云计算环境下知识约简算法

钱 进[1),2),3)] 苗夺谦[1),3)] 张泽华[1),3)]

[1)](同济大学计算机科学与技术系 上海 201804)

[2)](江苏技术师范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)

摘 要 知识约简是粗糙集理论的重要研究内容之一.经典的知识约简算法是假设所有数据一次性装入内存中,这显然不适合处理海量数据.为此,从属性(集)的可辨识性和不可辨识性出发,给出了可辨识和不可辨识对象对的概念及其性质,并阐述了它们与差别矩阵的关系.利用 MapReduce 设计了并行计算等价类的方法,提出了面向大规模数据的数据并行知识约简算法,讨论并实现了 3 种并行策略.最后,通过实验表明了云计算环境下知识约简算法是有效可行的,具有较好的可扩展性.

关键词 云计算;粗糙集;知识约简;数据并行;MapReduce

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2011.02332

Knowledge Reduction Algorithms in Cloud Computing

QIAN Jin[1),2),3)] MIAO Duo-Qian[1),3)] ZHANG Ze-Hua[1),3)]

[1)](*Department of Computer Science and Technology, Tongji University, Shanghai* 201804)

[2)](*School of Computer Engineering, Jiangsu Teachers University of Technology, Changzhou, Jiangsu* 213001)

[3)](*Key Laboratory of Embedded System & Service Computing, Ministry of Education of China, Tongji University, Shanghai* 201804)

Abstract Knowledge reduction is one of the important research issues in rough set theory. Classical knowledge reduction algorithms assume all the datasets can be loaded into the main memory, which are infeasible for large-scale datasets. Massive data with high dimensions makes attribute reduction a challenging task. To this end, the concepts and properties of discernibility and indiscernibility object pairs are given in terms of the discernibility and indiscernibility of the attribute(s). The relationship between discernibility matrix and them is illustrated in detail. Then, an algorithm of computing equivalence classes is designed for large-scale data in data parallel, and the corresponding knowledge reduction algorithms are proposed in cloud computing. Finally, three parallelism strategies are implemented and discussed. The experimental results demonstrate that knowledge reduction algorithms in cloud computing can scale well and efficiently process massive datasets on commodity computers.

Keywords cloud computing; rough set; knowledge reduction; data parallel; MapReduce

1 引 言

随着数据库技术的发展和信息系统的广泛使用,各行各业已经积累了大规模数据.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.粗糙集理论(Rough Set Theory)[1]是一种新的研究不精确、不确定性知识的数据分析

收稿日期:2011-03-30;最终修改稿收到日期:2011-10-31.本课题得到国家自然科学基金(60970061,61075056,61103067)、中央高校基本科研业务费专项资金、江苏省属高校自然科学资金项目(09KJD520004)资助.**钱 进**,男,1975 年生,博士研究生,讲师,主要研究方向为粗糙集理论、数据挖掘、云计算.E-mail: qjqjlqyf@163.com. **苗夺谦**(通信作者),男,1964 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为粗糙集理论、粒计算、Web 智能与模式识别、云计算.E-mail: dqmiao@tongji.edu.cn. **张泽华**,男,1981 年生,博士研究生,主要研究方向为粗糙集理论、不确定性推理等.

理论，目前正被广泛地应用于机器学习、数据挖掘、模式识别等领域. 在粗糙集理论中，知识约简是重要的研究内容之一，也是知识获取的关键步骤. 所谓知识约简是指保持知识库的分类能力不变的条件下，删除其中不必要的知识. 通过删除冗余知识，可以大大提高信息系统潜在知识的清晰度. 因此，研究如何提高知识约简算法效率是十分重要的工作.

目前已提出许多知识约简算法[2-15]，主要分为基于正区域的属性约简算法[2-4]、基于信息论的属性约简算法[5-6]，基于差别矩阵及在此基础上改进的属性约简算法[7-10]等. 众多学者主要研究如何处理不相容决策表[11-12]和如何提高知识约简算法效率[4,13-15]. 文献[11]研究了不相容决策表中最大分布约简、分布约简和可能约简之间的关系. 文献[12]具体探讨了 Pawlak 粗糙集模型下各种知识约简算法中性质保持的含义，给出了一个广义的性质保持定义. 文献[4]提出了基于 Hash 的正区域计算方法和知识约简算法，将基于正区域的知识约简算法时间复杂度降为 $O(|C|^2|U/C|)$. 文献[13]利用正向近似思想，提出了一种知识约简框架模型，能够将基于信息熵的约简算法时间复杂度降为 $O\left(|U||C|+\sum_{i=1}^{|C|}(|C|-i+1)\right)$. 文献[14]利用计数排序算法能够将基于差别矩阵的知识约简算法时间复杂度降为 $\max(O(|C||U|),O(|C|^2|U/C|))$. 文献[15]给出了改进的差别矩阵定义，对不相容对象进行了处理，并利用原有的约简结果对增量数据进行属性约简更新，从而提高了知识约简算法效率. 所有这些算法是假设所有数据能够一次性装入内存中，这显然无法处理大规模数据.

并行知识约简可能是解决海量数据挖掘问题的一个重要途径，而已有的并行知识约简算法[16]将并行遗传算法和协同进化算法相结合，对属性约简任务进行分解，从而提高了知识约简算法的效率. 然而，此类知识约简算法[16-17]也是假设将所有数据一次性装入内存中，不适合处理海量数据. 云计算(Cloud Computing)是近几年新提出的一种商业计算模型，是分布式计算、并行计算和网格计算的发展. 云计算先行者之一的 Google 公司提出了一个具有海量数据存储和访问能力的大型分布式文件系统 GFS(Google File System)[18]，同时提供了一种处理海量数据的并行编程模式 MapReduce[19]，这为海量数据挖掘提供了一个可行的解决方案. 云计算技术已经初步应用于机器学习领域[20]，但至今还没有真正应用到知识约简算法中[21].

本文深入研究了 MapReduce 并行编程技术，对现有知识约简算法进行具体剖析，利用属性(集)的可辨识性和不可辨识性，给出了可辨识和不可辨识对象对的定义和相关性质，结合 MapReduce 技术设计了适合大规模数据集的并行计算等价类的算法，并利用 Hadoop 开源平台实现了云计算环境下知识约简算法. 实验结果表明该算法不仅具有较好的扩展性，而且能够很好地处理海量数据.

2　相关理论

2.1　粗糙集相关概念

下面简要介绍本文主要用到的一些 Rough 集的基本概念，详细内容请参考文献[1,3,14].

定义 1[1].　五元组 $S=\langle U,C,D,V,f\rangle$ 是一个决策表，其中 $U=\{x_1,x_2,\cdots,x_n\}$ 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 $A\subseteq C\cup D$ 决定了一个二元不可区分关系 $IND(A)$：

$$IND(A)=\{(x,y)\in U\times U\mid \forall a\in A,f(x,a)=f(y,a)\}.$$

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

定义 2[1].　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 $A\subseteq C\cup D$，X 的下近似集与上近似集分别可以由 A 的基本集定义如下：

$$\underline{A}(X)=\bigcup\{Y\in U/A:Y\subseteq X\},$$

$$\overline{A}(X)=\bigcup\{Y\in U/A:Y\cap X\neq\varnothing\}.$$

定义 3[1].　在决策表 $S=\langle U,C,D,V,f\rangle$ 中，$\forall A\subseteq C,X\subseteq U$，用 $\underline{A}X$ 表示 X 的下近似集，决策属性 D 的 A_正区域 $POS_A(D)$ 定义为

$$POS_A(D)=\bigcup_{X\in U/D}\underline{A}X.$$

定义 4[1].　在决策表 $S=\langle U,C,D,V,f\rangle$ 中，$a\in C$，若 $POS_{C-\{a\}}(D)\neq POS_C(D)$，则称属性 a 在 C 中是不可缺少的；C 中所有不可缺少的属性集合称为 C 的核(简称核)，记为 $Core(C)$.

