

# 海量数据下不完备信息系统的知识约简算法

王 添,姜 麟,米允龙

(昆明理工大学 理学院,云南 昆明 650500)

**摘 要:**面向大规模的数据进行知识约简是近年来粗糙集理论研究的热点。传统不完备信息系统的知识约简是假设在初始时将所有需要处理的数据一次性地装入内存中,这明显不适合处理海量数据,更不适合处理含有缺失信息的海量数据。为此,深入剖析了带有缺失信息的数据特征,把缺失属性的值用该属性所有可能的取值表示,并结合知识约简算法中的可并行性,从属性(集)的可辨识性和不可辨识性出发,并在 MapReduce 框架下设计了可用来处理不完备信息系统的知识约简算法。实验结果表明,该算法是有效可行的,能够对不完备信息系统中的海量数据进行知识约简。

**关键词:**海量数据;云计算;粗糙集;不完备信息系统;约简;MapReduce

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2015)01-0137-06

doi:10.3969/j.issn.1673-629X.2015.01.031

## Knowledge Reduction Algorithms of Incomplete Information System in Massive Datasets

WANG Tian,JIANG Lin,MI Yun-long

(Faculty of Science,Kunming University of Science and Technology,Kunming 650500,China)

**Abstract:** Knowledge reduction for massive datasets has attracted many research interests in rough set theory. Traditional knowledge reduction algorithms of incomplete information system assume that all the datasets can be loaded into the main memory, which are obviously infeasible for large-scale datasets, especially for massive datasets with missing information. To this end, deeply analyze the characteristics of massive datasets with missing information, and allow the missing attribute value to take all possible values. Then, by combining the parallel computations used in classical knowledge reduction algorithms with the discernibility (indiscernibility) of the attributes, a knowledge reduction algorithm is designed for incomplete information systems under MapReduce framework. The experimental results demonstrate that this algorithm is effective and feasible, which can efficiently process massive datasets for knowledge reduction in incomplete information systems.

**Key words:** massive data; cloud computing; rough set; incomplete information system; reduction; MapReduce

## 0 引 言

随着信息科学的进步与发展,人们迎来了大数据时代。而大数据不仅仅是海量数据,人们更要挖掘出数据背后隐藏着的十分重要的信息。关于海量数据的处理,以往的集中式数据挖掘算法<sup>[1]</sup>已不能满足要求。Google 公司提出了分布式文件系统(Google File System, GFS)<sup>[2]</sup>和并行编程模式 MapReduce<sup>[3]</sup>,为处理海量数据挖掘提供了一个很好的平台。

1982 年波兰数学家 Pawlak Zdzislaw 提出了可用于研究不确定和不精确信息的数学方法<sup>[4]</sup>,即经典的粗集理论,现已成功应用于数据挖掘、人工智能、通信

信息处理等领域。而不完备信息系统<sup>[5-10]</sup>的数据挖掘已成为近年来粗糙集理论的研究热点,对于不完备信息系统的处理方法可以利用间接处理的方法对不完备信息系统进行扩展,并利用其扩展后的信息表进行属性约简。

然而在现实生活中海量数据的信息缺失是不可避免的,如何对带有缺失信息的海量数据进行知识的约简还没被涉及。对于带有缺失信息的不完备信息系统可以利用集值信息系统来处理,故文中结合了 MapReduce 编程的特点,分析了属性(集)的特点即可辨识性和不可辨识性并结合集值理论,设计了适合面向不完

收稿日期:2014-02-07

修回日期:2014-05-12

网络出版时间:2014-11-17

基金项目:云南省教育科研基金(2010Y389)

作者简介:王 添(1990-),男,硕士研究生,研究方向为粗糙集理论、数据挖掘;姜 麟,博士,教授,硕士生导师,研究方向为并行计算、智能算法。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20141117.2205.019.html>

