PLAR: Parallel Large-scale Attribute Reduction on Cloud Systems

Junbo Zhang[1,2], Tianrui Li[1], Yi Pan[2]
[1]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China*
[2]*Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA*
Email: JunboZhang86@163.com, trli@swjtu.edu.cn, pan@cs.gsu.edu

***Abstract*—Attribute reduction for big data is viewed as an important preprocessing step in the areas of pattern recognition, machine learning and data mining. In this paper, a novel parallel method based on MapReduce for large-scale attribute reduction is proposed. By using this method, several representative heuristic attribute reduction algorithms in rough set theory have been parallelized. Further, each of the improved parallel algorithms can select the same attribute reduct as its sequential version, therefore, owns the same classification accuracy. An extensive experimental evaluation shows that these parallel algorithms are effective for big data.**

***Keywords*-Attribute Reduction, Rough Set Theory, MapReduce, Big Data**

I. Introduction

In pattern recognition, machine learning and data mining, attribute reduction, also known as feature selection, is the technique of preprocessing data and has been attracted much attention in recent years [1], [2], [3], [4], [5], [6]. Due to the development of information acquirement and storage, a large number of features are acquired and stored for wide real-world applications, such as text mining and biology problems. An excessive amount of features will make the program too time consuming. Some of them may even guide the program to make wrong decisions. The purpose of feature selection is to improve the accuracy and performance of classifiers through removing most irrelevant and redundant features from the data. It also helps people to acquire better understanding about the data by telling them which are relevant features.

Rough set theory is a relatively new soft computing tool for dealing with inconsistent information in decision situations and plays an important role in the fields of pattern recognition, feature selection and knowledge discovery [7]. Attribute reduction in rough set theory provides a theoretic framework for consistency-based feature selection, which will retain the discernible ability of original features for the objects from the universe [3].

Attribute reduction from large data is an expensive preprocessing step, to accelerate this process, incremental techniques combined with traditional rough set based methods are widely researched [8], [9], [5], [3], [10], [11]. For example, Li et al. proposed an incremental method for dynamic attribute generalization, which can accelerate a heuristic process of attribute reduction by updating approximations incrementally. Qian et al. introduced positive approximation, which is a theoretic framework for accelerating a heuristic process [3]. Zhang et al. presented a matrix-based incremental method for fast computing rough approximations [9].

As these methods are still sequential and can not process big data with a cluster. MapReduce, by Google, is a popular parallel programming model and a framework for processing big data on certain kinds of distributable problems using a large number of computers, collectively referred to as a cluster [12]. It can help arrange the application in the cluster easily. Some MapReduce runtime systems were implemented, such as Hadoop [13], Twister [14], Phoenix [15] and Mars [16], which all can help developers to parallelize traditional algorithms by using MapReduce model. For example, Apache Mahout is machine learning libraries, and produces implementations of parallel scalable machine learning algorithms on Hadoop platform by using MapReduce [17].

In previous work, we developed the parallel algorithm for computing rough set approximations based on MapReduce [18]. Based on that, we proposed parallel rough set based knowledge acquisition using MapReduce [19], [20]. Further, in this paper, we present a parallel method based on MapReduce for attribute reduction. Four representative attribute reduction algorithms are parallelized with this method. The corresponding parallel large-scale attribute reduction algorithms are proposed and implemented with Hadoop [13]. Users not only deploy Hadoop on local cluster easily, but also on public cloud systems, such as Amazon EC2 and Microsoft Azure both support Hadoop through Amazon Elastic MapReduce [21] and HadoopOnAzure [22], respectively. The experimental results on the large data set KDD99 and the synthetic data set show that the proposed parallel method have a good speedup and can help improve classification accuracy.

The rest of this paper is organized as follows. Section II includes the elementary background introduction to MapReduce and attribute reduction based on rough sets. Section III proposes the simplification and decomposition of evaluation functions, and parallel methods based on MapReduce. Section IV presents the experimental analysis. The paper ends with conclusions and future work in Section V.

978-1-4799-2419-6/13 $31.00 © 2013 IEEE
DOI 10.1109/PDCAT.2013.36

IEEE computer society

II. PRELIMINARIES

In this section, we briefly review the parallel model MapReduce [12], some basic concepts, notations and results of rough sets and attribute reduction algorithms [7], [23].

A. MapReduce model

MapReduce is a parallel model first introduced by Google [12]. It is designed to handle and generate big data in distributed environment. It provides a convenient way to parallelize data analysis process. Its advantages include conveniences, robustness and scalability. MapReduce helps to split the large input data into many small blocks and assign small tasks to different devices. In the cluster, MapReduce model usually works with the distributed file system [24].

