ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2013/07(01)-0035-11
doi: 10.3778/j.issn.1673-9418.1206048

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-51616056

MapReduce框架下并行知识约简算法模型研究*

钱 进[1,2,3],苗夺谦[1,3+],张泽华[1,3],张志飞[1,3]
1. 同济大学 计算机科学与技术系,上海 201804
2. 江苏理工学院 计算机工程学院,江苏 常州 213001
3. 同济大学 嵌入式系统与服务计算教育部重点实验室,上海 201804

Parallel Algorithm Model for Knowledge Reduction Using MapReduce*

QIAN Jin[1,2,3], MIAO Duoqian[1,3+], ZHANG Zehua[1,3], ZHANG Zhifei[1,3]
1. Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
2. School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu 213001, China
3. Key Laboratory of Embedded System & Service Computing, Ministry of Education of China, Tongji University, Shanghai 201804, China
+ Corresponding author: E-mail: dqmiao@tongji.edu.cn

QIAN Jin, MIAO Duoqian, ZHANG Zehua, et al. Parallel algorithm model for knowledge reduction using MapReduce. Journal of Frontiers of Computer Science and Technology, 2013, 7(1): 35-45.

Abstract: Knowledge reduction for massive datasets has attracted many research interests in rough set theory. Classical knowledge reduction algorithms assume that all datasets can be loaded into the main memory of a single machine, which are infeasible for large-scale data. Firstly, this paper analyzes the parallel computations among classical knowledge reduction algorithms. Then, in order to compute the equivalence classes and attribute significance on different candidate attribute sets, it designs and implements the Map and Reduce functions using data and task parallelism. Finally, it constructs the parallel algorithm framework model for knowledge reduction using MapReduce, which can be used to compute a reduct for the algorithms based on positive region, discernibility matrix or information entropy. The experimental results demonstrate that the proposed parallel knowledge reduction algorithms can efficiently process massive datasets on Hadoop platform.

Key words: MapReduce; rough set; knowledge reduction; data parallel; task parallel

* The National Natural Science Foundation of China under Grant Nos. 60970061, 61075056, 61103067 (国家自然科学基金); the Fundamental Research Funds for the Central Universities of China (中央高校基本科研业务费专项资金).
Received 2012-05, Accepted 2012-07.

摘 要:面向大规模数据进行知识约简是近年来粗糙集理论研究热点。经典的知识约简算法是一次性将小数据集装入单机主存中进行约简,无法处理海量数据。深入剖析了知识约简算法中的可并行性;设计并实现了数据和任务同时并行的Map和Reduce函数,用于计算不同候选属性集导出的等价类和属性重要性;构建了一种MapReduce框架下并行知识约简算法模型,用于计算基于正区域、基于差别矩阵或基于信息熵的知识约简算法的一个约简。在Hadoop平台上进行了相关实验,实验结果表明,该并行知识约简算法模型可以高效地处理海量数据集。

关键词: MapReduce;粗糙集;知识约简;数据并行;任务并行

文献标志码:A **中图分类号:**TP181

1 引言

随着数据库技术的迅速发展以及信息系统的广泛应用,各行各业已经积累了海量数据。大量数据背后隐藏着许多重要的信息,人们希望能够对其进行更高层次的分析,以便更好地利用这些数据。但是传统的集中式数据挖掘算法[1]已经无法处理海量数据。近几年,Google公司提出了分布式文件系统(Google file system, GFS)[2]和并行编程模式MapReduce[3],这为海量数据挖掘提供了基础设施,同时也给数据挖掘研究提出了新的挑战。

粗糙集理论中知识约简[4]是数据挖掘中知识获取的关键步骤,而目前已提出的知识约简算法,如基于正区域的知识约简算法[5-6]、基于信息熵的知识约简算法[7-8]、基于差别矩阵的知识约简算法[9-10]等,无法将海量数据装入到单机内存中。并行知识约简[11-13]是解决海量数据挖掘的一种途径。然而,传统的并行知识约简算法[11]利用任务并行来计算所有约简,把计算不同的约简看成不同的任务,将每个任务分发到客户机上(任务并行),然后每个客户机计算小数据集上约简,最后得到所有约简。对于海量数据,必须使用数据并行策略。文献[12]提出了并行约简概念,将大规模数据随机划分为若干个子决策表,然后分别对各个子决策表计算正区域个数,选择最优单个候选属性。文献[13]将数据集中子决策表作为小粒度,分别计算小粒度上的约简,然后融合各个约简,从而获得一个约简。对于不一致决策表,由于这两种方法在各个子决策表上计算正区域或约简时并不交换信息,只能得到一个近似约简。目前,MapReduce技术实现了数据并行,并且能够处理大规模数据,已经初步应用于机器学习[14]、数据挖掘[15-17]等领域。因此,有必要进一步深入探讨如何利用MapReduce技术来实现并行知识约简算法,构建一种并行知识约简算法框架模型。

本文通过具体分析现有的基于正区域的、基于差别矩阵和基于信息熵的知识约简算法中可并行化操作,提出了一种基于MapReduce的等价类算法,构建了并行知识约简算法框架模型,利用Hadoop开源平台实现了数据和任务同时并行的知识约简算法。实验结果表明,该算法不仅具有高效性,而且能够处理大规模数据。

2 相关理论

2.1 粗糙集相关概念

下面,简要介绍粗糙集理论相关概念,其详细定义可参考文献[4,7-8,10,17]。

定义1[4] 四元组 $S=<U,C\cup D,V,f>$ 是一个决策表,其中 $U=\{x_1,x_2,\cdots,x_n\}$ 表示对象的非空有限集合,称为论域;C 表示条件属性的非空有限集,D 表示决策属性的非空有限集,$C\cap D=\varnothing$;$V=\bigcup_{a\in C\cup D}V_a$,$V_a$ 是属性 a 的值域;$f:U\times(C\cup D)\to V$ 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 $R\subseteq C\cup D$ 决定了一个二元不可区分关系 $IND(R)$:

$$IND(R)=\{(x,y)\in U\times U|\forall a\in R,f(x,a)=f(y,a)\}$$

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

不失一般性，假设决策表 S 仅有一个决策属性 $D=\{d\}$，其决策属性值映射为 $1,2,\cdots,k$，由 D 导出的 U 上划分记为 $U/D=\{D_1,D_2,\cdots,D_k\}$，其中 $D_i=\{x\in U|f(x,D)=i\},i=1,2,\cdots,k$。

