基于 Spark 和三路交互信息的并行深度森林算法

毛伊敏[1,2]，周展[1]，陈志刚[3]

（1. 江西理工大学信息工程学院，江西 赣州 341000；2. 韶关学院信息工程学院，广东 韶关 512026；
3. 中南大学计算机学院，湖南 长沙 410083）

摘 要：针对并行深度森林在处理大数据时存在冗余及无关特征过多、类向量过长、模型收敛速度慢以及并行化训练效率低等问题，提出了基于 Spark 和三路交互信息的并行深度森林（PDF-STWII）算法。首先，提出基于特征交互的特征选择（FSFI）策略过滤原始特征，剔除无关及冗余特征；其次，提出多粒度向量消除（MGVE）策略，融合相似类向量，缩短类向量长度；再次，提出级联森林特征增强（CFFE）策略提高信息利用率，加快模型收敛速度；最后，结合 Spark 框架提出多级负载均衡（MLB）策略，通过自适应子森林划分和异构倾斜数据划分，提高并行化训练效率。实验结果表明，所提算法能显著提升模型分类效果，缩短并行化训练时间。

关键词：Spark 框架；并行深度森林算法；特征选择；多级负载均衡

中图分类号：TN92

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2023143

Parallel deep forest algorithm based on Spark and three-way interactive information

MAO Yimin[1,2], ZHOU Zhan[1], CHEN Zhigang[3]

1. School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China
2. College of Information Engineering, Shaoguan University, Shaoguan 512026, China
3. College of Computer Science and Engineering, Central South University, Changsha 410083, China

Abstract: To address issues such as excessive redundancy and irrelevant features, long class vectors, slow model convergence, and low efficiency of parallel training in parallel deep forests, a parallel deep forest algorithm based on Spark and three-way interactive information was proposed. Firstly, a feature selection based on feature interaction (FSFI) strategy was proposed to filter the original features and eliminate irrelevant and redundant features. Secondly, a multi-granularity vector elimination (MGVE) strategy was proposed, which fused similar class vectors and shortened the class vector length. Subsequently, the cascade forest feature enhancement (CFFE) strategy was proposed to improve the utilization of information and accelerate the convergence speed of the model. Finally, a multi-level load balancing (MLB) strategy was proposed, combined with the Spark framework, to improve the parallelization efficiency through adaptive sub-forest division and heterogeneous skew data partitioning. Experimental results demonstrate that the proposed algorithm significantly improves the model classification effect and reduces the parallelization training time.

Keywords: Spark framework, parallel deep forest algorithm, feature selection, multilevel load balancing

收稿日期：2023−04−17；**修回日期：**2023−07−01

通信作者：陈志刚，czg@csu.edu.cn

基金项目：广东省重点提升基金资助项目（No.2022ZDJS048）；科技创新 2030-“新一代人工智能”重大基金资助项目（No.2020AAA0109605）

Foundation Items: Key Promotion Project of Guangdong Province (No.2022ZDJS048), “2030 Innovation Megaprojects”-New Generation Artificial Intelligence Project (No.2020AAA0109605)

0　引言

深度森林是 Zhou 等[1]提出的一种基于决策树结构的深度学习模型，其包含多粒度扫描和级联森林两大组成部分，因其超参数少、参数敏感度低及模型深度自适应等优点，已被广泛应用于网络流量分类[2]、文本分类[3]、故障诊断[4]、目标识别[5]、恶意代码分类[6]等领域。然而，随着新一代信息技术的革新和大数据时代的来临，各领域将产生亟待处理的海量数据，这些数据通常表现出数据量大、数据价值密度低等特性，深度森林难以有效处理这类数据，因此如何设计出适合处理大数据问题的深度森林算法已成为一大研究热点。

Spark[7]作为专门处理大规模数据问题开发的并行计算框架，因其出色的计算能力和良好的通用性，被广泛应用于企业项目开发和学术研究中。文献[8]提出了用于退网用户预测的并行深度森林（PDF-OGUP, parallel deep forest for off-grid user prediction）算法，为节省多粒度扫描阶段的空间占用，设计了基于下标的扫描算法，并以随机采样构建随机森林的方式减少所需内存空间。针对网络入侵问题，文献[9]设计了基于特征分割和深度并行随机森林（FS-DPRF, feature segmentation and deep structure of parallelized random forest）检测模型，提出了 RDD（resilient distributed datasets）层次替换策略解决了 RDD 重用问题，提高了作业效率。为进一步提高并行深度森林算法的计算能力，文献[10]结合 Spark 框架设计了一种全新的并行深度森林 BLB-gcForest（bag of little bootstraps-gcForest）算法。首先，该算法使用 BLB（bag of little bootstrap）自助采样法替换传统采样法，减少了大量特征在级联森林各层级中的传输，提高了计算效率和通信效率；其次，提出自适应子森林划分算法，以确保每个子森林并行计算的资源利用率最大化；最后，利用轮询机制来实现节点的负载均衡。以上列举的 3 种并行深度森林算法虽然在训练效率上有了一定的提升，但仍然存在以下不足。1) 在特征选择阶段，无法有效去除原始数据携带的大量冗余和无关特征，导致后续模型训练过程中存在冗余及无关特征问题。2) 在多粒度扫描阶段，输入的原始特征经过滑动窗口扫描后，将产生大量的特征子序列，拼接多个输出的类向量将导致类向量过长问题。3) 在级联森林训练阶段，级联森林的每一层都将拼接原始特征和上层特征作为本层输入，但相对于原始特征的维度，每层转化后的增广特征的维度则要小得多，这将导致增广特征被淹没[11]，使模型收敛速度缓慢。4) 在模型并行化训练阶段，子森林的划分粒度不能依据模型训练效果自适应确定，加之异构节点情况下存在中间数据倾斜，将导致模型并行训练效率低下。

针对上述问题，本文提出了基于 Spark 和三路交互信息的并行深度森林（PDF-STWII, parallel deep forest algorithm based on spark and three-way interactive information）算法，其主要工作如下。

1) 提出基于特征交互的特征选择（FSFI, feature selection based on feature interaction）策略，通过消除原始特征中存在的大量冗余及无关特征，解决了冗余及无关特征过多的问题。

2) 提出多粒度向量消除（MGVE, multi-granularity vector elimination）策略，通过将多粒度扫描产生的任意 2 个相似类向量融合为一个向量，解决了多粒度扫描过程中产生的类向量过长问题。

3) 提出了级联森林增强（CFFE, cascade forest feature enhancement）策略，密集连接所有级联层输出的增广特征的同时动态缩减部分原始特征，解决了模型收敛速度慢的问题。

4) 提出了多级负载均衡（MBL, multi-level load balancing）策略，通过自适应子森林划分（ASFS, adaptive sub-forest splitting）算法控制森林划分粒度和异构倾斜数据划分（HSDP, heterogeneous skew data partition）算法平衡异构数据的倾斜，提高了模型的并行化训练效率。

1　相关概念介绍

定义 1　互信息[12]常用来衡量变量之间的相关性程度，互信息越大，变量间的相关性越强，反之，则相关性越弱。反映随机变量 f_i 和 f_j 相关性的互信息 $I(f_i;f_j)$ 可定义为

$$I(f_i;f_j)=H(f_i)-H(f_i\mid f_j) \tag{1}$$

其中，$H(f_i)$ 为变量 f_i 的信息熵，表示变量不确定性程度；$H(f_i\mid f_j)$ 为变量 f_j 确定时 f_i 的条件熵 $I(f_i;f_j)<\min\{H(f_i),(f_j)\}$。