B. Rough sets

Given a pair $K = (U, R)$, where U is a finite and non-empty set called the universe, and $R \subseteq U \times U$ is an equivalence relation on U. The pair $K = (U, R)$ is called an approximation space. The equivalence relation R partitions the set U into several disjoint subsets. This partition of the universe forms a quotient set induced by R, denoted by U/R. If two elements $x, y \in U (x \neq y)$, are indistinguishable under R, we say x and y belong to the same equivalence class. The equivalence class including x is denoted by $[x]_R$.

An approximation space $K = (U, R)$ is characterized by an information system $S = (U, A)$, where $U = \{x_1, x_2, \cdots, x_n\}$ is a non-empty finite set of objects, called a universe. $A = \{a_1, a_2, \cdots, a_m\}$ is a non-empty finite set of attributes (features). Specifically, $S = (U, A)$ is called a decision table if $A = C \cup D$, where C is a set of condition attributes and D is a set of output or decision results, $C \cap D = \emptyset$.

Definition 1: Let $B = \{b_1, b_2, \cdots, b_l\} \subseteq C$ be a subset of condition attributes. The information set with respect to B for any object $x \in U$ can be denoted by

$$\overrightarrow{x}_B =< f(x, b_1), f(x, b_2), \cdots, f(x, b_l) > \quad (1)$$

An equivalence relation with respect to B called the indiscernibility relation, denoted by $IND(B)$, is defined as

$$IND(B) = \left\{(x, y)|(x, y) \in U \times U, \overrightarrow{x}_B = \overrightarrow{y}_B\right\} \quad (2)$$

Two objects x, y satisfying the relation $IND(B)$ are indiscernible by attributes from B.

The equivalence relation $IND(B)$ partitions U into some equivalence classes given by:

$$U/IND(B) = \{[x]_B|x \in U\} \quad (3)$$

where $[x]_B$ denotes the equivalence class determined by x with respect to B, $[x]_B = \{y \in U|(x, y) \in IND(B)\}$. For simplicity, $U/IND(B)$ will be replaced by U/B.

Definition 2: Let $B \subseteq C$ be a subset of condition attributes. The information set with respect to B for any $E \in U/B$ is denoted by

$$\overrightarrow{E}_B = \overrightarrow{x}_B, x \in E \quad (4)$$

Definition 3: Let $X \subseteq U$, and R be an equivalence relation. $U/R = \{E_1, E_2, \cdots, E_t\}$. The lower and upper approximations of X are defined as

$$\begin{cases} \underline{R}(X) = \{x \in U|[x]_R \subseteq X\} \\ \overline{R}(X) = \{x \in U|[x]_R \cap X \neq \emptyset\} \end{cases} \quad (5)$$

Definition 4: Given a decision table $S = (U, C \cup D)$. $U/D = \{Y_1, Y_2, \cdots, Y_n\}$ called the set of decision classes, means that the object set U is partitioned into n mutually exclusive crisp subsets by the decision attributes D. Given any subset $B \subseteq C$ and $IND(B)$ is the equivalence relation induced by B, then one can define the lower and upper approximations of the decision D as

$$\begin{cases} \underline{R}_B(D) = \{\underline{R}_B(Y_1), \underline{R}_B(Y_2), \cdots, \underline{R}_B(Y_n)\} \\ \overline{R}_B(D) = \{\overline{R}_B(Y_1), \overline{R}_B(Y_2), \cdots, \overline{R}_B(Y_n)\} \end{cases} \quad (6)$$

We denote $POS_B(D) = \bigcup_{Y_i \in U/D} \underline{R}_B(Y_i)$ which is called as a positive region of D with respect to the condition attribute set B. It is the set of all elements of U that can be uniquely classified to blocks of the partition U/D by means of B.

C. Forward attribute reduction in rough set theory

Each attribute reduction method preserves a particular property of a given information system, which is based on a certain predetermined heuristic function. In rough set theory, attribute reduction is to find some feature subsets that have the minimal features and retain some particular properties. In the forward greedy attribute reduction methods, two important measures of attributes are used for heuristic functions, which are inner importance measure and outer importance measure [3]. The inner importance measure is applicable to determine the significance of every attribute, while the outer importance measure can be used in a forward feature selection. It is deserved to point out that each kind of attribute reduction tries to preserve a particular property of a given decision table.

Fig. 1 shows the process of attribute reduction methods in sequential, where ε is a small positive real number used to control the convergence.

