Mining Ordering Rules Using Rough Set Theory

Y.Y. Yao and **Ying Sai**[1] {yyao,ying}@cs.uregina.ca
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2

[1]Permanent address:
Department of Computer Information Engineering
Shandong Finance Institute, Jinan, Shandong, China 250014

Abstract: *Many real world problems deal with ordering of objects instead of classifying objects, although majority of research in machine learning and data mining has been focused on the latter. Examples of such problems are ordering of consumer products produced by different manufactures, ranking of universities, and so on. Typically, an overall ordering of objects is given. In this paper, we formulate the problem of mining ordering rules as finding association between orderings of attribute values and the overall ordering of objects. An example of ordering rules may state that "if the value of an object x on an attribute a is ordered ahead of the value of another object y on the same attribute, then x is ordered ahead of y". For mining ordering rules, the notion of information tables is generalized to ordered information tables by adding order relations on attribute values, and rough set theory based algorithms are then used.*

Keywords: Information table, Learning, Ordering Rules, Ranking, Rough Sets.

1 Introduction

One of the basic tasks of inductive learning and data mining is to learn the knowledge for classification. It is not surprising that the majority research has been concentrated on such a task. This is particularly true for rough set theory based approaches, as the theory was originally developed in the context of classification [16, 13, 14]. In real world situations, we may be faced with many problems that are not simply classification [3, 15]. One such type of problems is the ordering of objects. In this paper, we study the less addressed problem of ordering.

Two familiar examples of ordering problems are the ranking of universities and the ranking of the consumer products produced by different manufactures. In both examples, we have a set of attributes that are used to describe the objects under consideration. Consider the example of ranking consumer products. Attributes may be the price of the products, warranty of the products, and other information. The values of a particular attribute, say the price, naturally induce an ordering of objects. The overall ranking of products may be produced by their market shares of different manufactures. The ordering of objects by attribute values may not necessarily be the same as the overall ordering of objects. In this setting, a number of important issues arise. It would be interesting to know which attributes play more important roles in determining the overall ordering, and which attributes do not contribute at all to the overall ordering. It would also be useful to know which subset of attributes would be sufficient to determine the overall ordering. The dependency information of attributes may also be valuable.

From the previous example, we can identify the problem of mining ordering rules. There is a set of objects described by a set of attributes. There is an ordering on values of each attribute, and there is also an overall ordering of objects. The overall ordering may be given by experts or obtained from other information, either dependent or independent of the orderings of objects according to their attribute values. We are interested in mining the association between the overall ordering and the individual orderings induced by different attributes. More specifically, we want to derive ordering rules exemplified by the statement that "if the value of an object x on an attribute a is ordered ahead of the value of another object y on the same attribute, then x is ordered ahead of y".

Typically, an ordering rule may not be exact. In order to capture the uncertainty associated ordering rules, two quantitative measures are used. They the accuracy and the coverage of the rules [19, 20, 27]. The former deals with the correctness of the rules, and the latter represents the extent to which the rule covers the positive instances. For mining such ordering rules, we first introduce the notion of ordered information tables as a generalization of information tables. With an ordered information table, an ordered decision logic language is given to define ordering rules. We transform an ordered information table into a standard information table, on which rough set theory based mining algorithms are applied.

Ordered information tables are related to ordinal information systems proposed and studied by Iwinski [7, 8]. Ordering induced by attribute values in information tables were also considered by Greco, Matarazzo and Slowinski [4, 5]. Their work on approximating preference relations by dominance relations is related to, but different from, our approach for mining ordering rules.

The major difference between the proposed method and the conventional rough set methods is the requirement of ordering relations on attribute values. Traditionally, rough set theory relied on the trivial equality relation $=$ on attribute values. It is suitable for mining classification rules, but inadequate for mining ordering rules. Once ordering relations are introduced on attribute values, traditional rough set methods can be applied immediately. In fact, we can simply use the traditional data mining software packages by transforming an ordered information table to a binary information table.

2 Ordered Information Tables

In many information processing systems, a set of objects are typically represented by their values on a finite set of attributes. Such information may be conveniently described in a tabular form [12, 14]. Formally, an information table is defined as a quadruple:

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

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 $a \in At$,
$I_a : U \to V_a$ is an information function.

