CN 43-1258/TP
ISSN 1007-130X

计算机工程与科学
Computer Engineering & Science

第 39 卷第 5 期 2017 年 5 月
Vol. 39, No. 5, May. 2017

文章编号:1007-130X(2017)05-0841-08

基于 MapReduce 的 Bagging 决策树优化算法*

张元鸣,陈　苗,陆佳炜,徐　俊,肖　刚
(浙江工业大学计算机科学与技术学院,浙江 杭州 310023)

摘　要:针对经典 C4.5 决策树算法存在过度拟合和伸缩性差的问题,提出了一种基于 Bagging 的决策树改进算法,并基于 MapReduce 模型对改进算法进行了并行化。首先,基于 Bagging 技术对 C4.5 算法进行了改进,通过有放回采样得到多个与初始训练集大小相等的新训练集,并在每个训练集上进行训练,得到多个分类器,再根据多数投票规则集成训练结果得到最终的分类器;然后,基于 MapReduce 模型对改进算法进行了并行化,能够并行化处理训练集、并行选择最佳分割属性和最佳分割点,以及并行生成子节点,实现了基于 MapReduce Job 工作流的并行决策树改进算法,提高了对大数据集的分析能力。实验结果表明,并行 Bagging 决策树改进算法具有较高的准确度与敏感度,以及较好的伸缩性和加速比。

关键词:决策树;Bagging;MapReduce 模型;大数据分析;准确性
中图分类号:TP390.027　　**文献标志码**:A
doi:10.3969/j.issn.1007-130X.2017.05.004

An optimized bagging decision tree algorithm based on MapReduce

ZHANG Yuan-ming, CHEN Miao, LU Jia-wei, XU-Jun, XIAO Gang
(College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: In order to address the shortcomings of overfitting and poor scalability of the C4.5 decision tree algorithm, we propose an optimized C4.5 algorithm with Bagging technique, and then parallelize it according to the MapReduce model. The optimized algorithm can obtain multiple new training sets that are equal to the initial training set by sampling with replacement. Multiple classifiers can be obtained by training the algorithm with these new training sets. A final classifier is generated according to a majority voting rule that integrates the training results. Then, the optimized algorithm is parallelized in three aspects, including parallel processing training sets, parallel selecting optimal decomposition attributes and optimal decomposition point, and parallel generating child nodes. A parallel algorithm based on job workflow is implemented to improve the ability of big data analysis. Experimental results show that the parallel and optimized decision tree algorithm has higher accuracy, higher sensitivity, better scalability and higher performance.

Key words: decision tree; Bagging; MapReduce model; big data analysis; accuracy

1　引言

决策树是用来对数据集进行分类的最有效方法,C4.5 作为决策树的经典分类算法,被广泛地应用于数据挖掘、模式识别、图像分割、文本分析等各个领域。它是一种以训练样本为基础的归纳学习算法,目标是监督学习,给定一个数据集,其中的每一个元组都能用一组属性值来描述,每一个元组属于一个互斥的类别中的某一类。它从一组(通常是

* 收稿日期:2017-01-20;修回日期:2017-03-25
基金项目:浙江省重大科技专项(2014C01408);浙江省公益性技术项目(2017C31014)
通信地址:310023 浙江省杭州市浙江工业大学计算机科学与技术学院
Address: College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, Zhejiang, P. R. China

数量越多越好)无序的训练实例或者说训练样本中推理得到以决策树为表示形式的分类规则,采用自顶向下的递归方式,在决策树的内部节点进行属性比较,根据属性值的不同走向该节点向下对应的分支,并最终在决策树的叶节点上得到结论,即类标号。从根到叶节点的一条路径就表示一条合取规则,整棵树就对应着一组析取表达式规则。

针对经典 C4.5 算法的不足,国内外研究者给出了一些优化和改进方法。Mantas 等[1]提出一种 Credal-C4.5 算法,该算法利用模糊概率思想获得特征与类变量之间的概率,同时使用一种新的属性分裂准则来评价此种模糊概率分布;陈家俊等[2]提出一种基于多尺度粗糙集模型的决策树改进算法,该算法引入尺度变量和尺度函数,采用不同尺度下的近似分类精度选择测试属性,同时结合抑制因子对决策树进行修剪;孟凡荣等[3]提出一种 PCA-DT 算法,该算法运用主成分分析方法来对信息增益和相关性度量这两个因子降维,得到一个综合指标,以此来选择优先划分的条件属性;王伟等[4]利用 NFPCA 算法思想对 C4.5 算法进行改进形成 NFPCA-in-C4.5 算法,该算法兼具降维和容噪功能,避免了降维中由特征信息损失和噪声残留造成的预测模型准确率大幅降低的问题;胡海斌等[5]提出了一种基于数据继承关系的 C4.5 算法,该算法具有较高的健壮性、精确度和较好的可理解性;孟祥福等[6]利用改进的决策树算法在 Web 查询结果集上自动构建一个带标签的分层分类树,以满足不同类型用户的个性化查询需求。

