Research Article
Combined Accelerator for Attribute Reduction: A Sample Perspective

Yan Chen, Jingjing Song, Keyu Liu, Yaojin Lin, and Xibei Yang

1School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China
2Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian 363000, China
3School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China

Correspondence should be addressed to Jingjing Song: songjingjing108@163.com

Received 21 November 2019; Revised 12 January 2020; Accepted 21 January 2020; Published 19 February 2020

Academic Editor: Giuseppe D’Aniello

Copyright © 2020 Yan Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the field of neighborhood rough set, attribute reduction is considered as a key topic. Neighborhood relation and rough approximation play crucial roles in the process of obtaining the reduct. Presently, many strategies have been proposed to accelerate such process from the viewpoint of samples. However, these methods speed up the process of obtaining the reduct only from binary relation or rough approximation, and then the obtained results in time consumption may not be fully improved. To fill such a gap, a combined acceleration strategy based on compressing the scanning space of both neighborhood and lower approximation is proposed, which aims to further reduce the time consumption of obtaining the reduct. In addition, 15 UCI data sets have been selected, and the experimental results show us the following: (1) our proposed approach significantly reduces the elapsed time of obtaining the reduct; (2) compared with previous approaches, our combined acceleration strategy will not change the result of the reduct. This research suggests a new trend of attribute reduction using the multiple views.

1. Introduction

With the rapid development of technology, a large number of high-dimensional data come into being such as video, image, and text. The high dimensionality of data [1–5] may contribute to a great challenge for decision-making and data analysis. Attribute reduction [6–12], as an efficient tool for reducing the dimensionality of data [13–15], has been widely concerned by many researchers.

In the topic of attribute reduction, there are two important notions for the process of obtaining the reduct. One fundamental notion is binary relation [16], and the other is rough approximation.

Firstly, binary relation is effective to conduct information granulation [17–21]. In the classical rough set [22–25], binary relation is an equivalence relation, and each single equivalence class can be regarded as an information granule [26]. Nevertheless, the equivalence relation can only deal with categorical data, and this somewhat limits the applications of the rough set. Therefore, several generalized binary relations and the corresponding rough sets have been proposed [27–30, 31]. Among those generalized rough approximations, the neighborhood rough set offers us a useful mechanism to process continuous data. This is mainly because the neighborhood relation can be directly obtained via computing the distance between samples. From the viewpoint of Granular Computing (GrC), the neighborhood rough set groups the samples into different neighborhood granules [16].

To obtain the neighborhood granule for each sample, all the other samples in the data should be scanned by using traditional approaches [26, 32, 33]. With the increasing of the size of data, the elapsed time of computing neighborhood granules will be unacceptable. To save the time consumption of computing neighborhood granules, Liu et al. [34] have introduced the hash function into the calculating of neighborhood. Such hash function can map all the samples into a series of buckets. Therefore, only samples in the adjacent buckets should be considered to obtain a neighborhood granule. It should be emphasized that such a
method provides a novel approach for reducing the time consumption of obtaining the neighborhood granule from the perspective of the sample.

Secondly, rough approximation is a general concept represented by a pair of sets, which is always characterized via the so-called upper and lower approximations. Following the detailed description of the mechanism of bucket, it is not difficult to observe that the bucket only reduces the scanning space of obtaining the neighborhood granule from the viewpoint of the sample. However, the mechanism of bucket takes all the samples into consideration when computing the lower approximation. If the scale of data is becoming larger, then the elapsed time of deriving lower approximation will be significantly increased. For such a reason, Qian et al. [35] have proposed the local approximation which aims to remove the useless information in the process of obtaining the lower approximation. Such a strategy only uses the samples in the target concept instead of the samples in the whole universe for deriving the lower approximation. Therefore, such a method fully meets the requirements of big data analysis.

However, no matter which strategy is employed, it can only accelerate the process of obtaining the reduct from one and only one point of view. Motivated by further reducing the elapsed time of obtaining the reduct, a new approach which combines the above two crucial notions will be proposed from the perspective of the sample. Firstly, the mechanism of bucket is used, which aims to reduce the scanning space of deriving neighborhood granules. Secondly, by using the local approximation, the useless information granules will be removed for quickly computing the lower approximation. Finally, the appropriate attributes will be selected until the intended constraint defined in the attribute reduction is satisfied. Obviously, the main contribution of our approach is to take both advantages of bucket and local approximation, and our proposed strategy will be detailed in Section 4.

