Local rough set: a solution to rough data analysis in big data

Yuhua Qian [*] [†a,b,c], Xinyan Liang[†a,b,c], Qi Wang[†a,b,c], Jiye Liang[b], Bing Liu[d], Andrzej Skowron[e,f], Yiyu Yao[g], Jianmin Ma[h], Chuangyin Dang[i]

[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*
[d]*Department of Computer Science, University of Illinois at Chicago, Illinois, USA*
[e]*University of Warsaw, Warsaw, Poland*
[f]*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*
[g]*Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada*
[h]*Department of Mathematics and Information Science, Faculty of Science, Chang'an University, Shaan'xi, China*
[i]*Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong*

Abstract

As a supervised learning method, classical rough set theory often requires a large amount of labeled data, in which concept approximation and attribute reduction are two key issues. With the advent of the age of big data however, labeling data is an expensive and laborious task and sometimes even infeasible, while unlabeled data are cheap and easy to collect. Hence, techniques for rough data analysis in big data using a semi-supervised approach, with limited labeled data, are desirable. Although many concept approximation and attribute reduction algorithms have been proposed in the classical rough set theory, quite often, these methods are unable to work well in the context of limited labeled big data. The challenges to classical rough set theory can be summarized with three issues: limited labeled property of big data, computational inefficiency and over-fitting in attribute reduction. To address these three challenges, we introduce a theoretic framework called local rough set, and develop a series of corresponding concept approximation and attribute reduction algorithms with linear time complexity, which can efficiently and effectively work in limited labeled big data. Theoretical analysis and experimental results show that each of the algorithms in the local rough set significantly outperforms its original counterpart in classical rough set theory. It is worth noting that the performances of the algorithms in the local rough set become more significant when dealing with larger data sets.

Keywords: Rough set theory; Local rough set; Concept approximation; Attribute reduction; Limited labeled data

1. Introduction

Rough set theory, originated by Pawlak [26], has become an effective tool for uncertainty management and uncertainty reasoning, and has a wide variety of applications in artificial intelligence [6, 12, 27, 28, 32]. One of the strengths of rough set theory is that all its parameters are obtained from a given sample set. This can be seen in the following paragraph from [26]: "The numerical value of imprecision is not pre-assumed, as it is in probability theory or fuzzy sets, but is calculated on the basis of approximations which are the fundamental concepts used to express imprecision of knowledge". In other words, instead of using, the rough data analysis (RSDA) utilizes solely the granular structure of the given sample set, expressed as classes of suitable binary relations. To date, rough data analysis has been widely applied in feature selection [2, 5, 13, 52] pattern recognition [42], uncertainty reasoning [38, 45], granular computing

[*] Corresponding author.
[†] Co-first author.

Email addresses: jinchengqyh@126.com (Yuhua Qian), liangxinyan48@163.com (Xinyan Liang), counter_king@163.com (Qi Wang), ljy@sxu.edu.cn (Jiye Liang), liub@cs.uic.edu (Bing Liu), skowron@mimuw.edu.pl (Andrzej Skowron), yyao@cs.uregina.ca (Yiyu Yao), majianmin@chd.edu.cn (Jianmin Ma), mecdang@cityu.edu.hk (Chuangyin Dang)

[28, 31, 32, 36, 54], data mining and knowledge discovery [22, 23, 33, 39, 46]. Over the past 30 years, we have witnessed the rapid development of rough set theory.

In rough set theory, concept approximation and attribute reduction are two key issues. Concept approximation includes two terms: lower approximation and upper approximation. Given a sample set U and a binary relation R, one can generate a granular structure U/R. Through using information granules from the granular structure, one can construct a rough set $\langle$*lower approximation*, *upper approximaton*$\rangle$ of any subset on the sample set. Feature selection in rough set theory is called attribute reduction, which is a common problem in pattern recognition, data mining and machine learning. Attributes that are irrelevant to recognition tasks can be omitted, which will not seriously impact the resulting classification (recognition) error, cf. [9, 10, 24]. In particular, the reduction of some attributes could not only be tolerable but even desirable relatively to the costs involved in such cases [25]. Attribute reduction in rough set theory offers a systematic theoretic framework for consistency-based feature selection, which does not attempt to maximize the class separability [14] but rather attempts to retain the discernible ability of original features for the objects from the universe [34, 35, 42]. For a given set of objects with class labels, classical rough set theory can be used to extract some decision rules through using its attribute reducts. This set of decision rules is called a rough classifier, which can be used to predict the class label of an unseen object. From this viewpoint, classical rough set theory could be regarded as a supervised learning method.

In order to generally study, we combine the Pawlak's rough set [26] and the decision-theoretic rough set [50, 49] into the same rough set model, as one representative, in this paper. Let (U, R) be an approximation space with R being an equivalence relation on U, and $U/R = \{[x]_R : x \in U\}$ the set of all equivalence classes generated by the equivalence relation R. Then for any $X \subseteq U$, the lower and upper approximations of the set X are defined by

$$\begin{cases} \underline{R}_{(\alpha,\beta)}(X) = \{x \mid P(X \mid [x]_R) \geq \alpha,\ x \in U\}, \\ \overline{R}_{(\alpha,\beta)}(X) = \{x \mid P(X \mid [x]_R) > \beta,\ x \in U\}, \end{cases} \tag{1}$$

respectively, where $P(\cdot)$ is a conditional function and α, β are two parameters from the decision-theoretic rough set.

From the definition of the generalized rough set above, we can see that the computation of its lower/upper approximation needs to scan all objects in a given universe, in which information granules to approximate a target concept must be beforehand obtained. Conveniently, we call this kind of rough sets *global rough sets*. In this paper, we call the $\underline{R}_{(\alpha,\beta)}(X)$ the global lower approximation of X and the $\overline{R}_{(\alpha,\beta)}(X)$ the global upper approximation of X, respectively.

However, with the rapid development of big data, the scale of a data set (number of objects and/or number of attributes) has become bigger and bigger. In spite of existing rough set models being very successful for rough data analysis, they still encounter some challenges and cannot effectively and efficiently analyze a large-scale data set. In what follows, we analyze these challenges one by one. These challenges are the main motivations for this study.

(a) Semi-supervised property of big data

In many real-world classification tasks and recognition problems, the state-of-the-art algorithms focus on training classifiers or regressors from a given training set. With the advent of the age of big data, the number of obtained objects becomes larger and larger. Machine learning in big data has been an important research area. However, rough set-based 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. In the context of big data, a data set to deal with could be represented as a data table shown in Table 1. In classical rough set theory however, we only use the data set $\{x_1, x_2, \cdots, x_p\}$ as a training set with class labels $\{d_1, d_2, \cdots, d_r\}$, which wastes a wealth of information provided by unlabeled data. Hence, techniques to automatically learn rough classifiers from big data in an semi-supervised way, with limited labeled data, are desirable. This is one motivation of rough data analysis in big data.

(b) Computational inefficiency

It is well known that, many concept approximation and attribute reduction algorithms have been proposed however, quite often, these methods are computationally time-consuming. For any rough set model induced by a binary relation, its lower/upper approximation is constructed based on beforehand computations of granular structures in which information granules are used to approximate a target concept [12, 19, 20, 29, 34, 43, 44]. While the computations of these information granules must scan all objects in a given universe. The time complexity of computing all

Table 1: A data table with limited labeled objects

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

information granules is $O(n^2)$ without pre-ranking of objects and $O(n \log n)$ with pre-ranking, where n is the number of objects from a given universe. However, this computation is still computationally time-consuming, which cannot satisfy the requirement of efficient computation to big data. This means that the large amount of data also challenges conventional concept approximation and attribute reduction approaches, in terms of how the algorithms can be scalable and efficient. This is the second motivation of this study.

(c) Over-Fitting in attribute reduction

For attribute reduction, in fact, we also need to consider robustness and sensitivity for noisy data. As we know, noise has great influence on modeling classification tasks [1], specially rough classifiers induced by an attribute reduct. If the measures used to evaluate significance of attributes in attribute reduction are sensitive to noisy objects, the performance of the trained classifier would be weak. To solve this issue, some extended versions of rough sets have been developed, which include variable precision rough set [55], decision theoretic rough set [49, 50], Bayesian rough set [17], Probabilistic rough set [48, 51], and so on. Each of these rough sets can be used to control the degree of uncertainty, misclassification and imprecise information. Comparing it with Pawlak rough set model, however, we can find that for these rough sets, lower/upper approximations of a target concept with the number of attribute increasing are often not monotonic, in which objects outside this target concept may be included. Hence, an attribute reduct obtained might have much more attributes, which could lead to an over-fitting problem and a weaker classifier. How to ensure the monotonicity of an attribute reduction process is also a motivation of this study.

To address these three challenges, through reviewing existing rough set models, this paper first presents a new rough set model for rough data analysis in big data, called local rough set, in which the computations of lower/upper approximations do not obtain information granules of all objects in advance, but only calculate those of objects within a target concept. Some interesting properties and measures in the local rough set will also be given. Under the framework of the local rough set, then we propose the LLAC algorithm for computing a local lower approximation of a target concept and the LARC algorithm for searching a local attribute reduct of a target concept. Moreover, based on the local rough set, we design the LLAD algorithm for calculating a local lower approximation of a target decision and the LARD algorithm for finding a local attribute reduct of a target decision. Unlike the corresponding classical algorithms in the global rough set, the time complexity of each of these four algorithms is linear, which can be well used for rough data analysis in big data. To highlight advantages of the local rough set, we also employ nine real data sets and an artificial data set for verifying the performance of these four algorithms. Experiment results show that the proposed local rough set and corresponding algorithms can significantly improve on two limitations of the global rough set for rough data analysis in big data.

The study is organized as follows. Several existing rough models in rough set theory are briefly reviewed and discussed in Section 2. In Section 3, we establish the local rough set model and investigate some of its main properties and measures. In Section 4, we aim to solve the problem of how to approximate a target concept and that of how to search an attribute reduct of a target concept in the framework of the local rough set. Section 5 contributes to solve the problem of how to approximate a target decision and that of how to find an attribute reduct of a target decision. Section 6 employs an artificial large-scale data set to verify scalability of the local rough set. Experiments in Sections 4-6 show that the algorithms in the local rough set outperform their original counterparts in the global rough set. Finally, Section 7 concludes this paper by bringing some remarks and discussions.

2. Existing rough set models

In this section, we review several representative rough set models in rough set theory. From the perspective of considering decision risk or not, the existing rough set models are divided into two classes: rough set models without decision risk and rough set models with decision risk. We use two subsections to review them in this section, respectively. Throughout this paper, we suppose that the universe U is a finite nonempty set.

2.1. Rough set models without decision risk

Rough set theory has become a well-established mechanism for uncertainty management. In the past 10 years, several extensions of the rough set model have been proposed in terms of various binary relations, such as tolerance rough set [15, 16], dominance rough set [7], neighborhood rough set [11], fuzzy rough set [4, 41], and so on. All these models do not take decision risk into account.

As we know, the lower/upper approximation in every rough set model is constructed by some existing information granules, while an information granule is a clump of objects drawn together by indistinguishability, similarity, connectivity and proximity of functionality. Granulation of an object set by a binary relation leads to a collection of granules. In this subsection, we mainly review Pawlak's rough set model, as one representative of proven rough set models based on various binary relations without decision risk.

The earliest rough set model originated by Pawlak [26], which is based on an equivalence relation. Let U be a finite and non-empty set called the universe and $R \subseteq U \times U$ an equivalence relation on U. Then $K = \langle U, R\rangle$ is called an approximation space [26]. The equivalence relation R partitions the set U into disjoint subsets. This partition of the universe is called a quotient set induced by R, denoted by U/R. If two objects $x, y \in U$, $(x \neq y)$ belong to the same equivalence class, we say that x and y are indistinguishable under the equivalence relation R, i.e., they are equal in R. We denote the equivalence class including x by $[x]_R$. Each equivalence class $[x]_R$ may be viewed as an information granule consisting of indistinguishable objects [22].

Given an approximation space $K = \langle U, R\rangle$ and an arbitrary subset $X \subseteq U$, one can construct a rough set of the set on the universe by elemental information granules in the following definition:

$$\begin{cases} \underline{R}(X) = \cup\{[x]_R \mid [x]_R \subseteq X, x \in U\}, \\ \overline{R}(X) = \cup\{[x]_R \mid [x]_R \cap X \neq \emptyset, x \in U\}, \end{cases} \tag{2}$$

where $\underline{R}(X)$ and $\overline{R}(X)$ are called lower approximation and upper approximation with respect to R, respectively. The order pair $\langle \underline{R}(X),\ \overline{R}(X)\rangle$ is called a rough set of X with respect to the equivalence relation R. Equivalently, they also can be written as [26]

$$\begin{cases} \underline{R}(X) = \{x \mid [x]_R \subseteq X, x \in U\}, \\ \overline{R}(X) = \{x \mid [x]_R \cap X \neq \emptyset, x \in U\}. \end{cases} \tag{3}$$

The Pawlak's rough set has the following interesting properties: for any $X, Y \subseteq U$,

(1) $\underline{R}(\emptyset) = \overline{R}(\emptyset) = \emptyset,\ \underline{R}(U) = \overline{R}(U) = U$;

(2) $\underline{R}(X) \subseteq X \subseteq \overline{R}(X)$,

(3) $\underline{R}(X \cap Y) = \underline{R}(X) \cap \underline{R}(Y), \overline{R}(X \cup Y) = \overline{R}(X) \cup \overline{R}(Y)$;

(4) $X \subseteq Y \Rightarrow \underline{R}(X) \subseteq \underline{R}(Y),\ \overline{R}(X) \subseteq \overline{R}(Y)$;

(5) $\underline{R}(X \cup Y) \supseteq \underline{R}(X) \cup \underline{R}(Y), \overline{R}(X \cap Y) \subseteq \overline{R}(X) \cap \overline{R}(Y)$;

(6) $\underline{R}(\sim X) =\sim \overline{R}(X),\ \overline{R}(\sim X) =\sim \underline{R}(X)$;

(7) $\underline{R}(\underline{R}(X)) = \overline{R}(\underline{R}(X)) = \underline{R}(X),\ \underline{R}(\overline{R}(X)) = \overline{R}(\overline{R}(X)) = \overline{R}(X)$.