为了提高 C4.5 算法对大数据[7]的处理分析能力,研究者们基于不同的并行模型对 C4.5 算法进行了并行化。张莹等[8]提出一种基于 SPMD 的 C4.5 并行决策树算法,缩短了并行决策树的构建时间,提高了算法的性能;孙媛等[9]提出了一种基于 Hadoop 的 C4.5 算法,在一定程度上解决了 C4.5 算法在处理大数据时计算量大和构建决策树时间长的问题;Ben-Haim 等[10]提出一种基于流的并行决策树算法,取得了较好的可扩展性和较高的效率。

本文针对经典 C4.5 决策树算法存在过度拟合和伸缩性差的问题,提出一种基于 Bagging 技术的决策树优化方法,并基于 MapReduce 并行编程对其进行了并行化。并行优化算法不仅有效解决了过度拟合的问题,提高了分类的准确性,而且也提高了对大数据分析的能力。实验结果表明,并行 Bagging 决策树优化算法具有较高的准确度与敏感度,并在机群环境下获得了较高的性能。

2 Bagging 决策树优化方法

2.1 决策树基本思想

经典决策树 C4.5 算法的输入属性可以是离散型或连续型数值,输出属性为分类型,以信息熵为基础计算信息增益率,并将信息增益率作为选择最佳分割点和最佳分割属性的度量。

假设训练样本集 $S=\{\{\boldsymbol{x}_1,y_1\},\{\boldsymbol{x}_2,y_2\},\cdots,\{\boldsymbol{x}_s,y_m\}\}$ 有 s 个样本,其中样本 x 是包含了 M 个属性的向量,y_i 为类别。S 被划分为 m 个类别,$i=\{1,\cdots,m\}$,$s\geqslant m$,第 i 个子集 S_i 的样本数为 s_i。则把集合 S 划分为 m 个类别的信息熵为:

$$I(S)=-\sum_{i=1}^{m}\frac{S_i}{S}\log_2\left(\frac{S_i}{S}\right) \tag{1}$$

测试属性 A 有 v 个取值 $\{a_1,a_2,\cdots,a_v\}$(对连续属性来说,$v=2$),将 S 划分为 v 个子集 $\{S_1,S_2,\cdots,S_v\}$。属性 A 划分的第 j 个子集 S_j 的样本数为 s_j,$j=\{1,2,\cdots,v\}$,其中分类为第 i 类的样本有 s_{ij} 个,则 $s_{1j}+s_{2j}+\cdots+s_{mj}=s_j$。则 S 被属性 A 划分为 v 个子集的信息熵为:

$$H(A)=E(I(S_j))=-\sum_{j=1}^{v}\frac{s_j}{s}\cdot I(S_j) \tag{2}$$

其中,属性 A 划分的子集 S_j 被划分为 m 个类别的信息熵为:

$$I(S_j)=-\sum_{i=1}^{m}\frac{s_{ij}}{s_j}\log_2\left(\frac{s_{ij}}{s_j}\right) \tag{3}$$

则 S 被属性 A 划分的信息增益为:

$$Gain(A)=I(S)-H(A) \tag{4}$$

属性 A 的分裂信息为:

$$SplitInfo(A,S)=-\sum_{j=1}^{v}\frac{s_j}{s}\log_2\left(\frac{s_j}{s}\right) \tag{5}$$

则 S 被属性 A 划分的信息增益率 $GainRatio(A,S)$ 为:

$$GainRatio(A,S)=\frac{Gain(A)}{SplitInfo(A,S)} \tag{6}$$

基于以上信息增益率的计算方法,C4.5 算法先对数据对象进行数据预处理,去掉与决策树无关的属性和高分支属性,将数值型属性进行归纳概化以及处理包含空缺值的属性,形成决策树的训练集。

在对训练集进行训练时,计算每个属性的信息增益和信息增益率,选择信息增益率最大的且同时获取的信息增益又不低于所有属性平均值的属性,作为当前的主属性节点,为该子节点所包含的样本

自行递归地执行上述过程,直到子集中的数据记录在主属性上取值都相同,或没有属性可供再划分使用为止,生成初始的决策树。

2.2 Bagging 决策树算法

然而,C4.5 算法的计算过程存在较高的方差,数据较小的变化会产生完全不同的分割,导致分类树过于复杂,包含了太多的噪声,即产生"过度拟合"现象。一旦产生过度拟合,分类模型会出现对训练数据准确度很高,而对测试数据准确度很低的现象。为解决"过度拟合"问题,本文提出了基于 Bagging 的 C4.5 优化算法,简称 Bagging-C4.5,该算法能够较好地解决过度拟合的问题。

Bagging 是用于多分类器集成的技术[11,12],其基本思想是:给定一个弱分类器和一个训练集,让弱分类器训练 T 轮,每轮的训练集由初始训练集中随机取出的 x 个训练样本组成,每轮训练完成后得到一个预测函数 h,训练 T 轮得到 T 个预测函数 $h_1,h_2,\cdots,h_T$。用此预测函数序列对样本集进行预测,然后按照多数投票规则得到最后的预测结果 h^*。