2. Preliminaries

2.1. Binary Relation. Without loss of generality, a decision system can be represented as DS = \langle U, AT \cup \delta \rangle in which U is the set of samples called the universe, AT is the set of conditional attributes, and \delta is the decision attribute.

Furthermore, \forall x_i \in U, a(x_i) indicates its value over the conditional attribute \( a \in AT \), and \( d(x_i) \) denotes its value of decision attribute.

With regard to a given decision system DS, to derive the result of information granulation over \( U \), various forms of binary relations have been explored in terms of different requirements. Two typical forms will be analyzed as follows:

(i) One is the equivalence relation. If information granulation is required over the categorical data, then the binary relation can be given by the equivalence relation. For example, in supervised learning task, the values over decision attribute in a decision system are categorical. To derive the result of information granulation over the decision attribute, the equivalence relation \( IND_d = \{(x_i, x_j) \in U \times U | d(x_i) = d(x_j)\} \) can be employed. Then, the result of information granulation (i.e., partition) \( (U/IND_d) = \{X_1, X_2, \ldots, X_s\} \) can be derived, and the value of \( s \) is the number of equivalence classes. Specifically, each equivalence class in such partition is referred to an information granule.

(ii) The other is the parameterized binary relation. If information granulation is required over the continuous data, then the binary relation can be given by parameterized binary relation. Take the neighborhood relation as an example, and the values over conditional attribute in the decision system are continuous; to derive the results of information granulation over the conditional attribute, the neighborhood relation \( \delta_A = \{(x_i, x_j) \in U \times U | d_A (x_i, x_j) \leq \delta\} \) can be used, in which \( d_A (x_i, x_j) \) is the distance between samples \( x_i \) and \( x_j \) over \( A \), and \( \delta \) is the radius. It follows that neighborhood system [36] can be derived. Specifically, each neighborhood obtained by \( \delta_A (x_i) = \{x_j \in U | (x_i, x_j) \in \delta_A\} \) is referred to as an information granule.

2.2. Neighborhood Rough Set

**Definition 1.** Given a decision system DS, \( \forall A \subseteq AT \) and \( \forall X_p \in (U/\text{IND}_d) \) (\( 1 \leq p \leq s \)), the neighborhood lower and upper approximations of \( X_p \) with respect to \( A \) are defined as

\[
N_A(X_p) = \{x_i \in U | \delta_A (x_i) \subseteq X_p \}; \quad (1)
\]

\[
\overline{N}_A(X_p) = \{x_i \in U | \delta_A (x_i) \cap X_p \neq \emptyset \}. \quad (2)
\]

The pair \([N_A(X_p), \overline{N}_A(X_p)]\) is referred to as a neighborhood rough set of \( X_p \) in terms of \( A \).

**Definition 2.** Given a decision system DS, the approximation quality of \( d \) over \( A \subseteq AT \) is defined as

\[
\gamma(A, d) = \frac{|U \setminus U_{d=1}N_A(X_p)|}{|U|}, \quad (3)
\]

in which \(|X|\) denotes the cardinality of set \( X \).
From the viewpoint of rough set theory, approximation quality reflects the percentage of the samples which belongs to one of the decision classes by the explanation of neighborhood lower approximation. The greater the value of approximation quality, the greater the degree of such belongingness.

2.3. Attribute Reduction. Attribute reduction [32, 37–41] aims to find an attribute subset without reducing some considered properties which are related to the raw attributes. Take the approximation quality-based attribute reduction as an example; such an attribute reduction is to obtain a minimal attribute subset that will not decrease the degree of the belongingness shown in Definition 2. The definition of the approximation quality reduce can be presented as follows.

**Definition 3.** Given a decision system DS, ∀A ⊆ AT, A is referred to as an approximation quality-based reduct if and only if

1. γ(A, d) ≥ γ(AT, d)
2. ∀A′ ⊂ A, γ(A′, d) < γ(A, d)

Following the detailed description, generating the approximation quality reduct should satisfy two conditions. The first condition implies that the value of approximation quality over AT should not be reduced, and the second condition indicates that each attribute in A is individually necessary.

