# 面向集成学习的流形近邻样本包络与分层多类型变换算法

颜 芳,马 洁,李勇明*,王 品,覃 剑,刘承宇
(重庆大学微电子与通信工程学院,重庆 400044)

**摘 要:** 集成学习是机器学习领域的重要分支和研究热点. 目前集成学习算法的主要范式是:基于原样本集得到多个样本子集,分别训练基分类器,集成基分类器结果. 这种做法的主要问题在于:由于各子集均来自原样本集,因此,各子集之间的多样性显著降低. 尤其当原样本集数据尺寸小、采样比率大、不平衡程度高时,这一问题非常严重. 此外,当原样本集可分度低时,重采样获得的样本子集的可分度改善也有限. 为解决这个问题,本文提出面向集成学习的流形近邻样本包络与分层多类型变换算法,旨在通过包络化机制和多类型样本变换将原样本集转化为具有差异性的分层包络样本集,从而提高样本子集的多样性和可分度. 首先设计流形近邻样本包络化机制,将原样本转化为样本包络. 然后对样本包络进行多类型样本变换,重构生成分层包络样本. 接着,设计基于联合结构域适应的层间一致性保持机制,保持变换前后样本分布的一致性,提高包络样本对原样本的高表征能力. 此后,针对各层包络样本集,分别进行特征降维和训练基分类器. 最后,采用二维决策融合机制得到最终分类结果. 实验部分采用了十余个数据集和多个相关算法用于验证. 结果表明,相较于原样本集,本文算法构造的分层包络样本集提高了样本子集的多样性,改进了集成学习性能,准确率最高提升了 18.56%. 与相关集成学习算法相比,准确率最高提升了 7.56%. 本文工作为现有集成学习算法改进研究提供了新思路,将直接基于原样本的集成学习范式转化为基于分层包络样本的集成学习新范式,具有参考价值.

**关键词:** 集成学习;包络学习;样本变换;近邻样本包络化;域适应;分类器集成
**基金项目:** 国家自然科学基金(No.U21A20448,No.61771080)
**中图分类号:** TP391.4 **文献标识码:** A **文章编号:** 0372-2112(2024)12-4125-17
**电子学报 URL:**http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20231002

# Manifold Nearest Neighbor Sample Envelope and Hierarchical Multitype Transform Algorithm for Ensemble Learning

YAN Fang, MA Jie, LI Yong-ming*, WANG Pin, QIN Jian, LIU Cheng-yu
(*School of Microelectronics and Communication Engineering*, *Chongqing University*, *Chongqing* 400044, *China*)

**Abstract:** Ensemble learning is an important branch and research hotspot in machine learning. The current main paradigm of ensemble learning algorithms is to obtain multiple sample subsets based on the original sample set, then to train the base classifiers separately and integrate the base classifier results. The main problem of this paradigm is that the diversity among subsets is significantly reduced since all subsets are derived from the original sample set. This problem is especially serious when the data size of the original sample set is small, the sampling ratio is large, and the degree of imbalance is high. In addition, the improvement in the divisibility of the sample subsets obtained by resampling is also limited when the divisibility of the original sample set is low. In order to solve this problem, this paper proposes a manifold nearest neighbor sample envelope and hierarchical multitype transformation algorithm for ensemble learning. It aims to improve the diversity and divisibility of the sample subset by transforming the original sample set into a hierarchical enveloped sample set with differentiation through the envelopment mechanism and the multitype sample transformation. First, the manifold nearest neighbor sample envelope mechanism is designed to transform the original samples into sample envelopes. Second, a multitype sample transformation is performed on the sample envelope to reconstruct and generate hierarchical envelope samples.

收稿日期:2023-10-26;修回日期:2024-03-08;责任编辑:赵克
*通讯作者:李勇明

Third, the inter-layer consistency preservation mechanism based on joint structure domain adaptation is designed to preserve the distribution consistency of the samples before and after the transformation. Thus, improving the high representation ability of the envelope samples to the original samples. Four, feature dimensionality reduction and basic classifier training are performed separately for each layer of the envelope sample set. Finally, the final classification results are obtained using the two dimensional decision fusion mechanism. More than ten datasets and several representative algorithms are used in the experimental part for validation. The results show that compared with the original sample set, the proposed algorithm improves the diversity of the sample subsets, which improves the ensemble learning performance with up to 18.56% accuracy improvement. Compared with related ensemble learning algorithms, the accuracy of this paper's algorithm has been improved by up to 7.56%. This paper provides a new idea for the improvement of existing ensemble learning algorithms, and it is valuable to transform the paradigm of "ensemble learning directly based on original samples" into a new paradigm of "ensemble learning based on hierarchical envelope samples".

Key words: ensemble learning; envelope learning; sample transformation; nearest neighbor sample enveloping; domain adaptation; classifier ensemble

Foundation Item(s): National Natural Science Foundation of China (No.U21A20448, No.61771080)

## 1 引言

集成学习是一种通过构建并结合多个基分类器来完成学习任务的机器学习方法. 它的基本思想是通过组合多个弱分类器,以获得更好的分类器[1]. 相比单个分类器,集成学习具有更好的泛化性和稳定性,能降低误差和提高预测效果. 最近集成学习模型被广泛用于高维数据分类、带噪声数据分类、非平衡学习等[2,3]. 尤其是对结构化数据来说,集成学习具有较明显优势.

集成学习算法的主要模式包括Boosting、Bagging和Stacking等. Boosting算法的核心思想是将多个弱分类器组合起来,通过不断调整样本权重,使每个弱分类器都能更好分类,最终得到1个强分类器. 但这种模式很容易受到噪声的影响,产生过拟合现象[4]. Bagging算法的核心是利用可重复取样技术采样产生多个数据集,分别在这些数据集上训练基分类器,进而对多个基分类器进行组合得到更稳定的基分类器. Bagging提供了一种实现多样性的机制,但没有提到任何识别难以分类情况的机制,这为改进留下了空间[5]. Stacking是一种多级训练算法,核心思想在于利用训练数据集训练初级基分类器,将这些初级学习的输出看成是新的训练数据集然后在新的训练数据集上训练元分类器,但容易产生过拟合问题[6].

现有集成学习方法的主要范式为:基于原样本集构建多个子集,然后对每个子集分别进行预处理和基于每个子集训练基分类器,再融合每个基分类器结果. 这一范式存在1个问题:各子集中样本均直接来自同个原样本集,因此,子集的多样性提升有限且可分性易受到原样本集限制,从而制约后续基分类器的准确性. 尤其当原样本集尺寸小、采样比率大、不平衡程度高时,这一问题非常严重. 例如:

(1)当样本集可分度差(或类重叠严重)时,通过重采样构建的样本子集的可分度也较差,导致训练的基分类器的准确率不佳. 为了最小化训练误差,基分类器的决策曲线往往会很复杂,导致模型复杂度高、过拟合风险高.

(2)在非平衡率高、采样率高等情况下,划分的各子集之间重叠性高,多样性差,制约了集成学习性能.

由此可见,解决该问题的关键在于如何有效改造原样本集,提升样本子集的多样性和可分度.

## 2 动机和贡献

为了解决上述问题,人们探索通过在集成学习过程中引入随机性来提高分类模型的多样性和准确性[7~10],主要分支有:(1)样本数据扰动法;(2)特征空间扰动法;(3)输出表示扰动法;(4)算法参数扰动法;(5)模型结构多样性法. (2)~(5)的方法没有改造原样本集,因而难以有效提升样本子集的多样性和可分度. 虽然样本数据扰动法可以改善样本集质量,但该方法通过扰动引入的新样本为原样本个体的变化,没有挖掘样本间的深层次关联信息,因此改进效果有限. 此外,对样本个体的扰动程度难以精准把控,很容易与错分样本、噪声样本混淆.

此外,研究者们还提出了样本加权[11]、基于深度网络的样本生成[12]等解决方法. 这些研究成果有助于改善样本集质量,但仍存在一些局限性:(1)样本加权只是对样本个体或样本子集赋予不同权重,仍然只针对样本个体的变换;(2)生成法是让生成的样本接近原样本,因此,生成的新样本仍然无法摆脱原样本空间;(3)没有挖掘样本间深层次关联信息,而研究表明这些信息有助于提升分类准确性.

根据认知理论可知,人类认知特性具有多尺度、多粒度特性. 这意味着人们在识别事物时不仅依据事物本身,而且还依赖事物之间的关联关系. 人类大脑所接

收到的信息往往不是孤立存在的,而是相互关联的.人类通过对物体之间的关系、联系进行分析和推理,更好地识别出周围环境中的事物.例如,人类在识别草原时大脑会综合考虑草原所包含的各种特征,包括草地的颜色、形态等,同时,也会考虑草原不同区域之间的关联关系,例如草地和天空的分界线、草地和树木的关系等.因此,受人类的识别机制启发,有必要挖掘样本间的关联信息,用于改善样本集质量,从而有效提高集成分类准确性.

近年来,有研究者做了相关探索研究.Sakar等人[14]通过线性变换挖掘语料样本间关联信息,并转化为新样本,显著提升了分类准确率.Li等人[15]通过对同一受试者的语料样本进行多类型变换,构建新样本,显著提高识别准确率.Mohamed等人[16]通过挖掘相邻行人之间的关联关系构建邻接矩阵(新样本),用于行人预测,取得高准确率.Xia等人[17]通过挖掘样本间关联信息并转换为粒球样本,改进了原样本质量,提升了分类准确率.这些研究结果表明,挖掘样本间关联信息并将其转换为新样本可以改善原样本质量限制,有助于提升分类准确性.

聚类分析是常用于分析样本间信息的方法,其目的是通过同时最小化簇内距离和最大化簇间距离来发现数据集中的簇结构[13].通过挖掘相邻样本的关联关系,形成聚类中心(新样本),提高原样本的可分度和多样性.然而,在聚类过程中,可能会将一些在局部上相似的数据点分配到不同簇中,或者将一些局部上不相似的数据点分配到同个簇中.除聚类外,常见的样本关联信息挖掘方法还有线性和卷积变换等.线性变换可以挖掘样本间的关联信息[14],但当数据不服从或近似服从标准正态分布时,仅考虑线性变换的算法表现将会受到较大影响.卷积变换可以对输入数据进行局部线性变换和非线性激活操作,捕捉数据中的空间关系和结构信息,但其计算参数数量较大会增加模型复杂度.这些变换方法原理不同,优缺点不同,因此,结合多种类型的样本变换方法可以互相补充,对样本质量的改善取得更好效果.另外需要说明的是,如同深度特征变换一样[18],样本变换也可能产生与原样本有很大差异的新样本,因此,需要减小样本变换前后的分布差异,确保新样本集能正确反映原样本集本质,而非对原样本进行扭曲.

图1说明了多类型样本变换用于解决原样本集低质量问题的有效性.当原样本集可分度低、样本数少、采样率高时,构建的样本子集1和2的可分度低、多样性差.通过对原样本集进行样本变换形成了变换后新样本集,其可分度显著提高.此外,由于变换后新样本集与原样本集异构,因此,新样本子集与原样本子集具有明显差异性.在同样子集数量情况下,子集间的多样性将更好.

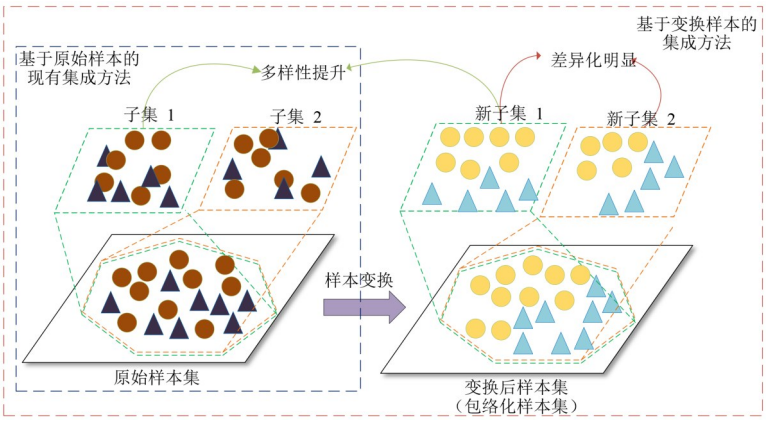



图1　基于原始样本的现有集成方法和基于变换样本的集成方法比较

基于以上分析,为解决目前集成学习算法面临的样本子集低质量问题,本文提出一种新的样本变换算法——流形近邻样本包络与分层多类型变换算法(Manifold Nearest Neighbor Sample Envelope and Hierarchical Multitype Transform,MNNSEHMT).该算法构造分层包络样本集,提高子集的多样性和可分度,有效解决上述问题.首先,设计流形近邻包络化机制为每个原样本构造样本包络.其次,针对样本包络设计多类型变换算子,构建分层包络样本集.然后,使用层间一致性保持机制来提升不同层包络样本集的表征能力.最后,基于上述构造的每层包络样本集,分别进行集成学习建模,形成新的面向集成学习的框架算法MNNSEHMT_EF(Ensemble leaning Framework algorithm based on MNNSEHMT).相比于现有原样本集,该算法构造的分层包络样本集有助于获得更高可分度和多样性;集成学习算法通过最佳权重融合各层样本集,获得更好建模效果.本文的主要贡献如下:

(1)分析了制约现有集成学习方法性能的共性关

键问题,即各子集来源于同个样本集,建模性能受限于原样本集的质量. 针对这个问题,本文提出了多类型样本变换来挖掘样本间深层关联信息,构造分层包络样本集,克服单一样本集的局限性,有效解决上述问题.