Bagging-C4.5 优化算法的步骤如算法 1 所示。

算法 1 Bagging-C4.5 优化算法

输入:训练样本集 S,迭代次数 T,根节点 R,阈值 n;

输出:分类器 H。

步骤:

for i=1 to T by 1 do

  S_i=$resampling(S)$;/* 利用重采样技术获得新训练集 S_i */

end for

for 所有 S_i do

  $Tree_i$=$\{R\}$;

  if S_i 都属于同一类别 C,或 S_i 中剩余样本数量少于 n,或 S_i 为空集

    返回 R 为叶节点,标记为类别 C;

  end if

  for 所有属性 do

    计算 $GainRatio$;

  endfor

  最佳分割属性 a_{best}=argmax($GainRatio$);

  $Tree_i$=$\{$创建 a_{best} 分叉$\}$;

  $\{S_v\}$=S_i 由 a_{best} 分割的子集;

  for 所有 S_v do

    $Tree_v$=$C4.5(S_v)$;

    $Tree_i$=$\{R,Tree_v\}$;

  end for

end for

H= argmax $\sum_{i=1}^{T} I(Tree_i)$; /* 通过投票集成训练结果 */

3 基于 MapReduce 的 Bagging 决策树算法

3.1 MapReduce Job 工作流

MapReduce 是一种被广泛应用于大数据分析的并行编程模型[13],该模型由一个负责元数据组织和任务调度的 Master 节点和若干个负责数据存储和计算的 Slave 节点组成。MapReduce 将任务分割成 Map 函数和 Reduce 函数,就能够实现任务的并行处理。Map 函数和 Reduce 函数处理的数据类型如下所示:

$$Map(key_1,value_1)\rightarrow List(key_2,value_2)$$

$$Reduce(key_2,List(value_2))\rightarrow(key_3,value_3)$$

Map 函数用来将输入的一组键值对(key_1/$value_1$)按用户逻辑进行处理,并输出中间结果(key_2/ $value_2$)。MapReduce 计算框架会自动合并中间结果中具有相同 key 值的 value 并交由 Reduce 函数处理。Reduce 函数按用户逻辑对中间结果进行处理,输出键值对 $key_3/value_3$。MapReduce 框架将问题分而治之,并为了减少计算过程中大量的数据通信开销按就近原则分配计算任务。

图 1 给出了 MapReduce 并行编程模型的工作流程,其执行的过程如下:用户将业务逻辑转变为 MapReduce 作业并提交到 Master 节点上;Master 节点采用就近原则将 Map 任务和 Reduce 任务分配给空闲的 Slave 节点;被分配到 Map 任务的 Slave 节点对输入分片中的每条记录执行 Map 函数,产生键值对($key_2/value_2$);Reduce 任务节点处理键值对($key_2/value_2$)集合,对相同键的值进行合并处理,执行用户提供的 Reduce 函数,并将最终结果保存到分布式文件系统上。

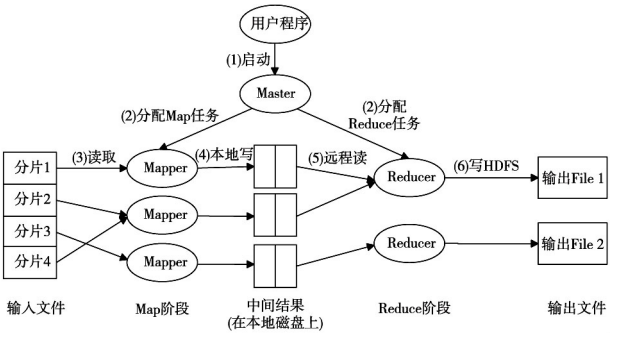


Figure 1 Execution procedure of MapReduce

图 1 MapReduce 执行流程

如果多个 MapReduce Job 之间存在相互依赖关系,就形成了 Job 工作流。图 2 给出了包含 3 个 Job 的工作流,Job1 和 Job2 相互独立,而 Job3 依赖于 Job1 和 Job2 的输出。

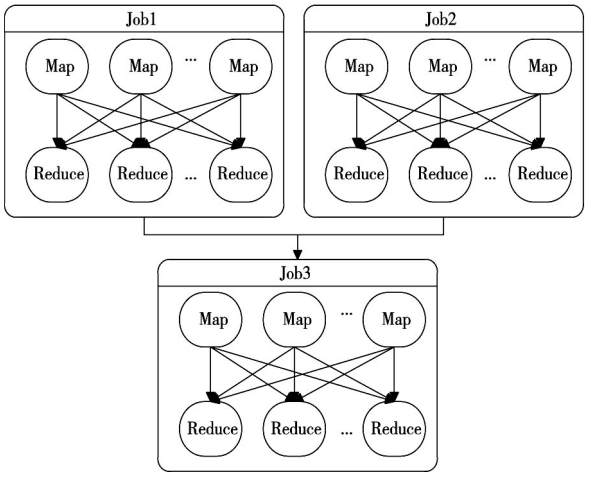


Figure 2 Workflow of three MapReduce Jobs

图 2 MapReduce Job 工作流实例

3.2 并行 Bagging 决策树算法

为了提高 Bagging-C4.5 处理大数据的性能,本文基于 MapReduce 并行编程模型对其进行了并行化,即并行 Bagging-C4.5 算法,该并行算法对训练集、选择最佳分割属性和最佳分割点、生成子节点进行了并行化处理器。

