Parallel Decision Tree Algorithm Based on Granular Computing

Liu Ping1,*, Wu Zhenggang2, Zhou Hao3, Yang Junping4 and Taorong Qiu5

The School of Information Engineering, Nanchang University, Nanchang 330031, China

Abstract: Large data processing has become a hot topic of current research. How to efficiently dig out useful information from large amounts of data becomes an important research direction in the field of data mining. Firstly, some granular concept about the decision tree was introduced in this article. Secondly, referring to granular computing, the improvement and parallelization ID3 algorithms were presented. Finally, the proposed algorithms were tested on two data sets. It can be concluded that the algorithm's classification accuracy improved. From the test on Hadoop platform, the results demonstrate that the parallel algorithms can efficiently process massive datasets.

Keywords: ID3, hadoop, granular computing, data mining, large data processing.

1. INTRODUCTION

With the advent of big data, data processing has attracted more and more attention of scholars. How to extract the better much important information behind the data has become a topic of growing concern. In the field of data mining, decision tree technology is the key technology. There are some common classification algorithms such as ID3 algorithm, C4.5 algorithm, KNN, SVM, etc.

In order to adapt to the demand of large data processing, in recent years Google company put forward a distributed file system (Google file system, GFS) and Map-Reduce parallel programming model [1-5], which provides the infrastructure for massive data mining, but the traditional data mining algorithms cannot be directly applied to the parallel platform. Therefore, parallelization of traditional data mining algorithm has become a research focus of scholars at home and abroad.

Granular computing (GrC) [6-10] is a new concept and computational model, and may be regarded as a label of family of theories, methodologies, and techniques that make use of granules. In the domain of data mining, GrC provides a conceptual framework for studying many issues.

This thesis is arranged as follows: In section 2 briefly introduces the traditional ID3 algorithm, the concepts about GrC are also introduced in this section. The defects of the traditional ID3 algorithm are discussed and an improved method is put forward. In section 3 discusses the parallelization of the improved ID3 algorithm. In section 4, some of the algorithm accuracy testing and the parallel efficiency testing have been done and the results of test are discussed. Conclusions are given in the last section.

2. THE TRADITIONAL ID3 ALGORITHM

2.1. The Basic Principle

ID3 [11-15] is a decision tree algorithm based on information entropy. A decision tree is a flowchart like tree structure, where each internal node (nonleaf node) denotes a judge on an attribute, each branch represents an attribute value, and each leaf node (terminal node) holds a class label. The core of the algorithm is to use the information entropy as the training sample set splitting measurement standard in the generation of decision tree.

The algorithm proceeds as follows [11-15]: Firstly, it calculates each attribute’s information entropy in turn to select a best attribute as the splitting attribute. Then the data are partitioned into subsets in accordance with the values of the splitting attribute. For each subset recursive implementation of the above process until the each tuple is correctly classified.

2.2. The Shortcomings of ID3 Algorithm

Although ID3 is a typical decision tree classification algorithm, but there are still shortcomings:

1) Because the attribute who has the maximum information entropy is selected as the splitting attribute, therefore, the choice will be more inclined to attribute which has more different attribute values, but not attribute values more is the best attribute.

2) The data set are bigger, the computation of the algorithm also increases more.

The traditional calculation formula for information entropy is expressed as:
The given probability distribution $P = (p_1, p_2, \ldots, p_n)$, then the information entropy of $P$ is defined as

$$I(P) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$ 

Suppose that $S = (U, A, D)$ is a set of data samples. Assume that class label attribute has $m$ different values, expressed as

$$U / D = \{D_1, D_2, \ldots, D_m\}, U / A = \{a_1, a_2, \ldots, a_l\},$$

$S_i$ is the number of samples in $D_i(i = 1, 2, \ldots, m)$, $s_i = |D_i|$, Equation (1) is the expectations of the information on a given sample classification.

