

DOI: 10.13382/j.jemi.B2307040

# 基于 PointCloudTransformer 和优化集成学习的 三维点云分类\*

于喜俊 段 勇

(沈阳工业大学信息科学与工程学院 沈阳 110870)

**摘要:**针对三维点云的不规则性和无序性所导致的难于提取特征并进行分类的问题,提出了一种融合深度学习和集成学习的三维点云分类方法。首先,训练深度学习点云分类网络 PointCloudTransformer,并使用主干网络提取点云特征,进而训练基分类器,获得基分类器集合;然后,针对集成学习算法设计基分类器选择模型,模型的优化目标为基分类器组合的差异性和平均总体精度。为了降低集成规模,本文基于增强后的白鲸优化算法提出了二元多目标白鲸优化算法,并使用该算法优化基分类器选择模型,获得集成剪枝方案集合;最后,采用多数投票法集成每个基分类器组合在测试集点云特征上的分类结果,获得最优基分类器组合,从而构建基于多目标优化剪枝的集成学习点云分类模型。在点云分类数据集上的实验结果表明,本文方法使用了更小的集成规模,获得了更高的集成精度,能够对多类别三维点云进行准确分类。

**关键词:** 三维点云分类;深度学习;集成学习;白鲸优化算法;多目标优化

**中图分类号:** TP391 **文献标识码:** A **国家标准学科分类代码:** 520.2

## Point cloud classification based on PointCloudTransformer and optimized ensemble learning

Yu Xijun Duan Yong

(School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China)

**Abstract:** Aiming at the difficulty of extracting features and classifying 3D point clouds due to their irregularity and disorder, a 3D point cloud classification method that fuses deep learning and ensemble learning is proposed. Firstly, the deep learning model PointCloudTransformer is trained to extract point cloud features and train base classifiers to establish a base classifier set. Subsequently, a base classifier selection model is designed for ensemble learning, and optimization objectives of the model are diversity and average overall accuracy of base classifiers. To reduce ensemble scale, binary multi-objective beluga optimization algorithm based on improved beluga optimization algorithm is proposed to optimize the base classifier selection model and obtain an ensemble pruning scheme set. Finally, the majority voting is used to ensemble the classification results of each base classifier combination on the test set to obtain the optimal base classifier combination, and an ensemble learning model of point cloud classification based on multi-objective optimization ensemble pruning is obtained. Experimental results on the point cloud classification dataset demonstrate that the method in this paper achieves higher ensemble accuracy with a smaller ensemble scale and can accurately classify multi-class 3D point clouds.

**Keywords:** 3D point cloud classification; deep learning; ensemble learning; beluga whale optimization; multi-objective optimization

## 0 引言

点云是由一组三维点坐标构成的数据集,用于描述物体或场景的三维形状和几何特征。点云分类是点云

数据处理过程中的重要环节,也是目前研究的热点方向。然而,由于点云数据的稀疏性、不规则性和无序性等缺点,提取点云特征并进行分类变得困难。

目前,点云分类方法主要分为基于机器学习的方法和基于深度学习的方法。基于机器学习的点云分类需要

使用法线估计、体素化和三维特征描述子等方法提取点云特征,然后使用分类器进行学习与分类。Zhao 等<sup>[1]</sup>从目标点云的空间分布、速度以及回波强度信息中提取特征向量,使用支持向量机(support vector machines, SVM)进行分类。胡海瑛等<sup>[2]</sup>将基于点基元和对象基元提取到的点云特征融合,获得多基元融合特征向量,使用随机森林(random forest, RF)进行分类。Tiwari 等<sup>[3]</sup>对比了 5 种算法的分类性能,实验结果表明,基于 Bagging 的决策树集成算法点云分类精度更高。Moorthy 等<sup>[4]</sup>使用径向有界最近邻域(radially bounded nearest neighbors)算法定义点的局部邻域,以提取不同空间尺度点的几何特征,分别使用 RF、XGBoost 以及 LightGBM 进行分类。

由于三维点云具有特殊的数据形式,无法直接输入到深度学习网络中。早期的深度学习点云分类网络采用将点云投影或切片为二维图像的方式进行输入,这种方式会丢失大量的空间结构信息。随着研究的深入,深度学习点云分类网络的输入形式已经实现了从二维图像到三维点云的转变。PointNet<sup>[5]</sup>是首个以三维点云作为输入的深度学习网络,被广泛应用于点云分类与分割任务。然而,PointNet 忽视了点云的局部特征提取。为了解决该问题,Qi 等<sup>[6]</sup>提出了 PointNet++,该网络对点云进行多次采样、分组和特征提取操作,有效解决了点云局部划分和提取局部特征问题,提高了分类精度。受到上述网络影响,近年来提出的点云分类网络仍然采用相似的方法提取点云局部特征,并结合其他结构进行特征提取与分类。其中,PointTransformer<sup>[7]</sup>使用自注意力(self-attention, SA)机制提取点云特征;PointMLP<sup>[8]</sup>放弃复杂的局部特征提取结构,提出几何仿射模块(geometric affine module),结合残差 MLP 结构提取点云特征。PointMLP-elite 作为 PointMLP 的精简版,虽然性能略有下降(下降幅度为 0.5%),但大幅降低网络参数量,提高了推理速度。

深度学习的优势在于自动提取特征,避免了复杂的人工特征提取过程。因此,研究人员使用深度学习网络提取特征,并将特征输入到集成学习算法中进行训练和预测。这种策略已经成功应用于诸多领域<sup>[9-10]</sup>。

在点云分类问题中,使用有效的分类器对基于三维点云提取的特征进行分类是关键。集成学习是一种有效的分类算法,具有较强的泛化性能。然而,随着基分类器集合规模的增大,集成模型的分类速度可能会显著降低,同时所需的存储空间也会增加。因此,需要进行集成剪枝。目前,集成剪枝方法可以分为基于排序的方法<sup>[11]</sup>、基于聚类的方法<sup>[12]</sup>、基于优化的方法<sup>[13]</sup>以及其他方法<sup>[14]</sup>。基分类器的差异性和平均总体精度是集成剪枝的两个重要指标,过分追求基分类器的差异性会降低其平均总体精度,而过度提高基分类器的平均总体精度也

会减小其差异性<sup>[15]</sup>。因此,找到两者之间的平衡状态是集成剪枝成功的关键。

基于此,本文使用训练后的深度学习点云分类网络提取点云特征,解决点云特征提取难的问题;基于点云特征训练基分类器集合,通过集成学习算法进行点云分类;将集成学习中的集成剪枝问题视为多目标组合优化问题,提出二元多目标白鲸优化算法进行多目标优化集成剪枝,解决基分类器的差异性和平均总体精度的平衡问题。

## 1 三维点云分类模型

点云分类是通过分析点云的几何、拓扑或语义特征,将点云整体或其中的点划分为不同类别或对象的过程。

本文提出了一种基于 PointCloudTransformer 深度学习网络和优化集成学习的三维点云分类方法,如图 1 所示。该方法基于 PCT(PointCloudTransformer)<sup>[16]</sup>提取三维点云特征,使用集成模型对点云整体进行分类。首先,将三维点云输入到训练后的 PCT 中,通过主干网络提取点云特征,输出 256 维特征向量;然后,将特征向量输入由集成剪枝获得的最优基分类器组合中,基分类器采用分类与回归树(classification and regression tree, CART),输出每个基分类器的分类结果;最后,通过多数投票法集成基分类器组合的分类结果,获得点云类别。

为了获得高质量的集成剪枝方案,提高三维点云分类精度,本文基于白鲸优化算法提出了二元多目标白鲸优化算法,并将其应用到集成剪枝过程中,借助元启发式多目标优化算法优秀的多目标权衡能力,实现在集成剪枝过程中自动平衡基分类器的差异性和平均总体精度的目标。

