改进随机森林算法在 Spark+Kudu 平台的并行化运用

庄巧蕙 [1]
ZHUANG Qiaohui

摘 要 多种因素可能对学生成绩造成影响，利用数据挖掘工具对学生的学习课程成绩进行预测分析，进而利用预测分析结果及时指正学生出现的不良学习行为，同时检查老师的教学效果。首先将改进随机森林算法在大数据平台上进行并行化改进后进行实践。然后通过将提出的改进随机森林算法进行并行化，将其运用到 Spark+Kudu 大数据平台上进行仿真实验。最后算法并行化主要根据随机森林算法中的决策树划分策略以及模拟退火算法构建多种群策略来进行。实验结果证明并行化策略能够有效提升数据集的分类效率，大幅度缩短算法执行时间。

关键词 随机森林算法；Spark；Kudu；决策树；模拟退火算法

doi：10.3969/j.issn.1672-9528.2024.02.015

0 引言

如何利用现代化的技术对数据规模庞大教育数据资源进行处理，并准确地发现隐藏的知识，不是一件容易的事。而近年来数据挖掘技术的出现，为解决这类难题提供了强有力的技术支持。利用数据挖掘技术从大量数据中发掘出有价值的数据或是相关的知识规律信息，并将分析挖掘出来的结果应用到教学管理工作中，对提高学校教学质量以及教育管理水平都将有着深远的意义。

大数据平台建设已经成为当前时代的重要课题之一。如何将数据挖掘工作与大数据平台结合进行处理，也是数据挖掘工作面临的重大挑战之一。数据挖掘算法与大数据平台的结合，既可以充分利用大数据平台的海量数据处理优势，也可以根据更深层次挖掘海量数据后隐藏的规则或原则，从而更加深刻理解事物的本质。

本课题研究中充分利用《数据库原理及应用》课程的特点，并将学生学习的具体行为等特征作为数据挖掘研究的一个重要点进行。

1 研究现状

教育数据挖掘是指运用教育学、计算机科学、心理学和统计学等多个学科的理论和技术来解决教育研究与教学实践中的问题。在大数据时代背景下，教育数据挖掘研究将迎来新的转折点。教育数据挖掘涉及的工作层面比较多，包括学生成绩预测、教学不足分析、学生自适应学习能力分析、学生成绩自动判别等子领域，本文研究范围将限定在学生成绩预测方面。

针对学生成绩预测特定方向的教育数据挖掘发展大致可分为两个时期，第 1 个时期是 20 世纪 80 年代—20 世纪末，研究者开始将数据挖掘技术用于教育领域，但是研究方法比较简单，研究成果很少，受当时的技术水平的限制；第 2 个时期则是从本世纪初至今，该领域的研究方法与研究成果快速发展。进入 21 世纪以来，互联网的普及引发了教育技术的变革，采用的数据挖掘技术更加多样化。2012 年，美国教育部发布的蓝皮书《通过教育数据挖掘和学习分析促进教与学》标志着教育数据挖掘工作已受到广泛关注。美国高等教育信息化协会 EDUCAUSE 在新时代的教育挑战中提出了教育数据挖掘的定义，即使用数据和模型来预测学生学习进展和成果，并在此基础上采取行动的能力。

近年多位国内外研究人员采用数据挖掘技术进行课程成绩数据的研究分析。EI-Halees 利用数据挖掘方法来研究学生学习习惯，并对于改进学生成绩提出针对性建议。Ayesha 在学生测验成绩集基础上采用 K-means 聚类方法来预测学生课程了解程度，进而为最终考试题目构建提供数据依据。整体来看，课程数据挖掘研究吸引大量的学者进行探索 [1]。

2 相关技术

2.1 决策树算法

决策树算法是一种根据数据属性特征建立额分裂树，并且随着研究深入，逐步发展出 ID3、C4.5、CART（classification and regression tree）分类回归树等改进决策树，其具有决策结果易于理解、性能强大等优势。最大缺点是可能出现弱拟合等问题 [2]。

1. 闽南理工学院 福建石狮 362700

决策树算法是随机森林算法的基分类器，因此在进行随机森林算法介绍之前，首先对于决策树进行简要说明。决策树是一种比较直观易于理解的分类算法，能够对于高维数据进行快速处理。决策树算法不需要任何参数进行预先设定，从而保证算法不必基于任何先验假设运行 [3]。