定义 5[3].　在决策表 $S=\langle U,C,D,V,f\rangle$ 中，记

$U/C=\{[x'_1]_C,[x'_2]_C,\cdots,[x'_s]_C\}$,$U'=\{x'_1,x'_2,\cdots,x'_s\}$,$U'_{POS}=\{x'_{i_1},x'_{i_2},\cdots,x'_{i_t}\}$,其中,$U'_{POS}$中对象为相容对象,$U'_{BND}$为$U'-U'_{POS}$,则$S'=(U'=U'_{POS}\cup U'_{BND},C,D,V,f)$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,其决策属性值映射为$1,\cdots,k$,由D导出的U上划分记为$U/D=\{D_1,D_2,\cdots,D_k\}$,其中,$D_i=\{x\in U\mid f(x,D)=i\}(i=1,2,\cdots,k)$.

定义 6[14]. 在简化决策表S'中,将U'_{BND}中所有矛盾对象集记为D_{k+1},其决策属性值标记为"?",映射为$k+1$.若$D_{k+1}=\varnothing$,则称决策表S'是相容(一致)决策表;否则是不相容(不一致)决策表.

若将决策表S中所有矛盾对象归为D_{k+1}类,则新划分$\{D_1,\cdots,D_k,D_{k+1}\}$既将属于不同决策类中相容对象$(D_1,\cdots,D_k)$分开,又将相容对象与矛盾对象$D_{k+1}$分开,这样的不一致决策表就可以看成"相容"决策表了.因此,相容决策表不过是不一致决策表的"特例",$D_{k+1}=\varnothing$. 因此,下文将所有决策表都看成"相容决策表".

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(Google File System)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:

$$\text{Map}:\langle in\_key,in\_value\rangle\rightarrow\{\langle key_i,value_i\rangle\mid i=1,\cdots,m\},$$

$$\text{Reduce}:(key,[value_1,\cdots,value_k])\rightarrow\langle final\_key,final\_value\rangle.$$

Map函数是接收一组输入键值对$\langle in_key,in_value\rangle$,然后通过某种计算,产生一组中间结果键值对$\langle key_i,value_i\rangle(i=1,\cdots,m)$;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$\langle final_key,final_value\rangle$.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 云计算环境下知识约简算法

假设决策表S共有k个不同决策属性值,相容对象的决策属性值分别映射为$1,\cdots,k$,将所有不相容对象的决策属性值映射为$k+1$.这样,整个决策表S可以看成由$k+1$个子决策表$D_1,D_2,\cdots,D_k,D_{k+1}$组成,每个子决策表包含同一类别的对象,其对象个数分别为$n^1,n^2,\cdots,n^{k+1}$.因此,决策表S是"相容决策表".假设属性a有r个不同属性值,将其映射为$1,\cdots,r$.记D_i中条件属性a的属性映射值为p的对象个数为n^i_p.显然,$n^j_1+n^j_2+\cdots+n^j_r=n^j(j=1,\cdots,k+1)$,$n^1_1+\cdots+n^1_r+\cdots+n^{k+1}_1+\cdots+n^{k+1}_r=n$.

3.1 云计算环境下可辨识对象对计算方法

一个可辨识的对象对是由决策属性值不同和条件属性组合值也不同的两个对象生成的.如果两个对象决策值不同,同时条件属性a上属性值也不同,则a能够辨识这两个对象(一个可辨识的对象对),即a具有一定的相对辨识能力.a能够辨识的对象对个数越多,说明a相对辨识能力越强.这时,可以用可辨识的对象对个数多少来衡量a相对辨识能力大小.

定义 7. 在相容决策表S中,$a\in C$,属性a能够辨识的对象对为

$$DOP_a=\{\langle x,y\rangle\mid f(x,a)\neq f(y,a),x\in D_i,y\in D_j\},$$

其中,$1\leq i<j\leq k+1$.

定义 8. 在相容决策表S中,$A\subseteq C$,属性集A能够辨识的对象对为

$$DOP_A=\{\langle x,y\rangle\mid\exists a\in A,f(x,a)\neq f(y,a),x\in D_i,y\in D_j\},$$

其中,$1\leq i<j\leq k+1$.

定理 1. 在相容决策表S中,若$A\subseteq C$,$\forall a\in A$,则有

$$DOP_A=\bigcup_{a\in A}DOP_a.$$

证明. 由定义7和定义8可以直接证得. 证毕.

假设由A导出的U上划分有r个等价类,记$U/A=\{A_1,A_2,\cdots,A_r\}$,将其属性组合值映射为$1,\cdots,r$. A能够辨识的对象对个数可根据下面定义来计算.

定义 9. 在相容决策表S中,$A\subseteq C$,属性集A能够辨识的对象对个数为$DIS^D_A=\sum_{1\leq i<j\leq k+1}\sum_{1\leq p<q\leq r}n^i_pn^j_q$.

由定义9可知,可辨识对象对个数的计算方法涉及到计算属性集A和属性D在U上导出的等价类,然后对它们进行交运算操作,计算相对比较复杂,所以只能在内存中计算出属性集A在小规模数据集上可辨识的对象对个数. 然而在云计算环境下,由于海量数据的不同等价类存储在若干个节点和文件中,而计算DIS^D_A涉及到多个不同等价类,因此,无法根据定义9计算对象对个数.如何快速计算

可辨识的对象对个数成为云计算环境下知识约简算法的关键问题.下面重点研究云计算环境下可辨识对象对个数的两种计算方法.

3.1.1　云计算环境下可辨识对象对个数的直接计算方法(DIS)

众所周知,属性集 A 能够区分 U/A 中任意两个等价类,这说明 A 具有一定的辨识能力.如果 A 只能将所有元素划分到一个等价类中,则 A 具有最弱的辨识能力.因此,可以利用属性集 A 的辨识能力大小[22]来计算 A 的相对辨识能力大小 DIS_A^D.

定义 10.　在相容决策表 S 中,$A\subseteq C$,属性集 A 具有的辨识能力大小定义为

$$DIS_{U,A}=\sum_{1\le p<q\le r}n_p\times n_q.$$

定义 11.　在相容决策表 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 $A_1,A_2,\cdots,A_r$ 中新增加的辨识能力大小之和,即

$$DIS_{U,A\cup\{c\}}-DIS_{U,A}=DIS_{A_1,\{c\}}+DIS_{A_2,\{c\}}+\cdots+DIS_{A_r,\{c\}}.$$

下面,根据属性集 A、D 和 $A\cup D$ 的辨识能力来计算属性集 A 的相对辨识能力大小.

定理 2.　在相容决策表 S 中,若 $A\subseteq C$,则 $DIS_A^D=DIS_{U,A}+DIS_{U,D}-DIS_{U,A\cup D}$.

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,$n_1^j+n_2^j+\cdots+n_r^j=n^j$ $(j=1,\cdots,k+1)$,则有

$$DIS_{U,A}=\sum_{1\le p<q\le r}n_p\times n_q,\ DIS_{U,D}=\sum_{1\le i<j\le k+1}n^i\times n^j.$$

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

$$DIS_{U,A\cup D}=\sum_{1\le l_1<l_2\le r(k+1)}n_{l_1}n_{l_2}.$$

由定义 11 得出

$$DIS_{U,A\cup D}=DIS_{U,A}+\sum_{1\le p\le r}\sum_{1\le i<j\le k+1}n_p^in_p^j.$$

因此,

$$\begin{aligned}&DIS_{U,A}+DIS_{U,D}-DIS_{U,A\cup D}\\&=DIS_{U,A}+DIS_{U,D}-\Big(DIS_{U,A}+\sum_{1\le p\le r}\sum_{1\le i<j\le k+1}n_p^in_p^j\Big)\\&=\sum_{1\le i<j\le k+1}n^i\times n^j-\sum_{1\le p\le r}\sum_{1\le i<j\le k+1}n_p^in_p^j\\&=\sum_{1\le i<j\le k+1}(n_1^i+n_2^i+\cdots+n_r^i)(n_1^j+n_2^j+\cdots+n_r^j)-\sum_{1\le i<j\le k+1}\sum_{1\le p\le r}n_p^in_p^j\\&=\sum_{1\le i<j\le k+1}\sum_{1\le p<q\le r}n_p^in_q^j\\&=DIS_A^D.\end{aligned}$$

证毕.