## 2 基于 PointCloudTransformer 提取三维点云特征

Transformer 使用自注意力(self-attention, SA)机制,使其能够更好地捕捉输入序列中的长距离依赖关系,从而能够处理任意长度的输入序列,并且不受输入序列的顺序影响,这种特性使得 Transformer 可以用于处理点云数据,解决由于点云数据无序性造成的学习困难问题。

PCT 是一种基于 Transformer 的深度学习网络,使用注意力机制学习三维点云特征。如图 1 所示,PCT 包含编码器(encoder)和分类解码器(classification decoder)。在 PCT 的编码器中,首先将三维点云坐标输入邻域嵌入(neighbor embedding)模块,将其嵌入到新的特征空间中;然后,将嵌入的特征输入到 4 个级联的偏置注意力(offset-attention, OA)模块中,以学习每个点的语义丰富

且具有判别性的表示;最后,通过线性层生成输出特征。在 PCT 的分类解码器中,首先对编码器输出的特征进行最大池化(max-pool)和平均池化(average-pool),然后将

池化后的特征输入到两个级联的 LBRD (Liner、BatchNorm、ReLU、Dropout) 层中,获得 256 维点云特征向量。

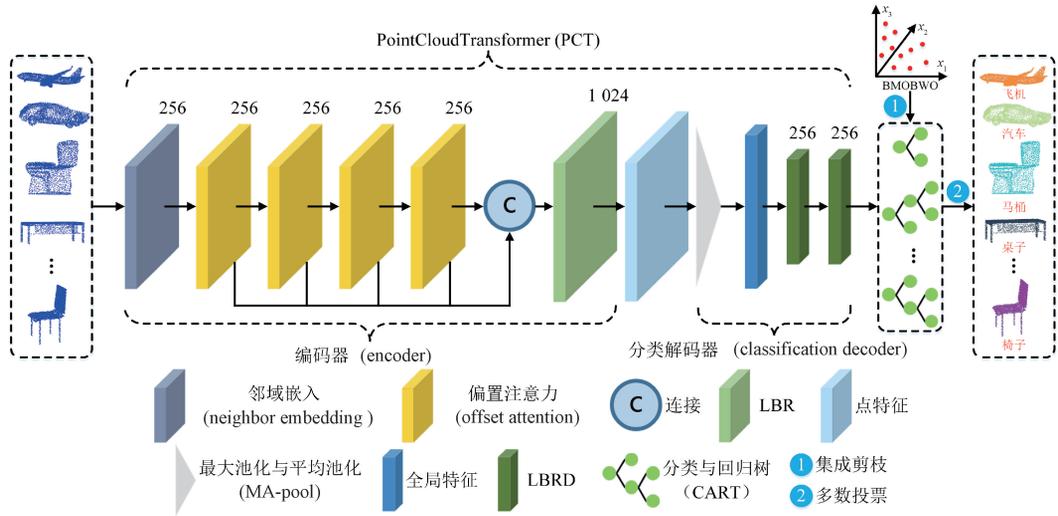


图1 融合 PCT 和优化集成学习的三维点云分类模型

Fig. 1 3D point cloud classification model based on PCT and optimized ensemble learning

邻域嵌入模块用于增强 PCT 的局部特征提取能力,包含 LBR (Liner、BatchNorm、ReLU) 层和采样-分组 (sample and group, SG) 层。SG 层首先使用最远点采样 (farthest point sampling, FPS) 算法对输入点云进行采样,并使用 k-NN 算法搜索每个采样点的 k 个最近邻点;然后令邻近点和采样点作差,并将差值和采样点拼接起来输入到 LBR 层中;最后通过最大池化获得点云局部特征。

偏置注意力层通过元素相减计算 SA 特征与输入特征之间的偏移量 (差值),并将这个偏移量输入到 LBR 层,如式(1)所示。

$$F_{out} = OA(F_{in}) = LBR(F_{in} - F_{sa}) + F_{in} \quad (1)$$

其中,  $F_{out}$  为输出点云特征,  $F_{in}$  为输入点云特征,  $F_{sa}$  为自注意力特征。经过实验证明, PCT 使用 OA 能够获得比使用 SA 更强的点云特征提取能力。

鉴于 PCT 优秀的点云特征提取能力,本文使用训练后的 PCT 提取三维点云特征,并将特征输入到集成学习算法中进行训练与测试。

### 3 基于多目标优化集成剪枝的集成学习三维点云分类

PCT 能够提取到比人工提取更具判别性的点云特征,从而提高点云分类精度的上限,使用有效的分类器逼近这个上限是点云分类问题的关键。

本文基于集成学习的思想,构建集成模型进行点云分类。为了降低集成规模,本文设计基分类器选择模型,

提出二元多目标白鲸优化算法优化模型,进行集成剪枝。具体过程为:首先,将 PCT 提取到的点云特征划分为训练集和验证集,通过预剪枝方法获得规模为  $L$  的基分类器集合;然后,将基分类器集合的总体精度 (overall accuracy, OA) 以及在验证集上的分类结果作为输入,使用二元多目标白鲸优化算法优化基分类器选择模型,输出集成剪枝方案集合;最后,根据每个集成剪枝方案选择基分类器组合对测试集点云特征进行分类,采用多数投票法集成每个基分类器组合的分类结果,获得测试集点云类别。在测试集上分类精度最高的基分类器组合就是最优组合。

#### 3.1 预剪枝方法

对于包含  $L$  个基分类器的集成学习模型,其中有  $2^L - 1$  个子集成组合,随着  $L$  的增大,直接使用二元多目标白鲸优化算法很难在搜索中平衡基分类器的差异性和平均总体精度。因此,本文通过预剪枝方法初始化基分类器集合,使其具有较高的平均总体精度,同时缩小集合规模,减小算法的搜索压力。具体过程为:首先,采用自助抽样法 (bootstrap method) 从训练集点云特征中抽取  $L$  个训练子集;然后,在每个训练子集上重复训练  $L'$  次,得到  $L$  组基分类器,每组包含  $L'$  个基分类器;最后,使用每组中的基分类器对测试集点云特征进行分类,将每组中总体精度最高的基分类器组成规模为  $L$  的基分类器集合。

#### 3.2 基分类器选择模型

本文针对集成学习算法设计基分类器选择模型,该

模型以难度度量 (Difficulty,  $\theta$ )<sup>[17]</sup> 作为衡量基分类器差异性的指标。对于  $L$  个基分类器, 存在离散型随机变量  $X' = 0, 1/L, 2/L, \dots, 1$ , 表示对某个样本做出正确分类的基分类器数量与基分类器总数的比值。 $\theta$  的计算如式(2)所示。

$$\theta = \text{var}(X') \quad (2)$$

式中:  $\theta$  的取值范围为  $[0, 0.25]$ ,  $\theta$  值越小, 表明基分类器间的差异越大。

假设每个基分类器的总体精度为  $OA_i$ , 从  $L$  个基分类器中选择  $l$  个基分类器, 其平均总体精度  $\overline{OA}$  的计算如式(3)所示。

$$\overline{OA} = \sum_{i=1}^l OA_i / l \quad (3)$$

综上所述, 基分类器选择模型的目标函数向量如下:

$$\mathbf{F}' = \min(\theta, -\overline{OA}) \quad (4)$$

其中,  $\mathbf{F}'$  表示模型的目标函数向量。

## 4 提出多目标白鲸优化算法应用于集成学习三维点云分类

本文使用元启发式多目标优化算法对集成学习三维点云分类模型进行多目标优化剪枝, 从而实现在集成剪枝过程中自动平衡基分类器的差异性和平均总体精度的目标, 该方法属于基于优化的集成剪枝方法。