并行 Bagging-C4.5 算法的实现过程为:多个节点同时扫描数据集的不同子集,避免对整个数据集扫描导致占用内存过大;设计子任务,并行计算信息增益率,寻找最佳分割属性和最佳分割点;递归调用算法,并行生成子节点,构建决策树。除此之外,为了方便进行加数计算,在并行计算过程中,可以把 s_{ij} 存入多维数组,形成 s_{ij} 计数矩阵。

图 3 给出了并行 Bagging-C4.5 算法的流程图,该并行算法包括 $JobTracker1$、$JobTracker2$ 和 $JobTracker3$ 三个作业,构建了一个 Job 工作流。首先 $JobTracker1$ 产生 T 个新训练集,然后 $JobTracker2$ 找出按每个属性分割的信息增益率最大值,接着 $JobTracker3$ 找出叶节点上各个属性的信息增益率最大值并完成叶节点的分割。循环进行 $JobTracker2$ 和 $JobTracker3$ 过程,直到树的生长结束,即满足剩余样本都属于同一类别,或剩余样本数量少于某一阈值,或无剩余样本这三种情况之一即可。

下面给出三个 MapReduce Job 的具体实现过程。

(1)$JobTracker1$ 的 Map 函数和 Reduce 函数

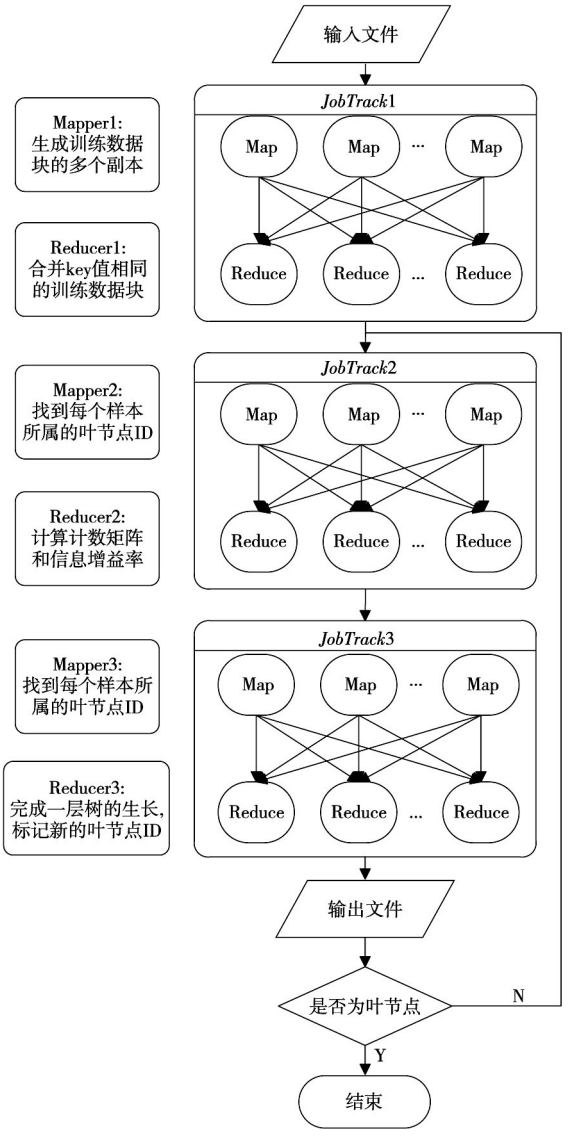


Figure 3 Flow diagram of the parallel bagging-C4.5 algorithm

图 3 并行 Bagging-C4.5 算法流程图

的实现过程如算法 2 和算法 3 所示,Map 函数将训练样本数据块划分为 T 份数据块,Reduce 函数归并 key 值相同的数据块并得到新训练集。

算法 2 $JobTracker1$ 的 Map 算法

输入:〈行号,{M 个属性向量}〉;

输出:〈整数,{M 个属性向量}〉。

步骤 1 载入所有训练样本数据块;

步骤 2 随机生成 T 个随机整数;

步骤 3 输出整数为 key,数据块内容为 $value$ 的 T 个 $key/value$ 对。

算法 3 $JobTracker1$ 的 Reduce 算法

输入:〈整数,{M 个属性向量}〉;

输出:〈行号,{M 个属性向量}〉。

步骤 1 将 key 值相同的训练样本数据块进行合并;

步骤 2 输出 $key/value$ 对。

(2)$JobTracker2$ 的 Map 函数和 Reduce 函数

的实现过程如算法 4 和算法 5 所示,Map 函数找到训练样本落入的叶节点,Reduce 函数计算并找到信息增益率最大值,对连续属性找出其最佳分割点。

算法 4 *JobTracker2* 的 Map 算法

输入:〈行号,{M 个属性的向量}〉;

