计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2020.20200490
57(8): 1594-1604, 2020

蒙德里安深度森林

贺一笑 庞 明 姜 远
(计算机软件新技术国家重点实验室(南京大学) 南京 210023)
(软件新技术与产业化协同创新中心(南京大学) 南京 210023)
(heyx@lamda.nju.edu.cn)

Mondrian Deep Forest

He Yixiao, Pang Ming, and Jiang Yuan
(*National Key Laboratory for Novel Software Technology* (*Nanjing University*), *Nanjing* 210023)
(*Collaborative Innovation Center of Novel Software Technology and Industrialization* (*Nanjing University*), *Nanjing* 210023)

Abstract Most studies about deep learning are built on neural networks, i.e., multiple layers of parameterized differentiable nonlinear modules trained by backpropagation. Recently, deep forest was proposed as a non-NN style deep model, which has much fewer parameters than deep neural networks. It shows robust performance under different hyperparameter settings and across different tasks, and the model complexity can be determined in a data-dependent style. Represented by gcForest, the study of deep forest provides a promising way of building deep models based on non-differentiable modules. However, deep forest is now used offline which inhibits its application in many real tasks, e.g., in the context of learning from data streams. In this work, we explore the possibility of building deep forest under the incremental setting and propose Mondrian deep forest. It has a cascade forest structure to do layer-by-layer processing. And we further enhance its layer-by-layer processing by devising an adaptive mechanism, which is capable of adjusting the attention to the original features versus the transformed features of the previous layer, therefore notably mitigating the deficiency of Mondrian forest in handling irrelevant features. Empirical results show that, while inheriting the incremental learning ability of Mondrian forest, Mondrian deep forest has a significant improvement in performance. And using the same default setting of hyperparameters, Mondrian deep forest is able to achieve satisfying performance across different datasets. In the incremental training setting, Mondrian deep forest achieves highly competitive predictive performance with periodically retrained gcForest while being an order of magnitude faster.

Key words machine learning; deep forest; Mondrian forest; ensemble methods; incremental learning

摘 要 大多数有关深度学习的研究都基于神经网络,即可通过反向传播训练的多层参数化非线性可微模块.近年来,深度森林作为一种非神经网络深度模型被提出,该模型具有远少于深度神经网络的超参数.在不同的超参数设置下以及在不同的任务下,它都表现出非常鲁棒的性能,并且能够基于数据确定模型的复杂度.以 gcForest 为代表的深度森林的研究为探索基于不可微模块的深度模型提供了一种可行的方式.然而,深度森林目前是一种批量学习方法,这限制了它在许多实际任务中的应用,如数据流的

收稿日期:2020-06-16;**修回日期**:2020-07-02
基金项目:国家自然科学基金项目(61673201)
This work was supported by the National Natural Science Foundation of China (61673201).
通信作者:姜远(jiangyuan@nju.edu.cn)

应用场景.因此探索了在增量场景下搭建深度森林的可能性,并提出了蒙德里安深度森林.它具有级联森林结构,可以进行逐层处理.设计了一种自适应机制,通过调整原始特征和经过前一层变换后的特征的权重,以进一步增强逐层处理能力,更好地克服了蒙德里安森林在处理无关特征方面的不足.实验结果表明:蒙德里安深度森林在继承蒙德里安森林的增量训练能力的同时,显著提升了预测性能,并能够使用相同的超参数设置在多个数据集上取得很好的性能.在增量训练场景下,蒙德里安深度森林取得了与定期重新训练的 gcForest 接近的预测准确率,且将训练速度提升一个数量级.

关键词 机器学习;深度森林;蒙德里安森林;集成学习;增量学习

中图法分类号 TP181

深度学习使得算法模型能够学得逐层抽象的数据表示[1].目前,大多数深度学习模型都是基于神经网络构建的,即可以通过反向传播训练的多层参数化可微模块[2].近年来,深度神经网络在与图像和语音相关的任务中取得了巨大的成功[3-5].