D. Four representative significance measures of attributes

For efficient attribute reduction, many heuristic attribute reduction methods have been developed in rough set theory, see [25], [26], [27], [28]. Further, from the viewpoint of heuristic functions, Qian et al. classified these attribute reduction methods into four categories: positive-region reduction, Shannon's conditional entropy reduction, Liang's

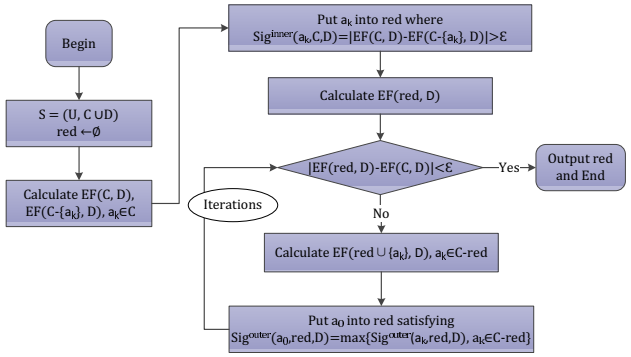


Figure 1. Process of the sequential method

conditional entropy reduction and combination conditional entropy reduction [3]. We here also only focus on these four representative attribute reduction methods.

Given a decision table $S = (U, C \cup D)$, $B \subseteq C$, the condition partition $U/B = \{X_1, X_2, \cdots, X_m\}$ and the decision partition $U/D = \{Y_1, Y_2, \cdots, Y_n\}$ can be obtained. Through these notations, we briefly review four types of significance measures of attributes as follows.

1) Positive-region based method (PR):

Definition 5: Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$, the dependency degree of D to B is defined as

$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}$,

where $POS_B(D) = \bigcup_{i=1}^{m} \{X_i \in U/B : X_i \subseteq Y_1 \vee X_i \subseteq Y_2 \vee \cdots \vee X_i \subseteq Y_n\} = \bigcup_{i=1}^{m} \{X_i \in U/B : |X_i/D| = 1\}$.

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

$Sig_{PR}^{inner}(a, B, D) = \gamma_B(D) - \gamma_{B-\{a\}}(D)$.

Definition 7: (PR-outer) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in C - B$. The significance measure of a in B is defined as

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

2) Shannon's conditional entropy based method (SCE):

Definition 8: Shannon's conditional entropy of D with respect to B is defined as

$H(D|B) = -\sum_{i=1}^{m} p(X_i) \sum_{j=1}^{n} p(Y_j|X_i) \log(p(Y_j|X_i))$,

where $p(X_i) = \frac{|X_i|}{|U|}$ and $p(Y_j|X_i) = \frac{|X_i \cap Y_j|}{|X_i|}$

Definition 9: (SCE-inner) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in B$. The significance measure of a in B is defined as

$Sig_{SCE}^{inner}(a, B, D) = H(D|B - \{a\}) - H(D|B)$.

Definition 10: (SCE-outer) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in C - B$. The significance measure of a in B is defined as

$Sig_{SCE}^{outer}(a, B, D) = H(D|B) - H(D|B \cup \{a\})$.

3) Liang's conditional entropy based method (LCE):

Definition 11: Liang's conditional entropy of D with respect to B is defined as

$E(D|B) = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|Y_j \cap X_i|}{|U|} \frac{|Y_j^c - X_i^c|}{|U|}$.

Definition 12: (LCE-inner) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in B$. The significance measure of a in B is defined as

$Sig_{LCE}^{inner}(a, B, D) = E(D|B - \{a\}) - E(D|B)$.

Definition 13: (LCE-outer) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in C - B$. The significance measure of a in B is defined as

$Sig_{LCE}^{outer}(a, B, D) = E(D|B) - E(D|B \cup \{a\})$.

4) Combination conditional entropy based method (CCE):

Definition 14: Combination conditional entropy of D with respect to B is defined as

$CE(D|B) = \sum_{i=1}^{m} (\frac{|X_i|}{|U|} \frac{C_{|X_i|}^2}{C_{|U|}^2} - \sum_{j=1}^{n} \frac{|X_i \cap Y_j|}{|U|} \frac{C_{|X_i \cap Y_j|}^2}{C_{|U|}^2})$,

where $C_{|X_i|}^2 = \frac{|X_i| \times (|X_i| - 1)}{2}$ denotes the number of the pairs of the objects which are not distinguishable each other in the equivalence class X_i.

Definition 15: (CCE-inner) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in B$. The significance measure of a in B is defined as

$Sig_{CCE}^{inner}(a, B, D) = CE(D|B - \{a\}) - CE(D|B)$.

Definition 16: (CCE-outer) Let $S = (U, C \cup D)$ be a decision table, $B \subseteq C$ and $\forall a \in C - B$. The significance measure of a in B is defined as

$Sig_{CCE}^{outer}(a, B, D) = CE(D|B) - CE(D|B \cup \{a\})$.

Intuitively, these four significance measures of attributes are listed in Table I.

