(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102214213 A
(43)申请公布日 2011.10.12

(21)申请号 201110143821.7

(22)申请日 2011.05.31

(71)申请人 中国科学院计算技术研究所
地址 100080 北京市海淀区中关村科学院南路 6 号

(72)发明人 庄福振 何清

(74)专利代理机构 北京律诚同业知识产权代理有限公司 11006
代理人 祁建国 梁挥

(51)Int.Cl.
*G06F 17/30*(2006.01)

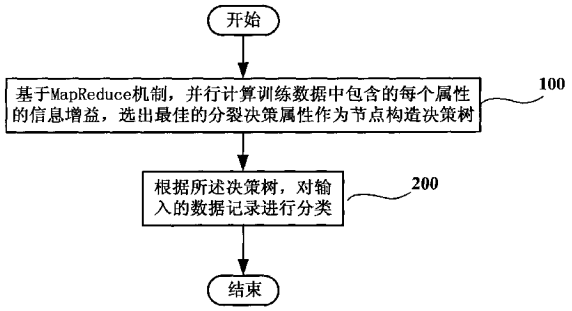
权利要求书 2 页 说明书 10 页 附图 3 页

(54)发明名称

一种采用决策树的数据分类方法和系统

(57)摘要

本发明公开了一种采用决策树的数据分类方法和系统。所述方法，包括下列步骤：基于MapReduce机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树；根据所述决策树，对输入的数据记录进行分类。其实现了基于MapReduce的并行决策树ID3算法，不仅可以处理大规模数据集，而且并行效率高，即实现构建决策树中节点内部以及同一层节点之间的并行计算。




CN 102214213 A

1. 一种采用决策树的数据分类方法，其特征在于，所述方法，包括下列步骤：

步骤 100，基于 MapReduce 机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树；

步骤 200，根据所述决策树，对输入的数据记录进行分类。

2. 根据权利要求 1 所述的采用决策树的数据分类方法，其特征在于，所述步骤 100，包括下列步骤：

步骤 110，启动一个进程，计算训练数据中包含的每个属性的信息增益，选出最大值作为根节点的分裂属性，并计算决策规则以及传给第一层的前缀信息：

步骤 120，判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，执行步骤 130；否则，执行步骤 130；

步骤 130，判断是否产生新的前缀信息，若是，则执行步骤 140；否则执行步骤 160；

步骤 140，决策树层数加一，判断当前决策树的层数是否小于训练数据中包含的所有属性的总数，若是，则执行步骤 150；否则执行步骤 160；

步骤 150，启动一个新的进程，计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息，返回步骤 120；

步骤 160，结束训练，根据计算得到的决策规则构建决策树。

3. 根据权利要求 1 所述的采用决策树的数据分类方法，其特征在于，在进行属性的信息增益计算时采用 MapReduce 函数采集数据，在 Map 函数中，根据头文件信息对读入的每一行样本进行解析，产生中间的 <key，value> 对，key 为前缀信息 + 类别信息 + 条件属性的名字 + 条件属性的值或者前缀信息 + 类别信息，若没有前缀信息，则为空，value 为 1，Map 函数的输入 key 和 value 分别为样本的在分布式文件系统上偏移位置和样本本身，Reduce 函数对中间 <key，value> 对进行融合。

4. 根据权利要求 1 所述的采用决策树的数据分类方法，其特征在于，所述步骤 200 中，构建完成的决策树保存在 HDFS 文件系统中，当进行测试时，每个节点从 HDFS 中提取决策树，对输入的数据记录进行分类。

5. 一种采用决策树的数据分类系统，其特征在于，所述系统，包括：

训练模块，用于基于 MapReduce 机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树；

分类模块，用于根据所述决策树，对输入的数据记录进行分类。

6. 根据权利要求 5 所述的采用决策树的数据分类系统，其特征在于，所述训练模块，包括：

信息增益计算模块，用于计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息：

决策规则判断模块，用于判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，触发前缀信息判断模块；否则，直接触发前缀信息判断模块；

前缀信息判断模块，用于判断是否产生新的前缀信息，若是，则触发阈值判断模块；否则结束训练，根据计算得到的决策规则构建决策树；

阈值判断模块，决策树层数加一，判断当前决策树的层数是否小于训练数据中包含的所有属性的总数，若是，则触发信息增益计算模块；否则结束训练，根据计算得到的决策规则构建决策树。

7. 根据权利要求 5 所述的采用决策树的数据分类系统，其特征在于，所述训练模块中，在进行属性的信息增益计算时采用 MapReduce 函数采集数据，在 Map 函数中，根据头文件信息对读入的每一行样本进行解析，产生中间的 <key，value> 对，key 为前缀信息 + 类别信息 + 条件属性的名字 + 条件属性的值或者前缀信息 + 类别信息，若没有前缀信息，则为空，value 为 1，Map 函数的输入 key 和 value 分别为样本的在分布式文件系统上偏移位置和样本本身，Reduce 函数对中间 <key，value> 对进行融合。

8. 根据权利要求 5 所述的采用决策树的数据分类系统，其特征在于，所述分类模块，控制每个节点将保存在 HDFS 文件系统中的构建完成的决策树提取出来，对输入的数据记录进行分类。

# 一种采用决策树的数据分类方法和系统

## 技术领域

[0001] 本发明涉及数据挖掘技术领域，特别是涉及一种采用决策树的数据分类方法和系统。

## 背景技术

[0002] 分类是数据挖掘中的一个重要课题。分类的目的是学会一个分类函数或分类模型（也常常称作分类器），该模型能把数据库中的数据项映射到给定类别中的某一个。分类可用于提取描述重要数据类的模型或预测未来的数据趋势。分类的目的是分析输入数据，通过在训练集中的数据表现出来的特性，为每一个类找到一种准确的描述或者模型。这种描述常常用谓词表示。由此生成的类描述用来对未来的测试数据进行分类。尽管这些未来的测试数据的类标签是未知的，我们仍可以由此预测这些新数据所属的类。注意是预测，而不能肯定。