由于 $DIS_{U,A}$,$DIS_{U,D}$ 和 $DIS_{U,A\cup D}$ 可以统一为 $\Big(\Big[\Big(\sum_{1\le i\le l}n_l\Big)^2-\sum_{1\le i\le l}(n_l)^2\Big]\Big/2\Big)$($n_l$ 为相应等价类中对象个数),而〈等价类,等价类中对象个数〉与〈key,$value$〉类似,故可以利用 MapReduce 并行计算等价类,从而可以计算可辨识对象对个数.

3.1.2　云计算环境下可辨识对象对个数的间接计算方法(NDIS)

一个可辨识的对象对是由决策属性值不同和条件属性组合值也不同的两个对象生成的.而一个不可辨识的对象对则是由条件属性组合值相同但决策属性值不同的两个对象产生的,这说明这些条件属性不能辨识这个对象对.于是,可以利用属性(集)的不可辨识性来间接计算可辨识的对象对个数.

定义 12.　在决策表 S 中,$a\in C$,属性 a 不能辨识的对象对为

$$\widetilde{DOP}_a=\{\langle x,y\rangle\mid f(x,a)=f(y,a),x\in D_i,y\in D_j\},$$

其中,$1\le i<j\le k+1$.

定义 13.　在决策表 S 中,$A\subseteq C$,属性集 A 不能辨识的对象对为

$$\widetilde{DOP}_A=\{\langle x,y\rangle\mid\forall a\in A,f(x,a)=f(y,a),x\in D_i,y\in D_j\},$$

其中,$1\le i<j\le k+1$.

定理 3.　在决策表 S 中,若 $A\subseteq C$,$\forall a\in A$,则有

$$\widetilde{DOP}_A=\bigcap_{a\in A}\widetilde{DOP}_a.$$

证明.　由定义 12 和定义 13 可以直接证得.

证毕.

定义 14.　在决策表 S 中,$A\subseteq C$,属性集 A 不能够辨识的对象对总数为 $\widetilde{DIS}_A^D=\sum_{1\le p\le r}\sum_{1\le i<j\le k+1}n_p^in_p^j$.

从定义 14 可知,由于 $\sum_{1\le p\le r}\sum_{1\le i<j\le k+1}n_p^in_p^j$ 能够统一为 $\Big[\Big(\sum_{1\le i\le l}n_l\Big)^2-\sum_{1\le i\le l}(n_l)^2\Big]\Big/2$,故它也能够利用 MapReduce 并行计算等价类,从而计算不可辨识的对象对个数.

定理 4.　在决策表 S 中,$A\subseteq C$,$c\in C-A$,则 $\widetilde{DIS}_{A\cup c}^D\le\widetilde{DIS}_A^D$.

证明.　由 A 导出的 U 上划分记为 $U/A=\{A_1,A_2,\cdots,A_r\}$,属性集合 $A\cup c$ 导出的 U 上划分 $U/\{A\cup c\}$ 是对 $A_1,A_2,\cdots,A_r$ 等价类的细化.任一等价类 $A_p(p=1,2,\cdots,r)$ 按决策属性 D 划分为 $k+1$ 个等价类 $A_p^1,A_p^2,\cdots,A_p^{k+1}$,其元素个数分别为 n_p^1,$n_p^2,\cdots,n_p^{k+1}$,则增加属性 c(有 m 个不同属性值)后

将 $A_p^1, A_p^2, \cdots, A_p^{k+1}$ 细化为 m 组等价类，则

$$A_{p,1}^1 \cup A_{p,2}^1 \cup \cdots \cup A_{p,m}^1 = A_p^1$$
$$A_{p,1}^2 \cup A_{p,2}^2 \cup \cdots \cup A_{p,m}^2 = A_p^2$$
$$\cdots$$
$$A_{p,1}^{k+1} \cup A_{p,2}^{k+1} \cup \cdots \cup A_{p,m}^{k+1} = A_p^{k+1}$$
$$n_{p,1}^1 + n_{p,2}^1 + \cdots + n_{p,m}^1 = n_p^1$$
$$n_{p,1}^2 + n_{p,2}^2 + \cdots + n_{p,m}^2 = n_p^2$$
$$\cdots$$
$$n_{p,1}^{k+1} + n_{p,2}^{k+1} + \cdots + n_{p,m}^{k+1} = n_p^{k+1}$$

对于等价类 A_p 来说，不能辨识的对象对数 $\widetilde{DIS}_A^D$ 为 $\sum_{1\le i<j\le k+1} n_p^i n_p^j$. 当增加属性 c 后，不能辨识的对象对数 $\widetilde{DIS}_{A\cup c}^D$ 为 $\sum_{1\le i<j\le k+1} n_{p,1}^i n_{p,1}^j + \sum_{1\le i<j\le k+1} n_{p,2}^i n_{p,2}^j + \cdots + \sum_{1\le i<j\le k+1} n_{p,m}^i n_{p,m}^j = \sum_{1\le i<j\le k+1} \sum_{1\le l\le m} n_{p,l}^i n_{p,l}^j$.

对于任意等价类 A_p 和决策属性值 i、j，都有 $n_{p,1}^i n_{p,1}^j + n_{p,2}^i n_{p,2}^j + \cdots + n_{p,m}^i n_{p,m}^j \le (n_{p,1}^i + n_{p,2}^i + \cdots + n_{p,m}^i)(n_{p,1}^j + n_{p,2}^j + \cdots + n_{p,m}^j) = n_p^i n_p^j$ 成立，故 $\widetilde{DIS}_{A\cup c}^D \le \widetilde{DIS}_A^D$.

性质 1. 在决策表 S 中，$P\subseteq Q\subseteq C$，则 $\widetilde{DIS}_Q^D \le \widetilde{DIS}_P^D$.

证明. 由定理 4，易证性质 1 成立.

从定理 4 和性质 1 可知，$\widetilde{DIS}_A^D$ 具有单调性，可用来进行属性约简. 下面探讨 DIS_A^D 和 $\widetilde{DIS}_A^D$ 与差别矩阵的关系. 假设“相容决策表 S”有 $k+1$ 个不同决策，按决策属性可生成非空差别矩阵元素个数为 $\sum_{1\le i<j\le k+1} n^i n^j$. 于是有下列定理成立.

定理 5. 在相容决策表 S 中，$A\subseteq C$，则 $DIS_A^D = \sum_{1\le i<j\le k+1} n^i n^j - \widetilde{DIS}_A^D$.

证明.

$$\begin{aligned}
&\sum_{1\le i<j\le k+1} n^i n^j - \widetilde{DIS}_A^D \\
&= \sum_{1\le i<j\le k+1} n^i n^j - \sum_{1\le p\le r} \sum_{1\le i<j\le k+1} n_p^i n_p^j \\
&= \sum_{1\le i<j\le k+1} (n_1^i + n_2^i + \cdots + n_r^i)(n_1^j + n_2^j + \cdots + n_r^j) - \\
&\quad \sum_{1\le i<j\le k+1} \sum_{1\le p\le r} n_p^i n_p^j \\
&= \sum_{1\le i<j\le k+1} \sum_{1\le p<q\le r} n_p^i n_q^j \\
&= DIS_A^D.
\end{aligned}$$

证毕.