For simplicity, in this definition we have considered only information tables characterized by a finite set of objects and a finite set of attributes. Each information function I_a is a total function that maps an object of U to exactly one value in V_a. The rows of the table correspond to objects of the universe, the columns correspond to a set of attributes, and each cell is the value of an object with respect to an attribute. An information table represents all available information and knowledge. Objects are only perceived, observed, or measured by using a finite number of properties. Similar representation schemes can be found in many fields, such as decision theory, pattern recognition, machine learning, data analysis, data mining, and cluster analysis [14].

With respect to the notion of information tables, there are extensive studies on the relationships between values of *different* attributes and relationships between values of the *same* attribute, i.e., the *horizontal* analysis and the *vertical* analysis of an information table.

Analysis of the horizontal relationships reveals the similarity, association, and dependency of different attributes [28]. Such relationships are normally characterized by the problem of determining the values of one set of attributes based on the values of another set of attributes. Two levels of dependencies, referred to as the *local* and *global* dependencies, may be observed. The local dependencies show how *one* specific combination of values on one set of attributes determines *one* specific combination of values on another set of attributes. The global dependencies show *all* combinations of values on one set of attributes determine *all* combinations of values on another set of attributes. Finding local dependencies is one of the main tasks of machine learning and data mining [10, 14]. For instance, the well-known association rules, which state the presence of one set of items implies the presence of another set of items, may be considered as a special kind of local dependencies. Functional dependency in relational databases is a typical example of global dependency [1, 2]. Attribute (data) dependency studied in the theory of rough sets is another example of global dependency [14].

Analysis of vertical relationships deals with semantic closeness of values of an attribute. Examples of vertical analysis include the discretization of real-valued attributes, and the use of order relations, concept hierarchies, fuzzy binary relations, similarity measures or distance functions on attribute values [6, 7, 9, 21, 26, 29].

An information table does not consider any semantic relationships between different attribute values of a particular attribute [24]. Different values of the same attribute are treated as distinct symbols without any connections, and hence horizontal analyses rely, to a large extent, on simple pattern matching. More specifically, one uses the trivial equality relation $=$ on values of an attribute. The standard rough set theory was therefore developed based on the trivial equality relation on attribute values [14]. Although the use of equality relation is sufficient for a number of tasks, such as classification, cluster analysis, and simple horizontal analysis of the data, it may be inadequate for other tasks such as approximate retrieval and ordering of objects. By incorporating semantics information, we may obtain different generalization of information tables [24].

Generalized information tables may be viewed as information tables with added semantics. For the problem of mining ordering rules, we introduce order relations on attribute values. An ordered information

	a	b	c	d	o
p_1	middle	3 years	\$200	heavy	1
p_2	large	3 years	\$300	very heavy	3
p_3	small	3 years	\$300	light	3
p_4	small	3 years	\$250	very light	2
p_5	small	2 years	\$200	very light	3

$\succ_a$: small $\succ$ middle $\succ$ large,
$\succ_b$: 3 years $\succ$ 2 years,
$\succ_c$: \$200 $\succ$ \$250 $\succ$ \$300,
$\succ_d$: very light $\succ$ light $\succ$ heavy $\succ$ very heavy,
$\succ_o$: 1 $\succ$ 2 $\succ$ 3.

Table 1: An ordered information table

table is define by:

$$OIT = (IT, \{\succ_a \mid a \in At\}),$$

where IT is the standard information table and $\succ_a$ is an order relation on attribute a. An ordering of values of a particular attribute a naturally induces an ordering of objects:

$$x \succ_{\{a\}} y \Longleftrightarrow I_a(x) \succ_a I_a(y), \tag{1}$$

where $\succ_{\{a\}}$ denotes an order relation on U induced by the attribute a. An object x is ranked ahead of another object y if and only if the value of x on the attribute a is ranked ahead of the value of y on a. The relation $\succ_{\{a\}}$ has exactly the same properties as that of $\succ_a$. For a subset of attributes $A \subseteq At$, we define:

$$\begin{aligned} x \succ_A y &\Longleftrightarrow \forall a \in A[I_a(x) \succ_a I_a(y)] \\ &\Longleftrightarrow \bigwedge_{a\in A} I_a(x) \succ_a I_a(y) \\ &\Longleftrightarrow \bigcap_{a\in A} \succ_{\{a\}} . \end{aligned} \tag{2}$$

That is, x is ranked ahead of y if and only if x is ranked ahead of y according to all attributes in A. The above definition is a straightforward generalization of the standard definition of equivalence relations in rough set theory, where the equality relation $=$ is used [14]. Mining ordering rules based on order relations is a concrete example of applications of our earlier studies on generalizations of rough set model with non-equivalence relations [23, 25].