[0003] 分类技术有很多，如决策树、贝叶斯网络、神经网络、遗传算法、关联规则等。其中，决策树技术是用于分类和预测的主要技术，决策树学习是以实例为基础的归纳学习算法。它着眼于从一组无次序、无规则的事例中推理除决策树表示形式的分类规则。它采用自顶向下的递归方式，在决策树的内部节点进行属性值的比较并根据不同属性判断从该节点向下的分支，然后进行剪枝，最后在决策树的叶节点得到结论。所以从根到叶节点就对应着一条合取规则，整棵树就对应着一组析取表达式规则。

[0004] 决策树 (Decision Tree) 又称为判定树，是运用于分类的一种树结构。其中的每个内部结点代表对某个属性的一次测试，每条边代表一个测试结果，叶结点代表某个类或者类的分布，最上面的结点是根结点。决策树分为分类树和回归树两种，分类树对离散变量做决策树，回归树对连续变量做决策树。1986 年 Quinlan 提出了著名的 ID3 算法 [Quinlan, 1986]。在 ID3 算法的基础上，1993 年 Quinlan 又提出了 C4.5 算法 [Quinlan,1993]。为了适应处理大规模数据集的需要，后来又提出了若干改进的算法，其中 SLIQ(supervised learning in quest) 和 SPRINT(scalable parallelizable induction of decision trees) 是比较有代表性的两个算法。

[0005] 构造决策树是采用自上而下的递归构造方法。决策树构造的结果是一棵二叉或多叉树，它的输入是一组带有类别标记的训练数据。二叉树的内部结点（非叶结点）一般表示为一个逻辑判断，如形式为 a ＝ b 的逻辑判断，其中 a 是属性，b 是该属性的某个属性值；树的边是逻辑判断的分支结果。多叉树 (ID3) 的内部结点是属性，边是该属性的所有取值，有几个属性值，就有几条边。树的叶结点都是类别标记。

[0006] 使用决策树进行分类，首先利用训练集建立并精化一棵决策树，建立决策树模型。这个过程实际上是一个从数据中获取知识，进行机器学习的过程。然后利用生成完毕的决策树对输入数据进行分类。对输入的记录，从根结点依次测试记录的属性值，直到到达某个叶结点，从而找到该记录所在的类。

[0007] 构造决策树最大的运算代价在于计算选择最佳分裂属性，因为选择分裂的时候，

对每个字段都考虑；对每个字段中的值先排序，然后再一一计算，最后选出最佳的分裂属性。常见的衡量准则有信息熵和 GiniIndex 等方法。决策树算法内部的并行性，实际上跟数据本身的存储有很大的关系，如果数据纵向划分存储，即每个节点只存储数据的部分属性，那么这种存储方式具有较好的并行性；而如果数据横向存储，分布在各个数据节点的那种情况，那么比较难用决策树算法进行并行化处理，特别是第一个根节点的分裂就只能串行执行，在根节点分裂完成以后，根节点下的两个子节点可以用相应的两台机器进行并行处理，以此类推。从本质上来讲，如果数据是横向存储且想得到全局分类决策树，那么很难进行并行处理，且节点之间不能并行，只能串行。

[0008] 为了处理大规模数据且数据都是横向划分的情况，现有的很多构造决策树的工作[宋，2007]都是分布式的。图 1 是现有技术中分布式构造决策树的工作示意图，如图 1 所示，把数据划分成很多小块，然后每个处理器对每块数据进行处理，构造一个局部的分类决策树，然后用这些子分类器对新的样本进行预测，最后对这些预测结果进行加权集成。还有另外一种方式是每计算一个节点的分裂属性时都要做一次同步，具体方法为：每个处理器对所分配的数据都计算一个局部最佳分裂属性，然后再同步得到全局最佳分裂属性，这样不断递归，最后得到最终的决策分类树。[张，2010]提出了一种基于 MapReduce 的 SPRINT 并行分类算法，不过其最终得到的模型并不是全局分类模型，而是局部最优模型，且仍然采用递归的方式（迭代次数不可控），即只实现了节点内部并行，没有实现同一层节点之间的并行。

[0009] 因此，现有的分类决策树很多实现是串行且基于内存，因此不能处理海量数据；而对于现有的分布式处理方式，虽然数据处理规模有了很大的提高，但是编程实现比较复杂和困难，另外构造的分类决策树不是全局的，只是很多局部子决策树的加权集成。更重要的是，递归实现效率也比较低，迭代过程不可控。

## 发明内容

[0010] 本发明的目的在于提供一种采用决策树的数据分类方法和系统。其实现了基于 MapReduce 的并行决策树 ID3 算法，不仅可以处理大规模数据集，而且并行效率高，即实现构建决策树中节点内部以及同一层节点之间的并行计算。

[0011] 为实现本发明的目的而提供的一种采用决策树的数据分类方法，所述方法，包括下列步骤：

[0012] 步骤 100，基于 MapReduce 机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树；

[0013] 步骤 200，根据所述决策树，对输入的数据记录进行分类。

[0014] 所述步骤 100，包括下列步骤：

[0015] 步骤 110，启动一个进程，计算训练数据中包含的每个属性的信息增益，选出最大值作为根节点的分裂属性，并计算决策规则以及传给第一层的前缀信息：

[0016] 步骤 120，判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，执行步骤 130；否则，执行步骤 130；

[0017] 步骤 130，判断是否产生新的前缀信息，若是，则执行步骤 140；否则执行步骤

160 ;

