

Approximation Maintenance When Adding a Conditional Value in Set-Valued Ordered Decision Systems

Xiaoyu Wang, Yuebin Su

College of Mathematics and Statistics, Sichuan University of Science and Engineering, Zigong, China
Email: 1099197627@qq.com, mathsyb@suse.edu.cn

How to cite this paper: Wang, X.Y. and Su, Y.B. (2024) Approximation Maintenance When Adding a Conditional Value in Set-Valued Ordered Decision Systems. *Open Journal of Applied Sciences*, 14, 411-424.
<https://doi.org/10.4236/ojapps.2024.142029>

Received: January 8, 2024

Accepted: February 24, 2024

Published: February 27, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The integration of set-valued ordered rough set models and incremental learning signify a progressive advancement of conventional rough set theory, with the objective of tackling the heterogeneity and ongoing transformations in information systems. In set-valued ordered decision systems, when changes occur in the attribute value domain, such as adding conditional values, it may result in changes in the preference relation between objects, indirectly leading to changes in approximations. In this paper, we effectively addressed the issue of updating approximations that arose from adding conditional values in set-valued ordered decision systems. Firstly, we classified the research objects into two categories: objects with changes in conditional values and objects without changes, and then conducted theoretical studies on updating approximations for these two categories, presenting approximation update theories for adding conditional values. Subsequently, we presented incremental algorithms corresponding to approximation update theories. We demonstrated the feasibility of the proposed incremental update method with numerical examples and showed that our incremental algorithm outperformed the static algorithm. Ultimately, by comparing experimental results on different datasets, it is evident that the incremental algorithm efficiently reduced processing time. In conclusion, this study offered a promising strategy to address the challenges of set-valued ordered decision systems in dynamic environments.

Keywords

Information Systems, Rough Set, Attribute Value, Incremental Method

1. Introduction

For the three cases of dynamic changes, the study of attribute values mainly fo-

cuses on the problem of coarsening and refining, which is the result of a certain change of an attribute value, and the attribute value domain does not change. However, when a condition value has changed, it will cause more than one attribute value to change. Luo [1] proposed a strategy and algorithm to quickly update approximations in set-valued ordered information systems for adding new condition values and removing old condition values to attribute values when the attribute value domain is changed. Each traversal in this approach requires a lookup of all objects. We find that we can preclassify all objects to reduce the time required for each traversal, and then consider the changes in the knowledge grains separately on the basis of classification.

The rest of the paper is organized as follows: in Section 2 some basic concepts about set-valued dominance relations rough set models are introduced. Section 3 describes dynamic update strategies for dominance sets and approximations in set-valued ordered decision systems when attribute values are changed. Section 4 introduces dynamic update algorithms in set-valued ordered decision systems when the attribute value is changed. In Section 5 the algorithms are evaluated experimentally by selecting some suitable datasets. In section 6 the paper concludes with conclusions and topics for further research.

2. Literature Survey

In 1982, Pawlak first proposed Rough Set Theory (RST) [2], a new mathematical tool mainly used to solve uncertain and imprecise mathematical problems. Set-valued information systems [3] are an extended model of traditional single-valued information systems, which can be used to handle missing values in incomplete information systems.

When attribute values change dynamically, Chen [4] first proposed the definition of coarsening and refining for dynamic changes in attribute values, and studied approximations maintenance when attribute values are coarsened and refined. Chen [5] divided attribute values into horizontal refining and vertical refining, proposed the definition of horizontal refining and vertical refining of attribute values, and discussed the principle of approximation updating under variable precision. Luo [6] proposed an incremental algorithm to formalize the dynamic characteristics of knowledge granules in hierarchical multi-criteria decision systems by refining and coarsening the dynamic characteristics of knowledge granules with the help of attribute value taxonomies. Chen [7] studied the approximation updating of attribute values during coarsening and refining with the help of extended feature dominance links in incomplete information systems. Li [8] performed approximation maintenance of dominance rough set with variation in the dominance matrix.

Luo [8] analyzed the approximation update problem caused by attribute value changes in set-valued ordered decision systems. Wang [10] investigated the problem of simultaneous changes in object sets and attribute values in multidimensional dynamics, and proposed an incremental algorithm for approximation

maintenance using the dominance-feature matrix approach. Luo [1] solved the knowledge updating problem caused by dynamic changes in condition values in set-value decision systems, and has given updated theories and algorithms for dynamically updating approximations of condition values with time.

3. Methods and Materials

3.1. Description of the Dataset

In this section, we briefly review some fundamental conceptions of set-valued ordered decision systems [11].

Definition 1 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be an information decision system, where $U = \{x_1, x_2, \dots, x_{|U|}\}$ is a set of all objects, known as the universe. $A = C \cup d$, where C is the collection of conditional attributes with preference relations, d is the decision attribute, and $C \cap d = \emptyset$. $V_A = V_C \cup V_d$, where V_C is a collection of conditional attributes C for all values and V_d is a collection of decision attributes d for all values. $f: U \times C \rightarrow 2^{V_C}$ is an object attribute mapping, which is a set value mapping. If all conditional attributes are in increasing or decreasing preference, then S is said to be a set-valued ordered decision system.