For simplicity, we also assume that there is a special attribute, called decision attribute. The ordering of objects by the decision attribute is denoted by $\succ_o$ and is called the overall ordering of objects.

An order relation should satisfy certain conditions. We consider the following two properties [22]:

Asymmetry :

$$x \succ y \Longrightarrow \neg(y \succ x),$$

Negative transitivity :

$$(\neg(x \succ y), \neg(y \succ z)) \Longrightarrow \neg(x \succ z).$$

An order relation satisfying these properties is called a weak order. An important implication of weak order is that the following relation,

$$x \sim y \Longleftrightarrow (\neg(x \succ y), \neg(y \succ x)), \tag{3}$$

is an equivalence relation. For two elements, if $x \sim y$ we say x and y are indiscernible by $\succ$. The equivalence relation $\sim$ induces a partition $U/\sim$ on U, and an order relation on $U/\sim$ can be defined by:

$$[x]_\sim \succ^* [y]_\sim \Longleftrightarrow x \succ y, \tag{4}$$

where $[x]_\sim$ is the equivalence class containing x. Moreover, $\succ^*$ is a linear order [22]. Any two distinct equivalence classes of $U/\sim$ can be compared. It is therefore possible to arrange the objects into levels, with each level consisting of indiscernible elements defined by $\succ$. In this study, we assume that all order relations are weak order. For a weak order, $\neg(x \succ y)$ can be written as $y \succeq x$ or $x \preceq y$, which means $y \succ x$ or $y \sim x$. For any two elements x and y, we have either $x \succ y$ or $y \succeq x$, but not both.

Example 1 *Suppose we have an ordered information table of a group of products produced by five manufactures as shown in Table 1. In this table, a, b, c, d, and o stand for size, warranty, price, weight, and overall ordering on a set of products, respectively. Based on orderings of attribute values, we obtain the following orderings of products:*

$$\begin{aligned} \succ_{\{a\}} &: [p_3, p_4, p_5] \succ^*_{\{a\}} [p_1] \succ^*_{\{a\}} [p_2], \\ \succ_{\{b\}} &: [p_1, p_2, p_3, p_4] \succ^*_{\{b\}} [p_5], \\ \succ_{\{c\}} &: [p_1, p_5] \succ^*_{\{c\}} [p_4] \succ^*_{\{c\}} [p_2, p_3], \\ \succ_{\{d\}} &: [p_4, p_5] \succ^*_{\{d\}} [p_3] \succ^*_{\{d\}} [p_1] \succ^*_{\{d\}} [p_2], \\ \succ_{\{o\}} &: [p_1] \succ^*_{\{o\}} [p_4] \succ^*_{\{o\}} [p_2, p_3, p_5]. \end{aligned}$$

For subsets $\{a, b\}$ *and* $\{c, d\}$*, we have:*

$$\begin{array}{lcl} \succ_{\{a,b\}} & : & \emptyset, \\ \succ_{\{c,d\}} & : & p_1 \succ_{\{c,d\}} p_2, \quad p_4 \succ_{\{c,d\}} p_2, \\ & & p_5 \succ_{\{c,d\}} p_2, \quad p_4 \succ_{\{c,d\}} p_3, \\ & & p_5 \succ_{\{c,d\}} p_3. \end{array}$$

By combining attributes a and b, all objects are put into the same class. On the other hand, it is interesting to note that $\succ_{\{c,d\}}$ *is not a weak order. That is, the intersection of two weak orders may not produce a weak order. This suggests that rules using simple condition* $\bigwedge_{a \in A} I_a(x) \succ_a I_a(y)$*, as in the standard rough set based methods, might not be very useful.*

3 An Ordered Decision Logic

For the purpose of mining ordering rules, we need to define the form (logic expressions) and interpretations of various expressions. This can be done by introducing an ordered decision logic language (ODL-language) for ordered information tables, a generalization of the Pawlak decision logic for standard information tables [14].