高效的多目标优化算法能够在集成剪枝过程中穷尽搜索决策空间, 提供优质的集成剪枝方案, 从而构建可靠的集成学习三维点云分类模型。白鲸优化 (beluga whale optimization, BWO)<sup>[18]</sup> 算法具有参数少、易于实现以及寻优能力强等优点, 本文基于 BWO 算法提出了多目标白鲸优化 (multi-objective beluga whale optimization, MOBWO) 算法, 以解决多目标优化问题。

### 4.1 结合 Tent 映射和反向学习增强多目标白鲸优化算法

BWO 算法采用随机方法初始化白鲸种群, 可能导致白鲸的初始位置在决策空间内分布不均匀。

本文结合 Tent 映射<sup>[19]</sup> 和反向学习 (opposition-based learning, OBL)<sup>[20]</sup> 初始化白鲸种群, 以获得分布均匀且高质量的初始种群, 增强 MOBWO 算法的优化性能。通过比较 Tent 映射种群及其反向种群对应位置白鲸的适应度值 (目标函数值), 选择较优白鲸组成新的初始种群。

Tent 映射对初值不敏感, 具有更好的遍历均匀性, 其公式如下:

$$z_{k+1} = \begin{cases} z_k / \beta, & 0 \leq z_k \leq \beta \\ (1 - z_k) / (1 - \beta), & \beta \leq z_k \leq 1 \end{cases} \quad (5)$$

其中,  $k$  为映射次数,  $z_k (z_0 \neq \beta)$  表示第  $k$  次的映射

函数值,  $\beta$  为常数, 通常取 0.5。

反向学习能够提供更优质的解, 对于  $d$  维决策空间内的点  $X(x_1, x_1, \dots, x_d)$ , 其反向点  $\check{X}(\check{x}_1, \check{x}_1, \dots, \check{x}_d)$  的计算如式(6)所示。

$$\check{x}_i = u_b + l_b - x_i \quad (6)$$

其中,  $u_b$  和  $l_b$  分别为决策变量的上界和下界。

为了进一步提高 MOBWO 算法的优化性能, 本文将反向学习引入白鲸的位置更新过程, 使算法能够以优质解进行迭代。在每次位置更新后, 算法通过式(6)计算当前解的反向解, 并比较两个解的适应度值, 保留两者中的较优解作为种群中的下一代个体。

### 4.2 使用外部存档保存帕累托最优解

多目标优化是指同时优化多个相互冲突的目标。多目标优化问题的解是由多个帕累托最优解组成的帕累托最优解集。以最小化问题为例, 对于一个具有  $d$  维决策变量,  $k (k \geq 2)$  个目标函数的优化问题, 其定义如式(7)所示。

$$\min F(X) = (f_1(X), f_2(X), \dots, f_k(X)), X \in \Omega \quad (7)$$

其中,  $X(x_1, x_1, \dots, x_d)$  为  $d$  维决策变量,  $\Omega$  为决策空间,  $F: \Omega \rightarrow \Lambda$  是从决策空间到目标空间的映射,  $\Lambda$  是由  $k$  个目标函数构成的目标空间。

本文通过建立一个外部存档来保存帕累托最优解。外部存档的基本思想是: 将算法在优化过程中获得的帕累托最优解加入存档, 并在后续优化过程中更新存档。通常, 外部存档容量有限。因此, 应该引入一种更新存档的机制, 避免超出其最大容量限制。

当白鲸移动到一个新位置时, 会面临以下 3 种情况之一: 1) 若新的解被存档内一个或多个解支配, 则舍弃这个解; 2) 若新的解没有被存档内的解支配且存档未满, 则将其加入存档并移出存档内被其支配的解; 3) 若新的解没有被外部存档内的解支配, 并且该解也未能支配存档内的解, 同时存档已满, 则选择存档内的一个解将其替换为新的解。理想帕累托前沿应该在目标空间内沿着前真实帕累托前沿均匀分布。若在外部分档已满时更新存档, 应将最密集区域内的某个解移出存档, 并将新的解加入存档。

本文使用每个解在目标空间内距其最近的 3 个解的距离总和计算其所在区域的密度。假设外部存档内存在解  $X$ , 其目标函数值向量为  $\mathbf{F}(X)$ 。解  $X$  的 3 个最近邻解为  $X_i (i = 1, 2, 3)$ ,  $\mathbf{F}(X_i) (i = 1, 2, 3)$  分别为 3 个解的目标函数值向量。则  $X$  距其最近邻 3 个解的距离总和  $D$  的计算如式(8)所示。

$$D = \sum_{i=1}^3 \sqrt{(f_1(X) - f_1(X_i))^2 + \dots + (f_k(X) - f_k(X_i))^2} \quad (8)$$

对于存有  $m$  个解的外部存档, 每个解所在区域的密

度  $\rho_i$  的计算如式(9)所示。

$$\rho_i = D_i / \sum_{i=1}^m D_i \quad (9)$$

其中,  $D_i$  为外部存档内第  $i$  个解在目标空间内距其最近3个解的距离总和。 $\rho_i$  值越小,表明该解所在区域的密度越大。若在外部存档已满时更新存档,将所在区域密度最大的解移出存档,即移出  $\rho_i$  值最小的解。

#### 4.3 基于外部存档中的解进行位置更新

BWO 算法通过模拟白鲸的群体行为寻找最优解,包括3个阶段:探索阶段(exploration phase)、开发阶段(exploitation phase)和鲸落(whale fall)。

在 BWO 算法中,进入探索和鲸落阶段的白鲸根据随机白鲸的位置更新自身的位置,进入开发阶段的白鲸同时根据随机白鲸和最优白鲸的位置更新自身的位置。BWO 算法通过平衡因子  $B_f$  平衡探索阶段和开发阶段:

$$B_f = B_0(1 - T/2T_{\max}) \quad (10)$$

其中,  $B_0$  为  $(0,1)$  之间的随机数,  $T$  为当前迭代,  $T_{\max}$  为算法最大迭代次数。当  $B_f \geq 0.5$  时,白鲸进入探索阶段,反之进入开发阶段。随着  $T$  的增加,  $B_f$  的范围从  $(0,1)$  缩小到  $(0,0.05)$ ,表明白鲸进入开发阶段的概率随着  $T$  的增大而增加。

本文 MOBWO 算法在平衡探索和开发阶段做出调整,当  $B_f \geq b$  时,白鲸进入探索阶段,反之进入开发阶段,  $b$  的取值会通过实验详细讨论。在 MOBWO 算法的开发阶段,当前最优白鲸从外部存档内选择。通过式(10)计算平衡因子  $B_f^*$ ,以此选择外部存档内的帕累托最优解更新白鲸的位置。当  $B_f^* > b$  时,选择外部存档中  $\rho_i$  值最大的解,反之选择  $\rho_i$  值最小的解。

在算法迭代初期,进入开发阶段的白鲸倾向于选择外部存档内所在区域密度最小的解来更新自身的位置,这有助于延伸帕累托前沿的覆盖范围;随着  $T$  的增加,白鲸将更加倾向于选择外部存档内所在区域密度最大的解进行位置更新,这有助于得到分布更加均匀的帕累托前沿。同理,随机白鲸将从外部存档内随机选择。因此,基于 BWO 算法,本文 MOBWO 算法的探索阶段、开发阶段和鲸落的位置更新公式如式(11)~(13)所示。

$$\begin{cases} X_{i,j}^{T+1} = X_{i,p_j}^T + (Archive_{r,p_j} - X_{i,p_j}^T) \cdot \\ \quad (1 + r_1) \sin(2\pi r_2), j = \text{even} \\ X_{i,j}^{T+1} = X_{i,p_j}^T + (Archive_{r,p_j} - X_{i,p_j}^T) \cdot \\ \quad (1 + r_1) \cos(2\pi r_2), j = \text{odd} \end{cases} \quad (11)$$

