CHAPTER-V

RELATIVE REDUCTS IN OBJECT-ORIENTED ROUGH SET MODELS
5.1. INTRODUCTION

This chapter propose a new Reduct in the Object-Oriented Rough Set Model called Relative Reduct by treating objects as instances of classes, and the structural hierarchies among them on, is-a relationship and has-a relationship[30]. It also proposes Heuristic-Filter Base Algorithm for finding Reducts. The Reduct finding in the Object-Oriented Rough Set Models suffer intense computation of Discernibility Matrix and positive regions. In order to improve the efficiency, in this chapter, a new computation model based on Relative Class Dependency has been introduced.

In earlier chapters of the thesis, Dynamic Reduct and its properties in the Object-Oriented Rough Set Models, Parallel Reduct and properties in the Object-Oriented Rough Set Model were discussed. However finding Reducts and Dynamic Reducts in the Object-Oriented Rough Set Model require intense computation of Discernibility Matrix and Parallel Reducts in the Object-Oriented Rough Set Models require intense computation of positive regions. Thus Reduct finding in the Object-Oriented Rough Set models suffers intense computation of Discernibility Matrix and positive regions. These two problems need to be eliminated by introducing Relative
Class Dependency. So the problem of feature selection in the Object-Oriented Rough Set Models is important issue need to be discussed. Moreover there was no attempt made regarding to Relative Reducts and Heuristic Filter-Based approach in the Object-Oriented Rough Set Models. Hence, in this chapter we made an attempt to accomplish this task.

The problem of feature selection can be defined as finding M relevant features among the N original features, where M ≤ N, according to the given selection criteria. Features can be redundant or irrelevant to the classes and classes with names analysis task. An irrelevant feature does not affect the underlying structure of the classes and classes with name in any way. A redundant feature does not provide anything new in describing the underlying structure. Feature selection is a task of searching for optimal subset of all available features. Its motivation is three-fold:

1) simplifying classes and classes with name analysis;

2) improving the accuracy of classes with name analysis;

3) reducing the classes with names. The last fold is particularly important for handling huge object-oriented databases.

Feature selection has long been an active research topic within statistics, Pattern Recognition, Machine Learning and Data Mining. Most researchers have demonstrated the interest in designing new methods and improving the performance of their algorithms. These methods can be
divided into two types: exhaustive or heuristic search. The exhaustive search probes all possible subsets chosen from the original features. This is prohibitive when the number of the original features is large. In practice, the heuristic search is the way out of this exponential computation and in general makes use of background information to approximately estimate the relevance of features. Although the heuristic search works reasonably well, it is certain that some features with high order correlation may be missed out.

An Object-Oriented Rough Set Model has been used to develop by finding classes and classes with name Reduct. Most existing Object-Oriented Rough Set Model suffers from intensive computation of either discernibility functions or positive regions to find classes and classes with name Reduct. In order to improve the efficiency, in this chapter, we develop a new computation model based on Relative Class Dependency.

Using the positive region defined in the (4.4) of Chapter -3, the Object-Oriented Rough Set Model of degree of dependency of \( N_{DEC} \) on \( N_{CON} \) is defined in the following way.

For \( P \in N_{CON}, Q \in N_{DEC} \), it is said that \( Q \) depends on \( P \) in a degree \( k(0 \leq k \leq 1) \), denoted \( P \rightarrow_{k} Q \), if:

\[
k = \gamma(Q) = \frac{|POS_{P}(Q)|}{|OORS|}
\]
The reduction of classes or classes with names is achieved by comparing equivalence relations generated by sets of classes and classes with names. Classes and classes with names are removed so that reduced set provides the same predictive capability of decision feature as the original. A Reduct R in the Object-Oriented Rough Set Model as classes or classes with name.

5.2 Quick Reduct Algorithm in Object-Oriented Rough Set Model

The Quick Reduct algorithm attempts to calculate a Reduct without exhaustively generating all possible Reducts. It starts off considering number conditional names and adds, in turn one at time, those classes and classes with names that result in the greatest increase in the Object-Oriented Rough Set Model dependency metric, until this produces value for the classes and classes with names set with largest positive region.

Algorithm:

QUICKREDUCT\((N_{\text{CON}}, N_{\text{DEC}})\)

\(N_{\text{CON}}\), the set of all conditional names.

\(N_{\text{DEC}}\), the set of decision names.