III. Parallelization

From description of attribute reduction algorithms, the most important and critical step is to calculate value of evaluation function (*e.g.*, conditional entropy). Hence, by parallelizing this step, attribute reduction algorithms can be run in parallel. Hence, we parallelize three marked steps in Fig 2, which are different with the sequential method. The detailed processes are described below in the next few subsections.

A. Simplification and decomposition of evaluation functions

In this subsection, to calculate the value of evaluation functions in parallel, we simplify and decompose all four evaluation functions as follows.

Given a decision table $S = (U, C \cup D)$, $B \subseteq C$, $U/B = \{X_1, X_2, \cdots, X_m\}$ and $U/D = \{Y_1, Y_2, \cdots, Y_n\}$ are condition and decision partitions, respectively.

(1) Decomposition of PR Evaluation Function

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|} = \sum_{i=1}^{m} \frac{|X_i| sgn_{PR}(X_i)}{|U|}$$

Table I
FOUR REPRESENTATIVE SIGNIFICANCE MEASURES OF ATTRIBUTES

	Evaluation Function (Stop Criterion)	$Sig_{\Delta}^{inter}(a,B,D)$ $(\forall a \in B)$	$Sig_{\Delta}^{outer}(a,B,D)$ $(\forall a \in C-B)$
PR	$\gamma_B(D) = \frac{\lvert POS_B(D)\rvert}{\lvert U\rvert}$	$\gamma_B(D) - \gamma_{B-\{a\}}(D)$	$\gamma_{B\cup\{a\}}(D) - \gamma_B(D)$
SCE	$H(D\lvert B) = -\sum_{i=1}^{m} p(X_i) \sum_{j=1}^{n} p(Y_j\lvert X_i)\log(p(Y_j\lvert X_i))$	$H(D\lvert B-\{a\}) - H(D\lvert B)$	$H(D\lvert B) - H(D\lvert B\cup\{a\})$
LCE	$E(D\lvert B) = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\lvert Y_j\cap X_i\rvert}{\lvert U\rvert}\frac{\lvert Y_j^c - X_i^c\rvert}{\lvert U\rvert}$	$E(D\lvert B-\{a\}) - E(D\lvert B)$	$E(D\lvert B) - E(D\lvert B\cup\{a\})$
CCE	$CE(D\lvert B) = \sum_{i=1}^{m}\left(\frac{\lvert X_i\rvert}{\lvert U\rvert}\frac{C^2_{\lvert X_i\rvert}}{C^2_{\lvert U\rvert}} - \sum_{j=1}^{n}\frac{\lvert X_i\cap Y_j\rvert}{\lvert U\rvert}\frac{C^2_{\lvert X_i\cap Y_j\rvert}}{C^2_{\lvert U\rvert}}\right)$	$CE(D\lvert B-\{a\}) - CE(D\lvert B)$	$CE(D\lvert B) - CE(D\lvert B\cup\{a\})$

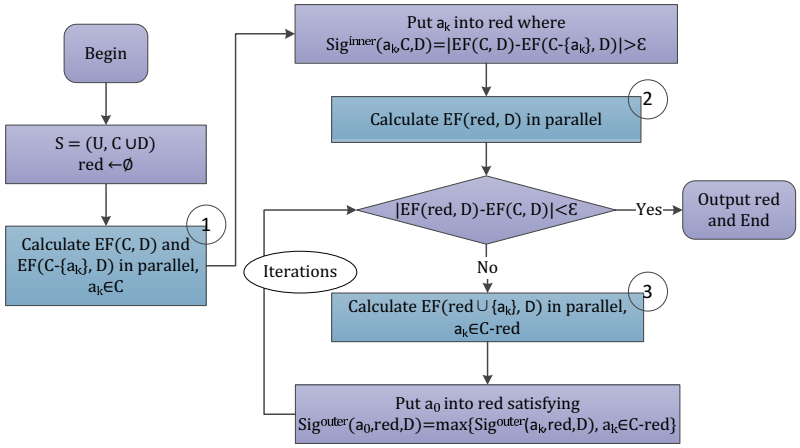


Figure 2. Process of the parallel method

where $sgn_{PR}(X_i) = \begin{cases} 1, & |X_i/D| = 1 \\ 0, & else \end{cases}$.

