DOI:10.13718/j.cnki.xsxb.2019.11.017

噪声大数据的 MapReduce 高度随机模糊森林算法①

王　梅[1]，　雒　芬[2]，　张保华[3]

1. 常州工程职业技术学院 实验实训教学部，江苏 常州 213164；
2. 河南理工大学 计算机科学与技术学院，河南 焦作 454000；
3. 常州工程职业技术学院 智能装备与信息学院，江苏 常州 213164

摘要：为解决日趋增长的噪声大数据分类问题，提出了一种高度随机模糊森林算法. 该算法在决策树学习中生成连续属性的模糊分区，并给出在 MapReduce 框架中所提算法的分布式实现，用于受属性噪声污染的大数据集中学习模糊决策树的集合，该分布式实现模型可以适应计算的有效分配策略，从而产生良好的可扩展性数据，这种分布式算法使得模糊随机森林能够处理大数据集的学习和分类. 高度随机模糊森林算法能够实现噪声大数据的高精度分类，为以后的大数据分析打下良好的基础. 实验结果表明，所提算法比现有算法准确率更高，在属性噪声情况下，该文分类准确率也高于随机森林算法，说明该文算法的可行性和有效性.

关　键　词：随机森林；模糊决策树；高度随机模糊森林；噪声大数据

中图分类号：TP311　　**文献标志码**：A　　**文章编号**：1000－5471(2019)11－0110－08

随着互联网的发展，各种数据呈指数型增长，Web 上的数据量以 exabytes(10^{18})和 zettabytes(10^{21})为单位进行衡量，到 2025 年预测互联网的数据量将超过全世界人口的大脑容量，数据的快速增长是由于数字传感器、通信、计算和存储方面的进步造成了巨大的数据集[1-2]. 大数据的特征体现在 4V 方面，即体积、速度、品种和准确度，而提取有用的知识和信息变得越来越困难，因此对于大数据的分类分析越来越重要[3-5].

决策树是数据挖掘中使用最广泛的分类方法之一，能够达到良好的精度水平. 决策树的一个特殊优点是与归纳有关，这通常需要对有限数量的参数进行操作[5]. 处理不确定数据时引入模糊决策树，模糊决策树中的每个节点都是由模糊集来表征的，因此每个实例可以激活不同的分支并到达多个叶子[6]. 但是，决策树的缺点是对轻微噪声非常敏感，因此在噪声大数据中效果并不好.

目前，随机森林被认为是最有效和最流行的分类工具之一[7-8]. 随机森林的基本算法包含 2 个随机元素：bagging(即不同的数据集，从整个数据集中取而代之，用于学习每个不同的树)以及随机属性选择. 在森林中的每个树中选择可用属性的子集，并确定相对最佳分割[9]. 模糊随机森林具有更高的预测精度和更少的参数方差，比模糊规则的系统更准确，并且比清晰的随机森林更能容忍噪声[10].

现有普通分类方法有卷积神经网络、支持向量机、K－最近邻、直方图条件熵、堆叠自编码器和粒子群优化算法等[11-12]，但是这些方法对于大数据普适性并不好，需要利用分布式计算重新设计支持算法来处理这样具有挑战性的数据集. 在这种情况下，MapReduce 框架被证明是一个非常好的选择，但是其他分类算法在 MapReduce 框架中的分布式实现对大数据的噪声属性却没有较好的效果. 文献[13]中提出一种基于

① 收稿日期：2018－07－25
基金项目：河南省科技攻关计划项目(162102310090).
作者简介：王　梅(1974－)，女，实验师，主要从事实践教学管理与计算机应用研究.

改进 Tri-Training 算法的健康大数据分类模型，该方法通过更改扩充样本训练集选取方式，剔除可能提高分类误差的样本，解决了随机选取基础分类器的扩充训练样本集会引入噪声这一问题. 但是这种方法是避免训练过程引入噪声问题，对于含噪声的大数据处理效果并不好. 文献[14]提出一种新的分布式最近邻分类中的原型简化分类方法，旨在将原始训练数据集表示为减少的实例数，加快分类过程并降低最近邻规则的存储要求和对噪声的敏感性，但是该方法对噪声大数据的分类精度并不高.

