Journal of Computer Applications
计算机应用,2015,35(4):1143 - 1147
ISSN 1001-9081
CODEN JYIIDU
2015-04-10
http://www.joca.cn

文章编号:1001-9081(2015)04-1143-05
doi:10.11772/j.issn.1001-9081.2015.04.1143

基于 Hadoop 的海量嘈杂数据决策树算法的实现

刘亚秋[1,2],李海涛[1,2],景维鹏[1,2*]

(1.东北林业大学 信息与计算机工程学院,哈尔滨 150040;
2.黑龙江省林业生态大数据存储与高性能(云)计算工程技术研究中心,哈尔滨 150040)

(*通信作者电子邮箱 nefujwp@gmail.com)

摘　要:针对当前决策树算法较少考虑训练集的嘈杂程度对模型的影响,以及传统驻留内存算法处理海量数据困难的问题,提出一种基于 Hadoop 平台的不确定概率 C4.5 算法——IP-C4.5 算法。在训练模型时,IP-C4.5 算法认为用于建树的训练集是不可靠的,通过用基于不确定概率的信息增益率作为分裂属性选择标准,减小了训练集的嘈杂性对模型的影响。在 Hadoop 平台下,通过将 IP-C4.5 算法以文件分裂的方式进行 MapReduce 化程序设计,增强了处理海量数据的能力。与 C4.5 和完全信条树(CCDT)算法的对比实验结果表明,在训练集数据是嘈杂的情况下,IP-C4.5 算法的准确率相对更高,尤其当数据嘈杂度大于 10% 时,表现更加优秀;并且基于 Hadoop 的并行化的 IP-C4.5 算法具有处理海量数据的能力。

关键词:Hadoop;C4.5;不确定概率;嘈杂数据;并行化

中图分类号:TP181　　**文献标志码**:A

Implementation of decision tree algorithm dealing with massive noisy data based on Hadoop

LIU Yaqiu[1,2], LI Haitao[1,2], JING Weipeng[1,2*]

(1. *College of Information and Computer Engineering, Northeast Forestry University, Harbin Heilongjiang* 150040, *China*;
2. *Heilongjiang Province Engineering Technology Research Center for Forestry Ecological Big Data Storage and High Performance Computing (Cloud Computing), Harbin Heilongjiang* 150040, *China*)

Abstract: Concerning that current decision tree algorithms seldom consider the influence of the level of noise in the training set on the model, and traditional algorithms of resident memory have difficulty in processing massive data, an Imprecise Probability C4.5 algorithm named IP-C4.5 was proposed based on Hadoop. When training model, IP-C4.5 algorithm considered that the training set used to design decision trees is not reliable, and used imprecise probability information gain rate as selecting split criterion to reduce the influence of the noisy data on the model. To enhance the ability of dealing with massive data, IP-C4.5 was implemented on Hadoop by MapReduce programming based on file split. The experimental results show that when the training set is noisy, the accuracy of IP-C4.5 algorithm is higher than that of C4.5 and Complete CDT (CCDT), especially when the data noise degree is more than 10%, it has outstanding performance; and IP-C4.5 algorithm with parallelization based on Hadoop has the ability of dealing with massive data.

Key words: Hadoop; C4.5; imprecise probability; noisy data; parallelization

0　引言

随着数据的爆炸式增长[1],数据挖掘技术被广泛应用在各个领域。在数据挖掘分类领域中,C4.5 算法[2]是 Quinlan 在 1993 年对 ID3 算法[3]的改进算法,其模型的构建忽视了用于建树的训练集的不可靠性所带来的误差。针对训练集的不可靠性,Walley 在 1996 年提出一种正式的不确定概率理论,称作不精确的狄利克雷模型(Imprecise Dirichlet Model, IDM)[4],其根据数据集来估计每个值的概率区间。基于这种不确定概率理论,许多学者研究并提出了分类树的改进算法:如 Abellan 等在 2003 年第一次提出在信条集上用不确定概率和不确定测量作为原始划分标准来建立信条决策树(Credal Decision Tree, CDT)[5],但存在类似于 ID3 的不足之处。文献[6]针对于文献[5]中不能处理连续数据以及缺失值的问题,提出了封装几个属性选择度量的方法来建立决策树,不足之处在于参数较多,实现复杂。关于此方面,在最近几年的研究中,文献[7]采用一种更加精简的方法,通过封装基于 IDM 理论的信条决策树的方法代替复杂封装决策树,经过实验偏方差误差分解分析,证明该方法以更少的参数和更为简单的实现取得了更为优秀的表现。在此基础上,文献[8]提出了一种完全信条树(Complete CDT, CCDT)算法,该算法改进了文献[7]中的属性选择度量,以完全不精确信息增益(Complete Imprecise Probability Information Gain, CIIG)代替文献[7]中的不精确信息增益(Imprecise Probability Information