Example 1. Table 1 shows a set-valued ordered decision system

$S = (U, C \cup \{d\}, V, f)$, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$,

$B = C = \{a_1, a_2\} = \{\text{Reading, Writing}\}$,

$V_B = \{E, F, G\} = \{\text{English, French, German}\}$, $V_d = \{\text{poor, good}\}$.

In Table 1, $f(x_2, a_1)$ indicates that x_2 will be read in English, French, and German, and $f(x_1, a_1)$ indicates that x_1 will be read in French only. It can be concluded that x_1 does not read as much language as x_2 . That is, in the attribute a_1 , object x_2 is at least dominant to object x_1 . This type of set-valued ordered decision system is called conjunctive set-valued ordered decision system (C-SODS) [11].

In C-SODS, the conjunctive dominance relation can be defined based on the set inclusion relation.

Definition 2 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS. $\forall B \subseteq C$, the

Table 1. A set-value ordered decision system.

U	Reading	Writing	d
x_1	{F}	{E, F}	poor
x_2	{E, F, G}	{E, F, G}	good
x_3	{E, F, G}	{E, F, G}	good
x_4	{E, F}	{E, F}	good
x_5	{E, F}	{E, F, G}	good
x_6	{F}	{E, F}	poor
x_7	{E, F, G}	{E, F, G}	good
x_8	{F}	{E, F, G}	poor

conjunctive dominance relation R_B on the domain U is defined as

$$R_B = \{(y, x) \in U \times U \mid y \succ_B x\} = \{(y, x) \in U \times U \mid f(y, a) \supseteq f(x, a), \forall a \in B\},$$

It is clear that the conjunctive dominance relation satisfies reflexive, transitive, and not symmetry.

Definition 3 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS. $\forall B \subseteq C, B \neq \emptyset$ two knowledge granules of x are called B -dominating set and B -dominated set, which are respectively defined as follows

$$[x]_B^+ = \{y \in U \mid y R_B x\},$$

$$[x]_B^- = \{y \in U \mid x R_B y\}.$$

Definition 4 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS. If the U can be divided into m equivalence sets, the equivalence sets are called decision sets. It can be expressed as:

$$Cl = \{Cl_n, n \in T\}, \text{ where } T = \{1, \dots, m\}.$$

Definition 5 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS. If the U can be divided into m decision sets according to the decision attribute d , upward union Cl_n^{\geq} and downward union Cl_n^{\leq} can be defined as:

$$Cl_n^{\geq} = \bigcup_{n' \geq n} Cl_{n'},$$

$$Cl_n^{\leq} = \bigcup_{n' \leq n} Cl_{n'}, \quad \forall n, n' \in T.$$

where $x \in Cl_n^{\geq}$ means “ x belongs to at least Cl_n ”; $x \in Cl_n^{\leq}$ means “ x belongs to at most Cl_n ”.

Definition 6 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS, for any $B \subseteq C$, the lower and upper approximations of Cl_n^{\geq} are respectively defined as follow:

$$\underline{R}_B(Cl_n^{\geq}) = \{x \in U \mid [x]_B^+ \subseteq Cl_n^{\geq}\}$$

$$\overline{R}_B(Cl_n^{\geq}) = \{x \in U \mid [x]_B^- \cap Cl_n^{\geq} \neq \emptyset\}$$

Definition 7 [11]. Let $S = (U, C \cup \{d\}, V, f)$ be a C-SODS, for any $B \subseteq C$, the lower and upper approximations of Cl_n^{\leq} are respectively defined as follow:

$$\underline{R}_B(Cl_n^{\leq}) = \{x \in U \mid [x]_B^- \subseteq Cl_n^{\leq}\}$$

$$\overline{R}_B(Cl_n^{\leq}) = \{x \in U \mid [x]_B^+ \cap Cl_n^{\leq} \neq \emptyset\}$$

Example 2. Continuing from Example 1, we can use the above definition to further compute the dominating and dominated sets, as well as the upper and lower approximations of the set-valued ordered decision system.

Dominating and dominated sets:

$$[x_1]_B^+ = [x_6]_B^+ = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, \quad [x_2]_B^+ = [x_3]_B^+ = [x_7]_B^+ = \{x_2, x_3, x_7\},$$

$$[x_4]_B^+ = \{x_2, x_3, x_4, x_5, x_7\}, \quad [x_5]_B^+ = \{x_2, x_3, x_5, x_7\}, \quad [x_8]_B^+ = \{x_2, x_3, x_5, x_7, x_8\}.$$

$$[x_1]_B^- = [x_6]_B^- = \{x_1, x_6\}, [x_2]_B^- = [x_3]_B^- = [x_7]_B^- = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, \\ [x_4]_B^- = \{x_1, x_6, x_4\}, [x_5]_B^- = \{x_1, x_4, x_5, x_6, x_8\}, [x_8]_B^- = \{x_1, x_6, x_8\}.$$

Decision classes:

$$Cl_1 = \{x_1, x_6, x_8\}, Cl_2 = \{x_2, x_3, x_4, x_5, x_7\}$$

Upward and downward unions of decision classes:

$$Cl_1^{\geq} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, Cl_2^{\geq} = \{x_2, x_3, x_4, x_5, x_7\} \\ Cl_1^{\leq} = \{x_1, x_6, x_8\}, Cl_2^{\leq} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