To find the approximation quality reduct, a lot of researching results have been put forward using various searching strategies [7, 8, 42, 43]. The forward greedy searching strategy has been widely accepted for its lower time complexity. Notably, the significance function is a necessary factor in the framework of forward greedy searching. This is mainly because each candidate attribute should be evaluated by the significance function. The definition of significance function is shown in the following.

**Definition 4.** Given a decision system DS, ∀A ⊆ AT, the significance of the candidate attribute a with respect to A is defined as

\[
\text{Sig}(a, A, d) = \gamma(A \cup \{a\}, d) - \gamma(A, d)
\]

\[
= \left| \frac{\bigcup_{P = 1}^{r} N_{A \cup \{a\}}(X_P)}{|U|} \right| - \left| \frac{\bigcup_{P = 1}^{r} N_{A}(X_P)}{|U|} \right|
\]

(4)

Following Definition 4, the detailed process of obtaining the reduct by using the forward greedy searching strategy can be shown in the following.

In Algorithm 1, the time complexity of the computing neighborhood relation is \(O(|U|^2)\). This is mainly because the distance between any two samples should be computed when computing the neighborhoods of all the samples. In the worst case, no attribute is redundant, and then all the attributes in the decision system should be added into the reduct. Therefore, the time complexity of Algorithm 1 is \(O(|U|^2 \cdot |AT|^2)\).

3. Bucket and Local Approximation-Based Attribute Reductions

3.1. Bucket-Based Attribute Reduction. As what has been indicated in Section 2.1, the neighborhood of one sample can be represented as the granule. In Algorithm 1, to obtain the granule, it is necessary to scan all the samples in the decision system. However, if the number of samples, i.e., the value of \(|U|\) increases to a large scale, then the time consumption of obtaining the granule can be unacceptable. To alleviate the phenomenon of excessive time consumption, Liu et al. [34] have proposed the mechanism of bucket which employs a hash function to map all the samples into a series of sequenced buckets, and then only samples in the adjacent buckets should be considered instead of the whole samples in the universe. The definition of the bucket is shown in the following.

**Definition 5** (see [34]). Given a decision system DS, \(x_0\) is a special sample constructed from \(U\), where \(\forall a \in AT\), \(a(x_0) = \min\{a(x); \forall x \in U\}\). Then, the samples in \(U\) can be divided into finite buckets \(B_0, \ldots, B_t\). \(B_t\) can be denoted as

\[
B_t = \{x_i \in U \mid \left| \frac{\text{dis}(x_0, x_i)}{\delta} \right| = t\}.
\]

(5)

If \(x_0 \in U\), then \(B_0 = \{x_0\}\); otherwise, \(B_0 = \emptyset\).

**Theorem 1** (see [34]). Given a decision system DS, \(B_0, \ldots, B_t\) are the buckets, then \(\forall x_i \in B_q\) \((q = 1, 2, \ldots, t - 1)\), and the neighborhood of \(x_i\) is only contained in \(B_{q-1} \cup B_q \cup B_{q+1}\); if \(x_i \in B_0\) then the neighborhood of \(x_i\) is only contained in \(B_0 \cup B_1\); if \(x_i \in B_t\), then the neighborhood of \(x_i\) is only contained in \(B_{t-1} \cup B_t\).

Notably, only samples in the adjacent buckets should be taken into consideration for obtaining the information granule through observing Theorem 1. Therefore, the scanning space of obtaining the information granule can be reduced. It follows that the elapsed time of calculating the information granule can be decreased. Following Definition 5 and Theorem 1, forward greedy searching based on the bucket is presented as follows.

Different from Algorithm 1, the number of samples which should be scanned is \(|U'| = |B_{q-1}| + |B_q| + |B_{q+1}|\) in Algorithm 2. Obviously, \(|U'| \geq |U|\). It follows that the mechanism of bucket can reduce the number of times to scan samples for obtaining an information granule. Therefore, the efficiency of obtaining the reduct can be improved by using Algorithm 2.

3.2. Local Approximation-Based Attribute Reduction. However, the mechanism of bucket shown in Section 3.1 only considers how to reduce the time consumption of obtaining the neighborhood of each sample. In the process of obtaining the reduct by considering the measure of approximation quality, the elapsed time of deriving the lower approximation is also time consuming. Following such
consideration, Qian et al. [35] have proposed a local approach which can reduce the time consumption of computing lower approximation.