With an ordered information table, an ODL-language is given as follows, similar to Pawlak decision logic language [14]. In the ODL-language, an atomic expression is given by $(a, \succ)$ or $(a, \preceq)$, where $a \in At$ and $\succ$ is an order relation on attribute a. If ϕ and ψ are expressions in the ODL-language, then so are $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \to \psi$, and $\phi \equiv \psi$. The semantics of the ODL-language can be defined in Tarski's style through the notions of a model and satisfiability. The model is an ordered information table OIT, which provides interpretation for symbols and expressions of the ODL-language. The satisfiability of an expression ϕ by an object pair (x, y), written $(x, y) \models_{OIT} \phi$ or in short $(x, y) \models \phi$ if OIT is understood, is given by:

(a1). $(x, y) \models (a, \succ)$ iff $x \succ_{\{a\}} y$,

(a2). $(x, y) \models (a, \preceq)$ iff $x \preceq_{\{a\}} y$,

(a3). $(x, y) \models \neg\phi$ iff not $(x, y) \models \phi$,

(a4). $(x, y) \models \phi \wedge \psi$ iff $(x, y) \models \phi$ and $(x, y) \models \psi$,

(a5). $(x, y) \models \phi \vee \psi$ iff $(x, y) \models \phi$ or $(x, y) \models \psi$,

(a6). $(x, y) \models \phi \to \psi$ iff $(x, y) \models \neg\phi \vee \psi$,

(a7). $(x, y) \models \phi \equiv \psi$ iff $(x, y) \models \phi \to \psi$ and $(x, y) \models \psi \to \phi$.

For an expression ϕ, the set $m_{OIT}(\phi)$ defined by:

$$m_{OIT}(\phi) = \{(x, y) \in U \times U \mid (x, y) \models \phi\}, \quad (5)$$

is called the meaning of the expression ϕ in OIT. If OIT is understood, we simply write $m(\phi)$. Obviously, the following properties hold [11, 14]:

(b1). $m((a, \succ)) = \{(x, y) \in U \times U \mid x \succ_{\{a\}} y\}$,

(b2). $m((a, \preceq)) = \{(x, y) \in U \times U \mid x \preceq_{\{a\}} y\}$,

(b3). $m(\neg\phi) = -m(\phi)$,

(b4). $m(\phi \wedge \psi) = m(\phi) \cap m(\psi)$,

(b5). $m(\phi \vee \psi) = m(\phi) \cup m(\psi)$,

(b6). $m(\phi \to \psi) = -m(\phi) \cup m(\psi)$,

(b7). $m(\phi \equiv \psi) = (m(\phi) \cap m(\psi)) \cup (-m(\phi) \cap -m(\psi))$,

where $-m(\phi)$ and $-m(\psi)$ are the corresponding complements of sets $m(\phi)$ and $m(\psi)$ in the universal set of $U \times U$. The meaning of an expression ϕ is therefore the set of all object pairs having the property expressed by the expression ϕ. In other words, ϕ can be viewed as the description of the set of object pairs $m(\phi)$. Thus, a connection between expressions of the ODL-language and subsets of $U \times U$ is established.

An expression ϕ is said to be true in an ordered information table OIT, written $\models_{OIT} \phi$ or $\models \phi$ for short when OIT is clear from the context, if and only if $m(\phi) = U \times U$. That is, ϕ is satisfied by all object pairs in the universal set $U \times U$. Two expressions ϕ and ψ are equivalent in OIT if and only if $m(\phi) = m(\psi)$. By definition, the following properties hold [14]:

(c1). $\models \phi$ iff $m(\phi) = U \times U$,

(c2). $\models \neg\phi$ iff $m(\phi) = \emptyset$,

(c3). $\models \phi \to \psi$ iff $m(\phi) \subseteq m(\psi)$,

(c4). $\models \phi \equiv \psi$ iff $m(\phi) = m(\psi)$.

Thus, we can study the relationships between concepts described by expressions of the ODL-language based on the relationships between their corresponding sets of object pairs.

Example 2 *Consider the ordered information table given by Table 1. We have the following results:*

$$\begin{array}{rcl} m((a, \succ)) & = & \{(p_3, p_1), (p_4, p_1), (p_5, p_1), \\ & & (p_3, p_2), (p_4, p_2), (p_5, p_2), \\ & & (p_1, p_2)\}, \\ m((o, \succ)) & = & \{(p_1, p_4), (p_1, p_2), (p_1, p_3), \\ & & (p_1, p_5), (p_4, p_2), (p_4, p_3), \\ & & (p_4, p_5)\}, \\ m((a, \succ) \to (o, \succ)) & = & U \times U - \\ & & \{(p_3, p_2), (p_3, p_1), (p_4, p_1), \\ & & (p_5, p_1), (p_5, p_2)\}, \end{array}$$

and

$$\models (b, \preceq) \wedge (b, \preceq) \to (o, \preceq).$$

In the OIT-language, we interpret $\to$ *as logic implication (material implication). With such a definition, one may not be able to find useful rules.*

4 Interpretation of Ordering Rules

In an ordered information table OIT, an atomic expression over a single attribute a is defined as either $(a, \succ)$ or $(a, \preceq)$. For a set of attributes $A \subseteq At$, an expression over A in OIT is defined by $\bigwedge_{a \in A} e(a)$, where $e(a)$ is an atomic expression over a. The set of all expressions over A in an ordered information table OIT is defined by $E(A)$. Relationships between attributes can be expressed as relationships between logic expression in the ODL-language.