其中,  $Archive_{r,p_j}$  为外部存档内随机帕累托最优解在第  $p_j$  维的位置。

$$\begin{cases} X_i^{T+1} = r_3 Archive_{\rho_{\max}} - r_4 X_i^T + C_1 \cdot L_F \cdot \\ \quad (Archive_r - X_i^T), B_f^* > b \\ X_i^{T+1} = r_3 Archive_{\rho_{\min}} - r_4 X_i^T + C_1 \cdot L_F \cdot \\ \quad (Archive_r - X_i^T), B_f^* \leq b \end{cases} \quad (12)$$

其中,  $Archive_{\rho_{\min}}$  和  $Archive_{\rho_{\max}}$  分别为外部存档内  $\rho_i$  值最小和最大的帕累托最优解。在每一次更新外部存档后,都需要计算外部存档内每个解的  $\rho_i$  值。

$$X_i^{T+1} = r_5 X_i^T - r_6 Archive_r + r_7 X_{step} \quad (13)$$

其中,  $Archive_r$  为外部存档内随机帕累托最优解的位置。

#### 4.4 二元多目标白鲸优化算法进行集成剪枝

本文将集成学习中的集成剪枝问题视为多目标组合优化问题。MOBWO 算法能够解决多目标优化问题,无法解决多目标组合优化问题。因此,本文使用转换函数将 MOBWO 算法离散化,提出二元多目标白鲸优化(binary multi-objective beluga whale optimization, BMOBWO)算法。具体做法为:保持算法的位置更新公式不变,设置决策变量取值范围为  $[-4,4]$ ,通过式(14)将白鲸位置向量的每一维实数值转化为该维取0或1的概率,然后通过式(15)确定该维的取值,获得对应的权重向量。

$$T(x) = 1/1 + e^{-2x} \quad (14)$$

$$x_{i,j}^{*T+1} = \begin{cases} 1, \text{rand} < T(x_{i,j}^{T+1}) \\ 0, \text{rand} \geq T(x_{i,j}^{T+1}) \end{cases} \quad (15)$$

式中:  $x_{i,j}^{*T+1}$  为第白鲸  $i$  在第  $j$  维的新权重,  $\text{rand}$  为  $(0,1)$  之间的随机数。

本文 BMOBWO 算法在 MOBWO 算法的基础上增加了一个权重向量存档,为集成学习三维点云分类模型中的每个基分类器赋予权重,权重为1代表该基分类器参与集成,权重为0代表该基分类器不参与集成。权重向量存档中的成员与帕累托最优解存档中的成员相对应。对于集成剪枝问题,权重向量存档中的每个成员都是集成剪枝方案,每个权重向量的维度  $d$  等于基分类器集合的规模  $L$ 。在种群初始化以及每一次位置更新后,通过式(15)计算每个解对应的权重向量,基于权重向量计算适应度值。

## 5 实验及结果分析

本章实验包括两部分,第1部分为 MOBWO 算法多目标优化性能测试实验,第2部分为融合 PCT 和优化集成学习的三维点云分类实验。

### 5.1 MOBWO 算法性能测试

#### 1) MOBWO 算法探索阶段和开发阶段的平衡

本节实验用于确定 4.3 节中  $b$  的取值,即 MOBWO 算法中白鲸进入探索阶段或开发阶段的概率。

本节使用 CEC2009<sup>[21]</sup> 中的 10 个无约束多目标优化测试函数来测试不同概率下 MOBWO 算法的优化性能。在 10 个测试函数中, F1~F7 包含两个优化目标, F8~F10 包含 3 个优化目标, F1~F9 的决策变量维度为 30, F10 的决策变量维度为 10。本节使用反世代距离 (inverse generational distance, IGD)<sup>[22]</sup> 作为多目标优化算法的评价指标。IGD 计算的是真实帕累托前沿中的每个成员到算法获得的帕累托前沿中距离最近的成员之间的平均距离。IGD 值越小,表明算法的优化性能越强:

$$IGD = \sqrt{\sum_{i=1}^M d_i^2} / M \quad (16)$$

其中,  $d_i$  为真实帕累托前沿中的第  $i$  个成员与算法获得的帕累托前沿中距离最近成员之间的欧氏距离,  $M$  为真实帕累托前沿中的成员总数。

当  $B_f \geq b$  时,白鲸进入探索阶段,反之进入开发阶段。 $b$  的值取自  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ 。设置 MOBWO

算法的白鲸种群数量  $n = 200$ ,最大迭代次数  $T_{max} = 1\ 000$ 。为了减小随机性对结果的影响,在每个测试函数上分别独立运行算法 30 次,并计算 IGD 的均值和标准差,结果如表 1 所示,表中加粗结果为每个测试函数上取得的最优解。

观察表 1,当  $b = 0.3$  时,MOBWO 算法在 4 个测试函数上取得了最优,在 F7 和 F10 上取得了次优。这证实了适当增大白鲸进入探索阶段的概率可以提高算法的优化性能。

究其本质,BWO 算法的 3 个阶段(探索阶段、开发阶段和鲸落)分别对应全局搜索、局部搜索以及避免局部最优解的过程。平衡全局搜索和局部搜索,可以使 BWO 算法更高效地搜索全局最优解。与单目标优化不同,多目标优化注重的是帕累托最优解集的整体质量,包括收敛性和分布性。收敛性表示算法获得的帕累托最优解与真实帕累托前沿的接近程度,而分布性表示帕累托最优解集在目标空间的分散程度。适当增大白鲸进入探索阶段的概率,可以使算法搜索更广阔的空间,从而获得更优质的帕累托最优解,因此,本文 MOBWO 算法中  $b$  的值为 0.3。

表 1 不同概率下 MOBWO 算法获得的 IGD 均值和标准差

Table 1 Mean and standard deviation of IGD obtained by MOBWO algorithm under different probabilities

