软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(4):1934−1944 [doi: 10.13328/j.cnki.jos.006841]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

多标记学习中基于交互表示的深度森林方法*

吕沈欢[1,2], 陈一赫[1,2], 姜 远[1,2]

[1](计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)
[2](软件新技术与产业化协同创新中心 (南京大学), 江苏 南京 210023)
通信作者: 姜远, E-mail: jiangy@lamda.nju.edu.cn

摘 要: 在多标记学习中, 每个样本都与多个标记关联, 关键任务是如何在构建模型时利用标记之间的相关性. 多标记深度森林算法尝试在深度集成学习的框架下使用逐层的表示学习来挖掘标记之间的相关性, 并利用得到的标记概率表示提升预测精度. 然而, 一方面标记概率表示与标记信息高度相关, 这会导致其多样性较低. 随着深度森林的深度增加, 性能会下降. 另一方面, 标记概率的计算需要我们存储所有层数的森林结构并在测试阶段逐一使用, 这会造成难以承受的计算和存储开销. 针对这些问题, 提出基于交互表示的多标记深度森林算法 (interaction-representation-based multi-label deep forest, iMLDF). iMLDF 从森林模型的决策路径中挖掘特征空间中的结构信息, 利用随机交互树抽取决策树路径中的特征交互, 分别得到特征置信度得分和标记概率分布两种交互表示. iMLDF 一方面充分利用模型中的特征结构信息来丰富标记间的相关信息, 另一方面通过交互表达式计算所有的表示, 从而使得算法无需存储森林结构, 大大地提升了计算效率. 实验结果表明: 在交互表示基础上进行表示学习的 iMLDF 算法取得了更好的预测性能, 而且针对样本较多的数据集, 计算效率比 MLDF 算法提升了一个数量级.

关键词: 深度森林; 多标记学习; 特征交互; 标记相关性; 表示学习
中图法分类号: TP18

中文引用格式: 吕沈欢, 陈一赫, 姜远. 多标记学习中基于交互表示的深度森林方法. 软件学报, 2024, 35(4): 1934–1944. http://www.jos.org.cn/1000-9825/6841.htm
英文引用格式: Lü SH, Chen YH, Jiang Y. Interaction-representation-based Deep Forest Method in Multi-label Learning. Ruan Jian Xue Bao/Journal of Software, 2024, 35(4): 1934–1944 (in Chinese). http://www.jos.org.cn/1000-9825/6841.htm

Interaction-representation-based Deep Forest Method in Multi-label Learning

LÜ Shen-Huan[1,2], CHEN Yi-He[1,2], JIANG Yuan[1,2]

[1](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)
[2](Collaborative Innovation Center of Novel Software Technology and Industrialization (Nanjing University), Nanjing, 210023)

Abstract: In multi-label learning, each sample is associated with multiple labels. The key task is how to use the correlation between labels when building the model. Multi-label deep forest (MLDF) algorithm attempts to mine the correlation between labels by using layer-by-layer representation learning under the framework of deep ensemble learning and use the obtained label probability representation to improve prediction accuracy. However, on the one hand, the label probability representation is highly correlated with the label information, which will lead to its low diversity. As the depth of the deep forest increases, the performance will decline. On the other hand, the calculation of label probability requires the storage of forest structures with all layers and the application of these structures one by one in the test stage, which will cause unbearable computational and storage overhead. To solve these problems, this study proposes interaction-representation-based MLDF (iMLDF). iMLDF mines the structural information in the feature space from the decision path of the forest model, extracts the feature interaction in the decision tree path by using the random interaction trees, and obtains two interaction

* 基金项目: 国家自然科学基金 (62176117)
收稿时间: 2022-03-15; 修改时间: 2022-10-19; 采用时间: 2022-12-02; jos 在线出版时间: 2023-07-28
CNKI 网络首发时间: 2023-07-31

representations of feature confidence score and label probability distribution, respectively. On the one hand, iMLDF makes full use of the feature structural information in the forest model to enrich the relevant information between labels. On the other hand, it calculates all the representations through interaction expressions so that the algorithm does not need to store all the forest structures, which greatly improves computational efficiency. The experimental results show that iMLDF algorithm achieves better prediction performance, and the computational efficiency is improved by an order of magnitude compared with MLDF for datasets with massive samples.

Key words: deep forest; multi-label learning; feature interaction; label correlation; representation learning

1 引 言

在多标记学习任务中, 一个训练样本往往对应着多个标记, 而学习任务则是学习模型使得对未见过的测试样本预测其对应的所有的标记[1]. 多标记学习任务在现实场景中应用非常广泛, 例如文本分类任务[2]、视频分类任务[3]、化学分类任务[4]等. 形式化定义来说, 我们令 $\mathcal{X}=\mathbb{R}^d$ 代表 d 维的特征空间, $\mathcal{Y}=\{y_1,y_2,\ldots,y_q\}$ 代表包含 q 个类别标记的标记空间. 由此我们可以给定一个多标记学习的训练集 $\mathcal{D}$, 其中 $\mathbf{x}_i\in\mathcal{X}$ 是一个 d 维的特征向量 $\mathbf{x}_i=(x_{i1},x_{i2},\ldots,x_{id})$ 且 $Y_i\in\mathcal{Y}$ 是 $\mathbf{x}_i$ 对应的与其相关的标记的集合. 多标记学习任务是为了学习一个预测模型 $h:\mathcal{X}\to 2^{\mathcal{Y}}$, 使得我们可以用其更好地预测未见样本对应的标记集合.

在多标记学习任务中, 探索和利用标记之间的相关性始终是一个受到关注的核心方法. 文献 [5] 通过将多标记学习问题转化为每个标记的独立二元分类问题. 虽然它旨在充分利用高性能的传统单标记分类器, 但当标记空间巨大时, 会导致较高的计算成本. 考虑到一个标记上的信息可能有助于学习其他相关标记的事实, 则研究标记之间的相关性对于提高多标记学习的性能至关重要[6]. 因此, 大多数现有方法都是通过以探索标记间相关性的方式利用标记信息来训练多标记示例[1]. 还有部分工作尝试利用特征空间中的数据结构来丰富多标记之间的相关信息, 以此来提升算法的预测性能[7]. 文献 [8] 尝试使用树结构模型来分阶段处理多标记任务中的标记相关性信息. 而文献 [9] 则通过深度神经网络来构造标记之间关系的嵌入空间, 从而学习标记之间的相关性.