定义2[4] 在决策表 S 中，对于每个决策类 $D_i\in U/D$ 和不可区分关系 $A\subseteq C$，D_i 的下近似集与上近似集分别可以由 A 的基本集定义为：

$$\underline{apr}_A(D_i)=\cup\{x\in U|[x]\subseteq D_i\}$$

$$\overline{apr}_A(D_i)=\cup\{x\in U|[x]\cap D_i\neq\varnothing\}$$

定义3[10] 在决策表 S 中，$\forall A\subseteq C$，正区域 $POS(D|A)$ 和边界域 $BND(D|A)$ 定义为：

$$POS(D|A)=\bigcup_{1\leqslant i\leqslant k}\underline{apr}_A(D_i)$$

$$BND(D|A)=\bigcup_{1\leqslant i\leqslant k}(\overline{apr}_A(D_i)-\underline{apr}_A(D_i))=U-POS(D|A)$$

定义4[7-8] 在决策表 S 中，$A\subseteq C$，A、D 在 U 上导出的划分分别为 $U/A=\{A_1,A_2,\cdots,A_r\}$，$U/D=\{D_1,D_2,\cdots,D_k\}$，则 A、D 在 U 的子集组成的 σ -代数上定义的概率分布为：

$$[U/A;p]=\begin{bmatrix}A_1 & A_2 & \cdots & A_r\\ p(A_1) & p(A_2) & \cdots & p(A_r)\end{bmatrix}$$

$$[U/D;p]=\begin{bmatrix}D_1 & D_2 & \cdots & D_k\\ p(D_1) & p(D_2) & \cdots & p(D_k)\end{bmatrix}$$

其中，$p(A_i)=\frac{|A_i|}{|U|}(i=1,2,\cdots,r),p(D_j)=\frac{|D_j|}{|U|}(j=1,2,\cdots,k)$，$|X|$表示集合 X 的基数。

定义5[7-8] 设 $U/A=\{A_1,A_2,\cdots,A_r\}$，$U/D=\{D_1,D_2,\cdots,D_k\}$，定义知识（属性集合）$A$ 的熵 $H(A)$ 为：

$$H(A)=-\sum_{i=1}^{r}p(A_i)\mathrm{lb}\,p(A_i)$$

知识（属性集合）D 相对于知识（属性集合）A 的条件熵 $H(D|A)$ 定义为：

$$H(D|A)=-\sum_{i=1}^{r}p(A_i)\sum_{j=1}^{k}p(D_j|A_i)\mathrm{lb}\,p(D_j|A_i)$$

其中，$p(D_j|A_i)=\frac{|D_j\cap A_i|}{|A_i|}(i=1,2,\cdots,r;j=1,2,\cdots,k)$。

定义6[17] 在决策表 S 中，$A\subseteq C,U/D=\{D_1,D_2,\cdots,D_k\}$，相对决策属性不可辨识关系定义为：

$$NDIS_A^D=\{<x,y>|\forall a\in A[f(x,a)=f(y,a)],x\in D_i,y\in D_j\}$$

其中，$D_i\in U/D,D_j\in U/D,1\leqslant i<j\leqslant k$。

属性集 A 不能辨识的对象对总数为：

$$NDIS(D|A)=\sum_{1\leqslant p\leqslant r1\leqslant i<j\leqslant k}n_p^in_p^j$$

其中，n_p^i 表示 A_p 中决策属性映射值为 i 的对象个数 $(p=1,2,\cdots,r;i=1,2,\cdots,k)$。

例1 表1为一个“相容”决策表，表中“?”为所有不相容对象的决策属性值。$U/C_1=\{\{1\},\{2,6\},\{3,4,5,7\}\},U/D=\{\{3\},\{4,5\},\{1\},\{2\},\{6,7\}\}$。根据定义3、定义5和定义6，则

$$POS(D|C_1)=1,BND(D|C_1)=6$$

$$H(D|C_1)=-\frac{2}{7}(\frac{1}{2}\mathrm{lb}(\frac{1}{2})+\frac{1}{2}\mathrm{lb}(\frac{1}{2}))-\frac{4}{7}(\frac{1}{4}\mathrm{lb}(\frac{1}{4})+\frac{1}{2}\mathrm{lb}(\frac{1}{2})+\frac{1}{4}\mathrm{lb}(\frac{1}{4}))=1.143$$

$$NDIS(D|C_1)=1\times1+[1\times(2+1)+2\times1]=6$$

Table 1　A “consistent” decision table

表1　一个“相容”决策表

U	C_1	C_2	C_3	C_4	C_5	D
1	1	1	2	1	3	3
2	2	1	1	1	3	4
3	3	1	2	2	2	1
4	3	3	3	3	2	2
5	3	3	3	3	3	2
6	2	1	1	2	1	?
7	3	1	3	1	2	?

2.2　云计算技术

为了解决海量数据的存储和计算问题，Google率先提供了GFS和MapReduce等云计算技术。MapReduce是一种处理海量数据的并行编程模式。用户不必关注MapReduce如何进行数据分割、负载均衡、容错处理等细节，只需要将实际应用问题分解成若干可并行操作的子问题，设计相应的Map和Reduce两个函数，就能将自己的应用程序运行在分布式系统上。其形式如下：

$$\mathrm{Map}:<in\_key,in\_value>\rightarrow\{<key_i,value_i>|i=1,2,\cdots,m\}$$

Reduce: $(key,[value_1,value_2,\cdots,value_k])\rightarrow$
$<final\_key,final\_value>$

Map函数是接收一组输入键值对 $<in\_key,in\_value>$，然后通过某种计算，产生一组中间结果键值对$<key_i,value_i>(i=1,2,\cdots,m)$；而Reduce函数对具有相同 *key* 的一组 *value* 值进行归并处理，最终形成 $<final\_key,final\_value>$。通过MapReduce编程模型，可以实现面向海量数据的知识约简算法。