收稿日期:2014-11-15;**修回日期**:2014-12-23。

基金项目:国家自然科学基金资助项目(31370565);哈尔滨市科技创新人才研究专项资金资助项目(2013RFXXJ089)。

作者简介:刘亚秋(1971 -),男,辽宁法库人,教授,博士,主要研究方向:高性能计算、嵌入式计算;　李海涛(1989 -),男,辽宁普兰店人,硕士研究生,主要研究方向:并行分布式计算、高性能计算;　景维鹏(1979 -),男,黑龙江鹤岗人,副教授,博士,主要研究方向:分布式计算、容错计算。

Gain,IIG),实验结果表明该方法可以生成更小的树,具有更高的准确率。无论是IIG,还是CIIG作为属性选择度量,虽然都提高了准确率,但是其属性选择度量都是基于信息增益的,即香农熵,所以存在多值偏向问题,并且需要数据预处理以及没有剪枝处理。关于数据集是嘈杂的实际应用方面,在最新的研究中,文献[9]将一种模糊ID3算法应用到绩效评估中,减小了训练集的嘈杂性所带来的误差,得到了更有效的分类结果。文献[10]考虑到大多数现有的模糊决策树没有系统地考虑模糊集合的隶属度的非线性特性的影响,提出一种基于广义Hartley度量模型和计算方法的广义模糊划分熵模糊ID3(Gntropy-based Fuzzy ID3, GFID3)算法,实验结果表明该算法具有数据结构和可操作性好、计算精度高的特点。以上两种方法虽然可以处理不确定环境的情况,但是实现较为复杂,并且适用情况受限。

云计算是在分布式计算、网格计算、虚拟化等技术基础上发展起来的一种基于互联网的商业计算服务模式[11],其中最能代表"云计算"的就是开源平台Hadoop,其框架最核心的设计就是MapReduce和Hadoop分布式文件系统(Hadoop Distributed File System, HDFS)。HDFS是Google的GFS(Google File System)[12]的一个开源实现,为分布式计算存储提供了底层支持。而MapReduce编程框架,也是Google的MapReduce[13]的开源实现,简单说来就是在映射(Map)阶段,将数据进行分割,传给合并(Reduce)输出。数据挖掘与Hadoop结合一直是研究的热点之一,在最新的研究中,文献[14]分别实现了基于粗糙集方法的几个具有代表性的MapReduce运行系统:Hadoop、Phoenix和Twister,并通过实验分析了各自的优缺点。文献[15]将C4.5算法进行MapReduce化以解决数据无法一次装入内存的情况,又通过设计特殊的Hashtable来减少读取外存I/O开销。文献[15]方法虽然提高了时间效率和具有伸缩性,但是并没有对算法分类精确度进行改进,只考虑了并行化程序设计;并且当数据量大时,Hashtable会占用大量内存。文献[16]提出了一种极端学习树来减少不确定性的启发模式,在该模型中,信息熵和模糊性作为属性选择度量,并进行高度并行学习算法设计,实验证明该算法可以降低时间复杂度。但是这种方法没有借助任何编程框架,如Hadoop,不易实现并行化设计。

综上所述,本文提出了不确定概率C4.5(Imprecise Probability C4.5, IP-C4.5)算法,解决了经典C4.5算法不适合处理嘈杂数据的问题;同时,以基于文件分裂的方式进行MapReduce并行化程序设计,以解决处理海量数据的问题,从而实现处理海量嘈杂数据的目的。