文献[2]认为,深度学习成功的关键在于逐层的处理、模型内的特征变换和足够的模型复杂度.由此提出了深度森林的一种具体实现 gcForest[6],它同时满足上述 3 个条件,而基于不可微的模块搭建,验证了深度学习不仅仅是深度神经网络.gcForest 由决策树森林组成.和深度神经网络相比,它不依赖于反向传播进行训练,其模型复杂度可以根据训练数据自动确定,同时超参数少,而且对于不同超参数的设置和在不同的数据集上都有着稳健的性能表现.

考虑到在很多动态环境的实际应用中,会不断接收到新的训练样本,增量学习引起了广泛的关注[7-10].不过 gcForest 的训练过程要求所有训练数据预先给出,如果后续获得了新的训练数据,gcForest 无法直接更新模型,而定期重新训练 gcForest 会带来昂贵的训练时间开销.因此我们希望设计可以增量训练的深度森林.

目前有一些将决策树森林扩展到增量/在线学习的有效方法.大多数现有的在线决策树森林需要维护和更新每个叶节点上的候选划分列表和它们相应的划分质量得分,因而时间和空间开销较大[11-12].蒙德里安森林(Mondrian forest, MF)[13]使用蒙德里安过程[14]来构建集成中的蒙德里安树,和大部分决策树的划分过程不同,蒙德里安树的划分选择不依赖于样本标记.与其他在线随机森林相比,蒙德里安森林增量更新速度更快,同时有更好的预测准确率,在相同训练数据量的条件下,蒙德里安森林具有与批量版本的随机森林有竞争力的准确性.

尽管蒙德里安森林有很多优点,但有 2 个问题阻碍了其性能的进一步提高.首先,蒙德里安森林始终基于原始特征进行学习,我们发现向森林中添加更多蒙德里安树并不是提高准确率的有效方法.其次,由于划分选择独立于样本标记,所以当无关特征较多时,蒙德里安森林会选择大量的无关特征用于划分,而导致其预测性能不理想.

本工作中,我们提出了蒙德里安深度森林(Mondrian deep forest, MDF),它以一种级联的方式集成了蒙德里安森林,使其既有深度森林的预测准确性,又有蒙德里安森林增量学习的能力.本文的主要贡献有 2 个方面:

1) 使用蒙德里安森林的级联搭建蒙德里安深度森林,级联的每层接收原始特征和前一层输出的变换后特征作为输入.同时进一步提出了一种自适应机制,通过调整原始特征和变换后特征的权重进一步提升性能.蒙德里安深度森林不仅提升了多个数据集上的预测性能,同时也改善了蒙德里安森林无法处理大量无关特征的问题;

2) 首次将深度森林拓展到增量学习的设定中,有效降低了深度森林在每次接收到新样本后的训练时间.蒙德里安深度森林取得了和定期重新训练的 gcForest 有竞争力的预测准确率,同时训练速度提升了一个数量级.

1 相关工作

1.1 深度森林

深度森林是一种非神经网络的深度模型,其中 gcForest[6]是第 1 个深度森林模型.gcForest 有着级联森林结构,级联的每层由多个决策树森林组成,包括随机森林[15]和完全随机森林[16].其中的每个决策树森林输出它估计的类别分布,形成类概率向量.这些类概率向量作为增广特征,和原始输入特征拼接在一起,共同输入下一层.级联每增加新的一层,可通过交叉验证估计整个级联的性能.如果没有达到

要求的性能提升,则终止训练过程.因此,深度森林可以自动确定级联层数,即根据数据自动确定模型复杂度.

在实际应用中,已将 gcForest 在一个工业分布式机器学习平台上实现,并用于某大型企业的现实世界非法套现检测,其性能表现超过了包括深度神经网络在内的其他方法[17].理论方面,文献[18]将深度森林重新形式化为加性模型的形式,并从间隔理论的角度为深度森林提供了一种新的理解方式.此外,文献[19]和文献[20]分别将深度森林拓展至多示例和多标记的学习问题并取得了好的效果.