[0018] 步骤 140,决策树层数加一,判断当前决策树的层数是否小于训练数据中包含的所有属性的总数,若是,则执行步骤 150 ;否则执行步骤 160 ;

[0019] 步骤 150,启动一个新的进程,计算在当前前缀信息下,当前训练数据中包含的每个属性的信息增益,选出最大值作为当前节点的分裂属性,并计算决策规则以及传给下一层的前缀信息,返回步骤 120 ;

[0020] 步骤 160,结束训练,根据计算得到的决策规则构建决策树。

[0021] 在进行属性的信息增益计算时采用 MapReduce 函数采集数据,在 Map 函数中,根据头文件信息对读入的每一行样本进行解析,产生中间的 <key, value> 对,key 为前缀信息 + 类别信息 + 条件属性的名字 + 条件属性的值或者前缀信息 + 类别信息,若没有前缀信息,则为空,value 为 1,Map 函数的输入 key 和 value 分别为样本的在 dfs 上偏移位置和样本本身,Reduce 函数对中间 <key, value> 对进行融合。

[0022] 所述步骤 200 中,构建完成的决策树保存在 HDFS 文件系统中,当进行测试时,每个节点从 HDFS 中提取决策树,对输入的数据记录进行分类。

[0023] 为实现本发明的目的还提供一种采用决策树的数据分类系统,所述系统,包括 :

[0024] 训练模块,用于基于 MapReduce 机制,并行计算训练数据中包含的每个属性的信息增益,选出最佳的分裂决策属性作为节点构造决策树 ;

[0025] 分类模块,用于根据所述决策树,对输入的数据记录进行分类。

[0026] 所述训练模块,包括 :

[0027] 信息增益计算模块,用于计算在当前前缀信息下,当前训练数据中包含的每个属性的信息增益,选出最大值作为当前节点的分裂属性,并计算决策规则以及传给下一层的前缀信息 :

[0028] 决策规则判断模块,用于判断是否产生了新的决策规则,若是,则将产生的新的决策规则保存到规则集中,同时删除当前训练数据中包含该规则的样本,产生新的数据集,触发前缀信息判断模块 ;否则,直接触发前缀信息判断模块 ;

[0029] 前缀信息判断模块,用于判断是否产生新的前缀信息,若是,则触发阈值判断模块 ;否则结束训练,根据计算得到的决策规则构建决策树 ;

[0030] 阈值判断模块,决策树层数加一,判断当前决策树的层数是否小于训练数据中包含的所有属性的总数,若是,则触发信息增益计算模块 ;否则结束训练,根据计算得到的决策规则构建决策树。

[0031] 所述训练模块中,在进行属性的信息增益计算时采用 MapReduce 函数采集数据,在 Map 函数中,根据头文件信息对读入的每一行样本进行解析,产生中间的 <key, value> 对,key 为前缀信息 + 类别信息 + 条件属性的名字 + 条件属性的值或者前缀信息 + 类别信息,若没有前缀信息,则为空,value 为 1,Map 函数的输入 key 和 value 分别为样本的在 dfs 上偏移位置和样本本身,Reduce 函数对中间 <key, value> 对进行融合。

[0032] 所述分类模块,控制每个节点将保存在 HDFS 文件系统中的构建完成的决策树提取出来,对输入的数据记录进行分类。

[0033] 本发明的有益效果是 :

[0034] 本发明实现了基于 MapReduce 的并行决策树算法,解决了传统串行决策树分类算

法不能处理的大规模数据的问题。实现了决策树算法 ID3 的完全并行，即不仅仅是同一节点最佳属性的选择是并行的，而且同一层的所有节点计算最佳属性也是并行的；更重要的是，实现了用循环的方式替换了递归，循环的次数可控，算法最大迭代次数不会超过数据的条件属性的个数。

## 附图说明

[0035] 图 1 是现有技术中分布式构造决策树的工作示意图；

[0036] 图 2 是本发明的一种采用决策树的数据分类方法的步骤流程图；

[0037] 图 3 是本发明中根据训练数据训练构造决策树的步骤流程图；

[0038] 图 4 是本发明的一种采用决策树的数据分类系统的结构示意图；

[0039] 图 5 是根据上述产生规则构造的决策树。

## 具体实施方式

[0040] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明的一种采用决策树的数据分类方法和系统进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0041] 本发明的一种采用决策树的数据分类方法和系统，通过对训练数据，基于 MapReduce 机制构造决策树，树的中间节点是分裂决策属性，树的叶结点都有类别标记，因此从根节点到叶节点就构成了一条判别规则。当构造完分类决策树后，就可以对测试样本进行分类。

[0042] 本发明中采用的并行分类决策树算法主要针对分类决策树，而且属性都是类别属性。在大规模数据处理中，连续型属性的处理代价相当高，所以可以把连续型属性通过预处理转化为类别型属性。

[0043] 下面结合上述目标详细介绍本发明的一种采用决策树的数据分类方法，图 2 是本发明的一种采用决策树的数据分类方法的步骤流程图，如图 2 所示，所述方法，包括下列步骤：

[0044] 步骤 100，计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性构造决策树；

[0045] 图 3 是本发明中根据训练数据训练构造决策树的步骤流程图，如图 3 所示，所述步骤 100，包括下列步骤：

[0046] 步骤 110，启动一个 job：计算训练数据中包含的每个属性的信息增益，选出最大值作为根节点的分裂属性并计算决策规则以及传给第一层的前缀信息：

[0047] 步骤 120，判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，执行步骤 130；否则，执行步骤 130；