(2)提出了一种样本变换新算法——流形近邻样本包络与分层多类型变换算法,挖掘原样本间深层关联信息,将原样本集转换为分层包络样本集,为集成学习提供更具多样性的样本信息,有助于提高分类准确率.

(3)基于联合结构域适应,提出一种层间一致性保持机制. 考虑不同层样本集的局部和全局分布的不一致性,提高变换后包络样本集的表征能力.

(4)基于上述提出的样本变换新算法,形成一种新的集成学习框架算法——流形近邻样本包络分层多类型变换集成算法,为现有集成学习的相关研究提供新思路.

## 3 相关工作

为了便于本文的描述,本节将介绍现有增强集成学习多样性、改善样本质量以及样本间关联信息挖掘的相关方法.

集成多样性,即个体基分类器之间的差异,是影响集成学习分类算法性能的一个重要因素. 如果将相同的基分类器组合在一起,集成后的分类器性能将与基分类器性能一致,无法获得任何性能提升. 因此探索增强基分类器之间的多样性的方法是十分必要的. 在集成学习中,主要从数据、参数和模型结构方面增强多样性.

(1)样本扰动法,即根据原始数据产生多个不同种类的数据子集,然后利用不同的数子集训练基分类器. 常见的方法有重采样法. 在此基础上,Ngo等人[19]提出了进化集成bagging方法,利用进化算法来搅乱和更新子集数据.

(2)特征空间扰动法,即从初始特征集中抽取若干特征子集,再基于每个特征子集训练基分类器. 常见的方法有随机子空间算法,随机森林算法.

(3)输出表示扰动法,即对输出表示进行操纵从而增强多样性. 常见的算法有翻转法、输出调整法[20]、ECOC法[21].

(4)算法参数扰动法,即通过使用不同的参数集来产生不同的基分类器. 例如,Lee等人[9]使用不同参数值多次运行C4.5算法的策略来获得不同的决策树,以构建集成系统. Zhao等人[10]提出了BoostForest,在每次生成根节点时都会随机选择不同的超参数训练.

(5)模型结构多样性法,即使基分类器的内部结构或者外部结构不同. 常见的方法有异质集成.

此外,为了改善样本质量从而提高分类性能,有研究者还提出了基于样本加权和基于深度学习的样本生成算法. Shu等人[11]提取1个显式加权函数,以样本损失和任务/类别特征作为输入,样本权重作为输出,让不同样本自适应得到不同权值. Deng等人[22]基于学习全局分布信息生成新样本,但缺乏生成样本可靠性的保证机制. Goodfellow等人[23]通过生成器与鉴别器的对抗博弈提高了生成样本的可靠性,但存在训练困难、模式崩溃等问题. Meng等人[24]通过长短期记忆自动编码器网络生成新的样本,新生成样本与原始样本相比得到增强.

近年来,有部分学者探索考虑相似样本之间的相关性来进行样本变换,重构高质量的新样本. Sakar等人[14]采用简单的线性样本变换方法对同一受试者内的原始样本进行重构,从而实现分类性能的改进. Li等人[15]通过对同一受试者内已有样本进行多类型变换,构建新的样本. Mohamed等人[16]通过挖掘相邻行人之间的社会关系来进行行人预测,以构建邻接矩阵(新样本). Xia等人[17]通过相似度构造颗粒球(新样本)用于后续分类,实现高鲁棒学习. 但是,这些研究中样本变换算子比较单一,且未针对集成学习进行研究.

## 4 本文方法

### 4.1 主要符号说明

表1中列出了与所提出方法有关的符号解释.

**表1 主要符号解释**

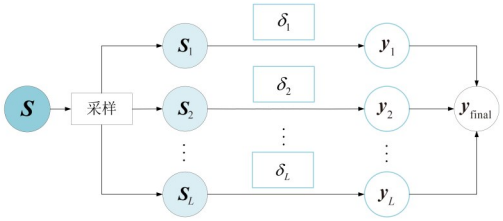
| 符号 | 解释 |
|---|---|
| $\boldsymbol{S}$ | 原始数据集(输入) |
| $\dot{\boldsymbol{S}}_i$ | 第$i$个样本的流形近邻样本包络 |
| $G$ | 样本的总数 |
| $N$ | 每个样本的特征数 |
| $L$ | 基分类器数 |
| $v$ | 每个样本包络的样本数量 |
| $\ell(\cdot)$ | 线性变换算子 |
| $\partial(\cdot)$ | 取整算子 |
| $\boldsymbol{D}$ | $\boldsymbol{S}$聚类后数据集 |
| $\varphi(\cdot)$ | 聚类算法 |
| $Q$ | 聚类中心数 |
| $\gamma(\cdot)$ | 卷积算子 |
| $\boldsymbol{E}_{\mathrm{A}},\boldsymbol{E}_{\mathrm{B}},\boldsymbol{E}_{\mathrm{C}}$ | A类型算子、B类型算子、C类型算子目标数据集 |
| $\boldsymbol{E}'_{\mathrm{A}},\boldsymbol{E}'_{\mathrm{B}},\boldsymbol{E}'_{\mathrm{C}}$ | $\boldsymbol{E}_{\mathrm{A}},\boldsymbol{E}_{\mathrm{B}},\boldsymbol{E}_{\mathrm{C}}$层间一致性保持机制的目标数据集 |
| $\boldsymbol{F}_{\mathrm{A}},\boldsymbol{F}_{\mathrm{B}},\boldsymbol{F}_{\mathrm{C}}$ | $\boldsymbol{E}'_{\mathrm{A}},\boldsymbol{E}'_{\mathrm{B}},\boldsymbol{E}'_{\mathrm{C}}$经过降维后的数据集 |
| $\Gamma(\cdot)$ | 矩阵扩展函数 |
| $\tilde{\boldsymbol{W}}$ | 带权邻接矩阵 |
| $\boldsymbol{R}$ | 距离矩阵 |
| $\boldsymbol{P}$ | 路径矩阵 |

### 4.2 问题建模

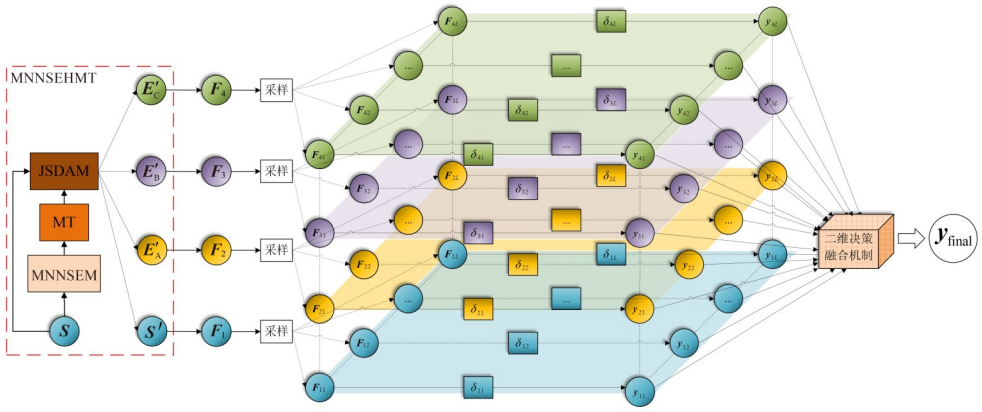
给定原始数据集 $\boldsymbol{S}=\begin{bmatrix}\bar{\boldsymbol{S}}_1\\ \bar{\boldsymbol{S}}_2\\ \vdots\\ \bar{\boldsymbol{S}}_G\end{bmatrix}=\begin{bmatrix}s_{11} & s_{12} & \cdots & s_{1N}\\ s_{21} & s_{22} & \cdots & s_{2N}\\ \vdots & \vdots & \vdots & \vdots\\ s_{G1} & s_{G2} & \cdots & s_{GN}\end{bmatrix}$ 作为整个集成模型的输入，其中 $\bar{\boldsymbol{S}}_i=\begin{bmatrix}s_{i1}\, s_{i2}\cdots s_{iN}\end{bmatrix}(i=1,2,\cdots,G)$ 表示 1 个样本. 样本对应的真实标签为 $\bar{\boldsymbol{y}}=(\bar{y}_1,\bar{y}_2,\cdots,\bar{y}_G)$. 现有集成学习算法通常基于 $\boldsymbol{S}$ 得到 $L$ 个子集 $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_L\}$，然后对每个子集进行处理和训练基分类器，利用组合策略得到最终预测结果 $\boldsymbol{y}_{\text{final}}$，最后计算得到错误率 $E$ 和多样性值 $D$. 本文要解决的问题为通过寻找合适的样本变换函数 trans(·)，提高样本子集的多样性和可分度，改善原样本低质量约束，即 min $E$ 和 min $D$. 具体而言，利用多个样本变换函数得到变换后的多层样本集 $\{\boldsymbol{S}_{\text{new}}^{(1)},\boldsymbol{S}_{\text{new}}^{(2)},\cdots,\boldsymbol{S}_{\text{new}}^{(K)}\}$，其中，$\boldsymbol{S}_{\text{new}}^{(1)}=\text{trans}^{(1)}(\boldsymbol{S}),\boldsymbol{S}_{\text{new}}^{(2)}=\text{trans}^{(2)}(\boldsymbol{S}),\cdots,\boldsymbol{S}_{\text{new}}^{(K)}=\text{trans}^{(K)}(\boldsymbol{S})$，分别基于 $\boldsymbol{S}_{\text{new}}^{(1)},\boldsymbol{S}_{\text{new}}^{(2)},\cdots,\boldsymbol{S}_{\text{new}}^{(K)}$ 采样得到 $(K+1)\times L$ 个子集 $\begin{bmatrix}\boldsymbol{S}_{11} & \boldsymbol{S}_{12} & \cdots & \boldsymbol{S}_{1L}\\ \boldsymbol{S}_{21} & \boldsymbol{S}_{22} & \cdots & \boldsymbol{S}_{2L}\\ \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{S}_{(K+1)1} & \boldsymbol{S}_{(K+1)2} & \cdots & \boldsymbol{S}_{(K+1)L}\end{bmatrix}$，对每个子集进行处理并训练 $(K+1)\times L$ 个基分类器，最后利用组合策略得到最终预测结果 $\boldsymbol{y}_{\text{final}}$.

### 4.3 算法设计

如前所述，本文拟解决的关键问题是如何通过样本变换来改善原样本低质量约束. 图 2($a$)显示了基于原始样本集的经典集成学习算法框架，通过采样方式构造样本子集，基于子集进行特征处理并训练子分类器 $\delta$，将子分类器结果融合得到最终预测标签 $\boldsymbol{y}_{\text{final}}$. 由于样本子集 $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_L\}$ 均采样来自于同一个样本集 $\boldsymbol{S}$，尤其是当 $\boldsymbol{S}$ 数据尺寸小，采样率高等情况下，样本子集 $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_L\}$ 的多样性较差. 此外，当 $\boldsymbol{S}$ 的可分度差时，样本子集 $\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_L\}$ 的可分度也难以有效改善. 因此，本文提出了一种流形近邻样本包络与分层多类型变换算法，如图 2($b$)所示. 首先，利用流形近邻包络化机制为每个样本构造流形样本包络，得到样本包络数据集. 其次，基于包络数据集使用多类型样本变换算子构建分层包络样本集. 然后，使用层间一致性保持机制保持样本变换前后的分布. 接着，对各层包络样本集采用主成分分析(Principal Component Analysis，PCA)进行特征降维，基于降维后的多个数据集采样训练多个基分类器. 最后，利用二维决策融合机制得到最终分类预测结果.




($a$) 现有的基于原始样本的集成学习框架




($b$) 本文算法主要框架

图 2　现有集成学习框架与本文算法对比

### 4.4 流形近邻样本包络与分层多类型变换算法

#### 4.4.1 流形近邻样本包络化机制

流形近邻样本包络化机制(Manifold Nearest Neighbor Sample Envelopment Mechanism, MNNSEM)通过Floyed算法计算各样本之间的流形距离,选择原样本和其$v-1$个近邻样本构造包络样本,得到的每个样本包络有$v$个样本. 为每个样本$\bar{\boldsymbol{S}}_i$构造近邻样本包络$\dot{\boldsymbol{S}}_i=\text{envelop}(\bar{\boldsymbol{S}}_i)$.