1.2 蒙德里安森林

在蒙德里安树中,各节点的划分选择不依赖于数据标记,这使其区别于绝大多数的决策树森林.在蒙德里安树的每个节点,根据节点内数据在各维度上的范围随机采样得到划分维度和划分点.同时蒙德里安树的每个节点与一个分裂时间对应.这样的划分机制和分裂时间机制使得蒙德里安树能够高效地更新.当一个新的训练样本出现,根据其与节点内已有数据的相对位置,蒙德里安树可以在 3 种方式中选择:1) 在当前划分之上引入一个更高层次的划分;2) 更新当前划分的范围使其包含新出现的训练样本;3) 将当前叶节点划分为 2 个子节点.也就是说,蒙德里安树可以对整棵树的结构进行修改,而其他增量随机森林只能更新叶节点[11-12].对于一个测试样本,一棵蒙德里安树输出各标记上的预测分布.蒙德里安森林是多个独立训练的蒙德里安树的集成,它的输出是其中各棵树的预测值的平均.

此外,蒙德里安森林还被应用于大规模回归任务中[21].文献[22]中得到了关于蒙德里安森林的一致性的理论结果,在随机森林的理论方面有所推进.

2 蒙德里安深度森林

本节我们提出了蒙德里安深度森林,它将增量学习的能力融入了级联森林的结构中.

2.1 级联森林结构

蒙德里安深度森林具有级联森林的结构,级联的每一层含有多个蒙德里安森林,它们的输入是经过前面的级联层处理后的特征信息,并将经过该层处理后的特征信息输出给下一层.

设 $\mathcal{D}$ 是 $\mathcal{X}\times\mathcal{Y}$ 上的联合分布,其中 $\mathcal{X}$ 是 D 维样本空间,$\mathcal{Y}=\{1,2,\cdots,C\}$ 是标记空间,训练和测试样本都是从 $\mathcal{D}$ 中独立同分布采样得到.令 $\boldsymbol{h}=(h_1,h_2,\cdots,h_T)$,$\boldsymbol{f}=(f_1,f_2,\cdots,f_T)$,其中 h_t 表示第 t 层蒙德里安森林,f_t 表示前 t 层蒙德里安森林的级联结构,则一个蒙德里安深度森林可以被形式化为 $(\boldsymbol{h},\boldsymbol{f})$.这里的 T 是级联总的有效层数.

在第 t 层,$f_t:\mathcal{X}\to\mathcal{Z}$ 的定义为

$$f_t(\boldsymbol{x})=\begin{cases}h_1(\boldsymbol{x}), & t=1;\\ h_t([\boldsymbol{x},f_{t-1}(\boldsymbol{x})],\alpha), & t>1.\end{cases}\tag{1}$$

假设每层只有一个蒙德里安森林,那么第 t 层 $h_t(\cdot)$ 的输出,也即 $f_t(\cdot)$ 的输出,是一个类概率向量 $(p_1,p_2,\cdots,p_C)$.h_1 的输入是原始特征 $\boldsymbol{x}$,而后每一层 h_t 的输入是原始特征 $\boldsymbol{x}$ 和前一层输出的变换后特征 $f_{t-1}(\boldsymbol{x})$ 拼接成的向量.我们称 $f_{t-1}(\boldsymbol{x})$ 为增广特征,如果每层含有多个蒙德里安森林,那么多个类概率向量会被拼接在一起共同作为增广特征.注意到与 gcForest 不同,这里我们引入自适应因子 α 来调整原始特征和增广特征的权重.