\(j=1\)

\(R = \{\}\)

\(R_i \leftarrow \{\text{number of condition names, and for each condition name } N_{\text{CON}}, N_{\text{CON}} \in (H_N(n))\}\)

\{\}

While \((j \leq R_i)\) {
If $\gamma_{\text{con}}(N_{\text{DEC}}) > \gamma_{\text{con},+}(N_{\text{DEC}})$

\[ l = \gamma_{\text{con}}(N_{\text{DEC}}) \]

\[ j = j + 1 \]

\}

\begin{align*}
N_{\text{DEC}} & \leftarrow \text{Another } N_{\text{DEC}} \\
T & \leftarrow R \cup I
\end{align*}

return R.

It is impossible to predict which combinations of classes and classes with names will lead to optimal Reduct in the Object-Oriented Rough Set Model based on changes in dependency with addition or deletion of single class. It does result in a close-to-minimal subset, through which is still useful in greatly reducing number of classes and classes with names.

5.3. HEURISTIC FILTER-BASED APPROACH

This algorithm begins with the considering all classes (R) of the Object-Oriented Rough Set Models. Additionally, a threshold value is required as stopping criterion to determine when Reduct in the Object-Oriented Rough Set Model is near enough to being a Reduct. The algorithm is outlined below

**Heuristic Filter-Based algorithm**

Select($N_{\text{CON}}, N_{\text{DEC}}$)
\( N_{\text{CON}} \), the set of all conditional names.

\( N_{\text{DEC}} \), the set of decision names.

\[
R \leftarrow \{ \text{Set of all classes} \{ C_i \} \in \text{OORS} \}
\]

\[
R_i \leftarrow \{ \text{number of condition names, and for each conditional name } N_{\text{CON}}, N_{\text{CON}} \in (H_n(n)) \}
\]

\( i=0, k=0; \)

\[
do \{
\text{While}(i \leq R_i) \{
\text{for}(j=1; j < R_i; j++) \{
\mid v_k \leftarrow \text{POS}_{CON_j}(N_{\text{DEC}}) \}
\quad k = k + 1
\}

i = i + 1
\}
\]

\[
N_{\text{DEC}} \leftarrow \text{Another } N_{\text{DEC}}
\]

\} until (\( N_{\text{DEC}} = \emptyset \))

Compare all \( v_k \), choose one with largest \( v_k \) and select \( N_{\text{CON}} \) of \( v_k \)

\[
R \leftarrow R \cup \{ C_i, N_{\text{CON}} \}
\]

Return R

**Example 17:** This example is in continuation of Example-15. The Parallel Reduct in the Model is as follows:
The following are the positive regions.

\[ \text{POS}_{\text{college}}(\text{department}) = 14 \]

\[ \text{POS}_{\text{college}}(\text{courses}) = 14 \]

\[ \text{POS}_{\text{department}}(\text{college}) = 14 \]

\[ \text{POS}_{\text{department}}(\text{courses}) = 14 \]

\[ \text{POS}_{\text{courses}}(\text{college}) = 14 \]

\[ \text{POS}_{\text{courses}}(\text{department}) = 14 \]

The one of Reduct of Object-Oriented Rough Set Model is

\[ R = \{ \text{Government University, Central University, Deemed University,} \]
\[ \text{Foreign University, Government University.department,} \]
\[ \text{Central University.department, Deemed University.department,} \]
\[ \text{Foreign University.versity.department} \} \]

Initially, only classes in the model are considered. The positive region of all condition names with different decision names are computed. For the one with largest positive region the corresponding condition name is selected. The selected condition name is accessed with each class and added to the \( R \). It can be taken as best Reduct in the object-Oriented Rough Set Model.

5.4. Relative Reduct

Let \( OORS(C, N, O) \) be the Object-Oriented Rough Set Model where \( C = (C, N, O) \), \( N = (N, N, N) \), \( O = (O, O, O) \) be the well defined class, name, object structures respectively. Let \( B \subseteq N_{\text{CON}} \), \( P \subseteq N_{\text{CON}} \), \( P \in H_N(n) \),
$Q \in N_{DEC}, Q \in H_N(n)$, then the following are Reative Reducts in the Model

1. $POS_B(Q) = POS_P(Q)$.

2. Minimum result of positive region represents objects with corresponding to classes and classes with names with respect to 1.

