Local neighborhood rough set

Qi Wang[a,b,c], Yuhua Qian [*a,b,c], Xinyan Liang[a,b,c], Qian Guo[a,b,c], Jiye Liang[b]

[a] *Institute of Big Data Science and Industry, Shanxi University, Taiyuan, 030006 Shanxi, China*
[b] *Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, 030006 Shanxi, China*
[c] *School of Computer and Information Technology, Shanxi University, Taiyuan,030006 Shanxi, China*

Abstract

With the advent of the age of big data, a typical big data set called limited labeled big data appears. It includes a small amount of labeled data and a large amount of unlabeled data. Some existing neighborhood-based rough set algorithms work well in analyzing the rough data with numerical features. But, they face three challenges: limited labeled property of big data, computational inefficiency and over-fitting in attribute reduction when dealing with limited labeled data. In order to address the three issues, a combination of neighborhood rough set and local rough set called local neighborhood rough set (LNRS) is proposed in this paper. The corresponding concept approximation and attribute reduction algorithms designed with linear time complexity can efficiently and effectively deal with limited labeled big data. The experimental results show that the proposed local neighborhood rough set and corresponding algorithms significantly outperform its original counterpart in classical neighborhood rough set. These results will enrich the local rough set theory and enlarge its application scopes.

Keywords: Rough set, local neighborhood rough set, concept approximation, attribute reduction, limited labeled data
2010 MSC: 00-01, 99-00

1. Introduction

Rough set theory was introduced by Pawlak [1, 2, 3] as a powerful soft computing tool for modelling and processing uncertainty information. It has been applied to feature selection [4, 5, 6, 7, 8], pattern recognition [9, 10], uncertainty reasoning [11], granular computing [12, 13, 14, 15], data mining and knowledge discovery [16, 17, 18, 19, 20, 21]. Over the past decades, it has an enormous impact on the uncertainty management and uncertainty reasoning.

There are two significant notions for rough set. One fundamental notion is concept approximation, in which a general concept represented by a set is always characterized via the so-called upper and lower approximations. Given a data set U and a binary relation R including equivalence relation, tolerance relation, neighborhood relation, dominance relation, and so on, and this given binary relation partitions a data set into a family of concepts, also called a granular structure U/R in granular computing, and each of which is called an information granule used to approximate a target concept [22, 23, 24]. One can get a rough set of any subset on the data set via employing information granule from U/R. The other important notion is attribute reduction which can be considered as a kind of specific feature selection [25, 26, 27, 28], whose objective is to reduce the number of attributes and to preserve a certain property that we want at the same time. In rough set theory, we are interested in the property of retaining the distinguishing ability provided by the originally whole attribute set [29, 30], rather than try to maximize the classification power [31, 32, 33, 34]. In other words, based on rough set theory, one can omit irrelevant and redundant attributes that will not influence the discriminability to to current recognition tasks [35, 29, 36, 37] and select useful features from a given data set. Given

[*]Corresponding author.
Email addresses: counter_king@163.com (Qi Wang), jinchengqyh@126.com (Yuhua Qian), liangxinyan48@163.com (Xinyan Liang), zcguoqian@163.com (Qian Guo), ljy@sxu.edu.cn (Jiye Liang)

Table 1: A data table with limited labeled objects

Objects	x_1	x_2	$\cdots$	x_p	$\cdots$	x_{n-1}	x_n
a_1	$a_1(x_1)$	$a_1(x_2)$	$\cdots$	$a_1(x_p)$	$\cdots$	$a_1(x_{n-1})$	$a_1(x_n)$
a_2	$a_2(x_1)$	$a_2(x_2)$	$\cdots$	$a_2(x_p)$	$\cdots$	$a_2(x_{n-1})$	$a_2(x_n)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_k	$a_k(x_1)$	$a_k(x_2)$	$\cdots$	$a_k(x_p)$	$\cdots$	$a_k(x_{n-1})$	$a_k(x_n)$
Class labels	d_1	d_2	$\cdots$	d_r			

a set of objects with class labels, some decision rules, which is called a rough classifier, can be obtained by utilizing attribute reduction induced by rough set model. We can predict the class label of an unseen object through using this set of decision rules. Considering this point, classical rough model can be thought as a supervised learning method.

Rough set theory is originally constructed on the basis of an equivalence relation. However, it is limited in many real-world applications. To overcome this limitation, ones extend the equivalence relation to other binary relations, such as similarity relation, tolerance relation, dominance relation and neighborhood relation, to generalize the classical rough sets. Among them, neighborhood rough sets are very important extension to deal with numeric data.

For convenience, we combine neighborhood rough set [38] with the decision-theoretic rough sets [39, 40] into the same rough set model, as a representative, called global neighborhood rough set in this paper. Let (U, N) be a neighborhood approximation space with N being neighborhood relation on U. The lower and upper approximations of the set X are defined as follows.

$$\begin{cases} \underline{N}_{\alpha}(X) = \{x | \mathcal{P}(X|\delta(x)) \geq \alpha, x \epsilon U\}, \\ \overline{N}_{\beta}(X) = \{x | \mathcal{P}(X|\delta(x)) > \beta, x \epsilon U\}. \end{cases} \tag{1}$$

where $\mathcal{P}(.)$ is a conditional function, $\delta(x)$ is neighborhood of x and α, β are two parameters from the decision-theoretic rough set:

The existing rough set models have made great achievement in rough data analysis, but they encounter some challenges when handling large-scale data sets. In what follows, we present a detailed description.

(a) Semi-supervised property of big data

Many state-of-the-art algorithms focus on classifiers or regressors from a given training set, where every object must be labeled. With the development of the age of the big data, one can get more data objects than ever. Some methods [41, 42, 43, 44] have been proposed to deal with stream data, such as data obtained from all kinds of sensors and that from social media, which increase dynamically. However, these models generally use labeled objects, and these unlabelled objects are not used to construct concept approximation for rough set-based supervised learning, where these algorithms require a large number of labeled data, and labelling these data is expensive and laborious. On the contrary, with the advent of Internet, obtaining unlabeled data becomes easy and cheap. Under the environment of big data, a data set to deal with could be represented as a data table shown in Table 1 (we can call it limited labeled decision table). In the original rough set model, only the data set $\{x_1, x_2, \cdots, x_p\}$ is used, which means that the model cannot use other information provided by unlabeled data. So a semi-supervised learning strategy is necessary, in which it can automatically learn rough classifiers from big data with limited labeled data. This is one motivation of rough data analysis in big data.

(b) Computational inefficiency

From the Equation (1), we can know that, calculating its lower/upper approximation needs to use all information granules obtained by scanning all objects, which is exceedingly time-costing. And its time complexity is $O(n^2)$ without pre-ranking and $O(n \log n)$ with pre-ranking [45, 46]. For a large-scale data set, they cannot effectively and efficiently work to satisfy the requirement in real world. How to reduce the time consumption is the second motivation of this study.

(c) Over-fitting in attribute reduction

The over-fitting degree in attribute reduction can be observed by the monotonicity of positive regions of a target decision, which is often measured by the accuracy of approximation in Eq. (9). It is a truth modeling classifier task which is influenced by noise easily [47]. So, we should consider robustness and sensitivity of attribute reduction to noise samples. If the measures used to evaluate significance of attribute in attribute reduction are robust to noisy objects, the performance of the trained classifier would be better. Some existing extended rough set model, such as variable precision rough set [48], decision rough set [39, 40], bayesian rough set [49], probabilistic rough set [50, 51, 40], etc., can be used to solve this issue. Each of these rough set models can be used to control the degree of uncertainty, misclassification and imprecise information. We can see that for these rough set models, lower/upper approximation of a target concept are often not monotonic with the number of attributes, where objects outside this target concept may be included. How to ensure the monotonicity of an attribute reduction process is also a motivation of this study.

In order to address these three challenges, a new rough set model for rough data analysis in big data, called local neighborhood rough set, is presented. To construct lower/upper approximations of a target concept under the learning framework of the local neighborhood rough set, it is unnecessary to compute information granules of all objects in advance. Only those of objects within a target concept need to be calculated. This saves a great amount of computing time and fully meets the needs of big data analysis. Some interesting properties and measures in the local neighborhood rough set will also be given. Based on the local rough set, the LLAC algorithm for computing a local lower approximation of a target concept and the LARC algorithm for searching a local attribute reduction of a target concept, were designed. Moreover, the LLAD algorithm for calculating a local lower approximation of a target decision and the LARD algorithm for finding a local attribute reduction of a target decision, will be proposed. The one of the advantages of these four algorithms is that their time complexity is linear. Hence, LNRS can fully be apply to rough data analysis in big data. At last, we use four real data sets from UCI and an artificial data set to verify the performance of these four algorithms. Corresponding experiment results show that these algorithms achieve a great success for rough data analysis in big data.

The remainder of this paper is organized as follows. In section 2, local rough set and neighborhood rough set are reviewed. In section 3, we first construct the local neighborhood rough set and explore its prime properties and measures. Section 4 provides solutions of how to compute the lower/up local approximation of a target concept and how to find an attribute reduction of a target decision in the local neighborhood rough set. In Section 5, we verify scalability of the local neighborhood rough set on an artificial large-scale data set. Finally, we conclude this paper by outlook for further research and discussion in Section 6.

2. Related work

In this section, we briefly review some basic concepts related to local rough set (LRS) [52] and neighborhood rough set (NRS) [53].

2.1. LRS

For obtaining a rough set $< lower\ approximation,\ upper\ approximation >$ of any subset on sample set, one first computes all the information granules by comparing the difference between any two objects from a given data set. This implies that a global rough set must observe the relationships between a target concept and each of the information granules. However, this is not a good strategy for approximating a target concept $X \subseteq U$. In fact, the information granules $\{[x] : [x] \cap X = \phi,\ x \in U\}$ are not useful for computing the lower/upper approximation of X. Indeed, we only need to calculate the information granules related to the target concept X. In particular, this kind of large-scale data sets $n >> | X |$ often exist in real applications (even we can have lots of labeled data, we still can obtain more unlabeled data. For examples, ImageNet consists of over 14 million hand-annotated images, its amount still is less than the amount of unlabeled images on Internet), where n and $|X|$ are the size of the data set and X, respectively. This time reduction improvement would be very useful for rough data analysis based on big data. According to this consideration, Qian et al.[52] reconstructed the rough set model as follows:

Definition 1. *[52] Let (U, R) be an approximation space and $\mathcal{D}$ is an inclusion degree defined on $\mathcal{P}(U)\times\mathcal{P}(U)$. Then, for any $X \subseteq U$, the $\alpha-$lower and $\beta-$upper approximations are defined by*

$$\underline{R}_{(LRS,\alpha)}(X) = \{x \mid \mathcal{D}(X/[x]_R) \geq \alpha,\ x \in X\}, \tag{2}$$

$$\overline{R}_{(LRS,\beta)}(X) = \{x \mid \mathcal{D}(X/[x]_R) > \beta,\ x \in U\} = \cup\{[x]_R \mid \mathcal{D}(X/[x]_R) > \beta,\ x \in X\}. \tag{3}$$

The pair $\langle \underline{R}_{(LRS,\alpha)}(X), \overline{R}_{(LRS,\beta)}(X)\rangle$ is called the LRS.