说明：DOP_A 中一个对象对将产生差别矩阵中包含属性集 A 中属性的一个差别元素，而 $\widetilde{DOP}_A$ 中一个对象对将生成差别矩阵中不包含属性集 A 中属性的一个元素. 对于相容（一致）决策表，两者个数之和正好等于差别矩阵中的元素个数. DIS_A^D 直接计算方法只能适用于相容决策表，而间接计算方法可以适用于任何决策表. 尽管两种方法都能够计算相容决策表的知识约简，但是由于 DIS_A^D 需要计算属性 A、D 和 $A\cup D$ 导出的等价类，而 $\widetilde{DIS}_A^D$ 仅计算出 A 导出的等价类，故知识约简算法效率会相差很大.

3.2 云计算环境下知识约简算法

从定理 2 和定理 5 可以看出，属性集 A 能够辨识的对象对个数 DIS_A^D 或不可辨识的对象对个数 $\widetilde{DIS}_A^D$ 都需要通过计算等价类来获得，而不同等价类是可以并行计算的. 因此，可以利用 MapReduce 并行编程技术处理大规模数据.

在一个 MapReduce 编程框架中，用户着重研究算法中可并行化操作，编写 Map 函数和 Reduce 函数，即可实现大规模数据并行处理. 具体而言，Map 函数主要完成不同数据块中等价类计算，而 Reduce 函数主要统计同一个等价类个数或计算同一个等价类所不能辨识的对象对个数. 根据 DIS_A^D 不同的计算方法，给出两种云计算环境下的知识约简算法.

3.2.1 云计算环境下的知识约简算法之一（DP-DIS）

下面先给出直接计算方法中一个属性集是否是约简的判断准则.

定理 6. 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) $DIS_A^D = DIS_C^D$；

(2) $\forall a\in A, DIS_{A-\{a\}}^D < DIS_A^D$.

证明. 由定理 2 容易证得. 证毕.

DP-DIS 算法包含 Map 函数（算法 1）、Reduce 函数（算法 2）和主程序 DP-DIS（算法 3）3 个算法，分别叙述如下.

算法 1. Map (Object *key*, Text *value*)

输入：已选属性集 A，候选属性 $c\in C-A$，决策属性 D，一个对象 *value*

输出：〈等价类，出现次数〉

//$Ac\_EquivalenceClass$、$D\_EquivalenceClass$、$AcD\_EquivalenceClass$ 为属性集 $A\cup c$、D 和 $A\cup c\cup D$ 导出的等价类

1. $Ac\_EquivalenceClass$ = “c”; $D\_EquivalenceClass$ = “D”; $AcD\_EquivalenceClass$ = “c D”;
2. For each attribute a in $A\cup c$
   $Ac_EquivalenceClass = Ac_EquivalenceClass + f(value, a) + $ “ ”;
3. EmitIntermediate 〈$Ac\_EquivalenceClass$, 1〉;
4. $D\_EquivalenceClass = f(value, D)$;

5. EmitIntermediate ⟨$D\_EquivalenceClass$,1⟩;

6. For each attribute a in $A\cup c\cup D$

$AcD\_EquivalenceClass=AcD\_EquivalenceClass+f(value,a)+$" ";

7. EmitIntermediate ⟨$AcD\_EquivalenceClass$,1⟩.

算法 2. Reduce (Text $EquivalenceClass$, Iterable⟨Int⟩ $values$).

输入:等价类 $EquivalenceClass$, $values[]$

输出:⟨$EquivalenceClass$,出现次数⟩

1. $Total=0$;

2. for $i=1$ to $values.size()$

$Total=Total+values[i]$;

3. Emit ⟨$EquivalenceClass$, $Total$⟩.

算法 3. 主程序 DP-DIS.

输入:一个相容决策表 S

输出:一个约简 Red

1. $Red=\varnothing$;

2. while(DIS^D_{Red} 不等于 DIS^D_C)

For each attribute c in $C-Red$

启动一个 Job,调用算法 1 的 Map 和算法 2 的 Reduce 函数,计算 $DIS^D_{Red\cup\{c\}}$;

$c_l=\max\limits_{c\in C-Red}\{DIS^D_{Red\cup\{c\}}\}$(若这样的 c_l 不唯一,则任选其一);

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

3. For each attribute c in Red

启动一个 Job,调用算法 1 的 Map 函数和算法 2 的 Reduce 函数;

if $DIS^D_{Red-\{c\}}=DIS^D_{Red}$

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

4. 输出 Red.

算法 1 中 Map 函数计算各个数据块中等价类及出现次数,算法 2 将所有数据块中相同的等价类进行汇总,而算法 3 则分别计算属性集 $A\cup c$、D 和 $A\cup c\cup D$ 的辨识能力大小,并根据各个候选属性的相对辨识能力确定最佳候选属性,重复上述过程,直到计算出约简.

3.2.2 云计算环境下知识约简算法之二(DP-NDIS)

由于同一个等价类可能决策属性值都相同,因此不可辨识的对象对个数为 0.为了节省存储空间,降低网络传输时间,不输出这类信息.

定理 7. 对于一个等价类 A_p, $A_p\subseteq BND_A(D)$ 充分必要条件为 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p>0$.

证明. 如果 A_p 属于 $BND_A(D)$,则 A_p 至少有两个不同的决策值.不妨设决策值为 i 和 j,则 $n^i_p>0$, $n^j_p>0$.于是有 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p>0$.

反之,若 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p>0$,假设 A_p 不属于 $BND_A(D)$,则 A_p 中所有对象的决策值相同.不妨设决策值为 i,则 $n^i_p>0$,其余的 $n^j_p=0$.于是,有 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p=0$,这与 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p>0$ 相矛盾.因此,A_p 属于 $BND_A(D)$. 证毕.

性质 2. 对于一个等价类 A_p, $A_p\subseteq POS_A(D)$,则 $\sum\limits_{1\le i<j\le k+1}n^i_pn^j_p=0$.

证明. 根据定理 7,容易证得. 证毕.

定理 8. 在决策表 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) $\widetilde{DIS}^D_A=\widetilde{DIS}^D_C$;

(2) $\forall a\in A$, $\widetilde{DIS}^D_{A-\{a\}}<\widetilde{DIS}^D_A$.

证明. 由定理 4 容易证得. 证毕.

下面,具体探讨如何实现 Map 函数(算法 4)、Reduce 函数(算法 5)和主程序 DP-NDIS(算法 6).

算法 4. Map (Object key, Text $value$)

输入:已选属性集 A,候选属性 $c\in C-A$,决策属性 D,一个对象 $value$

输出:⟨$Ac\_EquivalenceClass$,⟨$f(value,D)$,1⟩⟩

1. $Ac\_EquivalenceClass=$"c";

2. For each attribute a in $A\cup c$

$Ac\_EquivalenceClass=Ac\_EquivalenceClass+f(value,a)+$" ";

3. EmitIntermediate ⟨$Ac\_EquivalenceClass$, ⟨$f(value,D)$,1⟩⟩.

算法 5. Reduce(Text key, Iterable⟨Text⟩ $values$).

输入:同一个等价类 key 及对应的决策值与出现次数的列表 $values$

输出:⟨$EquivalenceClass$, $IndisObjectPairSum$⟩

//$EquivalenceClass$ 是等价类,$IndisObjectPairSum$ 是该等价类所不能辨识的对象对个数

1. $IndisObjectPairSum=0$;

2. For $i=1$ to $values.size()$

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

3. $IndisObjectPairSum=\sum\limits_{1\le i<j\le k+1}n^in^j$;

4. $EquivalenceClass=key.toString().split(" ")[0]$;

5. if $IndisObjectPairSum$ 不为 0

Emit ⟨$EquivalenceClass$, $IndisObjectPairSum$⟩.