不同于传统的多标记学习算法, 深度学习引入了表示学习的框架. 文献 [10] 在学习新的特征空间后, 在网络最后一层的输出上使用多标记分类器. 但是深度神经网络模型对于数据量的需求很大, 且其作为由可微部件构成的复杂系统更加适合处理数值建模问题, 比如图像分类问题、语音分类问题等. 通过认识到深度学习的本质在于逐层处理、模型特征转换和足够的模型复杂性, 文献 [11] 针对中小型混合建模数据集提出了深度森林模型并通过 gcForest 算法实现训练. 具有级联结构的深度森林模型可以像深度神经网络模型一样进行表示学习[12]. 与深度神经网络相比, 深度森林具有更少的超参数, 因此更易于训练. 文献 [13] 将深度森林拓展到了多标记学习的任务上, 探索了基于标记相关性表示学习的多标记深度森林方法. 作为多标记深度森林的一个拓展, 文献 [14] 通过使用标记补足方法解决了弱标记学习问题. 在每一层, 使用内部交叉验证方案对训练数据集的标记集进行补充, 即如果预测标记为正, 则在训练数据集中更改其标记.

尽管这些基于标记信息表示的深度森林在经验和理论上都显示出了巨大的潜力, 但我们认为基于标记预测的特征表示是一个关键缺陷. 首先, 正如文献 [11] 所述, 预测的标记概率提供的信息非常有限. 由于决策树集成后的随机森林已经是相当稳定的分类器, 这会导致特征表示的冗余且缺乏多样性, 这同样也是导致普通的 Stacking 算法无法直接构成深度模型的原因之一. 文献 [11] 提到这种基于 Stacking 的实现方式会在超过两层之后遭受严重的过拟合风险. 其次, 基于标记预测的表示在计算时依赖于多层森林模型的存储, 需要大量的存储空间和时间消耗. 因此, 如何针对多标记学习任务为深度森林模型设计信息量大、计算量小的特征表示是一个关键的问题.

在本文中, 我们提出了基于交互表示的多标记深度森林方法 (interaction-representation-based multi-label deep forest, iMLDF). 它从森林模型的决策路径中挖掘特征空间中的结构信息, 利用随机交互树抽取决策树路径中的特征交互, 分别得到了特征置信度得分和标记概率分布两种交互表示. 然后它会利用这些基于特征交互的表示与原始特征通过粘贴操作构造级联森林, 实现逐层的表示学习, 不断挖掘特征空间中更复杂的结构.

本文的主要贡献有以下两个方面.

(1) 首次设计了针对多标记森林模型的随机交互提取树方法, 充分利用了模型中的特征结构信息来丰富标记

间的相关信息. 这种基于交互的表示学习增加了多样性, 降低了过拟合风险.

(2) 利用交互表达式计算所有的表示, 从而使得算法无需存储森林结构, 极大地提升了深度森林的计算效率.

本文首先简要讨论一些相关工作. 其次, 介绍本文方法的技术细节. 第三, 报告对比研究的实验结果. 最后, 对本文进行总结.

2 相关工作

2.1 多标记深度森林

多标记深度森林方法 (multi-label deep forest, MLDF) 是一种基于森林模块构建的针对多标记学习任务的深度模型[13]. 该方法由级联森林结构实现逐层特征转化, 级联结构的每一层森林模块由两组多标记决策树森林组成, 包括预测聚类树随机森林 (random forest of predictive clustering trees, RF-PCT) 和预测聚类树极限随机森林 (extremely random forest of predictive clustering trees, ERF-PCT). 其中每个随机森林输出其得到的对应样本的标记概率分布, 生成基于标记概率向量的表示特征. 这些表示特征作为增广特征和原始特征拼接在一起, 一起成为下一层森林模块的输入. 为了模型的复杂度能自适应具体任务, 每一层级联森林训练结束都会通过交叉验证估计整个级联结构的性能, 如果达到停止条件, 则会终止训练过程. 除此之外, 在级联森林的逐层处理过程中, 预测概率分布使用基于衡量指标的置信度进行评估. 更具体地说, 如果来自当前层的预测概率分布比来自前一层的预测概率分布具有更好的置信度, 才更新它们. 文献 [14] 在多标记深度森林的基础上使用无监督数据来帮助挖掘特征空间中的结构信息, 从而使得深度森林模型获得更好的预测性能. 文献 [15] 则在深度森林的级联结构中加入类别选择的新机制, 加强了模型过滤标记信息中噪声的能力, 从而提升了预测性能. 另一种过滤噪声的方式是在级联结构中逐层筛选置信度低的样本进入下一层, 文献 [16,17] 通过置信度筛选的机制来缓解过拟合风险. 将在实际应用中, 文献 [18,19] 将多标记深度森林方法应用在了蛋白质标注问题上, 并取得了良好的性能. 文献 [20] 则是将多标记深度森林应用于流式数据任务, 并取得了不错的预测性能. 文献 [21] 将多标记深度森林归类为一种成功利用表示学习的多标记深度学习方法.

2.2 特征间交互信息

我们将特征交互定义为决策规则中的条件集合, 特征交互的简单形式为:

IF: 条件 为真 &…& 条件 为真; THEN: 交互表示激活.

这种特征交互信息最早被用于理解整个转录组、蛋白质的全基因组结合位点和许多其他分子过程如何通过高阶交互的方式驱动基因表达[19]. 文献 [22] 则进一步证明了通过随机森林中的集成决策树是可以恢复特征间的交互信息的, 且越是高阶的交互恢复的难度越高. 文献 [23] 利用输入特征的稳定高阶交互来生成多样性更高的表示特征, 并在此基础上构建深度森林的级联结构获得了更好的预测性能, 同时也降低了计算和存储开销. 由于原学习任务是多分类问题, 该方法中的交互表示计算只依赖于样本对应的唯一标记. 而本文在文献 [23] 计算的特征间交互信息 (交互激活特征区域) 的基础上从特征空间和标记空间两个不同的角度设计了新的表示特征, 其中局部激活区域的标记概率向量编码了局部空间中的标记间相关性. 这种对于标记间关系的建模方式是解决多标记学习任务的核心方法.

3 基于交互表示的多标记深度森林

本节我们提出了基于交互表示的多标记深度森林方法 (iMLDF), 它通过特征空间的结构挖掘交互信息融入了级联森林的特征表示中.