输出:〈{叶节点 id,属性 id},{属性值,类别值}〉。

步骤 1 载入所有训练样本;

步骤 2 找到它们落入的叶节点 id;

步骤 3 输出 $key/value$ 对。

算法 5 *JobTracker2* 的 Reduce 算法

输入:〈{叶节点 id,属性 id},{属性值,类别值}〉;

输出:〈行号,{叶节点 id,属性 id,属性分割,{$GainRatio$,计数矩阵}}〉。

步骤 1 计算按每个属性分割时对应的 s_{ij} 计数矩阵;

步骤 2 计算信息增益率,找到信息增益率最大值;

步骤 3 对连续属性找出其最佳分割点;

步骤 4 输出 $key/value$ 对。

(3)*JobTracker3* 的 Map 函数和 Reduce 函数的实现过程如算法 6 和算法 7 所示,Map 函数找到样本所属的叶节点,Reduce 函数完成一层树的生长并标记新节点。

算法 6 *JobTracker3* 的 Map 算法

输入:〈行号,{叶节点 id,属性 id,属性分割,{$GainRatio$,计数矩阵}}〉;

输出:〈叶节点 id,{属性 id,属性分割,{$GainRatio$,计数矩阵}}〉。

步骤 1 找到每个样本所属的叶节点 id;

步骤 2 输出 $key/value$ 对。

算法 7 *JobTracker3* 的 Reduce 算法

输入:〈叶节点 id,{属性 id,属性分割,{$GainRatio$,计数矩阵}}〉;

输出:〈行号,{叶节点 id,属性 id,最佳分割,{$GainRatio$,计数矩阵}}〉。

步骤 1 遍历每个叶节点上所有的分割属性;

步骤 2 找到有信息增益率最大值的分割属性和分割点作为该叶节点的最佳分割;

步骤 3 完成一层树的生长,并标记新的叶节点 id;

步骤 4 输出 $key/value$ 对。

4 实验与分析

4.1 实验环境

搭建了一个包括五个节点的机群作为实验环境,每个节点的主频为 2.1 GHz,内存都是 4 GB,节点上均安装了 Ubuntu14.04 操作系统,配置了 Hadoop 2.6.0 环境,将其中一个节点计算机配置为 Master,作为 NameNode 和 JobTracker 使用,其余四个节点作为 Slave,作为 DataNode 和 TaskTracker 使用,实验环境信息如表 1 所示。

Table 1 Configuration of cluster

表 1 机群节点配置

节点名称	IP 地址
Master	192.168.1.100
Slave1	192.168.1.101
Slave2	192.168.1.102
Slave3	192.168.1.103
Slave4	192.168.1.104
网关:192.168.1.1	
子网掩码:255.255.255.0	

4.2 实验数据

实验数据来自专业技术资格(职务)评审云平台中[14]的实际数据,并基于文本 ETL 技术将评审文档转换为结构化的数据集,为了分析申报对象的业务水平,选择与业绩相关的数据,包括论文、科研、获奖等数据,共包括 520 万个元组,这些数据集构成了本实验的全体数据集。

通过决策树可以实现对数据集的分类,如对于论文数据集,可以分为国际 SCI 期刊、国际 EI 期刊、国际普通期刊和国际会议,对于国内期刊,可以分为 A 类、B 类和普通期刊;对于科研项目可以分为国家级重点项目、国家级项目、省部级项目、厅级项目、重大横向项目和一般横向项目;对知识产权可以分为发明专利、实用新型、软件著作权;获奖情况可以分为国家级、省部级和厅级。根据分类结果,对每一类别数据赋予相应的权重,如国际 SCI 期刊 4 分、国际 EI 期刊 3 分、国际普通期刊 2 分、国际会议 2 分、国内 A 类 3 分、国内 B 类 2 分和国内普通期刊 1 分。根据分类及权重可以综合计算专业技术人员的业绩总分,从而为综合评价专业技术人员的业务水平提供依据。

本实验对比分析了 Weka 中的 J48 经典算法(C4.5 算法的标准实现)、并行 C4.5 算法[9]、Bagging-C4.5 算法和并行 Bagging-C4.5 算法,以验证本文提出的决策树优化算法的有效性。从数据集中随机抽取 150 万个元组作为训练样本集,设置并行 C4.5 中连续属性的分割点数量为 50,设置 Bagging-C4.5 和并行 Bagging-C4.5 的迭代次数为 8,设置决策树每个叶节点至少含有 2 个样本,且置信因数为 0.4。

4.3 算法准确度和敏感度

准确度 ACC 描述了分类器正确预测样本类别

的能力。其定义为：

$$ACC = \sum_{i=0}^{m} M_{ij} / \sum_{i,j=0}^{m} M_{ij} \times 100 \quad (7)$$

其中，m 为类别数，m 维混淆矩阵 $\boldsymbol{M}$ 是模型分类结果的计数矩阵，M_{ij} 代表实际是第 i 类而被分为第 j 类的样本个数，$i,j = 1,2,\cdots,m$。