算法 6. 主程序 DP-NDIS.

输入:一个决策表 S

输出:一个约简 Red

1. $Red=\varnothing$;

2. While(DIS^D_{Red} 不等于 DIS^D_C)

For each attribute c in $C-Red$

启动一个 Job,调用算法 4 的 Map 函数和算法 5 的 Reduce 函数,计算 $\widetilde{DIS}^D_{Red\cup c}$;

$c_l=\min\limits_{c\in C-Red}\{\widetilde{DIS}^D_{Red\cup c}\}$(若这样的 c_l 不唯一,则任

选其一);

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

3. For each attribute c in Red

启动一个 Job,调用算法 4 的 Map 函数和算法 5 的 Reduce 函数,计算 $DIS^D_{Red-\{c\}}$;

if $DIS^D_{Red-\{c\}}=DIS^D_{Red}$

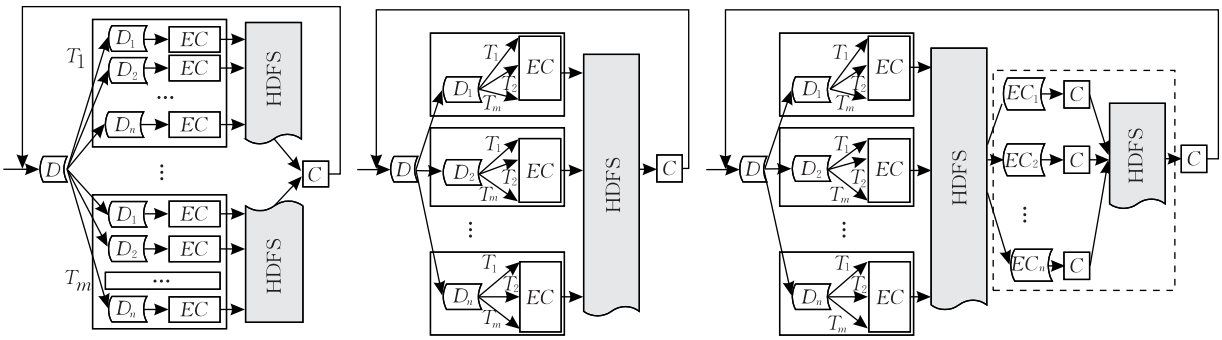
$Red=Red-\{c\}$;

4. 输出 Red.

算法 4 计算各个数据块中等价类及其不同决策值出现的次数,算法 5 统计同一个等价类中不同决策值出现的次数,并计算不可辨识的对象对个数,而算法 6 根据各个候选属性产生的不可辨识的对象对个数,确定最优候选属性,重复上述过程,直到计算出约简.

3.3 数据并行策略

传统的并行属性约简算法是假设将所有数据一次性装入内存中,这不适合处理大规模数据集.下面深入讨论知识约简算法中并行操作策略,如图 1 所示.图 1(a)先在每个任务中将大规模数据划分为多个数据分片,并行计算候选属性集导出的等价类,然后根据各个任务中所计算得到的不可辨识或可辨识对象对个数确定最佳候选属性.图 1(b)先将大规模数据划分为多个数据分片,然后对每个数据分片并行计算不同的候选属性集导出的等价类(任务并行),然后统计各个任务中不可辨识或可辨识对象对个数来确定最佳候选属性.图 1(c)是指当知识约简算法面对高维数据集时,任务并行方式将产生大规模的不可辨识或可辨识对象对,这时可以在图 1(b)基础上再以数据并行方式计算候选属性集的可辨识或不可辨识对象对总个数,最终确定最佳候选属性.



(a) data-parallelism (b) (data+task)-parallelism (c) (data+task+data)-parallelism

图 1 知识约简算法中的并行操作策略

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)说明本文提出的两种算法,其中表 1 中“?”为所有不相容对象的决策属性值.

表 1 一个“相容决策表”DT

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	?

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.下面分别阐述 DIS 算法和 NDIS 算法计算对象对个数过程.

(1) DP-DIS 算法计算候选属性 C_1 的相对可辨识的对象对过程

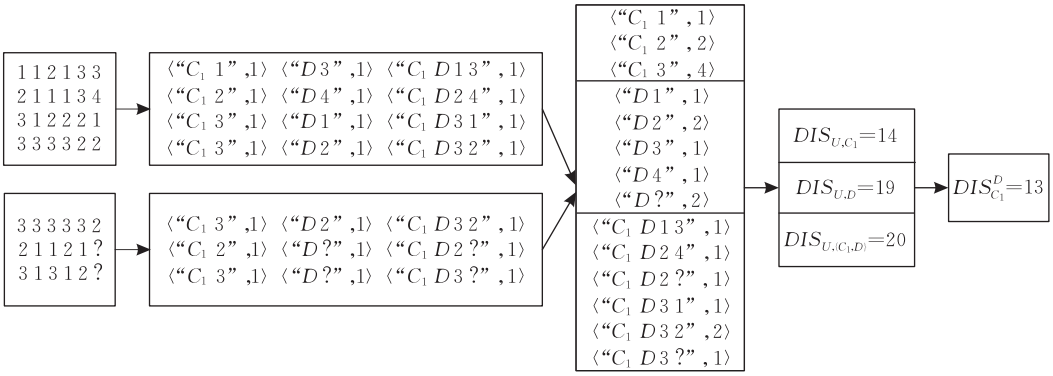
在 Map 阶段,对于第 1 个数据分片中第 1 个对象,分别生成属性 C_1 导出的等价类〈“C_1 1”,1〉、属性 D 导出的等价类〈“D 3”,1〉和属性集合$\{C_1,D\}$导出的等价类〈“C_1 D 1 3”,1〉3 个〈key,$value$〉对,其余对象计算过程类似.在 Reduce 阶段,对属性 C_1、属性 D 和属性集合$\{C_1,D\}$导出的等价类进行合并,如〈“C_1 2”,1〉和〈“C_1 2”,1〉合并为〈“C_1 2”,2〉;〈“C_1 3”,1〉、〈“C_1 3”,1〉、〈“C_1 3”,1〉和〈“C_1 3”,1〉合并为〈“C_1 3”,4〉;其余属性导出的等价类合并类似.最终,属性 C_1 的相对可辨识的对象对个数为 13(见图 2(a)).

(2) DP-NDIS 算法计算候选属性 C_1 的相对不可辨识的对象对过程

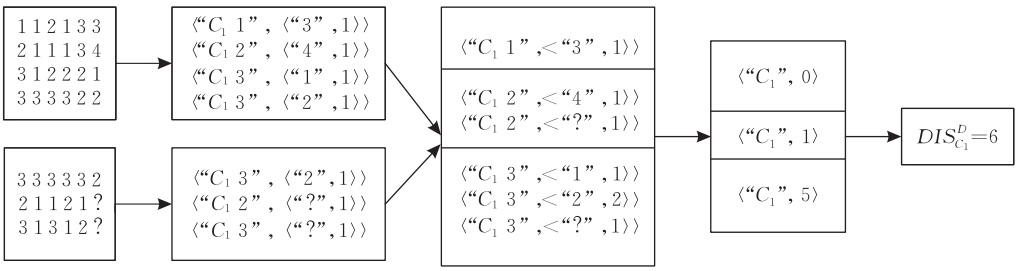
在 Map 阶段,对于第 1 个数据分片中第 1 个对象,生成属性 C_1 导出的等价类〈“C_1 1”,〈“3”,1〉〉,其余对象计算过程类似.在 Reduce 阶段,对属性 C_1 导出的等价类进行合并,如〈“C_1 3”,〈“2”,1〉〉和

〈“C_1 3”,〈“2”,1〉〉合并为〈“C_1 3”,〈“2”,2〉〉;其余等价类合并类似.最终,属性 C_1 的相对不可辨识的对象对个数为 6(见图 2(b)).