现有数据分类方法具有无法适应大数据分类及噪声大数据分类精度不高的问题. 针对这些问题，本文提出了一种在 MapReduce 中实现的高度随机模糊森林算法，并给出了 MapReduce 下的分布式实现. 模糊随机森林的生成需要一个初步的 bagging 步骤，在整个学习数据集上应用采样，以便得到与森林中模糊树的数量一样多的单个学习数据集. 在 C4.5 决策树上构建模糊树的连续属性分区，并在节点创建时生成模糊分区. 该算法能够更好地分配计算工作量，优化并节省计算费用，同时不会破坏整体集合的准确性. 实验结果表明，在训练数据存在属性噪声的情况下，所提出的算法比现有方法更有效.

1　MapReduce 框架

Mapreduce 是一个分布式运算程序的编程框架，其核心功能是在 hadoop 集群中分布式并行运行用户的业务代码. Mapreduce 框架具有以下几个优点：①突破大数据硬件资源限制；②降低了集群上分布式运行的开发难度和复杂度；③将用户工作重点由计算复杂度转移至业务开发上. Mapreduce 程序完整运行需要 3 个进程，如表 1 所示.

表 1　Mapreduce 程序的 3 个进程及作用

进　程	作　用
MRAppMaster	完成整个 mapreduce 程序的调度与协调
MapTask	管理 map 阶段的整个数据处理
ReduceTask	管理 reduce 阶段的整个数据处理

MapReduce 分布式运行完成一个任务代码，需要经过 4 个部分，分别是程序启动、Maptask 进程、MRAppMaster 过程和 Reducetask 进程，其核心部分是 Maptask 进程和 Reducetask 进程. Maptask 过程如图 1 所示.

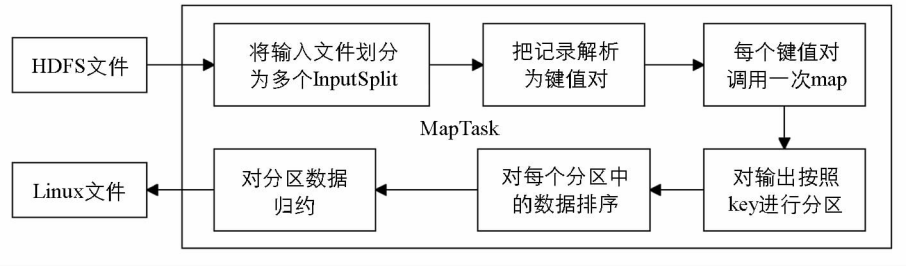

图 1　Maptask 过程图

由图 1 可知 Maptask 的过程为：经过对输入文件划分，得到大小固定的多个 InputSplit，解析 InputSplit 中的记录为键值对〈key，value〉，根据〈key，value〉对调用 map，结果输出 0 或多个〈key，value〉对. 对于输出的〈key，value〉对按照一定的 key 规则进行分区，得到分区的数量即为 Reducer 任务数量，之后对分区中〈key，value〉对进行排序，最终对数据进行归约处理，并将处理后的数据输出. 图 2 给给出了 Reducer 任务的过程图.

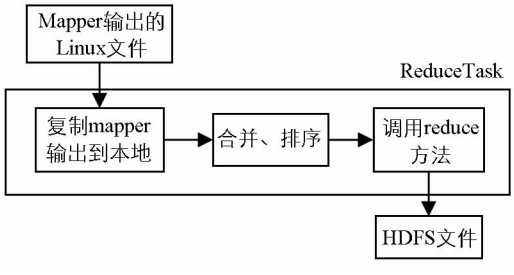

图 2　Reducetask 过程图

由图2可知Reducer任务的过程为：复制多个Mapper任务〈key，value〉对到Reducer本地，对所有复制〈key，value〉对数据进行合并与排序，之后每个〈key，value〉对调用reduce，将每次调用reduce的输出〈key，value〉对输出到HDFS文件中.

2 几种算法介绍

①用于生成模糊随机森林的模糊决策树学习算法，②高度随机模糊森林(highly random fuzzy forest，HRFF)算法，③MapReduce框架下HRFF算法的分布式实现.