Approximations:

$$\underline{R}_B(Cl_1^{\geq}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} \\ \underline{R}_B(Cl_2^{\geq}) = \{x_2, x_3, x_4, x_5, x_7\} \\ \overline{R}_B(Cl_1^{\geq}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} \\ \overline{R}_B(Cl_2^{\geq}) = \{x_2, x_3, x_4, x_5, x_7\} \\ \underline{R}_B(Cl_1^{\leq}) = \{x_1, x_6, x_8\} \\ \underline{R}_B(Cl_2^{\leq}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} \\ \overline{R}_B(Cl_1^{\leq}) = \{x_1, x_6, x_8\} \\ \overline{R}_B(Cl_2^{\leq}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

3.2. Incremental Approximations Updating Method When Adding a Conditional Value

With the variation of an information system, the structure of information granules in the information system may vary over time which leads to the change of knowledge induced by RST. In a dynamic decision system, the attribute value domain can be updated due to different requirements, such as adding new conditional values, removing old conditional values, or correcting errors. Changes in the attribute value domain can directly lead to changes in the preference relation between any pair of attributes of two objects, resulting in corresponding changes in knowledge granules.

Example 3. From **Table 1**, we consider

$V_B = \{E, F, G\} = \{\text{English, French, German}\}$. If we add Chinese to V_B , we will also use Chinese as a language criterion to judge Reading and Writing. Then, the attribute value domain will be updated to

$V_B = \{E, F, G, C\} = \{\text{English, French, German, Chinese}\}$. At this point, **Table 1** is updated to **Table 2**. x_1 is updated to $\{F, C\}$ under a_1 in **Table 2**, so after updating, x_1 and x_2 are not comparable in Reading.

When the preference relation changes, the discovered knowledge may become invalid, or some new implicit information may emerge in the whole updated information system. Rather than restarting from scratch by the non-incremental

Table 2. The condition value added into the set-value ordered decision system.

U	Reading	Writing	d
x_1	$\{F, C\}$	$\{E, F\}$	poor
x_2	$\{E, F, G\}$	$\{E, F, G\}$	good
x_3	$\{E, F, G, C\}$	$\{E, F, G\}$	good
x_4	$\{E, F\}$	$\{E, F\}$	good
x_5	$\{E, F\}$	$\{E, F, G\}$	good
x_6	$\{F, C\}$	$\{E, F\}$	poor
x_7	$\{E, F, G\}$	$\{E, F, G\}$	good
x_8	$\{F\}$	$\{E, F, G\}$	poor

or batch learning algorithm for each update, developing an efficient incremental algorithm to avoid unnecessary computations by utilizing the previous data structures or results is thus desired.

In this section, we discuss the variation of approximations in the dynamic C-SODS when the conditional value evolves over time while the object set and attribute set remain constant. For convenience, we assume the incremental learning process lasts two periods from time t to time $t + 1$. $\ast^{(t)}$ and $\ast^{(t+1)}$ represent the set-valued ordered decision system at time t and time $t + 1$, respectively. $\ast^{(+)}$ denotes adding a conditional value.

In the following, $V_a^{(t+1)}$ is denoted the updated attribute value domain of attribute a at $t + 1$, and v_a^+ is the conditional value that is added to $V_a^{(t)}$, where $V_a^{(t+1)} = V_a^{(t)} \cup v_a^+$.

Definition 8. For any $x \in U$, $a \in B$, where U^+ denotes the set of all objects with condition values added, and $\bar{U} = U - U^+$ denotes the set of objects without condition values added. If $v_a^+ \in f^{(t+1)}(y, a)$, then $y \in U^+$; if $v_a^+ \notin f^{(t+1)}(x, a)$, then $x \in \bar{U}$.

When $V_a^{(t)}$ changes, the dominating sets and dominated sets of y need to be updated. If x at time t dominates y , when y adds v_a^+ at time $t + 1$, and x does not contain v_a^+ . Thus, x no longer dominates y .

Property 1. If any $y \in U^+$, then $[y]_B^{+,(t+1)} = [y]_B^{+,(t)} - \bar{U}$.

Proof. If $x \in [y]_B^{+,(t)}$, $f^{(t+1)}(y, a) \not\subseteq f^{(t+1)}(x, a)$, then $x \notin [y]_B^{+,(t+1)}$. If $x \notin [y]_B^{+,(t)}$, $x \notin [y]_B^{+,(t+1)}$.

The approximation sets are close to the dominating sets and dominated sets. According to Property 1, only the dominating sets have changed, thus we just need to update $\underline{R}_B^{(t+1)}(CI_n^{\geq})$ and $\bar{R}_B^{(t+1)}(CI_n^{\leq})$.

Proposition 1. (1) If $\exists y \notin \underline{R}_B^{(t)}(CI_n^{\geq})$, and $\bar{U} \not\subseteq CI_n^{\geq}$, $[y]_B^{+,(t+1)} \subseteq CI_n^{\geq}$, then $\underline{R}_B^{(t+1)}(CI_n^{\geq}) = \underline{R}_B^{(t)}(CI_n^{\geq}) \cup \{y\}$.