1 IP-C4.5建模

IP-C4.5算法以Walley在1996年提出的不精确的狄利克雷模型[4]为理论基础,与经典C4.5的主要的不同之处便是属性度量的选取标准。经典的C4.5的属性分裂选取标准为信息增益率(Information Gain Ratio,IGR),而本文的改进算法用基于不确定概率的信息增益率(Imprecise Probability IGR, IP-IGR)来代替原有的,这样就克服了原有C4.5算法认为训练集是可靠的这一弊端,从而更加适合处理嘈杂的数据集。

1.1 经典C4.5的IGR模型

假设按照某属性A划分$\boldsymbol{D}$中元组,其中属性A具有V个不同值$\{a_1,a_2,\cdots,a_v\}$。如果A是离散的,则这些值直接对应于A上测试的V个输出。可以用属性A将$\boldsymbol{D}$划分V个分区$\{\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_v\}$,其中$\boldsymbol{D}_j$包含$\boldsymbol{D}$中的元组,它们的$A$值为$a_j$。则ID3算法所用的属性选择度量标准信息增益(Information Gain, IG)定义为原来的信息需求(仅基于类比例)与新的信息需求(对A划分之后)之间的差,即:

$$IG(A) = Info(\boldsymbol{D}) - Info_A(\boldsymbol{D}) \tag{1}$$

其中:$IG(A)$为属性A的信息增益,$Info(\boldsymbol{D})$为基于类比例的信息需求,$Info_A(\boldsymbol{D})$为对A划分之后新的信息需求。

设节点H代表分区$\boldsymbol{D}$的元组,选择具有最高信息增益的属性作为节点H的分裂属性。对$\boldsymbol{D}$中的元组分类所需要的期望信息如下:

$$Info(\boldsymbol{D}) = -\sum_{i=1}^{m} p_i \operatorname{lb}(p_i) \tag{2}$$

其中:p_i是$\boldsymbol{D}$中任意元组属于类C_i的非零概率,并用$|C_{i,\boldsymbol{D}}|/|\boldsymbol{D}|$估计;$|C_{i,\boldsymbol{D}}|$为类$C_i$在分区$\boldsymbol{D}$中所占的个数;$|\boldsymbol{D}|$为分区$\boldsymbol{D}$中总的元组数。

理想情况下,希望该划分产生元组的准确分类,$Info_A(\boldsymbol{D})$为对A划分之后所需信息的量由如式(3)度量:

$$Info_A(\boldsymbol{D}) = \sum_{j=1}^{v} \frac{|\boldsymbol{D}_j|}{|\boldsymbol{D}|} \times Info(\boldsymbol{D}_j) \tag{3}$$

其中:项$|\boldsymbol{D}_j|/|\boldsymbol{D}|$充当第$j$个分区的权重,$|\boldsymbol{D}_j|$为属于第$j$个分区的元组数,$Info(\boldsymbol{D}_j)$为第$j$个分区的信息期望。

使用信息增益作为属性选择标准的ID3算法具有多值偏向的弊端,所以C4.5[2]使用信息增益率代替ID3算法中的信息熵作为分类属性选择标准,则信息增益率(IGR)的定义为:

$$IGR(A) = \frac{IG(A)}{SplitInfo_A(\boldsymbol{D})} \tag{4}$$

其中$SplitInfo_A(\boldsymbol{D})$为使用分裂信息(Split information)值将信息增益规范化,定义如下:

$$SplitInfo_A(\boldsymbol{D}) = -\sum_{j=1}^{v} \frac{|\boldsymbol{D}_j|}{|\boldsymbol{D}|} \times \operatorname{lb}\left(\frac{|\boldsymbol{D}_j|}{|\boldsymbol{D}|}\right)$$

1.2 改进的属性分裂标准IP-IGR模型

经典的C4.5算法选取了信息增益率IGR作为属性选择标准,解决了ID3算法的多值偏向问题。但是信息增益率作为属性选择标准认为用于建模的训练集是可靠的,不可避免地接受了由于训练集的不可靠性所带来的误差。而本文提出的IP-C4.5可以减少这种误差,从而在处理嘈杂数据时具有更高的准确性。