如图 1 所示,假设有 3 个类要预测,那么对于每个样本,每个蒙德里安森林将输出一个 3 维的类

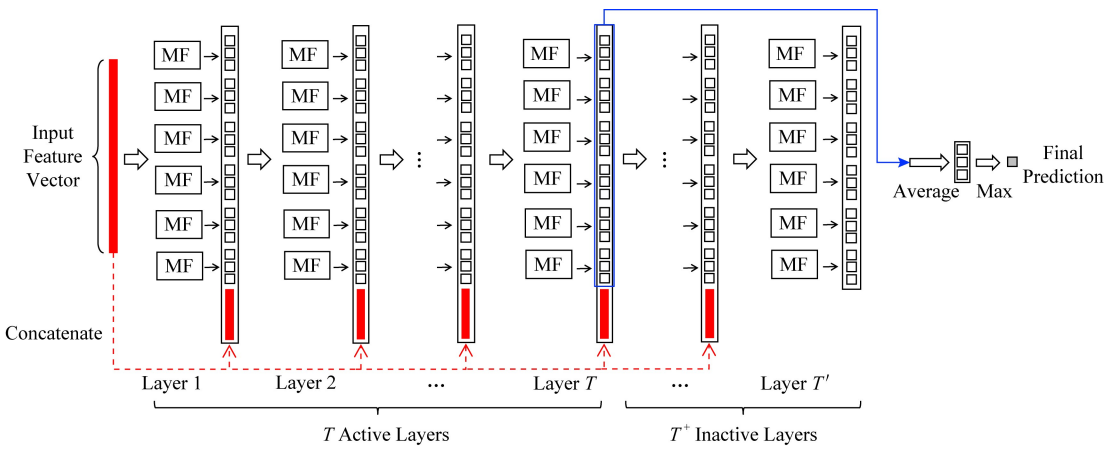


Fig. 1 Illustration of the model structure of Mondrian deep forest

图 1 蒙德里安深度森林结构

概率向量.假设级联的每一层含有 6 个蒙德里安森林,那么下一层在接收原始输入特征的同时还将接收 3×6=18 维增广特征.

每对$(\boldsymbol{h},\boldsymbol{f})$定义了一个蒙德里安深度森林模型$g:\mathcal{X}\to\mathcal{Y}$:

$$g(\boldsymbol{x})=\underset{c\in\{1,2,\cdots,C\}}{\arg\max}[f_T(\boldsymbol{x})]_c. \tag{2}$$

参考 gcForest 为了降低过拟合风险所采取的做法,我们使用交叉验证来生成类概率向量[2].与此同时,交叉验证的准确率可以用来估计级联增长至当前层的预测性能.如果新增长的层数没有带来一定的性能提升,那么训练过程将自动终止.这样蒙德里安深度森林自动确定了级联层数 T,也即实现了自适应地根据训练数据调整模型复杂度.

在一棵决策树中,每个内部节点 j 对应于一个划分(δ_j,ξ_j),其中 $\delta_j\in\{1,2,\cdots,D\}$表示划分维度,$\xi_j$ 表示该维度上的划分点.令 $R_1(j)$和 $R_r(j)$表示 2 个子节点,一个划分的定义为

$$\begin{aligned}R_1(j)&=\{\boldsymbol{x}\,|\,x_{\delta_j}\leqslant\xi_j\},\\ R_r(j)&=\{\boldsymbol{x}\,|\,x_{\delta_j}>\xi_j\}.\end{aligned} \tag{3}$$

对于每个节点 j,令 ℓ_{jd}和 u_{jd}分别表示节点 j 中训练数据沿维度 d 的最小值和最大值,并计算沿各维度 d 的数据范围 $r_{jd}=u_{jd}-\ell_{jd}$.在蒙德里安树中,以正比于节点内数据在各维度上范围 r_{jd}的概率采样得到划分维度 δ_j,即

$$Pr(\delta_j=d)=\frac{r_{jd}}{\sum\limits_{d\in\{1,2,\cdots,D\}}r_{jd}}. \tag{4}$$

其中,Pr 表示概率.