[0048] 每一层计算完成后，我们都会检查是否产生新的规则，如果产生新的规则就保存到规则集中，产生一个新的数据集，该数据集由原始数据集中去掉能被新规则包含的样本得到。这样下一层操作的数据对象是产生的新数据集，数据集会变得越来越小，最后直到没有新的数据集产生为止，算法结束。

[0049] 步骤 130，判断是否产生新的前缀信息，若是，则执行步骤 140 ；否则执行步骤 160 ；

[0050] 由于层与层之间的运算是串行的，上一层都会传递前缀信息给下一层节点，这些前缀信息包括从根节点到当前分支的分裂属性信息等。每一层要选择的分裂属性个数是由前缀信息决定的，每个前缀信息代表一个节点。

[0051] 步骤 140，决策树层数加一，判断当前决策树的层数是否小于训练数据中包含的所有属性的总数，若是，则执行步骤 150 ；否则执行步骤 160 ；

[0052] 本发明实现的并行决策树算法以循环代替了递归，每一层只需要一个 job 就可以完成，而不关心一个层里面有多少个节点。使得运行完程序所需要的最大 job 个数可预测（最大数目为样本集中条件属性的数目 numAttri），从而有利于控制程序的执行状态。而在递归中，无法预测还有多少节点要运算，这样就无法预测程序何时结束。而本发明中程序需要运行的最大层数由条件属性的个数决定，因此是可控制的。也就是可以设置程序中循环最大的迭代次数为 numAttri，而不是递归方式，无法估计程序何时执行结束。

[0053] 步骤 150，启动一个新的 job ：计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息 ：返回步骤 120 ；

[0054] 步骤 160，结束训练，根据计算得到的决策规则构建决策树。

[0055] 作为一种可实施方式，构建完成的决策树模型保存在 HDFS 文件系统中。当进行测试时，每个节点从 HDFS(HDFS 本身是一个分布式文件系统）中提取决策树模型，对测试样本进行分类。

[0056] 较佳地，在训练过程中，基于 MapReduce 机制，本发明实现了同一层内节点之间、节点内的并行计算，提高算法的执行效率。在进行属性的信息增益计算时采用 MapReduce 函数采集数据，在 Map 函数中，根据头文件信息对读入的每一行样本进行解析，产生中间的 <key，value> 对，key 为前缀信息 + 类别信息 + 条件属性的名字 + 条件属性的值或者前缀信息 + 类别信息（如果没有前缀信息，则为空），value 为 1。Map 函数的输入 key 和 value 分别为样本的在 dfs 上偏移位置和样本本身。Reduce 函数对中间 <key，value> 对进行融合，即对具有相同 key 的 value 进行统计。输入的 key 和 value 是 Map 函数的输出，输出的 key 与 Map 函数相同，value 为统计值。

[0057] MapReduce 是 Google 实验室提出的一种简化的分布式程序设计模型，用于处理和生成大量数据集。通过该模型，程序自动分布到一个由普通机器组成的超大机群上并发执行。[Dean J，Ghemawat S. ：MapReduce ：Simplified data processing on large clusters [J]. Communications of The ACM，vol ：51. pp ：107-113，2008. ]

[0058] Map 和 Reduce 是该模型中的两大基本操作。其中，Map 是把一组数据一对一的映射为另外的一组数据，Reduce 是对数据进行规约，映射规则与规约规则可由用户通过函数来分别指定。现实生活中很多任务的实现都是可以基于这样的映射规约模式。

[0059] 实现 MapReduce 操作的整个流程如下 ：

[0060] (1) 当用户程序调用 MapReduce 函数时，输入文件将被分隔成 M 块（每块大小通常为 16M 至 64M，可由用户通过参数控制），同时用户程序将被复制许多份存储于机群中。

[0061] (2) 机群中的某个节点将被视为主节点。主节点选择空闲节点，并指派给每个节

点一个 Map 任务或一个 Reduce 任务。最终将有 M 个 Map 任务和 R 个 Reduce 任务被指派完成。

[0062] (3) 被指派到 Map 任务的节点将读入所对应的输入文件块，输入的<key，value>对将通过用户自定义的 Map 函数得到一组同样用 <key，value> 对表示的中间结果集合。

[0063] (4) 中间结果对将会周期性的写入本地硬盘并通过分隔函数分隔成 R 个块。同时，存储中间结果的本地硬盘位置将会返回给主节点，以便其下一步指派 Reduce 任务。

[0064] (5) 被指派到 Reduce 任务的节点将远程读入中间结果。待中间结果数据全部被读入后，将会按照 key 来排序，使得具有同样 key 的中间结果聚集在一起。

[0065] (6) 含有相同 key 的 <key，value> 对通过用户自定义的 Reduce 函数进行归约，得到最终结果并写出输出文件。

[0066] MapReduce 通过把对数据集的大规模操作分发给网络上的每个节点来实现可靠性，每个节点会周期性的把完成的工作和状态信息返回给主节点。如果一个节点保持沉默超过一个预设的时间间隔，主节点就认为该节点失效了，并把分配给这个节点的数据发到别的节点，并且因此可以被其它节点所调度执行。

[0067] 由于 MapReduce 运行系统已考虑到了输入数据划分、节点失效处理、节点之间所需通信等各个细节，使得程序员可以不需要有什么并发处理或者分布式系统的经验，就可以处理超大规模的分布式系统资源。

[0068] 步骤 200，根据所述决策树，对输入的数据记录进行分类。

[0069] 相应于本发明的一种采用决策树的数据分类方法，还提供一种采用决策树的数据分类系统，图 4 是本发明的一种采用决策树的数据分类系统的结构示意图，如图 4 所示，所述系统，包括 ：

[0070] 训练模块 1，用于基于 MapReduce 机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树 ；