定义 2　对称不确定性[13]常用于相关特征选取，其通过归一化互信息修正了互信息在选取特征时存在的偏置。2 个随机变量 f_i 和 f_j 的对称不确定

性 $\mathrm{SU}(f_i,f_j)$ 可定义为

$$\mathrm{SU}(f_i,f_j)=\frac{2I(f_i;f_j)}{H(f_i)+H(f_j)} \tag{2}$$

从式(2)可知，$\mathrm{SU}(f_i,f_j)\in[0,1]$。

定义 3 三路交互信息[14]作为互信息的扩展可用来度量特征之间的交互性，其值可为正数、零和负数。当三路交互信息为正数时，2 个特征共同对标签提供的信息大于它们单独对标签提供信息的和，此时 2 个特征存在互补性；当三路交互信息为负数时，2 个特征对标签提供的信息存在冗余；当三路交互信息为零时，2 个特征提供给标签的信息是独立的。对于特征 f_i 和 f_j 及标签 C，三路交互信息 $I(f_i;f_j;C)$ 可表示为

$$I(f_i;f_j;C)=\sum\sum\sum p(f_i,f_j,C)\cdot \log\frac{p(f_i,f_j)p(f_i,C)p(f_j,C)}{p(f_i)p(f_j)p(C)} \tag{3}$$

其中，$p(f_i)p(f_j)p(C)$ 为三者的联合概率。

定义 4 近似马尔可夫毯[15]可用于冗余特征的检验，如果特征 f_j 是特征 f_i 的近似马尔可夫毯，则 2 个特征之间存在冗余，$\mathrm{SU}(f_j,C)\geqslant\mathrm{SU}(f_i,C)$ 和 $\mathrm{SU}(f_j,f_i)\geqslant\mathrm{SU}(f_i,C)$ 同时成立。

定义 5 皮尔逊相关系数常用来衡量 2 个向量之间的相似程度，取值范围为[–1,1]，其绝对值越大，相关性越强。当取值为正时，2 个向量呈正相关，当取值为负时，2 个向量呈负相关；当取值为零时，2 个向量无关。皮尔逊相关系数定义为

$$P(X,Y)=\frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}=\frac{\mathrm{E}((X-\mu_X)(Y-\mu_Y))}{\sigma_X\sigma_Y}=\frac{\mathrm{E}(XY)-\mathrm{E}(X)E(Y)}{\sqrt{\mathrm{E}(X^2)-\mathrm{E}^2(X)}\sqrt{\mathrm{E}(Y^2)-\mathrm{E}^2(Y)}} \tag{4}$$

其中，$\mathrm{cov}(X,Y)$ 表示 2 个向量之间的协方差，σ_X 和 σ_Y 分别表示向量 X 和向量 Y 的标准差，μ 表示向量均值，E 表示数学期望值。

2 PDF-STWII 算法说明

PDF-STWII 算法主要包括 4 个阶段：特征选择、多粒度扫描、级联森林训练、模型并行化训练。各阶段的主要任务如下。

1) 特征选择。提出 FSFI 策略，通过度量特征的相关性和冗余度，消除大量冗余及无关特征，同时挖掘出存在交互作用的特征，过滤大量冗余及无关特征。

2) 多粒度扫描。提出 MGVE 策略，融合任意 2 个相似类向量，缩短类向量长度。

3) 级联森林训练。提出 CFFE 策略，密集连接各层增广特征，同时逐层削减部分特征，防止增广特征被淹没，加快模型收敛速度。

4) 模型并行化训练。提出了 MBL 策略，其包含两方面内容。在算法并行处理层面，提出 ASFS 算法，通过分析子森林训练效果，自适应确定森林的划分粒度，提高算法并行度。在数据并行化处理方面，提出了 HSDP 算法，分析分布式异构环境中各计算节点的性能差异，将中间数据合理分配到各节点，以平衡中间数据倾斜，最终从算法和数据两方面提高模型并行化训练效率。

2.1 特征选择

针对原始数据集包含大量冗余及无关特征问题，提出的 FSFI 策略从特征相关性、冗余度和特征交互三方面综合考虑特征选取，高效剔除冗余无关特征。FSFI 包括无关特征过滤、冗余特征消除和特征综合评分。

2.1.1 无关特征过滤

在特征选择过程中，由于相对于特征的冗余度和交互性计算，特征的相关性计算更快，所以在特征选择的初始阶段，提出特征相关性系数（FRC）过滤大量无关特征，删除小于相关性阈值的特征，并利用 FRC 对特征排序。

定理 1 特征相关性系数（FRC）。已知数据集 $D\in\mathbf{R}^{n\times m}$，其中 n 和 m 分别为数据的样本量和特征，则 f_i 与标签 C 的相关性系数 FRC_i 定义为

$$\mathrm{FRC}_i=D\frac{I(f_i,C)}{\min(H(f_i),H(C))} \tag{5}$$

$$D=\sqrt{\frac{1}{n-1}\sum_{s=1}^{n}\left(f_{si}-\frac{1}{n}\sum_{s=1}^{n}f_{si}\right)^2} \tag{6}$$

其中，f_{si} 表示样本 s 中 f_i 的值。

证明 对标签具有较强区分度的特征，通常存在较大的方差，可用标准差反映特征 f_i 对类别的区分能力。D 为特征 f_i 的标准差，标准差越大，特征区分标签类别的能力越强；由互信息定义可知 $I(f_i;C)<\min\{H(f_i),H(C)\}$，互信息的大小受特征和标签信息熵的限制，直接使用互信息来衡量相关性时，具有越大信息熵的特征越有可能被选取，因

此将互信息 $I(f_i;C)$ 除以特征 f_i 和标签 C 的最小信息熵以消除偏置，最终将反映特征区分度的标准差和消除偏置的互信息相乘获得特征相关性系数 FRC，证毕。

2.1.2 冗余特征消除

经过无关特征初步过滤过程，特征的维度大幅缩减，但冗余特征并未消除，为此，在特征消除阶段提出冗余度指标 R 来衡量特征之间的冗余程度。冗余消除过程如下。首先，利用近似马尔可夫毯快速判断冗余特征并消除；然后，利用冗余度指标 R 计算特征间的冗余度，对比冗余度指标和冗余度阈值，进一步消除冗余特征。

定理 2　冗余度指标 R。已知存在特征 f_i 和特征 f_j，则计算特征间的冗余度指标 R 可表示为

$$R = P\mathrm{SU}(f_i,f_j) \tag{7}$$

$$P=\frac{1}{2}\left(\frac{\mathrm{SU}(f_i,C)}{H(f_i)}+\frac{\mathrm{SU}(f_j,C)}{H(f_j)}\right) \tag{8}$$

证明　$\mathrm{SU}(f_i,C)$ 为特征 f_i 与标签 C 的对称不确定性，根据对称不确定性定义可知，$\mathrm{SU}(f_i,C)$ 可度量特征 f_i 与标签 C 的相关信息量，同理，$\mathrm{SU}(f_i,f_j)$ 可度量 2 个特征之间的相关信息量，反映特征信息重叠大小。$H(f_i)$ 为 f_i 的信息熵，表示特征自身信息量的大小。当 $\frac{\mathrm{SU}(f_i,C)}{H(f_i)}$ 和 $\frac{\mathrm{SU}(f_j,C)}{H(f_j)}$ 越大时，在一个确定信息空间中的特征 f_i 和特征 f_j 的信息重叠概率也就越大，即越可能存在信息冗余。综上，P 可表示冗余概率，$\mathrm{SU}(f_i,f_j)$ 可表示冗余信息量，冗余概率和冗余信息量联立获得冗余度指标 R，证毕。

2.1.3 特征综合评分

经过无关特征过滤和冗余特征消除过程，剩余的特征都具有较高质量，为了进一步挖掘出更高质量的特征子集，从特征相关性、冗余度和特征交互性出发，设计特征综合评估函数 J_{FSFI}，获取更优特征子集。

定理 3　特征综合评估函数 J_{FSFI}。假设候选特征 f_i 与标签 C 的相关性为 $I(f_i;C)$，与已选特征 f_j 的冗余度为 $I(f_i;f_j)$，候选特征 f_i 和已选特征 f_j 对标签的交互性为 $I(f_i;f_j;C)$，特征综合评估函数 J_{FSFI} 可表示为

$$J_{\mathrm{FSFI}}=\arg\max_{f_i\in F'}(I(f_i;C)+\max_{f_j\in F_s}\left(\frac{I(f_i;C)-I(f_i;f_j)}{I(f_i;C)}I(f_i;f_j;C)\right)) \tag{9}$$