Floyed算法通过在图的带权邻接矩阵$\tilde{\boldsymbol{W}}$中用插入顶点的方法依次递推地构造距离矩阵$\boldsymbol{R}$,计算出各个样本之间的流形空间距离,得到最短路径. 具体流程为,先计算带权邻接矩阵$\tilde{\boldsymbol{W}}$,其中矩阵元素$w_{ij}$可根据$\bar{\boldsymbol{S}}_i$和$\bar{\boldsymbol{S}}_j$是否相邻设为欧式距离或无穷大;其次,将$\tilde{\boldsymbol{W}}$作为距离矩阵$\boldsymbol{R}$的初始值$\boldsymbol{R}^0=\tilde{\boldsymbol{W}}$,矩阵元素$r_{ij}^0=w_{ij}$. 进行第1次迭代,距离矩阵为$\boldsymbol{R}^1$,矩阵元素$r_{ij}^1=\min\left\{r_{ij}^0,r_{i1}^0+r_{1j}^0\right\}$表示从样本点$i\to j$只以样本点1作为中间点的路径最短路长度. 进行第2次迭代,距离矩阵为$\boldsymbol{R}^2$,矩阵元素$r_{ij}^2$表示从样本点$i\to j$以样本点1和样本点2作为中间点的路径最短路长度.

重复上述步骤遍历所有样本点,得到距离矩阵$\boldsymbol{R}^{\tilde{G}}$,矩阵元素为$r_{ij}^{\tilde{G}}=\min\left\{r_{ij}^{\tilde{G}-1},r_{iG}^{\tilde{G}-1}+r_{Gj}^{\tilde{G}-1}\right\}$表示从样本点$i\to j$的最短路长度. 其中,最短距离可根据下式计算:

$$r_{ij}=\begin{Bmatrix}\left\|\bar{\boldsymbol{S}}_i-\bar{\boldsymbol{S}}_j\right\|^2, & \text{if } \bar{\boldsymbol{S}}_i,\bar{\boldsymbol{S}}_j, & \text{邻居}\\ \min\{r_{ij},r_{ik}+r_{kj}\}, & & \text{其他}\end{Bmatrix} \tag{1}$$

在迭代距离矩阵$\boldsymbol{R}$的同时定义1个路径矩阵$\boldsymbol{P}$来记录所插入点的信息,初始化矩阵元素$p_{ij}^0=j$. 第$g$次迭代中相应$P^g$的生成表达式如下:

$$p_{ij}^g=\begin{cases}g, & \text{if } r_{ij}^{g-1}\geqslant r_{ig}^{g-1}+r_{gj}^{g-1}\\ p_{ij}^{g-1}, & \text{else}\end{cases} \tag{2}$$

若$p_{ij}^g=a_1$,则样本点$i\to j$的最短路径下1个样本点为$a_1$. 假设$p_{a_1j}^g=a_2,p_{a_2j}^g=a_3,\cdots,p_{a_mj}^g=j$,则样本点$i\to j$对应的最短路径为$i\to a_1\to a_2\to\cdots\to a_m\to j$.

MNNSEM如图3所示,图4给出原始样本集和近邻包络化样本集的样本空间分布图,不同颜色的点表示不同类别的样本.

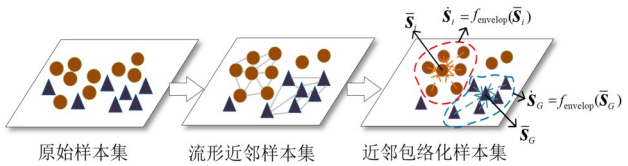
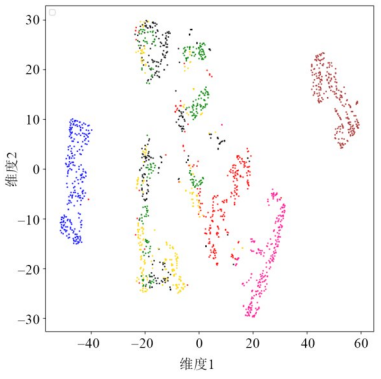
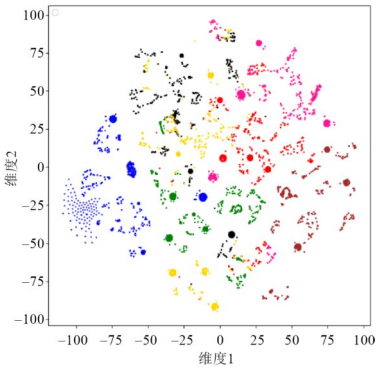



图3 MNE的示意图




(a) 原始样本集




(b) 近邻包络化样本集

图4 样本空间图

#### 4.4.2 多类型样本变换算子(Multitype Transform,MT)

(1)A类型变换算子——线性样本变换

基于近邻样本包络数据集$\dot{\boldsymbol{S}}$使用A类型算子获得数据集$\boldsymbol{E}_{\text{A}}=[\boldsymbol{E}_{\text{A1}}\quad\boldsymbol{E}_{\text{A2}}\quad\cdots\quad\boldsymbol{E}_{\text{AG}}]^{\text{T}}$,其中,每个样本表达式如下:

$$\boldsymbol{E}_{\text{A}i}=\ell\left(\dot{\boldsymbol{S}}_i\right)=\begin{bmatrix}\tilde{\boldsymbol{E}}_1\\ \tilde{\boldsymbol{E}}_2\\ \vdots\\ \tilde{\boldsymbol{E}}_6\end{bmatrix},(i=1,2,\cdots,G) \tag{3}$$

其中,$\ell(\cdot)$表示线性算子计算样本特征的集中趋势和离散程度,这6项指标为均值、中位数、裁剪均值、标准差、四分位距离和均值绝对误差,表达式分别如下:

$$\tilde{\boldsymbol{E}}_1=\frac{1}{v}\sum_{i=1}^{v}\bar{\boldsymbol{S}}_i \tag{4}$$

$$\tilde{\boldsymbol{E}}_2=\begin{Bmatrix}\bar{\boldsymbol{S}}_{(v+1)/2}, & v\text{是偶数}\\ \dfrac{\bar{\boldsymbol{S}}_{(v/2)}+\bar{\boldsymbol{S}}_{(v/2+1)}}{2}, & v\text{不是偶数}\end{Bmatrix} \tag{5}$$

$$\tilde{\boldsymbol{E}}_3=\frac{1}{v}\sum_{i=\partial\left(\frac{v}{4}\right)}^{v-\partial(v/4)}\boldsymbol{S}_i \tag{6}$$

其中,$\partial(\cdot)$为取整计算.

$$\tilde{\boldsymbol{E}}_4=\sqrt{\frac{\sum_{i=1}^{v}\left(\bar{\boldsymbol{S}}_i-\tilde{\boldsymbol{E}}_1\right)\cdot\left(\bar{\boldsymbol{S}}_i-\tilde{\boldsymbol{E}}_1\right)}{v-1}} \tag{7}$$

其中,"·"为点乘计算.

$$\tilde{\boldsymbol{E}}_5=\bar{\boldsymbol{S}}_{\partial\left(\frac{3}{4}v\right)}-\bar{\boldsymbol{S}}_{\partial\left(\frac{v}{4}\right)} \tag{8}$$

$$\tilde{\boldsymbol{E}}_6=\frac{1}{v}\sum_{i=1}^{v}\left|\bar{\boldsymbol{S}}_i-\tilde{\boldsymbol{E}}_1\right| \tag{9}$$

A类型算子示意图如图5所示.

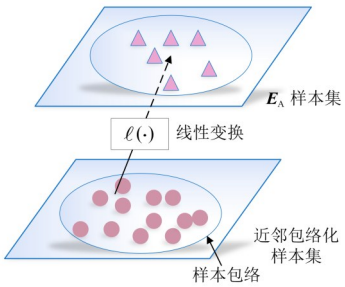



图5　A类型算子示意图

(2)B类型算子——聚类样本变换

通过$B$类型算子计算包络数据集$\dot{\boldsymbol{S}}$获得目标数据集$\boldsymbol{E}_{\mathrm{B}}$,该算子首先使用K均值聚类算法获得聚类后包络化样本集$\boldsymbol{D}=\left[\varphi\left(\dot{\boldsymbol{S}}_1\right)\quad\varphi\left(\dot{\boldsymbol{S}}_2\right)\quad\cdots\quad\varphi\left(\dot{\boldsymbol{S}}_G\right)\right]^{\mathrm{T}}$,$\boldsymbol{D}_i$的表示如下

$$\boldsymbol{D}_i=\varphi\left(\dot{\boldsymbol{S}}_i\right)=\begin{bmatrix}\hat{\boldsymbol{D}}_i^1\\ \hat{\boldsymbol{D}}_i^2\\ \vdots\\ \hat{\boldsymbol{D}}_i^Q\end{bmatrix},(i=1,2,\cdots,G) \tag{10}$$

其中,$\varphi(\cdot)$表示K均值聚类算法的计算,$\hat{\boldsymbol{D}}_i^Q$为第$i$个包络样本的第$q(q=1,2,\cdots,Q)$个聚类样本集,$Q$为每个包络样本的聚类中心数.

K均值聚类算法是无监督学习的杰出代表算法,其基本思想是根据数据的内在关系,将数据划分若干簇类,增大类内距离的同时减少类间距离. 在向量空间模型中,最常用的相似性度量是欧几里得距离,其表达如式(11)所示:

$$\mathrm{dist}\left(\bar{\boldsymbol{S}}_i,\bar{\boldsymbol{S}}_j\right)=\sqrt{\left(\bar{\boldsymbol{S}}_i-\bar{\boldsymbol{S}}_j\right)\left(\bar{\boldsymbol{S}}_i-\bar{\boldsymbol{S}}_j\right)^{\mathrm{T}}} \tag{11}$$

其中,$\bar{\boldsymbol{S}}_i$和$\bar{\boldsymbol{S}}_j$为2个单独样本. 平方和误差计算如下:

$$\mathrm{SSE}=\sum_{q=1}^{Q}\sum_{\bar{\boldsymbol{S}}_i\in\boldsymbol{K}_q}\left\|\bar{\boldsymbol{S}}_i-\boldsymbol{c}_q\right\|^2 \tag{12}$$

其中,$\boldsymbol{c}_q$表示第$q$个聚类中心,$\boldsymbol{K}_q$是以$\boldsymbol{c}_q$为聚类中心的样本集. 聚类中心的更新表达式为

$$\boldsymbol{c}_q=\frac{\sum_{\bar{\boldsymbol{S}}_i\in\boldsymbol{K}_q}\bar{\boldsymbol{S}}_i}{\left|\boldsymbol{K}_q\right|} \tag{13}$$

其中,$\left|\boldsymbol{K}_q\right|$为样本集$\boldsymbol{K}_q$的样本数.

然后,基于聚类后的包络化样本集$\boldsymbol{D}$使用线性算子获得目标数据集$\boldsymbol{E}_{\mathrm{B}}=\left[\boldsymbol{E}_{\mathrm{B1}}\quad\boldsymbol{E}_{\mathrm{B2}}\quad\cdots\quad\boldsymbol{E}_{\mathrm{B}G}\right]^{\mathrm{T}}$,$\boldsymbol{E}_{\mathrm{B}i}$的表达式为

$$\boldsymbol{E}_{\mathrm{B}i}=\ell\left(\boldsymbol{\phi}\left(\dot{\boldsymbol{S}}_i\right)\right)=\ell\left(\boldsymbol{D}_i\right)=\begin{bmatrix}\ell\left(\hat{\boldsymbol{D}}_i^1\right)\\ \ell\left(\hat{\boldsymbol{D}}_i^2\right)\\ \vdots\\ \ell\left(\hat{\boldsymbol{D}}_i^Q\right)\end{bmatrix},(i=1,2,\cdots,G) \tag{14}$$

B类型算子示意图如图6所示.