[0071] 分类模块 2，用于根据所述决策树，对输入的数据记录进行分类。其中，所述训练模块 1，包括 ：

[0072] 信息增益计算模块 11，用于计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息 ：

[0073] 决策规则判断模块 12，用于判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，触发前缀信息判断模块 13 ；否则，触发前缀信息判断模块 13 ；

[0074] 前缀信息判断模块 13，用于判断是否产生新的前缀信息，若是，则触发阈值判断模块 14 ；否则结束训练，根据计算得到的决策规则构建决策树 ；

[0075] 阈值判断模块 14，决策树层数加一，判断当前决策树的层数是否小于训练数据中包含的所有属性的总数，若是，则触发信息增益计算模块 11 ；否则结束训练，根据计算得到的决策规则构建决策树。

[0076] 对于输入的一段数据记录，启动第一个 job，信息增益计算模块 11 计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息 ；由于是第一个 job，因此还没有产生前

缀信息，当前训练数据即是输入的数据记录，当前节点即是根节点，传给下一层的前缀信息即是传给第一层的前缀信息。决策规则判断模块 12 判断是否产生了新的决策规则，若是，则将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集，触发前缀信息判断模块 13；如果没有产生新的决策规则，则直接触发前缀信息判断模块 13；所述前缀信息判断模块 13 判断是否产生新的前缀信息，若是，则触发阈值判断模块 14；决策树层数加一，阈值判断模块 14 判断当前决策树的层数是否小于训练数据中包含的所有属性的总数，若是，则启动一个新的 job，再次触发信息增益计算模块 11，计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息；如此反复执行直到不再产生新的前缀信息或者决策树的层数大于训练数据中包含的所有属性的总数，则结束训练，根据计算得到的决策规则构建决策树。

[0077] 为了更清楚地说明本发明的技术方案，下面以 weka3.5 中的数据 weather.nominal 为例来讲解构建决策树的过程。算法主要是针对结构化的 arff 格式的数据，arff 格式的数据说明如下：

[0078] 数据的名称，以 @relation 开头；

[0079] 数据中包含的样本整数，以 @totalnum 开头，如果样本数未知则设为 -1；

[0080] 数据中的属性，包括 nominal 和 numeric，都是以 @attribute 开头；

[0081] 文件中的数据以 @data 开头

[0082] weather.nominal 数据如下：

[0083] @relation weather.symbolic

[0084] @attribute outlook{sunny,overcast,rainy}

[0085] @attribute temperature{hot,mild,cool}

[0086] @attribute humidity{high,normal}

[0087] @attribute windy{TRUE,FALSE}

[0088] @attribute play{yes,no}

[0089] @data

[0090] sunny,hot,high,FALSE,no

[0091] sunny,hot,high,TRUE,no

[0092] overcast,hot,high,FALSE,yes

[0093] rainy,mild,high,FALSE,yes

[0094] rainy,cool,normal,FALSE,yes

[0095] rainy,cool,normal,TRUE,no

[0096] overcast,cool,normal,TRUE,yes

[0097] sunny,mild,high,FALSE,no

[0098] sunny,cool,normal,FALSE,yes

[0099] rainy,mild,normal,FALSE,yes

[0100] sunny,mild,normal,TRUE,yes

[0101] overcast,mild,high,TRUE,yes

[0102] overcast,hot,normal,FALSE,yes

[0103] rainy,mild,high,TRUE,no

[0104] 总共有 5 个属性,包括 4 个条件属性和 1 个决策属性。每个属性都是类别型的,包括有若干个属性值。

[0105] 1. 第一个 job 的输出如下 :

[0106] no5

[0107] no,humidity,high 4

[0108] no,humidity,normal 1

[0109] no,outlook,rainy 2

[0110] no,outlook,sunny 3

[0111] no,temperature,cool 1

[0112] no,temperature,hot 2

[0113] no,temperature,mild 2

[0114] no,windy,FALSE 2

[0115] no,windy,TRUE 3

[0116] yes 9

[0117] yes,humidity,high 3

[0118] yes,humidity,normal 6

[0119] yes,outlook,overcast 4

[0120] yes,outlook,rainy 3

[0121] yes,outlook,sunny 2

[0122] yes,temperature,cool 3

[0123] yes,temperature,hot 2

[0124] yes,temperature,mild 4

[0125] yes,windy,FALSE 6

[0126] yes,windy,TRUE 3

[0127] 通过计算各个属性的信息增益后,选出根节点的分裂属性为 outlook,然后产生一条新的规则 :outlook = overcast yes(confidence = 1.0),以及 2 条的前缀信息 :

[0128] outlook,sunny 2,3

[0129] outlook,rainy 3,2

[0130] 其中 outlook 是属性名称,sunny 是属性值,后面的两个数字 2 和 3 分别指 outlook = sunny 的分支中属于 yes 和 no 的样本数。

[0131] 2. 第二个 job 的工作

[0132] 由于产生了新的规则,因此要删除原始训练集中包含该规则的样本,产生新的数据集如下 :

[0133] rainy,cool,normal,FALSE,yes

[0134] rainy,cool,normal,TRUE,no

[0135] rainy,mild,high,FALSE,yes

[0136] rainy,mild,high,TRUE,no

[0137] rainy,mild,normal,FALSE,yes

[0138] sunny,cool,normal,FALSE,yes

[0139] sunny,hot,high,FALSE,no

[0140] sunny,hot,high,TRUE,no

[0141] sunny,mild,high,FALSE,no

[0142] sunny,mild,normal,TRUE,yes

[0143] 3. 第三个 job 的工作

[0144] 第三个 job 的任务跟第一个 job 类似,但是由于已经产生了前缀信息,所以在最后的 reduce 输出中都有前缀,结果如下 :