换句话说,某维度上的数据范围越大,该维度就越有可能被选为划分维度.通过归一化数据集可使得各维度在初始时有相等的概率被选为划分特征.而后从$[\ell_{j\delta_j},u_{j\delta_j}]$中均匀采样得到划分点 ξ_j.可以看出,划分选择是不依赖于样本标记的,当无关特征的比例很大时,每个划分选中有用特征的概率很低,这使得蒙德里安森林无法处理含有大量无关特征的数据集.

在蒙德里安深度森林中,每层接收前一层输出的类概率向量作为增广特征,通过级联结构逐层提升准确率.更进一步,我们在级联结构中引入了自适应机制来平衡原始特征和增广特征的权重,从而防止了增广特征被大量无关原始特征淹没,改善了蒙德里安森林在处理无关特征方面的不足.

我们提出了一种自适应机制,通过自适应因子 α,在随机采样划分特征的过程中调整增广特征和原始输入特征的权重.假设增广特征的维数是 D',那么和原始特征拼接后的总维数为 $D+D'$.则选中某个原始输入特征作为划分特征的概率为

$$Pr(\delta_j=d)=\frac{r_{jd}}{\sum\limits_{d=1}^{D}r_{jd}+\alpha\sum\limits_{d=D+1}^{D+D'}r_{jd}}, \tag{5}$$

$$d\in\{1,2,\cdots,D\},$$

选中某个增广特征的概率为

$$Pr(\delta_j=d)=\frac{\alpha r_{jd}}{\sum\limits_{d=1}^{D}r_{jd}+\alpha\sum\limits_{d=D+1}^{D+D'}r_{jd}}, \tag{6}$$

$$d\in\{D+1,D+2,\cdots,D+D'\}.$$

自适应因子 α 是一个可调参数,其设置方式有多种可能性.比如,可以逐层增大或者减小,或者每一层根据不同设置下的交叉验证准确率自适应地决定.本文采用一种简单的平衡策略,即对每个数据集,设置固定的自适应因子 $\alpha=D/D'$,则:

$$\begin{aligned}&Pr(\delta_j\in\{1,2,\cdots,D\})=\\ &Pr(\delta_j\in\{D+1,D+2,\cdots,D+D'\}).\end{aligned} \tag{7}$$

也就是说,算法在随机采样划分特征时,选中原始特征和选中增广特征的概率相同,可以较为均衡地结合二者的信息.第 3 节中展示了这样简单的设置方式在不同数据集上有稳定的好的表现.

2.2 批量训练和增量训练

图 1 展示了蒙德里安深度森林的模型结构.级联森林结构共有 T'层,但仅前 T 层在预测时被激活.这里的 T 根据 2.1 节中描述的方式在训练过程中自适应地确定.在批量训练的设定下,$T'=T$.

在增量训练的设定中,数据随时间分批到达,目标是用新获得的训练数据及时对模型进行更新,使模型充分利用已有的训练数据以期达到尽可能好的预测性能.深度森林 gcForest[6] 是一个批量学习模型.而对于蒙德里安深度森林,通过逐层更新其中的蒙德里安森林,可以更新整个级联结构.加上动态的对于有效层数的调整,我们得到了一个增量版本的蒙德里安深度森林.它可以从相对简单的模型开始,随着获得更多的训练数据逐步增加模型复杂度以提升性能.

用 $\mathcal{A}$ 表示生成级联森林结构中一层蒙德里安森林 h_t 的算法.设训练数据分为 K 小批依次到达,即$\{S^k\}_{k=1}^K$,h_t^k 表示使用第 k 批训练数据更新后的第 t 层蒙德里安森林.也就是说,当获得一批新的训练数据 S^k 时,第 t 层模型 h_t^{k-1} 将被更新为 h_t^k,更新方式为

$$h_t^k \leftarrow \begin{cases} \mathcal{A}(h_t^{k-1}, S^k), & t=1; \\ \mathcal{A}(h_t^{k-1}, S^k, f_{t-1}^k, \alpha), & t>1. \end{cases} \tag{8}$$