The boundary of X is denoted by $BN_R(X) = \overline{R}_{(LRS,\beta)}(X) - \underline{R}_{(LRS,\alpha)}(X)$, which we refer to as the local boundary region of X.

2.2. NRS

Let $< U, C \cup D >$ be a decision information system, where $U = \{x_1, x_2, \cdots, x_n\}$ is a finite and nonempty set of objects, x_i is a sample, C and D, described samples,are called condition attribute and decision attribute respectively.

For any $x_i, x_j, x_k \in U$, there exists a corresponding number Δ satisfying

$(1)\Delta(x_i, x_j) \geq 0, \Delta(x_i, x_j) = 0$ if and only if $x_i = x_j$,

$(2)\Delta(x_i, x_j) = \Delta(x_j, x_i)$, and

$(3)\Delta(x_i, x_j) + \Delta(x_j, x_k) \geq \Delta(x_i, x_k)$,

where Δ is a distance function and $< U, \Delta >$ is a metric space.

For any two samples $x_i = (x_i^1, x_i^2, ..., x_i^N)$ and $x_j = (x_j^1, x_j^2, ..., x_j^N)$ a general metric, named Minkowsky distance, is defined as

$\Delta(x_i, x_j) = (\sum_{k=1}^{N} |x_i^k - x_j^k|^P)^{\frac{1}{P}}$

In fact, the distance function has various forms, a detailed survey on this topic can refer to [34]. In this paper, we select Minkowsky distance, where $P = 2$ (also called Euclidean distance).

Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of x_i in feature space B is defined as

$\delta_B(x_i) = \{x_j | \Delta(x_i, x_j) \leq \delta\}$

Given an approximation space neighborhood approximation space $< U, N >$, where U is a set of objects and N is a neighborhood relation over U. For an any subset $X \subseteq U$, lower and upper approximation are defined as

$$\begin{cases} \underline{N}(X) = \{x | \delta(x) \subseteq X, x \epsilon U)\}, \\ \overline{N}(X) = \{x | X \cap \delta(x) \neq \phi, x \epsilon U)\}. \end{cases} \tag{4}$$

3. Local neighborhood rough set

Rough set is a useful tool for rough data uncertainty analysis. But these tasks based on the global rough set are time-consuming for many data sets, especially big data. In order to use rough set to handle big data efficiently, we introduce a general rough set framework, called local neighborhood rough set.

3.1. Construction of a local neighborhood rough set and its properties

In GNRS, all information granules are obtained by comparing otherness between any two objects from a given universe. The strategy means that we need to consider the relationship between a target concept and each information granule. However, this is not a good strategy for approximating target concept $X \subset U$. In fact, we only use the information granules $\{\delta(x) | \delta(x) \cap X \neq \phi, x \in X\}$, when approximating a target concept. This implies that the information granules $\{\delta(x) | \delta(x) \cap X = \phi, x \in X\}$ is useless for constructing lower/upper approximation of X. Hence, it is unnecessary to compute them. Namely, we only need to compute the $\{\delta(x) | \delta(x) \cap X \neq \phi, x \in X\}$, which can immensely reduce time consumption. In the context of big data, this kind of large-scale data sets $n >> | X |$ often exist, in which n and $| X |$ are the size of this data and X, respectively.

Based on the above analysis, in order to analyse big data by applying rough set theory, we reconstruct rough set model as follows:

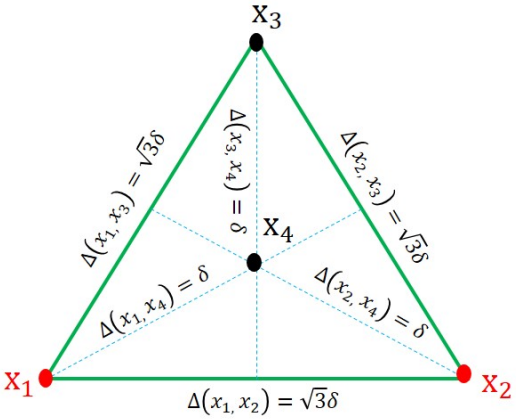


Figure 1: Space structure for the toy example.

Definition 2. *Let NAS* $= (U, N)$ *be a neighborhood approximation space, and* $\mathcal{D}$ *an inclusion degree defined on* $P(U) \times P(U)$*. Then for any* $X \subseteq U$*, the* $\alpha -$ *lower and* $\beta -$ *upper approximation are denoted by*

$$\begin{cases} \underline{N}_\alpha(X) = \{x | \mathcal{D}(X/\delta(x)) \geq \alpha, x \epsilon X\}, \\ \overline{N}_\beta(X) = \{x | \mathcal{D}(X/\delta(x)) > \beta, x \epsilon X\}. \end{cases} \tag{5}$$

where $\delta(x) = \{y | \Delta(x, y) \leq \delta\}$, Δ is a distance function, $\mathcal{D}(X/\delta(x)) = \frac{|X \cap \delta(x)|}{|\delta(x)|}$ referred to as the degree of inclusion [54].

The pair $\langle \underline{N}_\alpha(X), \overline{N}_\beta(X) \rangle$ is called a local neighborhood rough set. The boundary of X is denoted by $BN_N(X) = \overline{N}_\beta(X) - \underline{N}_\alpha(X)$.

Example 1. *Given a samples set* $U = \{x_1, x_2, x_3, x_4\}$*, corresponding label set* $D = \{y_1, y_2, *, *\}$ *('*' denotes that corresponding samples are not labeled), samples set with labels* $X = \{x_1, x_2\}$*, and* $\alpha = 0.5$*. Their space structure is shown in Fig. 1.*

For LNRS, we only need to obtain neighborhood relations for these objects from X. According to Fig 1, we have: $\delta(x_1) = \{x_1, x_4\}$, $\delta(x_2) = \{x_2, x_4\}$.
Further, there are $\mathcal{D}(X/\delta(x_1)) = \frac{1}{2}$ *and* $\mathcal{D}(X/\delta(x_2)) = \frac{1}{2}$.
Based on Definition 2, one can get $\underline{N}^{LNRS}_{\alpha=0.5}(X) = \{x_1, x_2\}$.
However, for GNRS, we need to obtain neighborhood relations for all objects from U. According to Fig 1, we get: $\delta(x_1) = \{x_1, x_4\}$, $\delta(x_2) = \{x_2, x_4\}$, $\delta(x_3) = \{x_3, x_4\}$, $\delta(x_4) = \{x_4, x_1, x_2, x_3\}$.
Further, there are $\mathcal{D}(X/\delta(x_1)) = \frac{1}{2}$, $\mathcal{D}(X/\delta(x_2)) = \frac{1}{2}$, $\mathcal{D}(X/\delta(x_3)) = 0$, *and* $\mathcal{D}(X/\delta(x_4)) = \frac{1}{2}$.
Based on Equation (1), we can obtain $\underline{N}^{GNRS}_{\alpha=0.5}(X) = \{x_1, x_2, x_4\}$

From Example 1, we can find that we only need to obtain these neighborhood relations of samples with labels in the context of LNRS. It is worth noting that obtaining these neighborhood relations of samples with labels needs to use the information hidden in these samples without labels. However, for GNRS model, one has to compute all neighborhood relations of all samples with labels and without labels.

The local neighborhood rough set degrades to a neighborhood rough set if $\alpha = 1$ and $\beta = 0$, which implies that the generalized model maintains the consistent ability to deal with uncertainty and does not change the idea of the latter (it can be seen as a type of global neighborhood rough set).

In fact, these objects from $\underline{N}_\alpha(X)$, can be further divided into two categories. $\underline{N}_\alpha(X) = \{x |\ \mathcal{D}(X/\delta(x)) \geq \alpha,\ x \in X\} = \{x |\ \mathcal{D}(X/\delta(x)) = 1,\ x \in X\} \cup \{x |\ 1 > \mathcal{D}(X/\delta(x)) \geq \alpha,\ x \in X\}$. Here, we denote $CL_N(X) = \{x |\ \mathcal{D}(X/\delta(x)) = 1,\ x \in X\}$, called a certain set of $\underline{N}_\alpha(X)$, and denote $PL_N(X) = \{x |\ 1 > \mathcal{D}(X/\delta(x)) \geq \alpha,\ x \in X\}$, called a possible set of $\underline{N}_\alpha(X)$. It is obvious that

$$\underline{N}_\alpha(X) = CL_N(X) \cup PL_N(X) \text{ and } |\underline{N}_\alpha(X)| = |CL_N(X)| + |PL_N(X)|.$$

Theorem 1. *Given two equivalence relations P, Q with $P \prec Q$, a target concept X and the parameter α. If $x \in CL_Q(X)$, then $x \in CL_P(X)$.*

Proof. If $x \in CL_Q(X)$, from the definition of $CL_Q(X)$, one has that $\mathcal{D}(X/\delta_Q(x)) = 1$, so $\frac{|X \cap \delta_Q(x)|}{|\delta_Q(x)|} = 1$, thus $\delta_Q(x) \subseteq X$. In addition, due to $P \prec Q$, we have that $\delta_P(x) \subseteq \delta_Q(x) \subseteq X$. Thus $\mathcal{D}(X/\delta_P(x)) = 1$. Then, the object $x \in CL_P(X)$.

However, for every object coming from the possible set $PL_N(X)$, the above property may not hold, as it is affected by the parameter α.

Compared with classical rough set, local rough set possesses some interesting properties.