There exists a partial relation $\preceq$ on the family of equivalence relations $\{R :\ R \in \mathbf{R}\}$ as follows: $P \preceq Q$ (or $Q \succeq P$) if and only if, for every $P_i \in U/P$, there exists $Q_j \in U/Q$ such that $P_i \subseteq Q_j$, where $U/P = \{P_1, P_2, \cdots, P_m\}$ and $U/Q = \{Q_1, Q_2, \cdots, Q_n\}$ are partitions induced by $P,\ Q \in \mathbf{R}$, respectively [37]. In this case, we say that Q is coarser than P, or P is finer than Q. If $P \preceq Q$ and $U/P \neq U/Q$, we say Q is strictly coarser than P (or P is strictly finer than Q), denoted by $P \prec Q$ (or $Q \succ P$). It becomes clear that $P \prec Q$ if and only if, for every $X \in U/P$, there exists $Y \in U/Q$ such that $X \subseteq Y$, and there exists $X_0 \in U/P$, $Y_0 \in U/Q$ such that $X_0 \subset Y_0$.

2.2. Rough set models with decision risk

In order to consider the decision risk [48] or enhance the robustness (to noisy data) [55] of a rough-set-based decision, many probabilistic approaches to rough sets with decision risk have been proposed, such as decision-theoretic rough set model [48, 51, 18], variable precision rough set model [55], Bayesian rough set model [17], and others. A fundamental difficulty with the probabilistic, variable precision, and parameterized approximations is the physical interpretation of the required threshold parameters, as well as systematic methods for setting the parameters. This difficulty has in fact been resolved in the decision-theoretic model of rough sets proposed by Yao [48, 49, 50, 51]. In this subsection, simply, we only review the decision-theoretic rough set model with an equivalence relation.

In the Bayesian decision procedure, a finite set of states can be written as $\Omega = \{\omega_1, \omega_2, \cdots, \omega_s\}$, and a finite set of m possible actions can be denoted by $A = \{a_1, a_2, \cdots, a_r\}$. Let $P(\omega_j|\mathbf{x})$ be the conditional probability of an object x being in state ω_j given that the object is described by $\mathbf{x}$. Let $\lambda(a_i|\omega_j)$ denote the loss, or cost, for taking action a_i when the state is ω_j, the expected loss associated with taking action a_i is given by

$$R(a_i|\mathbf{x}) = \sum_{j=1}^{s} \lambda(a_i|\omega_j)P(\omega_j|\mathbf{x}).$$

In the classical rough set theory, the approximation operators partition the universe into three disjoint classes $POS(A)$, $NEG(A)$, and $BND(A)$. Through using the conditional probability $P(X|[x])$, the Bayesian decision procedure can decide how to assign x into these three disjoint regions. The expected loss $R(a_i|[x])$ associated with taking the individual actions can be expressed as

$$\begin{aligned} R(a_1|[x]) &= \lambda_{11}P(X|[x]) + \lambda_{12}P(X^c|[x]),\\ R(a_2|[x]) &= \lambda_{21}P(X|[x]) + \lambda_{22}P(X^c|[x]),\\ R(a_3|[x]) &= \lambda_{31}P(X|[x]) + \lambda_{32}P(X^c|[x]), \end{aligned}$$

where $\lambda_{i1} = \lambda(a_i|X)$, $\lambda_{i2} = \lambda(a_i|X^c)$, and $i = 1, 2, 3$. When $\lambda_{11} \leq \lambda_{31} < \lambda_{21}$ and $\lambda_{22} \leq \lambda_{32} < \lambda_{12}$, the Bayesian decision procedure leads to the following minimum-risk decision rules:

(P) If $P(X|[x]) \geq \gamma$ and $P(X|[x]) \geq \alpha$, decision $POS(X)$;
(N) If $P(X|[x]) \leq \beta$ and $P(X|[x]) \leq \gamma$, decision $NEG(X)$;
(B) If $\beta \leq P(X|[x]) \leq \alpha$, decide $BND(X)$;

where

$$\begin{aligned} \alpha &= \tfrac{\lambda_{12}-\lambda_{32}}{(\lambda_{31}-\lambda_{32})-(\lambda_{11}-\lambda_{12})},\\ \gamma &= \tfrac{\lambda_{12}-\lambda_{22}}{(\lambda_{21}-\lambda_{22})-(\lambda_{11}-\lambda_{12})},\\ \beta &= \tfrac{\lambda_{32}-\lambda_{22}}{(\lambda_{21}-\lambda_{22})-(\lambda_{31}-\lambda_{32})}. \end{aligned}$$

If a loss function with $\lambda_{11} \leq \lambda_{31} < \lambda_{21}$ and $\lambda_{22} \leq \lambda_{32} < \lambda_{12}$ further satisfies the condition:

$$(\lambda_{12} - \lambda_{32})(\lambda_{21} - \lambda_{31}) \geq (\lambda_{31} - \lambda_{11})(\lambda_{32} - \lambda_{22}),$$

then $\alpha \geq \gamma \geq \beta$.

When $\alpha > \beta$, we have $\alpha > \gamma > \beta$. The decision-theoretic rough set has the decision rules:

(P1) If $P(X|[x]) \geq \alpha$, decide $POS(X)$;
(N1) If $P(X|[x]) \leq \beta$, decide $NEG(X)$;
(B1) If $\beta < P(X|[x]) < \alpha$, decide $BND(X)$.

Using these three decision rules, we get the probabilistic approximation [48], [50]:

$$\begin{cases} \underline{apr}_{(\alpha,\beta)}(X) = \{x \mid P(X|[x]) \geq \alpha,\ x \in U\}, \\ \overline{apr}_{(\alpha,\beta)}(X) = \{x \mid P(X|[x]) > \beta,\ x \in U\}. \end{cases} \tag{4}$$

In the framework of decision-theoretic rough sets, the Pawlak rough set model, the variable precision rough set model, the Bayesian rough set model and the 0.5-probabilistic rough set model can be pooled together and studied based on the notions of conditional functions.

3. Local rough set

In order to generally study, as the last section mentioned, we combine the Pawlak's rough set and the decision-theoretic rough set into the same rough set model, as one representative in the kind of global rough sets, in this paper. From the definition of the generalized rough sets, we know that both concept approximation and attribute reduction (two key issues in rough set theory) are computationally time-consuming for many data sets, especially big data. In order to promote effective applications of rough sets in big data, the objective of this study is to propose a novel and general rough set framework with efficiently computational performance.

3.1. Construction of a local rough set and its properties

In order to overcome the three limitations mentioned above, we develop a new rough set framework in this subsection and investigate some of its important properties.

From the definition of a decision-theoretic rough set, it can be seen that the conditional function $P(\cdot)$ needs to be given first. In fact, this conditional function has various forms [30, 40, 48]. In general, the function $P(\cdot)$ is often depicted as an inclusion degree with the following constraints.

Let $(U, \leq)$ be a partial set with $\leq$ being a partial order on the universe of discourse U. Then for any $x, y \in U$, there exists a corresponding number $\mathcal{D}(y/x)$ satisfying

(1) $0 \leq \mathcal{D}(y/x) \leq 1$,

(2) $x \leq y \Rightarrow \mathcal{D}(y/x) = 1$, and

(3) $x \leq y \leq z \Rightarrow \mathcal{D}(x/z) \leq \mathcal{D}(x/y)$,

where $\mathcal{D}$ is referred to as the degree of inclusion [53].

For example, Let U be a given universe, $\mathcal{P}(U)$ the power set of U, P a probability distribution on U, and $\subseteq$ an inclusion relation of sets. Then, $(\mathcal{P}(U), \subseteq)$ is a partial set. It is should be noted that $\forall E, F \in \mathcal{P}(U)$, if $\mathcal{D}(F/E) = \frac{|E \cap F|}{|E|}$ or $\mathcal{D}(F/E) = \frac{P(E \cap F)}{P(E)}$, then $\mathcal{D}$ is an inclusion degree on $\mathcal{P}(U)$. By (3) of the above definition, we can get that for any $X, Y, Z \subseteq U$,

$$X \subseteq Y \Rightarrow \mathcal{D}(X/Z) \leq \mathcal{D}(Y/Z). \tag{5}$$

Without loss of generality, we select $\mathcal{D}(F/E) = \frac{|E \cap F|}{|E|}$ as the conditional function in the next investigation of this paper.

From three limitations of global rough sets, as Section 1 mentioned, we know that all information granules are firstly computed through comparing the difference among any two objects from a given data set. This implies that a global rough set must observe the relationship between a target concept and each of information granules. However, this is not a good strategy for approximating a target concept $X \subseteq U$. In fact, the information granules $\{[x] : [x] \cap X = \emptyset, \ x \in U\}$ are useless for computing the lower/upper approximation of X. It is unnecessary time-wasting to compute them. In other words, we only need to calculate those information granules relating to the target concept X, which can significantly reduce computational time consumption for concept approximation. In particular, for a very large-scale data set, one often has that $n \gg |X|$, where n and $|X|$ are the number of this data set and that of X, respectively. This time-reduction performance will be very exciting for rough data analysis on big data.

Based on the consideration above, to efficiently apply rough set theory to big data, we can re-construct the rough set model as follows.

Definition 1. *Let (U, R) be an approximation space, and $\mathcal{D}$ 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*

$$\begin{cases} \underline{R}_{\alpha}(X) = \{x \mid \mathcal{D}(X/[x]_R) \geq \alpha, \ x \in X\}, \\ \overline{R}_{\beta}(X) = \cup\{[x]_R \mid \mathcal{D}(X/[x]_R) > \beta, \ x \in X\}. \end{cases} \tag{6}$$

The pair $\langle \underline{R}_{\alpha}(X), \overline{R}_{\beta}(X) \rangle$ is called a local rough set.

Table 2: A toy information system comprising data with limited labels

Project	Locus	Investment	Decision
x_1	Good	Very High	Yes
x_2	Good	Very High	Yes
x_3	Good	High	*
x_4	Common	Medium	No
x_5	Common	Medium	*
x_6	Common	Medium	No
x_7	Bad	Low	No
x_8	Bad	Very low	*

The boundary of X is denoted by $BN_R(X) = \overline{R}_\beta(X) - \underline{R}_\alpha(X)$, call the local boundary region of X.

It deserves to point out that, when $\alpha = 1$ and $\beta = 0$, the above local rough set will degenerate to a Pawlak's rough set. That is

$$\underline{R}_\alpha(X) = \{x \mid \mathcal{D}(X/[x]_R) \geq 1,\ x \in X\} = \{x \mid \{[x]_R \subseteq X,\ x \in U\}\} = \underline{R}(X),$$
$$\overline{R}_\beta(X) = \cup\{[x]_R \mid \mathcal{D}(X/[x]_R) > 0,\ x \in X\} = \{x \mid \{[x]_R \cap X \neq \emptyset,\ x \in U\}\} = \overline{R}(X).$$

In other words, the local rough set model will have the same form and semantics as the original Pawlak's rough set model (it can be seen as a type of global rough sets). From this point of view, this re-construction of a rough set does not change the idea of rough set theory originally proposed by Pawlak, and they have consistent ability to deal with uncertainty in data.

Example 1. *A toy information system is shown in Table 2. Suppose that $\alpha = 0.6, \beta = 0.4, X = \{x_1, x_2, x_4, x_7\}$, and $R = \{Locus\}$. For the attribute Decision, its attribute domain is $\{Yes,No\}$. The objects with decision values of * are not labeled.*

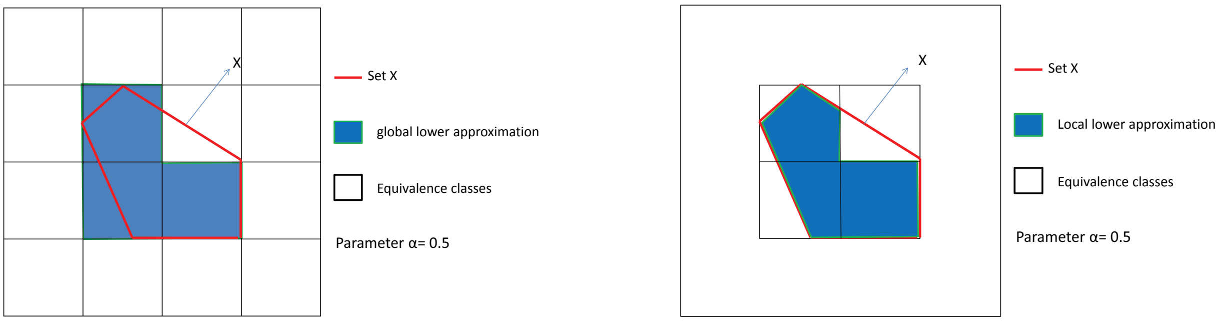
For the local rough sets, we only need to obtain equivalence classes for these objects from X. After a computation, we have

$[x_1]_R = \{x_1, x_2, x_3\}, [x_2]_R = \{x_1, x_2, x_3\}, [x_4]_R = \{x_4, x_5, x_6\}, [x_7]_R = \{x_7, x_8\},$
$P(X|[x_1]_R) = \frac{2}{3}, P(X|[x_2]_R) = \frac{2}{3}, P(X|[x_4]_R) = \frac{1}{3}, P(X|[x_7]_R) = \frac{1}{2},$
Thus,
$\underline{R}_{0.6}(X) = \{x_1, x_2\},$
$\overline{R}_{0.4}(X) = [x_1]_R \cup [x_2]_R \cup [x_7]_R = \{x_1, x_2, x_3, x_7, x_8\}.$

Unlike a global rough set, it can be seen from the definition of the rough set above, that the computation of its lower/upper approximation is only based on the information granules determined by objects within a target concept, not but those of all objects coming from a given universe, which can significantly reduce time consumption in computing approximations. Correspondingly, we call the kind of rough sets *local rough sets*. In this paper, for a local rough set, we call the $\underline{R}_\alpha(X)$ the local lower approximation of X and the $\overline{R}_\beta(X)$ the local upper approximation of X, respectively.