其中，F' 表示候选特征集，F_s 表示已选特征集。

证明　特征评估函数 J_{FSFI} 的目标在于每次从候选特征集 F' 中选取好的特征 f_i 使评估函数 J_{FSFI} 的值最大，好的特征应具有高相关性，且与已选特征具有低冗余度和高交互性，反映在函数中分别对应 $I(f_i;C)$、$\frac{I(f_i;C)-I(f_i;f_j)}{I(f_i;C)}$、$I(f_i;f_j;C)$。

当候选特征 f_i 与标签 C 的相关性较高时，$I(f_i;C)$ 越大，$\frac{I(f_i;C)-I(f_i;f_j)}{I(f_i;C)}$ 越大，J_{FSFI} 越大，候选特征 f_i 越容易被选择。

当候选特征 f_i 和已选特征 f_j 的冗余度较低时，$I(f_i;f_j)$ 越小，$\frac{I(f_i;C)-I(f_i;f_j)}{I(f_i;C)}$ 越大，J_{FSFI} 越大，候选特征 f_i 越容易被选择。

当候选特征 f_i 与已选特征 f_j 的交互性较高时，$I(f_i;f_j;C)$ 越大，J_{FSFI} 越大，候选特征 f_i 越容易被选择。

综上，特征评估函数 J_{FSFI} 在选择特征时能够有效挖掘出高相关性、低冗余度且具有交互作用的候选特征，证毕。

FSFI 的伪代码如算法 1 所示。

算法 1　FSFI

输入　特征集 F，相关性阈值 λ，冗余度阈值 $\delta m'$，最终选取的特征数目 m'

输出　已选特征集 F_s

1) 初始化 $F'=\varnothing,F_s=\varnothing,F_t=\varnothing$
2) 计算 F 中所有特征的 FRC
3) 将 $\mathrm{FRC}>\lambda$ 的特征放入 F' 并按降序排列
4) 当 F_s 中的 f_j 是 F' 中 f_i 的近似马尔可夫毯时，将 f_i 从 F' 中删除
5) 利用 R 与 δ 进一步删除 F' 中的冗余特征
6) 计算 J_{FSFI} 并将候选特征放入 F_t
7) 从 F_t 中选取使 J_{FSFI} 最大的特征 f_k
8) 将 f_k 放入 F_s 中并统计 F_s 中的特征数量
9) 重复步骤 7)和步骤 8)，直到 F_s 中特征数目为 m'

2.2 多粒度扫描

多粒度扫描[16]利用多种尺寸的滑动窗口对原始特征进行切片，随后将切片得到的多个窗口尺寸大小的特征子序列传入随机森林中进行训练，最后将训练得到的类向量拼接传入级联森林中训练。然而由于滑动窗口扫描得到的特征子序列存在大量相同特征，训练得到的大量类向量也相似，拼接大量相似类向量将使传入级联森林的类向量过长，增加级联森林训练开销。

针对多粒度扫描过程中产生的类向量过长问题，本节设计了 MGVE 策略将相似类向量融合。其具体过程如图 1 所示。

定理 4 相似类向量判定函数 $S(P(\boldsymbol{A},\boldsymbol{B}),\delta)$。已知在多粒度扫描阶段随机森林输出类向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，则 2 个向量的相似性判定表示为

$$S(P(\boldsymbol{A},\boldsymbol{B}),\delta)=\begin{cases}1, P(\boldsymbol{A},\boldsymbol{B})>\delta\\ 0, P(\boldsymbol{A},\boldsymbol{B})\leqslant\delta\end{cases} \tag{10}$$

其中，$P(\boldsymbol{A},\boldsymbol{B})$ 为向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的皮尔逊相关系数，δ 为设定的相似度阈值。当 $P(\boldsymbol{A},\boldsymbol{B})>\delta$ 时，$S(P(\boldsymbol{A},\boldsymbol{B}),\delta)=1$ 表明 2 个向量相似，反之不相似。

证明 由于 $P(\boldsymbol{A},\boldsymbol{B})$ 能直接反映 2 个向量之间的线性相关程度，同时每个随机森林输出的类向量为各个类别的概率，这使每个向量的内部概率值的和为 1。当用皮尔逊相关系数测得 2 个向量相关性越大时，2 个向量方向越趋于一致，此时 2 个向量内对应的各数值就越接近，2 个向量相似度越高，因此用皮尔逊相关系数与设定的阈值 δ 相比可判定 2 个向量是否相似，证毕。

MGVE 的伪代码如算法 2 所示。

算法 2 MGVE

输入 原始特征 F_s，相似度阈值 δ

输出 转化后特征 F_c

1) 初始化 $F_c=\varnothing, F_v=\varnothing$
2) 用大小为 m 的滑动窗口对 F_s 扫描
3) 训练窗口切片获取类向量 $\boldsymbol{V}_i$，$F_v=F_v\bigcup\boldsymbol{V}_i$
4) 对于 F_v 中的任意 2 个向量 $\boldsymbol{V}_a$ 和 $\boldsymbol{V}_b$
5) 如果 $P(\boldsymbol{V}_a,\boldsymbol{V}_b)>\delta$，则 $\boldsymbol{V}_c=\dfrac{\boldsymbol{V}_a+\boldsymbol{V}_b}{2}$
6) $F_v=F_v-\{\boldsymbol{V}_a,\boldsymbol{V}_b\}, F_v=F_v\bigcup\{\boldsymbol{V}_c\}$
7) $F_c=F_c\bigcup\{F_v\}$

2.3 级联森林训练

针对级联森林训练过程中模型收敛速度慢的问题，本节提出了 CFFE 策略，其主要过程如下。首先，密集连接每一层级联森林产生的增广特征；其次，为维持总的输入特征的维度不变，每一层级联森林训练后都根据训练效果给原始特征赋予不同的特征重要性权重 w，去除部分权重低的特征。具体过程如图 2 所示。

定理 5 特征 j 重要性权重 $w(j)$。假设 $w^{\mathrm{RF}_i}(j)$ 表示特征 j 是级联森林中第 i 个随机森林 RF_i 中的权重，m 个随机森林训练使用了特征 j，则特征 j 在本层的重要性权重 $w(j)$ 为

$$w(j)=\frac{\sum_{i=1}^{m}w^{\mathrm{RF}_i}(j)}{m} \tag{11}$$

证明 假设在构建决策树时，决策树 τ 内部的节点 i 被预测为类别 c 的概率为 $p(c)$，则节点 i 的信息熵 $E(i)$ 可表示为

$$E(i)=\sum_{\forall c}p(c)\ln\frac{1}{p(c)} \tag{12}$$

特征 j 将节点 i 划分为左右子节点，左右子节点的信息熵分别为 $E_{\mathrm{l}}(i)$ 和 $E_{\mathrm{r}}(i)$，则节点 i 被 j 划分的效果 $Q(i,j)$ 可表示为