2.1 单个模糊决策树

模糊决策树生成方法是基于最佳想法来选择最佳属性，在给定训练样例组每个节点处的分割. 为了测量一组例子中的杂质，首先需要确定模糊集的基数，清晰集的基数只是其元素的数量，模糊集 S 的基数是其中所有元素隶属度值的总和，可表示为

$$|S| = \sum_{e \in S} \mu_S(e) \tag{1}$$

其中，$\mu_S(e)$ 表示 e 到 S 的隶属度值. 通过上式可以来计算模糊集 S 的模糊熵.

$$H(S) = -\sum_c \frac{|S \to c|}{|S|} \log_2 \left(\frac{|S \to c|}{|S|} \right) \tag{2}$$

符号 $S \to c$ 表示包含 c 类所有样本集合 S 的子集，S 中的样本越不纯，熵值越高. 此外，根据该属性对样本进行排序，将属性的模糊信息增益定义为模糊熵的减少. 假设在属性 x 的模糊划分方面将集合 S 分成模糊子集 S_1，S_2，…，属性 x 相对于 S 的模糊信息增益可写为

$$G_f(x, S) = H_f(S) - \sum_i \frac{|S_i|}{|S|} H_f(S_i) \tag{3}$$

其中，$H_f(S)$ 表示 S 的模糊熵. 在每个子节点处，选择具有最大模糊信息增益的属性，然后在所有子节点上递归调用构建过程及其相应的数据，包含数据项和相关成员隶属度的列表. 根节点中的所有成员隶属度值都设置为1，这意味着根节点上的默认成员隶属函数始终返回1.

算法1中给出了构建单个模糊决策树的基本算法，构建模糊树过程通过调用find-best-attribute开始.

算法1：构建模糊决策树

```
struct NODE
    membership-f=lambda { return 1 }
    children=[ ]
    weights={ }
    function IS-LEAF(self)
        return size(self.children) == 0
    end function
end struct

function BUILD-FUZZY-TREE(node, data)
    if not is-terminal(node, data) then
        best-children←find-best-attribute(data)
        for child in best-children do
            child-data ← extract-child-data(child, data)
            build-fuzzy-tree(child, child-data)
            node.children.append(child)
        end for
    else
        UPDATE-WEIGHTS(node, data)
    end if
end function
```

本算法根据模糊信息增益确定了在当前节点划分数据的最佳属性. 评估遍历所有输入属性，并根据计算出的信息增益选择其中最好的属性.

当构建过程到达叶节点时，计算该节点的权重，可以将权重视为表示该节点中各种类的概率. 基于隶属度和分配给叶节点数据项的类来计算权重. c 类的权重是通过式(4) 获得的.

$$w_c = \frac{|S \rightarrow c|}{|S|} \tag{4}$$

使用构建模糊决策树来实现分类过程，算法 2 中给出了分类过程，提供给此过程的参数是分类已确定其成员隶属的新样本，由 data 的单个变量表示. 成员隶属度值设置为 1. total-weights 变量通过引用传递，应该包含过程结束时每个类的权重.

算法 2：模糊决策树分类

```
function CLASSIFY(data, total-weights, node=root)
  if node.is-leaf() then
    total-weights.add(t-norm(data.membership, node.weights))
  else
    for child in node.children do
      child-data ← data
      local-membership ← child.membership-f(data.item)
      child-data.membership←t-norm(data. membership, local-membership)
      CLASSIFY(child-data, total-weights, child)
    end for
  end if
end function
```

对于每个子节点，当前成员隶属度值使用子成员 membership-f 返回的值进行 t-normed，得到子节点的数据隶属度. 当执行分类时必须考虑所有子节点，而不仅仅是 $\mu \geqslant 0.5$ 的子节点，total-weights 变量也被传递给子节点以便更新.

通过树遍历一个新样本，直到抵达一个叶节点. 在叶节点上将当前隶属度通过 t-normed 赋给每个类权重，随后这些值被添加到总权重中的适当值. 通过这个过程，多叶节点可以在分类任务中被激活，每个叶节点的结果被添加到最终结果以决定不同类样本的隶属度.

2.2　高度随机的模糊森林算法

本文提出的高度随机模糊森林(HRFF)算法遵循随机森林算法的基本思想，使用随机选择的训练样本和特征来构建集合的基础模型. 这种随机数据选择有利于减轻大量数据项的计算负担，并避免高维中的特征选择问题.

HRFF 算法的特性在于引入输入属性模糊分区的随机化，这会影响模糊决策树构造的结果. 算法基本原则是：通过属性随机模糊划分学习的单个树不能表现出最佳性能，所有树共同补偿才能产生高的分类精度. 此外，采用随机分区的计算开销降低，这在大数据处理中尤为重要.

下面给出输入属性随机模糊划分的细节，由于分类特征的划分显而易见，本文只讨论数值属性的模糊划分，指定几个模糊集隶属函数来将属性的域划分为一组模糊区域. 这意味着一个样本可以遍历几个子节点，具体取决于它对模糊区域的成员隶属度值，选择梯形隶属函数，可以通过点 p 进行参数化.