朴素贝叶斯算法是根据统计学领域的先验概率来进行预测后验概率，其通常具有速度快并且准确率高等优势，但是经常出现属性之间存在依赖关系，导致分类准确率比较低的现象。朴素贝叶斯算法的性能可以与决策树算法相当 [4]。

支持向量机算法是通过构建超平面来进行高维、非线性的分类工作，其通常具有非常高的执行效率和分类准确率，但是其具有核函数构造复杂、数据敏感度缺失等缺点。

集成学习是通过组合多个分类器（随机森林算法、决策树算法、支持向量机算法）来进行投票分类，从而降低单个分类器的误差，提高整体的分类准确度。相关实验结果证实其具有更高的分类性能、稳定性和鲁棒性。

2.2 模拟退火算法

模拟退火算法的思想最早是由 Metropolis 提出的，并由 Kirkpatrick 应用到组合优化算法中，主要目标是解决优化过程陷入局部极小值，同时克服初值依赖性等问题。模拟退火算法主要模拟物理领域中的退火物理过程，是指将固体加热到足够高的温度，固体内部分子呈现出随机排列状态，然后逐步降温使其冷却，最终分子呈现出低能状态排列，固体达到某种稳定状态。模拟退火算法模仿退火物理过程，来解决优化问题中的全局最优解问题，从某一初始温度开始，随着温度的不断下降，结合概率突跳特性在解空间中随机寻找全局最优解。模拟退火算法之所以能够在优化问题中进行应用，主要是由于物质退火过程与组合优化问题之间具有比较高的相似性 [5]。

2.3 Spark 平台

近些年来，大数据平台逐渐火热，成为当前时代的研究与应用热点。Spark 平台作为 Apache 重点核心开源项目，已经逐步成为大数据平台的核心并行计算框架，目前已经在很多公司、研究机构、组织机构中进行实际应用。相比于传统 Hadoop 框架中的 MapReduce，Spark 框架充分利用内存计算以及数据整合处理，处理速度更快，这也是其能够得到广泛应用的原因之一。Spark 框架最新版本为 2.4.0。Spark 框架是基于 Scala 语言开发的，具有高并发模型、函数式编程、面向对象等特点，并使得 Spark 框架拥有更高的灵活性和性能。

Spark 框架采用 DAG（directed acyclic graph）有向无环图作为执行引擎，支持内存计算和循环数据流，从而具有执行速度快、运行高效等特点。Spark 框架具有多种语言接口，包括 Java、Scala、R 等语言，同时也支持 Spark Shell 交互式编程环境。Spark 框架支持多种底层数据存储文件系统，包括 Hadoop、Kudu、Hive、HBase 等。

Spark 生态圈包括多种组件，分别为 Spark SQL、Spark MLlib、Spark Streaming 以及 Spark GraphX。Spark SQL 采用分布式引擎，为用户提供方便的接口支撑 SQL 查询，目前主要用于交互式查询和批量数据流处理等应用场景。Spark MLlib 是基于 Spark Core 框架提供的机器学习算法库，使得用户可以基于海量数据进行一定的数据挖掘工作。Spark Streaming 作为一款流数据处理引擎，可以进行流式数据处理，具有高吞吐量以及高容错性等特点。Spark GraphX 可以用于进行图运算，充分融合数据并行以及图并行的优势，支撑大规模数据集的图计算 [6]。

Spark 整个体系结构在实现过程中充分借鉴了 Hadoop 平台的运行机制。在 Spark 框架运行过程中，主要有 Driver 和 Worker 两种角色，其中 Driver 核心在于是整个 Spark 集群的中心控制器，用于创建 SparkContext 对象，并完成执行任务的调度分发管理工作，Worker 主要利用 Executor 对象用于执行各个任务。通常在 Spark 集群中，Driver 作为主节点只有一个，而 Work 作为工作节点可以有很多个。除了上述两种角色之外，还需要有 Cluster Manager 资源管理器，主要负责集群上获取资源的外部服务，以及各种资源（CPU、内存）的分配与回收工作。为了完成各个任务，Worker 会从分布式文件系统获取执行计算所需的文件数据以及支撑配置文件，并将计算的 RDD 结果存储到 Cache 中，方便后续计算使用 [7]。