DP-DIS 和 DP-NDIS 两种算法整个计算过程如图 2 所示.



(a) DP-DIS算法计算属性C_1相对可辨识对象对个数



(b) DP-NDIS算法计算属性C_1相对不可辨识对象对个数

图 2　DP-DIS 和 DP-NDIS 两种算法计算候选属性 C_1 的重要性

4.2　实验结果

本节主要从运行时间、加速比(Speedup)和可扩展性(Scaleup) 3 个方面对所提出的云计算环境下知识约简算法的性能进行评价.将采用图 1(a)～(c)并行操作策略的算法分别标记为 DP-DIS/DP-NDIS、DTP-DIS/DTP-NDIS 和 DTDP-DIS/DTDP-NDIS.为了考察本文所提出的算法,选用 UCI 机器学习数据库(http://www.ics.uci.edu/～mlearn/MLRepository.html)中 DS1(tic-tac-toe-endgame, TicTac)和 DS2(mushroom)两个数据集用于测试算法正确性,用人工数据集 DS3、DS4、DS5 和 DS7 以及实际数据集 DS6(5000 倍 DS2)来测试性能. 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(http://lucene.apache.org/hadoop)和 Java 1.6.0_20 在 17 台普通计算机(Intel Pentium Dual-core 2.6GHz CPU, 2GB 内存)构建的云计算环境下进行实验,其中 1 台为主节点,16 台为从节点.

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	958	9	2
DS2	8124	22	2
DS3	10000000	30	10
DS4	20000000	30	10
DS5	40000000	30	10
DS6	40620000	22	2
DS7	100000	10000	10

(1)运行时间

首先,在两个小数据集 DS1(TicTac)和 DS2(mushroom)上比较了 DIS(传统方法)、DP-DIS、DP-NDIS、DTP-DIS 和 DTP-NDIS 5 种知识约简算法.从图 3 可以看出,小数据集不适宜使用 MapReduce 技术,而且使用数据和任务并行的知识约简算

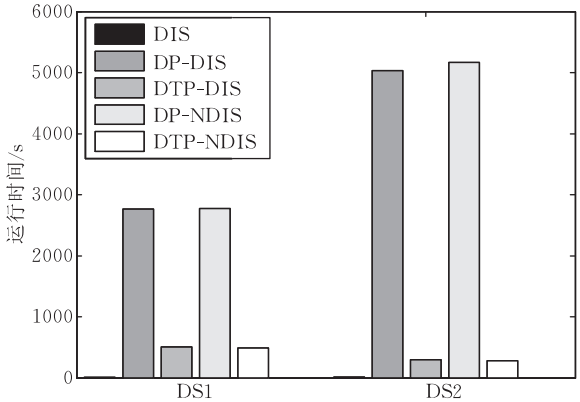


图 3　5 种知识约简算法运行时间比较

法比仅使用数据并行的算法运行时间更短.因此,下文着重讨论 DTP-DIS/DTP-NDIS 和 DTDP-DIS/DTDP-NDIS 两类基于数据和任务同时并行的知识约简算法.

其次,分别对 DS 3～6 四个数据集在 8 个节点上进行测试,运行时间如图 4、图 5 所示.从图 4 和图 5 可以看出,DTP-DIS 运行时间比 DTP-NDIS 更长.图 4 表明随着所选属性个数的增加,DTP-DIS 运行时间一直呈上升趋势,而 DTP-NDIS 运行时间呈现先上升后有所下降或较平稳上升.这是因为 DTP-DIS 生成的可辨识的对象对越来越多,故计算时间一直增加,而 DTP-NDIS 生成的不可辨识的对象对个数随着属性个数的增长先增加而后慢慢变少,故计算时间会先增加后减少,这一点可以从它们各自的计算公式上得到体现.