3.1 提取特征交互表示

当我们使用训练数据得到一个已完成训练的随机森林模型, 集成大量富有多样性的决策树使得其具有优秀的预测性能, 然而我们往往只能将其作为黑箱模型用以完成预测任务. 由于构成随机森林的基分类器决策树受到多种随机性的扰动, 因此随机森林中的单棵决策树的决策路径并不可靠. 但是相比于由标记信息主导的单棵决策树

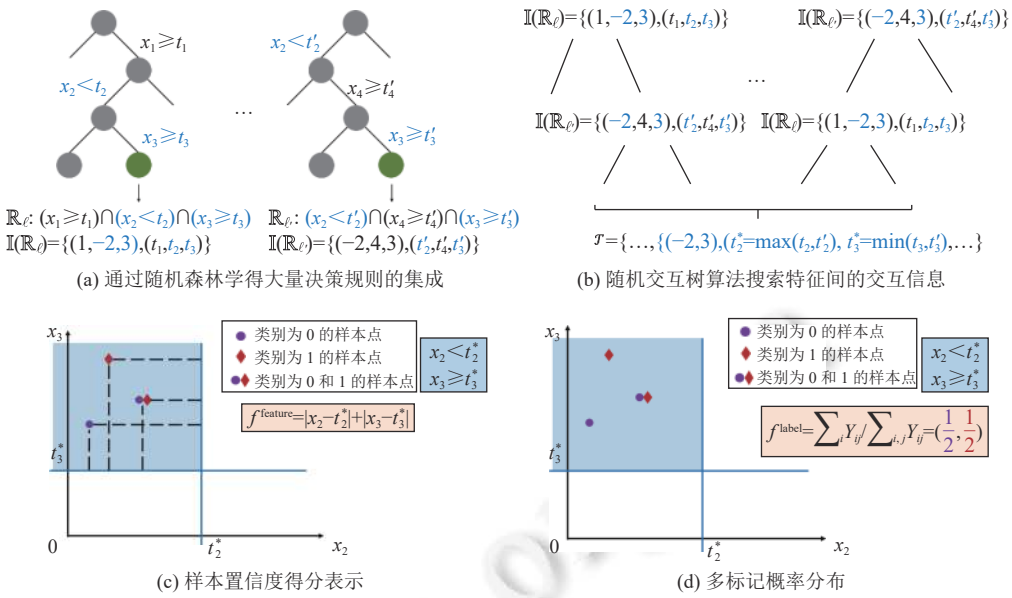
的决策路径, 通过随机森林中大量决策树的集成则可以筛选出在随机噪声的影响下仍然较为稳定出现的一部分决策路径, 这些决策路径在随机性的影响下是较为鲁棒的. 我们可以提取出所有决策路径中稳定性最高的一部分特征交互, 而这些特征交互编码了特征空间中的结构信息. 我们可以通过特征交互激活的特征空间区域得到两组表示特征: 特征置信度得分和标记概率分布.

考虑一个多标记学习任务, 我们有 n 个训练样本, 它们分别为 p 维的特征向量. 由此我们可以给定一个多标记学习的训练集 $S_n=\{(\mathbf{x}_1,Y_1),\ldots,(\mathbf{x}_n,Y_n)\}$. 其中对于 $\forall i\in\{1,\ldots,n\}$, $\mathbf{x}_i\in S$ 是一个 p 维的特征向量 $\mathbf{x}_i=(x_{i1},x_{i2},\ldots,x_{ip})$ 且 $Y_i\in\mathcal{Y}$ 是 $\mathbf{x}_i$ 对应的与其相关的标记的集合. 使用这些训练数据训练得到多个不同的森林模型之后, 我们可以从其中的决策树路径中得到大量的决策规则. 这些规则对于不同的特征类型会有所不同. 对于一个连续性特征 $x_{\cdot k}$, 它可以采取以下形式: 比如 $x_{\cdot k}<t_k$ 或者 $x_{\cdot k}\geqslant t_k$, 也可以记为 $x_{\cdot k}\gtrless t_k$. 而对于一个类别型特征, 我们则将其进行 one-hot 编码使得其决策规则和上述方式一致. 由此定义一条决策树路径为 $\mathbb{R}_j=\{(\Delta_k,t_k)|\forall(x_{\cdot k}t_k)\in 第j个决策路径\}$, 此处的 $\Delta_k\in\{-k,+k\}$ 表示 $x_{\cdot k}$ 是否在该决策路径中且其对应的决策规则的方向 $\gtrless$(+ 代表 $\geqslant$, –代表 <), t_k 表示其对应的决策规则的阈值. 我们可以拆解决策路径信息为两部分交互信息: 方向和下标信息 $\mathbb{I}_j^{\text{index}}$ 和阈值信息 $\mathbb{I}_j^{\text{threshold}}$. 图 1(a) 中的例子就是随机森林模型训练后得到的两个叶子节点 ℓ 和 ℓ' 所对应的两个决策规则: $\mathbb{R}_\ell=(x_{\cdot 1}\geqslant t_1)\cap(x_{\cdot 2}<t_2)\cap(x_{\cdot 3}\geqslant t_3)$ 和 $\mathbb{R}_{\ell'}=(x_{\cdot 2}<t_2')\cap(x_{\cdot 4}\geqslant t_4')\cap(x_{\cdot 3}\geqslant t_3')$. 在其被拆解为两部分交互信息之后, 可以表示为两个式子: $\mathbb{I}(\mathbb{R}_\ell)=\{(1,-2,3),(t_1,t_2,t_3)\}$ 和 $\mathbb{I}(\mathbb{R}_{\ell'})=\{(-2,4,3),(t_2',t_4',t_3')\}$.

算法 1. 多标记随机交互树 (multi-label random interaction trees, ML-RIT).

输入: 决策路径集合 $\mathcal{R}=\{\mathbb{R}_i|\mathbb{R}_i=\{\mathbb{I}_i^{\text{index}},\mathbb{I}_i^{\text{threshold}}\}\}_{i=1}^n$;
输出: $\mathcal{T}$ 激活区域的样本置信度得分和多标记概率分布.