[0145] outlook,rainy,no,temperature,cool 1

[0146] outlook,rainy,no,temperature,mild 1

[0147] outlook,rainy,no,humidity,high 1

[0148] outlook,rainy,no,humidity,normal 1

[0149] outlook,rainy,no,windy,TRUE 2

[0150] outlook,rainy,yes,temperature,cool 1

[0151] outlook,rainy,yes,temperature,mild 2

[0152] outlook,rainy,yes,humidity,high 1

[0153] outlook,rainy,yes,humi dity,normal 2

[0154] outlook,rainy,yes,windy,FALSE 3

[0155] outlook,sunny,no,temperature,hot 2

[0156] outlook,sunny,no,temperature,mild 1

[0157] outlook,sunny,no,humidity,high 3

[0158] outlook,sunny,no,windy,FALSE 2

[0159] outlook,sunny,no,windy,TRUE 1

[0160] outlook,sunny,yes,temperature,cool 1

[0161] outlook,sunny,yes,temperature,mild 1

[0162] outlook,sunny,yes,humidity,normal 2

[0163] outlook,sunny,yes,windy,FALSE 1

[0164] outlook,sunny,yes,windy,TRUE 1

[0165] 我们可以看到输出的每一条记录都会有一个前缀实际上每个前缀就对应着要找出一个分裂属性,因此这些输出中我们可以计算得到两个分裂属性。通过计算分类属性,我得到了以下 4 条规则 :

[0166] outlook = sunny,humidity = high no(confidence = 1.0)

[0167] outlook = sunny,humidity = normal yes(confidence = 1.0)

[0168] outlook = rainy,windy = TRUE no(confidence = 1.0)

[0169] outlook = rainy,windy = FALSE yes(confidence = 1.0)

[0170] 由于没有产生新的前缀信息,所以训练结束,最后得到分类决策树模型。

[0171] 图 5 是根据上述产生规则构造的决策树,如图 5 所示,最后产生的决策规则如下,其中参数 confidence 为 0.75,minNumObj 为 2。

[0172] 输出的规则 :

[0173] outlook = sunny

[0174] |humidity = high :no(3.0)

[0175] |humidity = normal :yes(2.0)

[0176] outlook = overcast :yes(4.0)

[0177] outlook = rainy

[0178] |windy = TRUE :no(2.0)

[0179] |windy = FALSE :yes(3.0)

[0180] 本发明的有益效果在于：

[0181] 1. 设计实现了并行决策树 ID3 算法，解决了分类决策树处理大规模数据的问题；

[0182] 2. 实现了算法的完全并行；即不仅仅是同一节点最佳属性的选择是并行的，而且同一层的所有节点计算最佳属性也是并行的；更重要的是，实现了用循环的方式替换了递归，循环的次数可控，最大次数不会超过数据的条件属性的个数；

[0183] 3. 实现的算法构造的分类决策树模型是全局的；解决了分布式算法下求解全局最优解比较难的问题；

[0184] 4. 基于 MapReduce 的并行编程机制；使得代码易于实现和理解，且能够很容易扩展的大规模集群中运行，提高算法的执行效率，减少执行时间。

[0185] 通过结合附图对本发明具体实施例的描述，本发明的其它方面及特征对本领域的技术人员而言是显而易见的。

[0186] 以上对本发明的具体实施例进行了描述和说明，这些实施例应被认为其只是示例性的，并不用于对本发明进行限制，本发明应根据所附的权利要求进行解释。

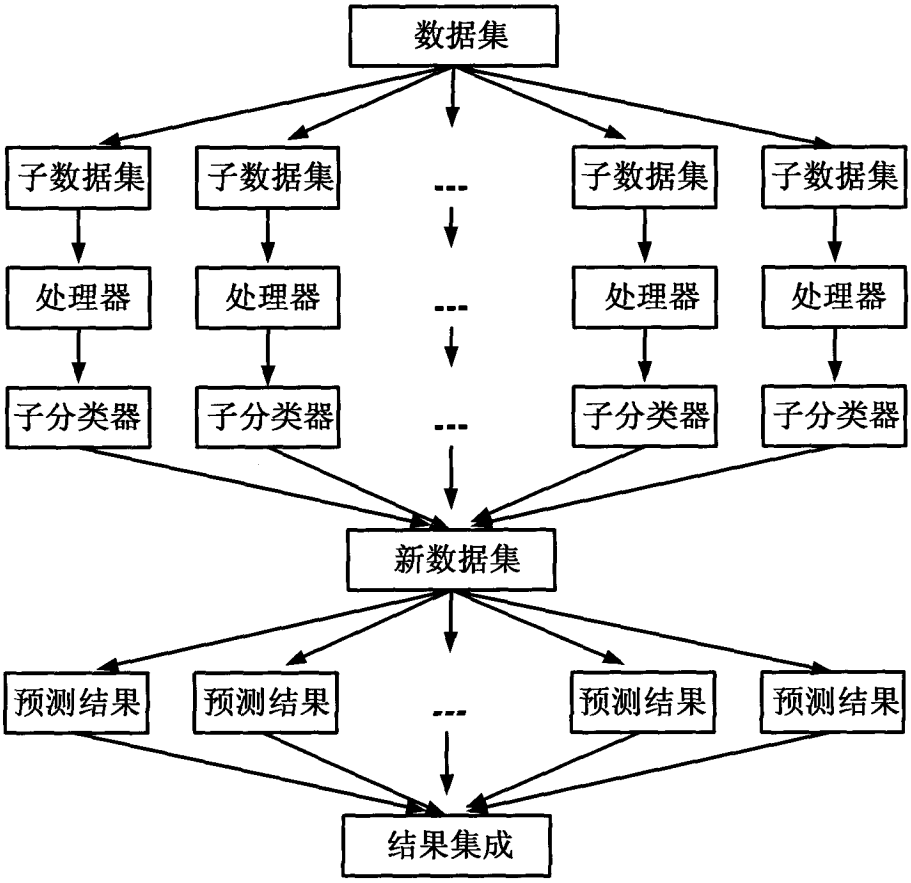

数据集
子数据集
子数据集
...
子数据集
子数据集
处理器
处理器
...
处理器
处理器
子分类器
子分类器
...
子分类器
子分类器
新数据集
预测结果
预测结果
...
预测结果
预测结果
结果集成