(2) Decomposition of SCE Evaluation Function

$$\begin{aligned} H(D|B) &= -\sum_{i=1}^{m} p(X_i) \sum_{j=1}^{n} p(Y_j|X_i)\log(p(Y_j|X_i)) \\ &= \sum_{i=1}^{m}\left[-\frac{1}{|U|}\sum_{j=1}^{n}|X_i\cap Y_j|\log(\frac{|X_i\cap Y_j|}{|X_i|})\right] \end{aligned}$$

(3) Decomposition of LCE Evaluation Function

$$\begin{aligned} E(D|B) &= \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{|Y_j\cap X_i|}{|U|}\frac{|Y_j^c - X_i^c|}{|U|} \\ &= \sum_{i=1}^{m}\left(\sum_{j=1}^{n}\frac{|Y_j\cap X_i|}{|U|}\frac{|X_i| - |Y_j\cap X_i|}{|U|}\right) \end{aligned}$$

(4) Decomposition of CCE Evaluation Function

$$\begin{aligned} CE(D|B) &= \sum_{i=1}^{m}\left(\frac{|X_i|}{|U|}\frac{C^2_{|X_i|}}{C^2_{|U|}} - \sum_{j=1}^{n}\frac{|X_i\cap Y_j|}{|U|}\frac{C^2_{|X_i\cap Y_j|}}{C^2_{|U|}}\right) \\ &= \sum_{i=1}^{m}\left[\frac{|X_i|^2\times(|X_i|-1)}{|U|^2\times(|U|-1)} - \right. \\ &\quad \left.\sum_{j=1}^{n}\frac{|X_i\cap Y_j|^2\times(|X_i\cap Y_j|-1)}{|U|^2\times(|U|-1)}\right] \end{aligned}$$

By above derivation, all four evaluation functions can be writen as the form $EF = \sum_{i=1}^{m} EF_i$. Hence, EF can be calculated in parallel because EF_i can be calculated independently.

B. Calculation of values of evaluation functions based on MapReduce

To introduce our method based MapReduce, we give the following definition first.

Definition 17: Given a decision table $S = (U, C \cup D, V, f)$. Let $S = \bigcup_{k=1}^{p} S_k$, where $S_k = (U_k, C\cup D, V, f)$.

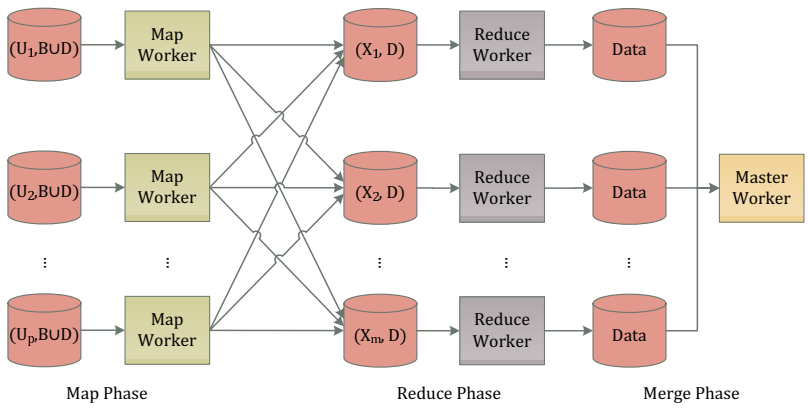


Figure 3. Calculation of the value of Evaluation Function using MapReduce

It satisfies: (1) $U = \bigcup_{k=1}^{p} U_k$; (2) $U_g \cap U_h = \emptyset$, $\forall g, h \in \{1, 2, \cdots, p\}$ and $g \neq h$. It means the decision table S is divided into m sub-decision tables. Then we call S_k is a sub-decision table of S.

Here, we design the parallel method for calculating the values of evaluation functions based on MapReduce. The data flow is shown in Fig. 3, includes three steps: Map, Reduce and Merge. Before that, the input data are divided into p blocks: $(U_1, B \cup D)$, $(U_2, B \cup D)$, $\cdots$, $(U_p, B \cup D)$, which have the similar size, and stored in different computing nodes.

- **Map phase:** Each Map Worker partitions $U_k, k \in \{1, 2, ..., p\}$ by the condition attribute set B (see Algorithm 1).
- **Reduce phase:** Each Reduce Worker collects the data which owns the same key $\overrightarrow{X_{iB}}$, and partitions X_i by the decision D. Then, EF_i is calculated according to Definitions. The detailed process is listed in Algorithm 2.
- **Merge phase:** Calculation of the sum of EF_i (see Algorithm 3). This step will be done in Master worker.

Algorithm 1: Map($key, value$)

Input:
//key: document name
//value: $S_k = (U_k, C \cup D)$
//Global variable: $B \subseteq C$
begin
  for *each* $x \in U_k$ **do** output.collect($\overrightarrow{x}_B$, $\overrightarrow{x}_D$);
  // Calculate $U_k/B = \{X_{k1}, X_{k2}, \cdots, X_{km}\}$
end

IV. Experimental Analysis

Our experiments run on the Apache Hadoop platform [13], which is open source software framework that supports

Algorithm 2: Reduce(*key, value*)

