are individually necessary. That is, none of the attributes in R is redundant and can be deleted. One can find an attribute reduct by sequentially deleting redundant attributes. Additional information on reduct construction methods can be find in rough set literature [4–8, 13].

For each equivalence class $[x]_R$, by condition (s1) we have $[x]_R \subseteq [x]_D$. Thus, a classification rule can be defined as:

$$\bigwedge_{a \in R} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x). \tag{11}$$

The family of classification rules is given by:

$$RS_1 = \{ \bigwedge_{a \in R} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x) \mid x \in U\}. \tag{12}$$

By the fact that $[x]_R = m(\bigwedge_{a \in R} a = I_a(x))$, rules in RS are non-overlapping and non-redundant. For a classification table, there may exist more than one attribute reduct. Different reducts produce different sets of rules.

A problem with rule set RS_1 is that some condition on the left hand side of a rule may be redundant. That is, there may exist a proper subset $R' \subset R$ such that $[x]_{R'} \subseteq [x]_D$, which produces a more general rule:

$$\bigwedge_{a \in R'} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x). \tag{13}$$

The next step in Pawlak data analysis is to simplify rules obtained in Step 1.

3.2 Step 2: Construction of an Attribute Reduct of an Object with Respect to Classification Attributes

With respect to a given reduct, one can produce a non-redundant rule set. The second step is to simplify each rule by removing redundant atomic formula in the left hand side of a rule. This is commonly known as the construction of a category reduct [7] or a minimal rule [8]. In this paper, we use the notion of an attribute reduct of an object, to be consistent with the discussion in Step 1.

Definition 8. *Given a consistent classification table $S = (U, At = C \cup D, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, a subset $R(x) \subseteq C$ is called an attribute reduct of x relative to D, if $R(x)$ satisfies the two conditions:*

(s2). $[x]_{R(x)} \subseteq [x]_D$;
(n2). $for\ any\ a \in R(x), \neg([x]_{R(x)-\{a\}} \subseteq [x]_D)$.

An attribute reduct R of a classification table is for all objects and an attribute reduct $R(x)$ of x is for only one object. Condition (s2) and (n2) can be similarly interpreted. In Pawlak's book, an attribute reduct R of the table is used to construct an attribute reduct $R(x)$ of an object, which is not necessary. We adopt the method presented in the paper by Pawlak and Showron [8], in which the entire set of condition attributes C is used instead of $R \subseteq C$. Let $REDUCT(x)$ denotes the set of all attribute reducts of x. It should be noted that the set of all reducts constructed from R may be different from the set of all reducts constructed from C.

By property (s2), for an attribute reduct $R(x)$ of x, we have $[x]_{R(x)} \subseteq [x]_D$, which produces a rule:

$$\bigwedge_{a \in R(x)} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x). \tag{14}$$

From the set of all reducts $REDUCT(x)$, we can construct a family of rules:

$$RS(x) = \{ \bigwedge_{a \in R(x)} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x) \mid R(x) \in REDUCT(x)\}. \tag{15}$$

Each rule in $RS(x)$ is sufficient to classify object x. By taking the union of the sets for all objects, we derive a family of rules for classifying all objects:

$$RS_2 = \bigcup \{RS(x) \mid x \in U\}. \tag{16}$$

There may exist overlapping and redundant rules in RS_2. The family $\{m(l) \mid l \rightarrow r \in RS_2\}$ is a covering of U. Rule learning in the second step of Pawlak approach is therefore based on coverings of universe.

3.3 Step 3: Construction of a Rule Reduct

The third step of Pawlak data analysis consists of constructing a rule set and simplifying the rule set by removing redundant rules. Instead of using RS_2, Pawlak constructs a rule set CRS by selecting a single rule from $RS(x)$. The family $\{m(l) \mid l \rightarrow r \in CRS\}$ is a covering of U and may not necessarily be a partition of U. There may exist redundant rules in CRS. It is therefore necessary to obtain a rule reduct of CRS.

Definition 9. *Given a set RS of consistent rules, a subset $RR \subseteq RS$ is called a rule reduct of RS if it satisfies the following conditions:*

$$\begin{aligned} &\text{(s3).} \quad \bigcup \{m(l) \mid l \rightarrow r \in RR\} = U \\ &\text{(n3).} \quad for\ any\ rule\ l_0 \rightarrow r_0 \in RR, \\ &\qquad\quad \bigcup \{m(l) \mid l \rightarrow r \in RR - \{l_0 \rightarrow r_0\}\} \neq U. \end{aligned}$$

Condition (s3) states that rules in RR are sufficient for classifying all objects and condition (n3) states that each rule in RR is necessary. Based on condition (n3), one can remove redundant rules to obtain a rule reduct. Like attribute reducts, there may exist more than one rule reduct.