1. **for** 树 ℓ 在 $\{1,2,\ldots,L\}$ 中 **do**
2. 令 ℓ 是深度为 D 的树. 令 J 树中节点的总数, 并对每一对父子节点对进行标注, 使得子节点的下标比父亲节点大. 对每一个节点 $j=1,\ldots,J$, 记节点 j 的父亲节点为 $pa(j)$, 令 i_j 是关于节点 j 的一个从训练数据中均匀采样得到的下标
3. 令 $T_1^{\text{index}}\leftarrow\mathbb{I}_{i_1}^{\text{index}}$, $T_1^{\text{threshold}}\leftarrow\mathbb{I}_{i_1}^{\text{threshold}}$
4. **for** j 在 $\{2,\ldots,J\}$ 中 **do**
5. $T_j^{\text{index}}\leftarrow\mathbb{I}_{i_j}^{\text{index}}\cap T_{pa(j)}$
6. $T_j^{\text{threshold}}\leftarrow()$
7. **for** 特征下标 i 在 T_j^{index} 中 **do**
8. **if** $\Delta_i\geqslant 0$ **then**
9. $T_j^{\text{threshold}}\leftarrow T_j^{\text{threshold}}+\left\{\max(t_i,t_i')|t_i\in\mathbb{I}_{i_j}^{\text{threshold}},t_i'\in T_{pa(j)}^{\text{threshold}}\right\}$
10. **else**
11. $T_j^{\text{threshold}}\leftarrow T_j^{\text{threshold}}+\left\{\min(t_i,t_i')|t_i\in\mathbb{I}_{i_j}^{\text{threshold}},t_i'\in T_{pa(j)}^{\text{threshold}}\right\}$
12. **end if**
13. **end for**
14. **end for**
15. $\mathcal{T}_\ell^{\text{index}}\leftarrow\left\{T_j^{\text{index}}:depth(j)=D\right\}$
16. $\mathcal{T}_\ell^{\text{threshold}}\leftarrow\left\{T_j^{\text{threshold}}:depth(j)=D\right\}$
17. $\mathcal{T}_\ell\leftarrow\{\mathcal{T}_\ell^{\text{index}},\mathcal{T}_\ell^{\text{threshold}}\}$
18. 得到交互 $\mathcal{T}=\bigcup_{\ell=1}^L\mathcal{T}_\ell$
19. **end for**
20. 通过 bootstrap 采样对交互进行评估和筛选
21. 计算 $\mathcal{T}$ 激活区域的样本置信度得分和多标记概率分布



(a) 通过随机森林学得大量决策规则的集成　(b) 随机交互树算法搜索特征间的交互信息

(c) 样本置信度得分表示　(d) 多标记概率分布

图 1　多标记随机交互树算法中提取特征交互的示意图

由于这些决策树路径只与其对应的叶子结点的少部分样本强相关, 我们选择通过设计多标记随机交互树算法来利用随机性筛选其中对多标记评估更加有益的决策规则, 可以获得具有更好泛化能力的特征交互. 我们在算法 1 中总结了多标记随机交互树算法 (multi-label random interaction trees, ML-RIT). ML-RIT 包含 L 棵交互树. 在每一棵交互树 ℓ 中, J 个样例的下标 $\{i_1,\ldots,i_J\}$ 从数据中均匀采样得到, 它们对应的决策路径由 $\{\mathbb{I}_{i_1},\ldots,\mathbb{I}_{i_J}\}$ 代表. 然后, 它使用决策路径的下标信息取交集操作 $\mathbb{I}_{i_1} \cap \ldots \cap \mathbb{I}_{i_J}$ 来保留交互信息并去除噪声特征. 最后保留下的每一组特征交互. 图 1(b) 展示了 ML-RIT 算法对于决策规则中交互信息的筛选操作, 通过随机生长交互树的方式来聚合重复出现的特征交互, 也就是说不断地对交互的特征下标信息和阈值信息取交集来得到更加稳定的交互信息. 图 1(a) 中的交互信息对应的聚合结果就是 $\{(-2,3),(\max(t_2,t'_2),\min(t_3,t'_3))\}$.

算法 1 得到的所有交互 $T_k \in \mathcal{T}$ 都会对应一个符合其决策规则的子空间区域. ML-RIT 方法则通过这些子空间区域计算得到两种表示输出, 分别为特征置信度得分:

$$f_k^{\text{feature}}(\mathbf{x}_i)=\begin{cases}\sum_j |x_{ij}-t_j|, & \forall i:\mathbf{x}_i \sim T_k \\ 0, & \text{其他}\end{cases},$$

和标记概率分布:

$$f_k^{\text{label}}(\mathbf{x}_i)=\begin{cases}\sum_{i:\mathbf{x}_i \sim T_k} Y_i / \sum_{i:\mathbf{x}_i \sim T_k,j} Y_{ij}, & \forall i:\mathbf{x}_i \sim T_k \\ 0, & \text{其他}\end{cases},$$

其中, $\mathbf{x}_i \sim T$ 表示 $\mathbf{x}_i$ 是在特征交互 T 对应的子空间区域中的. 图 1(c) 展示了特征置信度得分的含义, 即各个样本点位于交互对应激活区域的位置距离区域边界的 ℓ_1 距离. 图 1(d) 展示了标记概率分布的含义, 即在各个交互对应的激活区域内对所有样本统计其局部标记的概率分布.

3.2　级联森林结构

如图 2 所示, 在基于交互表示的多标记深度森林方法 (iMLDF) 中, 级联结构通过每层挖掘的特征间交互表示来实现. 每一层的森林模块包含两组多标记决策树森林组成, 包括预测聚类树随机森林 (RF-PCT) 和预测聚类树极限随机森林 (ERF-PCT). 每一层的森林模块会通过多标记随机交互树算法输出两组不同的特征交互表示: 区域置信度得分和区域标记概率分布, 而这些新的特征表示会通过粘贴操作和原始特征一起作为下一层的输入, 如此迭代循环直到触发停止条件.