基于不确定概率理论[4]的数学建模如下:在训练集上,假设属性A的变量W值属于$\{W_1,W_2,\cdots,W_k\}$,则$P_{W_j}(j=1,2,\cdots,k)$为W在训练集上的概率分布,由Walley在1996年提出的不精确的狄利克雷模型(IDM)[4]估计变量W_j的概率分布P_{W_j}值的区间为:

$$P_{W_j} \in \left[\frac{n_{W_j}}{T+s}, \frac{n_{W_j}+s}{T+s}\right]; j = 1,2,\cdots,k$$

其中:n_{W_j}是$W = W_j$频数;T为变量W的样本大小;s是一个给定的参数,由文献[4]可知,不依赖所选样本,表示不变性原理,其值越大,表示推理越谨慎,作者没有给出具体值,但是建议为1或2,在本文实验部分s取值为1。由此可知,关于变量W的概率分布$K(W)$为:

$$K(W) = \left\{p \mid p(W_j) \in \left[\frac{n_{W_j}}{T+s}, \frac{n_{W_j}+s}{T+s}\right], j = 1,2,\cdots,k\right\} \tag{5}$$

对于训练集 $\boldsymbol{D}$,假设有 m 个不同的类别 $C_i(i=1,2,\cdots,m)$,$A_i(i=1,2,\cdots,m)$ 为训练集属性,根据在概率分布 $K(W)$ 上的 p 的值,最大化表达式 $\sum_i p(A=a_i)Info(C\mid A=a_i)$,由式(2)便可以得到基于不确定概率的信息熵 $Info^*(K(W))$,定义为:

$$Info^*(K(W)) = \max\{Info(p)\mid p\in K(W)\} \tag{6}$$

其中:max 表示最大化;$Info(p)$ 为关于概率分布 $K(W)$ 不确定概率的信息熵,由式(2)可得。

对属性 A 进行关于变量 W 划分之后,新的信息需求 $Info_A^*(K(W))$ 所需信息的量由式(7)度量:

$$Info_A^*(K(W)) = \max\{Info(p_A)\mid p_A\in K_A(W)\} \tag{7}$$

其中 $Info(p_A)$ 为关于概率分布 $K_A(W)$ 不确定概率的信息熵。$K_A(Z)$ 的定义如下:

$$K_A(W)=\left\{p_A\mid p_A(W_j)\in\left[\frac{n_{W_j}}{T+s},\frac{n_{W_j}+s/l}{T+s}\right],j=1,2,\cdots,k\right\}$$

其中 l 是属性 A 的元组数量。

由式(6)~(7)可得,对于改进的属性选择度量 IP-IGR,本文作如下定义:

$$IP\text{-}IGR(A)=\frac{IP\text{-}IG(A)}{SplitInfo_A(\boldsymbol{D})} \tag{8}$$

其中 $IP\text{-}IG(A)$ 表示基于不确定概率的信息增益,定义为:

$$IP\text{-}IG(A)=Info^*(\boldsymbol{D})-Info_A^*(\boldsymbol{D})$$

2　IP-C4.5 算法实现

由于改进算法 IP-C4.5 是基于经典 C4.5 算法的,所以除了改进的属性选择度量外,其他部分和经典 C4.5 算法大体一致。

2.1　经典 C4.5 算法描述

C4.5 使用信息增益率作为分裂属性的选取标准,避免了信息增益所带来的多值偏向的问题。C4.5 的停止条件认为如果训练集为空或者达到设置的最小叶子实例数时便停止。此外,对于连续属性、缺失值,C4.5 也分别有自己的处理方法。树剪枝是重要的一步,为解决过度拟合数据问题,C4.5 使用悲观剪枝法,类似于代价复杂度方法,使用错误率来评估,以此对子树是否剪枝作出决定,通过加上一个惩罚项来调节从训练集得到的错误率以抵消出现的偏倚。一个 C4.5 决策树的建立过程可以简单地概括如下:

1)属性度量标准用于选择一个属性作为分裂节点;

2)决策树分支停止标准;

3)分配一个类标签或概率分布叶子节点的方法;