2.4 Kudu

Apache Kudu 是由 Cloudera 开源的分布式列式存储引擎，其弥补了传统 HDFS 和 HBase 的差距应用场景，具备更好的分析能力并能够应对快速的数据更新频率。Apache Kudu 可以支撑低延迟的随机读写以及高效的数据分析能力，其可以结合当前流程的各种大数据分析与查询工具，例如 Impala 工具、Spark 平台进行紧密结合。

对于大数据平台的存储技术，传统常用的技术主要有 HDFS 以及基于它发展而来的 Apache Parquet、Apache ORC 列式存储技术以及 Apache HBase、Cassandra 的 KV 类型半结构化存储技术。在企业应用场景中，通常使用 Parquent 进行静态数据分析，但是其不能更新数据且随机读写性能较差；而能够进行随机读写的 HBase 存储技术却并不能适用于进行数据分析，因此传统企业通常形成两套数据存储机制，即先进行 HBase 存储，然后定期进行 Parquent 数据同步并实时分析，造成资源浪费、分析时效性差、部分场景无法满足等问题，这也直接导致 Kudu 技术的诞生。

Kudu 工具可以和 Apache Hadoop 生态环境进行完美融合，充分利用各种通用性硬件进行横向扩展与部署，同时

支撑各种高可用性操作。Kudu 工具的典型优势包括：支持 OLAP 快速操作；无缝与 MapReduce、Spark 以及其他生态组件进行集成；与 Apache Impala 紧密结合；强大并松耦合的一致性模型；强大的顺序写入和随机读取性能；能够有效被 Cloudera 管理器进行管理。

从存储结构来看，Kudu 工具更像是一种结构化数据表存储系统，其中可以有任意数量的 table，并且需要为每个 table 预定义好相应的 schema。从底层实现角度来看，为了支撑大规模数据集扩展和集群建设，Kudu 将每个表划分为多个相似单元的 tablet，并且可以根据 hash 规则或者范围分区等策略进行配置，从而将上层分析负载进行均衡化并提供更为可靠的并发性。

在数据存储和支撑业务分析过程中，为了保证数据的始终安全性和可用性，Kudu 工具采用 Raft 一致性算法来复制针对每个 tablets 的操作行为。Raft 协议类似于 Paxos 协议确保每个写入操作在进行反馈客户端之前必须保证两个节点的写入成功，从而避免由某个机器宕机而导致的数据丢失问题。区别于最终一致性协议，Raft 一致性协议可以确保所有复制节点对于数据状态达到统一意见，同时借助于逻辑时钟和物理时钟融合技术可以提供严格镜像一致性。

3 改进算法流程

本文提出了一种综合优化的新算法，主要是将模拟退火算法融入到随机森林算法执行过程中，利用二进制编码、OOB 误差最小化、模拟退火操作，来获取最优的特征选择 O、决策树规模 K 和特征子集规模 N、决策树权重 W_t 的最佳组合取值[8]。

为了实现上述目标，设定改进随机森林算法的目标函数为：

$$f(K^*, O^*, \{Attribute_i | i = 1,2,...,M\}, \{w_j | j = 1,2,...,K\}) = \arg\min(\text{avg OOB error}) \quad (1)$$

优化变量：$K, O, \{Attribute_i | i = 1,2,...,M\}, \{w_j | j = 1,2,...,K\}$，其中 K, O 是实数，K 取值范围为 $[0,500]$，w_j 为取值范围为 $[0,15]$ 的整数。$Attribute_i$ 取值为 0 或 1，其中 0 表示该特征未被选择，而 1 则表示该特征已经被选中。优化变量可以采用二进制编码进行表示，主要由四个片段组成，如图 1 所示。