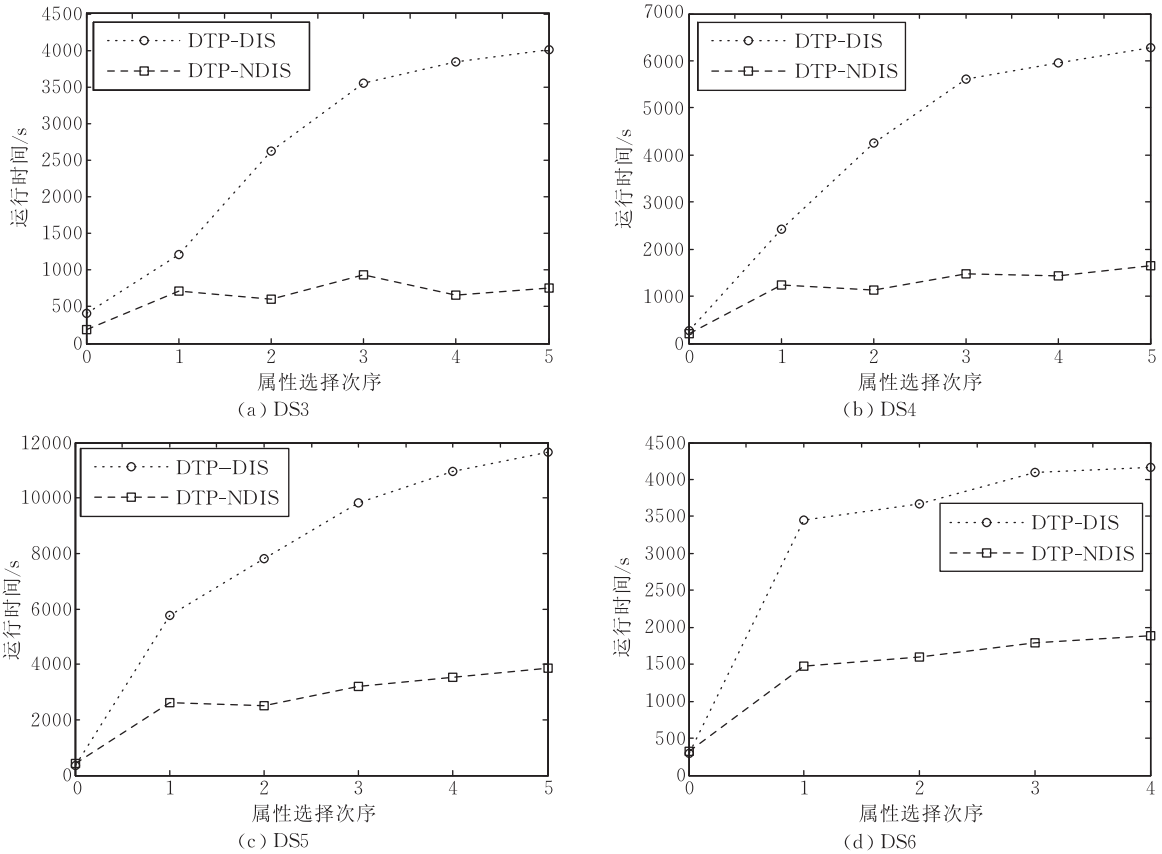


图 4 DTP-DIS 和 DTP-NDIS 在单次属性选择中运行时间比较

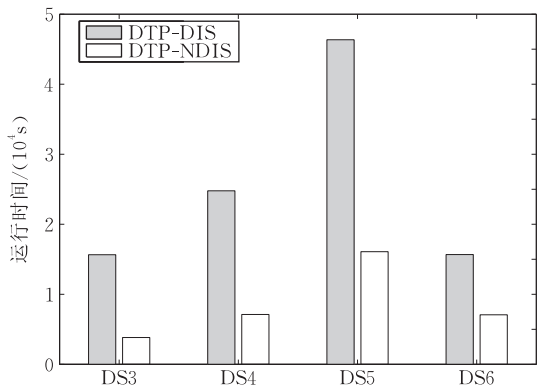
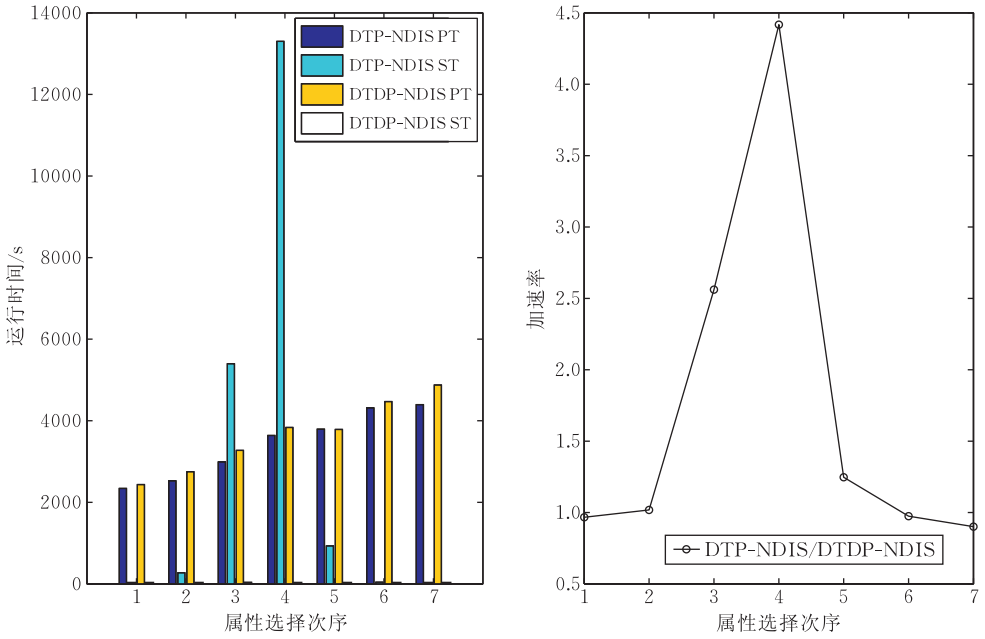


图 5 DTP-DIS 和 DTP-NDIS 在 DS3～6 上整个运行时间比较

由于 DTP-NDIS 算法比 DTP-DIS 算法运行时间更短,下面探讨 DTP-NDIS 算法.当 DTP-NDIS 遇到高维数据集时,其串行时间较长,这时可以考虑再次使用数据并行方式(图 1(c)).DTP-NDIS 算法和 DTDP-NDIS 算法在 DS7 的运行结果如图 6 所示,其中 PT 表示并行计算时间(含通信并行),ST 表示串行计算时间.从图 6(b)可以看出,在第 3、4 和 5 次选择属性时再次用数据并行方式,算法效率更高,其它情况基本相当或略低.至于何时再次使用数据并行方式值得进一步深入研究.

(2)加速比

加速比是指将数据集规模固定,不断增大计算机节点数时并行算法的性能.为了测定加速比,保持 DS3～7 的 5 个数据集大小不变,成倍增加计算机节点数至 16 台.一个理想的并行算法加速比是线性的,即当计算机节点数增加至 m 时,其加速比为 m.然而,由于存在计算机间通信开销、任务启动、任务



(a) 在DS7上运行时间　　(b) 在DS7上两种算法的加速率

图 6　DTP-NDIS 和 DTDP-NDIS 两种算法在 DS7 的运行结果

调度和故障处理等时间，其实际加速比低于理想的加速比. 图 7 显示了不同数据集的加速比大小. 从图 7 可以看出，云计算环境下知识约简算法具有较好的加速比. 图 7 中 DTP-NDIS 在高维数据集 DS7 上加速比较低，是因为产生了大量的不可辨识的对象对，在串行统计候选属性的对象对个数时运行时间过长，导致整个算法并行所占比例过低. 另外，较小数据集的加速比比较低，其主要原因是部分节点处于空闲状态.

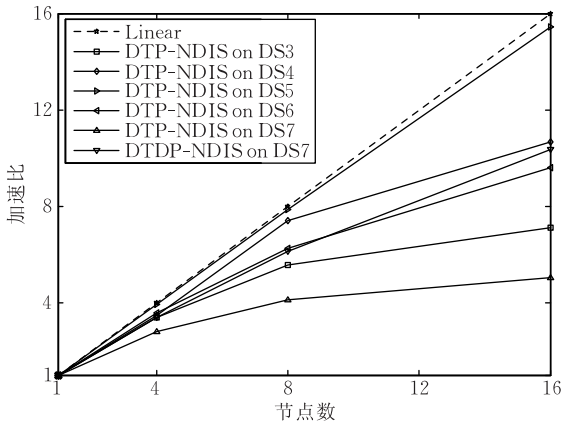


图 7　加速比

(3) 可扩展性

可扩展性是指按与计算机节点数成比例地增大数据集规模时并行算法的性能. 为了测定可扩展性，实验复制 1、2 和 4 倍的数据集 DS3、DS4，分别在 4、8 和 16 节点下运行. 图 8 显示了在不同节点上的可扩展性结果. 实验结果显示，DTP-NDIS 具有较好的可扩展性.

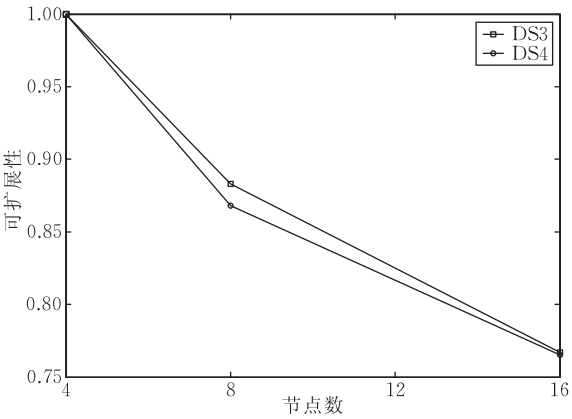


图 8　可扩展性

4.3　讨　论

传统的并行属性约简算法主要用来计算最小属性约简[16]，但仅仅对小数据集进行约简. 文献[17]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，重复这个过程，从而获取约简. 然而，对于不一致决策表，该方法并不能保证对各个子决策表计算正区域与对整个决策表计算正区域是等价的，因为各个子决策表之间计算正区域时并不交换信息. 同样，它也无法对较大的子决策表进行约简. 文献[21]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，再增加其它必要的候选属

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较大的条件属性集,删除冗余属性则变得十分困难.为了解决海量数据集的知识约简问题,本文利用 MapReduce 技术对大规模数据进行分片,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,计算候选属性的相对不可辨识对象对或可辨识对象对,选择最优单个候选属性,迭代此过程,直到获得约简.本文所提的算法先采用数据并行策略,再进行任务并行操作,这样大大地节省了 MapReduce 作业启动与调度时间,从而提高了云计算环境下知识约简算法效率.

5 结束语

传统的知识约简算法通过改进排序算法或利用较好的数据表示来快速计算等价类,将时间复杂度降为 $\max(O(|C||U|),O(|C|^2|U/C|))$,但只能在主存中处理小数据集.而目前已有的并行知识约简算法仅仅实现了约简任务的并行计算,同样不能处理大规模数据集.为了进行面向大规模数据集的知识约简,深入分析知识约简算法中可并行化操作,利用属性(集)的可辨识性和不可辨识性来设计适合处理海量数据的可辨识和不可辨识的对象对,提出了并行计算等价类方法和两类云计算环境下知识约简算法,讨论并实现了各种数据并行策略,并利用 Hadoop 在普通计算机的集群上进行实验.实验结果表明该知识约简算法具有较好的加速比,能够处理大规模数据集.