Through observing the process of Algorithms 1 and 2, equation (1) is adapted to compute the lower approximation, and then the value of approximation quality can be derived from the obtained lower approximation. Through observing equation (1), it is not difficult to observe that calculating the lower approximation should use all the information granules obtained by scanning all the samples, which is exceedingly time consuming. In fact, only the set of information granules \( \{ \delta_A(x_i) \mid \delta_A(x_i) \cap X_p \neq \emptyset, x_i \in U \} \) is useful, while the set of information granules \( \{ \delta_A(x_i) \mid \delta_A(x_i) \cap X_p = \emptyset, x_i \in U \} \) is useless. Therefore, the local rough approximation is defined as follows.

**Definition 6.** Given a decision system DS, \( \forall A \subseteq AT \) and \( \forall X_p \in (U/\text{IND}_d) \), \( \text{IND}_{d} \), the neighborhood lower and upper approximations of \( X_p \) with respect to \( A \) are defined as
the detailed algorithm is shown as follows.

The time consumption of obtaining the reduct can also be reduced, and then the time consumption of obtaining the reduct can also be reduced. The detailed algorithm is shown as follows.

In Algorithm 3, to compute the lower approximation of \(X_p\), only the samples in \(X_p\) instead of the whole samples in \(U\) should be used. Therefore, the time complexity of obtaining the lower approximation is \(O(|U| \cdot |X_p|)\). However, in Algorithm 1, the time complexity of obtaining the lower approximation is \(O(|U|^2)\). Therefore, by using the local approximation approach, the time of obtaining the reduct can be decreased.

\[
\bigcap_A (X_p) = \left\{ x_i \in X_p \mid \delta_A(x_i) \subseteq X_p \right\};
\]

\[
\overline{\bigcap_A} (X_p) = \left\{ x_i \in U \mid \delta_A(x_i) \cap X_p \neq \emptyset \right\}.
\]

By using equation (6), the elapsed time of obtaining the lower approximation can be decreased, and then the time consumption of obtaining the reduct can also be reduced. The detailed algorithm is shown as follows.

In Algorithm 3, to compute the lower approximation of \(X_p\), only the samples in \(X_p\) should be considered when obtaining an information granule. Then, the elapsed time of obtaining an information granule is decreased, and it follows that the time of obtaining the reduct can be saved.

(i) For the mechanism of bucket, it employs a hash function to map all the samples into a series of sequenced buckets, and only samples in the adjacent buckets should be considered when obtaining an information granule. Then, the elapsed time of obtaining an information granule is decreased, and it follows that the time of obtaining the reduct can be saved.

(ii) For the approach of local approximation, it can reduce the scope of obtaining a lower approximation, and then useless information can be removed. The elapsed time of computing the lower approximation is decreased, and it follows that the time of obtaining the reduct can also be saved.

Nevertheless, no matter which strategy is employed, it can only improve the time efficiency of obtaining the reduct from one and only one point of view. On the basis of this, we will propose a strategy which combines the above two approaches for further reducing the elapsed time of obtaining the reduct. Firstly, the mechanism of bucket is employed to obtain the information granule, which can reduce the scanning space of samples. Secondly, the useless information granules can be removed for quickly obtaining the lower approximation. Finally, the appropriate attributes can be selected by the measure of approximation quality until the constraint defined in the attribute reduction is satisfied.

In Algorithm 4, the time complexity of obtaining the lower approximation is \(O(|U'| \cdot |X_{p'}|)\). However, in Algorithm 2, the time complexity of obtaining the lower approximation is \(O(|U'| \cdot |U|)\). In Algorithm 3, the time complexity of obtaining the lower approximation is \(O(|U| \cdot |X_p|)\). Therefore, the time consumption of obtaining the reduct can be further decreased by using our Algorithm 4.

4. Experiments

To demonstrate the effectiveness of our proposed method in this paper, 15 UCI data sets have been selected to conduct experiments. The basic descriptions of data sets are shown in Table 1. All the experiments have been carried out on a personal computer with Windows 10, dual-core 2.60 GHz CPU, 8 GB memory. The programming language is MATLAB R2017b.

In our experiments, 5-fold cross-validation has been adopted to test the performances of different reducts. It follows that data have been divided into 5 parts of the same size. In each round of calculation, 80% of the samples have been regarded as the training data set for deriving the reduct, and the rest have been denoted as testing data for classification. Furthermore, 20 radii, i.e., 0.01, 0.02, . . . , 0.2 have been selected, which are recommended in [34].