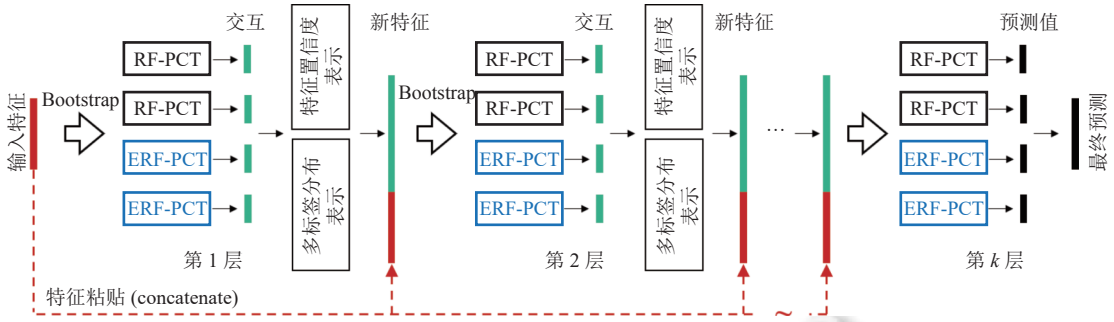


图 2 基于交互表示的多标记深度森林方法级联结构示意图

形式化地描述, 我们令 $\mathbf{h}=\{h_1,h_2,\ldots,h_K\}$, $\mathbf{f}=\{f_1,f_2,\ldots,f_K\}$, 其中 h_k 代表第 k 层的森林模块对应的函数, f_k 代表 k 层的深度森林挖掘得到的特征交互表示, 具体还可以分为 f_k^{feature} 和 f_k^{label} 两部分. 对于多标记随机交互树算法对应的函数我们用 $\mathcal{T}$ 表示, 该函数作用于森林函数 h_k 上, 记作 $\mathcal{T}\circ h_k$. 而 K 则代表停止条件触发时的深度. 由此我们可以定义基于交互表示的多标记深度森林方法在第 k 层的迭代式为:

$$f_k(\mathbf{x})=\begin{cases}\mathcal{T}\circ h_k(\mathbf{x}), & k=1\\ \mathcal{T}\circ h_k([\mathbf{x},f_{k-1}(\mathbf{x})]), & k>1\end{cases},$$

其中, $[a, b]$ 代表特征向量 a 和特征向量 b 的拼接.

由该迭代式计算得到的二元组 $(\mathbf{h},\mathbf{f})$ 即代表了一个基于交互表示的多标记随机森林 $g:\mathcal{X}\to\mathcal{Y}$:

$$g(\mathbf{x})=h_K([\mathbf{x},f_{K-1}(\mathbf{x})]).$$

深度森林最后一层森林模块输出的各个标记的预测概率值和阈值决定了对于每一个样本 $\mathbf{x}$ 输出的相关的标记集合 $\hat{Y}$.

4 实验部分

本文提出的基于交互表示的多标记深度森林方法是一种可以通过挖掘特征空间结构来丰富训练样本标记信息的多标记学习算法. 本节我们将 iMLDF 与一些经典的多标记学习算法进行对比, 有 RF-PCT[24,25]、DBPNN[26,27]、RAKEL[28]、MLKNN[29]、MLARAM[30]和基于标记表示的深度森林方法 MLDF[13]. 我们的目标是验证 iMLDF 可以在不同的度量上实现最佳性能, 尤其是和 MLDF 的对比结果验证了交互表示引入的特征空间结构有利于丰富标记间信息. 此外, 我们还验证了 iMLDF 相比于 MLDF 大幅缩减了测试阶段的计算开销和存储开销.

4.1 实验设置与数据集

我们选择了 7 个来自不同应用领域和不同规模的多标记分类基准数据集. 表 1 列出了这些数据集的基本统计数据. 所有数据集均来自多标记数据集的存储库. 数据集的大小各不相同: 从 207 个到 6000 个示例, 从 49 个到 1079 个特征, 从 5 个到 174 个标记. 它们按示例数升序排列. 对于下面进行的所有实验, 在不替换的情况下随机抽取 70% 的样本以形成训练集, 其余 30% 的样本用于创建测试集.

表 1 根据示例数量、特征数量和标记数量统计数据集.

数据集	领域	示例数	特征数	标记数
VirusPse-AAC	Biology	207	440	6
CAL500	Music	502	68	174
CHD_49	Medicine	555	49	6
Emotions	Music	593	72	6
Image	Image	2000	294	5
Slashdot	Text	3782	1079	22
Reuters-K500	Text	6000	500	103

表 2 列出了本文采用的 6 种在多标记学习[5]中广泛使用的评估方法: hamming loss, one-error, coverage, ranking loss, average precision 和 macro-AUC, "↓"意味着度量越低越好, "↑"意味着度量越高越好. 其中 coverage 是通过标记的数量进行归一化的, 因此所有评估度量值都在 [0, 1] 之间变化.

表 2 6 个多标记性能度量的定义

度量	公式
Hamming loss ↓	$\frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}\mathbb{I}\left[h_{ij}\neq y_{ij}\right]$
One-error ↓	$\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left[\arg\max f(\mathbf{x}_i)\notin Y_i^+\right]$
Coverage ↓	$\frac{1}{ml}\sum_{i=1}^{m}\mathbb{I}\left[\max_{j\in Y_i^+} rank_f(\mathbf{x}_i,j)-1\right]$
Ranking loss ↓	$\frac{1}{m}\sum_{i=1}^{m}\frac{\left\lvert S_{\text{rank}}^{i}\right\rvert}{\left\lvert Y_i^+\right\rvert\left\lvert Y_i^-\right\rvert}$
Average precision ↑	$\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\left\lvert Y_i^+\right\rvert}\sum_{j\in Y_i^+}\frac{\left\lvert S_{\text{precision}}^{ij}\right\rvert}{rank_f(\mathbf{x}_i,j)}$
Macro-AUC ↑	$\frac{1}{l}\sum_{j=1}^{l}\frac{\left\lvert S_{\text{macro}}^{j}\right\rvert}{\left\lvert Y_{\cdot j}^+\right\rvert\left\lvert Y_{\cdot j}^-\right\rvert}$

4.2 不同衡量指标下的性能

我们将 iMLDF 与以下 6 个对比算法进行比较: RF-PCT[24,25]、DBPNN[26,27]、RAKEL[28]、MLKNN[29]、MLARAM[30] 和 MLDF[13]. 其中, DBPNN (deep back propagation neural network) 是 DNN 方法用于多标记问题的代表算法; RAKEL (random K-label pruned sets) 和 RF-PCT (random forest of predictive clustering trees) 是多标记集成学习方法的代表; RAKEL 从标记集合中提取 M 个大小为 k 的子集, 并对每个子集进行修剪集训练, 然后结合修剪集分类器中的标记投票, 得到一个标记向量预测. RF-PCT 是基于 PCT 的集成, PCT 从多分类决策树改进, 它允许在树的叶子中有多个标记, 其公式可以通过对每个标记的标准值求和来修改; RF-PCT 通过自助法采样以及特征子集筛选来加强 PCT 的预测能力. MLKNN (multi-label K-nearest neighbours) 构建使用 K-nearest neighbors 找到与测试样本最接近的训练样本, 并使用贝叶斯推断选择最终分配的标记. MLARAM (multi-label fuzzy adaptive resonance associative map) 的目标是通过增加一个额外的 ART (adaptive resonance theory) 层来将学习到的原型聚类成大型的聚类, 从而提高分类速度. 在这种情况下, 所有原型的激活可以被其中一小部分的激活所取代, 从而显著减少分类时间.