$$Q(i,j)=\exp\{-(E_{\mathrm{l}}(i)+E_{\mathrm{r}}(i))\} \tag{13}$$

决策树 τ 总共有 N 个节点，特征 j 在决策树 τ 中的局部权重 $w^{\tau}(j)$ 可表示为

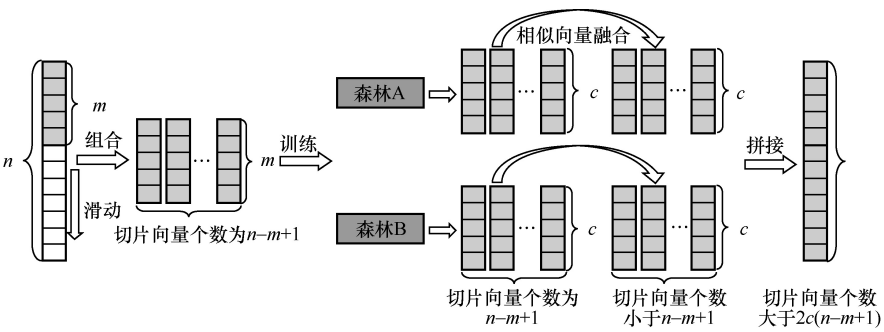


图 1 MGVE 过程

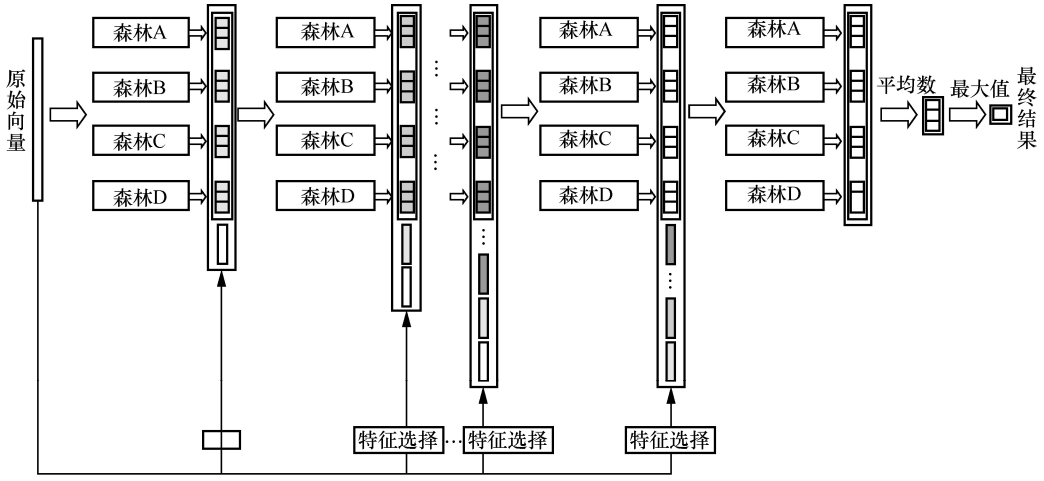


图 2　CFFE 过程

$$w^{\tau}(j)=\frac{\sum_{i=1}^{N}Q(i,j)}{N} \tag{14}$$

为评估决策树权重，使用袋外误差 δ 作为评估标准。设决策树 τ 的袋外误差为 δ^{τ}，则随机森林中决策树 τ 的归一化权重 γ^{τ} 可表示为

$$\gamma^{\tau}=\frac{\frac{1}{\delta^{\tau}}}{\max\limits_{\tau}\frac{1}{\delta^{\tau}}} \tag{15}$$

通过式(14)和式(15)，获得特征 j 在决策树 τ 中的局部权重 $w^{\tau}(j)$ 和决策树权重 γ^{τ}，则特征 j 在单个随机森林 RF 中的权重 $w^{\mathrm{RF}}(j)$ 可表示为

$$w^{\mathrm{RF}}(j)=\frac{\sum_{\forall\tau}\left(w^{\tau}(j)\gamma^{\tau}\right)}{\max\limits_{j}\sum_{\forall\tau}\left(w^{\tau}(j)\gamma^{\tau}\right)} \tag{16}$$

其中，$w^{\mathrm{RF}_i}(j)$ 表示特征 j 是级联森林中第 i 个随机森林 RF_i 中的权重，m 个随机森林训练使用了特征 j，则特征 j 在本层的权重为

$$w(j)=\frac{\sum_{i=1}^{m}w^{\mathrm{RF}_i}(j)}{m} \tag{17}$$

证毕。

2.4　模型并行化训练

针对模型并行化训练效率低的问题，本节提出了 MLB 策略，从算法和数据 2 个层面提升模型的并行化训练效率，包含算法层面的 ASFS 算法和数据层面的 HSDP 算法。

2.4.1　自适应子森林划分

在算法层面，为提高模型的并行化训练效率，本节提出了 ASFS 算法，其主要过程为如下。首先，采用自助采样法将采样特征分配到子森林中；然后，根据各个子森林的训练结果给每个子森林设定子森林权重系数 W_{SF}；最后，利用子森林的权重 W_{SF} 计算出整个森林划分得分因子 $\mathrm{score}_{\mathrm{F}}$ 以确定森林划分粒度。具体过程如图 3 所示。

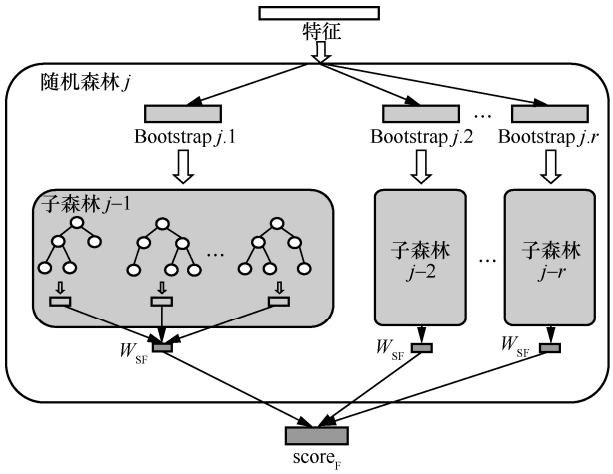


图 3　子森林划分

定理 6　子森林权重系数 $W_{\mathrm{SF}}(r)$。设第 r 个子森林中包含 Q 个决策树，利用 OOB 数据集验证获得第 i 个决策树的袋外误差 errOOB_i，则第 r 个子森林的权重系数 $W_{\mathrm{SF}}(r)$ 可表示为

$$W_{\mathrm{SF}}(r)=\sqrt{\frac{Q}{\frac{\mathrm{flu}}{\overline{\mathrm{errOOB}}}}} \tag{18}$$

$$\mathrm{flu}=\sum_{i=0}^{Q}\left|\mathrm{errOOB}_i-\overline{\mathrm{errOOB}}\right| \tag{19}$$

$$\overline{\text{errOOB}}=\frac{\sum_{i=0}^{Q}\text{errOOB}_i}{Q} \tag{20}$$

证明 由于 errOOB_i 为第 i 个决策树的袋外误差，$\overline{\text{errOOB}}$ 为子森林中决策树的袋外误差的均值，则 flu 可以统计子森林内决策树误差相距平均误差波动总和。误差波动总和越小，子森林预测的稳定性越高；平均袋外误差 $\overline{\text{errOOB}}$ 越小，子森林整体的预测准确率越高，所以 $W_{\text{SF}}(r)$ 可同时将子森林稳定性 flu 和准确率 $\overline{\text{errOOB}}$ 作为权重的考虑对象，使具有较高预测稳定性和高预测准确率的子森林获得高权重，证毕。

定理 7 森林划分得分因子 $\text{score}_{\text{F}}(s)$。将第 s 个森林划分为 r 个子森林，则第 s 个森林的森林划分得分因子为

$$\text{score}_{\text{F}}(s)=\frac{\sum_{i=0}^{r}(1-\overline{\text{errOOB}_i})W_{\text{SF}}(i)}{r} \tag{21}$$

证明 $1-\overline{\text{errOOB}_i}$ 为第 i 个子森林的平均预测准确率，准确率越高，子森林整体的分类能力越强。$W_{\text{SF}}(i)$ 为子森林权重系数，权重越大，子森林的稳定性越强、准确率越高，一个森林包含多个子森林，每个子森林的预测效果又包含准确率和稳定性两方面特性，因此结合两方面特性的 $\text{score}_{\text{F}}(s)$ 可表示子森林的整体预测效果，证毕。