(2) If $\exists y \in \bar{R}_B^{(t)}(CI_n^{\leq})$, and $\bar{U} \subseteq CI_n^{\leq}$, $[y]_B^{+,(t+1)} \cap CI_n^{\leq} = \emptyset$, then

$$\bar{R}_B^{(t+1)}(CI_n^{\leq}) = \bar{R}_B^{(t)}(CI_n^{\leq}) - \{y\}.$$

Proof. (1) According to the Property 1, let $\Delta^- = [y]_B^{+(t)} - CI_n^{\geq}$. Clearly, if $\bar{U} \cap \Delta^- = \emptyset$, and $[y]_B^{+(t+1)} \subseteq CI_n^{\geq}$, then $\underline{R}_B^{(t+1)}(CI_n^{\geq}) = \bar{R}_B^{(t)}(CI_n^{\geq}) \cup \{y\}$.

(2) According to Property 1, if $[y]_B^{+(t+1)} \cap CI_n^{\leq} = \emptyset$, then $\bar{R}_B^{(t+1)}(CI_n^{\leq}) = \bar{R}_B^{(t)}(CI_n^{\leq}) - \{y\}$.

Similarly to y , if y is dominated by x at time t , then it is no longer dominated by x when y adds v_a^+ at time $t + 1$ and x does not contain v_a^+ . Thus, y is no longer dominated by x .

Property 2. If any $x \in \bar{U}$, then $[x]_B^{-(t+1)} = [x]_B^{-(t)} - U^+$.

Proof. This Property can be proved by the same method as employed in Property 1.

Similar to Proposition 1, only the dominated sets changed in Property 4. We just need to discuss the updating theories for the $\underline{R}_B^{(t+1)}(CI_n^{\leq})$ and $\bar{R}_B^{(t+1)}(CI_n^{\geq})$.

Proposition 2. (1) If $\exists x \notin \underline{R}_B^{(t)}(CI_n^{\leq})$, and $U^+ \not\subseteq CI_n^{\leq}$, $[x]_B^{-(t+1)} \subseteq CI_n^{\leq}$, then

$$\underline{R}_B^{(t+1)}(CI_n^{\leq}) = \underline{R}_B^{(t)}(CI_n^{\leq}) \cup \{x\}.$$

(2) If $\exists x \in \bar{R}_B^{(t)}(CI_n^{\geq})$, and $U^+ \subseteq CI_n^{\geq}$, $[x]_B^{-(t+1)} \cap CI_n^{\leq} = \emptyset$, then $\bar{R}_B^{(t+1)}(CI_n^{\geq}) = \bar{R}_B^{(t)}(CI_n^{\geq}) - \{x\}$.

Proof. This proposition can be proved by the same method as employed in Proposition 1.

Example 4. Continuing from Example 3, we can use Properties 1 - 4 and Propositions 1 - 4 to update the dominating sets and dominated sets, and approximations.

Updating dominating sets and dominated sets:

$$\begin{aligned} [x_1]_B^{+(t+1)} &= [x_1]_B^{+(t)} - \{x_1, x_4, x_5, x_6, x_8\} = \{x_2, x_3, x_7\}, \\ [x_3]_B^{+(t+1)} &= [x_3]_B^{+(t)} - \{x_2, x_7\} = \{x_3\}, \\ [x_4]_B^{+(t+1)} &= [x_4]_B^{+(t)} - \{x_1, x_4, x_5, x_6, x_8\} = \{x_2, x_3, x_7\}, \\ [x_2]_B^{-(t+1)} &= [x_2]_B^{-(t)} - \{x_1, x_6\} = \{x_2, x_3, x_4, x_5, x_7, x_8\}, \\ [x_3]_B^{-(t+1)} &= [x_3]_B^{-(t)} - \{x_1, x_6\} = \{x_2, x_3, x_4, x_5, x_7, x_8\}, \\ [x_4]_B^{-(t+1)} &= [x_4]_B^{-(t)} - \{x_1, x_6\} = \{x_4\}, \\ [x_5]_B^{-(t+1)} &= [x_5]_B^{-(t)} - \{x_1, x_6\} = \{x_4, x_5, x_8\}, \\ [x_7]_B^{-(t+1)} &= [x_7]_B^{-(t)} - \{x_1, x_6\} = \{x_2, x_3, x_4, x_5, x_7, x_8\}, \\ [x_8]_B^{-(t+1)} &= [x_8]_B^{-(t)} - \{x_1, x_6\} = \{x_8\}. \end{aligned}$$

Updating approximations:

$$\begin{aligned} \underline{R}_B^{(t+1)}(CI_1^{\geq}) &= \underline{R}_B^{(t)}(CI_1^{\geq}), \quad \underline{R}_B^{(t+1)}(CI_2^{\geq}) = \underline{R}_B^{(t)}(CI_2^{\geq}), \quad \bar{R}_B^{(t+1)}(CI_1^{\geq}) = \bar{R}_B^{(t)}(CI_1^{\geq}), \\ \bar{R}_B^{(t+1)}(CI_2^{\geq}) &= \bar{R}_B^{(t)}(CI_2^{\geq}). \end{aligned}$$

$$\begin{aligned} \underline{R}_B^{(t+1)}(CI_1^{\leq}) &= \underline{R}_B^{(t)}(CI_1^{\leq}), \quad \underline{R}_B^{(t+1)}(CI_2^{\leq}) = \underline{R}_B^{(t)}(CI_2^{\leq}), \quad \overline{R}_B^{(t+1)}(CI_1^{\leq}) = \overline{R}_B^{(t)}(CI_1^{\leq}), \\ \overline{R}_B^{(t+1)}(CI_2^{\leq}) &= \overline{R}_B^{(t)}(CI_2^{\leq}). \end{aligned}$$

3.3. Static and Incremental Algorithms for Adding a Conditional Value in Dynamic C-SODS

3.3.1. Static Algorithm in C-SODS

In this section, we propose a static algorithm in C-SODS, namely **Algorithm 1**.