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2 (p_i), p_i = s_i / s = |U|$$

$$E(A) = \sum_{i=1}^{m} s_i \log_2 (s_i / s)$$

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2 (p_i)$$

Where $p_i = s_i / s$ is a sample of $s_i$ in the probability of belonging to class $D_i$.

The specific definition of the above three formula can reference literature [11-15].

In the process of constructing a decision tree the most important is the best splitting attribute selection, and the key to the best splitting attribute selection is to calculate the information entropy. From the above three formula can be seen, the calculation of critical information entropy is the probability distribution of condition attribute and the class attribute.

Assuming there are $n$ objects in the sample set and $m$ attributes of each object. In order to derive the probability distribution of attribute values, the time complexity of the system is at least $O(mn)$. If the algorithm is to deal with large data sets, one can imagine the time complexity. Therefore, in consideration of the above, a new method of calculating the information entropy is proposed.

### 2.3. The Improved Information Entropy Based on Granule Computing

Hypothesis that $A = C \cup D$ , $A = C \cup D$ be an information system(IS), Where $G_r = (\varphi, m(\varphi))$ is a set of objects, it is a non-empty finite set of objects, called the universe, $G_r = (\varphi, m(\varphi))$ is a set of attributes. $G_r = (\varphi, m(\varphi))$ is the condition attribute set, $G_r = (\varphi, m(\varphi))$ is the decision attribute set. An information granule [6] is defined as the tuple $G_r = (\varphi, m(\varphi))$, where $\varphi$ is expressed as the intensity of information granule $G_r$, and $m(\varphi)$ is expressed as the extension of information granule $G_r$.

**Definition 1:** (Elementary granule).

Let $G_r = (\varphi, m(\varphi))$ be an information granule [6]. If $\varphi_{y} = (a_i, v_i^j)$, $m(\varphi_{y}) = \{ u | f(u, a_i) = v_i^j, u \in U, a_i \in A \}$, where $m(\varphi_{y})$ refers to the object set whose attribute value of $a_i \in A$ is $v_i^j$. $v_i^j$ is the $j$-th attribute value of attribute $a_i$, then $G_r$ is called an elementary information granule. $G_r = ((a_i, v_i^j), m(a_i, v_i^j))$, if $a_i \in C$, then it is called the conditional granule; if $a_i \in D$, then called the decision granule.

**Definition 2:** (Size of information granule).

For $G_r = (\varphi, m(\varphi))$, size of information granule [6] can be defined as the extension of the extension of the information granule, it is expressed as $|m(\varphi)|$.

**Definition 3 (⊗ operation)**

Suppose that $G_{r1} = (\varphi, m(\varphi))$ and $G_{r2} = (\varphi, m(\varphi))$ are the Elementary granules, The operation of intersect between them is defined as: $G_{r1} \otimes G_{r2} = (\varphi \cup \varphi, m(\varphi) \cap m(\varphi))$ (4)

**Definition 4 (binomial granule)**

Suppose that $G_{r1} = (\varphi, m(\varphi))$ and $G_{r2} = (\varphi, m(\varphi))$ are the Elementary granules, binomial granule is defined as $G_r = G_{r1} \otimes G_{r2} = (\varphi \cup \varphi, m(\varphi) \cap m(\varphi))$

**Definition 5 (A combination of granules)**

If $G_{r_1} = (\varphi_1, m(\varphi_1))$, $G_{r_2} = (\varphi_2, m(\varphi_2))$, $G_{r_m} = (\varphi_m, m(\varphi_m))$ are the Elementary granules, A combination of granules is defined as $G_r = G_{r_1} \otimes G_{r_2} \otimes \ldots \otimes G_{r_m} = (\phi \cup \phi_1 \cup \ldots \cup \phi_m, m(\phi_1) \cap m(\phi_2) \ldots \cap m(\phi_m))$ (5)

**Definition 6:** (The information entropy of attribute: $Gain(U, c_i)$)

Without loss of generality, assume that there is only one decision attribute. Suppose that elementary granule $\varphi_{y} = (c_i, v_i^j)$, $m(\varphi_{y}) = \{ u | f(u, c_i) = v_i^j, u \in U, c_i \in A \}$