ASFS 的伪代码如算法 3 所示。

算法 3 ASFS

输入 级联层数 T，每层森林数 S，预设最大子森林数 R，子森林中树的数量 Q

输出 子森林划分矩阵 $\boldsymbol{P}[\,][\,]$

```
for i = 1:T
    for j = 1:S
        score[] = 0
        for k = 1:R
            计算 W_SF(k) 和 score[k]
        end for
            获取使 score[] 最大的 k
            P[i][j] = k
    end for
end for
根据划分矩阵划分子森林
```

2.4.2 异构倾斜数据划分

在数据层面，由于 Spark 在 Shuffle 阶段采用默认的哈希分区策略极易引起中间数据倾斜，严重影响模型的并行化训练效率，为此本文提出 HSDP 算法。平衡中间数据倾斜需进行如下操作。

1) 倾斜评估。Spark 以哈希分区作为默认的分区方式将产生 2 种数据倾斜情况：同一键值包含大量键值对，经过 Shuffle 过程被分配到同一分区，导致这一分区数据量巨大；大量不同键值对应同一分区索引，导致大量不同键对应的键值对分配到同一分区。以上 2 种数据倾斜情况在节点异构环境下将更加严重，对此，本文提出异构倾斜度量因子 D 来评估在节点异构条件下中间数据的倾斜程度。

定理 8 异构倾斜度量因子 D。假设中间数据包含 m 个不同的 key，且第 i 个 key 对应的数据容量为 Q_i，N 个桶对应 N 个计算节点，第 j 个桶包含的 key 表示为 $\{K_{1,j},K_{2,j},\cdots,K_{m,j}\}$，每个桶的数据量依次表示为 $q_1,\cdots,q_j,\cdots,q_N$，$q_{\text{avg}}$ 为所有桶的平均数据量，则异构倾斜度量因子 D 可表示为

$$D=\frac{D'}{q_{\text{avg}}} \tag{22}$$

$$D'=\sqrt{\frac{1}{N}\sum_{j=1}^{N}(q_j-q_{\text{avg}}\text{RC}_j)^2} \tag{23}$$

$$\text{RC}_j=\frac{\text{capability}_j}{\text{avg\_capability}} \tag{24}$$

其中，RC_j 表示第 j 个计算节点的相对计算能力。

证明 由于 q_{avg} 和 avg_capability 是实际环境中的固定值，于是可设定系数 α 表示两者的比例，即 $q_{\text{avg}}=\alpha\text{avg\_capability}$。$q_j-\alpha\text{capability}_j$ 为第 j 个桶的理论最大负载和实际负载的差值，D' 为实际负载和理论负载的标准差，实际负载和理论负载越接近，异构倾斜度量因子 D 越小，因此可用 D 作为异构倾斜度量因子来衡量中间数据倾斜程度，证毕。

2) 中间数据预测。为降低数据统计耗时，采用主从整体采样法预测中间数据。首先，从节点通过 RDD 操作计算所有 Map 任务的 mapPartitionsRddSize；然后，设置采样率 r，通过 sampleSize = $r$mapPartionsRddSize 计算总共的采样大小，根据 sampleSizePerPartion 计算每个 Map 任务采样的样本大小；其次，每个从节点利用 sampleSizePartion 的大小调用 RDD 的 sample 函数对 RDD 数据分区进行采样，统计出本地样本中 key 值记录，随后将 (K_i,Q_i) 传输到主节点；最后，主节点汇总每个 Map 任务的所

有样本数量，根据采样率得到中间数据集 $\{(K_1,Q_1),(K_2,Q_2),\cdots,(K_m,Q_m)\}$ 的整体分布情况。

3) 异构倾斜数据划分。通过整体采样方法获得中间数据的预测，根据节点的异构情况采用贪心策略将中间数据合理分配到各个桶中。

HSDP 的伪代码如算法 4 所示。

算法 4　HSDP

输入　中间数据 $\{(K_1,Q_1),(K_2,Q_2),\cdots,(K_m,Q_m)\}$

桶的平均容量 q_{avg}，桶的数量 N

输出　分区集合 P

1) for $j=1:N$ //初始化剩余容量
2) $\text{RB}[j]=q_{\text{avg}}\text{RC}_j$
3) end for
4) 遍历 $\{(K_1,Q_1),(K_2,Q_2),\cdots,(K_m,Q_m)\}$
5) $x=\text{hash}(K_i)\%N$ //默认哈希分区
6) if $\text{RB}[x]>Q_i$
7) $y=x$，$P=P\bigcup\{<K_i,K_j>\}$
8) else if $\exists\text{RB}[]>Q_i$
9) $y=$大于Q_i且剩余容量最小桶的id
10) $P=P\bigcup\{<K_i,y>\}$
11) else $\text{Temp}=\text{Temp}\bigcup\{<K_i,Q_i>\}$
12) 根据数据量的大小将 Temp 降序排列
13) 用剩余容量大的桶装大的中间数据
14) $P=P\bigcup\{<K_i,y>\}$

2.5　算法时间复杂度分析

PDF-OGUP、FS-DPRF 和 BLB-gcForest 等算法都基于 Spark 框架设计，且各自采用不同的优化策略提高算法性能，因此选取这 3 种算法与本文算法进行实验对比。

PDF-STWII 算法主要包括特征选择、多粒度扫描、级联森林训练、级联森林并行化训练。各阶段的时间复杂度分别标记为 T_1、T_2、T_3、T_4。

特征选择包括无关特征过滤、冗余特征消除、特征综合评分。已知数据样本量为 n，特征数目为 m，无关特征过滤遍历所有样本和特征，其时间复杂化度为 $O(nm)$；冗余特征消除需要计算近似马尔可夫毯和三路交互信息，需要的时间复杂度为 $O(m^2)$；特征综合评分阶段需要的时间复杂度为 $O(m^2n)$，因此特征选择时间 T_1 为

$$T_1=O(m^2n+m^2+mn) \tag{25}$$

在多粒度扫描阶段，时间复杂度主要取决于特征子集在随机森林训练以及类向量融合的时间开销。假设经过特征选择后的特征个数为 s，滑动窗口大小为 w，样本数目为 n，随机森林的个数为 N,则 T_2 为

$$T_2=O(s-w)+O(s(s-w)nN)+O(N^2) \tag{26}$$

其中，$O(s-w)$ 为窗口扫描时间复杂度，$O(s(s-w)nN)$ 为特征子集训练时间复杂度，$O(N^2)$ 为类向量融合的时间复杂度。

在级联森林训练阶段，假设传入级联森林的原始特征的个数为 v，样本数目为 n，每一层森林的个数为 N，每个森林包含 Q 棵树,级联森林层数为 L，则 T_3 为

$$T_3=O(Lv(nN)) \tag{27}$$

在模型并行化训练阶段中，时间复杂度主要由子森林划分、异构数据分区两部分组成。假设每一层森林的个数为 N，每个森林包含 Q 棵树，级联森林的层数为 L，每个森林可划分为 r 子森林，并行节点数量同样为 r，则 T_4 为

$$T_4=O(NLQ)+O(r^2) \tag{28}$$

其中，$O(NLQ)$ 为自适应子森林划分的时间复杂度，$O(r^2)$ 为异构数据分区的时间复杂度。

综上，PDF-STWII 算法的时间复杂度为

$$T_{\text{PDF-STWII}}=\frac{T_1+T_2+T_3+T_4}{r} \tag{29}$$

其中，r 为单个森林划分的子森林个数。

在大数据环境下，深度森林模型训练的时间复杂度主要取决于多粒度扫描阶段中输出的类向量长度和级联森林训练层数，即算法的时间复杂度 T 主要由 T_3 中的 v 和 L 决定。由于算法 PDF-OGUP、FS-DPRF 和 BLB-gcForest 都没在多粒度扫描阶段对相似类向量进行融合，从而使 $v_{\text{PFG-OGUP}}>v_{\text{PDF-STWII}}$，$v_{\text{FS-DPRF}}>v_{\text{PDF-STWII}}$，$v_{\text{BLB-gcForest}}>v_{\text{PDF-STWII}}$。又由于本文在级联森林中使用了 CFFE 策略加快了模型收敛，因此需要的训练层数相对更少，从而使 $L_{\text{PFG-OGUP}}>L_{\text{PDF-STWII}}$，$L_{\text{FS-DPRF}}>L_{\text{PDF-STWII}}$，$L_{\text{BLB-gcForest}}>L_{\text{PDF-STWII}}$。综上，相较于 PDF-OGUP、FS-DPRF 和 BLB-gcForest 算法，PDF-STWII 算法具有更低的时间复杂度。