For the example under consideration, all positive regions are equal, so no Relative Reducts can be obtained

### 5.5. The Notation of Dependency

Considering positive region as stated above.

Then, it can be said that $Q$ depends functionally on $P$ in degree $k(P,Q)$, denote symbolically as $P \Rightarrow_{k(P,Q)} \mathcal{D}$, where

$$k(P,Q) \equiv \frac{|POS_P(Q)|}{|Q|}.$$  

Clearly, $Q$ depends functionally on $P$ if and only if $k(P,Q)=1$ and we say in this case that $Q$ depends on $P$; when $k(P,Q)<1$, we say that $Q$ depends partially (in degree $k(P,Q)$) on $P$.

Here it is proposed to have a relative dependency measure which is defined as follows for classes and classes with name subset $R$:

$$\kappa_R(N_{DEC}) = \left[ \frac{[x]_{N_{con}}}{[x]_{N_{dec}}} \right]$$
It can be shown that R is Reduct in the Object-Oriented Rough Set Model which represents classes and classes with names.

An algorithm is feature selection based on this measure. It performs backward elimination of features, where classes are removed from the set of considered classes if the Relative Dependency equals 1 upon their removal. Classes are considered one at time, starting with first, evaluating their Relative Class Dependency.

5.6. Relative Class Dependency

Backward Elimination based on relative class dependency.

\( N_{CON} \), the set of conditional names.

\( N_{DEC} \), the decision name.

\( N_{DEC} \in H_N(n) \)

\( R \leftarrow \{\text{Set of all classes } (C), \text{Classes with names}(C,n)\} \)

\( R_i \leftarrow N_{CON}, N_{CON} \in H_N(n) \)

If \( \kappa_R(N_{DEC}) = 1 \)

\( R_i \leftarrow R_i - \{N_{CON}\} \)

\( R \leftarrow R - R_i //\text{removing class associated with name.} \)

return \( R \).
**Example 18:** This example is in continuation of Example-11. Then the Relative class dependency is as follows. Let $R$ be the set of classes and classes with names.

$$R = \left\{ \text{Government University, CentralUniversity, DeemedUniversity,} \right.$$  
$$\text{ForeignUniversity, Government University.department,}$$  
$$\text{CentralUniversity.department, DeemedUniversity.department,}$$  
$$\text{ForeignUniversity.department, GovernmentUniversity.courses,}$$  
$$\text{CentralUniversity.courses, DeemedUniversity.courses,}$$  
$$\text{ForeignUniversity.courses........} \right\}$$

\[
\kappa_R(N_{DEC}) = \left[ \frac{[x]_{N_{DEC}}}{[x]_{N_{DEC}}} \right] \\
\kappa_R(N_{college}) = \left[ \frac{[x]_{N_{department}}}{[x]_{N_{college}}} \right] = 14/14 = 1 \\
\kappa_R(N_{college}) = \left[ \frac{[x]_{N_{course}}}{[x]_{N_{college}}} \right] = 14/14 = 1 \\
\kappa_R(N_{department}) = \left[ \frac{[x]_{N_{course}}}{[x]_{N_{department}}} \right] = 14/14 = 1 \\
\kappa_R(N_{department}) = \left[ \frac{[x]_{N_{college}}}{[x]_{N_{department}}} \right] = 14/14 = 1 \\
\kappa_R(N_{courses}) = \left[ \frac{[x]_{N_{department}}}{[x]_{N_{courses}}} \right] = 14/14 = 1 \\
\kappa_R(N_{courses}) = \left[ \frac{[x]_{N_{college}}}{[x]_{N_{courses}}} \right] = 14/14 = 1
So removing college, courses, department, the following is Reduct of Object-Oriented Rough Set Model

\[ R = \{ \text{GovernmentUniversity, CentralUniversity, DeemedUniversity, ForeignUniversity} \} \]

We note that the algorithm is an arbitrary class with name that may be removed with respect to decision name. The output of the algorithm represents remaining classes and classes with names.

### 5.7. Conclusions

This chapter introduced a new Reduct called Relative Reduct by treating objects as instances of classes, and the structural hierarchies among them on, is-a relationship and has-a relationship. For determining the Reducts a Heuristic-Filter Base Algorithm is proposed. As the Reduct finding suffer intense computation of discernibility Matrix and positive regions, to improve the efficiency, a new computation model based on Relative Class Dependency is introduced to address the issues.