$G_r(\varphi_{y}) = ((c_i, v_i^j), m(c_i, v_i^j))$ $G_r(d_i) = ((d_k, v_k^j), m(d_k, v_k^j))$

$\omega_{ik} = G_r(\varphi_{y}) \otimes G_r(d_i)$

$Gain(U, c_i) \equiv \text{Entropy}(U) - \sum_{j \in \{1, 2, \ldots, k\}} \frac{|m(\varphi_{y})|}{|U|} \text{Entropy}(\omega_{ik})$ (6)

$\text{Entropy}(\omega_{ik}) \equiv \sum_{k=1}^{b} -p_k \log_2 p_k, p_k = \frac{|m(\varphi_{ik})|}{|m(\varphi_{y})|}$ (7)
3. THE PARALLEL ID3 ALGORITHM BASED ON GRANULAR COMPUTING

Through the above analysis, the time complexity calculation of each attribute’s information entropy is relatively high in the construction of decision tree, so computing attribute information entropy can be paralleled based on cloud computing platform.

The key of algorithm’s parallelization is the design of map function and reduce function.

3.1. The Granulation of Information System

Algorithm 1: Map(Object key, value) Input: Decision information system \( S = (U, A, V, f), A = C \cup D \), object \( x \in U \), attribute \( a \in A_c \).

Output: Elementary granule space \( G_{rs} = \{G_r\} \).

\( G_r = (\varphi, m(\varphi)) \)

(key: \( \varphi = (a_i, v_i) (1 \leq i \leq m) \); Value : \( m(\varphi) \))

\( // m = |A|, a_i \in A : v_i \) is the \( j-th \) attribute value of attribute \( a_i \)

\( // m(\varphi) = \{u \mid f(u, a_i) = v_i, u \in U, a_i \in A\} \)

for \( u \in DS_j \) do // DS_j : the \( i-th \) data slice

for \( (a_i \in A(1 \leq i \leq |A|)) \) do

if \( f(u, a_i) = v_i \)

\( \{\varphi = (a_i, v_i)\}; \)

\( m(\varphi) = \{u\}; \)

Emit \( <\varphi, m(\varphi)> \)

Algorithm 2: Reduce(Text \( \Phi \), Iterable <Text> values)

//The final result of processing the map function will be sent to the merger of the reduce function. When combined, have the same key is sent to the same reducer.

Input: granule \( \varphi = (a_i, v_i) (1 \leq i \leq m) \), \( m(\varphi) = \{u\}; \)

Output: \( <\text{granule } \varphi = (a_i, v_i) (1 \leq i \leq m) > \), union of \( m(\varphi) > \)

For \( i = 1 \) to values.size()

\( m(\varphi) = m(\varphi) + \text{values}[i]; \)

Emit \( <\varphi, m(\varphi)> \)

Through the parallel processing process, information system has been converted to a particle space, next based on the particle space, each attribute's information entropy can be calculated, the best splitting property can be found.

3.2. The Attribute Information Entropy Calculation Based on Granule

Algorithm 3: The formation of conditions granule space \( G_{rc} \) and decision granule space \( G_{rd} \)

Input: Elementary granule space \( G_{rs} = \{G_r\} \).

\( G_r = (\varphi, m(\varphi)) \)

Output: Condition granule space \( G_{rc} = \{G_r\} \).

Decision granule space \( G_{rd} = \{G_r\} \).

Step1: \( G_{rc} = null \); \( G_{rd} = null \)

Step2: for \( G_r = (\varphi, m(\varphi)) \in G_{rs} \) do // \( \varphi = (a, v) \)

\{if \( (a \in C) \)

\( G_{rc} = G_{rc} + \{G_r\}; \)

if \( (a \in D) \)

\( G_{rd} = G_{rd} + \{G_r\}; \)

Algorithm 4: According to the formula, calculate conditions information entropy.