3　实验结果分析

3.1　实验环境

为验证本文算法的性能表现，本文设计了相关实验。在硬件方面，本文实验设置 8 个计算节点，其中包括 1 个主节点和 7 个从节点。各个计算节点的硬件

配置均为 Intel(R) Core(TM) i7-11800H CPU、16 GB DDR4 RAM、1 TB SSD，实验中的计算节点处于同一局域网内，通过 1 GB/s 的以太网相连。在软件方面，各计算节点配置均为 Ubuntu16.04、Hadoop 2.7.4、JDK 1.8.0。各节点的详细配置如表 1 所示。

表 1 节点详细配置

节点类型	主机名	IP 地址
Master	M	192.168.110.1
Slaver	S_1	192.168.110.2
Slaver	S_2	192.168.110.3
Slaver	S_3	192.168.110.4
Slaver	S_4	192.168.110.5
Slaver	S_5	192.168.110.6
Slaver	S_6	192.168.110.7

3.2 实验数据与设置

实验数据。所有算法采用 4 个来自 UCI（university of California Irvine）公共数据库的数据集，分别为 Farm Ads、Susy、Connect-4 和 FMA，其中 Farm Ads 是从 12 个网站文本中搜集的各种有关农场动物的话题；Susy 是记录粒子在加速器条件下是否产生超对称粒子信号过程的数据集；Connect-4 数据集记录了四子棋游戏中所有合法的 8 层位置信息；FMA 记录了包括歌曲标题、专辑、艺术家等众多曲目信息。各数据集的详细信息如表 2 所示。

表 2 实验数据集

数据集	样本数/条	特征数/种	数据特点
Farm Ads	4413	54 877	样本少特征多
Susy	5 000 000	18	样本多特征少
Connect-4	67 557	42	样本特征适中
FMA	106 574	518	样本特征适中

实验设置。对于实验数据划分，采用所有算法数据划分一致性原则，即 70%为训练集，30%为测试集；对于模型参数，设数据的特征长度为 d，在多粒度扫描阶段中滑动窗口大小依次设置为 $\frac{d}{16}$、$\frac{d}{8}$、$\frac{d}{4}$，每个子森林中的决策树的数量初始化为随机森林中决策树数量的开方，每一层级联森林包含 2 个随机森林和 2 个完全随机森林。

3.3 评价指标

3.3.1 加速比

加速比是指同一任务在单处理器系统和在并行处理器系统中运行消耗的时间的比率，常用来衡量并行系统或程序并行化的性能和效果，加速比越大，算法的并行化程度越高，其定义如下

$$S_p = \frac{T_s}{T_P} \tag{30}$$

其中，T_s 表示在串行系统中的执行时间，T_P 表示在并行系统中的执行时间。

3.3.2 准确率

准确率（Accuracy）是指在分类模型中正确分类的样本数与总的样本数的比值，能够反映算法的分类能力，其定义为

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{31}$$

其中，TP、TN、FP、FN 在混淆矩阵中分别表示真正例、真反例、假正例、假反例。

3.4 算法性能的比较分析

算法整体性能需考虑多方面指标，为综合衡量算法性能，利用算法运行时间来度量算法训练速度，利用加速比来度量算法并行处理能力，利用准确率来度量算法分类性能。

3.4.1 算法运行时间对比分析

为检验 4 种算法训练速度，将 PDF-OGUP、FS-DPRF、BLB-gcForest 与本文算法（PDF-STWII）在上述 4 个数据集上进行对比实验，森林中决策树数量为 200，实验采用 10 折交叉验证方式，实验结果如图 4 所示。

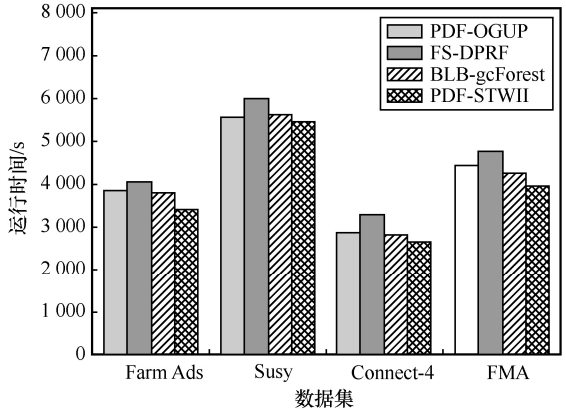


图 4 不同数据集上 4 种算法的运行时间

从图 4 中可知，在对 4 个数据集的测试中，本文算法所需要的运行时间最低，并且当数据集的特征数量越多时，本文算法相对其他算法缩短的运行时间比例也越大，在特征量最少的数据集 Susy 中，本文算法相比 PDF-OGUP、FS-DPRF、BLB-gcForest

运行时间分别减少了 2.62%、10.41%、3.41%；在特征量最多的数据集 Farm Ads 中，PDF-STWII 算法相比 PDF-OGUP、FS-DPRF、BLB-gcForest 运行时间分别减少了 13.8%、19.12%、10.76%。产生以上结果的主要原因如下。1) 本文算法设计了 FSFI 策略，消除了大量冗余及无关的特征，在不影响分类精度的前提下极大地减少了后续多粒度扫描和级联森林训练过程中输入的特征量，加快了模型的训练速度；2) 本文算法设计了 MGVE 策略，通过将 2 个相似的类向量融合为一个类向量，减少了级联森林训练过程中的特征维度，进而减少级联森林的训练开销。实验结果表明，PDF-FSIF 算法在处理高维大数据问题时具有良好性能。

3.4.2　加速比对比分析

为验证本文算法的并行计算能力，本文利用上述的 4 个数据集分别对 PDF-OGUP、FS-DPRF、BLB-gcForest 和本文算法在不同计算节点下进行算法加速比实验，实验采用 10 折交叉验证方式进行，森林中决策树数量设置为 200，实验结果如图 5 所示。

从图 5 可知，各算法的加速比均随着计算节点数量的增加而呈现不同程度的上升。当节点个数为 8 时，本文算法的加速比高于对比算法，在特征量最少的数据集 Farms Ads 中，本文算法的加速比分别比 PDF-OGUP、FS-DPRF、BLB-gcForest 高 0.32、0.52、0.18；在特征量最大的数据集 Susy 中，本文算法的加速比分别高 0.88、1.18、0.465；本文算法取得最高加速比的原因在于设计了 MLB 策略，从算法结构划分和中间数据合理分配 2 个层面的同时提高了模型的并行化训练效率，从而使算法在处理数据时具有更高的加速比。实验结果表明，PDF-STWII 算法在处理大数据问题时，具有较高加速比。

3.4.3　准确率对比分析

为验证本文算法的分类性能，实验选取准确率作为评估指标，将本文算法与对比的 PDF-OGUP、FS-DPRF 和 BLB-gcForest 算法在 4 个数据集上进行 10 折交叉验证实验，实验结果如图 6 所示。

从图 6 中可知，随着决策树数量的增加，4 种算法模型的分类准确率都有一定的提升，其主要原因在于随着决策树数量的增加，算法的泛化能力得到了增强，准确率随之提高。实验发现本文算法具有更高的准确率，当森林中决策树数量为 200 时，本文算法在 4 个数据集上的平均准确率相比