Fig. 1 shows the difference between a global rough set and a local rough set with the same universe and target concept, as well as the same equivalence relation, where the parameter $\alpha = 0.5$. In Fig. 1(a), the information granules determined by objects coming from the universe all need to compute and compare with the target concept X for obtaining its lower approximation. But in Fig. 1(b), we only need to calculate the information granules of objects coming from the target concept X, and also only compare them with the target concept for determining its lower approximation, which is an efficient computation strategy. In addition, unlike the lower approximation of the global rough set may overflow the range of a target concept when $1 > \alpha > 0$, the local rough set does not, in which its lower approximation must be included in the target concept. This can be seen a good property to reduce over-fitting degree in attribute reduction with the global rough set when $1 > \alpha > 0$.

In what follows, we address some interesting properties of a local rough set.



(a) Global rough sets

(b) Local rough sets

Figure 1: Difference between global rough set and local rough set

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

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

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

(3) $\underline{R}_\alpha(\emptyset) = \overline{R}_\beta(\emptyset) = \emptyset$,
$\underline{R}_\alpha(U) = \overline{R}_\beta(U) = U$;

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

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

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

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

Proof. (1) By $\alpha-$lower approximation in Definition 1, we can easily get that for any $X \subseteq U$ and $0 < \alpha \leq 1$, $\underline{R}_\alpha(X) \subseteq X$.

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

(3) Since for any $x \in U$, $x \notin \emptyset$ and $\mathcal{D}(\emptyset/[x]_R) = 0$. Then according to Definition 1, we have that for any $0 \leq \beta < \alpha \leq 1$, $\underline{R}_\alpha(\emptyset) = \overline{R'}_\beta(\emptyset) = \emptyset$. Furthermore, for any $x \in U$, $[x]_R \subseteq U$, and then $\mathcal{D}(U/[x]_R) = 1$. So, for any $0 \leq \beta < \alpha \leq 1$, we can get that $x \in [x]_R$, $\mathcal{D}(U/[x]_R) = 1 \geq \alpha$ and $\mathcal{D}(U/[x]_R) = 1 > \beta$, $\forall x \in U$. Thus, $\underline{R}_\alpha(U) = \overline{R}_\beta(U) = U$.

(4) Suppose $X \subseteq Y$. Take $x \in \underline{R}_\alpha(X)$, one has that $x \in X \subseteq Y$ and $\mathcal{D}(X/[x]_R) \geq \alpha$. Since $X \subseteq Y$, according to Eq. (5), we can get that $\mathcal{D}(X/[x]_R) \leq \mathcal{D}(Y/[x]_R)$. Thus, $\forall x \in Y$, one has $\mathcal{D}(Y/[x]_R) \geq \alpha$ and $x \in \underline{R}_\alpha(Y)$. By the arbitrariness of x, we have that $\underline{R}_\alpha(X) \subseteq \underline{R}_\alpha(Y)$. Analogously, we can prove that $X \subseteq Y$ implies $\overline{R}_\beta(X) \subseteq \overline{R}_\beta(Y)$.

(5) $\forall x \in \underline{R}_{\alpha}(X \cap Y)$, we have that

$$\begin{aligned}
x \in \underline{R}_{\alpha}(X \cap Y) \quad & \Leftrightarrow x \in X \cap Y,\ \mathcal{D}(X \cap Y/[x]_R) \geq \alpha \\
& \Leftrightarrow x \in X \text{ and } x \in Y,\ \mathcal{D}(X \cap Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in X \text{ and } x \in Y,\ \mathcal{D}(X/[x]_R) \geq \mathcal{D}(X \cap Y/[x]_R) \geq \alpha, \\
& \quad \mathcal{D}(Y/[x]_R) \geq \mathcal{D}(X \cap Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in X,\ \mathcal{D}(X/[x]_R) \geq \alpha, \text{ and } x \in Y,\ \mathcal{D}(Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in \underline{R}_{\alpha}(X) \text{ and } x \in \underline{R}_{\alpha}(Y) \\
& \Rightarrow x \in \underline{R}_{\alpha}(X) \cap \underline{R}_{\alpha}(Y),
\end{aligned}$$

so one can get that $\underline{R}_{\alpha}(X \cap Y) \subseteq \underline{R}_{\alpha}(X) \cap \underline{R}_{\alpha}(Y)$.

Meanwhile, $\forall x \in \overline{R}_{\beta}(X) \cup \overline{R}_{\beta}(Y)$, one has that

$$\begin{aligned}
x \in \overline{R}_{\beta}(X) \cup \overline{R}_{\beta}(Y) \quad & \Leftrightarrow x \in \overline{R}_{\beta}(X) \text{ or } x \in \overline{R}_{\beta}(Y) \\
& \Leftrightarrow x \in X,\ \mathcal{D}(X/[x]_R) > \beta, \text{ or } x \in Y,\ \mathcal{D}(Y/[x]_R) > \beta, \\
& \Rightarrow x \in X,\ \mathcal{D}(X \cup Y/[x]_R) \geq \mathcal{D}(X/[x]_R) > \beta, \\
& \quad \text{or } x \in Y,\ \mathcal{D}(X \cup Y/[x]_R) \geq \mathcal{D}(Y/[x]_R) > \beta, \\
& \Rightarrow x \in X \text{ or } x \in Y,\ \mathcal{D}(X \cup Y/[x]_R) > \beta, \\
& \Rightarrow x \in X \cup Y,\ \mathcal{D}(X \cup Y/[x]_R) > \beta, \\
& \Rightarrow x \in \overline{R}_{\beta}(X \cup Y),
\end{aligned}$$

then we obtain that $\overline{R}_{\beta}(X \cup Y) \supseteq \overline{R}_{\beta}(X) \cup \overline{R}_{\beta}(Y)$.

(6) For any $X, Y \subseteq U$, we have yhat

$$\begin{aligned}
x \in \underline{R}_{\alpha}(X) \cup \underline{R}_{\alpha}(Y) \quad & \Leftrightarrow x \in \underline{R}_{\alpha}(X) \text{ or } x \in \underline{R}_{\alpha}(Y) \\
& \Leftrightarrow x \in X,\ \mathcal{D}(X/[x]_R) \geq \alpha, \text{ or } x \in Y,\ \mathcal{D}(Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in X,\ \mathcal{D}(X \cup Y/[x]_R) \geq \mathcal{D}(X/[x]_R) \geq \alpha, \text{ or} \\
& \quad x \in Y,\ \mathcal{D}(X \cup Y/[x]_R) \geq \mathcal{D}(Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in X, \text{ or } x \in Y,\ \mathcal{D}(X \cup Y/[x]_R) \geq \alpha, \\
& \Rightarrow x \in \underline{R}_{\alpha}(X \cup Y),
\end{aligned}$$

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

Analogously, we can prove that $\overline{R}_{\beta}(X \cap Y) \subseteq \overline{R}_{\beta}(X) \cap \overline{R'}_{\beta}(Y)$.

(7) For any $0.5 < \alpha_1 < \alpha_2 \leq 1$, one has that

$$\begin{aligned}
x \in \underline{R}_{\alpha_2}(X) \quad & \Rightarrow x \in X,\ \mathcal{D}(X/[x]_R) \geq \alpha_2 \\
& \Rightarrow x \in X,\ \mathcal{D}(X/[x]_R) \geq \alpha_2 > \alpha_1 \\
& \Rightarrow x \in X,\ \mathcal{D}(X/[x]_R) \geq \alpha_1 \\
& \Rightarrow x \in \underline{R}_{\alpha_1}(X).
\end{aligned}$$

Then $0.5 < \alpha_1 < \alpha_2 \leq 1$ implies that $\underline{R}_{\alpha_2}(X) \subseteq \underline{R}_{\alpha_1}(X)$.

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

For simplicity, when $\alpha + \beta = 1$ and $\alpha \geq 0.5$, if we adopt the inclusion degree $D(X/Y) = \frac{|X \cap Y|}{|Y|}$, then $\underline{R_{\alpha}}$ and $\overline{R_{\beta}}$ are referred to as local variable $\alpha-$lower and local variable $\beta-$upper approximations, respectively.

In the local lower approximation $\underline{R}_{\alpha}(X)$ of a target concept X with respect to R, in fact, these objects can be further divided into two categories. We observe the following equation

$$\underline{R}_{\alpha}(X) = \{x \mid \mathcal{D}(X/[x]_R) \geq \alpha,\ x \in X\} = \{x \mid \mathcal{D}(X/[x]_R) = 1,\ x \in X\} \cup \{x \mid 1 > \mathcal{D}(X/[x]_R) \geq \alpha,\ x \in X\}.$$

Here, we denote $CL_R(X) = \{x \mid \mathcal{D}(X/[x]_R) = 1,\ x \in X\}$, called a certain set of $\underline{R}_{\alpha}(X)$, and denote $PL_R(X) = \{x \mid 1 > \mathcal{D}(X/[x]_R) \geq \alpha,\ x \in X\}$, called a possible set of $\underline{R}_{\alpha}(X)$. It is obvious that

$$\underline{R}_{\alpha}(X) = CL_R(X) \cup PL_R(X) \text{ and } |\underline{R}_{\alpha}(X)| = |CL_R(X)| + |PL_R(X)|.$$

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 $\mathcal{D}(X/[x]_Q) = 1$, so $\frac{|X \cap [x]_Q|}{|[x]_Q|} = 1$, thus $[x]_Q \subseteq X$. In addition, due to $P \prec Q$, we have that $[x]_P \subseteq [x]_Q \subseteq X$. Thus $\mathcal{D}(X/[x]_P) = 1$. Then, the object $x \in CL_P(X)$.

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

In real applications, an information system $S = (U, AT)$ is often employed for characterizing the relationship between objects and attributes in rough set theory, where U is a finite non-empty set of objects and AT is a finite non-empty set of attributes (predictor features) [26]. While for a classification problem, a decision table $S = (U, C \cup D)$ with $C \cap D = \emptyset$ is often used (it can be regarded as a special information system with $AT = C \cup D$), where an element of C is called a condition attribute, C is called a condition attribute set, an element of D is called a decision attribute, and D is called a decision attribute set [26]. Each non-empty subset $B \subseteq C$ determines an equivalence relation in the following way:

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

where $a(x)$ and $a(y)$ denote the values of objects x and y under a condition attribute a, respectively. This equivalence relation R_B partitions U into some equivalence classes (or equivalence information granules) given by

$U/R_B = \{[x]_B \mid x \in U\}$, for simplicity, U/R_B will be replaced by U/B,

where $[x]_B$ denotes the equivalence class determined by x with respect to B, i.e., $[x]_B = \{y \in U \mid (x, y) \in R_B\}$.

It is well known that in machine learning, a classifier is built using 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. The growing popularity of big data are available. Therefore, techniques to automatically do rough data analysis from big data in a semi-supervised way, with limited labeled data, are desirable.

In the above case, we assume that some objects coming from the entire universe U are labeled as r mutually exclusive crisp subsets $\{X^1, X^2, \cdots, X^r\}$ by the decision attributes D. For many large scale data sets, we often have that $|\bigcup\limits_{j=1}^{r} X^j| \ll |U|$. Given any subset $B \subseteq C$ and R_B is the equivalence relation induced by B, like a global rough set, in such a case, one also can define the local lower and local upper approximations of the decision attributes D as

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

Denoted by $POS_B(D) = \bigcup_{i=1}^{r} \underline{R_B}(X^i)$, it is called the local positive region of D with respect to the condition attribute set B. The local lower/upper approximation of B with respect to D can be easily computed through using the local rough set. It deserves to point out that in this paper, U/D is not the partition of the entire universe induced by the decision attributes D, but those classes with labels.

3.2. Several measures in the local rough set

Similar to Pawlak' rough set, uncertainty of a local rough set is also due to the existence of a boundary region in a local rough set. The greater the local boundary region of a local rough set, the lower is its accuracy (and vice versa). In order to more precisely express this idea in the local rough set, we give the formal definition of an accuracy measure as follows.

Definition 2. *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*

$$\alpha(B,\ X) = \frac{|\underline{R_B}(X)|}{|\overline{R_B}(X)|}, \tag{7}$$

where $X \neq \emptyset$ and $|X|$ means the cardinality of a set X.

Gediga [6] introduced a simple statistic for the precision of (deterministic) approximation of $X \subseteq U$ given R, which is not affected by the approximation of $\sim X$. This is just the relative number of objects in X which can be approximated by R. Given a local rough set, it becomes the same formula

$$\pi(R, X) = \frac{|\underline{R_B}(X)|}{|X|}. \tag{8}$$

Obviously, $\pi(R, X) \geq \alpha(R, X)$.

As Section 1 mentioned, the global lower approximation of a target concept may be not monotonic with number of attributes increasing. In next study, this precision of approximation π will be used to compare the monotonicity of the proposed local rough set and the global rough set for concept approximations in attribute reduction of a target concept.

Accuracy of approximation is another important measure in rough set theory, which is used to characterize the ability of a partition to approximate a decision [3, 6]. Given a decision table $S = (U, C \cup D)$ and $B \subseteq C$ an attribute subset, for a decision approximation in the local rough set, the accuracy of approximation of B with respect to D can be formally written as follows

$$\gamma(B, D) = \frac{\sum\{|\underline{R_B}(X)| : X \in U/D\}}{\sum\{|X| : X \in U/D\}} = \frac{|POS_B(D)|}{\sum\{|X| : X \in U/D\}}, \tag{9}$$

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

In this study, the accuracy of approximation γ is employed for investigating the monotonicity of a heuristic attribute reduction algorithm in the local rough set and that in the global rough set.

4. Computing approximation and attribute reduction of a target concept

It is well known that in rough set theory, approximation and attribute reduction of a target concept and those of a target decision have different computational mechanisms. Therefore, we will divide algorithms about the local rough set into two parts. In this section, we aim to solve the problem of how to compute approximation and attribute reduction of a target concept.

4.1. Computing the local lower approximation of a target concept

In this part, we propose an algorithm to compute a local lower approximation of a target concept, and verify its efficiency through employing serval real data sets.