函数		$b=0.1$	$b=0.2$	$b=0.3$	$b=0.4$	$b=0.5$
F1	Mean	$2.49 \times 10^{-3}$	$2.47 \times 10^{-3}$	<b><math>2.46 \times 10^{-3}</math></b>	$2.57 \times 10^{-3}$	$2.64 \times 10^{-3}$
	Std	$1.76 \times 10^{-4}$	$1.76 \times 10^{-4}$	<b><math>1.42 \times 10^{-4}</math></b>	$1.46 \times 10^{-4}$	$1.92 \times 10^{-4}$
F2	Mean	$1.78 \times 10^{-3}$	$1.70 \times 10^{-3}$	<b><math>1.68 \times 10^{-3}</math></b>	$1.99 \times 10^{-3}$	$2.45 \times 10^{-3}$
	Std	$3.90 \times 10^{-4}$	$3.32 \times 10^{-4}$	<b><math>2.93 \times 10^{-4}</math></b>	$3.96 \times 10^{-4}$	$5.12 \times 10^{-4}$
F3	Mean	$1.25 \times 10^{-2}$	<b><math>1.24 \times 10^{-2}</math></b>	$1.28 \times 10^{-2}$	$1.32 \times 10^{-2}$	$1.40 \times 10^{-2}$
	Std	$6.56 \times 10^{-4}$	$6.49 \times 10^{-4}$	$5.88 \times 10^{-4}$	$4.83 \times 10^{-4}$	<b><math>4.43 \times 10^{-4}</math></b>
F4	Mean	$1.97 \times 10^{-3}$	$1.95 \times 10^{-3}$	<b><math>1.93 \times 10^{-3}</math></b>	$2.03 \times 10^{-3}$	$2.17 \times 10^{-3}$
	Std	$7.12 \times 10^{-5}$	$5.77 \times 10^{-5}$	<b><math>4.93 \times 10^{-5}</math></b>	$5.41 \times 10^{-5}$	$9.89 \times 10^{-5}$
F5	Mean	$1.01 \times 10^{-1}$	$1.01 \times 10^{-1}$	<b><math>9.74 \times 10^{-2}</math></b>	$1.08 \times 10^{-1}$	$1.23 \times 10^{-1}$
	Std	<b><math>1.44 \times 10^{-2}</math></b>	$2.17 \times 10^{-2}$	$1.53 \times 10^{-2}$	$1.79 \times 10^{-2}$	$2.30 \times 10^{-2}$
F6	Mean	<b><math>1.27 \times 10^{-2}</math></b>	$1.28 \times 10^{-2}$	$1.41 \times 10^{-2}$	$1.59 \times 10^{-2}$	$1.82 \times 10^{-2}$
	Std	$1.60 \times 10^{-3}$	$1.64 \times 10^{-3}$	$1.40 \times 10^{-3}$	$1.02 \times 10^{-3}$	<b><math>9.79 \times 10^{-4}</math></b>
F7	Mean	$1.93 \times 10^{-3}$	<b><math>1.90 \times 10^{-3}</math></b>	$1.92 \times 10^{-3}$	$2.05 \times 10^{-3}$	$2.25 \times 10^{-3}$
	Std	$1.74 \times 10^{-4}$	$2.34 \times 10^{-4}$	$1.98 \times 10^{-4}$	$1.96 \times 10^{-4}$	<b><math>1.63 \times 10^{-4}</math></b>
F8	Mean	$1.26 \times 10^{-2}$	$1.16 \times 10^{-2}$	$1.17 \times 10^{-2}$	<b><math>1.15 \times 10^{-2}</math></b>	$1.25 \times 10^{-2}$
	Std	$8.89 \times 10^{-4}$	$1.68 \times 10^{-3}$	$1.60 \times 10^{-3}$	$1.65 \times 10^{-3}$	<b><math>4.84 \times 10^{-4}</math></b>
F9	Mean	$8.09 \times 10^{-3}$	<b><math>7.57 \times 10^{-3}</math></b>	$8.27 \times 10^{-3}$	$9.51 \times 10^{-3}$	$9.92 \times 10^{-3}$
	Std	$3.31 \times 10^{-3}$	<b><math>2.84 \times 10^{-3}</math></b>	$3.63 \times 10^{-3}$	$4.61 \times 10^{-3}$	$4.64 \times 10^{-3}$
F10	Mean	<b><math>1.58 \times 10^{-2}</math></b>	$1.84 \times 10^{-2}$	$1.66 \times 10^{-2}$	$1.74 \times 10^{-2}$	$1.67 \times 10^{-2}$
	Std	$2.51 \times 10^{-3}$	<b><math>9.70 \times 10^{-4}</math></b>	$4.06 \times 10^{-3}$	$1.51 \times 10^{-3}$	$3.96 \times 10^{-3}$

#### 2) MOBWO 算法与其他多目标优化算法对比

为了测试本文提出的 MOBWO 算法的多目标优化性能,本节将 MOBWO 算法与其他 5 种著名的元启发式多目标优化算法进行对比实验,包括多目标金鹰优化器

(multi-objective golden eagle optimizer, MOGEO)<sup>[23]</sup>、多目标灰狼优化器 (multi-objective grey wolf optimizer, MOGWO)<sup>[24]</sup>、非支配排序遗传算法 (non-dominated sorting genetic algorithm II, NSGA-II)<sup>[25]</sup>、多目标蚁狮优化

器 (multi-objective ant lion optimizer, MOALO)<sup>[26]</sup>。

对于所有算法,设置种群数量  $n = 200$ ,最大迭代次数  $T_{\max} = 1\ 000$ 。除了 NSGA-II 外,设置外部存档容量为 100。

将每种算法在 CEC2009 中的每个多目优化测试函数上独立运行 30 次后,计算 IGD 的均值和标准差,结果如表 2 所示。

表 2 各算法获得的 IGD 均值和标准差

Table 2 Mean and standard deviation of IGD obtained by each algorithm

函数		MOGEO	MOGWO	NSGA-II	MOALO	本文算法
F1	Mean	$3.96 \times 10^{-3}$	$5.77 \times 10^{-3}$	$4.47 \times 10^{-3}$	$7.67 \times 10^{-3}$	<b><math>2.46 \times 10^{-3}</math></b>
	Std	$4.77 \times 10^{-4}$	$9.73 \times 10^{-4}$	$1.30 \times 10^{-3}$	$6.58 \times 10^{-4}$	<b><math>1.42 \times 10^{-4}</math></b>
F2	Mean	$2.46 \times 10^{-3}$	$3.31 \times 10^{-3}$	$2.01 \times 10^{-3}$	$8.67 \times 10^{-3}$	<b><math>1.68 \times 10^{-3}</math></b>
	Std	<b><math>2.45 \times 10^{-4}</math></b>	$5.21 \times 10^{-4}$	$6.41 \times 10^{-4}$	$1.79 \times 10^{-3}$	$2.93 \times 10^{-4}$
F3	Mean	<b><math>8.91 \times 10^{-3}</math></b>	$1.21 \times 10^{-2}$	$1.09 \times 10^{-2}$	$1.72 \times 10^{-2}$	$1.28 \times 10^{-2}$
	Std	$1.33 \times 10^{-3}$	$2.43 \times 10^{-3}$	$1.30 \times 10^{-3}$	$1.94 \times 10^{-3}$	<b><math>5.88 \times 10^{-4}</math></b>
F4	Mean	$2.48 \times 10^{-3}$	$2.80 \times 10^{-3}$	$3.32 \times 10^{-3}$	$4.62 \times 10^{-3}$	<b><math>1.93 \times 10^{-3}</math></b>
	Std	$8.79 \times 10^{-5}$	$1.55 \times 10^{-4}$	$1.37 \times 10^{-4}$	$1.62 \times 10^{-3}$	<b><math>4.93 \times 10^{-5}</math></b>
F5	Mean	$1.99 \times 10^{-1}$	$2.36 \times 10^{-1}$	$4.99 \times 10^{-1}$	$3.15 \times 10^{-1}$	<b><math>9.74 \times 10^{-2}</math></b>
	Std	$6.12 \times 10^{-2}$	$1.25 \times 10^{-1}$	$5.77 \times 10^{-2}$	$8.26 \times 10^{-2}$	<b><math>1.53 \times 10^{-2}</math></b>
F6	Mean	$1.73 \times 10^{-2}$	$1.44 \times 10^{-2}$	$2.15 \times 10^{-2}$	$3.39 \times 10^{-2}$	<b><math>1.41 \times 10^{-2}</math></b>
	Std	$3.34 \times 10^{-3}$	$2.76 \times 10^{-3}$	$7.09 \times 10^{-3}$	$8.56 \times 10^{-3}$	$1.40 \times 10^{-3}$
F7	Mean	$2.10 \times 10^{-3}$	$4.62 \times 10^{-3}$	$6.66 \times 10^{-3}$	$1.08 \times 10^{-2}$	<b><math>1.92 \times 10^{-3}</math></b>
	Std	<b><math>1.04 \times 10^{-4}</math></b>	$9.80 \times 10^{-4}$	$5.75 \times 10^{-3}$	$5.08 \times 10^{-3}$	$1.98 \times 10^{-4}$
F8	Mean	$1.26 \times 10^{-2}$	$3.50 \times 10^{-2}$	$3.66 \times 10^{-2}$	$3.53 \times 10^{-2}$	<b><math>1.17 \times 10^{-2}</math></b>
	Std	$3.10 \times 10^{-3}$	$3.03 \times 10^{-2}$	$1.05 \times 10^{-2}$	$7.62 \times 10^{-3}$	$1.60 \times 10^{-3}$
F9	Mean	$1.48 \times 10^{-2}$	$1.14 \times 10^{-2}$	$3.43 \times 10^{-2}$	$3.06 \times 10^{-2}$	<b><math>8.27 \times 10^{-3}</math></b>
	Std	<b><math>3.43 \times 10^{-3}</math></b>	$5.49 \times 10^{-3}$	$7.85 \times 10^{-3}$	$5.03 \times 10^{-3}$	$3.63 \times 10^{-3}$
F10	Mean	$2.29 \times 10^{-2}$	$8.09 \times 10^{-2}$	$8.17 \times 10^{-2}$	$6.91 \times 10^{-2}$	<b><math>1.66 \times 10^{-2}</math></b>
	Std	$7.85 \times 10^{-3}$	$1.40 \times 10^{-1}$	$3.05 \times 10^{-2}$	$2.40 \times 10^{-2}$	<b><math>4.06 \times 10^{-3}</math></b>