Consider two subsets of attributes $A, B \subseteq At$. For two expressions $\phi \in E(A)$ and $\psi \in E(B)$, an ordering rule is read "if ϕ then ψ" and is denoted by $\phi \Rightarrow \psi$. The expression ϕ is called the rule's antecedent, while the expression ψ is called the rule's consequent. Mining ordering rules in an ordered information table may be formulated as finding association between orderings induced by attributes. In the general case, one is interested in finding associations between two arbitrary subsets of attributes. As a special case, the second subset may consist of a single attribute that defines the overall ordering of objects.

The ODL-language immediately offers one interpretation of the rule, $\phi \Rightarrow \psi$, in which $\Rightarrow$ is interpreted as the logical implication $\rightarrow$. In most cases, the expression $\phi \rightarrow \psi$ may not be true in an ordered information table. We thus have a quantitative measure associated with $\phi \Rightarrow \psi$ under logic implication interpretation:

$$T(\phi \Rightarrow \psi) = \frac{|m(\phi \rightarrow \psi)|}{|U \times U|}, \quad (6)$$

where $|\cdot|$ denotes the cardinality of a set. It measures the degree of truth of the expression $\phi \rightarrow \psi$ in an ordered information table. A problem with the logic implication interpretation can be seen as follows. For a pair of objects, if it does not satisfy ϕ, by definition, it satisfies $\phi \rightarrow \psi$. Thus, even if the degree of truth of $\phi \rightarrow \psi$ is very high, we may not use it to predict ordering as expressed by ψ.

Instead of using logic implication, we adopt a conditional probabilistic interpretation for ordering rules. A systematic analysis of probabilistic quantities associated with rules was given by Yao and Zhong [27]. In this paper, we choose to use two measures, called accuracy and coverage, proposed and studied by Tsumoto [19, 20].

An ordering rule $\phi \Rightarrow \psi$ may only reveal a part of the overall picture of the ordered information table from which it was derived [17]. It may happen that the ordered information table contains object pairs that match the rule's antecedent ϕ, but not satisfy the rule's consequent ψ. Hence, we are interested in the probability of the conclusion ψ being matched, given condition ϕ.

The quantity $accuracy(\phi \Rightarrow \psi)$ gives a measure of how trustworthy the rule is in drawing conclusion ψ on the basis of evidence ϕ, and is a frequency-based estimation of the conditional probability $Pr(\psi|\phi)$,

$$accuracy(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\phi)|}. \quad (7)$$

Expressions can be considered as criteria. An ordering rule states the extent to which orderings of objects by attributes in A determines orderings of objects by attributes in B. An ordering rule,

$$(a, \succ) \wedge (b, \preceq) \Rightarrow (c, \succ),$$

can be re-expressed as,

$$x \succ_{\{a\}} y \wedge x \preceq_{\{b\}} y \Rightarrow x \succ_{\{c\}} y.$$

That is, for two arbitrary objects x and y, if x is ranked ahead of y by attribute a, and at the same time, x is not ranked ahead of y by attribute b, then x is ranked ahead of y by attribute c. If $accuracy = 1$, the orderings by ϕ would determine the orderings by ψ. We thus have a strong association between the two orderings. A smaller value of *accuracy* indicates a weaker association.

Usually we also want a rule to be strong in the sense that it has a large support basis. However, what we consider to be "large" typically varies with how the decision values are distributed. The quantity $coverage(\phi \Rightarrow \psi)$ gives a measure of how well the antecedent ϕ describes the consequent ψ, and is a frequency-based estimation of the conditional probability $Pr(\phi|\psi)$,

$$coverage(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\psi)|}. \quad (8)$$

An ordering rule with higher coverage suggests that ordering of more pairs of objects can be derived from the rule.