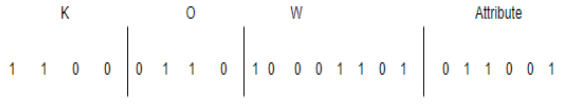

图 1 优化变量的二进制编码

在上述变量二进制编码的基础上，为了得到上述目标函数的最优解，设计的改进随机森林算法 IRFC 的流程如图 2 所示，其相应的流程如下。

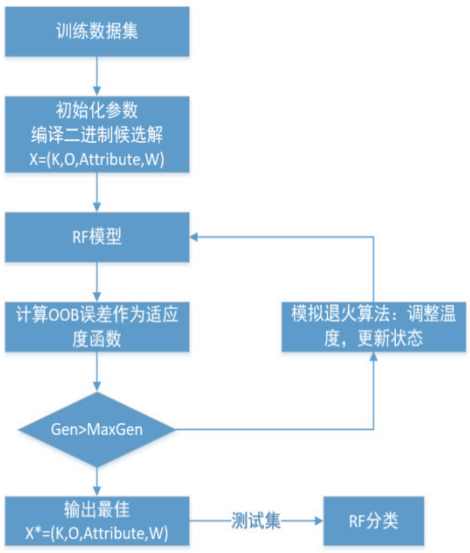

图 2 改进随机森林算法的算法整体流程图

第 1 步：设定初始化温度 $t=t_{max}$，迭代次数 $maxgen$，$k=1$，随机产生一个初始解 X_0，并令 $X_{best}=X_0$。

第 2 步：结合 RF 分类器，计算 $F=\max(1/f)$。

第 3 步：①若在此温度内达到内循环停止条件，则跳转到②，否则从当前最优解 X_{best} 的邻域 $N(X_{best})$ 中随机选择 X_{new}，并计算相关目标函数值 $f(X_{new})$，由此推算出相关增量值 $\Delta f=f(X_{new})-f(X_{best})$。按下面公式更新 X_{best}。

$$X_{best} = \begin{cases} X_{new}, & \Delta F < 0 \\ X_{new}, & \Delta F > 0 \text{ 且 } e^{\frac{-\Delta F}{t_k}} > \text{random}(0,1) \\ X_{best}, & \text{其他情况} \end{cases} \quad (2)$$

②设置 $t_{k+1}=d(t_k)$;$k=k+1$，若满足最终停止条件，则终止计算，输出最优解 X_{best}，否则返回①继续执行。

第 4 步：如果 $gen>maxgen$，否则返回第 2 步。

4 大数据平台搭建与算法并行化

4.1 大数据平台搭建

在本文搭建的大数据平台的基础架构中，Kudu 作为数据存储平台存储数据集，并为上层的 Spark 平台提供数据源。Spark 平台作为核心并行计算框架进行任务构建，并实现任务并行化处理。

本文搭建的大数据平台共有四台服务器组成，其中一台作为 Master（Driver）主节点，另外三台作为 Salve（Worker）从节点。Kudu 数据存储技术和 Spark 平台都是按照这种一主三从模式进行配置与安装。每个服务器的基本配置为 CentOS 6.8，Spark 版本为 2.2，内存为 16 GB，CPU 型号为 Intel Core i7 4700MQ。

在搭建 Spark 平台之前，需要首先进行 Kudu 数据存储软件的安装。详细的安装配置过程如下。

首先，配置 NTP 服务，在各个服务器配置 ntp，保证四

台服务器时间保证一致，处于同步状态。

其次，安装 Kudu-master。在 driver 节点通过 yum 方法远程安装 Kudu 软件包，并对 etc/kudu/conf/master.gflagfile 进行参数配置，并配置相关目录权限。

然后，安装 Kudu-tserver。在三个 worker 节点通过安装包进行解压并安装，同时也需要对 etc/kudu/conf/tserver.gflagfile 进行参数配置，并配置相关目录权限。

最后，启动服务。分别在 driver 节点和 tserver 节点启动 Kudu-master 和 Kudu-tserver 两个服务，启动后可以通过 driver 节点的 Web 界面观察 Kudu 各个节点的运行情况。

本文建设的 Spark 平台是基于 Kudu 存储文件系统建设的，因此安装流程是先建设 Kudu 存储平台再安装 Spark 平台。Spark 平台是基于 Scala 语言开发的，因此需要在默认安装 Spark 平台之前安装 Scala 运行环境。具体的安装环节如下。

首先，安装 Scala。下载 Scala 压缩包并进行解压，并需要在 /etc/profile 文件配置 Scala 环境变量，并通过重启或者 source 命令使得配置生效。