3 基于MapReduce的知识约简算法模型

3.1 基于MapReduce的知识约简算法中数据和任务并行性

从定义3、定义5和定义6可知，基于正区域（边界域）、基于差别矩阵（相对不可辨识关系）[10]和基于信息熵的计算形式都可以统一为 $\sum_{1\leqslant i\leqslant l}\square$（□为同一个等价类中某种计算），而<等价类，等价类中某种计算>与 $<key,value>$ 类似，故可以利用MapReduce并行计算等价类。

定义7 在决策表 S 中，$DS_i(i=1,2,\cdots,n)$ 为数据集 S 的子决策表，若 $S=\bigcup_{i=1}^{n}DS_i, DS_i\cap DS_j=\varnothing(i,j=1,2,\cdots,n;i\neq j)$，则子决策表 DS_i 称为 S 的数据分片。

定理1 在决策表 S 中，$A\subseteq C,U/A=\{A_1,A_2,\cdots,A_r\}$，$DS_i$ 为 S 的数据分片 $(i=1,2,\cdots,n)$，$S=\bigcup_{i=1}^{n}DS_i$，$DS_i/A=\{A_{i1},A_{i2},\cdots,A_{ir}\}$，则 $A_k=\bigcup_{i=1}^{n}A_{ik}(k=1,2,\cdots,r)$。

证明 由 A 导出的 U、DS_i 上划分分别记为 $U/A=\{A_1,A_2,\cdots,A_r\}$，$DS_i/A=\{A_{i1},A_{i2},\cdots,A_{ir}\}$。假设属性集 A 在某个对象上条件属性组合的映射值为 k，则所有在条件属性集上映射值为 k 的对象形成一个等价类，即在 S 上条件属性集的映射值为 k 的对象数应该等于在各个数据分片上条件属性集的映射值为 k 的对象数之和。由于 $S=\bigcup_{i=1}^{n}DS_i$，则有 $A_k=\bigcup_{i=1}^{n}A_{ik}$。 □

定理1表明在一个大规模数据集上计算候选属性集的等价类，与在若干个数据分片上分别计算并汇总得到的等价类是等价的。根据定理1，可以实现数据并行策略。

由于基于正区域的、基于不可辨识关系的和基于信息熵的知识约简算法都需要计算等价类，而MapReduce技术能够将大规模数据分解成若干个数据分片，且各个数据分片独立不重叠，故可进行面向海量数据的等价类计算，从而实现基于MapReduce的并行知识约简算法。在并行知识约简算法中，主要编写Map方法来完成不同数据块中等价类计算，Reduce方法来计算同一个等价类中正区域（边界域）对象的个数、信息熵或不可辨识的对象对个数。图1显示了基于MapReduce的知识约简算法中数据和任务并行性。

本文所提出的基于MapReduce的知识约简算法采用数据和任务同时并行方式，即先将大规模数据划分为多个数据分片（数据并行）；然后对各个数据分片并行计算不同的候选属性集导出的等价类，汇总等价类后计算各个候选属性的重要性（任务并行）；最后统计各个任务中候选属性的重要性来确定最佳候选属性。

3.2 基于MapReduce的并行知识约简算法模型

这里先给出四种属性重要性测度定义。

定义8 在决策表 S 中，$A\subseteq C,\forall c\in C-A$，在正区域下属性 c 重要性定义为：

$$Sig_{\mathrm{Pos}}(c,A,D)=|POS(D|A\cup c)|-|POS(D|A)|$$

定义9 在决策表 S 中，$A\subseteq C,\forall c\in C-A$，在边界域下属性 c 重要性定义为：

$$Sig_{\mathrm{Bnd}}(c,A,D)=|BND(D|A)|-|BND(D|A\cup c)|$$

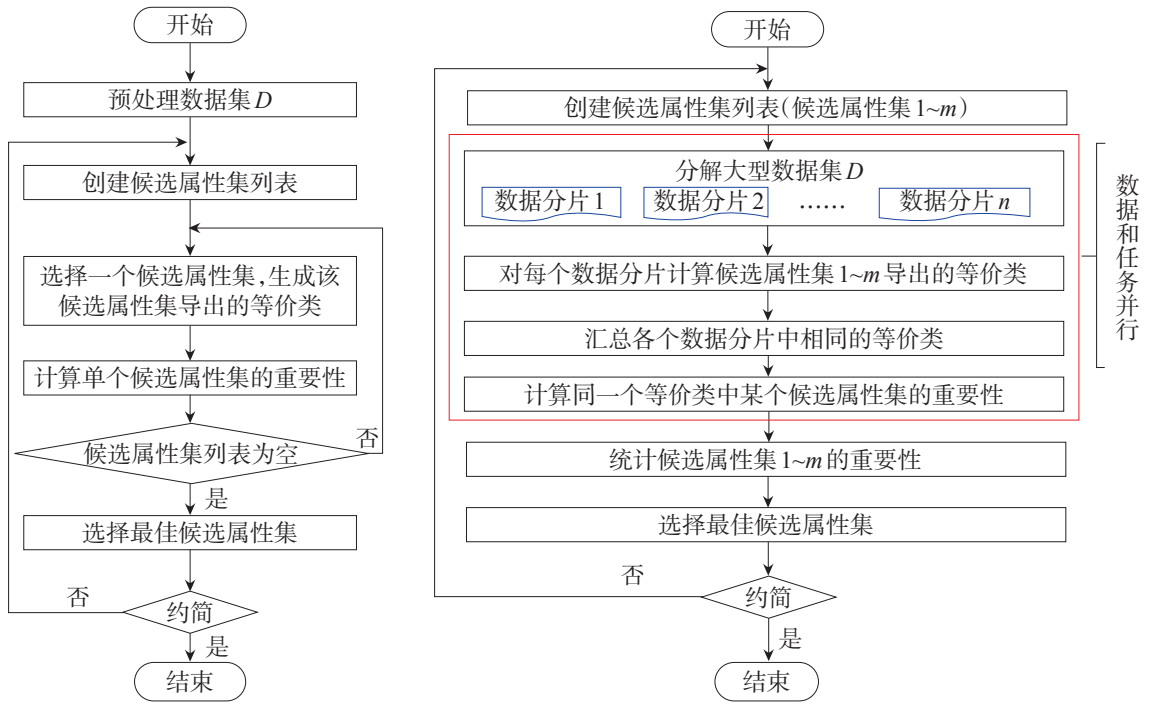
定义10 在决策表 S 中，$A\subseteq C,\forall c\in C-A$，在信息熵下属性 c 重要性定义为：

$$Sig_{\mathrm{Info}}(c,A,D)=H(D|A)-H(D|A\cup c)$$

定义11 在决策表 S 中，$A\subseteq C,\forall c\in C-A$，在相对不可辨识关系下属性 c 重要性定义为：

$$Sig_{\mathrm{NDIS}}(c,A,D)=NDIS(D|A)-NDIS(D|A\cup c)$$

说明 定义8和定义9只是计算公式不同，对于小数据集来说，计算效率差不多，但对于高维数据集来说，计算效率相差很大，为方便后面讨论，在此列出。定义11也可表示属性 c 在差别矩阵中的重要性，详细讨论见文献[17]。