Example 3 *From the data in Example 1, we can get, for example, two ordering rules:*

$$(b, \preceq) \wedge (c, \preceq) \Rightarrow (o, \preceq), \; accuracy = 8/8 = 1.0,$$
$$(c, \succ) \Rightarrow (o, \succ), \; accuracy = 5/8 = 0.625.$$

For these two ordering rules, the corresponding measures of coverage are:

$$(b, \preceq) \wedge (c, \preceq) \Rightarrow (o, \preceq), coverage = 8/13 = 0.615,$$
$$(c, \succ) \Rightarrow (o, \succ), coverage = 5/7 = 0.714.$$

The accuracy and coverage are not independent of each other, as both are related to the quantity $|m(\phi \wedge \psi)|$. It is desirable for a rule to be accurate as well as to have a high degree of coverage. In general, one may observe a trade-off between accuracy and coverage. A rule with higher coverage may have a lower accuracy, while a rule with higher accuracy may have a lower coverage.

5 Mining Ordering Rules

To mine ordering rules from an ordered information table, we present a rough set based approach. From an ordered information table, we can construct a binary information table. In the binary information, we consider all pairs of objects which are the Cartesian product $U \times U$. The information function is defined by:

$$I_a((x,y)) = \begin{cases} 1, & x \succ_{\{a\}} y, \\ 0, & x \preceq_{\{a\}} y. \end{cases} \quad (9)$$

Statements in ordered information tables can be translated into equivalent statements in the binary information table. For example, $x \succ_{\{a\}} y$ can be translated into $I_a((x,y)) = 1$. In the translation process, we will not consider object pairs of the form (x,x), as we are not interested in them.

In a binary information table, we define an equivalence relation E_A for a subset of attributes $A \subseteq At$:

$$(x,y)E_A(x',y') \iff (\forall a \in A) I_a((x,y)) = I_a((x',y')). \quad (10)$$

The overall ordering attribute o partitions all pairs of objects into two disjoint classes denoted by Cl_0 and Cl_1. The lower and upper approximations of $Cl_i (i = 1, 2)$ based on attributes in A are given by:

$$\underline{apr}(Cl_i) = \bigcup \{[(x,y)]_A \mid [(x,y)]_A \subseteq Cl_i\},$$
$$\overline{apr}(Cl_i) = \bigcup \{[(x,y)]_A \mid [(x,y)]_A \cap Cl_i \neq \emptyset\}, (11)$$

where $[(x,y)]_A$ is the equivalence class containing (x,y) induced by equivalence relation E_A. We can also find the *reduct* and *core* of the ordering attributes A to eliminate the redundant ones.

For each equivalence class $[(x,y)]_A \in \underline{apr}(Cl_i)$, we can draw a certain ordering rule:

$$\text{Des}([(x,y)]_A) \Rightarrow \text{Des}(Cl_i)$$

where $\text{Des}([(x,y)]_A)$ and $\text{Des}(Cl_i)$ denote the descriptions of the corresponding equivalence classes. For every ordering attribute $a \in A$, we can get an atomic expression in $\text{Des}([(x,y)]_A)$: $(a, \succ)$ if $I_a((x,y)) = 1$, and $(a, \preceq)$ if $I_a((x,y)) = 0$. The conjunction of these atomic expressions is $\text{Des}([(x,y)]_A)$. $\text{Des}(Cl_i)$ denotes one of the two atomic expressions for the overall ordering: $(o, \succ)$ if $i = 1$, and $(o, \preceq)$ if $i = 0$. That is, $\text{Des}([(x,y)]_A)$ and $\text{Des}(Cl_i)$ are expressions in *ODL*-language.

For each equivalence class $[(x,y)]_A \in \overline{apr}(Cl_i)$, we can draw a possible ordering rule with accuracy and coverage measures:

$$\text{Des}([(x,y)]_A) \Rightarrow \text{Des}(Cl_i),$$
$$accuracy = \frac{|m(\text{Des}([(x,y)]_A) \wedge \text{Des}(Cl_i))|}{|m(\text{Des}([(x,y)]_A))|},$$
$$coverage = \frac{|m(\text{Des}([(x,y)]_A) \wedge \text{Des}(Cl_i))|}{|m(\text{Des}(Cl_i))|}. (12)$$

The construction of possible rules is particularly useful for the analysis of large data sets where inconsistencies may considerably reduce the lower approximations and prevent discovery of strong rules.