算法 1 描述了蒙德里安深度森林的增量训练过程.当接收到第 1 批训练数据 S^1 时,从零开始训练一个蒙德里安深度森林.设估计的最优层数是 T,考虑到随着训练数据的增加可能需要更高的模型复杂度,我们在 T 层之后训练额外的 T^+层,它们暂时不参与预测,属于未激活的级联层,则级联的总层数为 $T'=T+T^+$,如图 1 所示.当获得新的训练数据 S^k ($k>1$)时,我们逐层更新蒙德里安森林.在更新过程中,全部 T'层都被更新.同时,当前批训练样本的交叉验证准确率随层数的变化趋势可用于决定是否激活更多的级联层用于预测.

在批量训练的设定下,算法 1 中的第 1 批训练数据 S^1 即包含了全部训练数据,模型不需要进行更新,将预训练备用层数 T^+ 设为 0,即得到了蒙德里安深度森林的批量版本.

算法 1. 蒙德里安深度森林.

输入:分为 K 小批到达的训练数据$\{S^k\}_{k=1}^K$、自适应因子 α、预训练备用层数 T^+、层数最大值 $T_{\max}$;

输出:蒙德里安深度森林模型 g^k.

① 初始化:$val_0=0$, $T'=\infty$;
② for $k\in\{1,2,\cdots,K\}$ do
③ 接收新到来的一批样本 S^k;
④ $t=1$;
⑤ while $t\leqslant\min(T_{\max},T')$ do
⑥ 根据式(8)得到 h_t^k;
⑦ 根据式(1)得到 f_t^k;
⑧ 计算交叉验证的准确率 val_t;
⑨ if $k=1\wedge T'=\infty\wedge val_t\leqslant val_{t-1}$ then
⑩ $T=t-1$; $T'=T+T^+$; /* 用第 1 批训练数据确定层数 */
⑪ end if
⑫ if $k>1\wedge t>T\wedge val_t>val_T$ then
⑬ $T=t$; /* 更新激活层数 */
⑭ end if
⑮ $t=t+1$;
⑯ end while
⑰ 根据式(2)得到 g^k;
⑱ end for

3 实 验

我们提出的蒙德里安深度森林(MDF)是一种可以进行增量学习的深度森林,本节我们将 MDF 与蒙德里安森林(MF)和 gcForest 进行比较.根据文献[13],在线随机森林方法 ORF-Denil[11] 和 ORF-Saffari[12] 性能接近,而 MF 和 ORF-Saffari 相比,训练时间短并且准确率更高,因此我们将 MF 作为基线方法.而 gcForest 是一种批量训练的深度森林,它能够在多种任务上达到非常优秀的性能表现[2,6],我们将使用它的准确率和训练时间作为参考.实验中,我们首先验证了 MDF 的分类准确率显著优于 MF,并且和 gcForest 很接近,接着展示了 MDF 增量训练的过程,对不同的方法在增量学习过程中的测试准确率和训练时间进行了比较.

3.1 实验设置和数据集

对所有数据集,MDF 使用相同的级联结构,具体来说,每层含有 6 个蒙德里安森林,每个森林含有 20 棵蒙德里安树.每个森林输出的类概率向量由 3 折交叉验证生成,交叉验证的准确率被用于估计级联增长至各层数时的预测性能.如果某层后续 3 层的性能估计没有提升则训练过程终止,取该层作为输出层.在 MDF 的增量设定中,我们令 $T^+=3$,也就是说,随着训练数据的增加,MDF 预测时的激活层数可以自适应地增加至多 3 层.作为对比的 MF 采用了 120 棵树和 2 000 棵树的配置,后文分别用 MF_{120} 和 MF_{2000} 表示.作为对比的 gcForest 采用和 MDF 同等的实验配置,即每层由 3 个随机森林和 3 个完全随机森林组成,每个森林含有 20 棵树,级联层数的确定方式也和 MDF 相同.在增量训练的实验中,考虑到 MDF 是基于文献[13]的 Python 版本的 MF 搭建的,我们自己实现了 Python 版本的随机森林和完全随机森林,基于它们搭建了 gcForest 以便于比较训练时间.