备信息系统中大规模数据的知识约简算法,并对该算法进行了实现。最后,通过实验表明海量数据下不完备信息系统的知识约简算法是有效可行的。

## 1 相关理论

### 1.1 粗糙集相关概念

此部分,将介绍文中主要用到的一些粗糙集的基本理论,详细内容请参考文献[4,11]。

定义1:五元组  $S = \langle U, C, D, V, f \rangle$  是一个决策表,其中  $U = \{x_1, x_2, \dots, x_i\}$  是研究对象的非空有限集, $U$ 称为论域; $C$ 称为条件属性的非空有限集合, $D$ 称为决策属性的非空有限集合,  $C \cap D = \emptyset$ ;  $V = \bigcup_{a \in C \cup D} V_a$ ,  $V_a$  是属性  $a$  的值域;  $f: U \times (C \cup D) \rightarrow V$  是一个信息函数,它为每个对象赋予一个信息值,即  $\forall a \in C \cup D, x \in U$ , 有  $f(x, a) \in V_a$ ; 每一个属性子集  $R \subseteq C \cup D$  确定一个二元不可区分关系  $\text{IND}(R)$ :

$$\text{IND}(R) = \{(x, y) \in U \times U \mid \forall a \in R, f(x, a) = f(y, a)\} \quad (1)$$

关系  $\text{IND}(R)$  构成了  $U$  的一个划分,用  $U/\text{IND}(R)$  表示,简记为  $U/R$ 。 $U/R$  中的任何元素:  $[x]R = \{y \mid \forall a \in R, f(x, a) = f(y, a)\}$  称为等价类。

定义2:在决策表  $S = \langle U, C, D, V, f \rangle$  中,对于每个子集  $X \subseteq U$  和不可区分关系  $R \subseteq C \cup D$ ,  $X$  的下近似集与上近似集分别由  $R$  导出并给出如下定义:

$$R(X)_- = \bigcup \{Y \in U/R \mid Y \subseteq X\}$$

$$R(X)^+ = \bigcup \{Y \in U/R \mid Y \cap X \neq \emptyset\}$$

定义3:在决策表  $S = \langle U, C, D, V, f \rangle$  中,使  $\forall A \subseteq C, X \subseteq U$ , 用  $AX_-$  表示  $X$  的下近似集,决策属性  $D$  的  $A$  正域  $\text{POS}_A(D) = \bigcup_{X \in U/D} AX_-$ 。

定义4:在决策表  $S = \langle U, C, D, V, f \rangle$  中,  $a \in C$ , 若  $\text{POS}_{C-a}(D) \neq \text{POS}_C(D)$ , 则属性  $a$  在  $C$  中是不可舍去的;  $C$  中一切不可缺少的属性集合称为  $C$  的核(简称核),记为  $\text{Core}(C)$ 。

定义5:在决策表  $S = \langle U, C, D, V, f \rangle$  中,记  $U/C = \{[X'_1]_C, [X'_2]_C, \dots, [X'_s]_C\}$ ,  $U' = \{x'_1, x'_2, \dots, x'_s\}$ ,  $U'_{\text{POS}} = \{x'_{i1}, x'_{i2}, \dots, x'_{ii}\}$ 。其中,  $U'_{\text{POS}}$  中对象为相容对象,  $U'_{\text{BND}} = U' - U'_{\text{POS}}$ , 则

$$S' = (U' = U'_{\text{POS}} \cup U'_{\text{BND}}, C, D, V, f) \quad (2)$$

为简化决策表。

不失一般性,假设决策表  $S$  仅有一个决策属性  $D$ , 并把它的决策属性值分别记为  $1, 2, \dots, k$ , 由  $D$  导出  $U$  上的分类记为  $U/D = \{D_1, D_2, \dots, D_k\}$ , 其中  $D_i = \{x \in U \mid f(x, D) = i\}$  ( $i = 1, 2, \dots, k$ )。

在简化决策表  $S'$  中,将  $U'_{\text{BND}}$  中所有矛盾对象记为  $D_{k+1}$ , 其决策值标记为“?”, 映射为  $k+1$ 。则新划分

$\{D_1, D_2, \dots, D_{k+1}\}$ , 这样,不一致决策表就可以看成“相容”决策表了。

### 1.2 不完备信息系统与集值理论

定义6:假设一个信息系统可以表示为  $S = \langle U, R, V, f \rangle$ 。其中,  $U$  为非空对象有限集合;  $C$  为条件属性集,  $D$  为决策属性集,  $R = C \cup D$  是属性集合;  $V = \bigcup_{r \in R} V_r$  是属性值集合,  $V_r$  表示属性  $R$  的值域;  $f: U \times R \rightarrow V$  是一个映射函数。若  $D$  为空, 则称信息系统为数据表, 不然称之为决策表。把属性子集  $B \subseteq C$  的缺失属性值记为“\*”, 把这种含有缺失属性值的信息系统称为不完备信息系统。