下面列出了比较方法的参数设置. 对于 RF-PCT, 将树的个数取为 500, 分裂指标为基尼指数, 不限制树的深度, 特征子集的大小为 $\sqrt{k}$. 对于 DBPNN, 选择两个大小分别为 100 和 50 的隐层, 激活函数为线性整流函数, 优化器为 Adam. 对于 RAKEL, 选择使用基尼指数为分裂标准的决策树作为基学习器, 每个标记分区的大小为 4. 对于 MLKNN, 设置近邻大小为 10, 光滑参数为 1.0. 对于 MLARAM, 设置 ART 网络的参数为 0.9. 对于 MLDF, 设置最大层数为 10, 每层中有 4 个 RF-PCT, 每个 RF-PCT 的树的个数为 100, 每棵树都使用基尼指数作为分裂指标, 并且不限制树的深度. 对于 iMLDF, 关于每一层中 RF-PCT 的设置与 MLDF 保持一致, 最大层数也为 10, 对于其中随机交集树参数的设置, 选择自助采样的次数为 10; 默认的交集树的数量是 20, 深度为 5.

我们对每个算法进行了 10 次实验. 记录 10 次训练/测试实验的平均度量值和标准偏差, 以进行比较研究. 表 3 报告了比较算法的详细实验结果, ↓(↑) 表示值越小 (越大), 性能越好, 粗体表示最好的预测性能. iMLDF 在每个评估指标方面实现最佳平均排名. 在 7 个基准数据集中, 在所有评估指标中, iMLDF 在 88.10% 的测试中排名第 1,

在 9.52% 的测试中排名第 2. 综上所述, iMLDF 在各种评估指标的广泛基准数据集上取得了与其他竞争者相比的最佳性能, 这验证了 iMLDF 的有效性.

表 3 7 个数据集上比较方法的预测性能

度量	算法	VirusPse-AAC	CAL500	CHD_49	Emotions	Image	Slashdot	Reuters-K500
Hamming loss ↓	RF-PCT	0.168±0.005	0.136±0.000	0.288±0.002	0.184±0.003	0.165±0.001	0.039±0.000	0.011±0.000
	DBPNN	0.199±0.008	**0.134±0.000**	0.333±0.013	0.217±0.004	0.175±0.002	0.048±0.001	0.011±0.000
	RAKEL	0.247±0.016	0.201±0.002	0.357±0.020	0.280±0.010	0.251±0.007	0.051±0.001	0.016±0.000
	MLKNN	0.186±0.000	0.146±0.000	0.430±0.000	0.307±0.000	0.247±0.000	0.054±0.000	0.014±0.000
	MLARAM	0.239±0.000	0.171±0.000	0.329±0.000	0.407±0.000	0.308±0.025	0.100±0.000	0.095±0.001
	MLDF	0.172±0.002	0.135±0.000	0.288±0.005	**0.172±0.005**	0.150±0.002	0.039±0.001	0.011±0.000
	iMLDF	**0.171±0.004**	0.135±0.000	**0.284±0.003**	0.177±0.005	**0.149±0.002**	**0.038±0.000**	**0.010±0.000**
One-error ↓	RF-PCT	0.432±0.022	0.098±0.002	0.236±0.004	0.260±0.010	0.289±0.006	0.411±0.002	0.384±0.003
	DBPNN	0.589±0.027	**0.090±0.004**	0.309±0.013	0.314±0.011	0.309±0.003	0.457±0.002	0.404±0.006
	RAKEL	0.684±0.060	0.523±0.027	0.407±0.033	0.492±0.040	0.499±0.007	0.556±0.010	0.631±0.005
	MLKNN	0.552±0.000	0.137±0.000	0.273±0.000	0.505±0.000	0.563±0.000	0.707±0.000	0.772±0.000
	MLARAM	0.631±0.000	0.360±0.000	0.415±0.000	0.641±0.000	0.346±0.001	0.511±0.000	0.794±0.001
	MLDF	0.426±0.002	0.096±0.002	**0.234±0.008**	0.262±0.010	0.269±0.004	**0.409±0.004**	**0.374±0.001**
	iMLDF	**0.400±0.013**	0.095±0.007	0.238±0.011	**0.249±0.005**	**0.263±0.006**	0.414±0.001	**0.374±0.001**
Coverage ↓	RF-PCT	0.171±0.007	0.743±0.003	0.450±0.001	0.265±0.002	0.174±0.002	0.105±0.001	0.055±0.001
	DBPNN	0.204±0.005	0.740±0.003	0.488±0.007	0.304±0.006	0.190±0.002	0.146±0.002	0.058±0.001
	RAKEL	0.279±0.009	0.967±0.002	0.574±0.014	0.431±0.016	0.307±0.007	0.278±0.003	0.333±0.002
	MLKNN	0.208±0.000	0.753±0.000	0.497±0.000	0.412±0.000	0.279±0.000	0.191±0.000	0.119±0.000
	MLARAM	0.296±0.000	0.917±0.000	0.568±0.000	0.483±0.000	0.209±0.000	0.248±0.000	0.197±0.000
	MLDF	0.178±0.004	0.739±0.000	0.451±0.006	0.266±0.002	0.166±0.002	0.105±0.002	**0.052±0.000**
	iMLDF	**0.161±0.004**	**0.736±0.003**	**0.442±0.004**	**0.265±0.006**	**0.164±0.002**	**0.103±0.000**	**0.052±0.000**
Ranking loss ↓	RF-PCT	0.181±0.008	0.182±0.001	0.206±0.001	0.134±0.003	0.148±0.003	0.092±0.001	0.042±0.001
	DBPNN	0.224±0.006	0.181±0.001	0.258±0.009	0.183±0.006	0.166±0.003	0.126±0.002	0.038±0.000
	RAKEL	0.640±0.035	0.694±0.007	0.574±0.022	0.532±0.015	0.533±0.006	0.524±0.007	0.528±0.003
	MLKNN	0.224±0.000	0.219±0.000	0.268±0.000	0.302±0.000	0.290±0.000	0.176±0.000	0.088±0.000
	MLARAM	0.643±0.000	0.436±0.000	0.505±0.000	0.585±0.000	0.241±0.001	0.415±0.000	0.264±0.000
	MLDF	0.187±0.004	0.179±0.000	0.208±0.004	0.137±0.002	0.138±0.002	0.090±0.002	**0.035±0.000**
	iMLDF	**0.170±0.002**	**0.178±0.001**	**0.202±0.005**	**0.133±0.004**	**0.136±0.003**	**0.088±0.001**	**0.035±0.000**
Average precision ↑	RF-PCT	0.731±0.013	0.498±0.001	0.796±0.001	0.819±0.005	0.811±0.003	0.692±0.001	0.696±0.001
	DBPNN	0.647±0.010	0.495±0.002	0.764±0.006	0.782±0.006	0.796±0.002	0.650±0.002	0.672±0.002
	RAKEL	0.565±0.014	0.266±0.004	0.718±0.011	0.676±0.011	0.662±0.005	0.540±0.007	0.413±0.004
	MLKNN	0.655±0.000	0.440±0.000	0.747±0.000	0.650±0.000	0.650±0.000	0.462±0.000	0.372±0.000
	MLARAM	0.562±0.000	0.359±0.000	0.743±0.000	0.570±0.000	0.770±0.001	0.574±0.000	0.370±0.000
	MLDF	0.728±0.008	0.500±0.000	0.794±0.006	0.819±0.004	0.823±0.002	**0.693±0.001**	**0.703±0.001**
	iMLDF	**0.751±0.005**	**0.502±0.002**	**0.800±0.005**	**0.824±0.002**	**0.827±0.004**	**0.693±0.001**	**0.703±0.001**
Macro-AUC ↑	RF-PCT	0.769±0.017	0.556±0.007	0.575±0.008	0.871±0.002	0.862±0.001	0.861±0.004	0.896±0.004
	DBPNN	0.741±0.011	0.550±0.005	0.575±0.012	0.811±0.006	0.845±0.004	0.820±0.004	0.910±0.005
	RAKEL	0.573±0.018	0.508±0.003	0.532±0.016	0.672±0.015	0.669±0.009	0.661±0.010	0.632±0.005
	MLKNN	0.654±0.000	0.521±0.000	0.498±0.000	0.674±0.000	0.723±0.000	0.664±0.000	0.715±0.000
	MLARAM	0.491±0.000	0.511±0.000	0.542±0.000	0.586±0.000	0.812±0.001	0.675±0.000	0.661±0.000
	MLDF	0.777±0.004	0.559±0.002	**0.589±0.020**	0.873±0.001	0.870±0.000	0.865±0.005	0.911±0.001
	iMLDF	**0.779±0.006**	**0.560±0.006**	0.573±0.006	**0.875±0.006**	**0.871±0.001**	**0.870±0.001**	**0.929±0.006**