Input: Condition granule space \( G_{rc} = \{G_r\} \).

Decision granule space \( G_{rd} = \{G_r\} \).

Output: The information entropy of each condition attribute. \( Gain(U, c_i) \)

step1: According to definition 6,

for \( G_r(\varphi_{ij}) \in G_{rc} \)

for \( G_r(d_i) \in G_{rd} \)

compute \( \omega_{ij} = G_r(\varphi_{ij}) \otimes G_r(d_i) \).

step 2: compute the condition information entropy

\( \varphi_{ij} = (c_i, v_i) \)

\( m(\varphi_{ij}) = \{u \mid f(u, c_i) = v_i, u \in U, c_i \in A\} \)

\( G_r(\varphi_{ij}) = ((c_i, v_i), m(c_i, v_i)) \)

\( G_r(d_i) = ((d_i, v_i), m(d_i, v_i)) \)

\( Gain(U, c_i) = \text{Entropy}(U) - \sum_{j=1,2,...,|V|} \frac{m(\varphi_{ij})}{|V|} \text{Entropy}(\omega_{ij}) \)
\[ \text{Entropy}(\omega_{jk}) = \sum_{k=1}^{K} - p_k \log_2 p_k = \left| \frac{m(\omega_{jk})}{m(\phi_j)} \right| \]

4. THE EXPERIMENTAL ANALYSIS

(1) The experimental environment:

Table 1. UCI Data set.

<table>
<thead>
<tr>
<th>Data set Name</th>
<th>Tuples Number</th>
<th>Attribute Number</th>
<th>Number Class</th>
<th>The Type of Attribute</th>
<th>Missing Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom</td>
<td>8124</td>
<td>22</td>
<td>2</td>
<td>Discrete</td>
<td>include</td>
</tr>
<tr>
<td>Nursery</td>
<td>12960</td>
<td>8</td>
<td>5</td>
<td>Discrete</td>
<td>Not include</td>
</tr>
<tr>
<td>Promoters</td>
<td>106</td>
<td>58</td>
<td>2</td>
<td>Discrete</td>
<td>Not include</td>
</tr>
</tbody>
</table>

(2) Data source:

The experimental data from the UCI(http://archive.ics.uci.edu/ml/datasets.html) data set

(3) The experimental analysis:

a) Test and analysis of algorithm accuracy

In the testing classification accuracy, randomly selected 90% of the data from the original data as training set to build a decision tree, select 10% of the data as a test set, repeat the test 10 times for each data set, taking the average value as the test accuracy rate.

The analysis of Table 1 and Table 2 of the experimental data, the conclusion can be drawn: the algorithm's classification accuracy is higher than that of the traditional algorithm, which also shows the effectiveness of the algorithm.

b) The running time and speedup ratio:

Some standard data test data are from UCI machine learning database set, in this experiment, each data set is amplified to 100M, 300M, 500M, 1000M using the duplication means and respectively running on clusters whose slaves number is 1, 2, 3 machines, the running results shown in the table below.

Table 2. The results of accuracy.

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>The Algorithm in this Paper</th>
<th>The Traditional Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushroom</td>
<td>99.80%</td>
<td>100%</td>
</tr>
<tr>
<td>Promoters</td>
<td>79%</td>
<td>76.4151%</td>
</tr>
<tr>
<td>Nursery</td>
<td>94.1%</td>
<td>78.9%</td>
</tr>
</tbody>
</table>

![Fig. (1). Running Time of Nursery Data Set.](image1)

![Fig. (2). Running Time of Mushroom Data Set.](image2)
CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This research is supported by the National Natural Science Foundation of China under grant No. 81260578 and No. 81460769 and No.61163023 and No.61070139.

REFERENCES


Received: September 16, 2014  Revised: December 23, 2014  Accepted: December 31, 2014

© Ping et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the (https://creativecommons.org/licenses/by/4.0/legalcode), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.