4)对树进行剪枝(postpruning)简化树结构。

2.2　IP-C4.5 算法描述

结合以上,IP-C4.5 算法主要思想描述如下:不精确概率的信息增益率(IP-IGR)作为每个分支节点的属性选择标准;当训练集 $\boldsymbol{D}$ 为空,即 $\boldsymbol{D}=\varnothing$,或者叶子中达到最小实例数或者比例时停止;连续值的处理以及树剪枝方法与经典 C4.5 类似。

具体算法描述如下:

Function IP-C4.5(A:属性集合,$\boldsymbol{D}$:训练集)。

Begin:

1)　*DecisionTree* = { }

2)　If $\boldsymbol{D}=\varnothing$ OR Conforming to stopping criteria standard then Exit;

3)　For all attribute A do
　　According to the model of IP-IGR: compute information-theoretic criteria if we split on A

4)　A_{best} = Best attribute selection criteria ccording to above computed result;

5)　*DecisionTree* = Create a decision node that A_{best} in the rootnode;

6)　$\boldsymbol{D}_v$ = Induced sub-datasets from $\boldsymbol{D}$ based on A_{best};

7)　For all $\boldsymbol{D}_v$ do
　　$DecisionTree_v$ = Function IP-C4.5($\boldsymbol{D}_v$)
　　Attach $DecisionTree_v$ to the corresponding branch of *DecisionTree*

8)　Return DecisionTree

End;

3　基于 Hadoop 的 IP-C4.5 算法实现

Hadoop 中的 MapReduce 编程框架对 Google 的 MapReduce 编程框架[13]的一个开源实现,简单说来就是映射(Map)和规约(Reduce)的过程,其处理大数据集的过程如图 1所示。

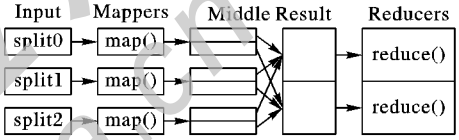

图 1　MapReduce 处理数据集过程

结合 MapReduce 编程框架思想和输入训练集的特点,本文提出了基于文件分裂的并行化程序设计方式。该方式通过两个 MapReduce 来实现,描述如下。

第一个 MapReduce:

Map1 阶段:读取数据集 $\boldsymbol{D}$,使用计数器 Counter 计算总记录数,同时将所有记录均改写成上述〈Key,Value〉形式。其中:Key 表示属性名;Value 表示属性值 + 类别 + ID 行号,并作为 Map1 的输出,同时计算出数据集 $\boldsymbol{D}$ 的不确定的信息熵。

Reduce1 阶段:MapReduce 的中间阶段会将相同 Key 的键值对放到一个 Reduce 中处理。对从 Map1 中得到的〈Key,Value〉,读取 Value 值,对不同属性值的类别用 TreeMap 分别计数,从而得到每个属性的不确定的信息熵,进而计算出每个属性的不确定的信息增益率,而信息增益率最大的即为最佳分裂属性。同时,在本阶段的开始还要判断终止条件,当从 Map1 中得到的〈Key,Value〉对中,相同 Key 的类别相同的结果占到一定比例(假设 80%)后,那么这一部分的分类就结束了或者说当不确定的信息熵计算为 0 时就结束。

第二个 MapReduce:

Map2 阶段:通过 MapReduce 之间的参数传递,通过 Conf 将第一个 MapReduce 中最佳分裂属性传递过来,将文件中的记录按照最佳分裂属性的属性值的不同取值写成〈Key,Value〉形式。

Reduce2 阶段:根据 Map2 阶段的〈Key,Value〉,通过改写 MulMultipleOutputFormat 多文件输出类,根据最佳分裂属性将源文件分裂,文件名以最佳分裂属性的不同属性值命名,文件内容为在该指定属性值下的其他属性值的记录。这样,就将文件分裂为不同的子文件,同时将这些文件放到一个队列中,以便递归地分别处理。