图 1

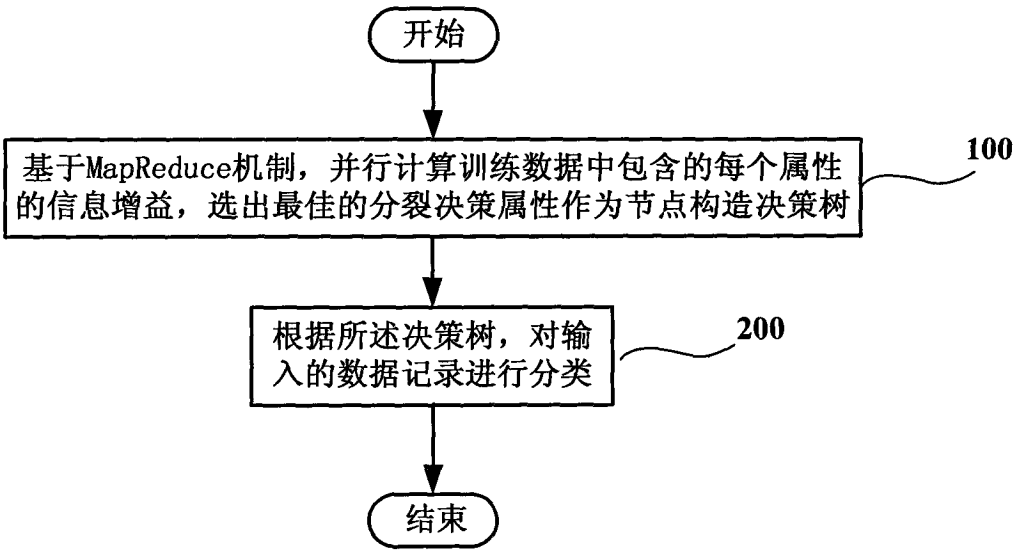

开始
基于MapReduce机制，并行计算训练数据中包含的每个属性的信息增益，选出最佳的分裂决策属性作为节点构造决策树
100
根据所述决策树，对输入的数据记录进行分类
200
结束

图 2

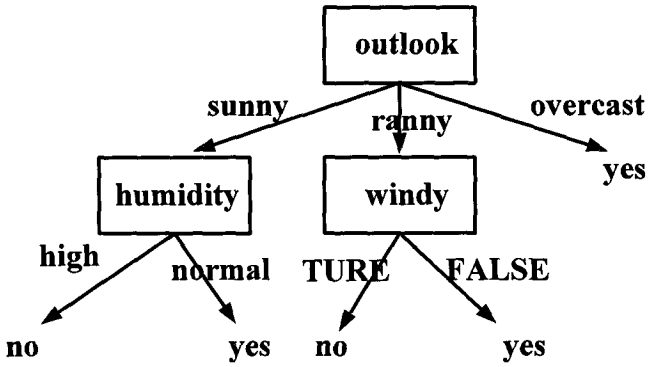

outlook
sunny
ranny
overcast
yes
humidity
windy
high
normal
TURE
FALSE
no
yes
no
yes