4.1.1. Algorithm

From Fig. 1 and the relative analysis, we know that a local rough set only needs to calculate information granules of objects within a given target concept, and also only compares them with the target concept for determining its lower/upper approximation. It is a more efficient strategy to obtain the lower approximation of a target concept than existing rough set models. In the following, we give the diagram of computing the lower approximation of a local rough set, which is formally depicted as follows.

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

```
Input: An information system $S = (U, AT)$, a target concept $X \subseteq U$ and a parameter value $\alpha$;
Output: Local lower approximation $LA$ of $X$ with respect to $AT$.

$Step$ 1: From $i = 1$ to $|X|$ Do
        {Compute $[x_i]_{AT}$ of $x_i$, $x_i \in X$}; //Generating equivalence classes to approximate the target concept
$Step$ 2: $LA \leftarrow \emptyset$, $i \leftarrow 1$;
$Step$ 3: While $i < |X|$ Do
        {
```

Table 3: The time complexities of computing the lower approximations of a global rough set and a local rough set

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

```
        If $\mathcal{D}(X/[x_i]_{AT}) \geq \alpha$
            then $LA \leftarrow LA \cup \{x_i\}$, $i \leftarrow i + 1$,
            otherwise $i \leftarrow i + 1$;
        }
    $Step$ 4: Return $LA$ and end.
```

In the above LLAC algorithm, Step 1 needs to compute $|X|$ equivalence classes through scanning the entire universe U, hence its time complexity is $O(|X||U|)$. However, the time complexity of computing all equivalence classes on U in a global rough set is $O(|U|^2)$. In Step 3, we compare the $|X|$ equivalence classes with the target concept X for obtaining its lower approximation, thus its time complexity is $O(|X|^2)$. But the time complexity of computing a global lower approximation is $O(|X||U|)$. Each of other steps of the LLAC algorithm is constant. In order to conveniently compare, we denote the algorithm of computing a global lower approximation by GLAC, where G means "global". To stress these findings, the time complexity of each step in GLAC and that in LLAC are shown as Table 3.

It can be seen from Table 3 that the time complexity of the LLAC algorithm is much lower than that of the GLAC algorithm. In addition, as we know, when a target concept is given, whose size is often far less than the size of the entire universe and can be seen as a constant. Hence, the time complexity of the LLAC algorithm is only linear $O(|X||U| + |X|^2)$ in terms of $|U|$. Therefore, one can draw a conclusion that the LLAC algorithm may significantly reduce the computational time for computing a lower approximation than the classical GLAC algorithm, which can work well in the context of big data.

For computing a local upper approximation, the same algorithm diagram can be employed, and the same time complexity can be also observed. We omit its discussion here.

In the next theorem, we quantitatively explore the ratio of time-reduction of the LLAC algorithm relative to the GLAC algorithm.

Theorem 2. *Given an information system $S = (U, AT)$ and a target concept $X \subseteq U$, the ratio of time-reduction 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 3, we have that

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

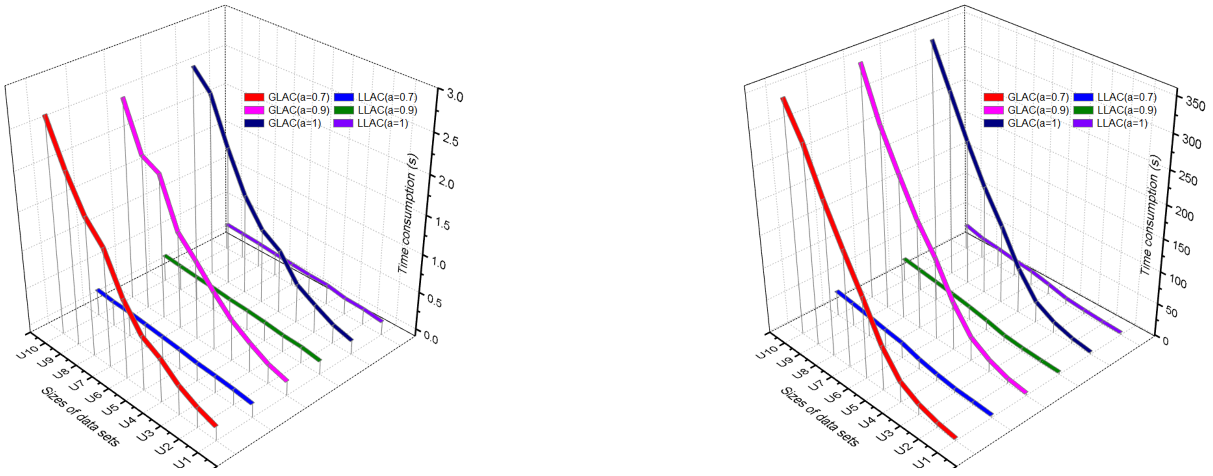
Hence, the ratio of time-reduction of the LLAC algorithm is $p = 1 - \frac{|X|}{|U|}$. This completes the proof.

From the above theorem, it can be seen that the LLAC algorithm is very efficient for computing a local lower approximation, which 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.

In many real large-scale data sets, the scale of a data set is very big, while that of a given target concept to observe is often smaller, i.e., $|X| \ll |U|$. Hence, when $|U| \rightarrow \propto$, one has that $\lim\limits_{|U| \rightarrow \propto} (1 - \frac{|X|}{|U|}) = 1$. This implies that the efficiency

Table 4: Data sets description

	Data sets	Cases	Features	Classes
1	Mushroom	5644	22	2
2	Tic-tac-toe	958	9	2
3	Dermatology	358	34	6
4	Kr-vs-kp	3196	36	2
5	Breast-cancer-wisconsin	683	9	2
6	Backup-large.test	376	35	19
7	Shuttle	58000	9	7
8	Letter-recognition	20000	16	26
9	Ticdata2000	5822	85	2



(a) Mushroom

(b) Shuttle

Figure 2: Times of LLAC and GLAC versus the size of universe on Mushroom and Shuttle

of the LLAC algorithm is amazing for rough data analysis in big data[1]. The bigger the scale of a data set, the more time-reduction of the LLAC algorithm.

4.1.2. Experimental analysis

In this subsection, in order to compare the LLAC algorithm with the classical GLAC algorithm, we employ nine UCI data sets [21] in Table 4 to verify the computational performance of the LLAC algorithm in the proposed local rough set, which are all categorical data (Shuttle and Ticdata2000 are preprocessed by discretization with entropy). In these nine data sets, Mushroom and Breast-cancer-wisconsin are two data sets with missing values. For uniform treatment of all data sets, we remove the objects with missing values.

In the experimental analysis, we compare the execution time of the LLAC algorithm with that of the GLAC algorithm vis-a-vis the size of universe. In the experiments in this paper, all algorithms are all run on a personal computer with Windows 7 and Xeon E5-1603 CPU 2.8 GHz, and ECC DDR3, 16 GB memory. The software being used is JAVA.

For the experimental design, we fix the size of target concept, which is the front 10% objects from each of these nine data sets. To distinguish the computational times, we divide each of these nine data sets into ten parts of equal

[1] In fact, the time consumption of the LLAC algorithm can be further decreased when a given data set is pre-ranked [47]. However, it is beyond the scope of this study. We here do not further discuss them.

Table 5: The computational times for concept approximation with LLAC and GLAC

Data sets	$\alpha = 0.7$		$\alpha = 0.9$		$\alpha = 1$	
	LLAC	GLAC	LLAC	GLAC	LLAC	GLAC
Mushroom	0.2633	2.6860	0.2647	2.5321	0.2582	2.5714
Tic-tac-toe	0.0342	0.1044	0.0340	0.1034	0.0341	0.1049
Dermatology	0.0115	0.0451	0.0116	0.0442	0.0115	0.0422
Kr-vs-kp	0.1831	1.4510	0.1817	1.3306	0.1793	1.3724
Breast-cancer-wisconsin	0.0260	0.0537	0.0246	0.0542	0.0254	0.0533
Backup-large.test	0.0105	0.0363	0.0117	0.0373	0.0110	0.0377
Shuttle	38.0850	346.1472	35.0175	353.4160	38.5338	347.3253
Letter-recognition	2.5850	24.2051	2.3535	21.6607	2.5630	24.4678
Ticdata2000	0.2476	2.0340	0.2635	1.9527	0.2684	1.9367

size. 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 the 2nd data set and the third part is regarded as the 3rd 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 show it vis-a-vis the size of universe. In this experiment, without loss of generality, we only let the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively.

The experimental results of these nine data sets are shown in Fig. 2 and Table 5. This figure displays more detailed change trend of each of two algorithms (LLAC and GLAC) with size of data set becoming increasing. Table 5 shows that the computational times of computing lower approximations of the same target concept using LLAC and GLAC algorithms on the nine data sets with various values of the parameter α.

One observation from Fig. 2[2] is that the computing time of each of these two algorithms increases with the increase of the size of data, and the other observation is that for each of values of the parameter α, the LLAC algorithm is consistently faster than the classical GLAC algorithm on the same universe and attribute set. In addition, we also can see that the differences between these two algorithms for time consumption are markedly larger when the size of the data set increases. From Table 5, it can be seen that as one of the important advantages of the LLAC algorithm, it only uses one tenth of the execution time used by GLAC. For example, the reduced time achieves 308.7915 seconds when $\alpha = 1$ on the data set Shuttle. One can say that for computing the local lower approximation of a target concept, the LLAC algorithm under the local rough set provides an efficient solution in the context of big data.

4.2. Searching a local attribute reduct of a target concept

In this subsection, we develop an algorithm for searching a local attribute reduct of a target concept.

4.2.1. Definition of a local attribute reduct of a target concept

It is well known that, one of important objectives of rough set theory is to learn a rough classifier. Given a target concept X, the classification problem aims to find an attribute subset to maximize the margin between X and $\sim X$. The found attribute subset is used to construct a corresponding rough classifier. The bigger the margin induced by an attribute subset, the stronger the rough classifier determined by it. In rough set theory, in general, the margin is characterized by the boundary region between the lower approximation and the upper approximation of X.

However, any rough set model with a parameter α has the limitation of its lower approximation being not monotonic, such that its attribute reduct found is often overfitting [35]. Beside this shortage, the searching process also spend much more computational time [35], [55]. In the following, we address attribute reduction of a target concept in the local rough set.

Firstly, we give the following definition of attribute reduct of a target concept in local rough sets.

[2] The time efficiency has been guaranteed by the theoretical evaluations, hence, we only shown 2 exemplary sub-figures on **Mushroom** and **Shuttle**. Fig. 5 and 7 have same considerations.

Definition 3. *Let $S = (U, AT)$ be an information system, $X \subseteq U$ a target concept and $B \subseteq AT$. If $|\underline{R_B}(X)| \geq |\underline{R_{AT}}(X)|$ and $|\underline{R_{B'}}(X)| \ngeq |\underline{R_{AT}}(X)|$ for any $B' \subset B$, then we call B a local attribute reduct of S with respect to X.*

From the above definition, in fact, there may exist multiple attribute reducts for a target concept in an information system. We assume that $\{B_1, B_2, \cdots, B_s\}$ are s local attribute reducts of S with respect to X. Similar to global rough sets, one also can define its core $Core = B_1 \cap B_2 \cap \cdots \cap B_s$. If the core is not an empty set, then the attributes from the core are indispensable for constructing a local attribute set of a target concept. Of course, it may be an empty set in some cases.

4.2.2. Algorithm

In local rough sets, when only one attribute reduct is needed, a heuristic algorithm can be designed with a greedy and forward search strategy. In a heuristic algorithm of finding a local attribute reduct, two important measures of attributes are used for heuristic functions, which are inner importance measure and outer importance measure. In the following, we give the versions of these two important measures of attributes in the context of local rough sets.

Definition 4. *Let $S = (U, AT)$ be an information system, $X \in U$, $B \subseteq AT$, $\forall a \in B$. The inner significance of a with respect to X is defined as*

$$Sig_{\alpha}^{inner}(a, B, X, U) = |\underline{R_B}_{\alpha}^{U}(X)| - |\underline{R_{B-\{a\}}}_{\alpha}^{U}(X)|.$$

Definition 5. *Let $S = (U, AT)$ be an information system, $X \in U$, $B \subseteq AT$, $\forall a \in AT - B$. The outer significance of a with respect to X is defined as*

$$Sig_{\alpha}^{outer}(a, B, X, U) = |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - |\underline{R_B}_{\alpha}^{U}(X)|.$$

The inner importance measure is applicable to determine the significance of every attribute, while the outer importance measure can be used in the forward feature selection process.

The computation of the two importance measures of attributes are main bodies of time consumption of a heuristic attribute reduction algorithm. Now we come back to see Theorem 1. From this theorem, we know that if any $x \in CL_Q(X)$, then $x \in CL_P(X)$, where $P, Q \subseteq AT$ with $P \prec Q$ are two attribute subsets. This means that $Q \subset P$. This principle implies that we do not recompute the equivalence classes of objects including in $CL_Q(X)$ and rejudge whether they belong to the local lower approximation or not for obtaining the local lower approximation $\underline{P}_{\alpha}(X)$. Taking this benefit into account, we want to introduce an efficient strategy of heuristic attribute reduction of a target concept.

In the following, we concentrate on the rank preservation of the outer significance measure of attribute based on the positive approximation encountered in an information system. Simply, we denote the certain set of $\underline{B}_{\alpha}(X)$ on the universe U by $CL_B^U(X) = \{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X\}$.