The meaning sets of the left hand side of rules in a rule set form a covering of the universe U; rule reducts are in fact reducts of a covering [15]. By Step 2, each rule in a rule reduct is a minimal rule.

4 Discussions and Variations of Pawlak Approach

Based on the formulation of the last section, we present several observations and variations of Pawlak three-step approach.

In our formulation, we in fact introduce a more general and generic notion of reducts. A reduct of a set is a subset of the set satisfying a pair of conditions, namely, a jointly sufficient condition (s) and an individually necessary condition (n). When the set is chosen to be set of attributes, we obtain attribute reducts of the table and attribute reducts of an object, respectively in Step 1 and Step 2. The former is defined globally for all objects; the latter is defined locally for a particular object. When the set is chosen to be a set of rules, we obtain rule reducts in Step 3.

The unified framework based on reducts has a number of advantages. First, one can apply a generic reduct construction algorithm for constructing any of the three types of reducts. In particular, one may use any of the three classes of algorithms, deletion, addition-deletion, and addition algorithms [13]. Second, all three steps can be viewed as different applications of the same data reduction method. Third, the same framework can be further applied to new situations where a reduct of a set is of interest.

In our formulation, we explicitly express intension and extension of a concept. A classification rule is expressed as a pair of two rules, one for extension and the other for intension: for $x \in U, P \subseteq C$,

$$\begin{cases} [x]_P \subseteq [x]_D, \\ \bigwedge_{a \in P} a = I_a(x) \rightarrow \bigwedge_{d \in D} d = I_d(x). \end{cases} \tag{17}$$

It enables us to see additional insights into Pawlak three-step approach. Steps 1 and 2 use both intensions and extensions. Step 3 only uses extensions.

In Pawlak three-way approach, both partitions and covering of a universe are used with respect to a complete classification table. The differences between the uses of partitions and covering are due to the requirements for overlapping and non-overlapping rules, rather than complete and incomplete tables.

The Pawlak three-step approach can be modified in several ways. It can be observed that the first step is not necessary. Thus, a two-step approach can be derived based only on Steps 2 and 3. In Step 2, attribute reducts are constructed based on both intensions and extensions. One may consider only extensions of concepts without reference to intensions. This can be formulated as a search for a reduct of the family of subsets of U given by:

$$\{[x]_P \mid P \subseteq C, [x]_P \subseteq [x]_D\}. \tag{18}$$

The results are a new rule learning method [12]. In Step 3, a rule reduct is defined independent of how a rule set is formed. For finding a set of non-redundant minimal rules, one can construct a reduct of either the rule set RS_2 or the rule set CRS.

5 Conclusion

By reformulating and re-interpreting Pawlak three-step approach to data analysis, this paper makes contributions to rough set based rule learning. A generic notion of reducts is introduced, of which attribute reducts and rule reducts are concrete cases. A classification rule is explicitly expressed as a pair of rules by using intensions and extensions of concepts. This enables us to reveal new features of reduct construction methods. Based on these findings, variations of Pawlak approach are suggested.

References

1. Blaszczynski, J., Slowinski, R., Szelag, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Information Sciences 181, 987–1002 (2011)
2. Cendrowska, J.: PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies 27, 349–370 (1987)
3. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13, 3–54 (1999)
4. Grzymala-Busse, J.: LERS - A system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
5. Grzymala-Busse, J., Rzasa, W.: Approximation space and LEM2-like algorithms for computing local coverings. Fundamenta Informaticae 85, 205–217 (2008)
6. Mi, J.S., Leung, Y., Wu, W.Z.: Dependence-space-based attribute reduction in consistent decision tables. Soft Computing 15, 261–268 (2011)
7. Pawlak, Z.: Rough sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
8. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177, 41–73 (2007)
9. Quinlin, J.R.: Induction of decision tree. Machine Learning 1, 81–106 (1986)
10. Van Mechelen, I., Hampton, J., Michalski, R.S., Theuns, P. (eds.): Categories and Concepts: Theoretical Views and Inductive Data Analysis. Academic Press, New York (1993)
11. Yao, Y.Y.: Interpreting concept learning in cognitive informatics and granular computing. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39, 855–866 (2009)
12. Yao, Y.Y., Deng, X.F.: A granular computing paradigm for concept learning (2011) (manuscript)
13. Yao, Y., Zhao, Y., Wang, J.: On reduct construction algorithms. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 297–304. Springer, Heidelberg (2006)
14. Zhao, Y., Yao, Y.Y., Yao, J.T.: Level construction of decision trees for classification. International Journal Software Engineering and Knowledge Engineering 16, 103–126 (2006)
15. Zhu, W., Wang, F.Y.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152, 217–230 (2003)