4.3 计算与存储开销

iMLDF 在具有较好的预测表现的同时, 在计算开销上相比于 MLDF 也有显著的优势. 为了展示计算开销的结果, 我们对比 iMLDF 和 MLDF 在多个数据集上的表现. 我们使用的硬件为: 16×3.70 GHz CPU 以及 128 GB 内存.

iMLDF 和 MLDF 都为 20 层, 每层的 RF-PCT 的超参数设置与之前保持一致. 表 4 展示了最终的结果. 可以看到 iMLDF 在测试时间和存储开销上远远好于 MLDF, 基本达到 10 倍以上的提升. 在数据量较大的数据集上, iMLDF 可以将深度森林方法的测试时间和存储开销做到和经典随机森林算法接近. 由于深度森林算法需要对所有样本进行逐层表示学习和表示特征的生成, 因此训练时间会远大于经典随机森林方法. 但是考虑到每一层森林的计算可以通过并行化计算降低时间开销, 一般我们认为这样的计算代价来换取性能的提升是可行的.

表 4　3 个数据集的训练时间、测试时间和内存使用率的比较结果

数据集	算法	训练时间 (s)	测试时间 (s)	存储 (MB)
Emotions	iMLDF	714.06	2.25	122
	MLDF	479.79	47.36	1 971
	RF-PCT	6.77	0.45	52
Slashdot	iMLDF	2 975.45	4.67	1 916
	MLDF	4 457.01	354.49	27 435
	RF-PCT	28.85	2.02	1 015
Reuters-K500	iMLDF	17 795.29	17.69	10 699
	MLDF	37 185.08	1 286.20	112 585
	RF-PCT	74.62	10.02	6 140

iMLDF 计算效率高的主要原因是: 由于 MLDF 生成的特征表示是基于 RF-PCT 预测的. 因此, MLDF 必须保存训练的所有层 RF-PCT 来为测试实例生成特征表示, 这实际上意味着保存了数万条未修剪的决策规则, 因此需要很大的内存成本. 除此之外, MLDF 需要花费大量时间对测试实例进行逐层预测. 而在另一方面, iMLDF 不依赖于基于预测的特征表示. 因此, 每一层只需要保存少量 (数十个) 特征交互, 测试实例可以很容易地基于这些交互生成特征表示.

5 总　结

在本文中, 我们针对多标记深度森林方法表示特征不够丰富且计算和存储开销过大的问题, 提出了基于交互表示的多标记深度森林算法. 该方法第 1 次提出利用交互的子区域的特征置信度和标记概率分布提取两种表示特征, 从而增加表示学习的多样性, 且使得表示特征的获取可以无需存储所有的森林结构. 因此, 本文方法在提升预测性能的同时大大降低了计算和存储开销. 实验表明, 该算法在大范围基准数据集上取得了良好的性能. 在未来的研究中, 我们还可以对随机交互提取算法进行理论分析, 给使用交互表示代替森林结构的存储一个理论保证. 这也会对我们未来设计高阶交互信息的提取算法有指导意义.