由于准确度高度依赖于数据分布，不同类别中的分类准确度可能不同，还需用敏感度评价分类模型的正确率，第 i 类的敏感度 SEN_i 可以定义为：

$$SEN_i = M_{ij} / \sum_{j=1}^{m} M_{ij} \times 100\% \quad (8)$$

为比较方便，取 m 个类别的敏感度平均值 SEN：

$$SEN = \frac{1}{m} \sum_{i=1}^{m} SEN_i = \frac{1}{m} \sum_{i=1}^{m} \frac{M_{ij}}{\sum_{j=1}^{m} M_{ij}} \times 100\% \quad (9)$$

从数据集中随机抽取一些元组用来测试分类规则的准确度和敏感度，这些测试样本集的名称和元组个数是：$A1(400)$、$A2(4\ 000)$、$A3(40\ 000)$、$A4(80\ 000)$，如 $A1$ 样本集中包含 400 个元组。以这四个样本作为输入得到的算法的准确度和敏感度如图 4 和图 5 所示。

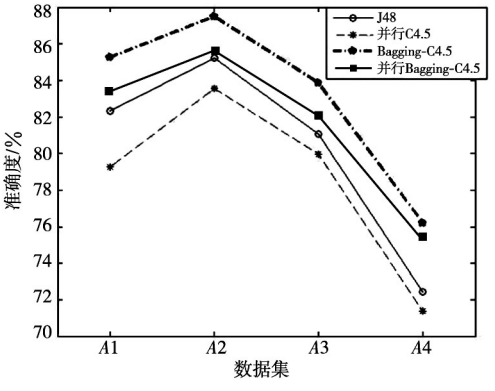


Figure 4 Accuracy of the four decision tree algorithms

图 4 四种决策树算法的准确度

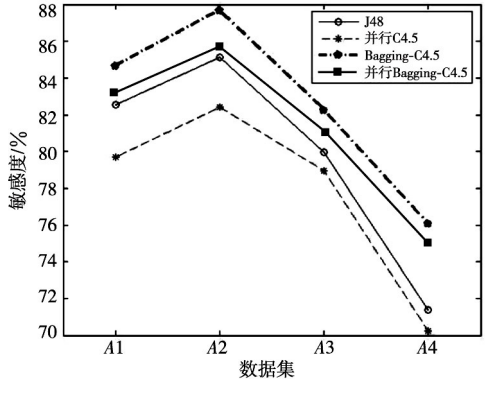


Figure 5 Sensitivity of the four decision tree algorithms

图 5 四种决策树算法的敏感度

根据实验结果可知，四种决策树算法的准确度和敏感度都在 70% 以上，说明所设计的专业技术资格评审数据分析方案是合理的。Bagging-C4.5 算法的准确度和敏感度都比 J48 算法要好，并行 Bagging-C4.5 算法的准确度和敏感度都比并行 C4.5 算法要好，而并行 Bagging-C4.5 算法的准确度和敏感度与 BBC4.5 算法相差不大，说明本文提出的基于 Bagging 的决策树优化方法具有较大的优势。

4.4 算法伸缩性

伸缩性表示算法对不同大小数据集的处理能力，为测试该特性，实验设置了四个不同大小的数据集，分别是：$B1$（4 000）、$B2$（40 000）、$B3$（400 000）和 $B4$（4 000 000），如 $B1$ 样本集中包含 4 000 个元组。根据这四个样本数据集再次运行四个算法后得到的结果如图 6 所示，其中 J48 算法和 Bagging-C4.5 算法是在单个节点上运行的，并行 C4.5 算法和 Bagging-C4.5 算法是在包含五个节点的集群上运行的。

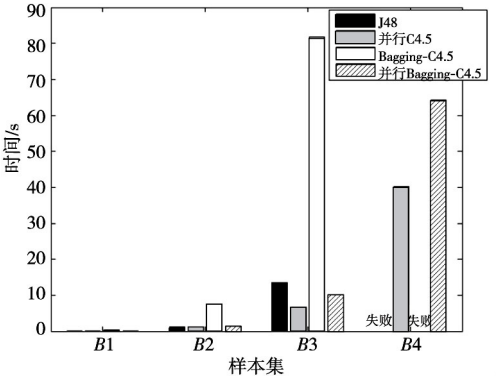


Figure 6 Scalability of the four decision tree algorithms

图 6 四种决策树算法的伸缩性

根据实验结果，Bagging-C4.5 算法由于重复取样及投票表决等过程的影响，算法效率比 J48 要低；并行 C4.5 算法运行时间比 J48 算法短；并行 Bagging-C4.5 算法运行时间比 Bagging-C4.5 算法短，但比 J48 算法长，说明并行化算法比串行算法具有较好的伸缩性；当数据集为 4 000 000 时，由于内存原因，J48 算法和 Bagging-C4.5 算法无法完成建模，伸缩性较差；而并行 C4.5 算法和并行 Bagging-C4.5 算法能够完成计算，说明了并行 C4.5 算法和并行 Bagging-C4.5 算法能够处理比较大的数据集，具有良好的伸缩性。