3.3.2. Incremental Algorithm When Adding a Conditional Value in C-SODS

In this section, we propose an approximation updating algorithm in C-SODS based on adding a conditional value, namely **Algorithm 2**.

4. Experimental Results

In this section, we perform the following experiments to test the performance of the proposed incremental algorithm for approximations when adding a conditional value. However, datasets cannot be found the set-valued in existing databases, we selected 6 datasets with missing values as shown in **Table 3** in UCI [12], and used the set of all missing values as a special case. All experiments are conducted on a personal computer with 11th Gen Intel(R) Core(TM) i5-11400H

Algorithm 1. Static algorithm in C-SODS

Input: (1) The set-valued information system;

(2) upward unions and downward unions: $CI_n^{\geq}, CI_n^{\leq} (0 \leq n \leq t)$

Output: $\underline{R}_B(CI_n^{\geq}), \overline{R}_B(CI_n^{\geq}), \underline{R}_B(CI_n^{\leq}), \overline{R}_B(CI_n^{\leq})$

```

1: for  $\forall x \in U$ 
2:    $\forall y \in U$ 
3:     if  $f(x, B) \subseteq f(y, B)$  then
4:        $[x]_B^+ \leftarrow [x]_B^+ \cup \{y\}$ 
5:     else
6:        $[x]_B^- \leftarrow [x]_B^- \cup \{y\}$ 
7:     if  $[x]_B^+ \subseteq CI_n^{\geq}$  then
8:        $\underline{R}_B(CI_n^{\geq}) \leftarrow \underline{R}_B(CI_n^{\geq}) \cup \{x\}$ 
9:     if  $[x]_B^- \cap CI_n^{\geq} = \emptyset$  then
10:       $\overline{R}_B(CI_n^{\geq}) \leftarrow \overline{R}_B(CI_n^{\geq}) \cup \{x\}$ 
11:    if  $[x]_B^- \subseteq CI_n^{\leq}$  then
12:       $\underline{R}_B(CI_n^{\leq}) \leftarrow \underline{R}_B(CI_n^{\leq}) \cup \{x\}$ 
13:    if  $[x]_B^+ \cap CI_n^{\leq} = \emptyset$  then
14:       $\overline{R}_B(CI_n^{\leq}) \leftarrow \overline{R}_B(CI_n^{\leq}) \cup \{x\}$ 
15: Return  $\underline{R}_B(CI_n^{\geq}), \overline{R}_B(CI_n^{\geq}), \underline{R}_B(CI_n^{\leq}), \overline{R}_B(CI_n^{\leq})$ 

```

Algorithm 2. Incremental algorithm when adding a conditional value in C-SODS

Input: (1) The set-valued ordered decision information system at time $t + 1$;
 (2) The dominating sets and dominated sets for each x at time t : $[x]_B^{+(t)}$, $[x]_B^{-(t)}$, $\forall x \in U$;
 (3) The upward unions and downward unions: CI_n^{\geq} , CI_n^{\leq} ($0 \leq n \leq t$);
 (4) The approximations at time t : $\underline{R}_B^{(t)}(CI_n^{\geq})$, $\bar{R}_B^{(t)}(CI_n^{\geq})$, $\underline{R}_B^{(t)}(CI_n^{\leq})$, $\bar{R}_B^{(t)}(CI_n^{\leq})$;
 (5) A object sets which add a conditional value: U^+ ; (6) A object sets in which the addition of a conditional value has not occurred: \bar{U} .
 Output: $\underline{R}_B^{(t+1)}(CI_n^{\geq})$, $\bar{R}_B^{(t+1)}(CI_n^{\geq})$, $\underline{R}_B^{(t+1)}(CI_n^{\leq})$, $\bar{R}_B^{(t+1)}(CI_n^{\leq})$

```

1: if  $v_a^+ \in f^{(t+1)}(y, a)$  then
2:    $U^+ \leftarrow U^+ \cup \{y\}$ 
3: else
4:    $\bar{U} \leftarrow \bar{U} \cup \{x\}$ 
5: for  $\forall y \in U^+$ 
6:    $[y]_B^{+(t+1)} \leftarrow [y]_B^{+(t)} - \bar{U}$ 
7:   if  $\exists y \notin \underline{R}_B^{(t)}(CI_n^{\geq})$  and  $\bar{U} \not\subseteq CI_n^{\geq}$  and  $[y]_B^{+(t+1)} \subseteq CI_n^{\geq}$  then
8:      $\underline{R}_B^{(t+1)}(CI_n^{\geq}) \leftarrow \underline{R}_B^{(t)}(CI_n^{\geq}) \cup \{y\}$ 
9:   if  $\exists y \in \bar{R}_B^{(t)}(CI_n^{\leq})$  and  $\bar{U} \subseteq CI_n^{\leq}$  and  $[y]_B^{+(t+1)} \cap CI_n^{\leq} = \emptyset$  then
10:     $\bar{R}_B^{(t+1)}(CI_n^{\leq}) \leftarrow \bar{R}_B^{(t)}(CI_n^{\leq}) - \{y\}$ 
11: for  $\forall x \in \bar{U}$ 
12:    $[x]_B^{-(t+1)} \leftarrow [x]_B^{-(t)} - U^+$ 
13:   if  $\exists x \notin \underline{R}_B^{(t)}(CI_n^{\leq})$  and  $U^+ \not\subseteq CI_n^{\leq}$  and  $[x]_B^{-(t+1)} \subseteq CI_n^{\leq}$  then
14:      $\underline{R}_B^{(t+1)}(CI_n^{\leq}) \leftarrow \underline{R}_B^{(t)}(CI_n^{\leq}) \cup \{x\}$ 
15:   if  $\exists x \in \bar{R}_B^{(t)}(CI_n^{\geq})$  and  $U^+ \subseteq CI_n^{\geq}$  and  $[x]_B^{-(t+1)} \cap CI_n^{\geq} = \emptyset$  then
16:      $\bar{R}_B^{(t+1)}(CI_n^{\geq}) \leftarrow \bar{R}_B^{(t)}(CI_n^{\geq}) - \{x\}$ 
17: Return  $\underline{R}_B^{(t+1)}(CI_n^{\geq})$ ,  $\bar{R}_B^{(t+1)}(CI_n^{\geq})$ ,  $\underline{R}_B^{(t+1)}(CI_n^{\leq})$ ,  $\bar{R}_B^{(t+1)}(CI_n^{\leq})$ 
    
```