图 5

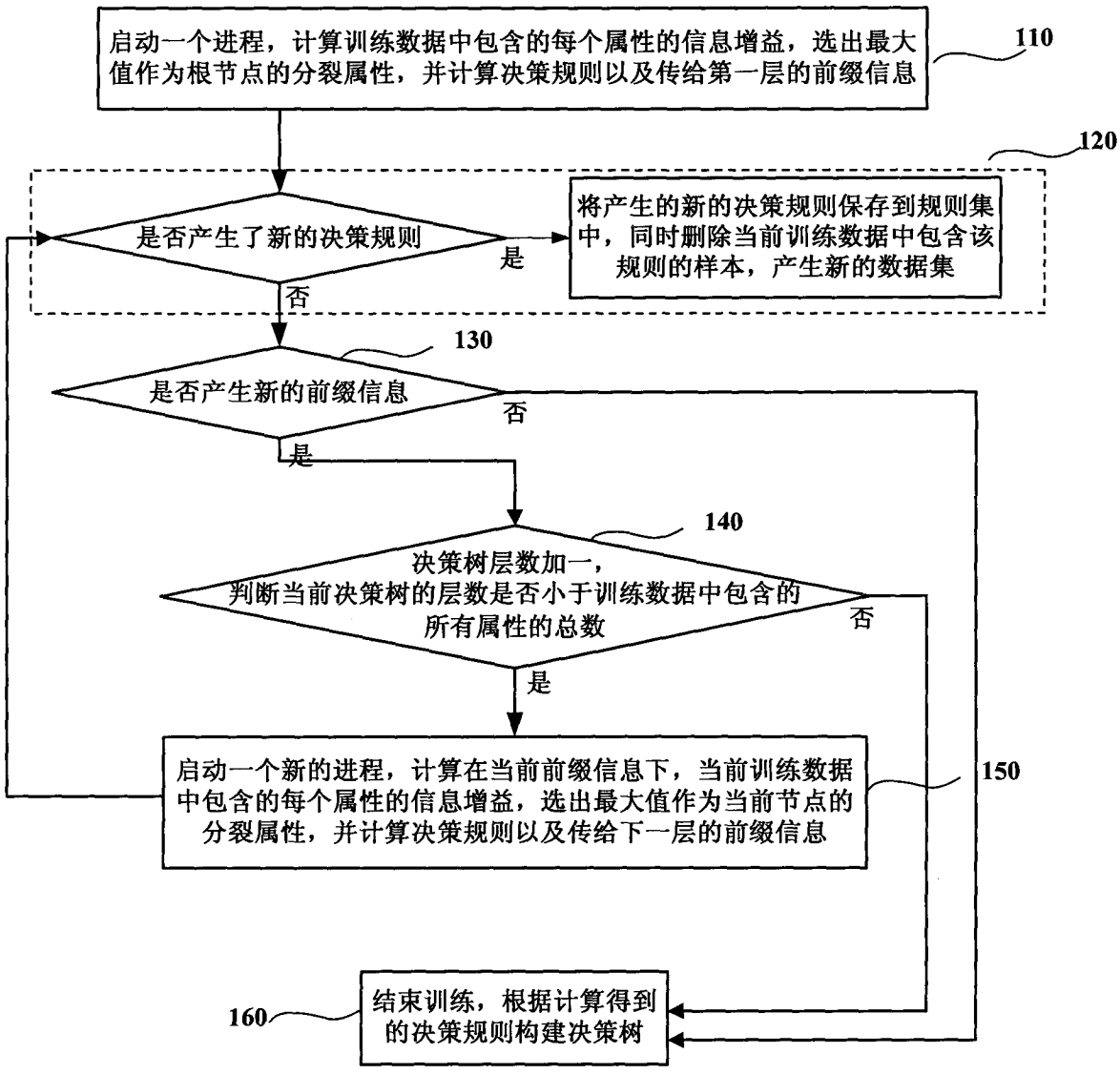

启动一个进程，计算训练数据中包含的每个属性的信息增益，选出最大值作为根节点的分裂属性，并计算决策规则以及传给第一层的前缀信息
110
120
是否产生了新的决策规则
是
将产生的新的决策规则保存到规则集中，同时删除当前训练数据中包含该规则的样本，产生新的数据集
否
130
是否产生新的前缀信息
否
是
140
决策树层数加一，
判断当前决策树的层数是否小于训练数据中包含的所有属性的总数
否
是
启动一个新的进程，计算在当前前缀信息下，当前训练数据中包含的每个属性的信息增益，选出最大值作为当前节点的分裂属性，并计算决策规则以及传给下一层的前缀信息
150
160
结束训练，根据计算得到的决策规则构建决策树

图 3

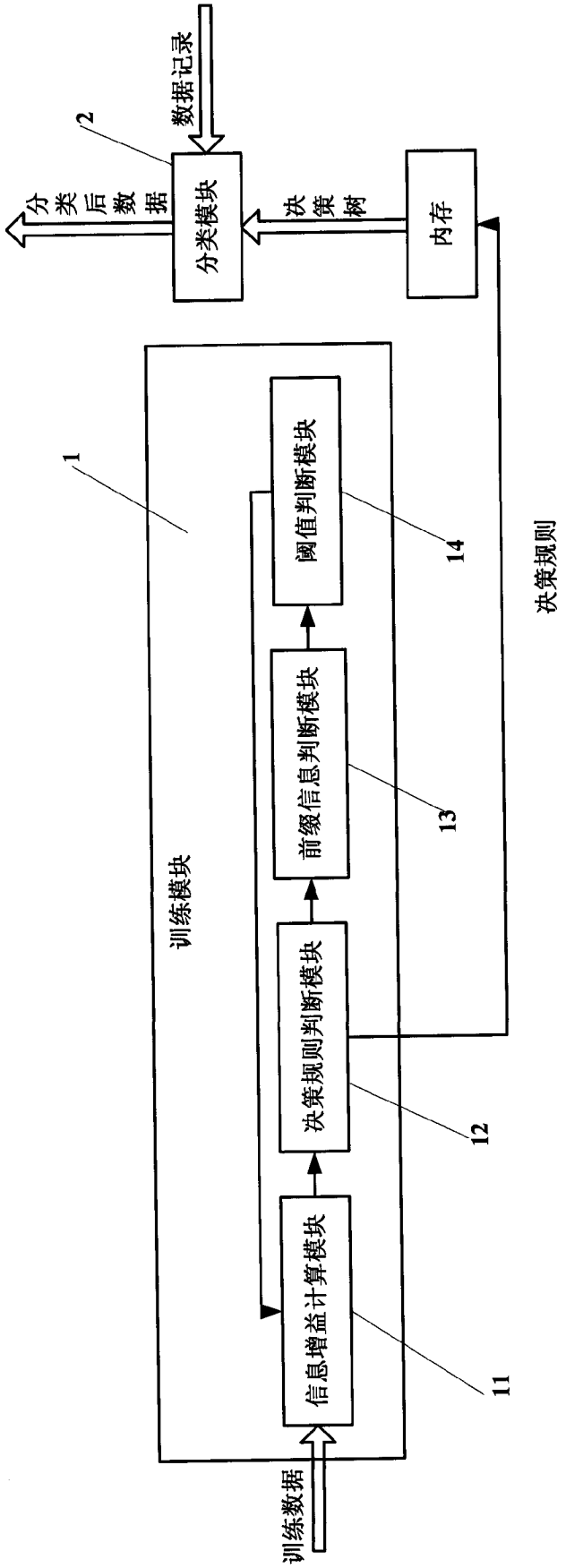

数据记录
2
分类后数据
分类模块
决策树
内存
1
阈值判断模块
14
前缀信息判断模块
13
决策规则判断模块
12
信息增益计算模块
11
训练模块
训练数据
决策规则

图 4