（a）The flowchart of classical knowledge reduction algorithm

（a）传统的知识约简算法流程图

（b）The flowchart of knowledge reduction algorithm in cloud computing

（b）云计算环境下知识约简算法流程图

Fig.1　Data and task parallelism in knowledge reduction algorithm using MapReduce

图1　基于MapReduce的知识约简算法中数据和任务并行性

令 $\Delta=\{\text{Pos},\text{Bnd},\text{Info},\text{NDIS}\}$，将定义8~定义11中所定义的属性 c 重要性统一表示为 $Sig_{\Delta}(c,A,D)$，$\Delta(D|C)$ 表示Pos、Bnd、Info、NDIS下的分类能力。

定理2　在相容决策表 S 中，$A\subseteq C$，A 是 C 相对于决策属性 D 的一个约简的充分必要条件为：

（1）$\Delta(D|A)=\Delta(D|C)$

（2）$\forall a\in A,\Delta(D|[A-\{a\}])<\Delta(D|A)$

证明　由定义3、定义5和定义6容易证得。　□

记基于正区域（边界域）、信息熵和相对不可辨识关系的知识约简算法所获得的约简为 Red_{Pos}（Red_{Bnd}）、Red_{Info} 和 Red_{NDIS}，其对应的并行知识约简算法所获得的约简为 $Red_{\text{DTP-Pos}}$（$Red_{\text{DTP-Bnd}}$）、$Red_{\text{DTP-Info}}$ 和 $Red_{\text{DTP-NDIS}}$。

定理3　在决策表 S 中，Red_{Pos}、Red_{Bnd}、Red_{Info}、Red_{NDIS} 和 $Red_{\text{DTP-Pos}}$、$Red_{\text{DTP-Bnd}}$、$Red_{\text{DTP-Info}}$ 和 $Red_{\text{DTP-NDIS}}$ 为 S 约简结果，则有 $Red_{\text{Pos}}=Red_{\text{DTP-Pos}}$、$Red_{\text{Bnd}}=Red_{\text{DTP-Bnd}}$、$Red_{\text{Info}}=Red_{\text{DTP-Info}}$ 和 $Red_{\text{NDIS}}=Red_{\text{DTP-NDIS}}$。

证明　由定理1可知，利用MapReduce技术在各个数据分片中计算等价类，最后汇总的等价类与在整个 S 上计算的等价类相同。由定义3、定义5、定义6和定理2，容易证得 $Red_{\text{Pos}}=Red_{\text{DTP-Pos}}$、$Red_{\text{Bnd}}=Red_{\text{DTP-Bnd}}$、$Red_{\text{Info}}=Red_{\text{DTP-Info}}$ 和 $Red_{\text{NDIS}}=Red_{\text{DTP-NDIS}}$。　□

具体而言，基于MapReduce的知识约简算法主要包括Map方法（算法1）、Reduce方法（算法2）和主程序（算法3）。

算法1　Map()

输入：已选属性集合 A，候选属性 $c\in C-A$，数据分片 DS_i。

输出：$<Ac\_EquivalenceClass,<d(x),1>>$。

//*Ac_EquivalenceClass* 是由 $A\cup c$ 导出的等价类

1. For $x\in DS_i$ do

2.　For $c\in C-A$ do

Emit $<Ac\_EquivalenceClass,<d(x),1>>$

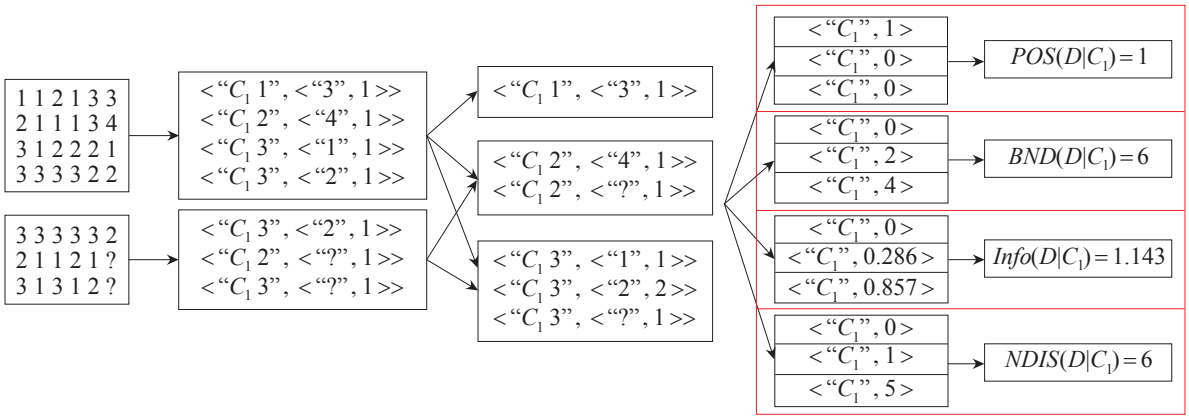


Fig.2 The result of the attribute significance on C_1 among four knowledge reduction algorithms

图2 四种知识约简算法计算属性 C_1 重要性的结果

算法 2 Reduce(string *Ac_EquivalenceClass*, *pairs* $[<d_1,n_1>,<d_2,n_2>,\cdots])$