算法流程如图 2 所示。

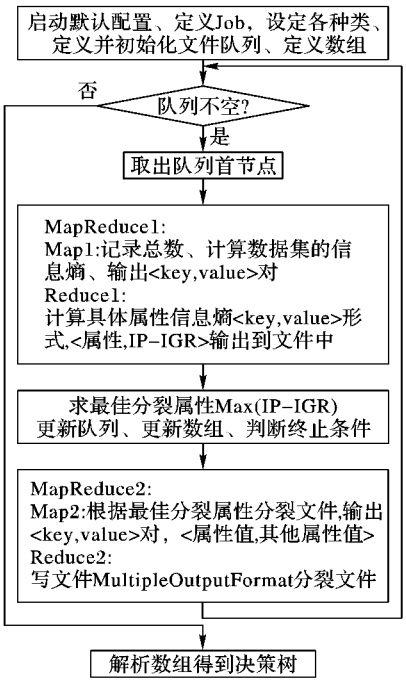

图 2 基于 MapReduce 的 IP-C4.5 算法流程

4 实验与分析

4.1 实验环境

为了检验 IP-C4.5 算法的性能，分别从串行和并行两个方面来说明。

1)串行方面。Windows 7 旗舰版(32 位)、Weka3.7.10 版本、eclipse-jee-indigo-SR2-win32 版本。

2)并行方面。4 台曙光服务器和一台普通 PC。曙光 I450-G10 配置为：塔式服务器，一个 InterXeon E5-2407 四核 2.2 GHz 处理器，8 GB 内存，硬盘 300 GB。PC 配置为：HP Compaq dx 2308、Intel Pentium E216 处理器、主频 1.8 GHz、1 GB 内存、160 GB 硬盘；曙光服务器作为虚拟机的物理载体，搭配上 XenServer-6.2，实现虚拟化，PC 机独立出来作为集群的 Master 节点(1 个)，管理 Slaves 节点(16 个)；每个节点安装 Centos6.4_final 作为系统平台，其内核为 2.6.32 版，在该平台上安装 JDK(jdk-6u31-linux-i586.bin)作为 Hadoop 的底层运行架构，在 JDK 上安装 Hadoop(hadoop-1.0.0.tar.gz)来构建 Hadoop 集群。

4.2 实验结果

串行实验：在相同的测试条件和环境下，利用 Weka[17] 中的 J48 源码作为参考，分别用 eclipse 实现了 IP-C4.5 算法和 CCDT 算法[8] 并重新编译进 Weka 工具中，进行经典 C4.5、CCDT 和 IP-C4.5 三个算法关于处理嘈杂数据能力的对比实验；并行实验：为了验证改进算法的处理嘈杂数据的能力，分别实现经典 C4.5 算法和文献[15]的 C4.5(记为 BC4.5)、IP-C4.5 算法，通过对比实验，验证本文改进算法处理嘈杂数据的能力、时间优越性、加速比、可扩展性等，所用数据集均来自 UCI 国际机器学习数据库[18]。

实验 1 基于 Weka 处理嘈杂数据的实验。

实验数据采用来自 UCI[18] 的 Iris、Breast-cancer、Adult、Spambase 这四个各具特色的数据集，用 Weka 工具中的 Filter-> AddNoise 功能分别向四个数据集中依次添加 0、5%、10%、15%、20%、25%、30% 的噪声，采用 10-fold cross-validation 来测试算法准确性。在相同的实验条件下，对经典 C4.5、CCDT 和 IP-C4.5 算法做关于正确率的实验。

正确率定义如下：

$$SuccessRate = N_{correct}/N_{all} \times 100\%$$

其中：$N_{correct}$ 为被正确分类的元组数，N_{all} 为总的元组数。

在串行条件下，由图 3 可知，经典的 C4.5 算法随着数据嘈杂度的增加，正确率明显低于其他两种算法，并且呈现忽高忽低的不稳定状态，而本文改进的算法要好得多，不但有更高的正确率，且表现稳定；图 4 表明在此数据集上所有算法预测的正确率均不高，分析原因为该数据集预测类别的相关性不强所致，但是随着嘈杂度的增加，本文算法仍然表现出优势。