Input:
//key: $\overrightarrow{X_{iB}}$
//value: $S_i' = (X_i, D)$, where $X_i = \bigcup_{k=1}^{p} X_{ki}$
begin
  $X_i/D = \{Y_{i1}, Y_{i1}, \cdots, Y_{in}\}$;
  //It is easy to verify that $Y_{ij} = X_i \cap Y_j$
  Double Result;
  switch *Method* **do**
    case *PR*
      if $|X_j/D| == 1$ **then**
        Result = $\frac{|X_j|}{|U|}$;
      else Result = 0;
      ;
    case *SCE*
      Result = $-\frac{1}{|U|} \sum_{j=1}^{n} |Y_{ij}| \log \frac{|Y_{ij}|}{|X_i|}$;
    case *LCE*
      Result = $\sum_{j=1}^{n} \frac{|Y_{ij}|}{|U|} \frac{|X_i| - |Y_{ij}|}{|U|}$;
    case *CCE*
      Result = $\frac{|X_i|^2(|X_i|-1)}{|U|^2(|U|-1)} - \sum_{j=1}^{n} \frac{|Y_{ij}|^2(|Y_{ij}|-1)}{|U|^2(|U|-1)}$;
    end
  endsw
  output.collect(Result);
end

Algorithm 3: Merge(V)

Input: The list of results from Reduce phase *V*
Output: The value of Evaluation Function
begin
  Double sum = 0;
  for *each* $val \in V$ **do** sum += val;
  //Calculate $EF = \sum_{i=1}^{m} EF_i$
  ;
  Output sum;
end

data-intensive distributed applications. Hadoop version 1.0.1 and Java 1.6.0.12 are used as MapReduce system.

The experiments are conducted on large clusters of compute machines. In detail, the task nodes consist of two kinds of machines. One kind of machines have 16 GB main memory and use AMD Opteron Processor 2376 with 2 Quad-Core CPUs (8 cores in all, each has a clock frequency of 2.3 GHz). The other kind of machines have 8 GB main memory and use Intel Xeon CPU E5410, comprising two Quad-Core CPUs (8 cores in all, each has a clock frequency of 2.33 GHz). The operating system in these machines is Linux CentOS 5.2 Kernel 2.6.18. Here, we use 1,2,4,8,16 and 32 cores in experiments.

A. Data sets

Two large data set are utilized in the experiments. One data set, named KDD99, is downloaded from the machine learning data repository, University of California at Irvine [29], which consists of approximately five million records. Each record consists of 1 decision attribute and 41 condition attributes, where 6 are categorical and 35 are numeric. Since our method can only deal with categorical attributes, we discretize the 35 numeric attributes firstly. Another data set, is synthetic and generated by means of the WEKA data generator [30]. Each record of that consist of 1 decision attribute and 20 condition attributes where 5 attributes are irrelevant and 15 are relevant. The data sets are outlined in Table II.

Table II
A DESCRIPTION OF REAL AND SYNTHETIC DATA SETS

	Data sets	Records	Attributes	Classes	Size
1	KDD99	4,898,421	41	23	0.48 GB
2	R15360k.A20.C10	15,360,000	20	10	1.70 GB

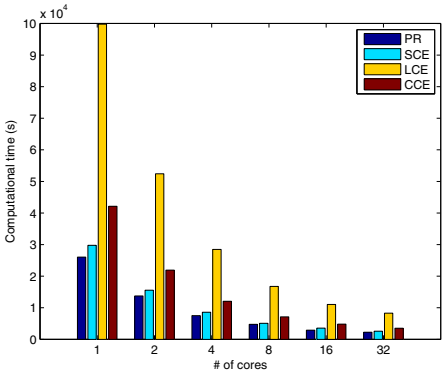
B. Performance

Figure 4 shows the computational time of four parallel methods with different cores on two data sets. As the number of the cores increases, the computational time of the parallel methods becomes smaller.

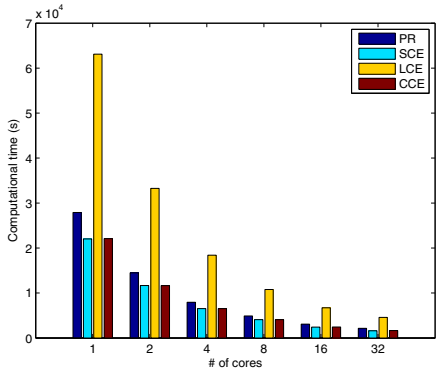
Further, we examine the speedup characteristic of proposed parallel methods.

The ideal parallel algorithm demonstrates linear speedup: a system with p times the number of cores yields a speedup of p. However, linear speedup is difficult to achieve because the communication cost increases with the increasing number of clusters.