对于任何含有缺失信息的系统从获取数据时的情况来看,人们很难得到实际意义上的准确值而往往都是近似值。既然是近似值,就需要从可用性的角度出发,把同一对象用“多值”的形式来取代“单值”。由于客观条件的约束和随机因素的干扰,同一对象取得的多个值,通常来说只是“相似”而不一定总是相同的。为了保险起见,可以将所有的这些“相似”的信息值都作为该对象的“属性值”。把这种对象在某个确定属性上取得“多值”的情形称为“集值”。

举例说明,为了方便讨论,同时也不失一般性,考虑缺失值的一种情况,将某个子属性的缺失值记作“\*”并将其看作是属性值域内所有可能的取值,即将“\*”看作是某一个属性所有取值的集合。

定义7<sup>[12]</sup>:称  $(U, C, D, F)$  是集值信息系统,其中  $U$  是有限非空的对象集,  $C$  是条件属性集,  $D$  是决策属性集,  $m$  表示非空有限的属性集合。令  $A = C \cup D$ ,  $F = \{f_l \mid l \leq m\}$  是  $U$  与  $A$  的关系集,  $f_l: U \rightarrow P_0(V_l)$  ( $l \leq m$ ),  $V_l$  是属性  $a_l$  的值域,  $P_0(V_l)$  是非空子集全体。

定义8<sup>[12-13]</sup>:设  $(U, A, V, f)$  是一个不完备的信息系统,由定义6得,若  $\forall a \in A, x \in U$ , 有  $f(x, a) = *$ , 令  $f(x, a) = V_a$ , 则称这是一个不完备信息系统向集值信息系统的一种转换,称转换后的信息系统为集值信息系统,若  $A = C \cup D$  且  $C \cap D = \emptyset$ , 则称集值信息系统为集值决策表。

定义9:在简化集值决策表  $\bar{S}'$  中,将  $\bar{U}'_{\text{BND}}$  中一切矛盾对象记为  $D_{k+1}$ , 并把它的决策属性值标识为“?”, 扩展成第  $k+1$  种情况。若  $D_{k+1} = \emptyset$ , 则称集值决策表  $\bar{S}'$  是相容集值决策表; 否则是不相容集值决策表。

例1:表1是一个不完备信息系统,其属性域为  $V_a = \{1, 2\}$ ,  $V_b = \{1, 2\}$ ,  $V_c = \{1, 2\}$ ,  $V_d = \{1, 2\}$ 。其中  $a, b, c$  为条件属性,  $d$  为决策属性。

表1所示的不完备信息系统可以转化为表2所示的集值信息系统。

表2所示的集值信息系统又可以转化成表3所示

的集值决策表。

表 1 不完备信息系统

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
$x_1$	2	1	1	1
$x_2$	1	*	1	1
$x_3$	*	*	2	1
$x_4$	2	1	1	2
$x_5$	*	2	1	2
$x_6$	1	2	1	*

表 2 集值信息系统

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
$x_1$	2	1	1	1
$x_2$	1	{2,1}	1	1
$x_3$	{2,1}	{2,1}	2	1
$x_4$	2	1	1	2
$x_5$	{2,1}	2	1	2
$x_6$	1	2	1	{2,1}

表 3 集值决策表

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
$x_1$	2	1	1	1
$x_2[1]$	1	2	1	1
$x_2[2]$	1	1	1	1
$x_3[1]$	2	2	2	1
$x_3[2]$	2	1	2	1
$x_3[3]$	1	2	2	1
$x_3[4]$	1	1	2	1
$x_4$	2	1	1	2
$x_5[1]$	2	2	1	2
$x_5[2]$	1	2	1	2
$x_6[1]$	1	2	1	2
$x_6[2]$	1	2	1	1

表 2 和表 3 就是对不完备信息系统(表 1)利用定义 7 和定义 8 的集值原理进行集值信息系统的转化,表 3 就是把表 2 的集值信息系统转化为集值决策表。由于实际应用中很少会出现大规模的同一属性下不同的取值,所以文中只考虑所有的属性取值只有两种的情况。

1.3 MapReduce 云计算技术

MapReduce<sup>[14]</sup>是由 Google 首先提出来的并行处理海量数据的编程方法。MapReduce 库先把输入的文件分解成若干个可并行执行的片段,再由主控程序 Master 分配任务给 Map 和 Reduce。执行过程可简单描述如下:

(1)将要执行的 MapReduce 程序复制到 Master 与

每一台 Job 机器中。

(2)Master 决定 Map 程序与 Reduce 程序,分别由哪些 Job 机器执行。

(3)将所有数据分块,分配给执行 Map 程序的 Job 机器运行。

(4)将 Map 后的结果存入执行 Map 程序的 Job 机器中。

(5)执行 Reduce 程序的 Job 机器远程读取每一份 Map 的结果,进行汇总、排序,同时执行 Reduce 程序。

(6)将使用者需要的运算结果输出。

用户只需要将实际问题分解成若干可并行处置的子问题,并构思对应的 Map 和 Reduce 函数,而不必关注 Map、Reduce 到底是如何进行数据分割、容错处理等细节。其形式如下:

Map: (*k*, *v*) →

$\{(k_i, v_i) \mid (i = 1, 2, \cdots, m)\},$

Reduce: (*k*, [*v*<sub>1</sub>, *v*<sub>2</sub>, ⋯, *v*<sub>*k*</sub>]) →

(final\_*k*, final\_*v*)

Map 函数是接收一组输入键值对 key-value,经 Map 计算后通过 Combine 类进行本地聚类,产生中间结果键值对(*k<sub>i</sub>*, *v<sub>i</sub>*)(*i* = 1, 2, ⋯, *m*); Reduce 函数将所有中间结果中含有相同 key 键的数据整合在一起,形成最终的 (final\_*k*, final\_*v*)。

2 云环境下不完备信息系统的知识约简

条件属性值与决策属性值均不相同的两个对象可生成一个可辨识对象对。即如果两个对象的条件属性 *a* 的值不同且决策属性的值也不同,就说 *a* 能够识别这两个对象即一对可辨识对象对。如何衡量 *a* 相对辨识能力的大小。这时候,不妨通过 *a* 可以辨识对象对的个数越多,说明 *a* 相对辨识能力越大来衡量。对于不完备信息系统可以通过第一部分的相关概念和例 1 的过程来进行转化使之成为集值决策表。

假定集值决策表 *S* 一共有 *k* 个不同决策属性值,把当中相容对象的决策属性值记为 1, 2, ⋯, *k*, 将所有不相容对象的决策属性值映射成 *k*+1。这样,整个集值决策表 *S* 就可以看作由 *k*+1 个子决策表 *D*<sub>1</sub>, *D*<sub>2</sub>, ⋯, *D*<sub>*k*+1</sub> 组成的一个“相容”集值决策表,其中每个子决策表对应同一类对象,其对象个数分别为 *n*<sup>1</sup>, *n*<sup>2</sup>, ⋯, *n*<sup>*k*+1</sup>。假设属性 *a* 分别有 1, 2, ⋯, *r* 个不同的属性值。可以把 *D<sub>i</sub>* 中条件属性 *a* 的值为 *p* 的对象个数记为 *n<sub>p</sub><sup>i</sup>*。不难得出 *n<sub>1</sub><sup>i</sup>* + *n<sub>2</sub><sup>i</sup>* + ⋯ + *n<sub>r</sub><sup>i</sup>* = *n<sup>i</sup>* (*j* = 1, 2, ⋯, *k* + 1), *n<sub>1</sub><sup>1</sup>* + ⋯ + *n<sub>r</sub><sup>1</sup>* + ⋯ + *n<sub>1</sub><sup>*k*+1</sup>* + ⋯ + *n<sub>r</sub><sup>*k*+1</sup>* = *n*。

定义 10:在相容集值决策表 *S* 中,当 *A* ∈ *C*, *r* ∈ *A* 时,若 POS<sub>*A*</sub>(*D*) = POS<sub>(*A* − {*r*})</sub>(*D*), 则称 *r* 在 *B* 中相对于属性 *D* 是可省略的,否则称 *r* 是 *B* 中相对于属性 *D* 是

不可省略的。若对  $C$  中的独立子集  $A \in C$ , 有  $\text{POS}_A(D) = \text{POS}_C(D)$ , 则称  $B$  为  $C$  的约简。

定义 11: 在相容集值决策表  $S$  中,  $a \in C$ , 属性  $a$  能够识别的对象对为  $\text{DOP}_a = \{ \langle x, y \rangle \mid f(x, a) \neq f(y, a), f(x, D) \neq f(y, D) \}$ , 其中  $1 \leq i < j \leq k+1$ 。

定义 12: 在相容集值策表  $S$  中,  $A \subseteq C$ , 属性集  $A$  能够辨识的对象对为  $\text{DOP}_A = \{ \langle x, y \rangle \mid \exists a \in A, f(x, a) \neq f(y, a), f(x, D) \neq f(y, D) \}$ , 其中  $1 \leq i < j \leq k+1$ 。