Property 1. *Given an approximation space (U, N), and an inclusion degree defined $\mathcal{D}$. Then, for any $X, Y \subseteq U, 0 \leq \beta < \alpha \leq 1$, the following properties hold:*

(1) $\underline{N}_\alpha(X) \subseteq X$;

(2) $\beta \in [0, min\{\mathcal{D}(X/\delta(x)) : x \in X\}] \Rightarrow X \subseteq \overline{N}_\beta(X)$;

(3) $\underline{N}_\alpha(\phi) = \overline{N}_\beta(\phi) = \phi, \quad \underline{N}_\alpha(U) = \overline{N}_\beta(U) = U$;

(4) $X \subseteq Y \Rightarrow \underline{N}_\alpha(X) \subseteq \underline{N}_\alpha(Y), \overline{N}_\beta(X) = \overline{N}_\beta(Y)$;

(5) $\underline{N}_\alpha(X \cap Y) \subseteq \underline{N}_\alpha(X) \cap \underline{N}_\alpha(Y)$,
$\overline{N}_\beta(X \cup Y) \supseteq \overline{N}_\beta(X) \cup \overline{N}_\beta(Y)$;

(6) $\underline{N}_\alpha(X \cup Y) \supseteq \underline{N}_\alpha(X) \cup \underline{N}_\alpha(Y)$,
$\overline{N}_\beta(X \cap Y) \subseteq \overline{N}_\beta(X) \cap \overline{N}_\beta(Y)$;

(7) $0.5 < \alpha_1 < \alpha_2 \leq 1 \Rightarrow \underline{N}_{\alpha_2}(X) \subseteq \underline{N}_{\alpha_1}(X)$,
$0 \leq \beta_1 < \beta_2 < 0.5 \Rightarrow \overline{N}_{\beta_2}(X) \subseteq \overline{N}_{\beta_1}(X)$.

Proof. (1)For $\forall x \in \underline{N}_\alpha(X)$, by $\alpha - lower$ approximation in Definition 1, we can get $x \in X$. So, for any $X \subseteq U$ and $0 < \alpha \leq 1$, one has $\underline{N}_\alpha(X) \subseteq X$.

(2) Since N is a neighborhood relation on U, $\forall x \in X$, one has that $x \in \delta(x)$ and $X \cap \delta(x) \neq \emptyset$. Hence, we get that $\mathcal{D}(X/\delta(x)) > 0$. Thus, when $\beta \in [0, \min\{\mathcal{D}(X/\delta(x)) : x \in X\})$, we have that $\forall x \in X$, $x \in \delta(x)$ and $\mathcal{D}(X/\delta(x)) > \beta$ hold. Therefore, from the definition of $\beta-$upper approximation in Definition 1, when $\beta \in [0, \min\{\mathcal{D}(X/\delta(x)) : x \in X\})$, we can get that $x \in \overline{N}_\beta(X)$, $\forall x \in X$, that is $X \subseteq \overline{N}_\beta(X)$.

(3)For $\forall x \in U, 0 \leq \beta < \alpha \leq 1$, we can get $\mathcal{D}(\phi/\delta(x)) = \frac{|\phi \cap \delta(x)|}{|\delta(x)|} = 0 \leq \beta < \alpha$, $x \notin \underline{N}_\alpha(\phi), x \notin \overline{N}_\beta(\phi)$. Thus, $\underline{N}_\alpha(\phi) = \overline{N}_\beta(\phi) = \phi$. Furthermore, We can get $\mathcal{D}(U/\delta(x)) = \frac{|U \cap \delta(x)|}{|\delta(x)|} = 1 \geq \beta > \alpha$, $x \in \underline{N}_\alpha(U), x \in \overline{N}_\beta(U)$. So, one can get that $\underline{N}_\alpha(U) = \overline{N}_\beta(U) = U$.

(4)For $\forall x \in U$,when $X \subseteq Y$, one can get easily $\mathcal{D}(X/\delta(x)) < \mathcal{D}(Y/\delta(x))$. For $\forall x \in \underline{N}_\alpha(X)$, when $X \subseteq Y$, we have $\mathcal{D}(Y/\delta(x)) \geq \mathcal{D}(X/\delta(x)) \geq \alpha$.Thus, $x \in \underline{N}_\alpha(Y).\underline{N}_\alpha(X) \subseteq \underline{N}_\alpha(Y)$. Analogously, we can prove that $X \subseteq Y$ implies $\overline{N}_\beta(X) = \overline{N}_\beta(Y)$.

(5)For $\forall x \in \underline{N}_\alpha(X \cap Y)$, one can get that

$x \in \underline{N}_\alpha(X \cap Y) \Rightarrow x \in X \cap Y, \mathcal{D}(X \cap Y/\delta(x)) \geq \alpha$
$\Rightarrow x \in X \wedge x \in Y, \mathcal{D}(X \cap Y/\delta(x)) \geq \alpha$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \mathcal{D}(X \cap Y/\delta(x)) \geq \alpha, x \in X$
$\wedge\ \mathcal{D}(Y/\delta(x)) \geq \mathcal{D}(X \cap Y/\delta(x)) \geq \alpha, x \in Y$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \alpha, x \in X \wedge \mathcal{D}(Y/\delta(x)) \geq \alpha, x \in Y$
$\Rightarrow x \in \underline{N}_\alpha(X) \wedge x \in \underline{N}_\alpha(Y)$
$\Rightarrow x \in \underline{N}_\alpha(X) \cap \underline{N}_\alpha(Y)$

from which one can get that $\underline{N}_\alpha(X \cap Y) \subseteq \underline{N}_\alpha(X) \cap \underline{N}_\alpha(Y)$.

Analogously, for $\forall x \in \overline{N}_\beta(X) \cup \overline{N}_\beta(Y)$,we can get that

$x \in \overline{N}_\beta(X) \cup \overline{N}_\beta(Y)$
$\Rightarrow x \in \overline{N}_\beta(X) \vee x \in \overline{N}_\beta(Y)$
$\Rightarrow \mathcal{D}(X/\delta(x)) > \beta, x \in X \vee \mathcal{D}(Y/\delta(x)) > \beta, x \in Y$

$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) \geq \mathcal{D}(X/\delta(x)) > \beta, x \in X \vee$
$\mathcal{D}(X \cup Y/\delta(x)) \geq \mathcal{D}(Y/\delta(x)) > \beta, x \in Y$
$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) > \beta, (x \in X \vee x \in Y)$
$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) > \beta, (x \in (X \cup Y))$
$\Rightarrow x \in \overline{N}_\beta(X \cup Y)$
then we obtain that $\overline{N}_\beta(X \cup Y) \supseteq \overline{N}_\beta(X) \cup \overline{N}_\beta(Y)$.

(6)For $\forall x \in \underline{N}_\alpha(X) \cup \underline{N}_\alpha(Y)$,we can get that
$x \in \underline{N}_\alpha(X) \cup \underline{N}_\alpha(Y)$
$\Rightarrow x \in \underline{N}_\alpha(X) \vee x \in \underline{N}_\alpha(Y)$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \alpha, x \in X \vee \mathcal{D}(Y/\delta(x)) \geq \alpha, x \in Y$
$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) \geq \mathcal{D}(X/\delta(x)) \geq \alpha, x \in X \vee$
$\mathcal{D}(X \cup Y/\delta(x)) \geq \mathcal{D}(Y/\delta(x)) \geq \alpha, x \in Y$
$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) \geq \alpha, (x \in X \vee x \in Y)$
$\Rightarrow \mathcal{D}(X \cup Y/\delta(x)) \geq \alpha, (x \in (X \cup Y))$
$\Rightarrow x \in \underline{N}_\alpha(X \cup Y)$ so one can get that $\underline{N}_\alpha(X) \cup \underline{N}_\alpha(Y) \subseteq \underline{N}_\alpha(X \cup Y)$.

Meanwhile, for $\forall x \in \overline{N}_\beta(X \cap Y)$,one can get that
$x \in \overline{N}_\beta(X \cap Y) \Rightarrow x \in X \cap Y, \mathcal{D}(X \cap Y/\delta(x)) \geq \beta$
$\Rightarrow x \in X, x \in Y, \mathcal{D}(X/\delta(x)) \geq \beta, \mathcal{D}(Y/\delta(x)) \geq \beta$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \beta, x \in X \wedge \mathcal{D}(Y/\delta(x)) \geq \beta, x \in Y$
$\Rightarrow x \in \overline{N}_\beta(X) \wedge x \in \overline{N}_\beta(Y)$
$\Rightarrow x \in \overline{N}_\beta(X) \cap \overline{N}_\beta(Y)$
from which one can get that $\overline{N}_\beta(X \cap Y) \subseteq \overline{N}_\beta(X) \cap \overline{N}_\beta(Y)$;

(7)For $0.5 < \alpha_1 < \alpha_2 \leq 1, \forall x \in \underline{N}_{\alpha_2}(X)$
$x \in \underline{N}_{\alpha_2}(X) \Rightarrow \mathcal{D}(X/\delta(x)) \geq \alpha_2, x \in X$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \alpha_2 > \alpha 1, x \in X$
$\Rightarrow \mathcal{D}(X/\delta(x)) \geq \alpha 1, x \in X \Rightarrow x \in \underline{N}_{\alpha_1}$
Then $0.5 < \alpha_1 < \alpha_2 \leq 1$ implies that $\underline{N}_{\alpha_2}(X) \subseteq \underline{N}_{\alpha_1}(X)$.

Analogously, we can prove that $0 \leq \beta_1 < \beta_2 < 0.5 \Rightarrow \overline{N}_{\beta_2}(X) \subseteq \overline{N}_{\beta_1}(X)$.

In real applications, for a classification problem, a decision table $S = (U, C \cup D)$ with $C \cap D = \emptyset$ is often used, where C and D are called condition attribute set and decision attribute set, respectively.

In machine learning, a classifier is built on a supervised learning algorithm with labeled training data. In rough set theory, a rough classifier is also learned from an object set with class labels (called a training set). However, in big data analysis, supervised learning often requires a large amount of labeled data, which is an expensive and laborious task and sometimes even infeasible. In contrast, unlabeled data are cheap and easy to obtain because a large amount of them can be easily collected. Therefore, techniques to automatically do rough data analysis on big data with limited labeled data in a semi-supervised way, are desirable.

In the above case, we assume that U is the entire universe, $U_{lable} \subseteq U$ is labeled sample set, $U_{unlable} = U - U_{lable}$ is unlabeled sample set, and U is parted into r mutually exclusive crisp subsets by the decision attributes D, i.e., $U_{lable}/D = \{X^1, X^2, \cdots, X^r\}$. For many large scale data sets, we often have that $|U_{lable}| \ll |U|$. Given any subset $B \subseteq C$ and N_B is the neighborhood relation induced by B, like a global rough set, in such a case, the local lower and local upper approximations of the decision attributes D with respect to B are defined as

$$\begin{cases} \underline{N_B}(D) = \{\underline{N_B}(X^1), \underline{N_B}(X^2), \cdots, \underline{N_B}(X^r)\}, \\ \overline{\underline{N_B}}(D) = \{\overline{\underline{N_B}}(X^1), \overline{\underline{N_B}}(X^2), \cdots, \overline{\underline{N_B}}(X^r)\}. \end{cases}$$

Denoted by $POS_B(D) = \bigcup_{i=1}^{r} \underline{N_B}(X^i)$, it is called the local positive region of D.