输入：等价类 *Ac_EquivalenceClass* 及对应的决策值列表 *pairs*。

输出：$<c\_EquivalenceClass, Sig_{\Delta}^{c}>$。

// Sig_{Δ}^{c} 是该等价类中属性 c 的重要性

1. For $<d,n>\in[<d_1,n_1>,<d_2,n_2>,\cdots]$ do

{统计不同决策值出现的次数 $(n_p^1, n_p^2, \cdots, n_p^k)$；}

2. 根据 $Sig_{\Delta}(c, A, D)$ 计算公式，计算该等价类中属性 c 的重要性 Sig_{Δ}^{c}；

3. 输出 $<c\_EquivalenceClass, Sig_{\Delta}^{c}>$。

算法3 主程序

输入：一个决策表 S。

输出：一个约简 Red。

1. $Red=\varnothing$；

2. 计算 $\Delta(D|C)$；

3. 启动一个 Job，执行算法 1 的 Map 方法和算法 2 的 Reduce 方法，根据计算出的 $Sig_{\Delta}^{c}(c\in C-Red)$，选择 $c_l=\{c|\underset{c\in C-Red}{Best}(Sig_{\Delta}^{c})\}$（若这样的 c_l 不唯一，则任选其一），$Red=Red\cup\{c_l\}$；

4. 重复步骤 3，直到 $\Delta(D|Red)=\Delta(D|C)$；

5. 启动一个 Job，执行算法 1 的 Map 方法和算法 2 的 Reduce 方法，从 Red 的尾部开始从后往前判断每个属性 c 是否可省，若 $\Delta(D|Red)=\Delta(D|Red-c)$，则说明 c 是可省的，$Red=Red-\{c\}$；

6. 输出 Red。

算法 1 计算各个数据分片中不同等价类；算法 2 统计同一个等价类中不同决策值出现次数，并根据 $Sig_{\Delta}(c, A, D)$ 来计算相应属性 c 的重要性；而算法 3 根据各个单个候选属性的重要性，确定一个最优候选属性。重复上述过程，直到计算出约简。

说明 基于边界域的、基于不可辨识关系的和基于信息熵的知识约简算法都是计算边界域中对象的不确定性，所以输出的 $<key, value>$ 的形式相同。而基于正区域的知识约简算法是输出正区域中对象的个数，如图 2 所示。因此，基于边界域的、基于不可辨识关系的和基于信息熵的并行知识约简算法可以统一成一种并行知识约简算法模型。

4 实例分析与实验结果

4.1 实例分析

用一个“相容决策表”（表 1）说明本文基于 MapReduce 的并行知识约简算法模型。假设将表 1 划分为两个数据分片，第 1 个数据分片包含第 1~4 条对象，第 2 个数据分片包含第 5~7 条对象，则整个计算过程如图 2 所示，这与例 1 的计算结果一致。

4.2 理论分析

下面，先从理论上分析加速比 *Speedup*[15]。设 T_1 表示单节点运行时间，T_m 表示多节点运行时间，其加速比记为 $Speedup=\frac{T_1}{T_m}$。具体为：

$$T_1(D)=T_{pp}(D)+T_{sp}(D)$$

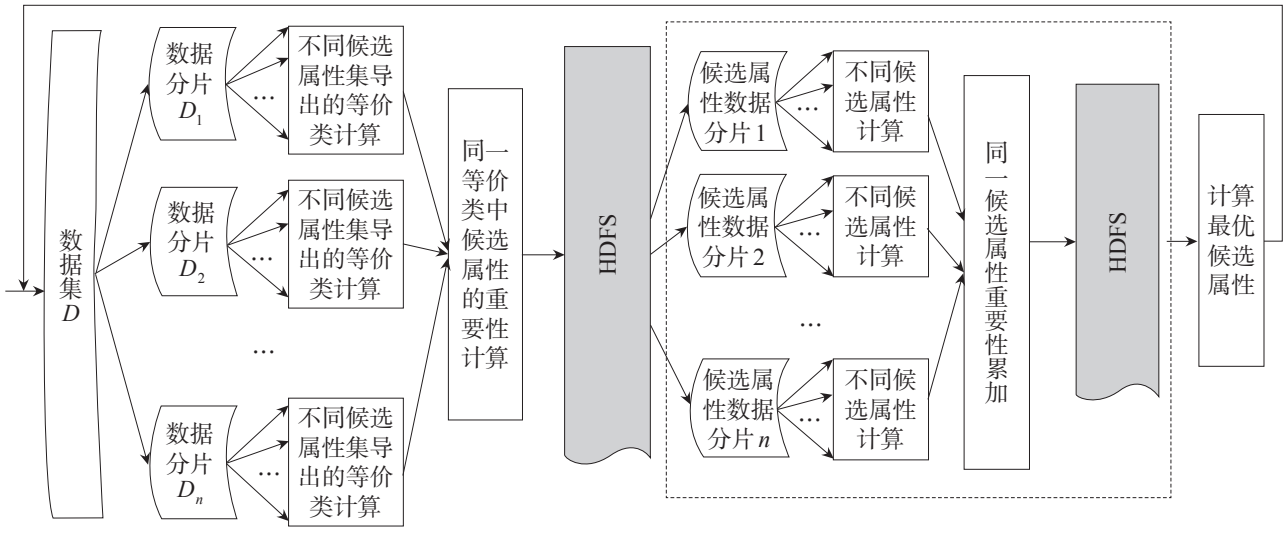


Fig.3　Data and task parallelism more times in knowledge reduction algorithms using MapReduce

图3　基于MapReduce的知识约简算法中数据和任务多次并行

$$T_{\mathrm{m}}(D,N)=\frac{T_{\mathrm{pp}}(D)}{N}+T_{\mathrm{sp}}(D)+T_{\mathrm{cp}}(D,N)$$

$$T_{\mathrm{cp}}(D,N)=(N-1)\left(c\times\frac{D}{S\times N}+c'\right)$$

其中，$T_{\mathrm{pp}}(D)$ 是对数据集 D 的并行处理时间；$T_{\mathrm{sp}}(D)$ 是对数据集 D 的串行处理时间；N 表示节点个数；$T_{\mathrm{cp}}(D,N)$ 表示 N 节点下对数据集 D 的通信处理时间；S 表示单个数据块的大小；c 表示对每个数据分片的传输时间；c' 表示节点间建立通话连接的通信时间。于是有

$$Speedup=\frac{T_1(D)}{T_{\mathrm{m}}(D,N)}=\frac{T_{\mathrm{pp}}(D)+T_{\mathrm{sp}}(D)}{\frac{T_{\mathrm{pp}}(D)}{N}+T_{\mathrm{sp}}(D)+(N-1)\left(c\times\frac{D}{S\times N}+c'\right)}<\frac{T_{\mathrm{pp}}(D)+T_{\mathrm{sp}}(D)}{\frac{T_{\mathrm{pp}}(D)}{N}+\frac{T_{\mathrm{sp}}(D)}{N}}=N$$