Table 3. Parameter values.

Datasets	Abbreviation	Samples	Attributes	Classes
Audiology (Standardized)	Audiology	226	69	24
Soybean (Large)	Soybean	307	35	19
Dermatology	Dermatology	366	34	6
Congressional Voting Records	CVR	435	16	2
Credit Approval	Credit	690	15	2
Mammographic Mass	Mammographic	961	6	2

2.69 GHz processor, 16.0 GB of RAM, and Windows 10 Home Edition. Algorithms based on python 3.9.12.

The purpose of the experiments in this section is to show the computation time of the static algorithm and the incremental algorithm for approximation maintenance when adding a conditional value on different number of object sets, and to prove that our proposed incremental algorithm is faster at updating approximations by comparing the computation time of the two algorithms. To better compare the time consumption of the two algorithms, we divide each dataset into ten equal parts. The first part is treated as dataset 1, the first and second parts are treated as dataset 2, the first, second and third parts are treated as dataset 3, ..., and the data of all parts are treated as dataset 10. All datasets obtained from these reorganizations will be used to calculate the time consumption of the two algorithms.

In this subsection, when a conditional value is added, the attribute value changes accordingly. We compare the time consumption of the static algorithm and the incremental algorithm on approximation updating for the 6 datasets in **Table 4**. First, for each dataset, we execute the static algorithm to compute dominating sets, dominated sets and approximations, and save all these results. Then, when a conditional value is added, we randomly select 10% attribute values in each dataset to add the conditional value, and execute the static algorithm and incremental algorithm procedures to update the approximations, respectively.

The specific time taken by the two algorithms to add condition values for approximation updating with the change in dataset size is shown in **Table 4**, and the unit of experimental results is second (s).

The experimental results in **Table 4** show that when a conditional value is added, both static and incremental algorithms show an upward trend in the time taken to compute approximations when the base of the object sets increases. The time consumption of the static algorithm is significantly higher than the incremental algorithm, and the time difference becomes larger with the increase of

Table 4. Time consumption of static and incremental algorithms when adding a conditional value with different sizes of data.

No.	Audiology Samples		Soybean		Dermatology		CVR		Credit		Mammographic	
	S.	I.	S.	I.	S.	I.	S.	I.	S.	I.	S.	I.
1	0.0019	0.0009	0.0039	0.0009	0.0049	0.0010	0.0069	0.0009	0.0179	0.0009	0.0300	0.0010
2	0.0069	0.010	0.0129	0.0009	0.0190	0.0019	0.0249	0.0009	0.0739	0.0019	0.1130	0.0030
3	0.0170	0.0029	0.0309	0.0019	0.0440	0.0039	0.0569	0.0019	0.1619	0.0029	0.2540	0.0069
4	0.0319	0.0049	0.0510	0.0059	0.0759	0.0069	0.0981	0.0029	0.5879	0.0059	0.4462	0.0129
5	0.0569	0.0109	0.0800	0.0089	0.1168	0.0099	0.1589	0.0049	0.4499	0.0109	0.7089	0.0200
6	0.0889	0.0179	0.1259	0.0160	0.1772	0.0129	0.2339	0.0060	0.6510	0.0149	0.9997	0.0300
7	0.1232	0.0230	0.1772	0.0270	0.2474	0.0179	0.3569	0.0089	0.8749	0.0189	1.3619	0.0409
8	0.1670	0.0319	0.2400	0.0339	0.3280	0.0240	0.4329	0.0129	1.2490	0.0260	1.7942	0.0540
9	0.2260	0.0359	0.3114	0.0459	0.4310	0.0330	0.5764	0.0169	1.6029	0.0349	2.2851	0.0691
10	0.2789	0.0480	0.4279	0.0649	0.5370	0.0449	0.7289	0.0210	1.9683	0.0419	2.8113	0.0840

the data set. Therefore, our proposed incremental method is more efficient than the static algorithm in terms of computational efficiency.