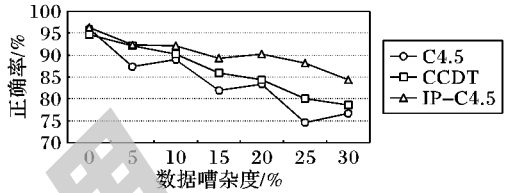

图 3 串行 Iris Set 嘈杂度实验

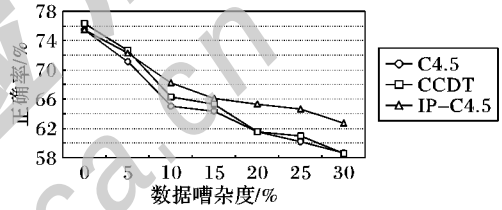

图 4 串行 Breast-cancer Set 嘈杂度实验

实验 2 基于 Hadoop 处理嘈杂数据的实验。

基于 Hadoop 平台的并行实验：实验数据集为 Adult 和 Spambase，在相同的实验环境下，分别对经典 C4.5(记为 mrC4.5)、IP-C4.5 以及并行化的 IP-C4.5(记为 mrIP-C4.5)做正确率实验，正确率的定义参考实验 1。

从图 5~6 可看出：对处理无噪声的数据，三个算法表现基本一致，但是随着噪声数据的增加，改进的 IP-C4.5 算法明显优于经典的 C4.5，并且 mrIP-C4.5 和 IP-C4.5 的曲线基本一致，也就是将改进的算法进行 MapReduce 并行化设计之后，并没有降低算法的预测正确率，从而说明，基于文件分裂的 IP-C4.5 算法具有可行性。

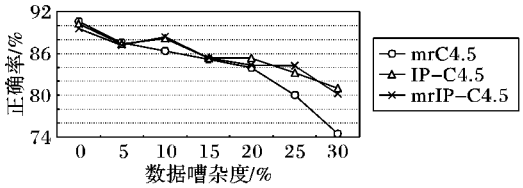

图 5 并行 Adult Set 嘈杂度实验

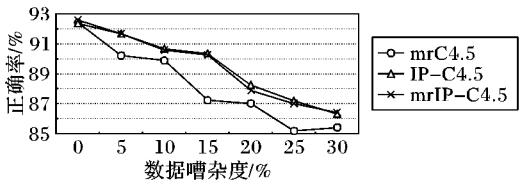

图 6 并行 Spambase Set 嘈杂度实验

实验 3 基于 MR 改进算法处理海量数据时间优越性的实验。

实验数据来自 UCI[18] 的 Poker Hand 数据集，该数据集具有 11 个属性，1 025 010 个元组，大小为 23 MB，采用数据复制

的手段,将数据集扩大到 200 MB、400 MB、600 MB、800 MB、1 000 MB。在相同实验条件下(DataNode 为 16 个),验证串行 C4.5、并行 BC4.5[15] 和并行 IP-C4.5 算法在处理大数据集上的时间优越性。

从图 7 可知,串行算法在处理大数据集时在时间上过高,而两个并行算法用时较少,尤其是本文基于文件分裂的 MapReduce 化的 IP-C4.5 算法,在时间上更是小于并行的 BC4.5,并且随着数据量的增大,用时增加幅度很小,这是因为基于文件分裂使得建树过程中每个分支得到单独处理,从而减少了时间。

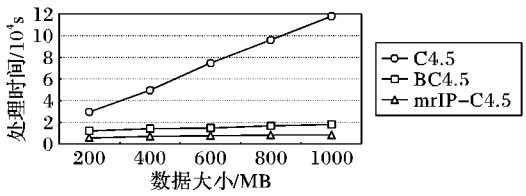

图 7　处理大数据的时间实验

实验 4　并行 IP-C4.5 的加速比及可扩展性实验。

加速比的定义如下:

$$S = T_s/T_n$$

其中:T_s 是单个节点的运行时间,T_n 是 n 个节点的运行时间。

效率定义为:

$$E = S/N$$

其中:S 为加速比,N 是节点数。

选用 BC4.5[15] 做对比实验,实验数据来自 UCI[18] 的 Poker Hand 数据集。

从图 8 可知,并行化的 IP-C4.5 算法具有更好的加速比,也就是说并行化的 IP-C4.5 随着节点数的增加,在减少时间上获得了更好的优势,从而说明具有高效性;从图 9 可知,在较大的数据集上,IP-C4.5 算法的曲线较为平滑,不会因为增加节点而使性能降低得过快,也就是说具有较好的可扩展性。