4.5 算法加速比

进一步测试并行 Bagging-C4.5 在不同节点数量上的加速比，分别设置集群节点数量为 1、2、3、4

和 5,在样本集 $B1$、$B2$、$B3$ 和 $B4$ 上再次运行该并行算法,得到结果如图 7 所示。

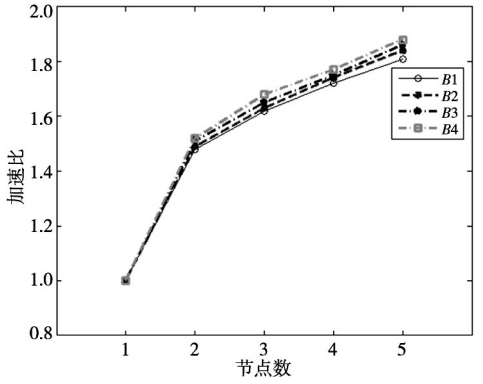


Figure 7 Speedup of the parallel Bagging-C4.5 algorithm

图 7 并行 Bagging-C4.5 算法的加速比

根据实验结果,随着节点数的增加,并行 Bagging-C4.5 算法的加速比得到明显提升,在较多节点情况下获得了较高的加速比。但是,随着节点数的增加,算法的加速比增加的倍数减少,这是由于节点增多的同时节点间通信导致的开销也增多,部分抵消了加速比增加的倍数。

5 结束语

C4.5 算法是用来对数据集进行自动分类的经典决策树算法,被广泛应用于数据挖掘、模式识别、图像分割、文本分析等各个领域,但是该算法存在过度拟合和伸缩性较差的问题。针对这些问题,本文给出了一种基于 Bagging 技术的决策树优化算法(Bagging-C4.5),该算法通过有放回采样得到多个与初始训练集大小相等的新训练集,并在每个训练集上进行训练,得到多个分类器;然后根据多数投票规则集成训练结果并得到最终的分类器。此外,还实现了一种基于 MapReduce 的 Bagging-C4.5 优化算法,通过对算法关键步骤的并行化提高了对大数据分析的能力。实验结果表明,本文提出的并行 Bagging 决策树优化算法具有较高的准确度与敏感度、较好的伸缩性以及较高的性能加速比。

参考文献:

[1] Mantas C J, Abellan J. Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data[J]. Expert Systems with Applications, 2014, 41(10): 4625-4637.

[2] Chen Jia-jun, Su Shou-bao, Xu Hua-li. Decision tree optimization algorithm based on multi-scale rough set model [J]. Journal of Computer Applications, 2011, 31(12): 3243-3246. (in Chinese)

[3] Meng Fan-rong, Jiang Xiao-yun, Tian Tian, et al. Decision tree construction method based on principal component analysis [J]. Journal of Chinese Computer Systemsm2008, 29(7): 1245-1249. (in Chinese)

[4] Wang Wei, Li Lei, Zhang Zhi-hong. Improvement of c4.5 algorithm with free noise capacity [J]. Computer Science, 2015, 42(12): 268-271. (in Chinese)

[5] Hu Hai-bin, Qiu Ming, Jiang Qing-shan. An improved classification algorithm of C4.5 based relationship [J]. Journal of Computer Research and Development, 2009, 46 (Suppl.): 491-496. (in Chinese)

[6] Meng Xiang-fu, Ma Zong-min, Zhang Xiao-yan. A categorization approach based on adapted decision tree algorithm for web databases query results [J]. Journal of Computer Research and Development, 2012, 49(12): 2656-2670. (in Chinese)

[7] Meng Xiao-feng, Ci Xiang. Big data management: Concept, techniques and challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169. (in Chinese)

[8] Zhang Ying, Bi Zhuo. Analysis of parallel C4.5 decision tree acceleration based on SPMD [J]. Computer Technology and Development, 2015, 25(1): 29-32. (in Chinese)

[9] Sun Yuan, Huang Gang. Analysis and study of C4.5 algorithm based on Hadoop platform [J]. Computer Technology and Development, 2014, 24(11): 83-86. (in Chinese)

[10] Ben-Haim Y, Tom-Tov E. A streaming parallel decision tree algorithm [J]. Journal of Machine Learning Research, 2010, 11(11): 849-872.

[11] Zhang Xiang, Zhou Ming-quan, Geng Guo-hua, Research on improvement of bagging chinese text categorization classifier [J]. Journal of Chinese Computer Systems, 2010, 31(2): 281-284. (in Chinese)

[12] He Liang, Song Qin-bao, Hai Zhen, et al. Bagging-based ensemble model and algorithm of k-NN prediction [J]. Control and Decision, 2010, 25(1): 48-52. (in Chinese)

[13] Cheng Xue-qi, Jin Xiao-long, Wang Yuan-zhuo, et al. Survey on big data system and analytic technology [J]. Journal of Software, 2014, 25(9): 1889-1908. (in Chinese)

[14] Zhang Y M, Ni K, Lu J W, et al. DOGCP: A domain-oriented government cloud platform based on paas [C] // Proc of IEEE International Conference on Cyber Security and Cloud Computing, 2015: 115-120.