To compare the difference in computational efficiency between the two algorithms, based on **Table 4**, we draw two trend lines to represent the changes in the two methods. **Figure 1** shows a more detailed trend for both algorithms. In each subplot of **Figure 1**, the x coordinate is related to the size of the dataset (starting from the once dataset to the tenth dataset), and the y coordinate is related to the running time of computing the approximations. Diamond marking and circular marking indicate the computational efficiency of incremental and static methods, respectively. From **Figure 1**, it can be seen that the computation time of the two algorithms increases as the amount of data increases. However, the incremental algorithm is much faster than the static algorithm due to the additional learning of the original information, which effectively reduces the time consumption of repeated computation. The more data sets vary, the greater the difference. To summarize, incremental algorithms are more efficient than static algorithms in terms of computation time when adding a conditional value.

5. Conclusion

In rough set theory, approximations plays a vital role in the processes of knowledge acquisition and decision-making. With the advent of the big data era, the volume of information is growing at an unprecedented rate, leading to dynamic changes in the knowledge obtained from rough sets over time. Using incremental methods for approximation maintenance can avoid redundant comparisons

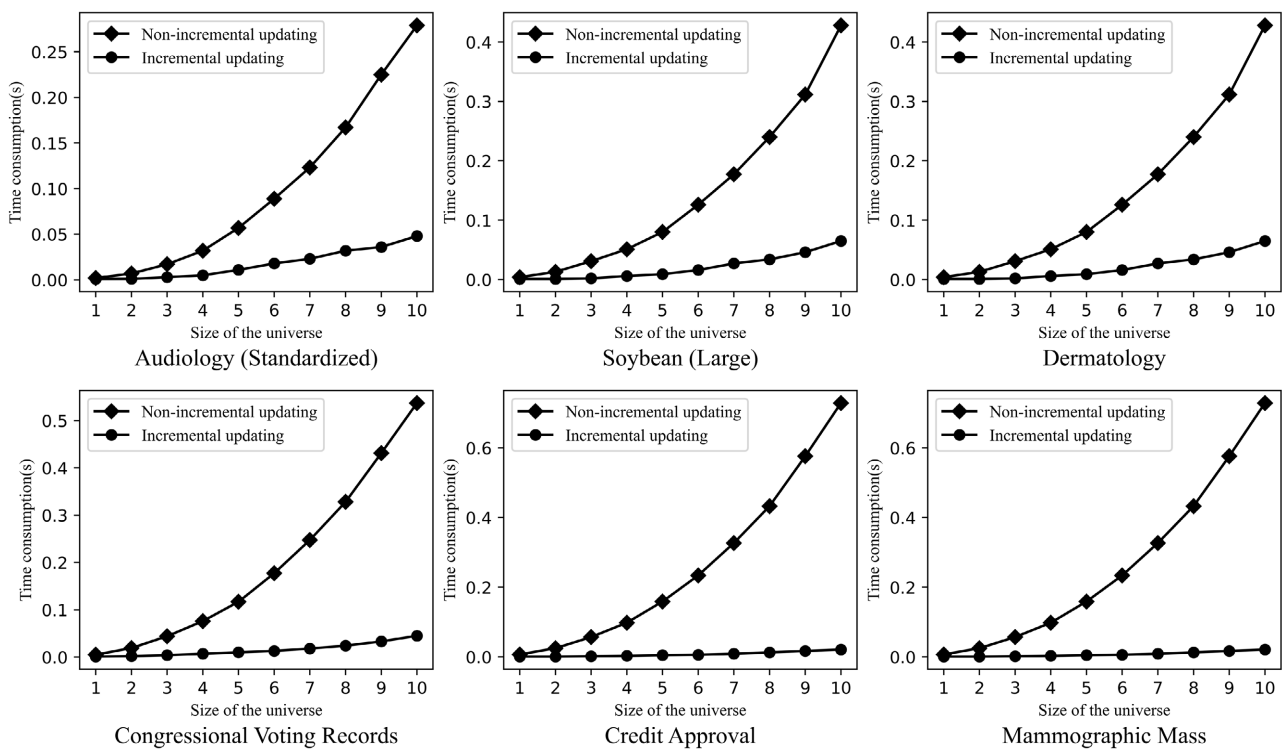


Figure 1. Comparison of static and incremental algorithms when adding a conditional value in different sizes of data.

of the original information and reduce computation time. In this paper, we extend the traditional single-valued information system to a set-valued ordered decision system, investigate the problem of updating approximations when adding a conditional value from an attribute value domain, and propose corresponding updating theories and algorithms. In this paper, we specifically consider changes which adding a conditional value in set-valued ordered decision systems. In future research, we plan to consider the change which removing a conditional value in set-valued ordered decision systems.