## 5.2 三维点云分类实验

为了验证本文算法在三维点云分类研究中的有效性和优越性,本节进行横向和纵向对比实验。横向对比实验将随机森林 (random forest, RF)、XGBoost、Bagging、支持向量机 (support vector machine, SVM)、分类与回归树 (classification and regression tree, CART) 与本文算法进行对比。纵向对比实验分别使用 PointNet、PointNet++ 以及 PointMLP-elite 提取三维点云特征,与 PCT 进行对比,采用的评价指标为总体精度 (overall accuracy, OA) 与平均精度 (average accuracy, AA)。

### 1) 三维点云分类数据集

本节分别在 ModelNet40 数据集<sup>[27]</sup>和 ScanObjectNN 数据集<sup>[28]</sup>上进行三维点云分类对比实验。

ModelNet40 数据集被广泛应用于三维点云分类研究中,该数据集包含 40 个类别,共计 12 311 个 3D 模型。本文使用官方提供的划分,将 9 843 个对象用于训练,2 468 个对象用于测试。

ScanObjectNN 数据集以真实世界中的对象为模型,

从表 2 可以看出,本文算法在求解多目标优化问题中表现出了明显的有效性和优越性,其在 9 个测试函数上的优化结果均优于其他算法,在 F1、F2、F4、F5、F7、F9、F10 上具有明显优势,并且本文算法在 7 个测试函数上提供了更稳定的结果。因此,使用 MOBWO 算法求解多目标优化问题,可以获得优质的解决方案。

共计 15 000 个对象。由于点云中存在背景、噪声以及遮挡,该数据集具有更高的难度与挑战性。

### 2) 深度学习点云分类网络训练参数设置

所有的深度学习点云分类网络均设置统一的训练参数,设置 batchsize 大小为 32,训练 300 轮,使用带动量的随机梯度下降 (stochastic gradient descent, SGD) 算法优化网络,设置动量为 0.9,权重衰减为 0.000 2,初始学习率为 0.1。使用余弦退火策略调整学习率,最小学习率设置为 0.005。对于每个点云样本,随机选择 1 024 个点输入到点云分类网络中。

### 3) 三维点云分类对比实验

设置 BMOBWO 算法的白鲸种群数量  $n = 200$ ,最大迭代次数  $T_{\max} = 1\ 000$ 。所有横向对比算法在训练集点云特征上通过 5 折交叉验证进行网格调参,集成学习算法 (RF、XGBoost、Bagging) 的基分类器均为 CART。为了减小随机性带来的影响,本文算法获得的总体精度、平均精度以及集成规模均为独立运行算法 30 次后获得的均值。

各算法在 ModelNet40 数据集上的获得的总体精度

以及平均精度如表 3~4 所示,表中第 1 列括号内的数值为点云分类网络在 ModelNet40 数据集上的总体精度与平均精度,这些数据均取自对应文献,由于 PointNet++ 和

PCT 在对应文献中均未给出在 ModelNet40 数据集上的平均精度,在本文的表 4 中也就没有给出对应数据,ScanObjectNN 数据集同理。

表 3 各算法在 ModelNet40 数据集上的总体精度

Table 3 Overall accuracy of each algorithm on the ModelNet40 dataset

(%)

点云特征提取网络	RF	XGBoost	Bagging	SVM	CART	本文算法
PointNet(89.2)	89.8	90.0	89.5	89.5	80.3	<b>90.4</b>
PointNet++(90.7)	90.8	91.1	90.6	90.8	85.5	<b>91.3</b>
PointMLP-elite(93.6)	93.2	93.4	92.3	93.2	86.9	<b>93.8</b>
PCT(93.2)	93.4	93.5	92.7	93.1	87.8	<b>94.0</b>

表 4 各算法在 ModelNet40 数据集上的平均精度

Table 4 Average accuracy of each algorithm on the ModelNet40 dataset

(%)

点云特征提取网络	RF	XGBoost	Bagging	SVM	CART	本文算法
PointNet(86.0)	85.4	85.7	85.7	85.1	75.4	<b>86.2</b>
PointNet++(-)	86.2	87.8	87.9	86.2	82.3	<b>88.1</b>
PointMLP-elite( <b>90.9</b> )	89.6	89.8	88.8	89.1	82.9	90.7
PCT(-)	89.8	90.7	90.2	90.2	85.1	<b>91.3</b>

从表 3 和 4 可以看出,集成学习算法的性能整体优于非集成学习算法。在集成学习算法(RF、XGBoost、Bagging)中,XGBoost 性能最强,RF 和 XGBoost 的总体精度在前 3 种网络上均有提高,在 PointMLP-elite 上低于原始网络,Bagging 仅在 PointNet 上的总体精度高于原始网络。在非集成学习算法(SVM、CART)中,CART 性能较弱,SVM 优于 Bagging,并且总体精度在前两种网络上有提高。

进行横向对比,本文算法的总体精度和平均精度均优于其他 5 种对比算法。进行纵向对比,虽然 PCT 的原始总体精度低于 PointMLP-elite,但结合本文算法后,总体精度与平均精度均取得最优。此外,4 种深度学习点云分类网络与本文算法结合后,在 ModelNet40 数据集上的总体精度较原始网络分别提高了 1.2%、0.6%、0.2%、0.8%。

本文算法(ours)以及集成学习算法(RF、XGBoost、Bagging)在 ModelNet40 数据集上的集成规模对比如图 2 所示。通过图 2 进一步分析,相比于其他 3 种集成学习算法,本文算法的集成规模最少,使用 PointNet、PointNet++ 以及 PCT 提取三维点云特征,集成规模仅为其他 3 种集成学习算法最小值的 50% 左右,但分类精度更高。

为了验证本文算法在真实世界点云数据上的分类性能,进一步在 ScanObjectNN 数据集上进行对比实验,相关实验设置与在 ModelNet40 数据集上的实验设置相同。各算法在 ScanObjectNN 数据集上的获得的总体精度以及平均精度如表 5~6 所示。图 3 为本文算法(ours)以及集成学习算法(RF、XGBoost、Bagging)在 ScanObjectNN 数