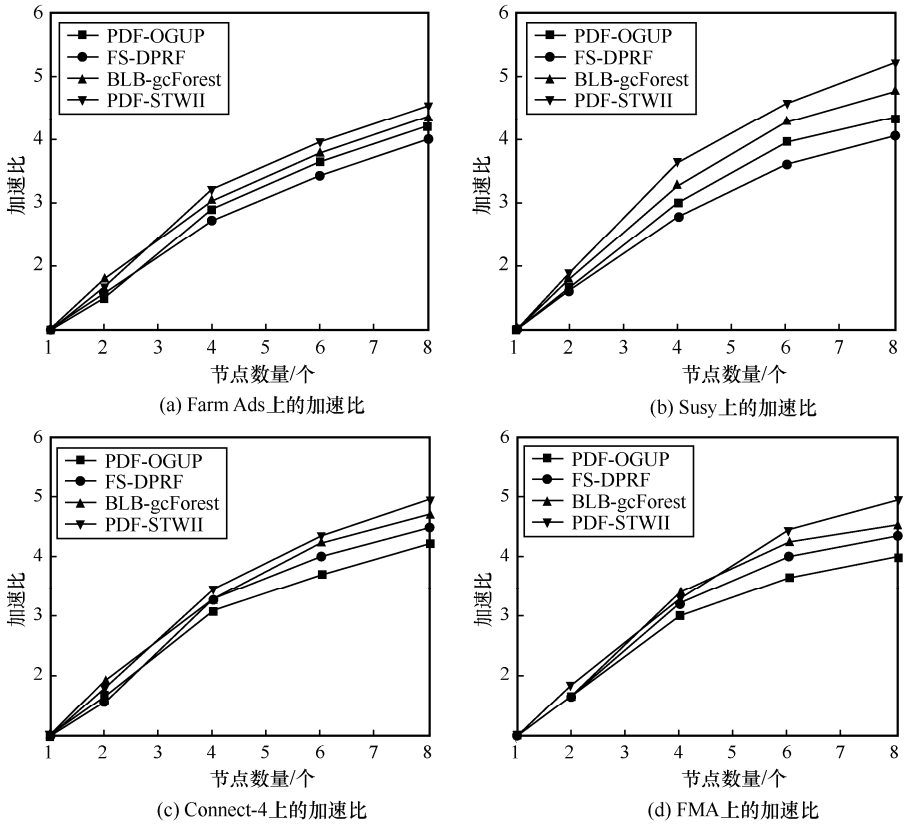


图 5　不同数据集上 4 种算法的加速比

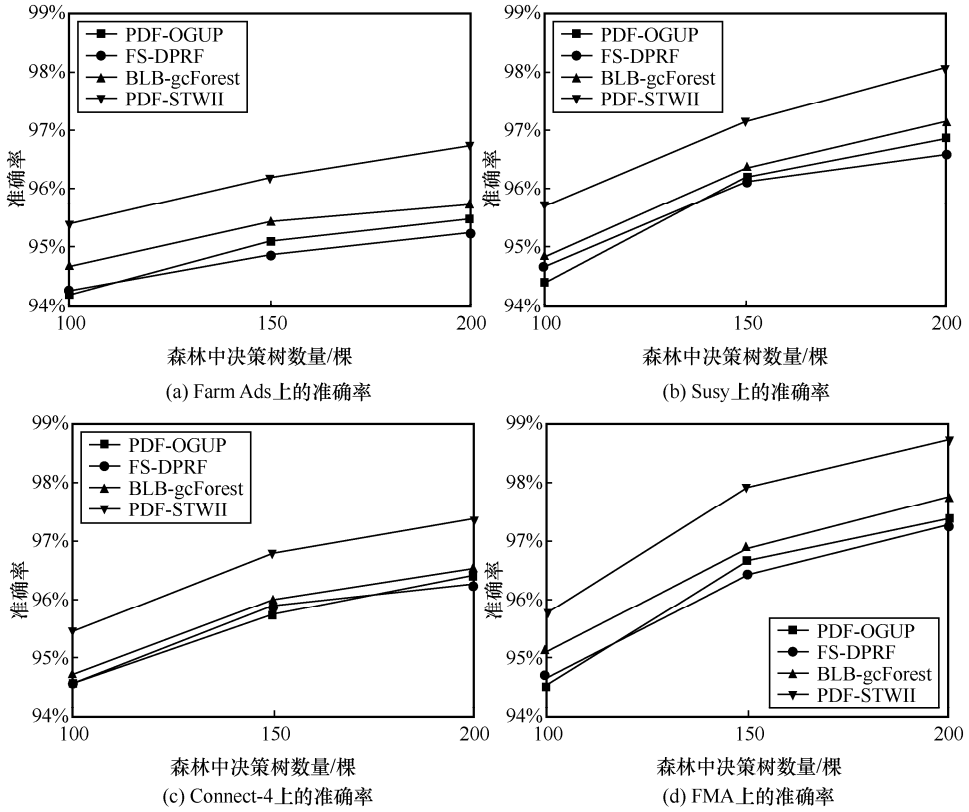


图 6　不同数据集上 4 种算法的分类精确度

PDF-OGUP、FS-DPRF 和 BLB-gcForest 分别提高了 1.24%、1.43%、0.96%，产生以上结果的原因如下。1) 本文算法设计了 FSFI 策略，消除大量冗余和无关特征，同时挖掘出具有交互作用的特征，提高了算法分类准确率；2) 本文算法设计了 CFFE 策略，密集连接增广特征，充分利用每一层级联森林的分类贡献，提高了模型的预测能力。实验结果表明，本文算法在大数据环境下具有优良的分类性能。

3.5　消融实验

为验证算法各策略的有效性和对算法模型的贡献，选取准确率和加速比作为评价指标，在上述 4 个数据集上设计消融实验，实验采用 8 个计算节点，森林中决策树数量为 200，实验结果由 10 折交叉验证获得，实验结果如表 3 所示。

从表 3 可知，各策略对加速比和准确率具有不同影响，其中，MBL 策略对算法加速比提升最明显，其次分别是 FSFI、MGVE 和 CFFE，在处理 4 类数据集时，相比无任何策略，使用了 MBL、FSFI、MGVE 和 CFFE 策略可将算法的平均加速比分别提升 19.04%、5.56%、3.64%和 1.98%。产生以上结果的原因如下。1) MBL 策略对森林自适应划分和平衡中间数据倾斜，能有效提高模型并行计算能力；2) FSFI 策略消除了原始特征中大量冗余无关特征，从而提高各计算节点的训练效率；3) MGVE 策略融合相似类向量，降低子森林训练开销，因此能一定程度提高加速比；4) CFFE 策略在级联森林训练过程中能够逐层削减少量特征，因此对加速比也有细微影响。

对算法准确率提升最大的是 CFFE 策略和 FSFI 策略，其次是 MGVE 策略和 MBL 策略，在处理 4 个数据集时，使用 CFFE、FSFI、MGVE 和 MBL 策略相比无任何策略，分别可将算法的平均准确率提升 1.98%、1.94%、0.45%和 0.21%。产生以上结果的原因如下。1) CFFE 策略密集连接各层增广特征，利用了每层森林的预测贡献；2) FSFI 策略消除了冗余无关特征并挖掘特征之间的交互信息；3) MGVE 策略将相似类向量融合对特征进行了转化，因此对准确率的提升有一定影响；4) MBL 策略主要划分森林结构和平衡中间数据倾斜，因此对准确率影响不大。综上，以上 4 种策略能有效应对大数据分类问题，且能有效提高算法加速比和准确率。