参 考 文 献

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

[2] Liu Shao-Hui, Shen Qiu-Jian, Wu Bin, Shi Zhong-Zhi, Hu Fei. Research on efficient algorithms for Rough set methods. Chinese Journal of Computers, 2003, 26(5): 524-529(in Chinese)
(刘少辉,盛秋戬,吴斌,史忠植,胡斐. Rough 集高效算法的研究. 计算机学报, 2003, 26(5): 524-529)

[3] Xu Zhang-Yan, Liu Zuo-Ping, Yang Bing-Ru, Song Wei. A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$. Chinese Journal of Computers, 2006, 29(3): 391-399(in Chinese)
(徐章艳,刘作鹏,杨炳儒,宋威. 一个复杂度为 $\max(O(|C||U|), O(|C|^2|U/C|))$ 的快速属性约简算法. 计算机学报, 2006, 29(3): 391-399)

[4] Liu Yong, Xiong Rong, Chu Jian. Quick attribute reduction algorithm with Hash. Chinese Journal of Computers, 2009, 32(8): 1493-1499(in Chinese)
(刘勇,熊蓉,褚健. Hash 快速属性约简算法. 计算机学报, 2009, 32(8): 1493-1499)

[5] Miao Duo-Qian, Hu Gui-Rong. A heuristic algorithm for reduction of knowledge. Journal of Computer Research & Development, 1999, 36(6): 681-684(in Chinese)
(苗夺谦,胡桂荣. 知识约简的一种启发式算法. 计算机研究与发展, 1999, 36(6): 681-684)

[6] Wang Guo-Yin, Yu Hong, Yang Da-Chun. Decision table reduction based on conditional information entropy. Chinese Journal of Computers, 2002, 25(7): 759-766(in Chinese)
(王国胤,于洪,杨大春. 基于条件信息熵的决策表约简. 计算机学报, 2002, 25(7): 759-766)

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

[8] Wang Jue, Wang Ju. Reduction algorithms based on discernibility matrix: The ordered attributes method. Journal of Computer Science and Technology, 2001, 16(6): 489-504

[9] Nguyen S H. Some efficient algorithms for Rough set methods//Proceedings of the Conference on Information Processing and Managements of Uncertainty in Knowledge Based Systems(IPMU1996). Granada, Spain, 1996: 1451-1456

[10] Hu Feng, Wang Guo-Yin. Quick reduction algorithm based on attribute order. Chinese Journal of Computers, 2007, 30(8): 1429-1435(in Chinese)
(胡峰,王国胤. 属性序下的快速属性约简算法. 计算机学报, 2007, 30(8): 1429-1435)

[11] Zhang Wen-Xiu, Mi Ju-Sheng, Wu Wei-Zhi. Approaches to knowledge reductions in inconsistent systems. International Journal of Intelligent System, 2003, 18(9): 989-1000

[12] Miao D Q, Zhao Y, Yao Y Y, Li H X, Xu F F. Relative reducts in consistent and inconsistent decision tables of the Pawlak Rough set model. Information Sciences, 2009, 179: 4140-4150

[13] Qian Y H, Liang J Y, Pedrycz W, Dang C Y. Positive approximation: An accelerator for attribute reduction in Rough set theory. Artificial Intelligence, 2010, 174: 597-618

[14] Qian J, Miao D Q, Zhang Z H, Li W. Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation. International Journal of Approximate Reasoning, 2011, 52(2): 212-230

[15] Yang Ming. An incremental updating algorithm for attribute reduction based on improved discernibility matrix. Chinese Journal of Computers, 2007, 30(5): 815-822(in Chinese)
(杨明. 一种基于改进差别矩阵的属性约简增量式更新算法. 计算机学报, 2007, 30(5): 815-822)

[16] Wang Li-Hong, Wu Geng-Feng. Attribute reduction based on parallel symbiotic evolution. Chinese Journal of Computers, 2003, 26(5): 630-635(in Chinese)
(王立宏，吴耿锋. 基于并行协同进化的属性约简. 计算机学报，2003，26(5)：630-635)

[17] Deng D Y, Yan D X, Wang J Y. Parallel reducts based on attribute significance//Yu J et al eds. The Fifth International Conference on Rough Set and Knowledge Technology. Beijing, 2010. Berlin Heidelberg: Springer-Verlag, 2010: 336-343

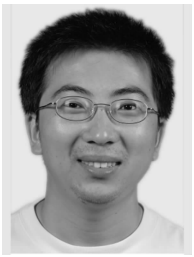
[18] Ghemawat S, Gobioff H, Leung S T. The Google file system. SIGOPS-Operating Systems Review, 2003, 37(5): 29-43

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

[20] Chu C T, Kim S, Lin Y A et al. MapReduce for machine learning on multicore//Schölkopf Bernhard, Platt John, Hofmann Thomas eds. Advances in Neural Information Processing Systems 19. MIT Press, 2006: 281-288

[21] Yang Y, Chen Z, Liang Z, Wang G. Attribute reduction for massive data based on Rough set theory and MapReduce//Yu J et al eds. The 5th International Conference on Rough Set and Knowledge Technology. Beijing, 2010. Berlin Heidelberg: Springer-Verlag, 2010: 672-678

[22] Xu Yan, Huai Jin-Peng, Wang Zhao-Qi. Reduction algorithm based on discernibility and its applications. Chinese Journal of Computers, 2003, 26(1): 97-103(in Chinese)
(徐燕，怀进鹏，王兆其. 基于区分能力大小的启发式约简算法及其应用. 计算机学报，2003，26(1)：97-103)



QIAN Jin, born in 1975, Ph. D. candidate. His research interests include rough set theory, data mining and cloud computing.

MIAO Duo-Qian, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include rough set theory, granular computing, Web intelligence and pattern recognition and cloud computing.

ZHANG Ze-Hua, born in 1981, Ph. D. candidate. His research interests include rough set theory and uncertainty reasoning.

Background

Rough set theory has been successfully applied in the fields such as data mining, pattern recognition, decision analysis, process control, image processing and medical diagnosis. Attribute reduction is one of the most important issues of rough set theory. Classical attribute reduction algorithms assume that all the data can be loaded into the main memory and compute equivalence classes using efficient sort algorithm or good data represention. When the number of the objects exceeds 1000,000, these algorithms can not acquire a reduct. On the other hand, the existing parallel attribute reduction algorithms only implement reduction methods in task parallel for small datasets, which are infeasible for handling large datasets. Massive data with high dimensions make attribute reduction a challenging task. In such case, developing an effective and fast search algorithm for attribute reduction is becoming increasingly important.

To address this issue, we first introduce the concepts and properties of discernibility and indiscernibility object pairs according to the discernibility and indiscernibility of the attribute(s). Then we discuss the relationship between discernibility matrix and discernibility/indiscernibility object pairs and parallel/serial computing parts. Moreover, we design an algorithm of computing equivalence classes for large-scale data in data parallel, and propose the corresponding knowledge reduction algorithms in cloud computing. Finally, we illustrate three parallelism strategies in detail and implement their algorithms. The experimental results demonstrate that knowledge reduction algorithms in cloud computing can scale well and efficiently process large-scale datasets on commodity computers.

The research is supported by the National Natural Science Foundation of China under Grant Nos. 60970061, 61075056, 61103067, the Fundamental Research Funds for the Central Universities, and the Natural Science Research Fund of Higher Education of Jiangsu Province under grant No. 09KJD520004. Our research work can enhance the performance of the attribute reduction algorithm for massive data and promote the applications of Rough set theory.