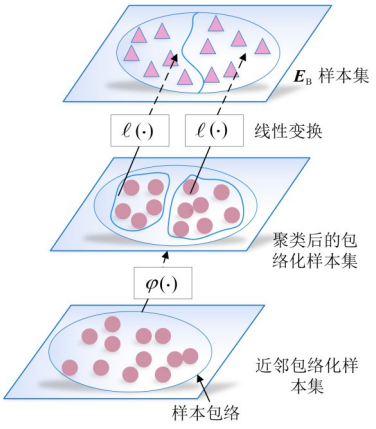



图6　B类型算子示意图

(3)C类型算子 - 卷积样本变换

C类型算子计算数据集$\dot{\boldsymbol{S}}$获得目标数据集$\boldsymbol{E}_C$. 首先,聚类后的包络化样本集$\boldsymbol{D}$利用A类型算子进行线性变换得到$\boldsymbol{E}_{\mathrm{A}}=\ell(\boldsymbol{D})$,然后将$\boldsymbol{D}$和$\boldsymbol{E}_{\mathrm{B}}$使用卷积变换算子计算得到数据集$\boldsymbol{E}_{\mathrm{C}}=\left[\boldsymbol{E}_{\mathrm{C1}}\quad\boldsymbol{E}_{\mathrm{C2}}\quad\cdots\quad\boldsymbol{E}_{\mathrm{C}G}\right]^{\mathrm{T}}$,$\boldsymbol{E}_{\mathrm{C}i}$的表达式为

$$\begin{aligned}\boldsymbol{E}_{\mathrm{C}i}&=\gamma\left(\boldsymbol{D}_i,\ell\left(\boldsymbol{D}_i\right)\right)=\gamma\left(\boldsymbol{D}_i,\boldsymbol{E}_{\mathrm{B}i}\right)\\&=\begin{bmatrix}\gamma\left(\hat{\boldsymbol{D}}_i^1,\ell\left(\hat{\boldsymbol{D}}_i^1\right)\right)\\ \gamma\left(\hat{\boldsymbol{D}}_i^2,\ell\left(\hat{\boldsymbol{D}}_i^2\right)\right)\\ \vdots\\ \gamma\left(\hat{\boldsymbol{D}}_i^Q,\ell\left(\hat{\boldsymbol{D}}_i^Q\right)\right)\end{bmatrix}=\begin{bmatrix}\boldsymbol{E}_{\mathrm{C}i}^1\\ \boldsymbol{E}_{\mathrm{C}i}^2\\ \vdots\\ \boldsymbol{E}_{\mathrm{C}i}^Q\end{bmatrix},(i=1,2,\cdots,G)\end{aligned} \tag{15}$$

其中,$\gamma(\cdot)$表示卷积算子,$\boldsymbol{E}_{\mathrm{C}i}^q$表示聚类后第$i$个包络样本中第$q(q=1,2,\cdots,Q)$类簇样本线性变换前后的卷积计算结果. $\boldsymbol{E}_{\mathrm{C}i}^q$的表达式为

$$\boldsymbol{E}_{Ci}^{q}=\gamma\left(\hat{\boldsymbol{D}}_{i}^{q},\ell\left(\hat{\boldsymbol{D}}_{i}^{q}\right)\right)=\gamma(\boldsymbol{U},\boldsymbol{T})=\left[\overline{\boldsymbol{E}_{1}'}\quad\overline{\boldsymbol{E}_{2}'}\quad\cdots\quad\overline{\boldsymbol{E}_{\tilde{m}}'}\quad\cdots\quad\overline{\boldsymbol{E}_{\tilde{M}}'}\right]^{\mathrm{T}}\tag{16}$$

其中，

$$\boldsymbol{U}=\begin{bmatrix}u_{11}&u_{12}&\cdots&u_{1N}\\u_{21}&u_{22}&\cdots&u_{2N}\\\vdots&\vdots&\vdots&\vdots\\u_{\tilde{I}1}&u_{\tilde{I}2}&\cdots&u_{\tilde{I}N}\end{bmatrix}=\begin{bmatrix}\bar{\boldsymbol{U}}_{1}\\\bar{\boldsymbol{U}}_{2}\\\vdots\\\bar{\boldsymbol{U}}_{\tilde{I}}\end{bmatrix}=[\boldsymbol{u}_{1}\quad\boldsymbol{u}_{2}\quad\cdots\quad\boldsymbol{u}_{N}],$$

$$\boldsymbol{T}=\begin{bmatrix}t_{11}&t_{12}&\cdots&t_{1N}\\t_{21}&t_{22}&\cdots&t_{2N}\\\vdots&\vdots&\vdots&\vdots\\t_{\tilde{M}1}&t_{\tilde{M}2}&\cdots&t_{\tilde{M}N}\end{bmatrix}=\begin{bmatrix}\bar{\boldsymbol{T}}_{1}\\\bar{\boldsymbol{T}}_{2}\\\vdots\\\bar{\boldsymbol{T}}_{\tilde{M}}\end{bmatrix}=[\boldsymbol{t}_{1}\quad\boldsymbol{t}_{2}\quad\cdots\quad\boldsymbol{t}_{N}],$$

$\boldsymbol{u}_{i}=[u_{1i}\quad u_{2i}\quad\cdots\quad u_{\tilde{I}i}]^{\mathrm{T}}$，$\boldsymbol{t}_{i}=[t_{1i}\quad t_{2i}\quad\cdots\quad t_{\tilde{M}i}]^{\mathrm{T}}$，$\tilde{I}$ 和 $\tilde{M}$ 分别为要进行卷积计算的2个数据集样本数.

$\overline{\boldsymbol{E}_{\tilde{m}}'}$ 的表达式为

$$\overline{\boldsymbol{E}_{\tilde{m}}'}=\sum_{i=1}^{\tilde{I}}\bar{\boldsymbol{T}}_{\tilde{m}}\cdot\boldsymbol{U}_{i},(\tilde{m}=1,2,\cdots,\tilde{M})\tag{17}$$

式(17)可进一步简化表示为

$$\overline{\boldsymbol{E}_{\tilde{m}}'}=\boldsymbol{u}'\widehat{\boldsymbol{t}}_{\tilde{m}}\tag{18}$$

其中，$\boldsymbol{u}'=[\boldsymbol{u}_{1}^{\mathrm{T}}\quad\boldsymbol{u}_{2}^{\mathrm{T}}\quad\cdots\quad\boldsymbol{u}_{N}^{\mathrm{T}}]$，$\widehat{\boldsymbol{t}}_{\tilde{m}}$ 由对矩阵 $\tilde{\boldsymbol{T}}_{\tilde{m}}$ 的列元素进行对角排列扩展得到. 而 $\widehat{\boldsymbol{T}}_{\tilde{m}}$ 是将 $1\times N$ 维度的矩阵 $\bar{\boldsymbol{T}}_{\tilde{m}}$ 通过复制扩展成 $\tilde{I}\times N$ 维度矩阵 $\tilde{\boldsymbol{T}}_{\tilde{m}}=\Gamma\left(\bar{\boldsymbol{T}}_{\tilde{m}}\right)=\begin{bmatrix}\bar{\boldsymbol{T}}_{\tilde{m}}\\\bar{\boldsymbol{T}}_{\tilde{m}}\\\vdots\\\bar{\boldsymbol{T}}_{\tilde{m}}\end{bmatrix}$.

进一步推导，式(16)的表达为

$$\boldsymbol{E}_{Ci}^{q}=[\mathrm{reshape}(\boldsymbol{u}'\widehat{\boldsymbol{T}})]^{\mathrm{T}}\tag{19}$$

其中，$\widehat{\boldsymbol{T}}=\left[\widehat{\boldsymbol{t}}_{1}\quad\widehat{\boldsymbol{t}}_{2}\quad\cdots\quad\widehat{\boldsymbol{t}}_{\tilde{m}}\quad\cdots\quad\widehat{\boldsymbol{t}}_{\tilde{M}}\right]$，$\mathrm{reshape}(\cdot)$ 表示将矩阵元素按照行优先的顺序重新排列成维度为 $N\times\tilde{M}$ 的矩阵.

C类型算子示意图如图7所示.

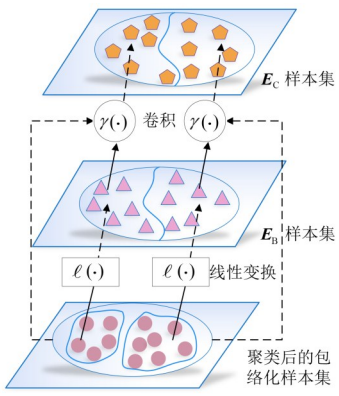



图7 C类型算子示意图

### 4.4.3 基于联合结构域适应的层间一致性保持机制

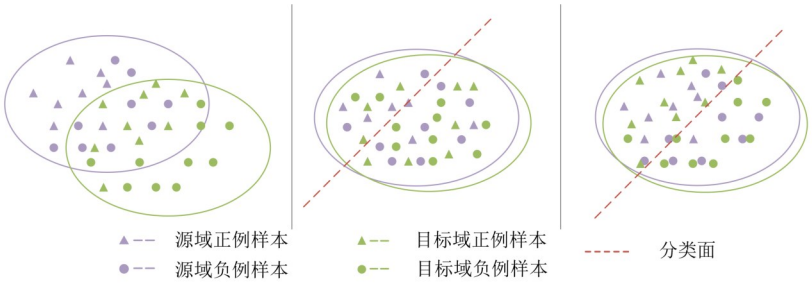
将变换前后的样本集分别作为源域和目标域，为保障变换后新样本集的高表征能力，设计了基于联合结构域适应的层间一致性保持机制对齐目标域与源域间的全局和局部分布. 将原始数据集 $\boldsymbol{S}$ 作为源域，将A类型变换算子变换后数据集 $\boldsymbol{E}_{\mathrm{A}}$，B类型变换算子变换后数据集 $\boldsymbol{E}_{\mathrm{B}}$ 和C类型变换算子变换后数据集 $\boldsymbol{E}_{\mathrm{C}}$ 分别作为目标域，使用基于联合结构域适应的层间一致性保持机制(Joint Structure Domain Adaptation Menchanism，JSDAM)获得域适应后的源域 $\boldsymbol{S}'$ 和目标数据集 $\boldsymbol{E}_{\mathrm{A}}'$、$\boldsymbol{E}_{\mathrm{B}}'$ 和 $\boldsymbol{E}_{\mathrm{C}}'$.

源域样本用 $\tilde{\boldsymbol{O}}$ 表示，目标域样本用 $\tilde{\boldsymbol{V}}$ 来表示. 无监督域适应一般假设：目标域和源域两者的条件概率分布相等(表示为 $Q_{\boldsymbol{O}}=Q_{\boldsymbol{V}}$)，但目标域和源域两者的边缘概率分布不相等(表示为 $P_{\boldsymbol{O}}\neq P_{\boldsymbol{V}}$). 数据分布对齐需要满足条件，变换后的源域和目标域的边缘概率分布相似 $P_{\boldsymbol{O}}'\approx P_{\boldsymbol{V}}'$ 且条件概率分布也相似 $Q_{\boldsymbol{O}}'\approx Q_{\boldsymbol{V}}'$，即最小化源域与目标域的样本中心距离，如式(20)第1项所示. 式(20)第2项和3项描述的是源域和目标域样本之间的局部结构信息. 目标函数可以表示如式(20)所示：

$$\begin{aligned}J_{\mathrm{JSDAM}}=\Bigg(&\left\|\frac{1}{n_{O}}\sum_{i=1}^{n_{O}}\boldsymbol{\phi}(\tilde{\boldsymbol{o}}_{i})-\frac{1}{n_{V}}\sum_{j=1}^{n_{V}}\boldsymbol{\phi}(\tilde{\boldsymbol{v}}_{j})\right\|_{\mathrm{H}}^{2}\\&+\frac{1}{2}\sum_{m=1}^{n_{O}}\sum_{n=1}^{n_{O}}\left\|\boldsymbol{\phi}(\tilde{\boldsymbol{o}}_{m})-\boldsymbol{\phi}(\tilde{\boldsymbol{o}}_{n})\right\|_{\mathrm{H}}^{2}\hat{S}_{mn}\\&+\frac{1}{2}\sum_{p=1}^{n_{V}}\sum_{d=1}^{n_{V}}\left\|\boldsymbol{\phi}(\tilde{\boldsymbol{v}}_{p})-\boldsymbol{\phi}(\tilde{\boldsymbol{v}}_{d})\right\|_{\mathrm{H}}^{2}\tilde{S}_{pd}\Bigg)\end{aligned}\tag{20}$$

其中，$\tilde{\boldsymbol{O}}=\left[\tilde{\boldsymbol{o}}_{1}\quad\tilde{\boldsymbol{o}}_{2}\quad\cdots\quad\tilde{\boldsymbol{o}}_{n_{o}}\right]^{\mathrm{T}}$ 和 $\tilde{\boldsymbol{V}}=\left[\tilde{\boldsymbol{v}}_{1}\quad\tilde{\boldsymbol{v}}_{2}\quad\cdots\quad\tilde{\boldsymbol{v}}_{n_{v}}\right]^{\mathrm{T}}$；$n_{\boldsymbol{O}}$ 和 $n_{\boldsymbol{V}}$ 分布为源域和目标域的样本数量；$\|\cdot\|_{\mathrm{H}}$ 是希尔伯特空间范数；$\boldsymbol{\phi}(\cdot)$ 为映射函数；$\hat{S}_{mn}$ 和 $\tilde{S}_{pd}$ 分别表示源域和目标域样本的亲和矩阵元素值，且 $m$ 与 $n$ 不相等，$p$ 与 $d$ 不相等.

图8是JSDAM的工作示意图. 图中紫色与绿色分别代表来自源域和目标域的样本，圆形和三角形分别代表负类和正类样本. 图8($a$)表示不同类别样本在源域和目标域中的数据分布情况，各域中样本类别区分度较高；图8($b$)给出了源域和目标域仅分布对齐后的样本分布和域分布情况，可以看出仅对齐分布而不保留样本邻域结构关系，分类边界划分受到影响；图8($c$)显示了JSDAM之后的样本分布和域分布，所有样本在域对齐后仍保持图8($a$)中原始的近邻关系，使得在缩小域间差距的同时又保留原始数据信息，使分类器更好地分类.