We perform the speedup evaluation on data sets with quite different sizes and structures. The number of processes varies from 1 to 32. Figure 5 shows the speedup of four methods over two data sets. As the result shows, the proposed parallel methods have a good speedup performance. KDD99 has a lower speedup curve, because the size of KDD99 is too small. As the size of the data set increases,



(a) KDD99



(b) R15360k.A20.C10

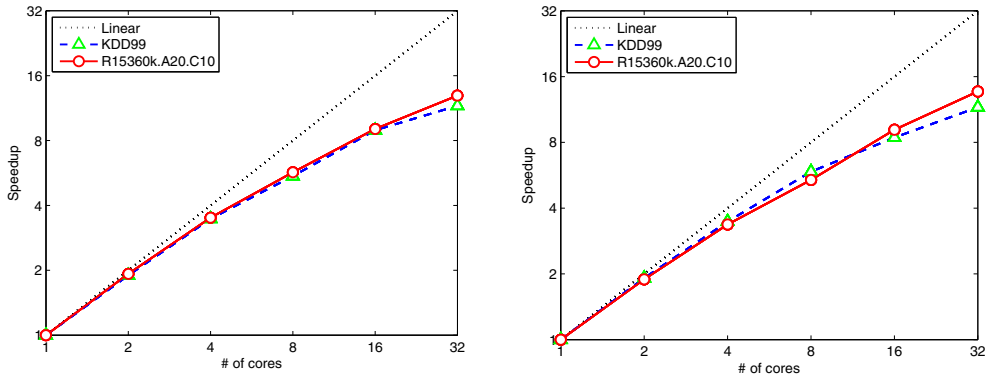
Figure 4. Computational Time

the speedup performs better. Therefore, the proposed parallel methods based on MapReduce can treat very large data efficiently. In fact, since the cluster is heterogeneous with two different kinds of machines, we run experiments on 1 core using the better machine and the proposed parallel methods would have a better speedup in a homogeneous setting.

V. CONCLUSIONS

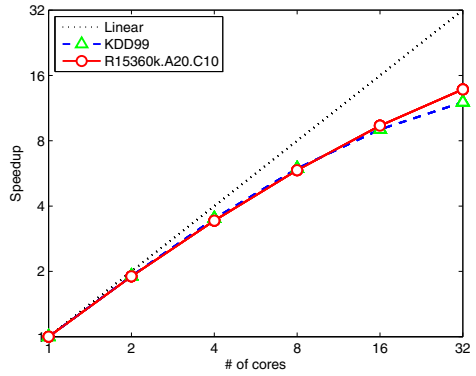
In this paper, we combined MapReduce technique with traditional attribute reduction methods, and designed a novel parallel method based on MapReduce for processing very large data set. The proposed large-scale attribute reduction algorithms can select features effectively and efficiently. It has a good speedup and helps to improve the classification accuracy.

Since the proposed parallel method is combined with classical rough set theory, it can only deal with categorical data. Hence, one of our future work is to present a parallel method combined with extended rough set models to process numerical, missing, set-valued and more composite data. Another work is to develop a parallel library for rough sets, which can help expand the application of rough sets in the field of data mining and deal with big data.

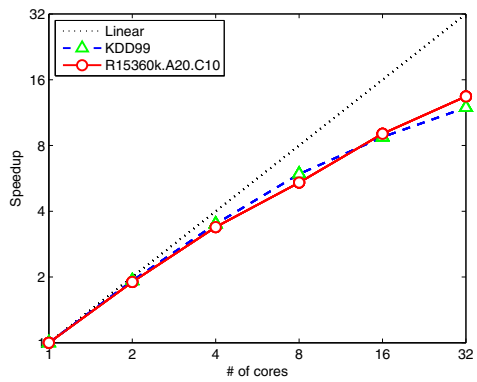


(a) PR

(b) SCE



(c) LCE



(d) CCE

Figure 5. Speedup of four methods

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation of China (Nos. 60873108, 61175047, 61100117) and NSAF (No. U1230117), the Fundamental Research Funds for the Central Universities (No. SWJTU11ZT08), the Research Fund of Traction Power State Key Laboratory, Southwest Jiaotong University (No. 2012TPL_T15), the Science and Technology Planning Project of Sichuan Province (No. 2012RZ0009), China, and the Fostering Foundation for the Excellent Ph.D. Dissertation of Southwest Jiaotong University 2012.

REFERENCES

[1] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial Intelligence*, vol. 151, no. 1-2, pp. 155–176, Dec. 2003.

[2] Q. Hu, W. Pedrycz, D. Yu, and J. Lang, "Selecting discrete and continuous features based on neighborhood decision error minimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 137–150, feb. 2010.

[3] Y. Qian, J. Liang, W. Pedrycz, and C. Dang, "Positive approximation: An accelerator for attribute reduction in rough set theory," *Artificial Intelligence*, vol. 174, no. 9-10, pp. 597–618, Jun. 2010.