其次，Spark 安装，在 Driver 节点下载 Spark 2.2 安装包进行解压，并配置 Spark 环境变量并重启生效。

然后，进入 Spark 的 conf 配置文件夹中，复制 spark-env.sh.template 文件为 spark-env.sh 脚本文件，并需要在该文件中配置 Scala_home、Java_home、Spark_master_ip 等参数信息。另外还需要在 conf 文件夹里面修改 slaves 文件，添加 work 节点的 IP 地址或者机器名信息。

最后，启动 Spark。将 Driver 节点中 Spark 文件以及配置文件信息通过 SCP 方式分发到各个 Worker 节点，并赋予相关启动权限和设置环境变量。在 Driver 节点中执行 Spark-all.sh 命令启动整个 Spark 集群。另外可以通过 WebUI 界面查看集群的运行状态。

4.2 算法并行化

为了能够将改进随机森林算法在 Spark 集群中执行，需要对算法进行适当调整，从而能够在 Spark 平台高度并行化执行。改进随机森林算法的并行策略包括随机森林算法的并行策略以及模拟退火算法的并行策略两个策略。

随机森林算法的并行策略：将随机森林算法执行过程中的决策树构建均匀分配到集群各个节点中，从而使得决策树的构建可以在不同集群节点得以执行，从而实现决策树的执行并行化。在每个节点上，不同决策树的生长也可以并发执行 [9]。

模拟退火算法的并行策略：考虑利用多种集群优化机制来引入模拟退火算法，从而支撑并行化工作。在模拟退火算法初始期间，随机产生多个进化种群，并分别计算每个种群的目标函数值，产生每个种群的最优候选解，最后从所有候选解中再次选择最优解作为算法最终的全局最优解。每个种群都可以在独立 worker 节点上进行运算 [10]。

通过上述两个并行策略，可以将改进随机森林算法改造成并行算法，从而可以在 Spark 大数据并行计算平台上运行。

5 总结

本文将改进随机森林算法在大数据平台上进行并行化改进并进行实践。先将提出的改进随机森林算法进行并行化，再将其运用到 Spark+Kudu 大数据平台上进行仿真实验。算法并行化主要根据随机森林算法中的决策树划分策略以及模拟退火算法构建多种群策略来进行。实验结果证明并行化策略能够有效提升数据集的分类效率，同时也从侧面证实了本文选择的 Spark 并行计算框架以及 Kudu 存储平台是非常合适进行并行计算的。

参考文献：

[1] 罗可，林睦纲，郝东妹 . 数据挖掘中分类算法综述 [J]. 计算机工程 ,2005(1):3-5+11.

[2] 栾丽华，吉根林 . 决策树分类技术研究 [J]. 计算机工程，2004, 30(9):94-96.

[3] 张棪，曹健 . 面向大数据分析的决策树算法 [J]. 计算机科学，2016, 43(S1):374-379+383.

[4] 黄春华，陈忠伟，李石君 . 贝叶斯决策树方法在招生数据挖掘中的应用 [J]. 计算机技术与发展，2016, 26(4):114-118.

[5] 杨若黎，顾基发 . 一种高效的模拟退火全局优化算法 [J]. 系统工程理论与实践，1997, 17(5): 30-36.

[6] 王日升 . 基于 Spark 的一种改进的随机森林算法研究 [D]. 太原：太原理工大学，2017.

[7] 牛志华 . 基于 Spark 分布式平台的随机森林分类算法研究 [D]. 北京：中国民航大学，2017.

[8] 梁世磊 . 基于 Hadoop 平台的随机森林算法研究及图像分类系统实现 [D]. 厦门：厦门大学，2014.

[9] 钱雪忠，秦静，宋威 . 改进的并行随机森林算法及其包外估计 [J]. 计算机应用研究，2018, 35(6):57-60.

[10]CHEN J, LI K, TANG Z, et al. A parallel random forest algorithm for big data in a spark cloud computing environment[J]. IEEE transactions on parallel and distributed systems, 2017, 28(4): 919-933.

【作者简介】

庄巧蕙（1990—），女，福建泉州人，硕士，讲师，研究方向：电子商务、计算机科学与技术。

（收稿日期：2023-11-17）