Note that, in the following subsections, “AQR,” “BAQR,” “LAQR,” and “BLAQR” denote reducts which are derived by Algorithms 1–4, respectively.

4.1. Comparisons of Elapsed Time. In this experiment, the elapsed time of obtaining the reduct by using four different algorithms will be compared. Moreover, the standard deviation is also lower than the ones produced by using four different algorithms. It follows that the time of computing the reduct is possibly to be increased if the value of \(\delta\) increases.

(1) With the increasing value of \(\delta\), the increasing trends can be observed for the elapsed time of computing reduct by using four algorithms in most cases. This is mainly because the number of attributes in the reduct is possibly to be increased if the value of \(\delta\) increases.

(2) Compared with Algorithm 1, both Algorithm 2 and Algorithm 3 emerge lower time consumption. Moreover, Algorithm 3 requires lower time consumption than Algorithm 2 does. Take “Cardiotocography” as an example, if \(\delta\) is 0.1, then the elapsed time of obtaining reducts by using three different algorithms is 15.7143, 14.3278, and 14.0373 seconds, respectively.

(3) Compared with the process of computing BAQR and LAQR, the process of searching BLAQR requires lower elapsed time. Take “Diabetic Retinopathy Debrecen” as an example; if \(\delta\) is 0.15, then the elapsed time of obtaining BLAQR is 2.9222 seconds. The elapsed time of obtaining BAQR and LAQR is 3.9703 and 3.1344 seconds, respectively. Such results indicate that our proposed strategy has speeded up the process of finding the reduct significantly.

(4) Through observing Tables 2 and 3, Algorithm 4 comes with a lower mean time. Furthermore, standard deviation is also lower than the ones produced.
**Algorithm 3:** Local approximation-based forward greedy searching strategy for obtaining the reduct.

**Inputs:** Decision system DS = \( \langle U, AT \cup d \rangle \), radius \( \delta \).

**Outputs:** An approximation quality reduct \( A \).

1. Compute \( (U/IND_d) = [X_1, X_2, \ldots, X_s] \);
2. Compute \( \gamma(AT, d) \);
3. \( A \leftarrow \emptyset \);
4. Do
   (i) \( \forall X_p \in (U/IND_d) \), \( \forall x_i \in X_p \), compute \( \delta_A(x_i) \);
   (ii) Compute the lower approximation based on \( \delta_A(x_i) \) by using equation (7);
   (iii) \( \forall a \in AT - A \), compute \( \text{Sig}(a, A, d) \) based on the lower approximation, and then select \( b \) such that \( \text{Sig}(b, A, d) = \max\{\text{Sig}(a, A, d) : \forall a \in AT - A\} \);
   (iv) \( A = A \cup \{b\} \);
   (v) Compute \( \gamma(A, d) \);
   While \( \gamma(A, d) \leq \gamma(AT, d) \);
5. Do
   (i) \( \forall c \in A \), compute \( \gamma(A - \{c\}, d) \);
   (ii) If \( \gamma(A - \{c\}, d) \geq \gamma(AT, d) \)

\[ A = A - \{c\} \]

End

While \( A \) changes and \( |A| > 1 \);

6. Return \( A \).

---

**Algorithm 4:** Bucket and local approximation for obtaining the reduct.

**Inputs:** Decision system DS = \( \langle U, AT \cup d \rangle \), radius \( \delta \).

**Outputs:** An approximation quality reduct \( A \).

1. Compute \( (U/IND_d) = [X_1, X_2, \ldots, X_s] \);
2. Compute \( \gamma(AT, d) \);
3. \( A \leftarrow \emptyset \);
4. \( \forall X_p \in (U/IND_d) \), \( \forall x_i \in X_p \), Hash \( (x_i, B_k) \);
5. Do
   (i) For each \( x_i \in X_p \)
     For each \( x_j \in B_{k-1} \cup B_k \cup B_{k+1} \)
     If \( \text{dis}(x_i, x_j) \leq \delta \)
     \( x_j \) is the neighbor of \( x_i \);
   End
   End
   Compute \( \delta_A(x_i) \);
   End
   (ii) Compute the lower approximation based on \( \delta_A(x_i) \) by using equation (7);
   (iii) \( \forall a \in AT - A \), compute \( \text{Sig}(a, A, d) \) based on the lower approximation, and then select \( b \) such that \( \text{Sig}(b, A, d) = \max\{\text{Sig}(a, A, d) : \forall a \in AT - A\} \);
   (iv) \( A = A \cup \{b\} \);
   (v) Compute \( \gamma(A, d) \);
   While \( \gamma(A, d) \leq \gamma(AT, d) \);
6. Do
   (i) \( \forall c \in A \), compute \( \gamma(A - \{c\}, d) \);
   (ii) If \( \gamma(A - \{c\}, d) \geq \gamma(AT, d) \)

\[ A = A - \{c\} \]