(*a*) 原始数据源域与目标域的数据分布　(*b*) 仅对齐两域后的数据分布　(*c*) 联合结构域适应后的数据分布

图8　联合结构域适应示意图

目标 $P'_O \approx P'_V$ 且 $Q'_O \approx Q'_V$，假设通过1个映射 $\boldsymbol{\phi}(\cdot)$ 可以得到 $P_O\left(\boldsymbol{\phi}(\tilde{\boldsymbol{O}})\right) \approx P_V\left(\boldsymbol{\phi}(\tilde{\boldsymbol{V}})\right)$ 且 $Q_O\left(\boldsymbol{Y}_O \middle| \boldsymbol{\phi}(\tilde{\boldsymbol{O}})\right) \approx Q_V\left(\boldsymbol{Y}_V \middle| \boldsymbol{\phi}(\tilde{\boldsymbol{V}})\right)$. 问题核心是最小化数据域之间的距离度量进而使分布变得一致. 参考迁移成分分析(Transfer Component Analysis,TCA),使用最大均值差异(Maximum Mean Discrepancy,MMD)来衡量2个分布之间的距离,写为

$$\operatorname{dist}(\tilde{\boldsymbol{O}}, \tilde{\boldsymbol{V}}) = \left\| \frac{1}{n_O} \sum_{i=1}^{n_O} \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_i) - \frac{1}{n_V} \sum_{j=1}^{n_V} \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_j) \right\|_{\mathrm{H}}^2 \tag{21}$$

通过最小化式(21)就可以找出映射 $\boldsymbol{\phi}(\cdot)$,但是直接求解上式很难得出结果而且容易使 $\boldsymbol{\phi}(\cdot)$ 陷入局部最小值. 通过学习内核矩阵 $\hat{\boldsymbol{K}}$,将源域和目标域嵌入到公共的低维空间中. 具体来说,经过映射后的源域 $\tilde{\boldsymbol{O}}$ 和目标域 $\tilde{\boldsymbol{V}}$ 可以表示成 $\begin{bmatrix} \left\langle \boldsymbol{\phi}(\tilde{\boldsymbol{O}})\ \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \right\rangle & \left\langle \boldsymbol{\phi}(\tilde{\boldsymbol{O}})\ \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \right\rangle \\ \left\langle \boldsymbol{\phi}(\tilde{\boldsymbol{V}})\ \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \right\rangle & \left\langle \boldsymbol{\phi}(\tilde{\boldsymbol{V}})\ \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \right\rangle \end{bmatrix}$,那么 $\hat{\boldsymbol{K}} = \begin{bmatrix} \hat{\boldsymbol{K}}_{O,O} & \hat{\boldsymbol{K}}_{O,V} \\ \hat{\boldsymbol{K}}_{V,O} & \hat{\boldsymbol{K}}_{V,V} \end{bmatrix}$. 再对其求迹,得出源域和目标域之间的样本距离等于 $\operatorname{tr}(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}})$,且服从约束 $\hat{\boldsymbol{K}}$. $\hat{\boldsymbol{M}}$ 是MMD矩阵,其矩阵元素如下所示:

$$M_{ij} = \begin{cases} \dfrac{1}{n_O n_O}, \tilde{\boldsymbol{o}}_i, \tilde{\boldsymbol{o}}_j \in \tilde{\boldsymbol{O}} \\ \dfrac{1}{n_V n_V}, \tilde{\boldsymbol{v}}_i, \tilde{\boldsymbol{v}}_j \in \tilde{\boldsymbol{V}} \\ -\dfrac{1}{n_O n_V}, \text{其他}. \end{cases} \tag{22}$$

减小域间的边缘概率分布差异可能会破坏样本之间原始的关系结构,从而导致有用信息的丢失. 因此,基于局部保留投影算法(Locality Preserving Projections,LPP)中保留数据邻域结构的思想,引入亲和力矩阵获取样本的局部结构信息. 源域和目标域的保持邻域结构定义为

$$\frac{1}{2} \sum_{m=1}^{n_O} \sum_{n=1}^{n_O} \left\| \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_m) - \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_n) \right\|_{\mathrm{H}}^2 \hat{S}_{mn} + \frac{1}{2} \sum_{p=1}^{n_V} \sum_{d=1}^{n_V} \left\| \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_p) - \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_d) \right\|_{\mathrm{H}}^2 \tilde{S}_{pd} \tag{23}$$

上述就是JSDAM的2个部分,第1部分使源域和目标域的分布保持一致,第2部分保留样本原始邻域结构. 前者在较大范围内减小源域和目标域之间的分布差异,以便分类器可以更好地匹配数据. 后者保留局部范围内每个域中样本之间的近邻结构,使原始有效信息不受域对齐的影响. 因此,将齐域分布与局部结构保留结合在一起. JSDAM目标函数式(20)可以进一步表示为

$$\begin{aligned} J_{\mathrm{JSDAM}} = \operatorname{tr}\Bigg( & \frac{1}{n_O^2} \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \boldsymbol{I} \boldsymbol{I}^{\mathrm{T}} \boldsymbol{\phi}(\tilde{\boldsymbol{O}})^{\mathrm{T}} + \frac{1}{n_V^2} \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \boldsymbol{I} \boldsymbol{I}^{\mathrm{T}} \boldsymbol{\phi}(\tilde{\boldsymbol{V}})^{\mathrm{T}} \\ & - \frac{1}{n_O n_V} \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \boldsymbol{I} \boldsymbol{I}^{\mathrm{T}} \boldsymbol{\phi}(\tilde{\boldsymbol{V}})^{\mathrm{T}} - \frac{1}{n_O n_V} \boldsymbol{\phi}(\tilde{\boldsymbol{O}})^{\mathrm{T}} \boldsymbol{I}^{\mathrm{T}} \boldsymbol{I} \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \Bigg) \\ & + \frac{1}{2} \sum_{m=1}^{n_O} \sum_{n=1}^{n_O} \operatorname{tr}\Big( \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_m) \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_m)^{\mathrm{T}} + \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_n) \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_n)^{\mathrm{T}} \\ & - \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_m) \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_n)^{\mathrm{T}} - \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_n) \boldsymbol{\phi}(\tilde{\boldsymbol{o}}_m)^{\mathrm{T}} \Big) \hat{S}_{mn} \\ & + \frac{1}{2} \sum_{p=1}^{n_V} \sum_{d=1}^{n_V} \operatorname{tr}\Big( \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_p) \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_p)^{\mathrm{T}} + \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_d) \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_d)^{\mathrm{T}} \\ & - \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_p) \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_d)^{\mathrm{T}} - \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_d) \boldsymbol{\phi}(\tilde{\boldsymbol{v}}_p)^{\mathrm{T}} \Big) \tilde{S}_{pd} \end{aligned} \tag{24}$$

根据矩阵求迹的性质和先前的定义,可以将式(24)简化为式(25):

$$\begin{aligned} J_{\mathrm{JSDAM}} = & \operatorname{tr}(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}) + \operatorname{tr}\left( \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \boldsymbol{\phi}(\tilde{\boldsymbol{O}})^{\mathrm{T}} \hat{\boldsymbol{L}}_O \right) \\ & + \operatorname{tr}\left( \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \boldsymbol{\phi}(\tilde{\boldsymbol{V}})^{\mathrm{T}} \hat{\boldsymbol{L}}_V \right) \end{aligned} \tag{25}$$

其中,核矩阵 $\hat{\boldsymbol{K}}$ 和MMD矩阵 $\hat{\boldsymbol{M}}$ 都是经第1步迁移后的样本计算得到的,$\hat{\boldsymbol{K}}_O = \boldsymbol{\phi}(\tilde{\boldsymbol{O}}) \boldsymbol{\phi}(\tilde{\boldsymbol{O}})^{\mathrm{T}}$, $\hat{\boldsymbol{K}}_V = \boldsymbol{\phi}(\tilde{\boldsymbol{V}}) \boldsymbol{\phi}(\tilde{\boldsymbol{V}})^{\mathrm{T}}$. $\hat{\boldsymbol{L}}_O = \hat{\boldsymbol{H}}_O - \hat{\boldsymbol{S}}$ 和 $\hat{\boldsymbol{L}}_V = \hat{\boldsymbol{H}}_V - \tilde{\boldsymbol{S}}$ 分别为源域和目标域的拉普拉斯矩阵,$\hat{\boldsymbol{H}}_O$ 和 $\hat{\boldsymbol{H}}_V$ 分别为源域和目标域的度矩阵,$\hat{\boldsymbol{S}}$ 为源域的亲和矩阵,$\tilde{\boldsymbol{S}}$ 为目标域的亲和矩阵. 可将式(25)改写为式(26):

$$J_{\mathrm{JSDAM}} = \operatorname{tr}(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}) + \operatorname{tr}(\hat{\boldsymbol{K}} \cdot \hat{\boldsymbol{L}}) \tag{26}$$

式中,$(\cdot)$ 表示 $\hat{\boldsymbol{K}}$ 和 $\hat{\boldsymbol{L}}$ 的点乘,$\hat{\boldsymbol{L}} = \begin{bmatrix} \hat{\boldsymbol{L}}_O & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{L}}_V \end{bmatrix}$.

采用统一的内核学习方法,利用显式的低秩表示得到式(27),并对目标函数进行最小化:

$$\min_{\boldsymbol{W}} \ \mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}\boldsymbol{W})+\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\hat{\boldsymbol{L}}^{*}\boldsymbol{W})+\lambda\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}) \quad \text{s.t. } \boldsymbol{W}^{\mathrm{T}}\hat{\boldsymbol{K}}\hat{\boldsymbol{H}}\hat{\boldsymbol{K}}\boldsymbol{W}=\boldsymbol{I} \tag{27}$$

为了使问题的解决方案唯一,在式中引入了约束. $\boldsymbol{W}$是映射矩阵,$\hat{\boldsymbol{L}}^{*}$是$\hat{\boldsymbol{K}}$和$\hat{\boldsymbol{L}}$的点乘,$\lambda\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W})$是正则项. 基于拉格朗日乘子法,式(27)可重新表述为

$$\min_{\boldsymbol{W}} \mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}+\hat{\boldsymbol{L}}^{*}+\lambda\boldsymbol{I})\boldsymbol{W}\right)-\mathrm{tr}((\boldsymbol{W}^{\mathrm{T}}\hat{\boldsymbol{K}}\hat{\boldsymbol{H}}\hat{\boldsymbol{K}}\boldsymbol{W}-\boldsymbol{I})\boldsymbol{Z}) \tag{28}$$

其中,$\boldsymbol{Z}$是包含拉格朗日乘子项的对角矩阵,对$\boldsymbol{W}$趋于0求导得:

$$(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}+\hat{\boldsymbol{L}}^{*}+\lambda\boldsymbol{I})\boldsymbol{W}=\hat{\boldsymbol{K}}\hat{\boldsymbol{H}}\hat{\boldsymbol{K}}\boldsymbol{W}\boldsymbol{Z} \tag{29}$$

式(29)的解就是$(\hat{\boldsymbol{K}}\hat{\boldsymbol{M}}\hat{\boldsymbol{K}}+\hat{\boldsymbol{L}}^{*}+\lambda\boldsymbol{I})^{-1}\hat{\boldsymbol{K}}\hat{\boldsymbol{H}}\hat{\boldsymbol{K}}$的前$\tilde{d}$维特征向量,$\tilde{d}\leqslant n_{\boldsymbol{O}}+n_{\boldsymbol{V}}$. 最后,通过降维后的映射矩阵对源域和目标域进行矩阵乘法运算得到$\boldsymbol{S}'$和$\boldsymbol{E}'_{\mathrm{A}}$、$\boldsymbol{E}'_{\mathrm{B}}$、$\boldsymbol{E}'_{\mathrm{C}}$.