3.2. Several measures in the local neighborhood rough set

Like Pawlak' rough set, uncertainty of a local neighborhood rough set is caused by the existence of a boundary region. As we know, the greater the boundary region of rough set is, the weaker its accuracy is. In the local rough set, in order to express this idea, the formal definition of an accuracy measure is given as follows:

Definition 3. *Let $S = (U, AT)$ be an information system, $X \subseteq U$ and $B \subseteq AT$ an attribute subset. The accuracy measure of X by B is defined as*

$$\alpha(B,\ X) = \frac{|\underline{N_B}(X)|}{|\overline{N_B}(X)|}, \tag{6}$$

Gediga [55] introduced a simple statistic for the approximation precision of X by B. In local neighborhood rough set, it can be defined as

$$\pi(N, X) = \frac{|\underline{N_B}(X)|}{|X|}. \tag{7}$$

In rough set theory, accuracy of approximation is employed for describing the ability of a partition to approximate a decision[47],[48]. For a decision table $S = (U, C \cup D)$ and an attribute subset $B \subseteq C$, given a local rough set, the approximation accuracy of B with respect to D can be given as follows:

$$\gamma(B, D) = \frac{\sum\{|\underline{N_B}(X)| : X \in U/D\}}{\sum\{|X| : X \in U/D\}} = \frac{|POS_B(D)|}{\sum\{|X| : X \in U/D\}}. \tag{8}$$

where $X \in U/D$ means a labeled class.

In this paper, the precision of approximation π and γ is used to compare the monotonicity of the proposed local rough set and the global rough set in attribute reduction of a target concept and a target decision respectively.

4. Computing approximation of a target concept and attribute reduction of a target decision

In this section, we focus on approximation and attribute reduction of a target concept.

4.1. Computing the local lower approximation of a target concept

In this part, an algorithm computing a local lower approximation of a target concept is designed, and its efficiency is verified on several real data sets.

4.1.1. Algorithm

From Definition 1 and relative analysis, we know, in order to obtain lower/upper approximation of local rough set, we only need to compute information granules of objects within a given target concept. Here in below, we give corresponding algorithm description.

Algorithm 1. Computing the local lower approximation of a target concept(LLAC)

Input: An information system $S = (U, AT)$, a target concept set X, inclusion degree α and neighborhood size δ;

Output: Local $\alpha - lower$ approximation LA of X.

(1) for each $x \in X$, compute $\delta(x)$

(2) $LA \leftarrow \phi, i \leftarrow 1$.

(3) for each $x \in X$

if $\mathcal{D}(X/\delta(x) \geq \alpha$

$LA \leftarrow LA \cup \{x\}$

(4) return LA and end.

For convenience, we name the algorithm of computing a global lower approximation GLAC. In this part, we evaluate time complexity of the LLAC algorithm. Step 1 needs to compute $|X|$ neighborhood information granules through scanning the entire universe U, thus its time complexity is $O(|X||U|)$. But the time complexity of calculating all neighborhood information granules is $O(|U|^2)$ for the classical GLAC. In Step 3, in order to get lower approximation of the LLAC, we only need to compare the $|X|$ neighborhood information granules with the concept X. Hence, its time complexity is $O(|X|^2)$. However, the time complexity for obtaining lower approximation of global rough set is $O(|X||U|)$. Each of other steps of the LLAC and GLAC is constant. To intuitively compare, the time complexity of each step in LLAC and that GLAC are shown in Table 2. From Table 2, we can see that the time complexity of the GLAC is much higher than that of the LLAC. In the next theorem, we quantitatively evaluate the ratio of time-reduction of the LLAC algorithm relative to the GLAC algorithm.

Table 2: The time complexities of the LLAC algorithm and GLAC algorithm

Algorithms	Step 1	Step 3	Other steps
Global lower approximation	$O(\|U\|^2)$	$O(\|X\|\|U\|)$	Constant
Local lower approximation	$O(\|X\|\|U\|)$	$O(\|X\|^2)$	Constant

Table 3: Datasets description

Id	Datasets	Cases	Features	Classes
1	EEG	14980	14	2
2	Hill-Valley	606	100	2
3	Magic	19020	10	2
4	Occupancy Detection	20560	5	2

Theorem 2. *Given an information system $S = (U, AT)$ and a target concept $X \subseteq U$, the speedup ratio of the LLAC algorithm is $p = 1 - \frac{|X|}{|U|}$ relative to the GLAC algorithm.*

Proof. From the time complexity of LLAC algorithm and that of a global lower approximation in Table 2, we have that

$$\begin{aligned} p &= (O(|U|^2 + |X||U|) - O(|X||U| + |X|^2))/O(|U|^2 + |X||U|) \\ &= O(|U|^2 + |X||U| - |X||U| - |X|^2)/O(|U|^2 + |X||U|) \\ &= O(|U|^2 - |X|^2)/O(|U|^2 + |X||U|) \\ &= O((|U| + |X|)(|U| - |X|))/O(|U|(|U| + |X|)) \\ &= O(|U| - |X|)/O(|U|) \\ &= 1 - \frac{|X|}{|U|}. \end{aligned}$$

Hence, the ratio of time-reduction of the LLAC algorithm is $p = 1 - \frac{|X|}{|U|}$. This completes the proof.

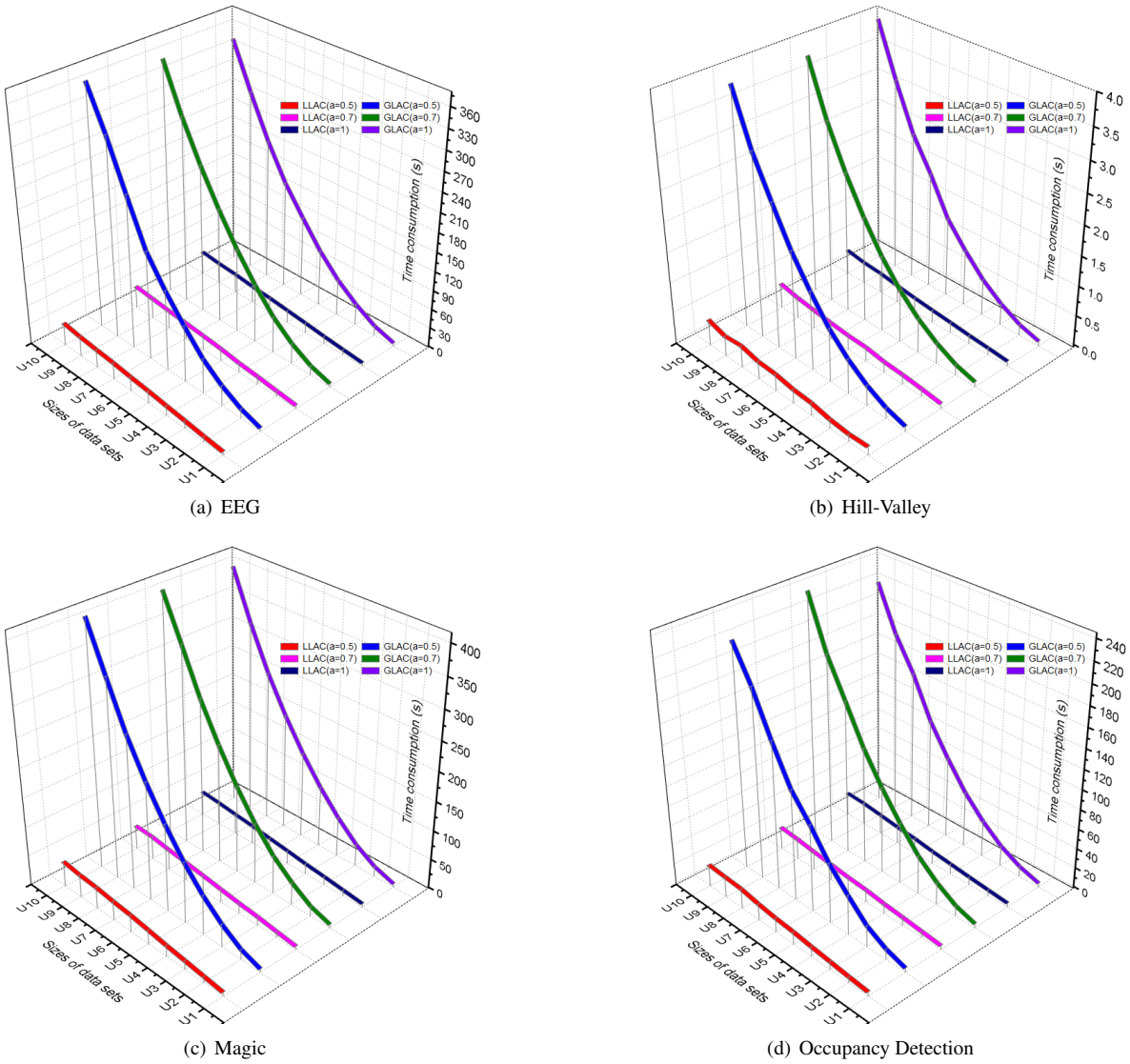
From the above theorem, we can see that the LLAC algorithm can improve $\frac{O(|U|^2+|X||U|)}{O(|X||U|+|X|^2)} = \frac{|U|}{|X|}$ times than the corresponding global lower approximation for computational time. Therefore, it is more efficient for computing a local lower approximation. Particularly, the size $|X|$ of the target concept is far less than the size $|U|$ of the universe, i.e., $|X| << |U|$ in the context of big data. Then $|X|$ may be regarded as a constant. In other words, the time complexity of LLAC is line $O(|X||U|+|X|^2)$ in term of $|U|$. Hence, the LLAC algorithm can be highly efficient for big data analysis.

4.1.2. Experimental analysis

In the experiments in this paper, all algorithms are all run on a personal computer with Windows 10 and Core(TM) I7-4790 CPU 3.6 GHz, and ECC DDR3, 8 GB memory. The software being used is JAVA. Without loss of generality, we only let the parameter $\alpha = 0.5$, $\alpha = 0.7$, and $\alpha = 1$, respectively and $\delta = 0.001$.

For experimental design, we fix the size of target concept, which is the front 10% objects from each of these four data sets. To distinguish the computational time, we divide each of these four data sets into ten parts of equal size. These samples from the first part are labeled, and these samples from other parts are not labeled. The first part is regarded as the 1st data set, the combination of the first part and the second part is viewed as the 2nd data set, ..., the combination of all ten parts is viewed as the 10th data set. These data sets can be used to calculate time used by each of the LLAC algorithm and the GLAC algorithm and Fig. 2 shows it vis-a-vis the size of universe.