[4] Q. Hu, Z. Xie, and D. Yu, "Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation," *Pattern Recognition*, vol. 40, no. 12, pp. 3509–3521, Dec. 2007.

[5] Y. Qian, J. Liang, W. Pedrycz, and C. Dang, "An efficient accelerator for attribute reduction from incomplete data in rough set framework," *Pattern Recognition*, vol. 44, no. 8, pp. 1658–1670, Aug. 2011.

[6] Y. Yang, Z. Chen, Z. Liang, and G. Wang, "Attribute reduction for massive data based on rough set theory and MapReduce," in *Rough Set and Knowledge Technology*, ser. Lecture Notes in Computer Science, J. Yu, S. Greco, P. Lingras, G. Wang, and A. Skowron, Eds. Springer Berlin / Heidelberg, 2010, vol. 6401, pp. 672–678.

[7] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, no. 1, pp. 3–27, 2007.

[8] T. Li, D. Ruan, W. Geert, J. Song, and Y. Xu, "A rough sets based characteristic relation approach for dynamic attribute generalization in data mining," *Knowledge-Based Systems*, vol. 20, no. 5, pp. 485–494, Jun. 2007.

[9] J. Zhang, T. Li, D. Ruan, and D. Liu, "Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems," *International Journal of Approximate Reasoning*, vol. 53, no. 4, pp. 620–635, Jun. 2012.

[10] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Information Sciences*, vol. 178, no. 18, pp. 3577–3594, Sep. 2008.

[11] J. Liang, F. Wang, C. Dang, and Y. Qian, "A group incremental approach to feature selection applying rough set technique," *Knowledge and Data Engineering, IEEE Transactions on*, no. 99, p. accpet, 2012.

[12] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[13] T. White, *Hadoop: The Definitive Guide*, 2nd ed. O'Reilly Media / Yahoo Press, 2010.

[14] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: a runtime for iterative mapreduce," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 810–818.

[15] J. Talbot, R. M. Yoo, and C. Kozyrakis, "Phoenix++: modular mapreduce for shared-memory systems," in *Proceedings of the second international workshop on MapReduce and its applications*, ser. MapReduce '11. New York, NY, USA: ACM, 2011, pp. 9–16.

[16] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang, "Mars: a mapreduce framework on graphics processors," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, ser. PACT '08. New York, NY, USA: ACM, 2008, pp. 260–269.

[17] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Greenwich, CT, USA: Manning Publications Co., 2011.

[18] J. Zhang, T. Li, D. Ruan, Z. Gao, and C. Zhao, "A parallel method for computing rough set approximations," *Information Sciences*, vol. 194, no. 0, pp. 209–223, Jul. 2012.

[19] J. Zhang, T. Li, and Y. Pan, "Parallel rough set based knowledge acquisition using mapreduce from big data," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, ser. BigMine '12. New York, NY, USA: ACM, 2012, pp. 20–27.

[20] J. Zhang, J.-S. Wong, T. Li, and Y. Pan, "A comparison of parallel large-scale knowledge acquisition using rough set theory on different mapreduce runtime systems," *International Journal of Approximate Reasoning*, 2013, doi: http://dx.doi.org/10.1016/j.ijar.2013.08.003.

[21] *Amazon Elastic MapReduce*, http://aws.amazon.com/elasticmapreduce/.

[22] *HadoopOnAzure*, https://www.hadooponazure.com/.

[23] Z. Pawlak and A. Skowron, "Rough sets and boolean reasoning," *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.

[24] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 29–43, Oct. 2003.

[25] J. Y. Liang, K. S. Chin, C. Y. Dang, and R. C. M. Yam, "A new method for measuring uncertainty and fuzziness in rough set theory," *International Journal of General Systems*, vol. 31, no. 4, pp. 331–342, 2002.

[26] J. Y. Liang and Z. B. Xu, "The algorithm on knowledge reduction in incomplete information systems," *International Journal of Uncertainty Fuzziness and Knowledge-based Systems*, vol. 10, no. 1, pp. 95–103, Feb. 2002.

[27] Y. Qian and J. Liang, "Combination entropy and combination granulation in rough set theory," *International Journal of Uncertainty Fuzziness and Knowledge-based Systems*, vol. 16, no. 2, pp. 179–193, Apr. 2008.

[28] D. Slezak, "Approximate entropy reducts," *Fundamenta Informaticae*, vol. 53, no. 3-4, pp. 365–390, Dec. 2002.

[29] D. Newman, S. Hettich, C. Blake, and C. Merz, *UCI Repository of Machine Learning Databases*, University of California, Department of Information and Computer Science, Irvine, CA, 1998. (http://archive.ics.uci.edu/ml/).

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.