### 4.5 二维决策融合机制

基于上述的MNEMT方法,结合二维决策融合机制,得到新的集成学习框架算法MNEMT_EF. 二维决策融合机制具体的步骤如下:第1步,对原始训练样本$\boldsymbol{S}_{\mathrm{train}}$和MNEMT得到的$\boldsymbol{E}'_{\mathrm{A}}$、$\boldsymbol{E}'_{\mathrm{B}}$和$\boldsymbol{E}'_{\mathrm{C}}$使用PCA算法降维,得到降维后的数据集$\boldsymbol{F}_1$、$\boldsymbol{F}_2$、$\boldsymbol{F}_3$和$\boldsymbol{F}_4$. 第2步,对测试样本$\boldsymbol{S}_{\mathrm{test}}$进行相应的样本变换操作,得到$\boldsymbol{E}'_{\mathrm{A\,test}}=\Upsilon\left(\boldsymbol{S}_{\mathrm{test}},\boldsymbol{E}'_{\mathrm{A}}\right)$,$\boldsymbol{E}'_{\mathrm{B\,test}}=\Upsilon\left(\boldsymbol{S}_{\mathrm{test}},\boldsymbol{E}'_{\mathrm{B}}\right)$,$\boldsymbol{E}'_{\mathrm{C\,test}}=\Upsilon\left(\boldsymbol{S}_{\mathrm{test}},\boldsymbol{E}'_{\mathrm{C}}\right)$,相应的降维后测试样本为$\boldsymbol{F}_{1\mathrm{test}}=\Upsilon\left(\boldsymbol{S}_{\mathrm{test}},\boldsymbol{F}_1\right)$,$\boldsymbol{F}_{2\mathrm{test}}=\Upsilon\left(\boldsymbol{E}'_{\mathrm{A\,test}},\boldsymbol{F}_2\right)$,$\boldsymbol{F}_{3\mathrm{test}}=\Upsilon\left(\boldsymbol{E}'_{\mathrm{B\,test}},\boldsymbol{F}_3\right)$和$\boldsymbol{F}_{4\mathrm{test}}=\Upsilon\left(\boldsymbol{E}'_{\mathrm{C\,test}},\boldsymbol{F}_4\right)$,$\Upsilon(a,b)$表示基于后项$b$元素范式计算前项$a$元素. 第3步,基于$\boldsymbol{F}_1$、$\boldsymbol{F}_2$、$\boldsymbol{F}_3$和$\boldsymbol{F}_4$训练二维基分类器,相应测试样本集$\boldsymbol{F}_{1\mathrm{test}}$、$\boldsymbol{F}_{2\mathrm{test}}$、$\boldsymbol{F}_{3\mathrm{test}}$、$\boldsymbol{F}_{4\mathrm{test}}$测试二维基分类器,得到二维基分类器的预测标签矩阵$\boldsymbol{Y}=\begin{bmatrix}y_{11}&y_{12}&\cdots&y_{1L}\\y_{21}&y_{22}&\cdots&y_{2L}\\y_{31}&y_{32}&\cdots&y_{3L}\\y_{41}&y_{42}&\cdots&y_{4L}\end{bmatrix}$. 第4步,采用投票法对二维基分类器的预测标签融合,得到最终预测标签. 对于样本$\bar{\boldsymbol{S}}_i$最终预测结果表达式如下:

$$\boldsymbol{y}_{\mathrm{predict}}^{(\bar{\boldsymbol{S}}_i)}=\boldsymbol{y}^{(\bar{\boldsymbol{S}}_i)}_{\underset{c^*}{\mathrm{argmax}}\,\mathrm{Count}(\boldsymbol{y}_{c^*}^{(\bar{\boldsymbol{S}}_i)})},(c^*=1,2,\cdots,C^*) \tag{30}$$

其中,Count(·)为计数函数. 设样本$\bar{\boldsymbol{S}}_i$的预测值类型为1个$1\times C^*$维向量$(\boldsymbol{y}_1^{(\bar{\boldsymbol{S}}_i)},\boldsymbol{y}_2^{(\bar{\boldsymbol{S}}_i)},\cdots,\boldsymbol{y}_{C^*}^{(\bar{\boldsymbol{S}}_i)})$,则预测值$\boldsymbol{y}_{c^*}^{(\bar{\boldsymbol{S}}_i)}$出现的次数可表示为

$$\mathrm{Count}(\boldsymbol{y}_{c^*}^{(\bar{\boldsymbol{S}}_i)})=\sum_{i=1}^{4}\sum_{j=1}^{L}f_{\mathrm{com}}(\boldsymbol{y}_{ij},\boldsymbol{y}_{c^*}^{(\bar{\boldsymbol{S}}_i)}) \tag{31}$$

其中,$f_{\mathrm{com}}(a,b)$为对比函数,当$a=b$时$f_{\mathrm{com}}(a,b)=1$,否则$f_{\mathrm{com}}(a,b)=0$. 本文提出的算法伪代码如算法1所示.

### 4.6 复杂度分析

本节旨在评估所提出算法的计算复杂性. 其计算复杂度由MNEMT和二维决策融合机制2个因素组成.

**算法1 流形近邻样本包络与分层多类型变换堆栈集成框架算法**

**输入**:原始样本集$\boldsymbol{S}$

**输出**:最终预测结果$\boldsymbol{y}_{\mathrm{predict}}$

流程:

1. 基于MNE构造流形近邻包络样本$\dot{\boldsymbol{S}}$
2. 基于MT对$\dot{\boldsymbol{S}}$作多类型样本变换,获得$\boldsymbol{E}_{\mathrm{A}}$,$\boldsymbol{E}_{\mathrm{B}}$,$\boldsymbol{E}_{\mathrm{C}}$
3. 设计基于联合结构域适应的层间一致性保持机制,分别对$\boldsymbol{S}$和$\boldsymbol{E}_{\mathrm{A}}$,$\boldsymbol{E}_{\mathrm{B}}$,$\boldsymbol{E}_{\mathrm{C}}$的分布做域适应,以保持层间样本分布的一致性,获得$\boldsymbol{E}'_{\mathrm{A}}$,$\boldsymbol{E}'_{\mathrm{B}}$和$\boldsymbol{E}'_{\mathrm{C}}$
4. 使用PCA降维,得$\boldsymbol{F}_1,\boldsymbol{F}_2,\boldsymbol{F}_3,\boldsymbol{F}_4$
5. 基于数据集$\boldsymbol{F}_1,\boldsymbol{F}_2,\boldsymbol{F}_3,\boldsymbol{F}_4$分别构造样本子空间并训练$4\times L$个基分类器,获得相应预测标签.
6. 根据式(30)和式(31)融合预测标签,得到最终预测结果$\boldsymbol{y}_{\mathrm{predict}}$

MNEMT的计算复杂度为$O(2NGG)+2O(g'g)+O(N(n_{\boldsymbol{O}}+n_{\boldsymbol{V}})^2)$,其中,$G$为原始样本数量,$N$为特征数,MT变换后样本数量为$g'$,变换前样本数量为$g$,$n_{\boldsymbol{O}}$和$n_{\boldsymbol{V}}$分别为源域和目标域样本个数. 二维基分类器矩阵融合的计算复杂度为$O\left((d_k f^3+n\cdot\log(n)\cdot N)KL\right)$,其中,$L$和$K$分别为包络样本空间层数和基分类器的个数,$f$和$n$分别为每个子集的特征个数和样本个数,$d_k$为子特征集的个数. 因此,提出算法的计算复杂度为$O\left(\max\{(2NGG+2g'g+N(m_s+m)^2,(d_k f^3+n\cdot\log(n)\cdot N)KL\}\right)$.

## 5 试验结果与分析

为了验证本文提出算法的有效性,本节设置了3组实验. 第1组实验通过消融法来验证MNEMT的有效性. 第2组实验为算法对比实验,包括准确率、样本集多样性等指标的比较. 第3组实验为提出算法的参数分析.

### 5.1 实验环境

为了验证本文算法有效性,本文选择了19个代表性的UCI和LIBSVM公共数据集. 数据集的详细信息可在链接(https://archive.ics.uci.edu/ml/index.php)和(https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)中找到. 这些数据集包含了医学、生物学、物理学等多个领域数据,样本数由150~7 000不等,特征数由4~5 000不等,类别数由2~7类不等,且被广泛用于集成学习算法的研究与验证. 表2显示了数据集的基本信息.

算法使用了10折交叉验证法. 算法的参数设置如下,实验中使用了随机森林、旋转森林,树的数量设置为50(实验设置范围为15 255 075 100 200 500),子空间数量设置40(实验设置范围为20~40),包络样本数设为7(实验设置范围为2~9). 通过网格搜索方法确定上述超参数. 实验在64位Windows 10计算机上运行,硬件参数为Intel i5-9400 CPU,2.8 GHz,16 GB内存,使用软件为MATLAB软件的R2018b版本.

本文采用准确率(Accuracy)、错误率(Error)、精确率(Precision)、召回率(Recall)、$F_1$分数($F_1$score)和

表2　数据集基本信息

| 数据集 | 样本数 | 特征数 | 类别数 |
|---|---|---|---|
| Balance | 625 | 4 | 3 |
| Banknote | 1 372 | 4 | 2 |
| Biodeg | 1 055 | 41 | 2 |
| Breast | 286 | 9 | 2 |
| German | 1 000 | 24 | 2 |
| Heart1 | 270 | 13 | 2 |
| Iris | 150 | 4 | 3 |
| Liver | 345 | 6 | 2 |
| Ionosphere | 351 | 34 | 2 |
| gisette | 7 000 | 5 000 | 2 |
| Seeds | 210 | 7 | 3 |
| Segment | 2 310 | 18 | 7 |
| Sonar | 208 | 60 | 2 |
| Thyroid | 215 | 5 | 3 |
| Transfusion | 748 | 4 | 2 |
| Vote | 435 | 16 | 2 |
| Wdbc | 569 | 30 | 2 |
| Wine | 178 | 13 | 3 |
| Pima | 768 | 8 | 2 |

Kappa作为衡量算法的指标. Kappa评估2个分类器输出之间的一致性水平,同时纠正偶然性. 将集合中基分类器的平均Kappa值作为集合的Kappa值. Kappa值越高,多样性越低.

### 5.2　消融研究-有效性验证

为了验证MNEMT的有效性,本文利用消融法在不同层样本集上进行了实验. 记录了10次实验的均值分类准确率,每层样本集都使用C4.5作为基分类器,不同样本集的分类结果如表3所示. 表3中样本集包括:原始样本集s0,第1层包络样本集s1,第2层包络样本集s2,第3层包络样本集s3,第1层与第2层包络样本集融合s1&s2以及基于所有样本集的投票法融合sall. 在表3中列出来部分实验结果,更多结果和分析见链接:https://pan.baidu.com/s/1ySkWBWpPawM1hlTCg8P6OA? pwd=7jwv .

由表3可知,对于包络样本集s1和s2,在大部分数据集上其分类准确率、精确度、召回率和$F_1$分数均高于原样本集s0. 对于包络样本集s3,其在不同数据集上表现存在差异,在Pima数据集上准确率低于原样本集,而在Ionosphere、Breast和Sonar数据集上s3准确率相较于原样本集s0准确率分别提高了2.84%、5.38%和9.21%. 对于第1层与第2层包络样本集融合s1&s2而言,多个数据集性能指标都低于多层样本集融合的结果. 对于多层样本集融合sall,在多个数据集上获得了最高准确率. 由此体现出不同分层样本集对分类准确率的影响不同存在互补性,综合不同样本层进行集成学习是必要和有效的. 第3层包络样本集s3虽然在多个数据集上性能指标低于原样本集s0和s1&s2,但是所有层融合sall的结果高于s1&s2,这是因为第3层包络样本集增加了集成多样性,对于最后的多层样本融合结果而言是有益的.

### 5.3　方法比较

为了验证本文算法的有效性,本文将经典的集成学习算法与提出的MNEMT_EF进行比较. 对比算法包括 Rotation Forest、Bagging、Random forests、Adaboost、Xgboost、LightGBM算法.

#### 5.3.1　与经典集成学习算法比较

在19个数据集下,进行了6种经典集成学习算法和本文算法的对比实验. 其中,6个数据集的分类准确率、精确度、召回率和$F_1$分数如表4所示. 其他数据集实验结果和分析见链接:https://pan.baidu.com/s/1ySkWBWpPawM1hlTCg8P6OA? pwd=7jwv. 实验所用基分类器为决策树. 本文方法MNEMT_EF在19个数据集的10个中取得最高准确率. 其中在Liver数据集上比Adaboost算法准确率提高了13.94%,在Sonar数据集上,相较于准确率第2高的Adaboost算法准确率提升了7.64%.

在Wine数据集上,本文算法相较于最高准确率的算法仅依次低了0.53%和1.87%. 总的来说,在准确率方面本文算法在大多数据集上取得最优结果且提升巨大,在部分数据集上与获得最优结果的对比算法差距较小,结果表明本文算法是有效的. 算法在不同数据集上表现的差异性实验$t$检验结果如表5所示,显著性水平为0.05,将P-value小于0.05的数值标注加粗.

如表5所示,本文算法与所有算法不存在显著差异性,Bagging算法与Random Forests、Xgboost和LightGBM算法均存在显著差异性,Rotation forest与Random forest也存在显著差异性.

#### 5.3.2　与最新集成学习算法比较

在本节中,将本文算法与目前最新的集成学习算法进行了比较,基于不同的数据集,比较结果如表6所示.

由表6可知,本文算法在Breast数据集取得了最优分类准确率为82.05%,在Liver数据集上取得了最高准确率为80.35%,在Sonar数据集上达到最高准确率为91.31%. 在数据集Wine上与取得最优准确率的算法相差2.41%,在Pima数据集上与最优准确率相差了6.94%. 总的来说,本文算法的创新是有效的,相较于其他的代表性算法有效提高了分类准确率.