Fig. 2 displays the change trends of both LLAC and GLAC with the increase of data set size (the x-coordinate and y-coordinate pertain to the size of the data set and the computing time, respectively). From Fig. 2, we can see the computing time of these two algorithms increases with the increase of the size of data. In addition, for each value of the parameter α, the LLAC algorithm is consistently faster than the GLAC algorithm on the same universe and attribute set. The differences between these two algorithms for time consumption are markedly larger when the size of the data set increases. Table 4 shows that the computational time of LLAC and GLAC algorithms on the tenth data



(a) EEG (b) Hill-Valley

(c) Magic (d) Occupancy Detection

Figure 2: Time of LLAC and GLAC versus the size of universe

set in the nine data sets. From Table 4, we observe that it only uses one tenth of the run time used by GLAC. One can say that for computing the local lower approximation of a target concept, the LLAC algorithm under the local rough set is efficient for handling big data.

4.2. Computing local attribute reduction of a target decision

In this subsection, we develop a heuristic, greedy and forward search algorithm for searching a local attribute reduction of a target decision algorithm and verify its efficiency through employing several real data sets.

4.2.1. Definition of a local attribute reduction of a target decision

The aim of attribute reduction is to find a subset of attributes so that the classification task has the maximal consistency in the reduction set. However, the lower approximation of any rough set model with a parameter $\alpha \neq 1$ is not monotonic, which often causes over-fitting problem [56, 48]. This is induced by the very strong constraint

Table 4: The computational time for concept approximation with LLAC and GLAC

Datasets	$\alpha = 0.5$		$\alpha = 0.7$		$\alpha = 1$	
	LARC(s)	GARC(s)	LARC(s)	GARC(s)	LARC(s)	GARC(s)
EEG	34.0398	371.5881	33.8306	361.4159	35.0514	350.7959
Hill-Valley	0.3924	3.8723	0.3790	3.8501	0.3714	4.0022
Magic	42.1641	414.2564	40.2146	410.2744	40.4122	404.7643
Occupancy Detection	20.7668	223.5289	21.4709	241.3208	21.7125	224.1059

$\underline{N_B}(X) = \underline{N_{AT}}(X)$ of an attribute reduction [30, 9]. Besides, much more run time is spent in the searching process. For global rough set model, the task of attribute reduction of a target decision has the same limitation, where the stopping criterion $POS_B(D) = POS_C(D)$ is not only too strong but also time-consuming. To address the two limitations of low efficiency and over-fitting, we introduce the definition of local attribute reduction of a target decision as follows

Definition 4. *Let $S = (U, C \cup D)$ be a decision table and $B \subseteq C$. If $|POS_B(D)| \geq |POS_C(D)|$ and $|POS_{B'}(D)| \ngeq |POS_C(D)|$ for any $B' \subset B$, then we call B a local attribute reduction of S.*

From the above definition, we can know that there may be multiple attribute reductions for a target decision with class labels. Let $\{B_1, B_2, \cdots, B_s\}$ be s local attribute reductions of S with respect to D, its core can be written $Core = B_1 \cap B_2 \cap \cdots \cap B_s$. The attributes from the core are indispensable for constructing a local attribute set of a target decision. It is noted that sometimes the core can be empty.

4.2.2. Algorithm

In fact, Definition 3 may induce multiple attribute reductions for a target decision with class labels. But in most applications, it is enough to find one of them. At present, some heuristic algorithms based on greedy and forward search algorithms have been designed based on the significance measures of attributes in global rough set. For this kind of attribute reduction approaches, two important measures of attributes are used for heuristic functions, which are inner importance measure and outer importance measure. In this local rough set, we can also develop a local positive-region based attribute reduction algorithm, in which the significance measures of attributes are defined as follows:

Definition 5. *Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in B$. The inner significance measure of a in B is defined as*

$$Sig^{inner}(a, B, D, U) = \gamma_B(D) - \gamma_{B-\{a\}}(D),$$

where $\gamma_B(D) = \frac{|POS_B(D)|}{|U|}$, and $POS_B(D)$ is the local positive region of B with respect to D.

Definition 6. *Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in C - B$. The outer significance measure of a in B is defined as*

$$Sig^{outer}(a, B, D, U) = \gamma_{B\cup\{a\}}(D) - \gamma_B(D).$$

From the definition and relation analysis, we see the computation of the two important measures of attributes is the main part of time consumption of a heuristic attribute reduction algorithm. To reduce the computational time, we will want to introduce an efficient strategy of heuristic attribute reduction, in which we concentrate on the rank preservation of the significance measures of attributes in a decision table. Simply, we denote the certain set $POS_B(D)$ of on the universe U by $CP_B^U(D) = \cup\{\delta_B^D(x) | \mathcal{D}(X/\delta_B^D(x)) = 1, x \in X, X \in U/D\}$, called the certain positive region.

Theorem 3. *Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $U' = U - CP_B^U(D)$. For $\forall a, b \in C - B$, if $Sig^{outer}(a, B, D, U) \geq Sig^{outer}(b, B, D, U)$, then $Sig^{outer}(a, B, D, U') \geq Sig^{outer}(b, B, D, U')$.*

Proof. From the definition of $Sig^{outer}(a, B, D) = \gamma_{B\cup\{a\}}(D) - \gamma_B(D)$, we know that its value only depends on the dependency function $\gamma_B(D) = \frac{|POS_B(D)|}{|U|}$.

Since $U^{'} = U - \cup\{\delta_B^U(x)|\mathcal{D}(X/\delta_B^U(x)) = 1, x \in X, X \in U/D\}$, so there must exist $X \in U/D$ such that $1 > \mathcal{D}(X/\delta_B^U(x)) \geq \alpha$ for $\forall x \in U^{'}$, then the object x belongs to the positive region $POS_B^{U^{'}}(D)$. Hence $\forall\alpha > 0$, we have that

$POS_B^U(D) = \{x|\mathcal{D}(X/\delta_B^U(x)) \geq \alpha, x \in X,\ X \in U/D\}$
$= \{x|1 > \mathcal{D}(X/\delta_B^U(x)) \geq \alpha,\ x \in X, X \in U/D\} \cup \{x|\mathcal{D}(X/\delta_B^U(x)) = 1,\ x \in X, X \in U/D\}$
$= POS_B^{U^{'}}(D) \cup \{x|\mathcal{D}(X/\delta_B^U(x)) = 1,\ x \in X, X \in U/D\}$
From the above equation and the definition of local positive region, we have that
$|POS_{B\cup\{a\}}^{U^{'}}(D)| - |POS_B^{U^{'}}(D)|$
$= |POS_{B\cup\{a\}}^{U}(D)| - |\{x|\mathcal{D}(X/\delta_B^U(x)) = 1, x \in X, X \in U/D\}| - |POS_B^{U^{'}}(D)|$
$= |POS_{B\cup\{a\}}^{U}(D)| - (|\{x|\mathcal{D}(X/\delta_B^U(x)) = 1, x \in X, X \in U/D\}| + |POS_B^{U^{'}}(D)|)$
$= |POS_{B\cup\{a\}}^{U}(D)| - (|\{x|\mathcal{D}(X/\delta_B^U(x)) = 1, x \in X, X \in U/D\} \cup POS_B^{U^{'}}(D)|)$
$= |POS_{B\cup\{a\}}^{U}(D)| - |POS_B^{U}(D)|$
Therefore, we have
$\frac{Sig^{outer}(a,B,D,U)}{Sig^{outer}(a,B,D,U^{'})} = \frac{\gamma_{B\cup\{a\}}^{U}(D)-\gamma_B^U(D)}{\gamma_{B\cup\{a\}}^{U^{'}}(D)-\gamma_B^{U^{'}}(D)}$
$= \frac{|U^{'}|}{|U|}\frac{|POS_{B\cup\{a\}}^{U}(D)|-|POS_B^{U}(D)|}{|POS_{B\cup\{a\}}^{U^{'}}(D)|-|POS_B^{U^{'}}(D)|}$
$= \frac{|U^{'}|}{|U|}\frac{|POS_{B\cup\{a\}}^{U}(D)|-|POS_B^{U}(D)|}{|POS_{B\cup\{a\}}^{U}(D)|-|POS_B^{U}(D)|} = \frac{|U^{'}|}{|U|}$.

Because $\frac{|U^{'}|}{|U|} \geq 0$ and $Sig^{outer}(a, B, D, U) \geq Sig^{outer}(b, B, D, U)$, hence $Sig^{outer}(a, B, D, U^{'}) \geq Sig^{outer}(b, B, D, U^{'})$. This completes the proof.

From the above theorem, one can see that the rank of attributes in the process of attribute reduction will remain unchanged after reducing the certain positive region. This mechanism can be used to improve the computational performance of a heuristic attribute reduction algorithm.

In a forward greedy attribute reduction approach, starting with the attribute with the maximal inner importance, we take the attribute with the maximal outer significance into the attribute subset in each loop until this attribute subset satisfies the stopping criterion, and then we can get an attribute reduction of a target decision. Formally, a forward greedy attribute reduction algorithm for searching a local attribute reduction with respect to a given target decision can be formulated as follows.

Algorithm 2. A forward greedy local attribute reduction algorithm for a target decision(LARD)

Input: An decision table $S = (U, C \cup D)$, labeled sample set $U_{lable} \subseteq U$, inclusion degree α and neighborhood size δ;

Output: *red*.

(1) $red \leftarrow \phi$.
(2) compute $Sig^{inner}(a_k, C, D, U), k \leq |C|$.
(3) $red \leftarrow a_k$, where $Sig^{inner}(a_k, C, D, U) = max(Sig^{inner}(a_k, C, D, U), a_k \in AT)$.
(4) $i \leftarrow 1, R_1 \leftarrow red, U_1 = U_{lable}, U_1/D = X_i^j, j \leq r;$.
(5) while $|\underline{POS_{red}}_{\alpha}^{U_i}(D)| < |\underline{POS_C}_{\alpha}^{U_i}(D)|$
$U_{i+1} = \cup_{x\in X} - CL_{red}^{U_i}(X_i)$;
$X_{i+1} = X_i - CL_{red}^{U_i}(X_i)$;
$red \leftarrow red \cup a_0$, where $Sig^{outer}(a_k, red, X_i, U_i) = max(Sig^{outer}(a_k, red, X_i, U_i), a_k \in C - red)$;
$R_i \leftarrow R_{i-1} \cup \{a_0\}$;
(6) return *red* and end

In order to conveniently compare, we denote the algorithm for finding an attribute reduction of a target decision in the context of global rough set by GARD.