定理 1: 在相容集值策表  $S$  中, 若  $A \subseteq C, \forall a \in A$ , 则有  $\text{DOP}_A = \bigcup_{a \in A} \text{DOP}_a$ 。

证明: 可由定义 9 和定义 10 证得。

定义 13: 在相容集值决策表  $S$  中,  $A \subseteq C$ , 属性集  $A$  能够确定的对象对的个数为

$$\text{DIS}_A^D = \sum_{1 \leq i < j \leq k+1} \sum_{1 \leq p < q \leq r} n_p^i n_q^j \quad (3)$$

定义 14: 在相容集值决策表  $S$  中,  $A \subseteq C$ , 属性集  $A$  拥有的识别能力的大小定义为

$$\text{DIS}_{U,A} = \sum_{1 \leq p < q \leq r} n_p n_q \quad \text{DIS}_{U,A} = \sum_{1 \leq p < q \leq r} n_p n_q \quad (4)$$

定义 15<sup>[15]</sup>: 在相容集值决策表  $S$  中,  $A \subseteq C, c \in C \cup D$ , 则由属性  $c$  新增加的辨识能力定义为属性  $c$  分别在  $A_1, A_2, \dots, A_r$  中新增加的辨识能力大小之和, 即

$$\sum_{i=1}^r \text{DIS}_{A_i, \{c\}} = \text{DIS}_{U,A \cup \{c\}} - \text{DIS}_{U,A} = \text{DIS}_{A_1, \{c\}} + \text{DIS}_{A_2, \{c\}} + \dots + \text{DIS}_{A_r, \{c\}} \quad (5)$$

定理 2: 在相容集值决策表  $S$  中, 若  $A \subseteq C$ , 则

$$\text{DIS}_A^D = \text{DIS}_{U,A} + \text{DIS}_{U,D} - \text{DIS}_{U,A \cup D} \quad (6)$$

证明: 假设  $D_i$  中条件属性集  $A$  的属性值映射为  $p$  的对象个数为  $n_p^i$

$$n_1^j + n_2^j + \dots + n_r^j = n^j (j = 1, 2, \dots, k+1) \Rightarrow$$

$$\text{DIS}_{U,A} = \sum_{1 \leq p < q \leq r} n_p n_q, \text{DIS}_{U,D} = \sum_{1 \leq i < j \leq k+1} n^i n^j。$$

由属性集  $A \cup D$  导出的等价类细分为  $\{A_1^1, A_1^2, \dots, A_1^{k+1}, A_2^1, A_2^2, \dots, A_2^{k+1}, \dots, A_r^1, A_r^2, \dots, A_r^{k+1}\}$ , 其对象个数记为  $n_l = (l = 1, 2, \dots, (k+1)r)$ , 则有  $\text{DIS}_{U,A \cup D} =$

$\sum_{1 \leq l_1 < l_2 \leq (k+1)r} n_{l_1} n_{l_2}$ 。由定义 13 可知,  $\text{DIS}_{U,A \cup D} = \text{DIS}_{U,A} + \sum_{1 \leq p \leq r} \sum_{1 \leq i < j \leq k+1} n_p^i n_p^j$ , 所以

$$\begin{aligned} \text{DIS}_{U,A} + \text{DIS}_{U,D} - \text{DIS}_{U,A \cup D} &= \text{DIS}_{U,A} + \text{DIS}_{U,D} - \\ &(\text{DIS}_{U,A} + \sum_{1 \leq p \leq r} \sum_{1 \leq i < j \leq k+1} n_p^i n_p^j) = \sum_{1 \leq i < j \leq k+1} (n_1^i + n_2^i + \\ &\dots + n_r^i) (n_1^j + n_2^j + \dots + n_r^j) - \sum_{1 \leq i < j \leq k+1} \sum_{1 \leq p \leq r} n_p^i n_p^j = \\ &\sum_{1 \leq i < j \leq k+1} \sum_{1 \leq p < q \leq r} n_p^i n_q^j \end{aligned} \quad (7)$$

因为上述理论中的主要计算对象  $n$  为等价类中的对象个数, 像这样 (等价类, 对象个数) 的形式与 Map 函数需要输入 (key, value) 键值对相似, 故可用 MapRe-

duce 并行计算等价类来计算可识别对象对的个数。

不完备信息系统的知识约简算法如下所述。

算法一: 不完备信息系统转化为集值信息系统, 并求相应的二元划分。

输入: 一个不完备的信息系统  $S$ ;