Although the rules we get from reducts are somewhat simplified rules, there are many methods introduced in the area of rough set which can induce a set of minimal ordering rules from an information table. For example, using Rosetta [18], a rough set toolkit for analyzing data, we can get a set of minimal ordering rules based on reducts that discern on a per object basis.

Example 4 *The ordered information table in Example 1 can be transformed into the binary information table, given by Table 2. Using rough set theory, we can get the reduct of the given set of ordering attributes: $\{b, c\}$, which produces the following rough set approximations:*

$$\begin{aligned} \underline{apr}(Cl_1) &= \emptyset, \\ \underline{apr}(Cl_0) &= \{[b=0, c=0]\}, \\ \overline{apr}(Cl_1) &= \{[b=1, c=0], [b=0, c=1]\}, \\ \overline{apr}(Cl_0) &= \{[b=1, c=0], [b=0, c=1], \\ & \quad [b=0, c=0]\}, \end{aligned}$$

The lower approximations produce an ordering rule:

$$R_1 : \quad (b, \preceq) \wedge (c, \preceq) \Rightarrow (o, \preceq), \ accuracy = 1, \ coverage = 0.615.$$

That is, if $x \preceq_{\{b\}} y$ and $x \preceq_{\{c\}} y$ then $x \preceq_{\{o\}} y$. Examples of possible ordering rules obtained from the upper approximations are:

$$R_2 : \quad (c, \succ) \Rightarrow (o, \succ), \ accuracy = 0.625, \ coverage = 0.714,$$
$$R_3 : \quad (b, \succ) \Rightarrow (o, \succ), \ accuracy = 0.5, \ coverage = 0.286.$$

That is, if $x \succ_{\{c\}} y$ then $x \succ_{\{o\}} y$ with accuracy 0.625 and coverage 0.714. If $x \succ_{\{b\}} y$ then $x \succ_{\{o\}} y$ with accuracy 0.5 and coverage 0.286.

In our approach, the interpretation of an ordered information table and the translation to a binary information table are crucial. Once we obtain the binary information table, any existing rough set based data mining algorithm can be used to mine ordering rules.

One may also use other types of translation methods. For example, we may consider two strict order relations $\succ$ and $\prec$, instead of $\succ$ and $\preceq$. Alternatively, one may translate an ordered information table into a three-valued information table, corresponding to $\succ$, $\prec$, and $\sim$. It is important to realize that the framework presented in this paper can be easily applied with very simple modification.

Object	a	b	c	d	o
$(1,2)$	1	0	1	1	1
$(1,3)$	0	0	1	0	1
$(1,4)$	0	0	1	0	1
$(1,5)$	0	1	0	0	1
$(2,1)$	0	0	0	0	0
$(2,3)$	0	0	0	0	0
$(2,4)$	0	0	0	0	0
$(2,5)$	0	1	0	0	0
$(3,1)$	1	0	0	1	0
$(3,2)$	1	0	0	1	0
$(3,4)$	0	0	0	0	0
$(3,5)$	0	1	0	0	0
$(4,1)$	1	0	0	1	0
$(4,2)$	1	0	1	1	1
$(4,3)$	0	0	1	1	1
$(4,5)$	0	1	0	0	1
$(5,1)$	1	0	0	1	0
$(5,2)$	1	0	1	1	0
$(5,3)$	0	0	1	1	0
$(5,4)$	0	0	1	0	0

Table 2: A binary information table derived from an ordered information table

6 Conclusions

Ordering of objects is a fundamental issue in human decision making and may play a significant role in the design of intelligent information systems. This problem is considered from the perspective of data mining. The commonly used attribute value approaches are extended by introducing order relations on attribute values. Mining ordering rules is formulated as the process of finding associations between orderings on attribute values and the overall ordering of objects. These ordering rules tell us, or explain, how objects should be ranked according to orderings on their attribute values. A rough set based approach is used to illustrate our basic ideas.

The proposed solution for mining ordering rules is simple. Our main contribution is the formulation of the problem, and the translation of the problem to existing data mining problem. Consequently, one can directly apply any existing data mining algorithms for mining ordering rules. Depending on the specific problem, one may use different translation methods.

References

[1] Bell, A. Discovery and maintenance of functional dependencies by independencies, *Proceedings of KDD-95*, pp. 27-32, 1995.

[2] Butz, C.J., Wong, S.K.M., and Yao, Y.Y. On data and probabilistic dependencies, *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1692-1697, 1999.