Theorem 3. *Let $S = (U, AT)$ be an information system, $X \subseteq U$, $B \subseteq AT$ and $U' = \bigcup\{[x]_B^U, x \in X\} - CL_B^U(X)$. For $\forall a, b \in AT - B$, if $Sig_{\alpha}^{outer}(a, B, X, U) \geq Sig_{\alpha}^{outer}(b, B, X, U)$, then $Sig_{\alpha}^{outer}(a, B, X, U') \geq Sig_{\alpha}^{outer}(b, B, X, U')$.*

Proof. From the definition of $Sig_{\alpha}^{outer}(a, B, X, U) = |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - |\underline{R_B}_{\alpha}^{U}(X)|$, we know that its value only depends on the local lower approximations. Since $U' = \bigcup\{[x]_B^U, x \in X\} - \{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X\}$, so $1 > \mathcal{D}(X/[x]_B^U) \geq \alpha$ for $x \in U'$, and $\mathcal{D}(X/[x]_B^U) = 0$ or $\mathcal{D}(X/[x]_B^U) = 1$ for $x \notin U - U'$. Hence $\forall \alpha > 0$, we have that

$$\begin{aligned}
\underline{R_B}_{\alpha}^{U}(X) &= \{x \mid \mathcal{D}(X/[x]_B^U) \geq \alpha,\ x \in X\} \\
&= \{x \mid \mathcal{D}(X/[x]_B^{U'}) \geq \alpha,\ x \in X\} \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) \geq \alpha,\ x \in X\} \\
&= \{x \mid 1 > \mathcal{D}(X/[x]_B^{U'}) \geq \alpha,\ x \in X\} \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) = 0,\ x \in X\} \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) = 1,\ x \in X\} \\
&= \{x \mid 1 > \mathcal{D}(X/[x]_B^{U'}) \geq \alpha,\ x \in X\} \cup \{\emptyset\} \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) = 1,\ x \in X\} \\
&= \{x \mid 1 > \mathcal{D}(X/[x]_B^{U'}) \geq \alpha,\ x \in X\} \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) = 1,\ x \in X\} \\
&= \underline{R_B}_{\alpha}^{U'}(X) \cup \{x \mid \mathcal{D}(X/[x]_B^{U-U'}) = 1,\ x \in X\}
\end{aligned}$$

From the above equation and the definition of local lower approximation, we know that

$$\begin{aligned}|\underline{R_{B\cup\{a\}}}_{\alpha}^{U'}(X)| - |\underline{R_B}_{\alpha}^{U'}(X)| &= |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - |\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X\}| - |\underline{R_B}_{\alpha}^{U'}(X)| \\ &= |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - (|\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X\}| + |\underline{R_B}_{\alpha}^{U'}(X)|) \\ &= |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - |\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X\} \cup \underline{R_B}_{\alpha}^{U'}(X)| \\ &= |\underline{R_{B\cup\{a\}}}_{\alpha}^{U}(X)| - |\underline{R_B}_{\alpha}^{U}(X)|\end{aligned}$$

Therefore, $\forall a, b \in AT - B$, if $Sig_{\alpha}^{outer}(a, B, X, U) \geq Sig_{\alpha}^{outer}(b, B, X, U)$, then $Sig_{\alpha}^{outer}(a, B, X, U') \geq Sig_{\alpha}^{outer}(b, B, X, U')$. 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 set. 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 reduct. Formally, a forward greedy attribute reduction algorithm for searching a local attribute reduct with respect to a given target concept can be written as follows.

Algorithm 2. A forward greedy local attribute reduction algorithm for a target concept (LARC)

```
Input: An information system $S = (U, AT)$, the parameter $\alpha$ and $X \subseteq U$;
Output: One reduct $red$.
Step 1: $red \leftarrow \emptyset$; //$red$ is the pool to conserve the selected attributes.
Step 2: Compute $Sig^{inner}(a_k, AT, X, U)$, $k \leq |AT|$; //$Sig^{inner}(a_k, AT, X, U)$ is the inner significance of attribute $a_k$.
Step 3: Put $a_k$ into $red$, where $Sig^{inner}(a_k, AT, X, U) = \max\{Sig^{inner}(a_k, AT, X, U),\ a_k \in AT\}$;
Step 4: $i \leftarrow 1$, $R_1 \leftarrow red$, $X_1 = X$ and $U_1 = U$;
Step 5: While $|\underline{R_{red}}_{\alpha}^{U_i}(X_i)| < |\underline{R_{AT}}_{\alpha}^{U_i}(X_i)|$ Do //This provides a stopping criterion.
        {
          $U_{i+1} = \cup_{x \in X_i}[x]_{red}^{U_i} - CL_{red}^{U_i}(X_i)$,
          $X_{i+1} = X_i - CL_{red}^{U_i}(X_i)$,
          $i \leftarrow i + 1$,
          $red \leftarrow red \cup \{a_0\}$, where $Sig^{outer}(a_0, red, X_i, U_i) = max\{Sig^{outer}(a_k, red, X_i, U_i), a_k \in C - red\}$;
                                   //$Sig^{outer}(a_k, C, X_i, U_i)$ is the outer significance of the attribute $a_k$.
          $R_i \leftarrow R_{i-1} \cup \{a_0\}$,
        }
Step 6: Return $red$ and end.
```

In order to conveniently compare, we denote the algorithm of finding an attribute reduct in the context of global rough sets by GARC.

In the above LARC algorithm, Step 1 needs to compute $|AT|$ local lower approximations using the LLAC algorithm, hence its time complexity is $O(|AT|(|X|^2 + |X||U|))$. However, the time complexity of this step in global rough sets is $O(|AT|(|U|^2 + |X||U|))$. Step 3 only scans the $|AT|$ attributes, hence it time complexity is $O(|AT|)$, and this step in global rough sets has the same case. In Step 5, we begin with the core and add an attribute with the maximal significance into the set in each stage until finding a reduct. This process is called a forward reduction algorithm whose time complexity is $O(\sum_{i=1}^{|AT|}(|AT| - i + 1))(|X_i|^2 + |X_i||U_i|))$. However, the time complexity of this step in a classical heuristic algorithm is $O(\sum_{i=1}^{|AT|}(|AT| - i + 1))(|U|^2 + |X||U|))$. Each of other steps of the LLAC algorithm is constant. To stress these findings, the time complexity of each step in the LARC algorithm and the GARC algorithm is shown as Table 6.

It can be seen from Table 6 that the time complexity of the LARC algorithm is much lower than that of the GARC algorithm. Hence, one can draw a conclusion that the LARC algorithm may significantly reduce the computational time for attribute reduction from an information system, which can efficiently work in the context of big data.

Table 6: The time complexities of the LARC algorithm and the GARC algorithm

Algorithms	Step 2	Step 3	Step 5	Other steps
GARC	$O(\vert AT\vert(\vert U\vert^2 + \vert X\vert\vert U\vert))$	$O(\vert AT\vert)$	$O(\sum_{i=1}^{\vert AT\vert}(\vert AT\vert - i + 1))(\vert U\vert^2 + \vert X\vert\vert U\vert))$	Constant
LARC	$O(\vert AT\vert(\vert X\vert\vert U\vert + \vert X\vert^2))$	$O(\vert AT\vert)$	$O(\sum_{i=1}^{\vert AT\vert}(\vert AT\vert - i + 1))(\vert X_i\vert\vert U_i\vert + \vert X_i\vert^2))$	Constant

Table 7: The number of attributes in attribute reducts of LARC and GARC

Data sets	Original features	LARC algorithm			GARC algorithm		
		Selected features			Selected features		
		$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
Mushroom	22	18	18	18	22	18	18
Tic-tac-toe	9	8	8	8	8	8	8
Dermatology	34	5	5	5	5	5	5
Kr-vs-kp	36	31	31	31	36	31	31
Breast-cancer-wisconsin	9	9	9	9	9	9	9
Backup-large.test	35	4	4	4	4	4	4
Shuttle	9	8	8	8	8	8	8
Letter-recognition	16	16	16	16	16	16	16
Ticdata2000	85	24	24	24	24	24	24

In fact, for many real large-scale data sets, the size of U_i and that of X_i can be quickly reduced in the process of forward searching. Hence, the LARC algorithm will possess much better time-reduction performance for attribute reduction from limited big data. The bigger the scale of a data set, the more time-reduction of the LARC algorithm.

4.2.3. Experimental analysis

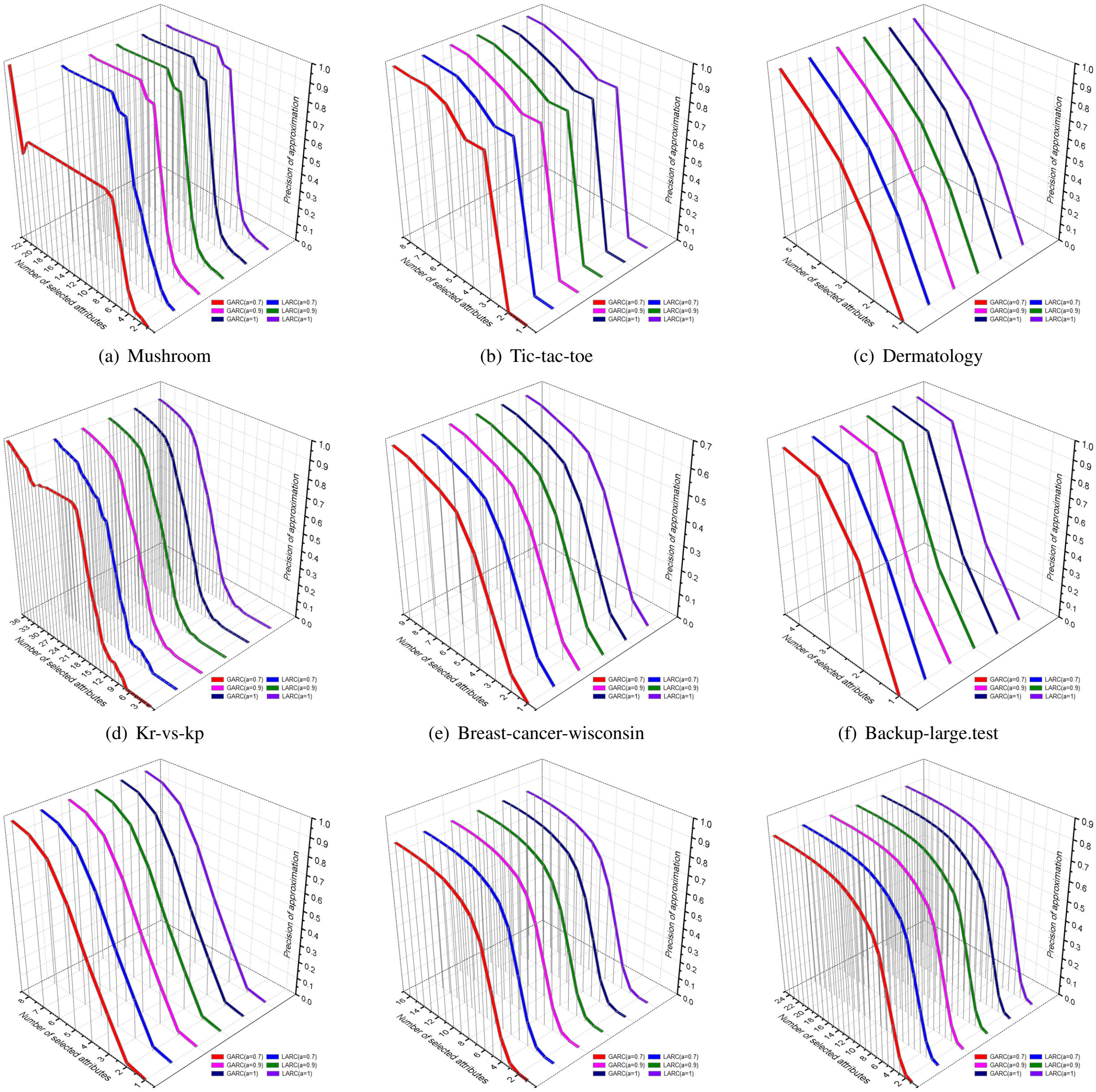
In this experimental analysis, we want to verify the advantages of the LARC algorithm from two sides: over-fitting and efficiency.

- Over-Fitting issue

The over-fitting degree of an attribute reduction algorithm can be observed by the monotonicity of attribute reducts, which is measured by the precision of approximation $\pi(R, X)$ in Eq. (8). Given an information system $S = (U, AT)$, $X \subseteq U$, $P = \{R_1, R_2, \cdots, R_n\}$ a family of attributes with $R_1 \geq R_2 \geq \cdots \geq R_n$ ($R_i \in 2^{AT}$). Let $P_i = \{R_1, R_2, \cdots, R_i\}$, we denote the precision of approximation of X with respect to P_i by $\pi(P_i, X) = \frac{|\underline{R_i}_{\alpha}(X)|}{|X|}$.

In this experiment, we observe both the precision of approximation and the number of selected attributes in the forward searching process for the LARC algorithm and the GARC algorithm. In general, if the precision of approximation monotonically increases and the number of attributes in an attribute reduct induced an attribute reduction algorithm is much smaller, we can say that the algorithm may be much better. It is because that the smaller number of attributes may have the same approximation ability. For the experiment, we take the front 10% objects from each of the nine data sets as its corresponding target concept. Fig. 3 shows the precision of approximation of X used by each of the LLAC algorithm and the GLAC algorithm and show it vis-a-vis the number of attributes, where the parameter is assigned as $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively. In each of these sub-figures, the x-coordinate pertains to the number of the attributes, while the y-coordinate concerns the value of precision of approximation. Table 6 displays the number of attributes in attribute reducts obtained by the LARC algorithm and the GARC algorithm with the same parameter value, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively.