在图 3 中，可以看到使用 $p = 4$，$q = 0.5$ 生成的隶属函数，属性的域为 $U = [0, 10]$，这里仅定义了两个隶属函数，以加快决策树构建和通过非常小的子节点的遍历过程. 用于确定梯形函数确切形状的点 a 和 b 被放置在各自段的特定百分比 $q \in [0, 1]$ 上，即 $a = p - q(p - U_{\min})$，$b = p + q(U_{\max} - p)$. 为了减少学习算法的运行时间，如果该样本具有相对于相关模糊区域 $\mu \geqslant 0.5$ 的隶属度值，只对子节点传递一个样本.

原则上，有可能通过寻找参数 p 的最佳值来优化模糊分区，但是当数据集很大时，会导致非常高的计算成本. 另一方面，随机森林的关键思想是即使单个决策树可能无法正确分类，基础模型的组合也会产生

统计上准确的结果. 同样，可以应用类似想法来支持在属性的模糊分区中随机选择参数 p 的值. 尽管在大多数情况下随机分区可能导致单个决策树的性能较差，但整体集合将通过对来自单个决策树的结果求平均来做出正确的决策. 因此，在本文 HRFF 算法中，学习集合中的每个单个模糊决策树的同时，将随机值分配给每个节点处的参数 p.

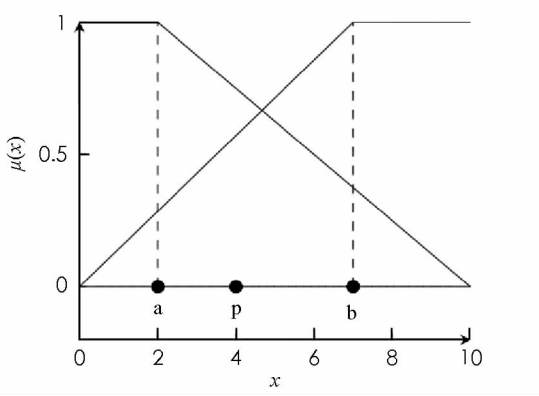

图 3 用 $p=4$，$q=0.5$ 生成的模糊集隶属函数

2.3 HRFF 的 MapReduce 分布式实现

将本文算法扩展到大数据集上，最广泛使用的方法是在计算机上分布执行，本文选择 MapReduce 作为执行此操作的框架.

首先描述标准分布方法，将整个数据集分成几部分，因为整个数据集太大而无法通过树学习程序进行处理. 用 R 表示数据部分的数量. 如果数据项的原始计数是 D，则每个 R 部分获得 $\frac{D}{R}$ 的数据项，然后利用每个 R 部分生成自己的随机森林. 另外一个参数是每个森林中的树木数量，本文用 V 表示. 所以，总的来说会有 RV 个树.

在算法 3 中给出了 mapper 和 reducer 函数. 首先，对输入数据进行了一些初步假设. 输入文件中的每一行都标记为训练数据或验证数据，本文指定如果行以字母 v 开头，则该行是验证行，否则默认为训练行. 本算法在建立随机模糊森林的同时对结果进行验证，这样做是为了每个工作者获得一些训练数据及所有验证数据，以便可以立即预测所有验证样本的类.

算法 3：用于 HRFF 算法的 Mapper 和 reducer 函数

```
function MAP(line):
  if is-training(line) then
    MAP-TO(random mod R, line)
  else
    for r in [0, R) do
      EMIT(r, line)
    end for
  end if
end function

function REDUCE(id, lines):
  validation-data ← [ ]
  training-data ← [ ]
  for line in lines do
    if is-training(line) then
      training-data.append(STRING-TO-DATA(line))
    else
      validation-data.append(STRING-TO-DATA(line))
    end if
  end for
  forest ← BUILD-HRFF(training-data)
  for v in validation-data do
    EMIT(v, forest.predict(line))
  end for
end function
```

算法 3 中显示的 mapper 逐行读取文件，如果遇到训练行则会发出一个在[0，R]范围内的随机 key，其中只有一个 reducer 得到该 key，如果 mapper 遇到验证行，则会向所有 reducers 发出 key. 然后，reducers

运行 reduce 函数，获取与其 id 相关的所有行，reducer 循环遍历与之关联的所有行，并将其放入 validation-data 和 training-data 组中. 使用 training-data，用 V 树构建随机森林，其中 V 是算法的参数，reducer 会预测每条线的隶属度.