说明　基于MapReduce的并行知识约简算法很难达到理想的加速比 N。除了通信开销以外，由于知识约简算法中计算各个候选属性的重要性是串行过程，当处理海量高维数据集时，串行时间将会很长，其加速比显著降低。这时，可以再次考虑数据并行方式，如图3所示，其中虚线部分为并行计算候选属性重要性。

4.3　实验测试与分析

4.3.1　实验环境

为了进一步考察本文算法，选用UCI机器学习数据库（http://www.ics.uci.edu/~mlearn/MLRepository.html）中mushroom和gisette两个数据集，分别复制5 000次和17次形成DS4和DS5，以及人工数据集DS1~DS3和DS6来测试算法性能。表2列出了不同数据集的特性。利用开源云计算平台Hadoop 0.20.2和Java 1.6.0_20实现了基于MapReduce的并行知识约简算法——基于正区域的算法（DTP-Pos算法）、基于边界域的算法（DTP-Bnd算法）、基于信息熵的算法（DTP-Info算法）和基于差别矩阵的（相对不可辨识关系）算法（DTP-NDIS算法），并在9台计算机（Intel Pentium Q8400 2.6 GHz quad-core CPU，3 GB内存）构

Table 2　The characteristics of datasets

表2　数据集特性

数据集	对象数	条件属性数	决策属性值个数
DS1	10 000 000	30	10
DS2	20 000 000	30	10
DS3	40 000 000	30	10
DS4	40 620 000	22	2
DS5	102 000	5 000	2
DS6	100 000	10 000	10

建的云计算环境下进行了实验,其中1台为主节点,8台为从节点。

4.3.2 运行时间

对6个数据集DS1~DS6在8个从节点下进行测试,运行时间如图4所示。从图4可以看出,Bnd算法、Info算法和NDIS算法是十分相似的,因为它们都是计算边界域中信息不确定性,而Pos算法是随着属性个数增加,其单次运行时间不断增长。图5比较了Pos和Bnd两种算法,加速率= T_{Pos} / T_{Bnd} ,当遇到高维数据集时运行时间存在明显差异。这是因为Pos算