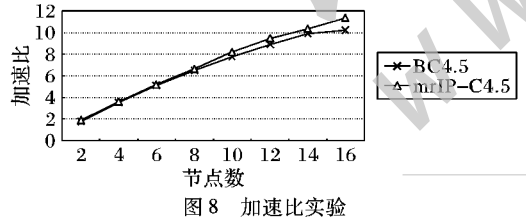

图 8　加速比实验

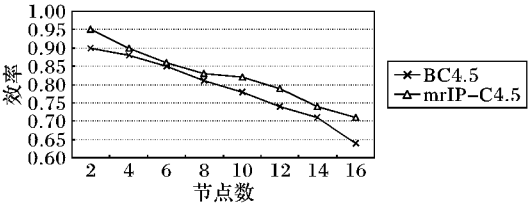

图 9　可扩展性实验

5　结语

基于不确定概率理论,本文提出一种 Hadoop 平台下改进的 C4.5 算法。实验结果表明,改进的算法具有处理嘈杂数据的能力;并且,从时间复杂度、加速比以及可扩展性方面验证了该算法具有处理海量数据的能力。以后的工作中将进一步考虑运用本文算法针对于某些领域的数据(比如林业数据),进行真正的数据挖掘研究。

参考文献:

[1] GANTZ J, REINSEL D. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east — United States [EB/OL]. [2010-10-10]. http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

[2] QUINLAN J R. C4.5: programs for machine learning[M]. Burlington: Morgan Kaufmann Publishers, 1993: 17 - 42.

[3] QUINLAN J R. Induction of decision trees[J]. Machine Learning, 1986, 1(1): 81 - 106.

[4] WALLEY P. Inferences from multinomial data: learning about a bag of marbles [J]. Journal of the Royal Statistical Society, Series B: Methodological, 1996, 58(1): 3 - 57.

[5] ABELLAN J, MORAL S. Building classification trees using the total uncertainty criterion [J]. International Journal of Intelligent Systems, 2003, 18(12): 1215 - 1225.

[6] ABELLAN J, MASEGOSA A R. An experimental study about simple decision trees for bagging ensemble on datasets with classification noise [C]// Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, LNCS 5590. Berlin: Springer-Verlag, 2009: 446 - 456.

[7] ABELLAN J, MASEGOSA A R. Bagging schemes on the presence of class noise in classification [J]. Expert Systems with Applications, 2012, 39(8): 6827 - 6837.

[8] MANTAS C J, ABELLAN J. Analysis and extension of decision trees based on imprecise probabilities: application on noisy data [J]. Expert Systems with Applications, 2014, 41(5): 2514 - 2525.

[9] LI Y, JIANG D, LI F. The application of generating fuzzy ID3 algorithm in performance evaluation [J]. Procedia Engineering, 2012, 29: 229 - 234.

[10] JIN C, LI F, LI Y. A generalized fuzzy ID3 algorithm using generalized information entropy [J]. Knowledge-Based Systems, 2014, 64: 13 - 21.

[11] ARMBRUST M, FOX A, GRIFFITH R, *et al.* A view of cloud computing [J]. Communications of the ACM, 2010, 53(4): 50 - 58.

[12] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system [C]// SOSP 2003: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003: 29 - 43.

[13] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1): 107 - 113.

[14] ZHANG J, WONG J, LI T, *et al.* A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems [J]. International Journal of Approximate Reasoning, 2014, 55(3): 896 - 907.

[15] DAI W, JI W. A MapReduce implementation of C4.5 decision tree algorithm [J]. International Journal of Database Theory and Application, 2014, 7(1): 49 - 60.

[16] WANG R, HE Y-L, CHOW C-Y, *et al.* Learning ELM-tree from big data based on uncertainty reduction [J]. Fuzzy Sets and Systems, 2015, 258: 79 - 100.

[17] WITTEN I H, FRANK E. Data mining: practical machine learning tools and techniques [M]. Burlington: Morgan Kaufmann Publishers, 2005.

[18] HETTICH S, BLAKE C L, MERZ C J. UCI repository of machine learning database [EB/OL]. [2014-04-04]. http://archive.ics.uci.edu/ml/#/.