由于每个 reducer 为每个数据项生成单个预测，因此对于每个单独的项总共有 R 个预测，为了能够组合这些预测，将使用此 reducer 作为下一步的 mapper，然后以数据项为 key 进行预测，能够将所有预测组合在一个数据项下. 算法 4 中给出了另一个 reducer 算法，能够将来自所有随机森林的预测组合成一个整体预测，这就为每个单独的类加上了预测，最后的决定是隶属度值最高的类.

算法 4：随机模糊森林分布式 reducer 函数

```
function reduce-results(item, predictions):
  prediction ← combine-predictions(predictions)
  if is-correct-prediction(item, prediction) then
    write-to-output(1)
  end if
end function
```

3　实验结果与分析

本文实验的大数据集及通过在其上运行所选算法获得的结果见表 2.

表 2　大规模数据集

数据集	特征数	项目数	属性值
KDDCup1999	4.8×10^6	41	18MB
POKER HAND	1×10^6	18	23MB
SUSY	5×10^6	10	880MB
HIGGS	11×10^6	48	2.5G
ARTIFICIAL-HIGGS	55×10^6	48	12.5G

表 2 给出了 5 个大数据集，所有这些数据集都来自 UCI 机器学习库. 表 2 中前 4 个数据集都是真实的数据集，最后一个是基于 HIGGS 数据集人工生成的，通过获取 HIGGS 数据集中的每个项目并生成 4 个新项目来生成此人工数据集.

通过运行模糊随机森林(Fuzzy Random Forest，FRF)和高度随机模糊森林 HRFF 算法，本文使用准确度来比较算法的性能. 实验将在数据中引入一些噪声，用一定的概率分布对其进行扰动. 扰动以这样的方式完成，即它增加或减去该值的某个百分比. 所以，以一定的概率应用变换 $(1+q)x\rightarrow x$，实数 q 在 $[-Q, Q]$ 范围内，其中 $Q\in[0, 1]$. 参数 Q 确定扰动的强度，在本文实验中将 Q 设置为扰动的概率(表 3).

表 3　大数据集的分类准确性结果

数据集	$Q=0$		$Q=0.4$	
	HRFF	FRF	HRFF	FRF
KDDCup1999	84.41	80.03	80.2	76.34
POKER HAND	93.12	88.49	92.87	86.49
SUSY	100	98.32	100	97.15
HIGGS	77.13	71.02	75.45	68.7
ARTIFICIAL-HIGGS	75.7	68.1	74.23	66

在表 3 中可以看到对于大数据集的分类结果. 由此可以得出，本文的算法优于 FRF，在 $Q=0.4$ 的干扰下，本文算法依然有较好的分类准确性. 为了证明本文算法的可扩展性，选择 SUSY 数据集样本进行实验. 理论上，本文算法应该与 MapReduce 可执行核的数量成线性关系.

表4 算法可扩展性分析

核的数量	运行时间/s
8	1500
12	1200
16	900
20	660
24	540

图4给出表4中数据的绘制图，可以得到本文算法几乎与节点的大小成线性关系，这表明本文算法适用于具有足够计算能力的更大数据集.

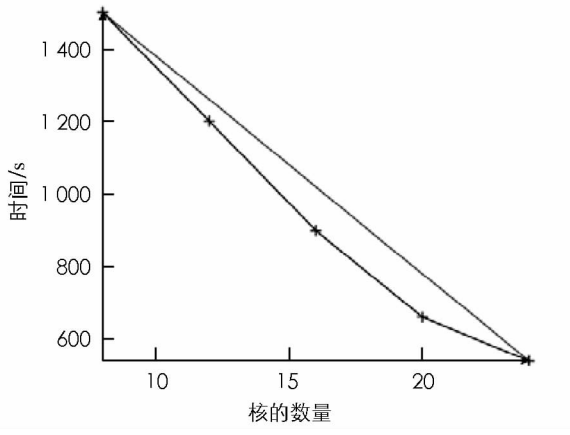

图4 算法可扩展性分析

4 结 论

本文提出了高度随机模糊森林算法来处理大数据分类问题，并给出了HRFF在MapReduce框架下的分布式实现. 该算法的基本思想是在分割数值属性时随机化模糊分区，且能够更好地分配计算工作量，允许模糊分区的随机性用于模糊决策树的输入变量. 实验结果表明，在具有属性噪声的情况下，本文HRFF算法比FRF算法分类更准确. 通过改变通常的数据分布过程来改进算法的并行化功能，HRFF也可以很好地适应并行处理器的数量. 未来工作将研究在噪声影响下，提高分类精度的方法.