In the above LARD algorithm, Step 1 needs to compute $|C|$ local lower approximations for r labeled class using the LLAC algorithm. Hence, its time complexity is $O(|C|(\sum_{j=1}^{r}|X^j|^2 + \sum_{j=1}^{r}|X^j||U|))$. However, the time complexity of this

Table 5: The time complexities of the LARD algorithm and the GARD algorithm

Algorithms	Step 2	Step 3	Step 5	Other steps
GARD	$O(\|C\|(\|U\|^2 + \sum_{j=1}^{r} \|X^j\|\|U\|))$	$O(\|C\|)$	$O(\sum_{i=1}^{\|C\|}(\|C\| - i + 1))(\|U\|^2 + \sum_{j=1}^{r} \|X^j\|\|U\|))$	Constant
LARD	$O(\|C\|(\sum_{j=1}^{r} \|X^j\|\|U\| + \sum_{j=1}^{r} \|X^j\|^2))$	$O(\|C\|)$	$O(\sum_{i=1}^{\|C\|}(\|C\| - i + 1))(\sum_{j=1}^{r} \|X_i^j\|\|U_i\| + \sum_{j=1}^{r} \|X_i^j\|^2))$	Constant

step in global rough set is $O(|C|(|U|^2 + \sum_{j=1}^{r} |X^j||U|))$. Step 3 needs to scan the $|C|$ attributes, hence, its time complexity is $O(|C|)$, and this step in global rough set has the same case. In Step 5, we add an attribute with the maximal significance into the set in each stage until finding a reduction. This process is called a forward reduction algorithm whose time complexity is $O(\sum_{i=1}^{|C|}(|C| - i + 1))(\sum_{j=1}^{r} |X_i^j|^2 + \sum_{j=1}^{r} |X_i^j||U_i|))$. However, the time complexity of this step in a classical heuristic algorithm is $O(\sum_{i=1}^{|C|}(|C| - i + 1))(|U|^2 + \sum_{j=1}^{r} |X^j||U|))$. Each of other steps of the LARD algorithm is constant. To stress these findings, the time complexity of each step in the LARD algorithm and the GARD algorithm is shown in Table 5.

It can be seen from Table 5 that the time complexity of the LARD algorithm is much lower than that of the GARD algorithm. Hence, one can draw a conclusion that the LARD algorithm may significantly reduce the computational time for attribute reduction of a target decision, which can efficiently work in the context of big data.

4.2.3. Experimental analysis

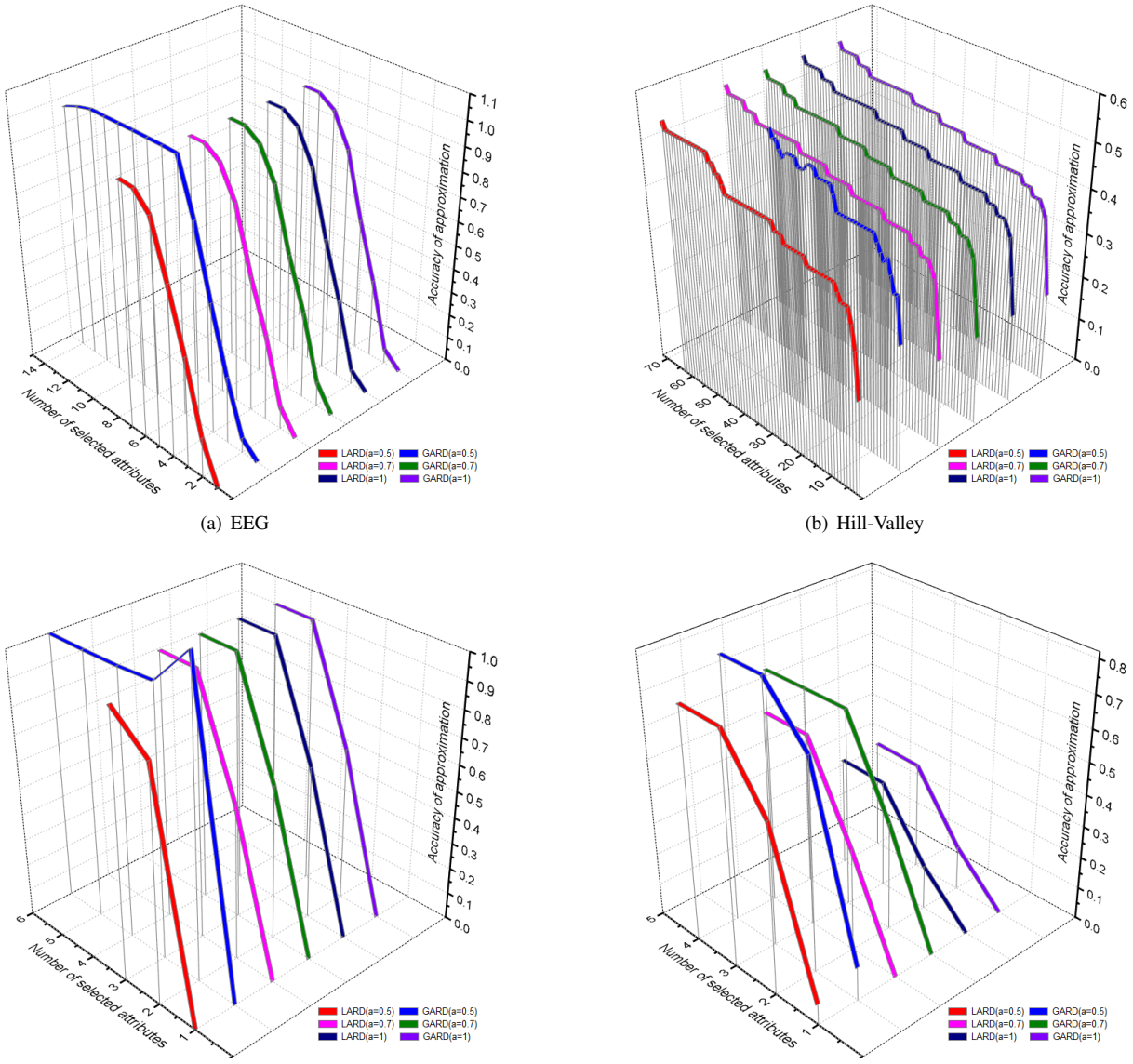
In this experimental analysis, we want to verify the advantages of the LARD algorithm from three aspects: monotonicity, efficiency and generality.

• Over-fitting issue

The over-fitting degree in attribute reduction can be observed by the monotonicity of positive regions of a target decision, which is often measured by the accuracy of approximation in Eq. (9). Given a decision table $S = (U, C \cup D)$, labeled sample set $U_{lable} \subseteq U$, and unlabeled sample set $U_{unlable} = U - U_{lable}$. $X_1, X_2, \cdots, X_r \in U_{lable}/D$ are r classes with labels, $P = \{N_1, N_2, \cdots, N_n\}$ a family of attributes with $N_1 \succeq N_2 \succeq \cdots \succeq N_n$ ($R_i \in 2^{AT}$). Let $P_i = \{N_1, N_2, \cdots, N_i\}$, we denote the accuracy of approximation of D with respect to P_i by $\gamma(P_i, D) = \frac{\sum\{|R_{P_i}(X)|:\, X \in U_{lable}/D\}}{\sum\{|X|:\, X \in U_{lable}/D\}} = \frac{|POS_{R_i}(D)|}{\sum\{|X|:\, X \in U_{lable}/D\}}$, where $X \in U_{lable}/D$ means each of those classes with labels. We use the same experiment settings. U_{lable} includes these samples from the first part, and $U_{unlable}$ consists of these samples from other parts.

If an attribute reduction algorithm possesses two properties that the quality of its approximation monotonically increases and the number of an attribute reduction induced by the algorithm is smaller. In general, we can say that the algorithm may be much better. In this experiment, we observe the two measures for the LARD algorithm and the GARD algorithm. Fig. 3 displays the quality of approximation of D used by each of the LARD algorithm and the GARD algorithm and shows it vis-a-vis the number of attributes (the x-coordinate and y-coordinate concern the number of the attributes and the value of precision of approximation, respectively). Table 6 shows the number of attributes in attribute reductions obtained by the LARD algorithm and the GARD algorithm.

Fig. 3 detailedly shows that the accuracy of approximation of the LARD algorithm monotonically increases with the increase of the number of attributes on these four data sets. However, the accuracy of approximation of the GARD algorithm does not always monotonically increases with the increase of the number of attributes. For instance, in Fig. 3(c) , we can obviously see that the accuracy of approximation of the GARD algorithm with $\alpha = 0.5$ is not monotonic as the number of attributes increases. We also can see from Table 6 that the number of attributes selected by the LARD algorithm is usually not larger than that selected by the GARD algorithm with the same parameter value on the same data set. The reason is that positive region of a target decision for global rough set may be beyond its own region, which leads to that these values of accuracy of approximation of the GARD algorithm may be larger than 1, which clearly causes over-fitting in attribute reduction. Based on these results, one can say that the LARD algorithm may effectively reduce the over-fitting degree in attribute reduction for a target decision.



(a) EEG (b) Hill-Valley

(c) Magic (d) Occupancy Detection

Figure 3: Accuracy of approximation of a target decision versus number of attributes

• Efficiency of the LARD algorithm

We still take the same settings as Section 4.1.2. to verity the efficiency of LARD algorithm. In this part, to compare computational time of LARD algorithm with the GARD algorithm vis-a-vis the size of universe, we take the front 10% objects from each of these nine data sets as its corresponding target decision. The experimental results are shown in Fig. 4 and Table 7. And this Figure displays more detailed tendency of computational time change of each algorithm with the increase of data set size. Table 7. shows that the computational time of attribute reductions of the same target decision using LARD and GARD algorithms on the nine data sets with different values of the parameter α. What excites us is that the LARD algorithm is consistently faster than the GARD algorithm on the same universe. Additionally, the computational time of the LARD algorithm is much smaller than that of the GARD algorithm. For

Table 6: The number of attributes in attribute reductions of LARD and GARD

Datasets	Original features	Selected features					
		LARD			GARD		
		$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 1$
EEG	14	7	8	8	8	8	8
Hill-Valley	100	63	71	41	71	71	71
Magic	10	3	4	4	6	4	4
Occupancy Detection	5	4	4	4	4	5	4

Table 7: The computational time for concept approximation with LARD and GARD

Datasets	$\alpha = 0.5$		$\alpha = 0.7$		$\alpha = 1$	
	LARD(s)	GARD(s)	LARD(s)	GARD(s)	LARD(s)	GARD(s)
EEG	1063.6885	28422.2514	1059.8917	17968.5868	1059.4966	18002.7213
Hill-Valley	284.5502	3051.7323	656.5356	6174.3999	317.5171	6313.0248
Magic	631.4789	12663.8582	869.9849	8473.5255	657.6703	8482.9844
Occupancy Detection	237.2134	3407.0483	226.5693	4000.121	234.7231	3402.2773

example, the LARD algorithm only takes 1/18 computational time of the GARD algorithm on the data set EEG when $\alpha = 1$. One can say that for computing the attribute reduction of a target decision, the LARD algorithm under the local rough set provides a very efficient solution in the context of limited labeled big data.