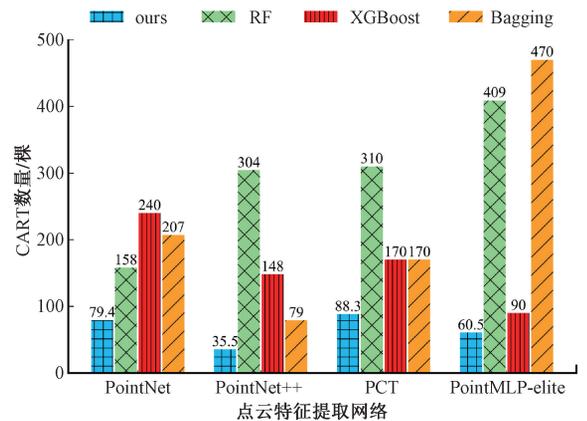


图 2 各算法在 ModelNet40 数据集上的集成规模对比  
Fig. 2 Comparison of ensemble scale of each algorithm on the ModelNet40 dataset

据集上的集成规模对比。

在 ScanObjectNN 数据集上进行横向对比,本文算法的总体精度和平均精度均优于其他 5 种对比算法。进行纵向对比,结合本文算法后,PointMLP-elite 在 ScanObjectNN 数据集上取得了最高的总体精度与平均精度,PCT 次之。PointNet、PointNet++ 以及 PointMLP-elite 与本文算法结合后,在 ScanObjectNN 数据集上的总体精度较原始网络分别提高了 1.3%、0.9%、0.5%。结合图 3 分析,本文算法(ours)的集成规模最小,但具有更高的分类精度。

综上所述,本文算法具有有效性和优越性,可以用于解决三维点云分类问题。

表 5 各算法在 ScanObjectNN 数据集上的总体精度

Table 5 Overall accuracy of each algorithm on the ScanObjectNN dataset

(%)

点云特征提取网络	RF	XGBoost	Bagging	SVM	CART	本文算法
PointNet(68.2)	68.5	68.9	68.3	68.2	62.3	<b>69.5</b>
PointNet++(77.9)	78.1	78.5	77.5	78.0	71.3	<b>78.8</b>
PointMLP-elite(83.8)	82.4	83.6	82.1	83.2	75.5	<b>84.3</b>
PCT(-)	80.4	81.1	80.0	80.2	74.0	<b>81.6</b>

表 6 各算法在 ScanObjectNN 数据集上的平均精度

Table 6 Average accuracy of each algorithm on the ScanObjectNN dataset

(%)

点云特征提取网络	RF	XGBoost	Bagging	SVM	CART	本文算法
PointNet(63.4)	63.3	63.7	62.2	61.3	57.0	<b>63.9</b>
PointNet++(75.4)	74.3	74.9	73.9	75.2	67.2	75.1
PointMLP-elite(81.8)	79.5	80.0	79.1	79.9	71.9	<b>82.4</b>
PCT(-)	77.5	78.8	76.9	77.4	71.1	<b>78.9</b>

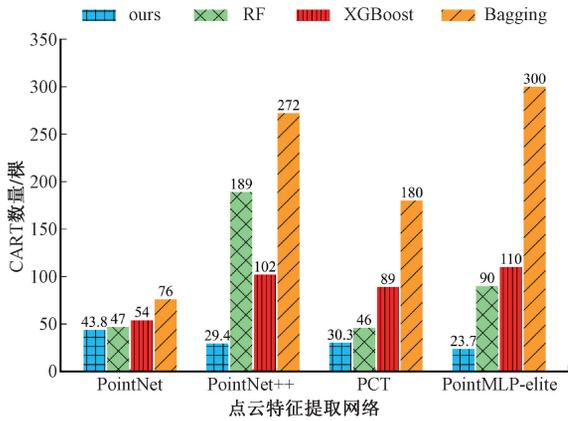


图 3 各算法在 ScanObjectNN 数据集上的集成规模对比

Fig. 3 Comparison of ensemble scale of each algorithm on the ScanObjectNN dataset

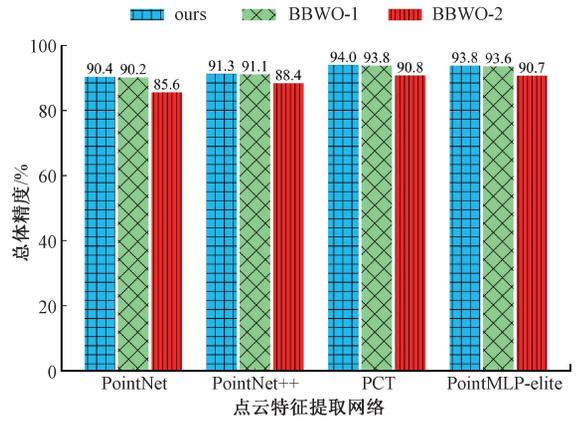


图 4 多目标和单目标优化集成剪枝在

ModelNet40 数据集上的总体精度

Fig. 4 Overall accuracy of ensemble pruning based on multi-objective and single objective optimization on the ModelNet40 dataset

4) 多目标优化与单目标优化集成剪枝三维点云分类性能对比

为了验证多目标优化集成剪枝相较于单目标优化集成剪枝的优势,本文将二元白鲸优化(binary beluga whale optimization, BBWO)算法作为对比算法, BBWO-1 和 BBWO-2 分别表示以难度度量  $\theta$  和平均总体精度  $\overline{OA}$  为优化目标进行单目标优化集成剪枝,将这两种算法与本文多目标优化集成剪枝进行对比。

显然,当以  $\overline{OA}$  作为优化目标进行单目标优化集成剪枝时,只需选择一个总体精度最高的基分类器,本文将这类解定义为不可行解。为了减小多数投票法的随机性,当算法以  $\overline{OA}$  为目标进行集成剪枝时,基分类器最小保留数量设置为 3。分别在 ModelNet40 数据集上运行本文算法和两种对比算法 30 次,得到三维点云分类总体精度以及集成规模,如图 4~5 所示。

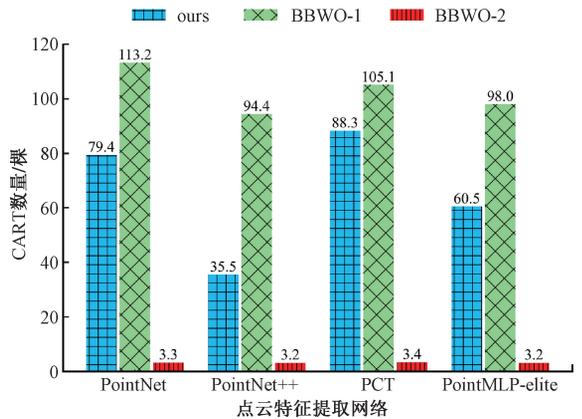


图 5 多目标和单目标优化集成剪枝在

ModelNet40 数据集上的集成规模对比

Fig. 5 Comparison of ensemble scale of ensemble pruning based on multi-objective and single objective optimization on the ModelNet40 dataset

从图 4 可以看出,使用  $\theta$  作为集成剪枝的目标比同时使用  $\theta$  和  $\overline{OA}$  的性能低 0.2%。结合图 5 进一步分析,当以  $\overline{OA}$  为目标进行集成剪枝时,集成规模会大幅减少,但总体精度较低。当以  $\theta$  为目标进行集成剪枝时,总体精度较高,但集成规模会增加。使用 BMOBWO 算法同时以  $\theta$  和  $\overline{OA}$  为目标进行集成剪枝,可以自动平衡二者的影响,既能保证基分类器间具有较大的差异,又能使基分类器组合具有较高的平均总体精度,并且大幅缩小集成规模,证明了多目标优化集成剪枝的性能优于单目标优化集成剪枝。

## 6 结 论

针对三维点云分类中由于点云数据的不规则性和无序性带来的特征提取和分类难问题,本文提出了一种基于 PointCloudTransformer 提取点云特征的三维点云分类算法,该算法使用集成学习进行分类。为了在集成剪枝过程中自动平衡基分类器的差异性和平均总体精度,本文将集成学习中的集成剪枝问题视为多目标组合优化问题,提出了二元多目标白鲸优化(BMOBWO)算法,以基分类器的难度度量和平均总体精度为目标进行多目标优化集成剪枝。在点云分类数据集上的实验结果表明,本文算法点云分类精度更高,集成规模更小。