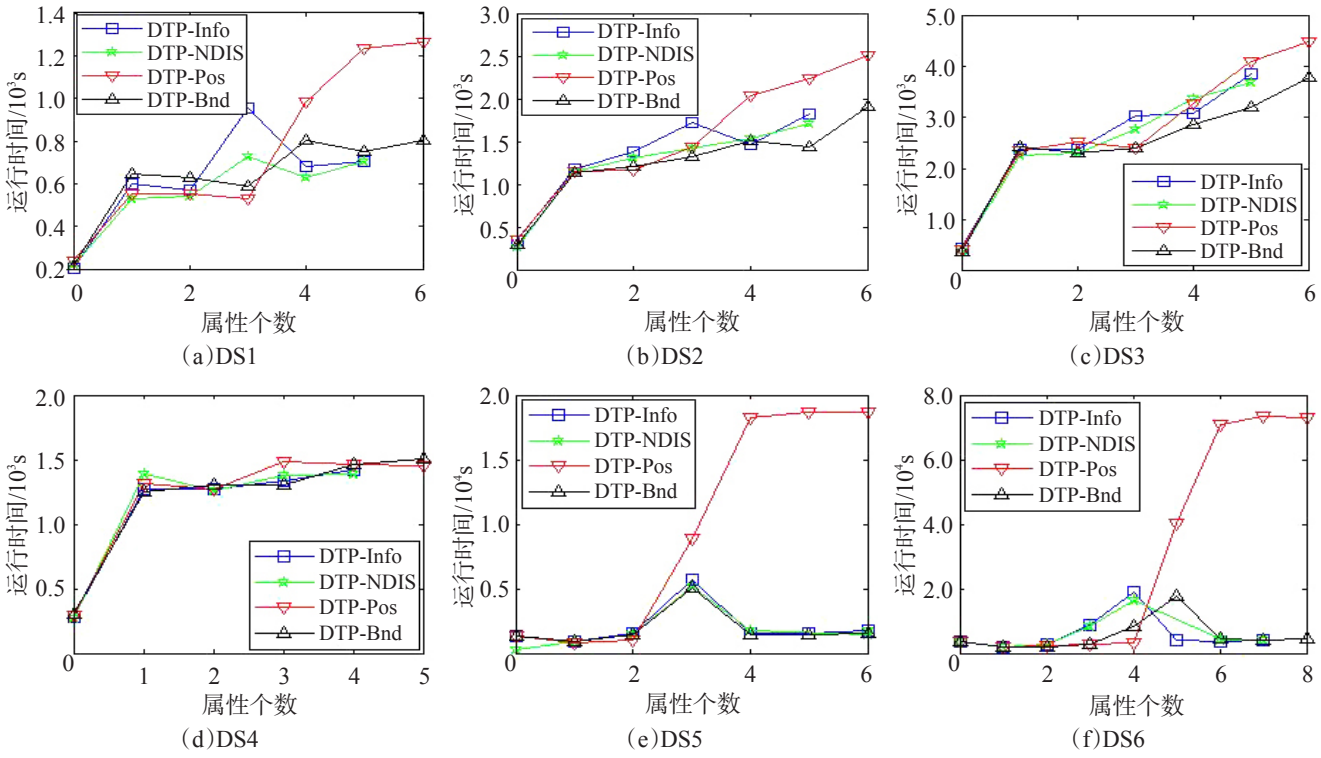


Fig.4 The running time on DS1~DS6 among different knowledge reduction algorithms

图4 不同知识约简算法在DS1~DS6上的运行时间

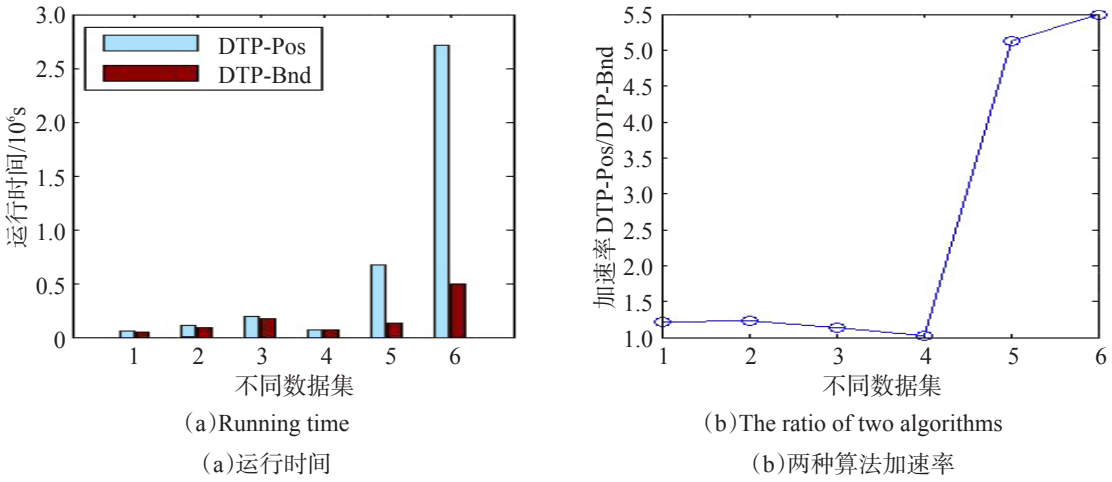
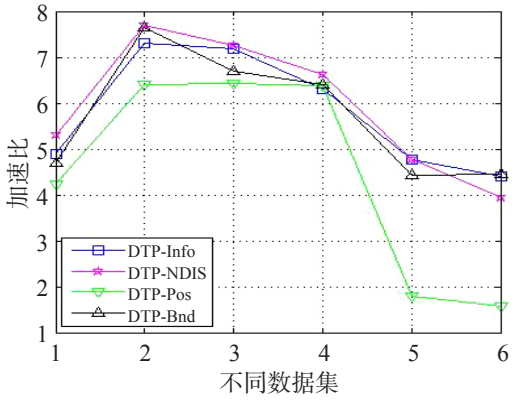


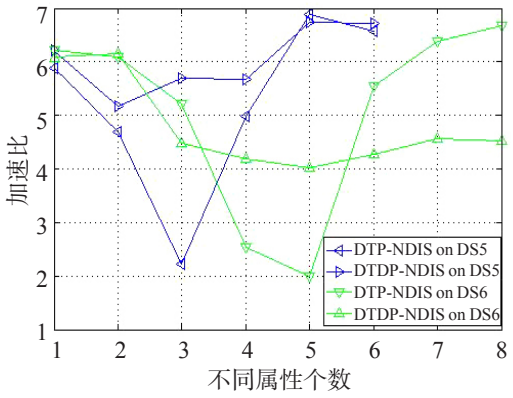
Fig.5 The comparison of DTP-Pos and DTP-Bnd algorithms

图5 DTP-Pos和DTP-Bnd两种算法比较



(a)The speedup of four algorithms

(a)四种算法加速比



(b)The speedup for computing each attribute

(b)计算单个属性的加速比

Fig.6　The speedups of four algorithms and each attribute

图6　四种算法和单个属性的加速比

法随着属性个数增加，在Reduce阶段生成了巨大的<*key*, *value*>，造成串行计算时间过长，而Bnd算法随着属性个数增加，运行时间先增加后减少，其串行计算效率较高。

4.3.3　加速比

分别对数据集DS1~DS6进行测试，其加速比如图6所示。从图6可以看出，基于MapReduce的知识约简算法的加速比并不呈线性比。这是因为基于MapReduce的知识约简算法除了进行Map和Reduce、数据传输和网络通信等并行操作，还存在串行计算。当属性个数较少(数据集DS1~DS5)时，基于MapReduce的知识约简算法加速比较好。DS1、DS5和DS6加速比相对较差，其中DS1是因为数据集偏小，部分节点处于空闲状态，没有发挥作用。当遇到高维数据集(DS5和DS6)时，知识约简算法的加速比较低，尤其是DTP-Pos算法，主要是因为串行计算时间相对较长，这时可以考虑数据和任务多次并行方式(图3)。

下面主要剖析NDIS算法。NDIS算法中的串行计算可以采用数据并行方式，本文将这种算法记为DTDP-NDIS算法。图6(b)显示了DTP-NDIS和DTDP-NDIS算法选择单个属性的加速比。从图6(b)可以看出，DTDP-NDIS在许多单个属性选择上加速比高于DTP-NDIS。DTDP-NDIS算法在DS5上选择第2、3和4个候选属性和在DS6上选择第4和5个候选属性时，加速比最好。这是因为Reduce阶段产生了巨大的<*key*, *value*>，这时再次采用数据并行方式，能够大大降低原串行计算时间；而选择其他属性时，并行计算时间与串行计算时间相当或更长。至于何时再次采用数据并行方式值得进一步研究。

4.3.4　规模增长性

为了测试规模增长性，将数据集DS2分别复制1、2、4和8倍在8节点下运行，如图7所示。实验结果表明，知识约简算法的规模增长性较好。

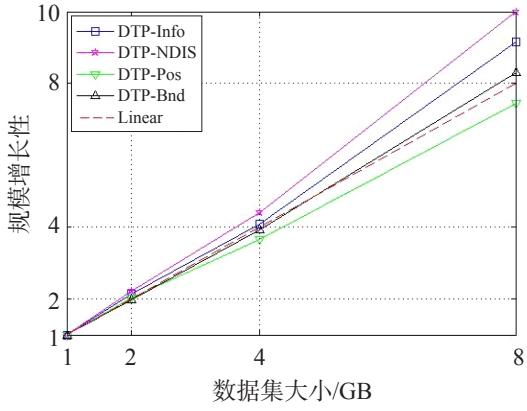


Fig.7　The scaleup for dataset DS2

图7　数据集DS2规模增长性

4.4　讨论

文献[16]先利用MapReduce技术分解大规模数据，对每个数据分片进行约简，然后合并各个数据分

片的约简,再增加其他必要的候选属性,最后删除冗余属性。然而,该方法中合并后的候选约简有可能是整个条件属性集,删除冗余属性则变得十分困难。文献[17]利用MapReduce技术对各个数据分片计算不同候选属性集的等价类,然后汇总并合并相同的等价类,计算边界域中对象对个数,选择最优单个候选属性,迭代此过程,直到获得约简。本文在文献[17]基础上,先采用数据并行策略,再进行任务并行操作,构建了一种基于MapReduce的并行知识约简算法模型,这样大大节省了MapReduce作业启动与调度时间,从而提高了基于MapReduce的知识约简算法的效率。