End

While \( A \) changes and \( |A| > 1 \);

7. Return \( A \).
Table 1: Descriptions of data sets.

<table>
<thead>
<tr>
<th>ID</th>
<th>Data sets</th>
<th>Samples</th>
<th>Attributes</th>
<th>Decision classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cardiotocography</td>
<td>2126</td>
<td>21</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Contraceptive Method Choice</td>
<td>1473</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Crowdsourced Mapping</td>
<td>10845</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Diabetic Retinopathy Debrecen</td>
<td>1151</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>German</td>
<td>1000</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Letter Recognition</td>
<td>20000</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>Magic Gamma Telescope</td>
<td>19020</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Page Blocks Classification</td>
<td>5473</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Magic Gamma Telescope</td>
<td>19020</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Statlog (image segmentation)</td>
<td>2310</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Statlog (Landsat satellite)</td>
<td>6435</td>
<td>36</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>Wall-Following Robot Navigation</td>
<td>5456</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>Waveform Database Generator (version 1)</td>
<td>5000</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>Waveform Database Generator (version 2)</td>
<td>5000</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

![Figure 1: Continued.](image-url)
by AQR, BAQR, and LAQR for obtaining reducts in most cases.

4.2. Comparisons of Stability-Based Reducts. In this section, the stabilities of reducts will be compared using four different algorithms. For more details about computation of the stability, see [20]. Moreover, the standard deviation [44] is used to characterize the dispersion degree of stability. The lower the value of the standard deviation, the higher the stability of reducts.
### Table 3: The elapsed time ($\delta = 0.2$).

<table>
<thead>
<tr>
<th>ID</th>
<th>Algorithm 1 (seconds)</th>
<th>Algorithm 2 (seconds)</th>
<th>Algorithm 3 (seconds)</th>
<th>Algorithm 4 (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.8818 ± 2.6218</td>
<td>15.1942 ± 2.4092</td>
<td>13.9933 ± 2.1455</td>
<td>13.3126 ± 2.0835</td>
</tr>
<tr>
<td>2</td>
<td>1.7975 ± 0.0730</td>
<td>1.6874 ± 0.0721</td>
<td>1.4172 ± 0.0507</td>
<td>1.3127 ± 0.0642</td>
</tr>
<tr>
<td>3</td>
<td>712.6823 ± 154.1671</td>
<td>658.0492 ± 140.5334</td>
<td>405.5043 ± 79.2855</td>
<td>350.7878 ± 65.0520</td>
</tr>
<tr>
<td>4</td>
<td>4.2553 ± 0.6203</td>
<td>4.1227 ± 0.5770</td>
<td>3.2243 ± 0.4658</td>
<td>3.0570 ± 0.4148</td>
</tr>
<tr>
<td>5</td>
<td>3.7099 ± 0.3600</td>
<td>3.5396 ± 0.3346</td>
<td>2.6141 ± 0.2216</td>
<td>2.4199 ± 0.1884</td>
</tr>
<tr>
<td>6</td>
<td>768.9334 ± 49.6549</td>
<td>742.6549 ± 32.5468</td>
<td>530.3251 ± 60.3075</td>
<td>502.3758 ± 41.4835</td>
</tr>
<tr>
<td>7</td>
<td>277.8121 ± 33.1599</td>
<td>271.3526 ± 24.4083</td>
<td>166.8020 ± 36.9960</td>
<td>160.3587 ± 26.4904</td>
</tr>
<tr>
<td>8</td>
<td>30.0322 ± 1.7625</td>
<td>29.3092 ± 1.3176</td>
<td>26.8057 ± 1.0691</td>
<td>26.1383 ± 0.9010</td>
</tr>
<tr>
<td>9</td>
<td>8.0012 ± 1.9094</td>
<td>7.7373 ± 1.8071</td>
<td>6.0683 ± 1.4523</td>
<td>5.8254 ± 1.3521</td>
</tr>
<tr>
<td>11</td>
<td>15.3631 ± 2.8326</td>
<td>14.5911 ± 2.6272</td>
<td>12.7235 ± 2.3736</td>
<td>11.9391 ± 2.1424</td>
</tr>
<tr>
<td>12</td>
<td>480.8548 ± 117.1881</td>
<td>437.4929 ± 105.1775</td>
<td>387.4126 ± 93.4885</td>
<td>343.9659 ± 81.3193</td>
</tr>
<tr>
<td>13</td>
<td>149.2419 ± 6.4690</td>
<td>138.7982 ± 4.6385</td>
<td>111.6941 ± 6.4658</td>
<td>101.2385 ± 4.1834</td>
</tr>
<tr>
<td>14</td>
<td>91.9483 ± 13.2181</td>
<td>90.3759 ± 13.9434</td>
<td>68.5839 ± 9.1511</td>
<td>66.9474 ± 9.8640</td>
</tr>
<tr>
<td>15</td>
<td>188.2401 ± 28.2228</td>
<td>184.2565 ± 29.4879</td>
<td>141.8513 ± 19.8201</td>
<td>137.8270 ± 21.1095</td>
</tr>
</tbody>
</table>