表 3　　消融实验结果

数据集	算法策略	加速比	准确率
Farm Ads	所有策略	4.53(+0.79)	96.74%(+3.23%)
	仅 FSFI	3.91(+0.17)	95.39%(+1.88%)
	仅 MGVE	3.86(+0.12)	93.83%(+0.32%)
	仅 CFFE	3.81(+0.06)	94.82%(+1.31%)
	仅 MBL	4.31(+0.57)	93.62%(+0.11%)
	无任何策略	3.74(±0)	93.51%(±0)
Susy	所有策略	5.24(+1.23)	98.06%(+3.91%)
	仅 FSFI	4.26(+0.25)	95.71%(+1.56%)
	仅 MGVE	4.15(+0.14)	94.54%(+0.39%)
	仅 CFFE	4.09(+0.08)	96.53%(+2.38%)
	仅 MBL	4.94(+0.93)	94.43%(+0.28%)
	无任何策略	4.01(±0)	94.15%(±0)
Connect-4	所有策略	4.97(+0.99)	97.4%(+3.52%)
	仅 FSFI	4.20(+0.22)	95.51%(+1.63%)
	仅 MGVE	4.13(+0.15)	94.26%(+0.38%)
	仅 CFFE	4.06(+0.08)	95.82%(+1.94%)
	仅 MBL	4.69(+0.71)	94.07%(+0.19%)
	无任何策略	3.98(±0)	93.88%(±0)
FMA	所有策略	4.96(+1.04)	98.75%(+4.03%)
	仅 FSFI	4.15(+0.23)	96.94%(+2.22%)
	仅 MGVE	4.08(+0.16)	95.32%(+0.6%)
	仅 CFFE	4.01(+0.09)	96.53%(+1.81%)
	仅 MBL	4.69(+0.77)	94.93%(+0.21%)
	无任何策略	3.92(±0)	94.72%(±0)

4　结束语

为解决深度森林算法在处理大数据存在的不足，本文提出了 PDF-STWII 算法。首先，提出了 FSFI 策略以消除原始特征中存在的大量冗余及无关特征；其次，提出了 MGVE 策略，通过将相似的 2 个类向量合并为一个类向量，解决了多粒度阶段中产生的类向量过长问题；随后，提出了 CFFE 策略，通过密集连接增广特征，提高信息利用率，加快了模型收敛速度；最后，提出了 MLB 策略，通过自适应子森林划分和异构倾斜数据划分，解决了模型并行化训练效率低的问题。实验结果表明，PDF-STWII 算法在处理大数据问题时具有良好的并行化训练效率和分类性能。

虽然 PDF-STWII 算法在并行化训练效率和分类精度上有了一定的提升，但仍存在以下不足：1)在多粒度向量消除策略中，利用求均值的方式将 2 个向量融合为一个向量会丢失部分信息；2) 在大数据环境中，本文算法难以有效处理不平衡数据分类问题。上述问题将作为未来的重点研究对象。

参考文献：

[1] ZHOU Z, FENG J. Deep forest: towards an alternative to deep neural networks[J]. arXiv Preprint, arXiv: 1702.08835, 2017.

[2] 戴瑾, 王天宇, 王少尉. 基于深度森林的网络流量分类方法[J]. 国防科技大学学报, 2020, 42(4): 30-34.
DAI J, WANG T Y, WANG S W. Network traffic classification method based on deep forest[J]. Journal of National University of Defense Technology, 2020, 42(4): 30-34.

[3] 牛振东, 石鹏飞, 朱一凡, 等. 基于深度随机森林的商品类超短文本分类研究[J]. 北京理工大学学报, 2021, 41(12): 1277-1285.
NIU Z D, SHI P F, ZHU Y F, et al. Research on classification of commodity ultra-short text based on deep random forest[J]. Transactions of Beijing Institute of Technology, 2021, 41(12): 1277-1285.

[4] 邵怡韦, 陈嘉宇, 林翠颖, 等. 小训练样本下齿轮箱故障诊断: 一种基于改进深度森林的方法[J]. 航空学报, 2022, 43(8): 112-126.
SHAO Y W, CHEN J Y, LIN C Y, et al. Gearbox fault diagnosis with small training samples: an improved deep forest based method[J]. Acta Aeronautica et Astronautica Sinica, 2022, 43(8): 112-126.

[5] 李璐, 杜兰, 何浩男, 等. 基于深度森林的多级特征融合 SAR 目标识别[J]. 电子与信息学报, 2021, 43(3): 606-614.
LI L, DU L, HE H N, et al. Multi-level feature fusion SAR automatic target recognition based on deep forest[J]. Journal of Electronics & Information Technology, 2021, 43(3): 606-614.

[6] 卢喜东, 段哲民, 钱叶魁, 等.一种基于深度森林的恶意代码分类方法[J]. 软件学报, 2020, 31(5): 1454-1464.
LU X D, DUAN Z M, QIAN Y K, et al. Malicious code classification method based on deep forest[J]. Journal of Software, 2020, 31(5): 1454-1464.

[7] 毛伊敏, 甘德瑾, 廖列法, 等. 基于 Spark 框架和 ASPSO 的并行划分聚类算法[J]. 通信学报, 2022, 43(3): 148-163.
MAO Y M, GAN D J, LIAO L F, et al. Parallel division clustering algorithm based on Spark framework and ASPSO[J]. Journal on Communications, 2022, 43(3): 148-163.

[8] LI X B, SUN Y, ZHANG F Z, et al. Potential off-grid user prediction system based on Spark[J]. ZTE Communications, 2019, 17(2): 26-37.

[9] LIU Z P, SU N, QIN Y W, et al. A deep random forest model on Spark for network intrusion detection[J]. Mobile Information Systems, 2020, 2020: 1-16.

[10] CHEN Z X, WANG T, CAI H B, et al. BLB-gcForest: a high-performance distributed deep forest with adaptive sub-forest splitting[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(11): 3141-3152.

[11] 庞明. 新型深度森林模型的研究[D]. 南京: 南京大学, 2020.
PANG M. Study on new deep forest model[D]. Nanjing: Nanjing University, 2020.

[12] 李占山, 杨云凯, 张家晨. 基于熵权法的过滤式特征选择算法[J]. 东北大学学报(自然科学版), 2022, 43(7): 921-929.
LI Z S, YANG Y K, ZHANG J C. Filtering feature selection algorithm

based on entropy weight method[J]. Journal of Northeastern University (Natural Science), 2022, 43(7): 921-929.

[13] 顾翔元, 郭继昌, 李重仪, 等. 基于对称不确定性和三路交互信息的特征子集选择算法[J]. 天津大学学报(自然科学与工程技术版), 2021, 54(2): 214-220.
GU X Y, GUO J C, LI C Y, et al. Feature subset selection algorithm based on symmetric uncertainty and three-way interaction information[J]. Journal of Tianjin University (Science and Technology), 2021, 54(2): 214-220.

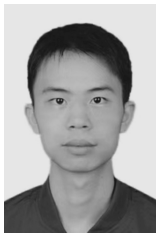
[14] 肖利军, 郭继昌, 顾翔元. 一种采用冗余性动态权重的特征选择算法[J]. 西安电子科技大学学报, 2019, 46(5): 155-161.
XIAO L J, GUO J C, GU X Y. Algorithm for selection of features based on dynamic weights using redundancy[J]. Journal of Xidian University, 2019, 46(5): 155-161.

[15] 张俐, 王枞, 郭文明. 利用近似马尔可夫毯的最大相关最小冗余特征选择算法[J]. 西安交通大学学报, 2018, 52(10): 141-145.
ZHANG L, WANG C, GUO W M. A feature selection algorithm for maximum relevance minimum redundancy using approximate Markov blanket[J]. Journal of Xi'an Jiaotong University, 2018, 52(10): 141-145.

[16] 杨晓晖, 张圣昌. 基于多粒度级联孤立森林算法的异常检测模型[J]. 通信学报, 2019, 40(8): 133-142.
YANG X H, ZHANG S C. Anomaly detection model based on multi-grained cascade isolation forest algorithm[J]. Journal on Communications, 2019, 40(8): 133-142.

[作者简介]

毛伊敏（1970− ），女，新疆伊犁人，博士，江西理工大学教授、博士生导师，主要研究方向为数据挖掘、大数据安全与隐私保护。



周展（1998− ），男，江西丰城人，江西理工大学硕士生，主要研究方向为数据挖掘、大数据。

陈志刚（1964− ），男，湖南益阳人，博士，中南大学教授、博士生导师，主要研究方向为网络与分布式计算、机会网络。