## 参考文献

- [ 1 ] ZHAO Z, SONG Y, CUI F, et al. Point cloud features-based kernel SVM for human-vehicle classification in millimeter wave radar [ J ]. IEEE Access, 2020, 8: 26012-26021.
- [ 2 ] 胡海瑛, 惠振阳, 李娜. 基于多基元特征向量融合的机载 LiDAR 点云分类[J]. 中国激光, 2020, 47(8): 237-247.  
HU H Y, HUI ZH Y, LI N. Airborne LiDAR point cloud classification based on multiple-entity eigenvector fusion[J]. Chinese Journal of Lasers, 2020, 47(8): 237-247.
- [ 3 ] TIWARI G, GUPTA S. Fitness activity classification using mmwave radar point-cloud and machine learning[C]. 2023 IEEE Applied Sensing Conference (APSCON). IEEE, 2023: 1-3.
- [ 4 ] MOORTHY S M K, CALDERS K, VICARI M B, et al. Improved supervised learning-based approach for leaf and wood classification from LiDAR point clouds of forests[J]. IEEE Transactions on Geoscience and Remote Sensing, 2019, 58(5): 3057-3070.
- [ 5 ] QI C R, SU H, MO K, et al. Pointnet: deep learning on point sets for 3D classification and segmentation [ C ]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 652-660.
- [ 6 ] QI C R, YI L, SU H, et al. Pointnet ++: Deep hierarchical feature learning on point sets in a metric space [ C ]. Advances in Neural Information Processing Systems, 2017: 5099-5108.
- [ 7 ] ZHAO H, JIANG L, JIA J, et al. Point transformer [ C ]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021: 16259-16268.
- [ 8 ] MA X, QIN C, YOU H, et al. Rethinking network design and local geometry in point cloud: A simple residual MLP framework [ J ]. ArXiv preprint arXiv: 2202.07123, 2022.
- [ 9 ] 王忠民, 李和娜, 张荣, 等. 融合卷积神经网络与支持向量机的表情识别[J]. 计算机工程与设计, 2019, 40(12): 3594-3600.  
WANG ZH M, LI H N, ZHANG R, et al. Fusing convolutional neural network and support vector machine for expression recognition[J]. Computer Engineering and Design, 2019, 40(12): 3594-3600.
- [ 10 ] 马建, 邓晓刚, 王磊. 基于深度集成支持向量机的工业过程软测量方法[J]. 化工学报, 2018, 69(3): 1121-1128.  
MA J, DENG X G, WANG L. Industrial process soft sensor method based on deep learning ensemble support vector machine [ J ]. CIESC Journal, 2018, 69(3): 1121-1128.
- [ 11 ] 夏平凡, 倪志伟, 朱旭辉, 等. 基于双错测度的极限学习机选择性集成方法[J]. 电子与信息学报, 2020, 42(11): 2756-2764.  
XIA P F, NI ZH W, ZHU X H, et al. Selective ensemble method of extreme learning machine based on double-fault measure [ J ]. Journal of Electronics & Information Technology, 2020, 42(11): 2756-2764.
- [ 12 ] 孟军, 张晶, 姜丁菱, 等. 结合近邻传播聚类的选择性集成分类方法 [ J ]. 计算机研究与发展, 2018, 55(5): 986-993.  
MENG J, ZHANG J, JIANG D L, et al. Selective ensemble classification integrated with affinity propagation clustering [ J ]. Journal of Computer Research and Development, 2018, 55(5): 986-993.
- [ 13 ] ZHU X, NI Z, NI L, et al. Improved discrete artificial fish swarm algorithm combined with margin distance minimization for ensemble pruning [ J ]. Computers & Industrial Engineering, 2019, 128: 32-46.
- [ 14 ] HAMMOUDEH A, FRAIHAT M, ALMOMANI M. Selective ensemble model for telecom churn prediction [ C ]. 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT).

- IEEE, 2019: 485-487.
- [15] 朱旭辉, 倪志伟, 程美英, 等. 融合改进二元萤火虫算法和边界最小化测度的集成剪枝方法[J]. 计算机学报, 2019, 42(6): 1252-1273.  
ZHU X H, NI ZH W, CHENG M Y, et al. Improved binary glowworm swarm optimization combined with margin distance minimization for ensemble pruning[J]. Chinese Journal of Computers, 2019, 42 ( 6 ): 1252-1273.
- [16] GUO M H, CAI J X, LIU Z N, et al. PCT: Point cloud transformer[J]. Computational Visual Media, 2021, 7: 187-199.
- [17] 杨春, 殷绪成, 郝红卫, 等. 基于差异性的分类器集成: 有效性分析及优化集成[J]. 自动化学报, 2014, 40(4): 660-674.  
YANG CH, YIN X CH, HAO H W, et al. Classifier ensemble with diversity: Effectiveness analysis and ensemble optimization [ J ]. Acta Automatica Sinica, 2014, 40(4): 660-674.
- [18] ZHONG C, LI G, MENG Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm [ J ]. Knowledge-Based Systems, 2022, 251: 109215.
- [19] YU Y, GAO S, CHENG S, et al. CBSO: A memetic brain storm optimization with chaotic local search [ J ]. Memetic Computing, 2018, 10: 353-367.
- [20] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution algorithms [ C ]. 2006 IEEE International Conference on Evolutionary Computation. IEEE, 2006: 2010-2017.
- [21] ZHANG Q, ZHOU A, ZHAO S, et al. Multiobjective optimization test instances for the cec 2009 special session and competition [ J ]. University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-objective Optimization Algorithms, Technical Report, 2008, 264: 1-30.
- [22] SIERRA M R, COELLO COELLO C A. Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance [ C ]. Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, 2005: 505-519.
- [23] MOHAMMADI-BALANI A, NAYERI M D, AZAR A, et al. Golden eagle optimizer: A nature-inspired metaheuristic algorithm [ J ]. Computers & Industrial Engineering, 2021, 152: 107050.
- [24] MIRJALILI S, SAREMI S, MIRJALILI S M, et al. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization [ J ]. Expert Systems with Applications, 2016, 47: 106-119.
- [25] SESHADRI A. A fast elitist multiobjective genetic algorithm: NSGA-II [ J ]. MATLAB Central, 2006, 182: 182-197.
- [26] MIRJALILI S, JANGIR P, SAREMI S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems [ J ]. Applied Intelligence, 2017, 46: 79-95.
- [27] WU Z, SONG S, KHOSLA A, et al. 3D shapenets: A deep representation for volumetric shapes [ C ]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1912-1920.
- [28] UY M A, PHAM Q H, HUA B S, et al. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data [ C ]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 1588-1597.

### 作者简介



于喜俊, 2021 年于沈阳化工大学获得学士学位, 现为沈阳工业大学硕士研究生, 主要研究方向为计算机视觉、深度学习。

E-mail: yxj980920@163.com

**Yu Xijun** received his B. Sc. degree from Shenyang University of Chemical Technology in 2020. He is now a M. Sc. candidate at Shenyang University of Technology. His main research interests include computer vision and deep learning.



段勇 (通信作者), 沈阳工业大学信息科学与工程学院教授, 博士生导师, 主要研究方向为自主机器人、机器学习、计算机视觉。

E-mail: duanyong0607@126.com

**Duan Yong** (Corresponding author) is now a professor and Ph. D. supervisor at School of Information Science and Engineering, Shenyang University of Technology. His main research interests include autonomous robot, machine learning and computer vision.