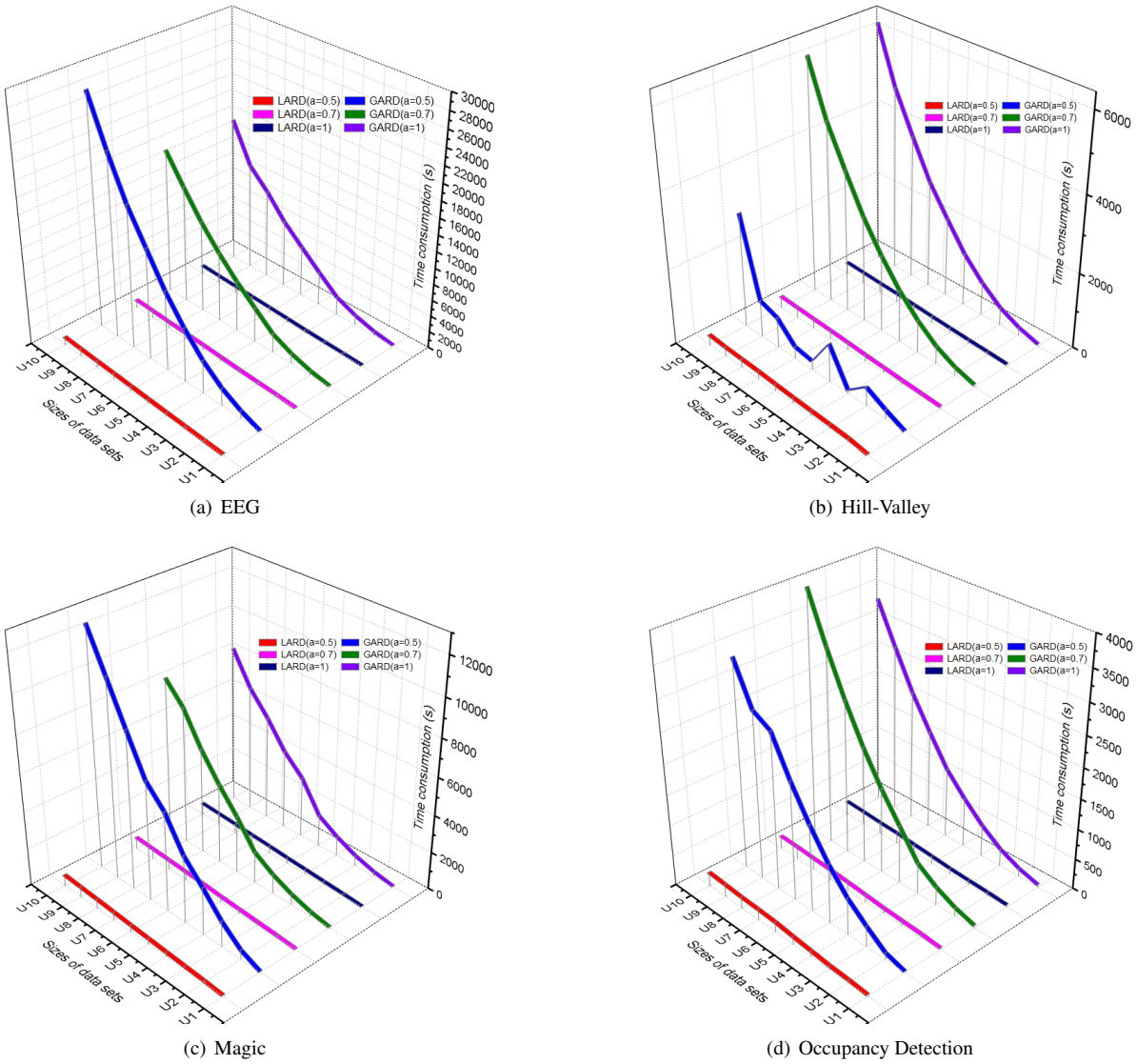
• Generalization of classifiers induced by attribute reduction of the LARD algorithm

The purpose of this experiment is to test classification quality induced by attribute reduction of the LARD algorithm comparing with those of the GARD algorithm. As we know, a given classifier must obtain the same classification accuracy on the same attribute reduction. Table 6 shows that almost the same attribute reduction is obtained by using these two algorithms when $\alpha = 0.7$ and 1.0 respectively. Hence, the corresponding classifiers have almost the same classification accuracy for a given classifier. However, these two algorithms would obtain different attribute reductions as the parameter α becoming much bigger. In the following, let $\alpha = 0.5$, all data are labeled, i.e. $U_{label} = U$. The results of attribute reduction with the LARD algorithm and the GARD algorithm and their classification accuracies are observed through employing SVM and KNN (coming from Weka 3.6.10, K=5), which are shown in Table 8.

In Table 8, $(\cdot)$ means the number of attributes in an attribute reduction obtained by an algorithm. It is easy to see from Table 8 that when $\alpha = 0.5$, the number of attributes in the attribute reduction induced by the LARD algorithm is consistently bigger than that induced by the GARD algorithm on the same data set. In addition, we can also see that for both classifiers SVM and KNN, the corresponding classification accuracies of classifiers induced by attribute reduction of the LARD algorithm are mostly higher than those of the GARD algorithm. This implies that one must select a reasonable α to obtain attribute reduction at least as good as the existing methods. Compared with the original attribute set, LARD algorithm can obtain comparable accuracy. However, the number of attribute sets in our algorithm is only about thirty percent to fifty percent of that of the original attribute set.

Table 8: Classification accuracies of classifiers induced by attribute reductions with LARD and GARD ($\alpha = 0.5$)

Datasets	SVM			KNN		
	Original	LARD	GARD	Original	LARD	GARD
EEG	0.5512 (14)	0.5511 (7)	0.5512 (1)	0.8379 (14)	0.8311 (7)	0.604 (1)
Hill-Valley	0.6204 (100)	0.6254 (55)	0.4834 (1)	0.5313 (100)	0.5215 (55)	0.49 (1)
Magic	0.7914 (10)	0.7842 (3)	0.7174 (1)	0.8364 (10)	0.8088 (3)	0.6771 (1)
Occupancy Detection	0.9884 (5)	0.9884 (4)	0.8262 (1)	0.9929 (5)	0.9931 (4)	0.8554 (1)



(a) EEG (b) Hill-Valley

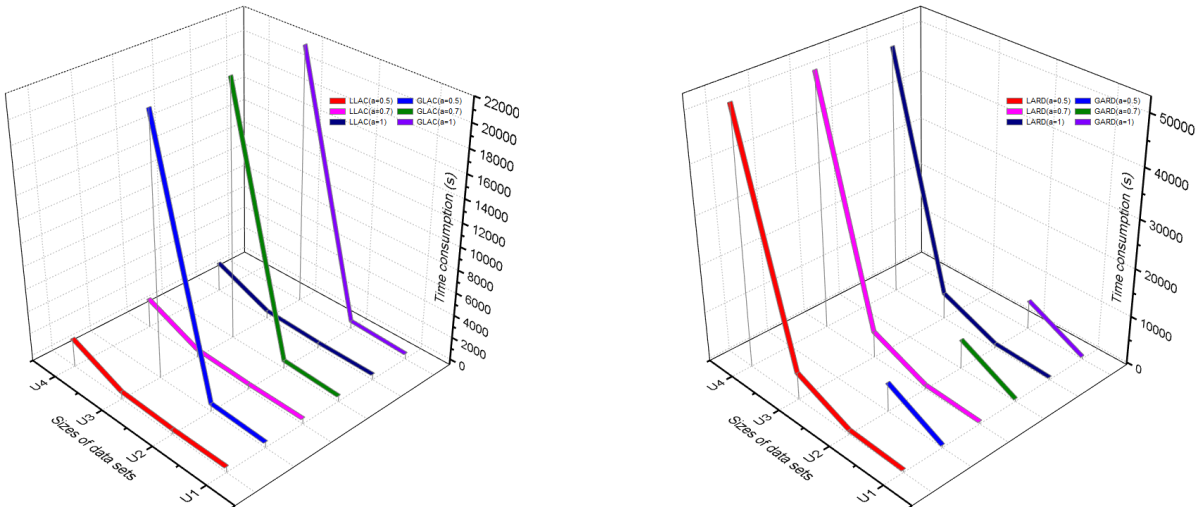
(c) Magic (d) Occupancy Detection

Figure 4: Time consumption of LARD and GARD versus the size of the universe

5. Scalability tests to big data

The purpose of this section is to test the scalability of the local rough set for rough data analysis on very large data sets. One large scale data set SUSY from UCI [[57], which has $1,000,000$ objects, 18 features and 2 classes, is used in this experiment. We obtain eight data sets by taking the front $U_1 = \frac{|U|}{10^3}$, $U_2 = \frac{|U|}{10^2}$, $U_3 = \frac{|U|}{10}$, and $U_4 = |U|$ from each of these two data sets. The number of labeled objects is set $\frac{|U|}{10^3}$, where $|U|$ is the size of the data set.

We tested scalability of the two algorithms (LLAC and LARD) in the local rough set on this large data set, which is the scalability against the number of objects for a given target concept and a given set of labeled objects. Fig. 5 (a) shows the computational time of the LLAC and GLAC algorithms calculating the lower approximations on different numbers of objects and Fig. 5 (b) shows those of the LARD and GARD algorithms for finding the attribute reductions of a given decision on different numbers of objects, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively. And we cannot get results of GLAC for U_4 and GARD for U_3 as well as U_4 with 24 hours.



(a) LLAC vs GLAC

(b) LARD vs GARD

Figure 5: Time consumption of LLAC, GLAC, LARD and GARD on SUSY

One important observation from these tables is that the run time of the two algorithms in the framework of the local rough set tends to increase linearly as the numbers of objects increase. For a given target concept/decision, the computational time of each algorithm on the data set with $\frac{|U|}{10^i}$ objects is almost 10 times longer than that of this algorithm on the data set with $\frac{|U|}{10^{i+1}}$ objects. This observation is consistent with the linear time complexity of each of these two algorithms in the local rough set. Hence, we can say that the proposed local rough set is an effective and efficient approach to rough data analysis in limited labeled big data.