5 结论

为了进行面向大规模数据集的知识约简,通过深入分析传统的基于正区域、基于信息熵和基于差别矩阵(或相对不可辨识关系)的知识约简算法中可并行化操作,提出了基于MapReduce的并行知识约简算法框架模型。利用云计算开源平台Hadoop在普通计算机的集群上进行了实验,实验结果表明,本文的知识约简算法可以处理大规模数据集。

References:

[1] Han J, Kamber M. Data mining—concepts and techniques[M]. 2nd ed. [S.l.]: Morgan Kaufman Publishers, 2006.

[2] Ghemawat S, Gobioff H, Leung S T. The Google file 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]. Communications of the ACM, 2008, 51 (1): 107-113.

[4] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356.

[5] Xu Zhangyan, Liu Zuopeng, Yang Bingru, et al. A quick attribute reduction algorithm with complexity of max($O(|C||U|)$, $O(|C|^2|U/C|)$)[J]. Chinese Journal of Computers, 2006, 29(3): 391-399.

[6] Qian Yuhua, Liang Jiye, Pedrycz W, et al. Positive approximation: an accelerator for attribute reduction in rough set theory[J]. Artificial Intelligence, 2010, 174(9/10): 597-618.

[7] Miao Duoqian, Hu Guirong. A heuristic algorithm for reduction of knowledge[J]. Journal of Computer Research and Development, 1999, 36(6): 681-684.

[8] Wang Guoyin, Yu Hong, Yang Dachun. Decision table reduction based on conditional information entropy[J]. Chinese Journal of Computers, 2002, 25(7): 759-766.

[9] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[M]//Słowiński R. Intelligent Decision Support Handbook of Applications and Advances of the Rough Set Theory. Dordrecht: Kluwer Academic Publishers, 1992: 311-362.

[10] Qian Jin, Miao Duoqian, Zhang Zehua, et al. Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation[J]. International Journal of Approximate Reasoning, 2011, 52(2): 212-230.

[11] Wang Lihong, Wu Gengfeng. Attribute reduction based on parallel symbiotic evolution[J]. Chinese Journal of Computers, 2003, 26(5): 630-635.

[12] Deng Dayong, Yan Dianxun, Wang Jiyi. Parallel reducts based on attribute significance[C]//Proceedings of the 5th International Conference on Rough Set and Knowledge Technology (RSKT '10) , Beijing, 2010. Berlin, Heidelberg: Springer-Verlag, 2010: 336-343.

[13] Liang Jiye, Wang Feng, Dang Chuangyin, et al. An efficient rough feature selection algorithm with a multi-granulation view[J]. International Journal of Approximate Reasoning, 2012, 53(6): 912-926.

[14] Chu C T, Kim S, Lin Y A, et al. MapReduce for machine learning on multicore[C]//Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS 2006). Cambridge, MA: MIT Press, 2006: 281-288.

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

[16] Yang Yong , Chen Zhengrong, Liang Zhu, et al. Attribute reduction for massive data based on rough set theory and MapReduce[C]//Proceedings of the 5th International Conference on Rough Set and Knowledge Technology (RSKT '10), Beijing, 2010. Berlin, Heidelberg: Springer-Verlag, 2010: 672-678.

[17] Qian Jin, Miao Duoqian, Zhang Zehua. Knowledge reduction algorithms in cloud computting[J]. Journal of Computers, 2011, 34(12): 2332-2343.

附中文参考文献:

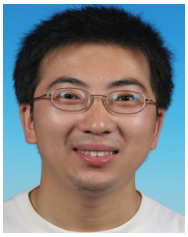
[5] 徐章艳, 刘作鹏, 杨炳儒,等. 一个复杂度为max($O(|C||U|)$, $O(|C|^2|U/C|)$)的快速属性约简算法[J]. 计算机学报, 2006, 29(3): 391-399.

[7] 苗夺谦, 胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展, 1999, 36(6): 681-684.

[8] 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7): 759-766.

[11] 王立宏, 吴耿锋. 基于并行协同进化的属性约简[J]. 计算机学报, 2003, 26(5): 630-635.

[17] 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.

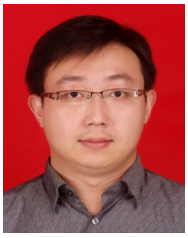


QIAN Jin was born in 1975. He is a Ph.D. candidate at Tongji University, a lecturer at Jiangsu University of Technology, and the member of CCF. His research interests include rough set theory, data mining and cloud computing, etc.

钱进(1975—),男,江苏泰兴人,同济大学博士研究生,江苏理工学院讲师,CCF会员,主要研究领域为粗糙集理论,数据挖掘,云计算等。

MIAO Duoqian was born in 1964. He received his Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences in 1997. Now he is a professor and Ph.D. supervisor at Tongji University, and the senior member of CCF. His research interests include rough set theory, granular computing, text mining and cloud computing, etc.

苗夺谦(1964—),男,山西祁县人,1997年于中国科学院自动化研究所获得博士学位,现为同济大学教授、博士生导师,CCF高级会员,主要研究领域为粗糙集,粒计算,文本挖掘,云计算等。



ZHANG Zehua was born in 1981. He is a Ph.D. candidate at Tongji University, and the student member of CCF. His research interests include rough set theory, granular computing and uncertainty reasoning, etc.

张泽华(1981—),男,山西怀仁人,同济大学博士研究生,CCF学生会员,主要研究领域为粗糙集理论,粒计算,不确定性推理等。

ZHANG Zhifei was born in 1986. He is a Ph.D. candidate at Tongji University, and the student member of CCF. His research interests include rough set theory and text mining, etc.

张志飞(1986—),男,江苏南通人,同济大学博士研究生,CCF学生会员,主要研究领域为粗糙集理论,文本挖掘等。