#### 5.3.3　多样性比较

本节比较了本文算法中各层样本集的多样性,也比较了不同算法的分类器多样性. 为了客观验证,本文在balance、breast、ionosphere、liver、pima和wdbc数据集

表3 不同层样本集的分类结果比较 单位:%

| 数据集 | 性能指标 | 原样本集s0 | 第1层包络样本集s1 | 第2层包络样本集s2 | 第3层包络样本集s3 | 第1层与第2层包络样本集融合s1&s2 | 多层样本集融合sall |
|---|---|---|---|---|---|---|---|
| sonar | acc | 74.93 | 82.21 | 86.57 | 84.14 | 87.10 | **91.31** |
| | pre | 76.03 | 84.75 | 88.10 | 85.81 | 88.93 | **91.54** |
| | rec | 74.96 | 82.32 | 86.79 | 83.97 | 87.88 | **91.82** |
| | $F_1$ | 73.19 | 80.27 | 85.67 | 82.64 | 86.27 | **90.62** |
| Wine | acc | 93.27 | 87.78 | 92.12 | 85.42 | 95.56 | **97.22** |
| | pre | 93.72 | 89.28 | 90.41 | 86.24 | 94.01 | **94.81** |
| | rec | 93.84 | 87.30 | 91.05 | 85.40 | 93.63 | **95.74** |
| | $F_1$ | 93.35 | 86.98 | 89.99 | 83.75 | 93.63 | **95.18** |
| Pima | acc | **71.88** | 72.01 | 68.62 | 71.87 | 68.62 | 70.06 |
| | pre | 69.07 | 76.62 | 69.11 | 77.15 | 73.03 | **77.36** |
| | rec | **65.84** | 61.20 | 55.17 | 60.86 | 55.52 | 57.57 |
| | $F_1$ | **66.16** | 59.80 | 49.29 | 59.06 | 50.40 | 53.52 |
| Breast | acc | 72.28 | 72.25 | 78.40 | 77.66 | 77.67 | **82.05** |
| | pre | 65.81 | 70.50 | 75.63 | 74.62 | 72.07 | **80.31** |
| | rec | 67.00 | 69.10 | 65.34 | 68.55 | 69.02 | **71.50** |
| | $F_1$ | 65.83 | 65.97 | 64.86 | 67.80 | 68.04 | **71.36** |
| Liver | acc | 61.79 | 75.70 | 70.50 | 74.27 | 79.79 | **80.35** |
| | pre | 60.56 | 75.97 | 72.19 | 73.63 | 80.66 | **81.46** |
| | rec | 60.78 | 75.68 | 68.13 | 73.53 | 78.48 | **78.55** |
| | $F_1$ | 60.15 | 74.84 | 67.17 | 72.72 | 78.43 | **78.79** |
| Ionosphere | acc | 86.04 | 89.75 | 80.06 | 88.88 | 87.47 | **92.74** |
| | pre | 87.46 | 93.13 | 83.05 | 91.35 | 87.21 | **94.21** |
| | rec | 82.33 | 85.61 | 74.41 | 85.34 | 83.45 | **88.48** |
| | $F_1$ | 83.64 | 87.60 | 73.60 | 86.64 | 83.69 | **90.31** |

注:加粗数据表示最优结果.

上基于相同的基分类器数量,分别计算原样本集s0,包络样本集s1、s2、s3,不同层样本集融合sall的Kappa-Error指标. 选取了其中2个数据集(liver和breast)的结果图展示如图9. 在图中纵轴表示基分类器对的分类误差,横轴表示基分类器对的Kappa值.

为了对比本文算法(Proposed)与典型算法的基分类器多样性,本文选择了典型算法:Rotation forest(Rof)、Bagging(Bag),Randon Forests(Rf)、AdaBoost(Ada),Xgboost(Xgb)、LightGBM(Lgb),并在对比实验中采用相同的基分类器数量,在balance、breast、ionosphere、liver、pima和wdbc数据集下计算了Kappa-error图. 选取了其中2个数据集(liver和ionosphere)的结果图展示如图10.

由图10所示,代表本文方法(Proposed)的紫色点分布区域较大,且大部分紫色表征点分布在所有表征点的下方偏左位置. 这说明本文算法具有较低的error值和较低Kappa值,相较于其他算法具有较高的多样性.

### 5.4 参数分析

本节选取7个数据集,比较不同算法在不同基分类器数下的分类性能表现,如表7所示. 如表7所示,对于Balance数据集,算法的最优基分类器数量是75,对于German数据集,其最优基分类器数量是75,对于Vote数据集,其最优基分类器数量是75,对于Wine数据集最优基分类器数量是25、50、75、500. 由以上分析可知,本文算法中使用的最优基分类器数需要通过验证集调参的方式进行选取.

本文算法和典型集成学习算法在不同分类器数量下的分类准确率如图11所示. 在Heart1数据集图11($a$)上本文算法能在不同分类器数下达到的最高准确率,且准确率受分类器个数影响较小,在图11($b$)中,本文算法在分类器数量为75时取得了最高的分类准确率.

**表4　与典型集成学习算法的对比**　　单位:%

| 数据集 | 性能指标 | Rotation Forest | Bagging | Random Forests | Adaboost | Xgboost | LightGBM | MNEMT_EF |
|---|---|---|---|---|---|---|---|---|
| sonar | acc | 80.74 | 74.45 | 83.17 | 83.67 | 82.67 | 80.79 | **91.31** |
| | pre | 80.44 | 74.19 | 84.95 | 83.64 | 81.97 | 80.47 | **91.54** |
| | rec | 80.41 | 74.09 | 81.84 | 83.05 | 81.44 | 80.21 | **91.82** |
| | $F_1$ | 79.28 | 73.09 | 81.46 | 82.57 | 81.11 | 79.38 | **90.62** |
| Wine | acc | 92.12 | 96.11 | 97.75 | 96.08 | **97.75** | 96.63 | 97.22 |
| | pre | 92.93 | 95.60 | 98.06 | 96.71 | **98.13** | 96.91 | 94.81 |
| | rec | 92.55 | 96.30 | 97.86 | 96.90 | **97.86** | 96.71 | 95.74 |
| | $F_1$ | 92.33 | 95.63 | **97.89** | 96.50 | 97.88 | 96.71 | 95.18 |
| Pima | acc | 73.45 | **77.35** | 76.31 | 74.62 | 75.79 | 75.53 | 70.06 |
| | pre | 70.72 | **75.05** | 74.25 | 72.53 | 74.39 | 73.24 | 77.36 |
| | rec | 69.56 | **74.87** | 72.91 | 72.03 | 70.51 | 72.39 | 57.57 |
| | $F_1$ | 69.63 | **74.63** | 72.87 | 71.62 | 71.08 | 72.26 | 53.52 |
| Breast | acc | 67.95 | 70.44 | 73.70 | 70.09 | 70.79 | 74.49 | **82.05** |
| | pre | 59.85 | 65.39 | 65.58 | 64.20 | 64.62 | 72.09 | **80.31** |
| | rec | 59.16 | 65.07 | 63.51 | 62.53 | 64.64 | 60.12 | **71.50** |
| | $F_1$ | 57.82 | 64.38 | 61.81 | 61.31 | 63.08 | 59.92 | **71.36** |
| Liver | acc | 67.30 | 70.45 | 74.50 | 66.41 | 72.22 | 69.87 | **80.35** |
| | pre | 65.94 | 69.44 | 73.94 | 65.22 | 71.94 | 68.85 | **81.46** |
| | rec | 65.28 | 69.86 | 72.89 | 65.29 | 71.87 | 66.62 | **78.55** |
| | $F_1$ | 65.07 | 69.35 | 72.76 | 64.79 | 71.26 | 66.39 | **78.79** |
| Ionosphere | acc | 92.59 | 93.75 | **94.03** | 91.45 | 93.45 | 93.17 | 92.74 |
| | pre | 92.10 | **94.24** | 93.86 | 90.94 | 93.84 | 93.93 | 94.21 |
| | rec | 92.02 | 92.35 | **92.88** | 90.42 | 91.99 | 91.65 | 88.48 |
| | $F_1$ | 91.45 | 93.03 | **93.05** | 90.31 | 92.55 | 92.35 | 90.31 |

注:加粗数据表示最优结果.

**表5　对比算法 $t$ 检验**

| 数据集 | Rotation forest | Bagging | Random forests | AdaBoost | Xgboost | LightGBM | MNEMT_EF (proposed) |
|---|---|---|---|---|---|---|---|
| Rotation forest | — | 0.759 8 | **0.012 1** | 0.484 7 | **0.004 5** | 0.070 9 | 0.179 9 |
| Bagging | 0.759 8 | — | **0.004 7** | 0.693 7 | **0.011 6** | **0.020 6** | 0.140 6 |
| Randon forests | **0.012 1** | **0.004 7** | — | **0.023 4** | 0.727 3 | 0.061 1 | 0.652 9 |
| AdaBoost | **0.484 7** | 0.693 7 | **0.023 4** | — | **0.037 7** | 0.089 0 | 0.077 3 |
| Xgboost | **0.004 5** | **0.011 6** | 0.727 3 | 0.037 7 | — | 0.154 2 | 0.614 2 |
| LightGBM | 0.070 9 | **0.020 6** | 0.061 1 | 0.089 0 | 0.154 2 | — | 0.387 6 |
| MNEMT_EF (proposed) | 0.179 9 | 0.140 6 | 0.652 9 | 0.077 3 | 0.614 2 | 0.387 6 | — |

注:加粗数据表示P-value小于0.05的数值.

表 6 与最新集成算法比较 单位:%

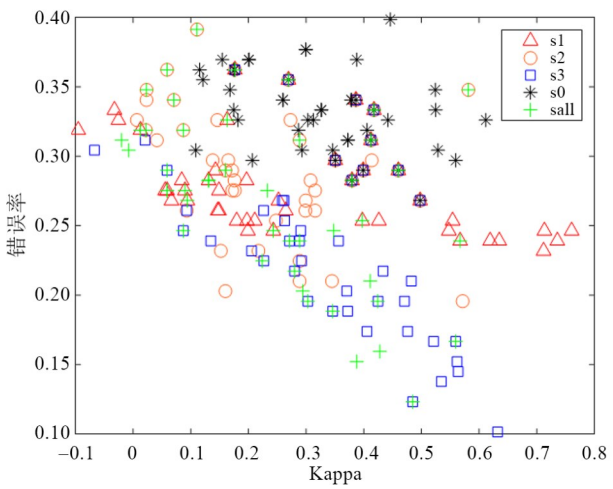
| 数据集 | Algorithm | Acc |
|---|---|---|
| Breast | ELBAD[25] | 74.31 |
| | HBoost[26] | 76.47 |
| | ECBIW[27] | 72.20 |
| | MNEMT_EF(proposed) | **82.05** |
| Ionosphere | ELBAD[25] | **93.56** |
| | E_RSRR[7] | 93.40 |
| | CSME[28] | 91.78 |
| | HSCE-G&Lc[29] | 86.90 |
| | ICSRD[30] | 89.40 |
| | ECL-MAOD[31] | 91.78 |
| | FLT[32] | 92.77 |
| | GRD[33] | 91.40 |
| | MNEMT_EF(proposed) | 92.74 |
| Liver | ECBIW[27] | 63.10 |
| | HBoost[26] | 74.67 |
| | CSME[28] | 72.42 |
| | ECL-MAOD[31] | 67.53 |
| | ECCS[34] | 71.80 |
| | MNEMT_EF(proposed) | **80.35** |
| Pima | E_RSRR[7] | 76.30 |
| | ECBIW[27] | 72.61 |
| | GRD[33] | **77.00** |
| | MNEMT_EF(proposed) | 70.06 |
| Sonar | ECBIW[27] | 73.28 |
| | CSME[28] | 91.02 |
| | FLT[32] | 82.26 |
| | ICSRD[30] | 80.10 |
| | ECL-MAOD[31] | 88.19 |
| | ECCS[34] | 83.20 |
| | HF2HM[35] | 85.20 |
| | MNEMT_EF(proposed) | **91.31** |
| Wine | ELBAD[25] | 97.74 |
| | ECBIW[27] | 94.52 |
| | HBoost[26] | 98.81 |
| | CSME[28] | 98.30 |
| | HSCE-G&Lc[29] | 75.90 |
| | ICSRD[30] | 97.90 |
| | ECL-MAOD[31] | 98.23 |
| | ILCS-MD[33] | **99.63** |
| | ECCS[34] | 99.20 |
| | HF2HM[35] | 92.60 |
| | MNEMT_EF(proposed) | 97.22 |

注:加粗数据表示最优结果.

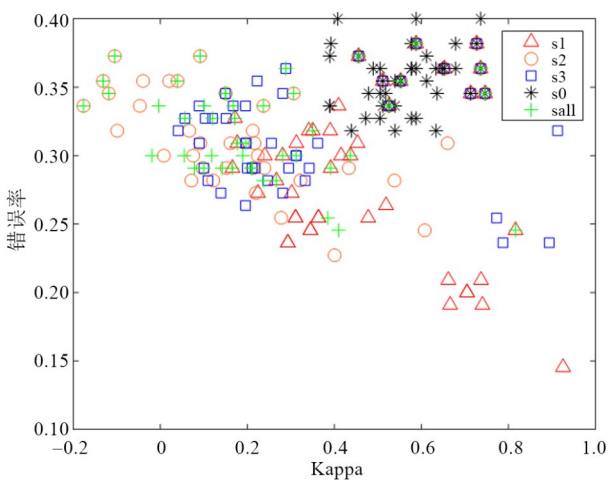
表 7 本文算法在不同分类器数量情况的分类准确率 单位:%

| 数据集 | 基分类器个数 | | | | | | |
|---|---|---|---|---|---|---|---|
| | 15 | 25 | 50 | 75 | 100 | 200 | 500 |
| Balance | 82.54 | 84.13 | 82.54 | **90.25** | 79.37 | 84.13 | 82.54 |
| German | 79.10 | 79.60 | 78.90 | **81.10** | 79.84 | 79.80 | 79.70 |
| Heart1 | 91.11 | 90.00 | 91.48 | **92.96** | 92.59 | 92.59 | **92.96** |
| Iris | 90.67 | 90.00 | **93.33** | 90.67 | 92.33 | 91.78 | 91.78 |
| Vote | 94.32 | 94.68 | 92.84 | **95.60** | 93.06 | 93.51 | 94.68 |
| Transfusion | **76.34** | 76.07 | 76.03 | 76.28 | 76.10 | 76.10 | 76.28 |
| Wine | 96.77 | **97.22** | **97.22** | **97.22** | 96.89 | 96.89 | **97.22** |

注:加粗数据表示最优结果.