[15] Schölkopf B, Platt J, Hofmann T. Map-Reduce for machine learning on multicore[M] // Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2006: 281-288.

[16] Li Jun-hua. Cloud computing and several data mining algorithms parallelization with MapReduce[D]. Chengdu: University of Electronic Science and Technology of China, 2010. (in Chinese)

[17] Yang Lai, Shi Zhong-zhi, Liang Fan, et al. Parallel approach in data mining based on Hadoop cloud platform [J]. Journal

of System Simulation,2013,25(5):936-944. (in Chinese)

[18] Zhang C,Li F,Jestes J. Efficient parallel kNN joins for large data in MapReduce[C]// Proc of International Conference on Extending Database Technology,2012:38-49.

[19] Luo X,Qiu K,Liang P,et al. Speeding up large-scale next generation sequencing data analysis with pBWA[J]. Journal of Applied Bioinformatics & Computational Biology,2012,1(1):1.

[20] Lu Q, Cheng X H. The research of decision tree mining based on Hadoop[C]// Proc of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012),2012:798-801.

[21] Chang K W,Roth D. Selective block minimization for faster convergence of limited memory large-scale linear models[C]// Proc of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,2011:699-707.

[22] Moretti C,Steinhaeuser K,Thain D,et al. Scalingup classifiers to cloud computers[C]// Proc of the 8th IEEE International Conference on Data Mining,2008:472-481.

附中文参考文献:

[2] 陈家俊,苏守宝,徐华丽. 基于多尺度粗糙集模型的决策树优化算法[J]. 计算机应用,2011,31(12):3243-3246.

[3] 孟凡荣,蒋晓云,田恬,等. 基于主成分分析的决策树构造方法[J]. 小型微型计算机系统,2008,29(7):1245-1249.

[4] 王伟,李磊,张志鸿. 具有容噪特性的 C4.5 算法改进[J]. 计算机科学,2015,42(12):268-271.

[5] 胡海斌,邱明,姜青山. 一种基于数据继承关系的 C4.5 分类优化算法[J]. 计算机研究与发展,2009,46(Suppl.):491-496.

[6] 孟祥福,马宗民,张霄雁. 基于改进决策树算法的 Web 数据库查询结果自动分类方法[J]. 计算机研究与发展,2012,49(12):2656-2670.

[7] 孟小峰,慈祥. 大数据管理:概念、技术与挑战[J]. 计算机研究与发展,2013,50(1):146-169.

[8] 张莹,毕卓. 基于 SPMD 的 C4.5 并行决策树加速分析[J]. 计算机技术与发展,2015,25(1):29-32.

[9] 孙媛,黄刚. 基于 Hadoop 平台的 C4.5 算法的分析与研究[J]. 计算机技术与发展,2014,24(11):83-86.

[11] 张翔,周明全,耿国华,等. Bagging 中文文本分类器的改进方法研究[J]. 小型微型计算机系统,2010,31(2):281-284.

[12] 何亮,宋擒豹,海振,等. 基于 Bagging 的组合 k-NN 预测模型与方法[J]. 控制与决策,2010,25(1):48-52.

[13] 程学旗,靳小龙,王元卓,等. 大数据系统和分析技术综述[J]. 软件学报,2014,25(9):1889-1908.

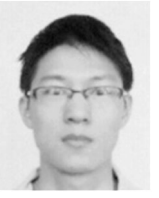
[16] 李军华. 云计算及若干数据挖掘算法的 MapReduce 化研究[D]. 成都:电子科技大学,2010.

[17] 杨来,史忠植,梁帆,等. 基于 Hadoop 云平台的并行数据挖掘方法[J]. 系统仿真学报,2013,25(5):936-944.

作者简介:

张元鸣(1977-),男,河南濮阳人,博士,副教授,CCF 会员(19490M),研究方向为并行计算和云计算。**E-mail**:zym@zjut.edu.cn

ZHANG Yuan-ming, born in 1977, PhD, associate professor, CCF member (19490M), his research interests include parallel computing, and cloud computing.



陈苗(1991-),男,浙江绍兴人,硕士,研究方向为大数据处理。**E-mail**:2111312003@zjut.edu.cn

CHEN Miao, born in 1991, MS, his research interest includes big data processing.

陆佳炜(1981-),男,浙江湖州人,硕士,讲师,研究方向为云计算和服务计算。**E-mail**:viivan@zjut.edu.cn

LU Jia-wei, born in 1981, MS, lecturer, his research interests include cloud computing, and service computing.

徐俊(1979-),男,浙江永嘉人,硕士,高级实验师,研究方向为云计算和服务计算。**E-mail**:xujun@zjut.edu.cn

XU-Jun, born in 1979, MS, senior experimentalist, his research interests include cloud computing, service computing.

肖刚(1965-),男,浙江上虞人,博士,教授,研究方向为云计算和智能信息系统。**E-mail**:xg@zjut.edu.cn

XIAO Gang, born in 1965, PhD, professor, his research interests include cloud computing, and intelligent information system.