输出: 集值信息系统  $\bar{S}$ 。

(1) Class Map(key, value)

(2) for each  $x \in U$  do

(3) if  $f(x, a) = "*" , a \in R$ , then

$f(x, a) = V_{a_i}, x = x[i]$ ;

(4) if  $f(x, b) = "x" , b \in R$ , then

fillMap(); //调用下面的 Method fillMap 函数

(5) Let key =  $x_R$ ; //  $x_R$  为属性的取值

(6) Let value =  $x[i]$ ; //  $x[i]$  为对象

(7) Eimt(key, value);

(8) Method fillMap() //此处定义了递归调用函数来进行连续的数据处理

(9) if  $f(x, b) = "*" , b \in R$ , then

$f(x, b) = V_{b_i}, x = x[i]$ ;

(10) if  $f(x, c) = "*" , c \in R$ , then

fillMap(); //递归调用

(11) Let key =  $x_R$ ;

(12) Let value =  $x[i]$ ;

(13) Eimt(key, value);

算法二: Map(Object key, Test value)。

输入: 初始条件属性集  $I$  (初始时空), 条件属性  $a \in C - I$ , 决策属性  $D$ , 对象 value;

输出: <等价类, 出现次数>。

(1) Ia\_EqClass = "a"; D\_EqClass = "D"; Da\_EqClass = "E"

//此处分别为属性集  $I \cup a, D$  和  $I \cup a \cup D$  导出的等价类

(2) for each attribute  $c \in I \cup a$

Ia\_EqClass = Ia\_EqClass + f(value, c);

(3) Emit(Ia\_EqClass, 1);

(4) D\_EqClass = f(value, D);

(5) Emit(D\_EqClass, 1);

(6) for each attribute  $c \in I \cup a \cup D$

Da\_EqClass = Da\_EqClass + f(value, c);

(7) Emit(Da\_EqClass, 1);

算法三: Reduce(String EqClass, Int value)。

输入: 可划分类 EqClass, values[] //此处的可划分类由算法二得到;

输出: <EqClass, 出现次数>。

(1) Record = 0;

(2) for i from 1 to value.size(); Record = Record +

value[  $i$  ];

(3) Eimt<EqClass,Record>;

算法四:主程序。

输入:一个不完备的信息系统  $S$ ;

输出:一个约简 Red。

(1) 启动一个 Job 调用算法一的 Map 函数,使不完备的信息系统  $S$  完备化为一个相容的决策表  $S'$ ;

(2) Red =  $\varnothing$ ;

(3) While(  $DIS_{Red}^D$  不等于  $DIS_C^D$  )

for each attribute  $c \in C - Red$

启动一个 Job,调用算法二的 Map 和算法三的 Reduce 函数,计算  $DIS_{Red \cup \{c\}}^D$ ;

(4)  $c_l = \max_{c \in C - Red} \{DIS_{Red \cup \{c\}}^D\}$  (若其中  $c_l$  不唯一,任取其一)

Red = Red -  $\{c_l\}$ ;

(5) for each attribute  $c \in C - Red$  启动一个 Job,

调用算法二的 Map 函数和算法三的 Reduce 函数;

if  $DIS_{Red - \{c\}}^D = DIS_{Red}^D$  Red = Red -  $\{c\}$ ;

(6) 输出 Red。

3 实验测试与分析

实验的测试与分析,主要是从实验的加速比 (Speedup)、可扩展性 (Scaleup) 以及运行时间 3 个方面对所提出的 MapReduce 框架下不完备信息系统知识约简算法的性能进行评价。选用 UCI 机器学习数据库 (<http://archive.ics.uci.edu/ml/datasets.html>) 中的带有缺失数据的两个数据集 Mammographic Mass Data Set 和 Breast Cancer Wisconsin Data Set (分别记为:  $MMDS_1, BCDS_1, MMDS_2, BCDS_2$ )。为了更好的测试,将  $MMDS_1$  和  $MMDS_2$  分别复制 600 次和 6 000 次,  $BCDS_1$  和  $BCDS_2$  分别复制 6 000 次和 60 000 次来进行测试,数据集如表 4 所示。选用 9 台普通计算机 (1 个主节点和 8 个从节点,配置:CPU 为 2.8 GHz,内存为 2 G)。Hadoop (<http://Hadoop.apache.org/>) 是 Apache 组织中一个十分成功地专注于 DFS 和 MapReduce 的开源项目。本实验采用 Hadoop 0.20.2 和 Java 1.7.0\_17,并在 Centos\_6.3 Linux 系统下构建 MapReduce 环境。