[3] Cohen, W.W., Schapire R.E., and Singer Y. Learning to order things, *Advances in Neural Information Processing Systems*, **10**, 1998.

[4] Greco, S., Matarazzo, B., and Slowinski, R. The use of rough sets and fuzzy sets in MCDM, In: *Advances in Multiple Criteria Decision Making*, Gal, T., Hanne, T., and Stewart, T. (Eds.) Kluwer Academic Publishers, Boston, pp. 14.1-14.59, 1999.

[5] Greco, S., Matarazzo, B., and Slowinski, R. Rough approximation of a preference relation by dominance relations, *European Journal of Operational Research* **117**, 63-83, 1999.

[6] Han, J., Cai, Y., and Cercone, N., Data-driven discovery of quantitative rules in data bases, *IEEE Transactions on Knowledge and Data Engineering*, **5**, 29-40, 1993.

[7] Iwinski, T.B. Ordinal information system, I, *Bulletin of the Polish Academy of Sciences, Mathematics*, **36**, 467-475, 1988.

[8] Iwinski, T.B. Ordinal information system, II, *Bulletin of the Polish Academy of Sciences, Technical Sciences*, **39**, 157-170, 1991.

[9] Klir, G.J. and Yuan, B. *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice Hall, New Jersey, 1995.

[10] Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (Eds.) *Machine Learning*, Tioga, 1983.

[11] Orlowska, E. Reasoning about vague concepts, *Bulletin of Polish Academy of Science, Mathematics*, **35**, 643-652, 1987.

[12] Pawlak, Z. Information systems: theoretical foundations, *Information Systems*, **6**, 205-218, 1981.

[13] Pawlak, Z. Rough classification, *International Journal of Man-Machine Studies*, **20**, 469-483, 1984.

[14] Pawlak, Z. *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[15] Pawlak, Z., Slowinski, R. Rough set approach to multi-attribute decision analysis, *European Journal of Operational Research*, **72**, 443-359, 1994.

[16] Polkowski, L. and Skowron, A. (Eds.), *Rough Sets in Knowledge Discovery, I, II*, Physica-Verlag, Heidelberg, 1998.

[17] Aleksander, O. *Discernibility and Rough Sets in Medicine: Tools and Application*, PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[18] Rosetta, a rough set toolkit for analyzing data, *http://www.idi.ntnu.no/ aleks/rosetta/*, March, 2000.

[19] Tsumoto, S. Modelling medical diagnostic rules based on rough sets, *Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence, 1424*, Springer-Verlag, Berlin, pp. 475-482, 1998.

[20] Tsumoto, S. Automated discovery of plausible rules based on rough sets and rough inclusion, *Proceedings of PAKDD'99, Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, 210-219, 1999.

[21] Wasilewska, A. Conditional knowledge representation system – model for an implementation, *Bulletin of Polish Academy of Science, Mathematics*, **37**, 63-69, 1990.

[22] Yao, Y.Y. Measuring retrieval effectiveness based on user preference of documents, *Journal of the American Society for Information Science*, **46**, 133-145, 1995.

[23] Yao, Y.Y. Generalized rough set models, in: *Rough Sets in Knowledge Discovery*, Polkowski, L. and Skowron, A. (Eds.), Physica-Verlag, Heidelberg, pp. 286-318, 1998

[24] Yao, Y.Y. Information tables with neighborhood semantics, in: *Data Mining and Knowledge Discovery: Theory, Tools, and Technology II*, Dasarathy, B.V. (Ed.), Society for Optical Engineering, Bellingham, Washington, pp. 108-116, 2000.

[25] Yao, Y.Y. and Lin, T.Y. Generalization of rough sets using modal logic, *Intelligent Automation and Soft Computing, An International Journal*, **2**, 103-120, 1996.

[26] Yao, Y.Y., Wong, S.K.M., and Lin, T.Y. A review of rough set models, in: Lin, T.Y. and Cercone, N. (Eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Academic Publishers, Boston, pp. 47-75, 1997.

[27] Yao, Y.Y. and Zhong, N. An analysis of quantitative measures associated with rules, *Proceedings of PAKDD'99, Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, 479-488, 1999.

[28] Yao, Y.Y. and Zhong, N. On association, similarity and dependency of attributes, *Proceedings of PAKDD'00, Lecture Notes in Computer Science, 1805*, Springer-Verlag, Berlin, 138-141, 2000.

[29] Yao, Y.Y. and Zhong, N. Granular computing using information tables, manuscript, 2000.