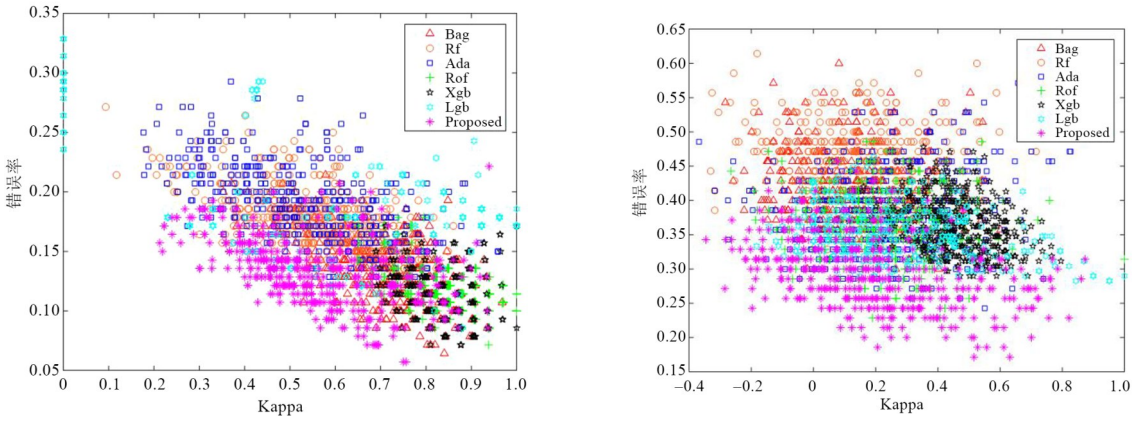



(a) Liver数据集



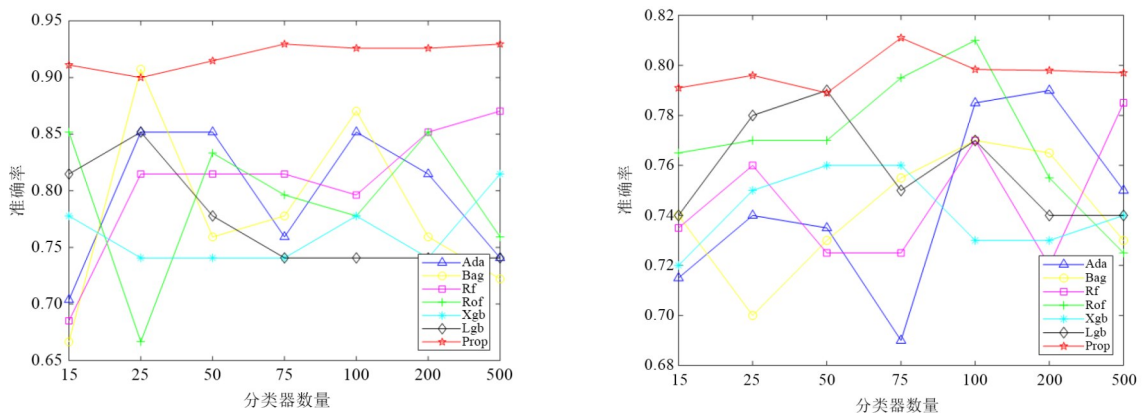

(b) Breast数据集

图9 不同层样本集的多样性比较




(*a*) Ionosphere数据集　　(*b*) Liver数据集

图10　不同算法的基分类器多样性比较




(*a*) Heart1数据集　　(*b*) German数据集

图11　不同算法的基分类器数量与分类准确率关系图(基于UCI数据集)

## 6　结论

集成学习算法是机器学习领域的重要分支. 目前的集成学习算法主要基于原样本集得到多个样本子集,然后进行建模集成. 这种集成学习范式存在1个共性问题:各子集内样本均来自同一个原样本集,导致各子集之间的多样性显著降低. 尤其当原样本集数据尺寸小、采样比率大、不平衡程度高时,样本子集的质量将非常低.

此外,当原样本集可分度低时,通过重采样获得的样本子集的可分度改善也很有限. 针对此问题,本文提出了一种解决方法——面向集成学习的流形近邻样本包络与分层多类型变换算法(MNEMT_EF). 该算法首先采用流形近邻样本包络化机制构建样本包络,并设计了多类型样本变换算子,基于样本包络构造分层包络样本集. 然后使用层间一致性保持机制,保持变换前后样本的分布一致性. 最后,采用二维决策融合机制得到最终分类结果. 本文方法最大的特点在于通过多类型样本变换构建了具有差异性的分层包络样本集,提高样本子集的多样性和可分度,克服了现有集成学习方法基于同一原样本集的局限性. 本文通过消融法实验和对比实验,从模型准确率和分类器多样性方面,验证了本文算法创新是有效的.

与现有基于原始样本的集成学习算法不同,本文提出了一种基于变换后多层包络样本的集成学习框架. 该集成学习框架中多类型样本变换算法、子空间采样方法、决策融合机制都可用相关具体算法进行替换,因此,本文提出的框架算法具有较好的通用性. 在今后的研究工作将考虑结合深度学习和本文提出的框架算法,来进一步验证本文算法在深度学习中的有效性.

### 参考文献

[1] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.
ZHOU Z H. Machine Learning[M]. Beijing: Tsinghua University Press, 2016. (in Chinese)

[2] MOHAMMED A, KORA R. A comprehensive review on ensemble deep learning: Opportunities and challenges[J]. Journal of King Saud University - Computer and Information Sciences, 2023, 35(2): 757-774.

[3] 刘金平, 何捷舟, 马天雨, 等. 基于KELM选择性集成的复杂网络环境入侵检测[J]. 电子学报, 2019, 47(5): 1070-1078.
LIU J P, HE J Z, MA T Y, et al. Selective ensemble of KELM-based complex network intrusion detection[J]. Acta Electronica Sinica, 2019, 47(5): 1070-1078. (in Chinese)

[4] FREUND Y, SCHPIRE R E. Experiments with a new boosting algorithm[C]//13th International Conference on Machine Learning. Morgan: Kaufmann, 1996, 96: 148-156.

[5] BREIMAN L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.

[6] SHI J Q, LI C X, YAN X H. Artificial intelligence for load forecasting: A stacking learning approach based on ensemble diversity regularization[J]. Energy, 2023, 262: 125295.

[7] 朱鹏飞, 胡清华, 于达仁. 基于随机化属性选择和邻域覆盖约简的集成学习[J]. 电子学报, 2012, 40(2): 273-279.
ZHU P F, HU Q H, YU D R. Ensemble learning based on randomized attribute selection and neighborhood covering reduction[J]. Acta Electronica Sinica, 2012, 40(2): 273-279. (in Chinese)

[8] CUI S Z, WANG Y Z, YIN Y Q, et al. A cluster-based intelligence ensemble learning method for classification problems[J]. Information Sciences, 2021, 560: 386-409.

[9] LEE S J, XU Z Z, LI T, et al. A novel bagging C4.5 algorithm based on wrapper feature selection for supporting wise clinical decision making[J]. Journal of Biomedical Informatics, 2018, 78: 144-155.

[10] ZHAO C M, WU D R, HUANG J, et al. BoostTree and BoostForest for ensemble learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(7): 8110-8126.

[11] SHU J, YUAN X, MENG D Y, et al. CMW-net: Learning a class-aware sample weighting mapping for robust deep learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(10): 11521-11539.

[12] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[13] NIE F P, LI Z H, WANG R, et al. An effective and efficient algorithm for K-means clustering with new formulation[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(4): 3433-3443.

[14] SAKAR B E, ERDEM ISENKUL M, SAKAR C O, et al. Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings[J]. IEEE Journal of Biomedical and Health Informatics, 2013, 17(4): 828-834.

[15] LI Y M, LIU C Y, WANG P, et al. Envelope multi-type transformation ensemble algorithm of Parkinson speech samples[J]. Applied Intelligence, 2023, 53(12): 15957-15978.

[16] MOHAMED A, QIAN K, ELHOSEINY M, et al. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 14412-14420.

[17] XIA S, LIU Y, DING X, et al. Granular ball computing classifiers for efficient, scalable and robust learning[J]. Information Sciences, 2019, 483: 136-152.

[18] HINTON G E, SALAKHUTDINOV R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[19] NGO G, BEARD R, CHANDRA R. Evolutionary bagging for ensemble learning[J]. Neurocomputing, 2022, 510: 1-14.

[20] BREIMAN L. Randomizing outputs to increase prediction accuracy[J]. Machine Learning, 2000, 40(3): 229-242.

[21] DIETTERICH T G, BAKIRI G. Solving multiclass learning problems via error-correcting output codes[J]. Journal of Artificial Intelligence Research, 1995, 2: 263-286.

[22] DENG X L, DAI Z G, SUN M D, et al. Variational autoencoder based enhanced behavior characteristics classification for social robot detection[C]//International Conference on Security and Privacy in Digital Economy. Singapore: Springer, 2020: 232-248.

[23] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[24] MENG F Y, LIU H, LIANG Y S, et al. Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition[J]. IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society, 2019, 28(11): 5281-5295.

[25] SHIUE Y R, YOU G R, SU C T, et al. Balancing accuracy and diversity in ensemble learning using a two-phase artificial bee colony approach[J]. Applied Soft Computing, 2021, 105: 107212.

[26] KADKHODAEI H R, MOGHADAM A M E, DEHGHAN M. HBoost: A heterogeneous ensemble classifier based on the Boosting method and entropy measurement [J]. Expert Systems with Applications, 2020, 157: 113482.
[27] MAO S S, LIN W S, JIAO L C, et al. End-to-end ensemble learning by exploiting the correlation between individuals and weights[J]. IEEE Transactions on Cybernetics, 2021, 51(5): 2835-2846.
[28] JAN Z M, VERMA B. Multiple elimination of base classifiers in ensemble learning using accuracy and diversity comparisons[J]. ACM Transactions on Intelligent Systems and Technology, 2020, 11(6): 1-17.
[29] YANG Y, JIANG J M. Hybrid sampling-based clustering ensemble with global and local constitutions[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(5): 952-965.
[30] JAN Z, MUNOS J C, ALI A. A novel method for creating an optimized ensemble classifier by introducing cluster size reduction and diversity[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(7): 3072-3081.
[31] ASAFUDDOULA M, VERMA B, ZHANG M J. A divide-and-conquer-based ensemble classifier learning by means of many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(5): 762-777.
[32] ARMANO G, TAMPONI E. Building forests of local trees[J]. Pattern Recognition, 2018, 76: 380-390.
[33] KIZILOZ H E. Classifier ensemble methods in feature selection[J]. Neurocomputing, 2021, 419: 97-107.
[34] MD JAN Z, VERMA B. Evolutionary classifier and cluster selection approach for ensemble classification[J]. ACM Transactions on Knowledge Discovery from Data, 2020, 14(1): 1-18.
[35] WANG L J, MO T L, WANG X T, et al. A hierarchical fusion framework to integrate homogeneous and heterogeneous classifiers for medical decision-making[J]. Knowledge-Based Systems, 2021, 212: 106517.

## 作者简介

**颜　芳**　女,1979年10月出生,甘肃省天水人. 重庆大学微电子与通信工程学院副教授. 主要研究领域为智能信息处理.
E-mail: yanfang@cqu.edu.cn

**马　洁**　女,1998年11月出生,云南腾冲人. 重庆大学微电子与通信工程学院博士研究生. 主要研究领域为数据信息处理、机器学习.
E-mail: 609290268@qq.com

**李勇明**　男,1976年9月出生,四川绵阳人. 重庆大学微电子与通信工程学院教授,博士生导师. 主要研究领域为医学信号处理、机器学习.
E-mail: yongmingli@cqu.edu.cn

**王　品**　女,1979年11月出生,江苏盐城人. 重庆大学微电子与通信工程学院副教授,硕士生导师. 主要研究领域为图像处理与识别.
E-mail: wangpin@cqu.edu.cn

**覃　剑**　男,1977年5月出生,陕西宝鸡人. 重庆大学微电子与通信工程学院副教授. 主要研究领域为视频分析及传输.
E-mail: qinjian@cqu.edu.cn

**刘承宇**　男,1998年3月出生,广西桂林人. 重庆大学微电子与通信工程学院硕士研究生. 主要研究领域为机器学习、帕金森语音识别.
E-mail: 719702750@qq.com