表 4 数据集描述

数据集	记录数	属性数
MMDS <sub>1</sub>	576 600	6
BCDS <sub>1</sub>	4 194 000	10
MMDS <sub>2</sub>	5 766 000	6
BCDS <sub>2</sub>	41 940 000	10

3.1 运行时间

让数据集  $BCDS_1, MMDS_1, BCDS_2, MMDS_2$  分别在 1 ~ 8 个节点上运行,运行结果如表 5 所示。

表 5 算法运行时间 s

数据集	节点数				
	1	3	5	7	8
MMDS <sub>1</sub>	205	175	136	100	80
BCDS <sub>1</sub>	1 139	568	405	278	235
MMDS <sub>2</sub>	1 038	513	365	245	204
BCDS <sub>2</sub>	11 400	4 573	3 217	1 970	1 633

从表 5 可以看出,同一数据集随着节点数增加运行时间不断减少。并且,随着数据量越大,递减的效果越好。这表明对于大数据,用 MapReduce 并行编程能取到很好效果。同时,可以看出,虽然  $BCDS_1$  记录数比  $MMDS_2$  少得多,但是属性比  $MMDS_2$  多,造成运行的时间比  $MMDS_2$  要多。这也可看出属性在约简算法中是十分重要的。

3.2 加速比 (Speedup)

为了测试加速比,固定数据集规模,不断增加数据节点。通常,线性加速比是十分理想的,但是,随着节点增加,集群之间的通信时间会不断增加,因此,一般很难达到理想的线性加速比。公式如下<sup>[16]</sup>:

Speedup(  $m$  ) =  $T_1/T_m$  (8)

其中,  $T_1$  是固定规模数据集在一个节点上的运行时间;  $T_m$  是固定规模数据集在  $m$  个节点上的运行时间。

图 1 显示了不同数据集随着节点增加的加速性能比。从图中可见,数据量越大加速比越好,如  $BCDS_2$  比  $MMDS_1$  的加速比好得多。因此,对于海量数据集,用 MapReduce 进行并行计算能取得很好的效果。

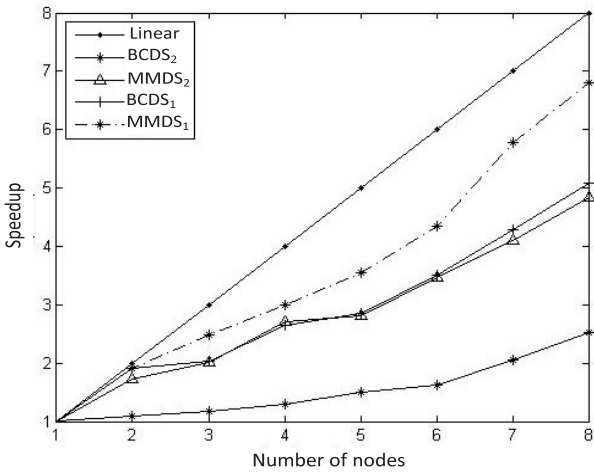


图 1 加速比

3.3 可扩展性

可扩展性指的是与计算机节点数成比例地增大数据集规模的性能,如对于数据集  $BCDS_1$  来讲,用

699 000 记录数在一个节点上运行, 则用  $699\ 000 * 2$  记录数在两个节点上运行。公式如下<sup>[14]</sup>:

$$\text{Scaleup}(\text{DB}, m) = T_{\text{DB}_1} / T_{\text{DB}_m} \quad (9)$$

其中,  $T_{\text{DB}_1}$  是 DB 数据集在一个节点上的运行时间;  $T_{\text{DB}_m}$  是  $m * \text{DB}$  数据集在  $m$  个节点上运行的时间。