It is easy to see from Fig. 3 that the precision of the approximation of the LARC algorithm monotonically increases with the number of attributes added to each of the nine data sets. However, the GARC algorithm does not. For example, in subfigures (a) and (d), the precision of approximation of the GARC algorithm is not monotonic as the number of attributes increases. To achieve the same value of precision of approximation, the GARC algorithm needs to add much more attributes, which may cause over-fitting in attribute reduction of a target concept. From Table 7, we



(a) Mushroom (b) Tic-tac-toe (c) Dermatology

(d) Kr-vs-kp (e) Breast-cancer-wisconsin (f) Backup-large.test

(g) Shuttle (h) Letter-recognition (i) Ticdata2000

Figure 3: Monotonicity of precision of approximation of a target concept versus number of attributes

Table 8: The computational times for attribute reductions with LARC and GARC

Data sets	α = 0.7		α = 0.9		α = 1	
	LARC	GARC	LARC	GARC	LARC	GARC
Mushroom	8.8181	103.8508	8.5825	151.8479	8.5207	149.1398
Tic-tac-toe	0.1503	0.8132	0.1465	0.8046	0.1462	0.8985
Dermatology	0.1174	0.5952	0.1190	0.6274	0.1203	0.6918
Kr-vs-kp	8.0904	121.8135	9.1224	131.0399	9.5779	130.4910
Breast-cancer-wisconsin	0.1430	0.4400	0.1420	0.4624	0.1417	0.4997
Backup-large.test	0.1129	0.2488	0.1164	0.2478	0.1190	0.2797
Shuttle	852.6764	9760.5057	825.6931	9742.6694	824.2109	9747.1383
Letter-recognition	125.9547	3472.5252	110.9230	3447.8506	107.1654	3441.2356
Ticdata2000	64.6641	2400.0551	66.8567	2384.5602	64.4674	2415.4694

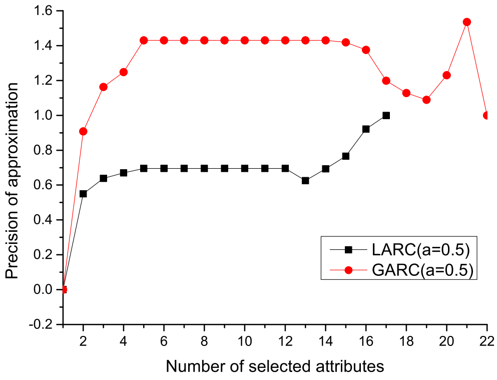
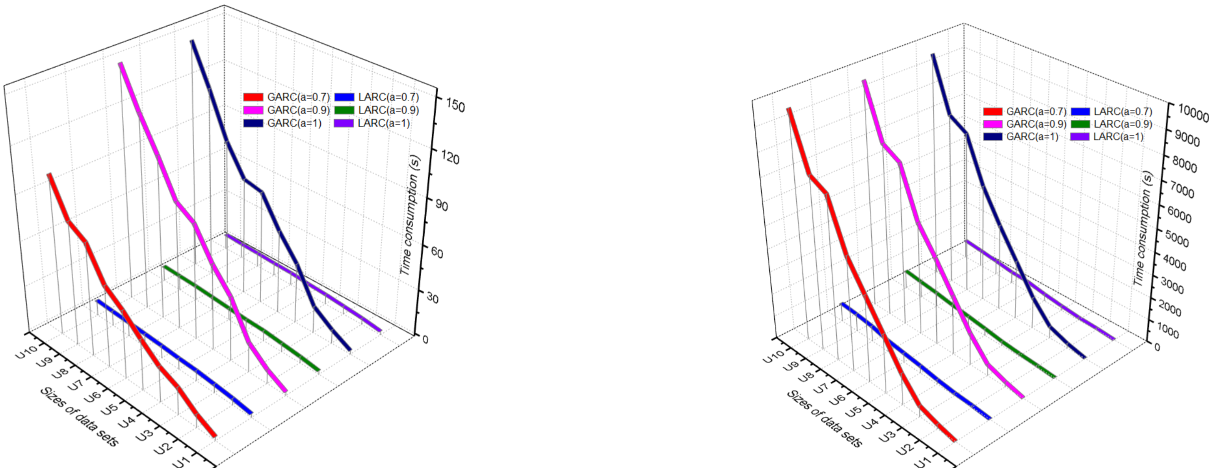


Figure 4: Precision of approximation of a target concept versus number of attributes on Mushroom



(a) Mushroom

(b) Shuttle

Figure 5: Times of LARC and GARC versus the size of the universe on Mushroom and Shuttle

also can see that the number of attributes selected by the LARC algorithm is not more than that selected by the GARC algorithm with the same parameter value on the same data set. For instance, the GARC algorithm may lead to over-fitting phenomenon of an attribute reduct of a target concept on Mushroom and Kr-vs-kp. To stress this phenomenon, we also tested these two algorithms with the parameter $\alpha = 0.5$ on Mushroom data set, which is shown as Fig. 4. From Fig. 4, one can see that the values of precision of approximation coming from the GARC algorithm are often over one, which clearly causes over-fitting in attribute reduction. This is because that the global lower approximation of a target concept may be beyond its own region. From these results, one can say that the LARC algorithm under the local rough set may be a good solution for reducing the over-fitting degree in attribute reduction for a target concept.

- Efficiency of the LARC algorithm

For verifying the efficiency of the LARC algorithm, we still take the same experimental setting as Section 4.1.2. In this part, we fix the size of target concept (the front 10% objects from each of these nine data sets) to compare computational time of the LARC algorithm with the GARC algorithm vis-a-vis the size of universe, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively. The experimental results on these nine data sets are shown in Fig. 5 and Table 8. Fig. 5 displays more detailed change trend of each of two algorithms (LARC and GARC) with size of data set becoming increasing. Table 8 shows that the computational times of attribute reductions of the same target concept using LARC and GARC algorithms on the nine data sets with different values of the parameter α.

It is easy to note from Fig. 5 that the computing time of each of these two algorithms increases with the increase of the size of data. From the two subfigures in Fig. 5, we also can see that for each of values of the parameter α, the LARC algorithm is consistently faster than the classical GARC algorithm on the same universe. In addition, we also can see that the differences between these two algorithms for time consumption are profoundly larger when the size of the data set increases. As one of the important advantages of the LARC algorithm, it can be seen from Table 8 that the time reduction performance of the LARC algorithm is very clear compared to the GARC algorithm. For example, the LARC algorithm only takes 1/30 computational time of the GARC algorithm on the data set Letter-recognition. One can say that for computing the attribute reduct of a target concept, the LARC algorithm under the local rough set provides an very efficient solution in the context of big data.

5. Computing approximation and attribute reduction of a target decision

5.1. Computing local lower approximation of a target decision

In this part, we develop an algorithm to compute a local lower approximation of a target decision.

From Algorithm 1 and its related analysis, we know that the LLAC algorithm provides a more efficient strategy to obtain the lower approximation of a target concept than existing rough set models. Based on this diagram, naturally, we can design a corresponding algorithm for computing the local lower approximation of a target decision, which is formally depicted as follows.

Algorithm 3. Computing the local lower approximation of a target decision (LLAD)

```
Input: A decision table $S = (U, C \cup D)$ and a parameter value $\alpha$;
Output: Local lower approximation $LA$ of a target decision $D = \{X^1, X^2, \cdots, X^r\}$.

Step 1: $LA \leftarrow \emptyset$;
Step 2: From $j = 1$ to $r$ Do
        {
          From $i = 1$ to $|X^j|$ Do
            {Compute $[x_i]_C$ of $x_i$, $x_i \in X^j$}; //Generating equivalence classes to approximate every target concept
          $LA_j \leftarrow \emptyset$, $i \leftarrow 1$;
          While $i < |X^j|$ Do
            {
              If $\mathcal{D}(X^j/[x_i]_C) \geq \alpha$
                then $LA_j \leftarrow LA_j \cup \{x_i\}$, $i \leftarrow i + 1$,
                otherwise $i \leftarrow i + 1$;
            }
```

```
    }
$Step$ 3: $LA \leftarrow \{LA_1, LA_2, \cdots, LA_r\}$;
$Step$ 4: Return $LA$ and end.
```

In the above LLAD algorithm, Step 2 needs to compute r local lower approximations, hence its time complexity is $O(\sum_{j=1}^{r} |X^j||U|) = |\bigcup_{j=1}^{r} X^j||U|$. It also does not need to compute all equivalence classes with the time complexity $O(|U|^2)$, which is an efficient strategy for big data.

5.2. *Searching a local attribute reduct of a target decision*

In this subsection, we develop an algorithm for searching a local attribute reduct of a target decision.

5.2.1. *Definition of a local attribute reduct of a target decision*

As Section 4.2.1 mentioned, any rough set model with a parameter $\alpha \neq 1$ has the limitation of its lower approximation being not monotonic, such that its attribute reduct found is often overfitting [35], [55]. This is caused by the constraint condition $\underline{R_B}(X) = \underline{R_{AT}}(X)$ of an attribute reduction [12], [34], which is a very strong constraint. Beside this shortage, the searching process also spend much more computational time. For attribute reduction of a target decision, a global rough set model has the same limitations, in which the constraint condition $POS_B(D) = POS_C(D)$ of an attribute reduction is also too strong and the corresponding algorithm is also time-consuming [12], [34].

In order to overcome two limitations of low efficiency and over-fitting, we introduce the definition of attribute reduct of a target decision in the local rough set. If we want to find an attribute subset such that the local positive region induced by it is at least the same as that induced by the original attribute set, then a local attribute reduct can be defined as follows.

Definition 6. *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 reduct of S.*

In fact, the definition may induce to multiple attribute reducts for a target decision with class labels. Let $\{B_1, B_2, \cdots, B_s\}$ be s local attribute reducts of S with respect to D, we also can define its core $Core = B_1 \cap B_2 \cap \cdots \cap B_s$. If the core is not an empty set, then the attributes from the core are indispensable for constructing a local attribute set of a target decision. Sometimes the core may be an empty set.

5.2.2. *Algorithm*

In fact, there may be multiple local attribute reducts for a given decision table. It can also be proven that finding the minimal local attribute reduct of a decision table is NP hard. When only one attribute reduct is needed, based on the significance measures of attributes, some heuristic algorithms have been proposed in global rough sets, most of which are greedy and forward search algorithms. These search algorithms start with a nonempty set, and keep adding one or several attributes of high significance into a pool each time until the constraint condition is satisfied. In 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. The inner importance measure is applicable to determine the significance of every attribute, while the outer importance measure can be used in a forward attribute reduction.

The idea of attribute reduction using positive region was first originated by Grzymala-Busse in Refs. [8], and the corresponding algorithm ignores the additional computation of choice of significant attributes. Hu and Cercone proposed a heuristic attribute reduction method, called positive-region reduction (PR), which keeps the positive region of target decision unchanged [12]. For this local rough set, we also can develop a local positive-region based attribute reduction algorithm, in which the significance measures of attributes are defined as follows.

Definition 7. *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 8. *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*

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

As mentioned above, these two significance measures of attributes provide some heuristics to guide the mechanism of forward searching a feature subset. To obtain an efficient strategy of heuristic attribute reductions, we concentrate on the rank preservation of the significance measures of attributes in a decision table. Simply, we also denote the certain set of $POS_B(D)$ on the universe U by $CP_B^U(D) = \bigcup\{[x]_B^U \mid \mathcal{D}(X/[x]_B^U) = 1, x \in X, X \in U/D\}$, called the certain positive region. It is interesting that one proves the following theorem of rank preservation.

Theorem 4. *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 $U' = U - \bigcup\{[x]_B^U \mid \mathcal{D}(X/[x]_B^U) = 1, x \in X, X \in U/D\}$, so there must exist $X \in U/D$ such that $1 > \mathcal{D}(X/[x]_B^U) \geq \alpha$ 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

$$\begin{aligned}
POS_B^U(D) &= \{x \mid \mathcal{D}(X/[x]_B^U) \geq \alpha,\ x \in X,\ X \in U/D\} \\
&= \{x \mid 1 > \mathcal{D}(X/[x]_B^U) \geq \alpha,\ x \in X,\ X \in U/D\} \cup \{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X,\ X \in U/D\} \\
&= POS_B^{U'}(D) \cup \{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X,\ X \in U/D\}
\end{aligned}$$

From the above equation and the definition of local positive region, we have that

$$\begin{aligned}
|POS_{B\cup\{a\}}^{U'}(D)| - |POS_B^{U'}(D)| &= |POS_{B\cup\{a\}}^U(D)| - |\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X, X \in U/D\}| - |POS_B^{U'}(D)| \\
&= |POS_{B\cup\{a\}}^U(D)| - (|\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X, X \in U/D\}| + |POS_B^{U'}(D)|) \\
&= |POS_{B\cup\{a\}}^U(D)| - (|\{x \mid \mathcal{D}(X/[x]_B^U) = 1,\ x \in X, X \in U/D\} \cup POS_B^{U'}(D)|) \\
&= |POS_{B\cup\{a\}}^U(D)| - |POS_B^U(D)|
\end{aligned}$$

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 reduct of a target decision. Formally, a forward greedy attribute reduction algorithm for searching a local attribute reduct with respect to a given target decision can be written as follows.

Algorithm 4. A forward greedy local attribute reduction algorithm for a target decision (LARD)

```
Input: Decision table $S = (U, C \cup D)$ and the parameter $\alpha$;
Output: One reduct $red$.

Step 1: $red \leftarrow \emptyset$; //$red$ is the pool to conserve the selected attributes.
Step 2: Compute $Sig^{inner}(a_k, C, D, U)$, $k \leq |C|$; //$Sig^{inner}(a_k, C, D, U)$ is the inner significance of the attribute $a_k$.
Step 3: Put $a_k$ into $red$, where $Sig^{inner}(a_k, C, X, U) = \max\{Sig^{inner}(a_k, C, U, D),\ a_k \in C\}$;
```

Table 9: The time complexities of the LARD algorithm and the GARD algorithm

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

```
$Step$ 4: $i \leftarrow 1, R_1 \leftarrow red, U_1 = U, U_1/D = \{X_i^j,\ j \leq r\}$;
$Step$ 5: While $|POS_{red}^{U_i}(D)| < |POS_C^{U_i}(D)|$ Do //This provides a stopping criterion.
    {
    $U_{i+1} = \bigcup_{j=1}^{r}\{[x]_{red}^{U_i},\ x \in X_i^j, X_i^j \in U_i/D\} - \bigcup_{j=1}^{r}\{CL_{red}^{U_i}(X_i^j),\ X_i^j \in U_i/D\}$,//Compute the reduced universe.
    $X_{i+1}^j = X_i^j - CL_{red}^{U_i}(X_i^j),\ X_i^j \in U_i/D$,//Update every gradually reduced target concept.
    $i \leftarrow i + 1$,
    $red \leftarrow red \cup \{a_0\}$, where $Sig^{outer}(a_0, red, U_i, D) = max\{Sig^{outer}(a_k, red, U_i, D), a_k \in C - red\}\}$,
                                        //$Sig^{outer}(a_k, C, U_i, D)$ is the outer significance of the attribute $a_k$.
    $R_i \leftarrow R_{i-1} \cup \{a_0\}$.
    }
$Step$ 6: return $red$ and end.
```

In order to conveniently compare, we denote the algorithm of finding an attribute reduct of a target decision in the context of global rough sets 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 $O(|C|(\sum\limits_{j=1}^{r} |X^j|^2 + \sum\limits_{j=1}^{r} |X^j||U|))$. However, the time complexity of this step in global rough sets is $O(|C|(|U|^2 + \sum\limits_{j=1}^{r} |X^j||U|))$. Step 3 needs to scan the $|C|$ attributes, hence it time complexity is $O(|C|)$, and this step in global rough sets has the same case. In Step 5, we add an attribute with the maximal significance into the set in each stage until finding a reduct. This process is called a forward reduction algorithm whose time complexity is $O(\sum_{i=1}^{|C|}(|C| - i + 1))(\sum\limits_{j=1}^{r} |X_i^j|^2 + \sum\limits_{j=1}^{r} |X_i^j||U_i|))$. However, the time complexity of this step in a classical heuristic algorithm is $O(\sum_{i=1}^{|C|}(|C| - i + 1))(|U|^2 + \sum\limits_{j=1}^{r} |X^j||U|))$. 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 as Table 9.