**Figure 2:** Continued.
Following Figure 2, it is not difficult to observe the following:

(1) The stabilities of reducts by using four algorithms are the same. That is to say, the derived reducts by using four different algorithms are the same. This is mainly because the mechanism of bucket only reduces the scanning times of samples, and the obtained neighborhood relation does not change. Furthermore, the local approach only removes the useless granule, and the obtained lower approximation also does not change. Therefore, our proposed strategy which combines the mechanism of bucket and local approximation does not change the result of the reduct.

(2) The standard deviations of reducts are also the same. It follows that the dispersion degrees of stabilities for obtaining the reduct by using four algorithms are the same.

Because the stability-based reducts by using four algorithms are the same, the classification accuracies derived from AQR, BAQR, LAQR, and BLAQR are also the same. Therefore, the results of classification will not be shown in this paper.

4.3. Comparisons of Elapsed Time over Incremental Data.

In this section, the time tendency with the incremental scale of data will be shown. The whole universe has been divided into 11 groups. Firstly, only the first group will be used to obtain the reduct and the corresponding elapsed time will be recorded; secondly, the second group will be added into the first group for computing reduct and the corresponding elapsed time will be recorded, and so on; and finally, all the groups will be combined for computing reduct and the corresponding elapsed time will be recorded.

Following Figure 3, it is not difficult to observe the followings:

(1) With the increasing of the volume of data, the elapsed time of obtaining the reduct by using four different algorithms is increasing. This is mainly because when the scale of the sample increases, the elapsed time of calculating the neighborhood relation and lower approximation increases. Therefore, the time consumption of obtaining the reduct will be increased.

(2) Though the data have been divided into a series of groups, it is also not difficult to observe that the elapsed time of obtaining the reduct by using our approach is less than that of other methods.
Figure 3: Continued.
Furthermore, with the size of data becoming large, our approach has obvious advantages in time consumption compared with other algorithms.

5. Conclusions and Future Perspectives

In this paper, we have designed an accelerator for calculating the reduct using the concepts of both bucket and local approximation. Our proposed strategy not only reduces the scanning space of obtaining information granules but also removes the useless granules for computing the lower approximations. Furthermore, the experimental results show that our proposed approach can significantly reduce the time consumption of obtaining the reduct. In addition, compared with the previous research studies, the stability-based reducts are the same which implies that the obtained reducts are unchanged by our strategy.

The following topics deserve our further investigations:

1. The proposed strategy is only realized from the viewpoint of sample; some other acceleration strategies which take the attribute into account can be further studied.

2. Single granularity is applied to our proposed strategy; multiple granularities [16] can be further considered in the accelerator.

Data Availability

The (UCI) data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (nos. 61906078 and 61572242) and the Key Laboratory of Data Science and Intelligence Application, Fujian Province University (no. D1901).

References


[10] K. Qin, B. Li, and Z. Pei, “Attribute reduction and rule acquisition of formal decision context based on object proportions,”