参考文献：

[1] FERNÁNDEZ A, DEL RÍO S, LÓPEZ V, et al. Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce, and Programming Frameworks [J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2014, 4(5): 380-409.

[2] 梁吉业，冯晨娇，宋 鹏. 大数据相关分析综述 [J]. 计算机学报，2016，39(1)：1-18.

[3] WU X, ZHU X, WU G Q, et al. Data Mining with Big Data [J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(1): 97-107.

[4] GANDOMI A, HAIDER M. Beyond the Hype: Big Data Concepts, Methods, and Analytics [J]. International Journal of Information Management, 2015, 35(2): 137-144.

[5] RUTKOWSKI L, JAWORSKI M, PIETRUCZUK L, et al. Decision Trees for Mining Data Streams Based on the Gaussian Approximation [J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(1): 108-119.

[6] SEGATORI A, MARCELLONI F, PEDRYCZ W. On Distributed Fuzzy Decision Trees for Big Data [J]. IEEE Transactions on Fuzzy Systems, 2018, 26(1): 174-192.

[7] GENUER R, POGGI J M, TULEAU-MALOT C. VSURF: an R Package for Variable Selection Using Random Forests [J]. The R Journal, 2015, 7(2): 19-33.

[8] SCORNET E, BIAU G, VERT J P. Consistency of Random Forests [J]. The Annals of Statistics, 2015, 43(4): 1716-1741.

[9] RISTIN M, GUILLAUMIN M, GALL J, et al. Incremental Learning of Random Forests for Large-Scale Image Classification [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(3): 490-503.

[10] GADOMER L, SOSNOWSKI Z A. Fuzzy Random Forest with C-Fuzzy Decision Trees [C]//IFIP International Conference on Computer Information Systems and Industrial Management. Cham: Springer International Publishing, 2016.

[11] HAIXIANG G, YIJING L, YANAN L, et al. BPSO-Adaboost-KNN Ensemble Learning Algorithm for Multi-Class Imbalanced Data Classification [J]. Engineering Applications of Artificial Intelligence, 2016, 49: 176-193.

[12] 叶学义，宋倩倩，高　真，等. 基于直方图条件熵的水声数据分类算法 [J]. 计算机工程，2016，42(11)：244-248，254.
[13] 唐校辉，廖　欣，陈雷霆，等. 基于改进 Tri-Training 算法的健康大数据分类模型研究 [J]. 现代计算机(专业版)，2017(20)：21-25.
[14] TRIGUERO I，PERALTA D，BACARDIT J，et al. MRPR：A MapReduce Solution for Prototype Reduction in Big Data Classification [J]. neurocomputing，2015，150：331-345.

MapReduce Highly Random Fuzzy Forest Algorithm for Noisy Large Data

WANG　Mei[1]，　LUO　Fen[2]，　ZHANG Bao-hua[3]

1. *Experimental Training and Teaching Department，Changzhou Vocational Institute of Engineering，Changzhou Jiangsu 213164，China*；
2. *School of Computer Science and Technology，Henan Polytechnic University，Jiaozuo Henan 454000，China*；
3. *School of Intelligent Equipment and information，Changzhou Vocational Institute of Engineering，Changzhou Jiangsu 213164，China*

Abstract：In order to solve the problem of increasing noise big data classification，a highly random fuzzy forest algorithm has been proposed，which generates fuzzy partitions of continuous attributes in decision tree learning，and gives a distributed implementation of the proposed algorithm in MapReduce framework. Learning a set of fuzzy decision trees in a large data set contaminated by attribute noise，the distributed implementation model can adapt to the effective allocation strategy of the calculation，thereby generating good scalability data，and the distributed algorithm enables the fuzzy random forest to process learning and classification of big data sets. The highly random fuzzy forest algorithm can achieve high-precision classification of noisy big data，laying a good foundation for future big data analysis. The experimental results show that the proposed method has higher classification accuracy rate than the existing algorithm. In the case of attribute noise，the classification accuracy rate is higher than the random forest algorithm，which shows the feasibility and effectiveness of the proposed algorithm.

Key words：random forest，fuzzy decision tree，highly random fuzzy forest，noise big data

责任编辑　夏　娟