It can be seen from Table 9 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.

5.2.3. Experimental analysis

In this experimental analysis, we want to verify the advantages of the LARD algorithm from three sides: 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 are often measured by the accuracy of approximation in Eq. (9). Given a decision table $S = (U, C \cup D)$, $X_1, X_2, \cdots, X_r \in U/D$ are r classes with labels, $P = \{R_1, R_2, \cdots, R_n\}$ a family of attributes with $R_1 \succeq R_2 \succeq \cdots \succeq R_n$

Table 10: The number of attributes in attribute reducts of LARD and GARD

Data sets	Original features	LARD algorithm			GARD algorithm		
		Selected features			Selected features		
		$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
Mushroom	22	18	18	18	22	18	18
Tic-tac-toe	8	8	8	8	8	8	8
Dermatology	34	6	6	6	6	6	6
Kr-vs-kp	36	31	31	31	36	31	31
Breast-cancer-wisconsin	9	9	9	9	9	9	9
Backup-large.test	35	6	6	6	5	6	6
Shuttle	9	8	8	8	8	8	8
Letter-recognition	16	16	16	16	16	16	16
Ticdata2000	85	24	24	24	24	24	24

($R_i \in 2^{AT}$). Let $P_i = \{R_1, R_2, \cdots, R_i\}$, we denote the accuracy of approximation of D with respect to P_i by $\gamma(P_i, D) = \frac{\sum\{|\underline{R_{P_i}}(X)|:\ X \in U/D\}}{\sum\{|X|:\ X \in U/D\}} = \frac{|POS_{R_i}(D)|}{\sum\{|X|:\ X \in U/D\}}$, where $X \in U/D$ means each of those classes with labels.

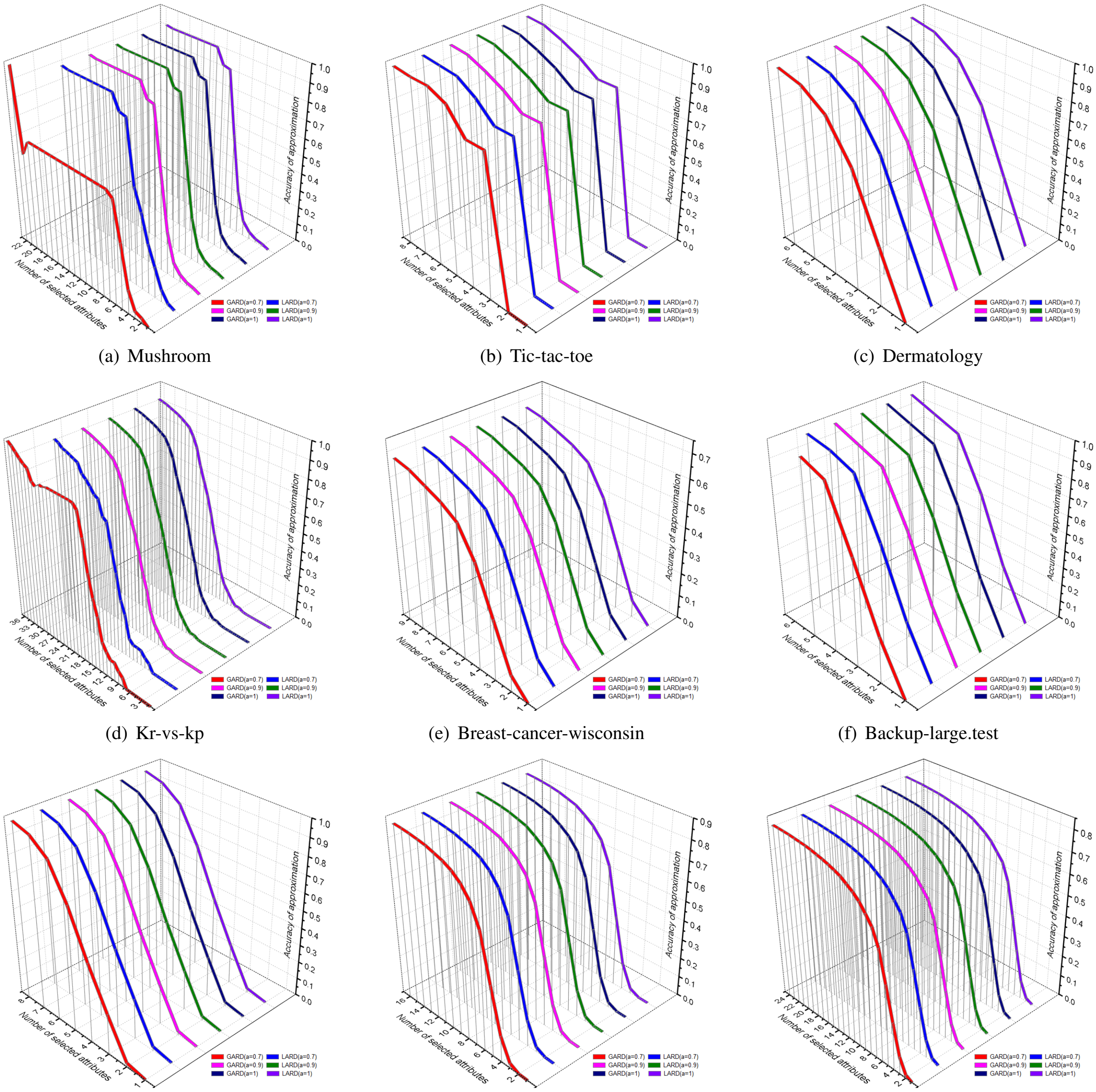
In this experiment, we observe both the quality of approximation and the number of selected attributes in the forward searching process for the LARD algorithm and the GARD algorithm. In general, if the quality of approximation monotonically increases and the number of attributes in an attribute reduct induced an attribute reduction algorithm is much smaller, we can say that the algorithm may be much better. It is because that much smaller number of attributes may induce a much better classifier. For the experiment, we take the front 10% objects from each of the nine data sets as its corresponding target decision. Fig. 6 shows the quality of approximation of D used by each of the LARD algorithm and the GARD algorithm and show it vis-a-vis the number of attributes, where the parameter is assigned as $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively. In each of these sub-figures, the x-coordinate pertains to the number of the attributes, while the y-coordinate concerns the value of precision of approximation. Table 10 displays the number of attributes in attribute reducts obtained by the LARD algorithm and the GARD algorithm with the same parameter value, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively.

It is easy to see from Fig. 6 that the accuracy of approximation of the LARD algorithm monotonically increases with the number of attributes adding on each of the nine data sets. However, in subfigures (a) and (d), we can see that the accuracy of approximation of the GARD algorithm is not monotonic as the number of attributes is added. The GARD algorithm needs to search much more attributes to achieve the same value of accuracy of approximation, which may cause over-fitting in attribute reduction of a target decision. From Table 10, we also can see that the number of attributes selected by the LARD algorithm is usually not more than that selected by the GARD algorithm with the same parameter value on the same data set. For example, the GARD algorithm may lead to over-fitting phenomenon of an attribute reduct of a target decision on Mushroom and Kr-vs-kp. Like Fig. 4, the values of accuracy of approximation coming from the GARD algorithm may be also over one, which could cause over-fitting in attribute reduction. This is because that global positive region of a target decision may be beyond its own region. From these findings, we can say that the LARD algorithm under the local rough set may be a good solution for reducing the over-fitting degree in attribute reduction for a target decision.

- Efficiency of the LARD algorithm

To verify the efficiency of the LARD algorithm, we still take the same experimental setting as Section 4.1.2. In this experimental analysis, we fix the size of target decision (the front 10% objects from each of these nine data sets) to compare computational time of the LARD algorithm with the GARD algorithm vis-a-vis the size of universe, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively. The experimental results of these nine data sets are shown in Fig. 7 and Table 11. This figure displays more detailed change trend of each of two algorithms (LARD and GARD) with size of data set becoming increasing.

It is easy to note from Figure 7 that the computing time of each of these two algorithms increases with the increase of the size of data. From the two subfigures in Fig. 7, we also can see that for each value of parameter α, the LARD



(a) Mushroom

(b) Tic-tac-toe

(c) Dermatology

(d) Kr-vs-kp

(e) Breast-cancer-wisconsin

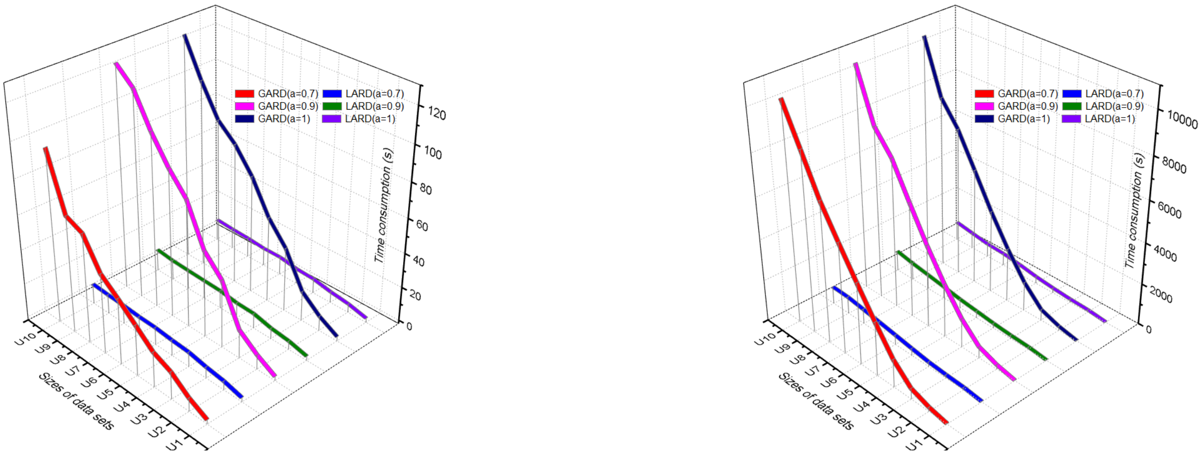
(f) Backup-large.test

(g) Shuttle

(h) Letter-recognition

(i) Ticdata2000

Figure 6: Accuracy of approximation of a target decision versus number of attributes



(a) Mushroom

(b) Shuttle

Figure 7: Times of LARD and GARD versus the size of the universe on Mushroom and Shuttle

Table 11: The computational times for attribute reductions with LARD and GARD

Data sets	α = 0.7		α = 0.9		α = 1	
	LARD	GARD	LARD	GARD	LARD	GARD
Mushroom	10.1489	97.5596	11.1063	126.2411	10.7848	127.4461
Tic-tac-toe	0.1494	0.9330	0.1457	0.8947	0.1517	0.9083
Dermatology	0.1773	0.7636	0.1732	0.7915	0.1769	0.8007
Kr-vs-kp	9.5103	113.0699	10.8262	189.1183	10.4646	185.6964
Breast-cancer-wisconsin	0.1648	0.4549	0.1628	0.5291	0.1603	0.4801
Backup-large.test	0.1370	0.4282	0.1446	0.3965	0.1470	0.4410
Shuttle	869.9849	10404.8801	1008.1572	10675.9573	924.8207	10725.3392
Letter-recognition	211.2411	3235.1833	185.2882	3394.3294	183.4979	3458.2517
Ticdata2000	84.3644	2525.6332	83.6232	2512.7273	81.5344	2595.8942

Table 12: Classification accuracies of classifiers induced by attribute reductions with LARD and GARD (α = 0.5)

Data sets	SVM		NN		C4.5	
	LARD	GARD	LARD	GARD	LARD	GARD
Mushroom	0.9991 (5)	0.6357 (1)	0.9998 (5)	0.6357 (1)	0.9993 (5)	0.6354 (1)
Tic-tac-toe	0.7432 (7)	0.6534 (1)	0.8580 (7)	0.6534 (1)	0.8351 (7)	0.6534 (1)
Dermatology	0.6285 (5)	0.5224 (4)	0.6089 (5)	0.4330 (4)	0.6089 (5)	0.4553 (4)
Kr-vs-kp	0.9584(30)	0.5222 (1)	0.9687(30)	0.5222 (1)	0.9944(30)	0.5222 (1)
Breast-cancer-wisconsin	0.9634 (4)	0.8551 (1)	0.9532 (4)	0.8594 (1)	0.9561 (4)	0.8521 (1)
Backup-large.test	0.8085 (8)	0.6862 (5)	0.8059 (8)	0.8271 (5)	0.8431 (8)	0.8617 (5)
Shuttle	0.9632 (4)	0.9211 (2)	0.9994 (4)	0.9540 (2)	0.9996 (4)	0.9542 (2)
Letter-recognition	0.7307 (9)	0.6169 (7)	0.9212 (9)	0.8333 (7)	0.8691 (9)	0.8090 (7)
Ticdata2000	0.9402(42)	0.9402 (1)	0.8985(42)	0.9402 (1)	0.9387(42)	0.9402 (1)

algorithm is consistently faster than the classical GARD algorithm on the same universe. In addition, we also can see that the differences between these two algorithms for time consumption are profoundly larger when the size of the data set increases. As one of the important advantages of the LARD algorithm, it can be seen from Table 11 that the time reduction performance of the LARD algorithm is very obvious compared to the GARD algorithm. For example, the LARD algorithm only takes about 1/30 computational time of the GARD algorithm on the data set Ticdata2000. One can say that for computing the attribute reduct of a target decision, the LARD algorithm under the local rough set provides an very efficient solution in the context of limited labeled big data.