6. Conclusions and further work

With the advent of the age of big data, the data scale becomes larger and larger, while labelling a massive amounts of data is an expensive, time-consuming and laborious task, sometimes even infeasible. As a supervised learning method, classical neighborhood rough set model mainly faces three issues for numerical big data analysis, including semi-supervised property of big data, computational inefficiency and over-fitting in attribute reduction. To solve the issues, we introduce local neighborhood rough set and design corresponding approximation and attribute reduction algorithms in this paper. we have verified the performance of these algorithms through employing nine real data sets and an artificial data set. The experimental analysis shows that the proposed local rough set and corresponding algorithms significantly improve three limitations of the global rough set. It is worth noting that the performances of the algorithms in the local rough set become more significant when analyzing huger data sets. Hence, the local rough set can be regarded as an effective solution to rough data analysis in big data. For further study, there are still many attractive and significant issues under the theoretical framework of local rough set model such as rough classifiers with semi-supervised learning and corresponding applications. These research results probably will be important to handle limited big data and may effectively promote the development of rough data analysis in big data in the future.

Acknowledgment

This study was supported by the National Natural Science Fund of China (Nos. 61672332, 61322211, 61432011, U1435212, 11671006 and 61603173), the Program for New Century Excellent Talents in University (No. NCET-12-1031), the Program for the Outstanding Innovative Teams of Higher Learning Institutions of Shanxi, the Program for the Young San Jin Scholars of Shanxi, and the National Key Basic Research and Development Program of China (973) (Nos. 2013CB329404 and 2013CB329502), the Key Science and Technology Program of Shanxi (No. MQ2014-09).

References

[1] Z. Pawlak, Rough sets: theoretical aspects of reasoning about data, Kluwer Academic Publishers.

[2] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 117 (1) (2007) 28–40.

[3] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 117 (1) (2007) 3–27.

[4] H. Zhao, P. Wang, Q. Hu, Cost-sensitive feature selection based on adaptive neighborhood granularity with multi-level confidence, Information Sciences 366 (2016) 134–149.

[5] H. Ju, H. Li, X. Yang, X. Zhou, B. Huang, Cost-sensitive rough set: A multi-granulation approach, Knowledge-Based Systems 123 (2017) 137–153.

[6] J. Liang, F. Wang, Y. Qian, C. Dang, A group incremental approach to feature selection applying rough set technique, IEEE Transactions on Knowledge and Data Engineering 26 (2) (2014) 294–308.

[7] Y. Yao, X. Zhang, Class-specific attribute reducts in rough set theory, Information Sciences 418 (2016) 601–618.

[8] X. Yue, Y. Chen, D. Miao, J. Qian, Tri-partition neighborhood covering reduction for robust classification, International Journal of Approximate Reasoning 83 (2017) 371–384.

[9] R. W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognition Letters 24 (2003) 833–849.

[10] J. Wei, S. Wang, X. Yuan, Ensemble rough hypercuboid approach for classifying cancers, IEEE Transactions on Knowledge and Data Engineering 22 (3) (2010) 381–391.

[11] Y. She, X. He, H. Shi, Y. Qian, A multiple-valued logic approach for multigranulation rough set model, International Journal of Approximate Reasoning 82 (2017) 270–284.

[12] T. Feng, J. Mi, Variable precision multigranulation decision-theoretic fuzzy rough sets, Knowledge-Based Systems 91 (2016) 93–101.

[13] Y. Qian, H. Cheng, J. Wang, J. Liang, W. Pedrycz, C. Dang, Grouping granular structures in human granulation intelligence, Information Sciences 382–383 (2017) 150–169.

[14] Y. Qian, J. Liang, Y. Yao, C. Dang, Mgrs: a multi-granulation rough set, Information Sciences 180 (6) (2010) 949–970.

[15] Y. Sang, J. Liang, Y. Qian, Decision-theoretic rough sets under dynamic granulation, Knowledge-Based Systems 91 (2016) 84–92.

[16] Y. Chen, X. Yue, H. Fujita, S. Fu, Three-way decision support for diagnosis on focal liver lesions, Knowledge-Based Systems 127 (2017) 85–99.

[17] B. Huang, H. Li, G. Feng, Y. Zhuang, Inclusion measure-based multi-granulation intuitionistic fuzzy decision-theoretic rough sets and their application to issa, Knowledge-Based Systems 138 (2017) 220–231.

[18] D. Liu, D. Liang, C. Wang, A novel three-way decision model based on incomplete information system, Knowledge-Based Systems 91 (2016) 32–45.

[19] W. Wu, Y. Qian, T. Li, S. Gu, On rule acquisition in incomplete multi-scale decision tables, Information Sciences 378 (2016) 282–302.

[20] H. Li, L. Zhang, X. Zhou, B. Huang, Cost-sensitive sequential three-way decision modeling using a deep neural network, International Journal of Approximate Reasoning 85 (2017) 68–78.

[21] J. Hu, T. Li, H. Wang, H. Fujita, Hierarchical cluster ensemble model based on knowledge granulation, Knowledge-Based Systems 91 (2016) 179 – 188.

[22] X. H. Hu, N. Cercone, Learning in relational databases: a rough set approach, International Journal of Computational Intelligence 11 (2) (1995) 323–338.

[23] Y. Qian, X. Liang, G. Lin, Q. Guo, J. Liang, Local multigranulation decision-theoretic rough sets, International Journal of Approximate Reasoning 82 (2017) 119–137.

[24] R. Slowinski, D. Vanderpooten, A generalized definition of rough approximations based on similarity, IEEE Transactions on Knowledge and Data Engineering 12 (2) (2000) 331–336.

[25] H. Dou, X. Yang, X. Song, H. Yu, W. Wu, J. Yang, Decision-theoretic rough set: A multicost strategy, Knowledge-Based Systems 91 (2016) 71–83.

[26] M. Kryszkiewicz, P. Lasek, Fun: fast discovery of minimal sets of attributes functionally determining a decision attribute, Transactions on Rough Sets 9 (1) (2008) 41–73.

[27] S. Xu, X. Yang, H. Yu, D. Yu, J. Yang, E. C. Tsang, Multi-label learning with label-specific feature reduction, Knowledge-Based Systems 104 (2016) 52–61.

[28] Y. Yao, Y. Zhang, Attribute reduction in decision-theoretic rough set models, Information Sciences 178 (17) (2008) 3356–3373.

[29] Y. Huang, T. Li, C. Luo, H. Fujita, S. Horng, Dynamic variable precision rough set approach for probabilistic set-valued information systems, Knowledge-Based Systems 122 (2017) 131–147.

[30] Y. Qian, J. Liang, W. pedrycz, C. Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artifical Intelligence 174 (9) (2010) 597–618.

[31] R. Jensen, Q. Shen, A distance measure approach to exploring the rough set boundary region for attribute reduction, IEEE Transactions on Knowledge and Data Engineering 22 (3) (2007) 305–317.

[32] K. Kira, L. A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: AAAI, 1992, pp. 129–134.

[33] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.

[34] M. Z. Li, B. Yu, Z. D. W. O. Rana, Grid service discovery with rough sets, IEEE Transactions on Knowledge and Data Engineering 20 (6) (2008) 851–862.

[35] Q. Hu, Z. X. Xie, D. R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (2007) 3509–3521.

[36] H. Liu, R. Setiono, Feature selection via discretization, IEEE Transactions on Knowledge and Data Engineering 9 (4) (1997) 642–645.

[37] R. Quinlan, Induction of decision rules, Information Sciences 1 (1) (1986) 81–106.

[38] T. Y. Lin, K. J. Huang, Q. Liu, Rough sets, neighborhood systems and approximation, in: Proceedings of the Fifth International Symposium on Methodologies of Intelligent Systems, Vol. 22, 1990, pp. 130–141.

[39] Y. Yao, Three-way decisions with probabilistic rough sets, Information Sciences 180 (2010) 341–353.

[40] Y. Yao, The superiority of three-way decisions in probabilistic rough set models, Information Sciences 181 (2011) 1080–1096.
[41] X. Yang, T. Li, H. Fujita, D. Liu, Y. Yao, A unified model of sequential three-way decisions and multilevel incremental processing, Knowledge-Based Systems 134 (2017) 172–188.
[42] Y. Jing, T. Li, H. Fujita, Z. Yu, B. Wang, An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view, Information Sciences 411 (2017) 23–38.
[43] C. D. Maio, G. Fenza, V. Loia, F. Orciuoli, Making sense of cloud-sensor data streams via fuzzy cognitive maps and temporal fuzzy concept analysis, Neurocomputing 256 (2017) 35–48.
[44] C. D. Maio, G. Fenza, V. Loia, F. Orciuoli, Unfolding social content evolution along time and semantics, Future Generation Computer Systems 66 (2017) 146–159.
[45] H. Wang, Nearest neighbors by neighborhood counting, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 942–953.
[46] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.
[47] A. BenDavida, L. Stering, T. Tran, Adding monotonicity to learning algorithms may impair their accuracy, Expert Systems with Applications 36 (2009) 6627–6634.
[48] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences 46 (1993) 39–59.
[49] D. Slezak, W. Ziarko, The investigation of the bayesian rough set model, International Journal of Approximate Reasoning 40 (2005) 381–91.
[50] J. T. Yao, Y. Y. Yao, W. Ziarko, Probabilistic rough sets: approximations, decision-making and applications, International Journal of Approximate Reasoning 49 (4) (2008) 253–254.
[51] Y. Yao, Probabilistic rough set approximations, International Journal of Approximate Reasoning 49 (2008) 255–271.
[52] Y. Qian, X. Liang, Q. Wang, J. Liang, B. Liu, A. Skowron, Y. Yao, J. Ma, C. Dang, Local rough set: a solution to rough data analysis in big data, International Journal of Approximate Reasoning, Accepted.
[53] Q. Hu, D. R. Yu, J. F. Liu, C. X. Wu, Neighborhood rough set based heterogeneous feature subset selection, Information Sciences 178 (18) (2008) 3577–3594.
[54] W. X. Zhang, Y. Leung, Theory of including degrees and its applications to uncertainty inferences, International Journal of Approximate Reasoning (1996) 496–501.
[55] I. D. G. Gediga, Rough approximation quality revisited, Artificial Intelligence 132 (2001) 219–234.
[56] Y. Qian, J. Liang, W. pedrycz, C. Y. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, Pattern Recognition 44 (8) (2011) 1658–1670.
[57] M. Lichma, UCI machine learning repository, `http://archive.ics.uci.edu/ml`, university of California, Irvine, School of Information and Computer Sciences (2013).