图 2 显示了各数据集的可扩展性。显然, 各数据集都有很好的可扩展性。同时, 数据集规模越大, 可扩展性越好, 越稳定。

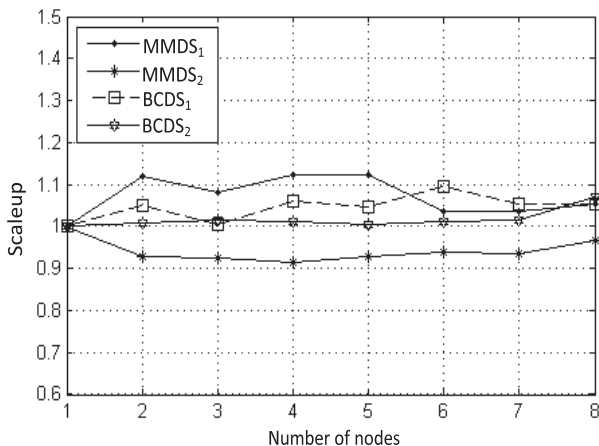


图 2 可扩展性

## 4 结束语

传统不完备信息系统的属性约简算法是采用启发式算法或分辨矩阵式算法来提高效率, 但只适合处理中小型数据集。为了面向含有缺失信息的海量数据进行知识约简, 分析了如何有效地对不完备信息系统进行处理使其完备化并利用属性(集)可辨识性特征来研究知识约简方法中的可并行性, 设计适合处理带有缺失信息的海量数据的知识约简算法。使用普通计算机集群进行实验, 实验结果表明该算法是正确有效的, 能够处理存在数据缺失的大规模数据集。

### 参考文献:

[1] Han Liangxiu, Liew C S, van Hemert J V, et al. A generic parallel processing model for facilitating data mining and integration[J]. Parallel Computing, 2011, 37(3): 157-171.

[2] Ghemawat S, Gobioff H, Leung S T. The Google files system [J]. ACM SIGOPS Operating Systems Review, 2003, 37(5): 29-43.

[3] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters [J]. Communication of the ACM, 2008, 51(1): 107-113.

[4] 王国胤. Rough 集理论与知识获取 [M]. 西安: 西安交通大学出版社, 2001.

[5] Liang Jiye, Xu Zongben. The algorithm on knowledge in incomplete information systems [J]. International Journal of Uncertainty Fuzziness and Knowledge-based Systems, 2002, 10(1): 95-103.

[6] Kryszkiewicz M. Rules in incomplete information systems [J]. Information Sciences, 1999, 113: 271-292.

[7] 周玉新, 周军, 梅红岩, 等. 一种不完备信息系统的约简方法 [J]. 计算机技术与发展, 2007, 17(9): 109-112.

[8] Zhang Wenxiu, Mi Jusheng. Incomplete information system and its optimal selections [J]. Computers and Mathematics with Applications, 2004, 48(5-6): 691-698.

[9] 李海涛, 章德斌. 基于决策树的不完备信息系统规则提取方法 [J]. 计算机工程与科学, 2009, 29(10): 68-69.

[10] 鄂旭, 邵良杉, 周津, 等. 一种新的不完备信息系统属性约简算法 [J]. 重庆邮电大学学报: 自然科学版, 2010, 22(5): 648-651.

[11] 李金海, 吕跃进. 决策系统的快速属性约简算法 [J]. 电子科技大学学报, 2007, 36(6): 1237-1240.

[12] 杨习贝, 张再跃, 张明. 集值信息系统中的模糊优势关系粗糙集 [J]. 计算机科学, 2011, 38(2): 234-237.

[13] Meng Zuqiang, Shi Zhongzhi. A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets [J]. Information Sciences, 2009, 179: 2774-2793.

[14] 朱晓峰, 李玲娟, 徐小龙, 等. 基于 MapReduce 的关联规则增量更新算法 [J]. 计算机技术与发展, 2012, 22(4): 115-118.

[15] 钱进, 苗夺谦, 张泽华. 云计算环境下知识约简算法 [J]. 计算机学报, 2011, 34(12): 2332-2343.

[16] Xu Xiaowei, Jager J, Kriegel H P. A fast parallel clustering algorithm for large spatial databases [J]. Data Mining and Knowledge Discovery, 1999, 3(3): 263-290.

海量数据下不完备信息系统的知识约简算法

作者：[王添](#)，[姜麟](#)，[米允龙](#)，[WANG Tian](#)，[JIANG Lin](#)，[MI Yun-long](#)

作者单位：[昆明理工大学 理学院, 云南 昆明, 650500](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015(1)

引用本文格式：[王添](#).[姜麟](#).[米允龙](#).[WANG Tian](#).[JIANG Lin](#).[MI Yun-long](#) [海量数据下不完备信息系统的知识约简算法](#)[期刊论文]-[计算机技术与发展](#) 2015(1)