References:

[1] Zhang QW, Zhong Y, Zhang ML. Feature-induced labeling information enrichment for multi-label learning. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. New Orleans: AAAI, 2018. 545. [doi: 10.1609/aaai.v32i1.11656]

[2] Zhou P, El-Gohary N. Ontology-based multilabel text classification of construction regulatory documents. Journal of Computing in Civil Engineering, 2016, 30(4): 04015058. [doi: 10.1061/(ASCE)CP.1943-5487.0000530]

[3] Ray J, Wang H, Tran D, Wang YF, Feiszli M, Torresani L, Paluri M. Scenes-objects-actions: A multi-task, multi-label video dataset. In: Proc. of the 15th European Conf. on Computer Vision. Munich: Springer, 2018. 660–676. [doi: 10.1007/978-3-030-01264-9_39]

[4] Cheng X, Zhao SG, Xiao X, Chou KC. iATC-mHyb: A hybrid multi-label classifier for predicting the classification of anatomical therapeutic chemicals. Oncotarget, 2017, 8(35): 58494–58503. [doi: 10.18632/oncotarget.17028]

[5] Tsoumakas G, Katakis I. Multi-label classification: An overview. Int'l Journal of Data Warehousing and Mining, 2007, 3(3): 1–13. [doi: 10.4018/jdwm.2007070101]

[6] Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: Maimon O, Rokach L, eds. Data Mining and Knowledge Discovery Handbook. Boston: Springer, 2009. 667–685. [doi: 10.1007/978-0-387-09823-4_34]

[7] Wang J, Yang Y, Mao JH, Huang ZH, Huang C, Xu W. CNN-RNN: A unified framework for multi-label image classification. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2285–2294. [doi: 10.1109/CVPR.2016.

251]

[8] Vens C, Struyf J, Schietgat L, Džeroski S, Blockeel H. Decision trees for hierarchical multi-label classification. Machine Learning, 2008, 73(2): 185–214. [doi: 10.1007/s10994-008-5077-3]

[9] Liu SY, Song XH, Ma ZC, Ganaa ED, Shen XJ. MoRE: Multi-output residual embedding for multi-label classification. Pattern Recognition, 2022, 126: 108584. [doi: 10.1016/j.patcog.2022.108584]

[10] Zhang ML, Zhou ZH. Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. on Knowledge and Data Engineering, 2006, 18(10): 1338–1351. [doi: 10.1109/TKDE.2006.162]

[11] Zhou ZH, Feng J. Deep forest. National Science Review, 2019, 6(1): 74–86. [doi: 10.1093/nsr/nwy108]

[12] Lyu SH, Yang L, Zhou ZH. A refined margin distribution analysis for forest representation learning. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 5530–5540.

[13] Yang L, Wu XZ, Jiang Y, Zhou ZH. Multi-label learning with deep forest. In: Proc. of the 24th European Conf. on Artificial Intelligence. Santiago de Compostela: ECAI, 2020. 1634–1641.

[14] Wang QW, Yang L, Li YF. Learning from weak-label data: A deep forest expedition. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence, 2020, 34(4): 6251–6258. [doi: 10.1609/aaai.v34i04.6092]

[15] Chen YN, Weng W, Wu SX, Chen BH, Fan YL, Liu JH. An efficient stacking model with label selection for multi-label classification. Applied Intelligence, 2021, 51(1): 308–325. [doi: 10.1007/s10489-020-01807-z]

[16] Ma PF, Wu YX, Li Y, Guo L, Li Z. DBC-Forest: Deep forest with binning confidence screening. Neurocomputing, 2022, 475: 112–122. [doi: 10.1016/j.neucom.2021.12.075]

[17] Ma PF, Wu YX, Li Y, Guo L, Jiang H, Zhu XQ, Wu XD. HW-Forest: Deep forest with hashing screening and window screening. ACM Trans. on Knowledge Discovery from Data, 2022, 16(6): 123. [doi: 10.1145/3532193]

[18] Yu QZ, Dong ZH, Fan XY, Zong LC, Li Y. HMD-AMP: Protein language-powered hierarchical multi-label deep forest for annotating antimicrobial peptides. arXiv:2111.06023, 2021.

[19] Basu S, Kumbier K, Brown J B, Yu B. Iterative random forests to discover predictive and stable high-order interactions. Proc. of the National Academy of Sciences of the United States of America, 2018, 115(8): 1943–1948. [doi: 10.1073/pnas.1711236115]

[20] Liang SP, Pan WW, You DL, Liu Z, Yin L. Incremental deep forest for multi-label data streams learning. Applied Intelligence, 2022, 52(12): 13398–13414. [doi: 10.1007/s10489-022-03414-6]

[21] Liu WW, Wang HB, Shen XB, Tsang IW. The emerging trends of multi-label learning. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2022, 44(11): 7955–7974. [doi: 10.1109/TPAMI.2021.3119334]

[22] Behr M, Wang Y, Li X, Yu B. Provable boolean interaction recovery from tree ensemble obtained via random forests. Proc. of the National Academy of Sciences of the United States of America, 2022, 119(22): e2118636119. [doi: 10.1073/pnas.2118636119]

[23] Chen YH, Lyu SH, Jiang Y. Improving deep forest by exploiting high-order interactions. In: Proc. of the 2021 IEEE Int'l Conf. on Data Mining. Auckland: IEEE, 2021. 1030—1035. [doi: 10.1109/ICDM51629.2021.00118]

[24] Kocev D, Vens C, Struyf J, Džeroski S. Tree ensembles for predicting structured outputs. Pattern Recognition, 2013, 46(3): 817–833. [doi: 10.1016/j.patcog.2012.09.023]

[25] Nakano FK, Pliakos K, Vens C. Deep tree-ensembles for multi-output prediction. Pattern Recognition, 2022, 121: 108211. [doi: 10.1016/j.patcog.2021.108211]

[26] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504–507. [doi: 10.1126/science.1127647]

[27] Read J, Reutemann P, Pfahringer B, Holmes G. MEKA: A multi-label/multi-target extension to weka. Journal of Machine Learning Research, 2016 17(1): 667–671.

[28] Tsoumakas G, Vlahavas IP. Random *k*-labelsets: An ensemble method for multilabel classification. In: Proc. of the 18th European Conf. on Machine Learning. Warsaw: ECML, 2007. 406–417.

[29] Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition, 2007, 40(7): 2038–2048. [doi: 10.1016/j.patcog.2006.12.019]

[30] Benites F, Sapozhnikova E. HARAM: A hierarchical ARAM neural network for large-scale text classification. In: Proc. of the 2015 IEEE Int'l Conf. on Data Mining Workshop. Atlantic City: IEEE, 2015. 847–854. [doi: 10.1109/ICDMW.2015.14]

吕沈欢(1995－), 男, 博士, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘.

姜远(1976－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘.

陈一赫(1997－), 男, 硕士, 主要研究领域为机器学习, 数据挖掘.