Acknowledgements

This work was supported by Open Fund of Sichuan Provincial Key Laboratory of Artificial Intelligence (2018RYJ02), Talent Introduction Project of Sichuan University of Science and Engineering (2019RC08) and University Student Innovation and Entrepreneurship Training Program Project, Sichuan University of Science and Engineering (cx2022147).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

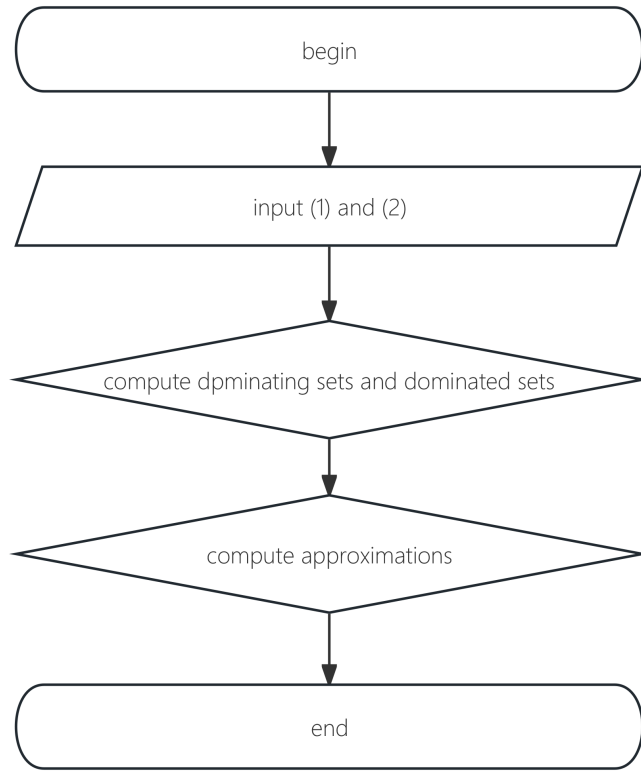
- [1] Luo, C., Li, T., Chen, H. and Lu, L. (2015) Fast Algorithms for Computing Rough Approximations in Set-Valued Decision Systems While Updating Criteria Values. *Information Sciences*, **299**, 221-242. <https://doi.org/10.1016/j.ins.2014.12.029>
- [2] Beaubouef, T. and Petry, F.E. (2009) Rough Sets. Database Technologies: Concepts, Methodologies, Tools, and Applications. <https://dl.acm.org/doi/book/10.5555/1550811>
- [3] Yao, Y. (2001) Information Granulation and Rough Set Approximation. *International Journal of Intelligent Systems*, **16**, 87-104. [https://doi.org/10.1002/1098-111X\(200101\)16:1%3C87::AID-INT7%3E3.0.CO;2-S](https://doi.org/10.1002/1098-111X(200101)16:1%3C87::AID-INT7%3E3.0.CO;2-S)
- [4] Chen, H., Li, T., Qiao, S. and Ruan, D. (2010) A Rough Set Based Dynamic Maintenance Approach for Approximations in Coarsening and Refining Attribute Values. *International Journal of Intelligent Systems*, **25**, 1005-1026. <https://doi.org/10.1002/int.20436>
- [5] Chen, H., Li, T. and Zhang, J. (2010) A Method for Incremental Updating Approximations Based on Variable Precision Set-Valued Ordered Information Systems. 2010 *IEEE International Conference on Granular Computing*, San Jose, 14-16 August 2010, 96-101. <https://doi.org/10.1109/GrC.2010.115>
- [6] Luo, C., Li, T., Chen, H., Fujita, H. and Yi, Z. (2018) Incremental Rough Set Approach for Hierarchical Multicriteria Classification. *Information Sciences*, **429**, 72-87. <https://doi.org/10.1016/j.ins.2017.11.004>
- [7] Chen, H., Li, T. and Ruan, D. (2012) Maintenance of Approximations in Incomplete Ordered Decision Systems While Attribute Values Coarsening or Refining. *Knowledge-Based Systems*, **31**, 140-161. <https://doi.org/10.1016/j.knosys.2012.03.001>
- [8] Li, S. and Li, T. (2015) Incremental Update of Approximations in Dominance-Based Rough Sets Approach under the Variation of Attribute Values. *Information Sciences*,

294, 348-361. <https://doi.org/10.1016/j.ins.2014.09.056>

- [9] Luo, C., Lu, L., Li, T., Zeng, A. and Chen, H. (2013) An Approach for Incremental Maintenance of Approximations in Set-Valued Ordered Decision Systems While Updating Criteria Values. 2013 *Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, Edmonton, 24-28 June 2013, 819-824. <https://doi.org/10.1109/IFSA-NAFIPS.2013.6608506>
- [10] Wang, S., Li, T., Luo, C., Hu, J., Fujita, H. and Huang, T. (2020) A Novel Approach for Efficient Updating Approximations in Dynamic Ordered Information Systems. *Information Sciences*, **507**, 197-219. <https://doi.org/10.1016/j.ins.2019.08.046>
- [11] Qian, Y., Dang, C., Liang, J. and Tang, D. (2009) Set-Valued Ordered Information Systems. *Information Sciences*, **179**, 2809-2832. <https://doi.org/10.1016/j.ins.2009.04.007>
- [12] Kelly, M., Longjohn, R. and Nottingham, K. (2023) The UCI Machine Learning Repository. <https://archive.ics.uci.edu>

Appendix

Algorithmic 1. Flow Chart



Algorithmic 2. Flow Chart