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

In this experiment, we want to classification quality of induced by attribute reduction of the LARD algorithm comparing with those of the GARD algorithm. As we know, the same attribute reduct must have the same classification accuracy for a given classifier. From Table 10, it can be seen that these two algorithms obtained almost the same attribute reducts when $\alpha = 0.7$, 0.9 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 reducts as the parameter α becoming much smaller. In the following, let $\alpha = 0.5$, we observe the results of attribute reduction with the LARD algorithm and the GARD algorithm and their classification accuracies through employing SVM, NN and C4.5 (coming from Weka 3.6.10), which are shown in Table 12.

In Table 12, $(\cdot)$ means the number of attributes in an attribute reduct obtained by an algorithm. It is easy to see from Table 12 attribute reduct induced by the LARD algorithm is almost consistently bigger than that induced by the GARD algorithm on the same data set. In addition, we also can see that for each of classifiers SVM, NN and C4.5, the corresponding classification accuracies of classifiers induced by attribute reduction of the LARD algorithm are almost higher than those of the GARD algorithm [3]. This implies that for a given α, the proposed method/approach allow us to obtain attribute reducts at least as good as the existing methods.

6. 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. An artificial data set is used in this experiment. This data set has $10,000,000$ (10^7) objects and 6 attributes, where 100 objects were labeled.

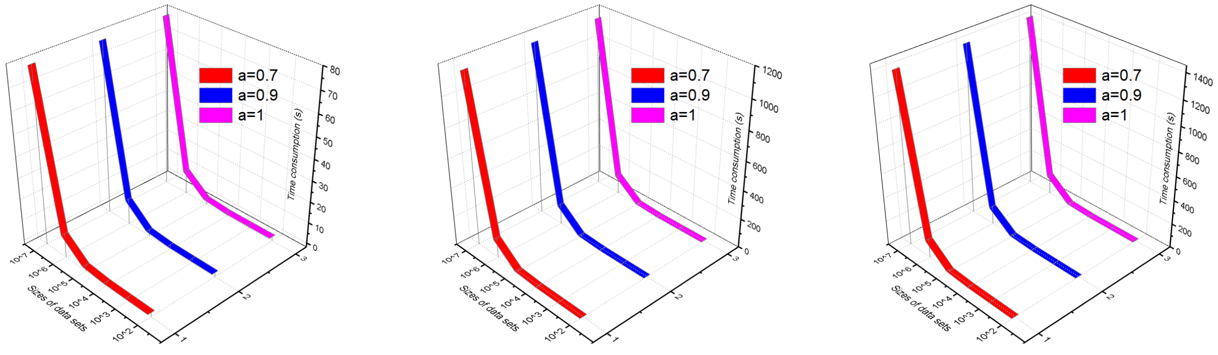
We tested scalability of the three algorithms (LLAC, LARC and LARD) in the local rough set on this large data set, which is the scalability against the number of objects for a given target concept and a given set of labeled objects. Fig. 8(a) shows the computational times of the LLAC algorithm calculating local lower approximations on different numbers of objects, Fig. 8(b) displays computational times of the LARC algorithm to search local attribute reducts of a given target concept on different numbers of objects, and Fig. 8(c) shows those of the LARD algorithm for finding local attribute reducts of a given decision on different numbers of objects, where the parameter $\alpha = 0.7$, $\alpha = 0.9$ and $\alpha = 1$, respectively.

One important observation from these figures is that the run time of the three 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 10^{i+1} objects is almost 10 times longer than that of this algorithm on the data set with 10^i objects. This observation is consistent with the linear time complexity of each of these three 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.

7. Conclusions

With the advent of the age of big data, the number of obtained objects in databases becomes larger and larger, while labeling a large amount of data is an expensive and laborious task and sometimes even infeasible. As a supervised

[3] For Ticdata2000 datasets, we obtain 24 attributes from the original 84 attributes when $\alpha = 1$, which suggests there are lots of redundant attributes. Furthermore, we find we can get almost same accuracy (94%) for lots of single attribute. Hence, we conjecture the result is caused by heavily attributes redundance.



(a) LLAC (b) LARC (c) LARD

Figure 8: Times of LLAC, LARC and LARD versus sizes of data sets (s)

machine learning method, classical rough set theory mainly face three challenges for rough set analysis in big data, which are semi-supervised property of big data, computational inefficiency and over-fitting in attribute reduction.

To address the challenges of classical rough set theory, in this study, a theoretic framework called local rough set has been proposed and some of its important properties have been also given. Unlike the classical rough set theory, the local rough set does not obtain information granules of all objects in a given data set beforehand, but only compute those of objects coming from a target concept. Based on this framework, an algorithm for computing a local lower approximation of a target concept (LLAC) and an algorithm for searching a local attribute reduct of a target concept (LARC) have been proposed. Each of these two algorithms has a linear time complexity, which can significantly improve computational performance. Furthermore, we have designed the LLAD algorithm for calculating a local lower approximation of a target decision and the LARD algorithm for finding a local attribute reduct of a target decision. To highlight advantages of the local rough set, we have also employed nine real data sets and an artificial data set for verifying the performance of these four algorithms. Experiment results have shown 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 visible when dealing with larger data sets. Hence, the local rough set can be regarded as an effective solution for rough data analysis in big data.

Within the broad area of local rough set, this study delivers preliminary albeit interesting and promising results. In future works, there are many interesting and important issues to be investigated. For instance, extension of local rough set with various binary relations, attribute reduction, rough classifiers with semi-supervised learning, multigranulation local rough set, and their applications. The studies along these lines would significantly promote the research of rough data analysis in big data.

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 Polish National Science Centre (NCN) grants (DEC- DEC-2011/01/D /ST6/06981, DEC-2012/05/B/ ST6/03215, DEC-2013/09/B/ST6/01568) as well as by the Polish National Centre for Research and Development (NCBiR) under the grant (OROB/0010/03/001)

[1] Ben-David, A., Sterling, L., Tran, T. D., 2009. Adding monotonicity to learning algorithms may impair their accuracy. Expert Systems with Applications 36 (3), 6627–6634.

[2] Bhatt, R. B., Gopal, M., 2005. On fuzzy-rough sets approach to feature selection. Pattern Recognition Letters 26 (7), 965–975.

[3] Dntsch, I., Gediga, G., 1998. Uncertainty measures of rough set prediction. Artificial Intelligence 106 (1), 109–137.

[4] Dubois, D., Prade, H., 1990. Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 191–209.

[5] Eskandari, S., Javidi, M. M., 2016. Online streaming feature selection using rough sets. International Journal of Approximate Reasoning 69, 35–57.

[6] Gediga, G., Dntsch, I., 2001. Rough approximation quality revisited . Artificial Intelligence 132 (2), 219–234.

[7] Greco, S., Matarazzo, B., Slowinski, R., 1999. Rough approximation of a preference relation by dominance relations. European Journal of Operational Research 117 (1), 63–83.

[8] Grzymala-Busse, J. W., 1992. LERS-A System for Learning from Examples Based on Rough Sets. Springer Netherlands.

[9] Guyon, I., Elisseeff, A., 2003. An introduction to variable feature selection. Journal of Machine Learning Research 3, 1157–1182.

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

[11] Hu, Q., Yu, D., Liu, J., Wu, C., 2008. Neighborhood rough set based heterogeneous feature subset selection. Information Sciences 178 (18), 3577–3594.

[12] Hu, X., Cercone, N., 1995. Learning in relational databases: A rough set approach. Computational Intelligence 11 (2), 323–338.

[13] Jensen, R., Shen, Q., 2008. Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. Wiley-IEEE Press.

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

[15] Kryszkiewicz, M., 1998. Rough set approach to incomplete information systems. Information Sciences 112 (14), 39–49.

[16] Kryszkiewicz, M., 1999. Rules in incomplete information systems. Information Sciences 113 (34), 271–292.

[17] Lezak, D., Ziarko, W., 2005. The investigation of the bayesian rough set model. International Journal of Approximate Reasoning 40 (1), 81–91.

[18] Liang, D., Liu, D., Kobina, A., 2016. Three-way group decisions with decision-theoretic rough sets. Information Sciences 345, 46–64.

[19] Liang, J., Wang, F., Dang, C., Qian, Y., 2012. An efficient rough feature selection algorithm with a multi-granulation view. International Journal of Approximate Reasoning 53 (6), 912–926.

[20] Liang, J., Wang, F., Dang, C., Qian, Y., 2013. A group incremental approach to feature selection applying rough set technique. IEEE Transactions on Knowledge & Data Engineering 26 (2), 294–308.

[21] Lichma, M., 2013. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, university of California, Irvine, School of Information and Computer Sciences.

[22] Lin, T. Y. T., 2000. Data mining and machine oriented modeling: A granular computing approach. Applied Intelligence 13 (2), 113–124.

[23] Liu, D., Li, T., Liang, D., 2014. Incorporating logistic regression to decision-theoretic rough sets for classifications. International Journal of Approximate Reasoning 55 (1), 197–210.

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

[25] Pavlenko, T., 2001. On feature selection, curse-of-dimensionality and error probability in discriminant analysis. Journal of Statistical Planning & Inference 115 (2), 565–584.

[26] Pawlak, Z., 1982. Rough sets. International Journal of Parallel Programming 11 (5), 341–356.

[27] Pedrycz, W., 2013. Granular Computing: Analysis and Design of Intelligent Systems. CRC Press.

[28] Pedrycz, W., Bargiela, A., 2002. Granular clustering: a granular signature of data. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society 32 (2), 212–224.

[29] Pedrycz, W., Vukovich, G., 2002. Feature analysis through information granulation and fuzzy sets. Pattern Recognition 35 (4), 825–834.

[30] Polkowski, L., Skowron, A., 1996. Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15 (4), 333–365.

[31] Qian, Y., Cheng, H., Wang, J., Liang, J., Pedrycz, W., Dang, C., 2017. Grouping granular structures in human granulation intelligence. Information Sciences 382383, 150–169.

[32] Qian, Y., Li, S., Liang, J., Shi, Z., Wang, F., 2014. Pessimistic rough set based decisions: A multigranulation fusion strategy . Information Sciences 264 (6), 196–210.

[33] Qian, Y., Liang, J., Dang, C., 2010. Incomplete multigranulation rough set. IEEE Transactions on Systems, Man and Cybernetics: Part A 40 (2), 420–431.

[34] Qian, Y., Liang, J., Pedrycz, W., Dang, C., 2010. Positive approximation: An accelerator for attribute reduction in rough set theory. Artificial Intelligence 174 (9), 597–618.

[35] Qian, Y., Liang, J., Pedrycz, W., Dang, C., 2011. An efficient accelerator for attribute reduction from incomplete data in rough set framework. Pattern Recognition 44 (8), 1658–1670.

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

[37] Qian, Y., Zhang, H., Li, F., Hu, Q., Liang, J., 2014. Set-based granular computing: A lattice model. International Journal of Approximate Reasoning 55 (3), 834–852.

[38] She, Y., He, X., 2012. On the structure of the multigranulation rough set model. Knowledge-Based Systems 36 (6), 81–92.

[39] Skowron, A., 1995. Extracting laws from decision tables: A rough set approach. Computational Intelligence 11 (2), 371–388.

[40] Skowron, A., Stepaniuk, J., 1996. Tolerance approximation spaces. Fundamenta Informaticae 27 (2,3), 245–253.

[41] Sun, B., Ma, W., Xiao, X., 2016. Three-way group decision making based on multigranulation fuzzy decision-theoretic rough set over two universes. International Journal of Approximate Reasoning 81, 87–102.

[42] Swiniarski, R. W., Skowron, A., 2003. Rough set methods in feature selection and recognition. Pattern Recognition Letters 24 (6), 833–849.

[43] Wang, G. Y., Yu, H., Yang, D. C., 2002. Decision table reduction based on conditional information entropy. Chinese Journal of Computers 25 (7), 759–766.

[44] Wang, G. Y., Zhao, J., An, J., Wu, Y., 2005. A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fundamenta Informaticae 68 (3), 289–301.

[45] Wu, W. Z., Zhang, M., Li, H. Z., Mi, J. S., 2005. Knowledge reduction in random information systems via dempster-shafer theory of evidence. Information Sciences 174 (3-4), 143–164.

[46] Xu, J., Miao, D., Zhang, Y., Zhang, Z., 2017. A three-way decisions model with probabilistic rough sets for stream computing. International Journal of Approximate Reasoning 88, 1–22.

[47] Yan, X. Z., Beijng, 2006. A quick attribute reduction algorithm with complexity of $max(o(|c||u|), o(|c|\ 2|u/c|))$. Chinese Journal of Computers 29 (3), 391–399.

[48] Yao, Y., 2008. Probabilistic rough set approximations. International Journal of Approximate Reasoning 49 (2), 255–271.

[49] Yao, Y., 2010. Three-way decisions with probabilistic rough sets. Information Sciences 180 (3), 341–353.

[50] Yao, Y., 2011. The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181 (6), 1080–1096.

[51] Yao, Y. Y., 2003. Probabilistic approaches to rough sets. Expert Systems 20 (5), 287–297.

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

[53] Zhang, W. X., Leung, Y., 1996. Theory of including degrees and its applications to uncertainty inferences. International Journal of Approximate Reasoning, 496–501.

[54] Zhang, X., Miao, D., 2013. Two basic double-quantitative rough set models of precision and grade and their investigation using granular computing. International Journal of Approximate Reasoning 54 (8), 1130–1148.

[55] Ziarko, W., 1993. Variable precision rough set model. Journal of Computer & System Sciences 46 (1), 39–59.