MF 含有 2 个超参数.考虑到实验的可比性,我们使用和文献[13]相同的设置.具体来说,一个是生长期限 λ,我们令 $\lambda=\infty$使得该参数不会限制蒙德里安树的深度.另一个是层次规范化稳定过程[23]中的参数 γ,令 $\gamma=10D$,其中 D 表示特征维数.MDF 中作为基本单元的 MF 和作为对比方法的 MF_{120} 和 MF_{2000} 都使用上述的参数设置.此外,MDF 还含有额外的参数 α,参考 2.1 节,我们令 $\alpha=D/D'$,其中 D'表示增广特征的维数.

实验使用和文献[13]相同的数据集,即 USPS, SATIMAGE, LETTER 和 DNA[24].并参照文献[13]的做法,抽取 DNA 数据集中的第 61～120 维特征生成 DNA60 数据集,这是因为 MF 无法处理大量的无关特征,所以需要选取最相关的 60 维特征.但我们同时保留了原本的 DNA 数据集,以展示

MDF 可以处理含有无关特征的数据集.我们使用和文献[13]相同的训练集/测试集划分,具体信息如表 1 所示:

Table 1 Dataset Information

表 1 数据集信息

Dataset	# Dim	# Label	N_{train}	N_{test}
USPS	256	10	7 291	2 007
SATIMAGE	36	6	3 104	2 000
LETTER	16	26	15 000	5 000
DNA60	60	3	1 400	1 186
DNA	180	3	1 400	1 186

3.2 批量设定下的实验结果

我们比较了 MDF 和 MF_{120},MF_{2000},gcForest 的性能,表 2 中展示了在每个数据集上进行 10 次实验得到的测试准确率的均值和标准差,括号中是级联结构的平均层数.注意到 MF_{120} 等价于我们实验设置下的一层 MDF,而 MF_{2000} 含有比 MDF 更多的树,即有更高的模型复杂度.表格中用加粗字体标注了每个数据集上平均准确率最高的结果.

比较表 2 中 MF_{120} 和 MF_{2000} 的性能,我们发现增加树的棵数难以提升准确率.而 MDF 的测试准确率在 5 个数据集上均显著优于 MF_{120} 和 MF_{2000},这说明搭建深度模型是有效的提升性能的方式.与此同时,根据 MDF 在 5 个数据集上的平均层数,算得分别平均用了 792,420,528,456,1 248 棵树.也就是说,MDF 使用比 MF_{2000} 更少的树达到了更好的预测性能.如果把蒙德里安树看作基本的结构单元,则实验结果验证了搭建深度模型是比搭建更宽的模型更为有效的提升性能的方式.

Table 2 Mean Test Accuracy (%) and Standard Deviation

表 2 批量设定下的测试准确率(%)均值和标准差

Dataset	MF_{120}	MF_{2000}	MDF	gcForest
USPS	93.14±0.29	93.50±0.07	**94.62±0.26 (6.6)**	94.46±0.07 (6.9)
SATIMAGE	89.45±0.26	89.59±0.12	**91.02±0.45 (3.5)**	90.69±0.25 (3.1)
LETTER	96.57±0.11	96.94±0.09	**97.19±0.11 (4.4)**	96.53±0.12 (1.8)
DNA60	90.89±0.38	91.79±0.26	94.49±0.23 (3.8)	**95.08±0.34 (3.0)**
DNA	66.79±0.72	65.11±0.23	92.01±0.71(10.4)	**94.74±0.43 (7.2)**

Note: The best results are in bold; the average layer numbers of the cascade are in the parentheses.

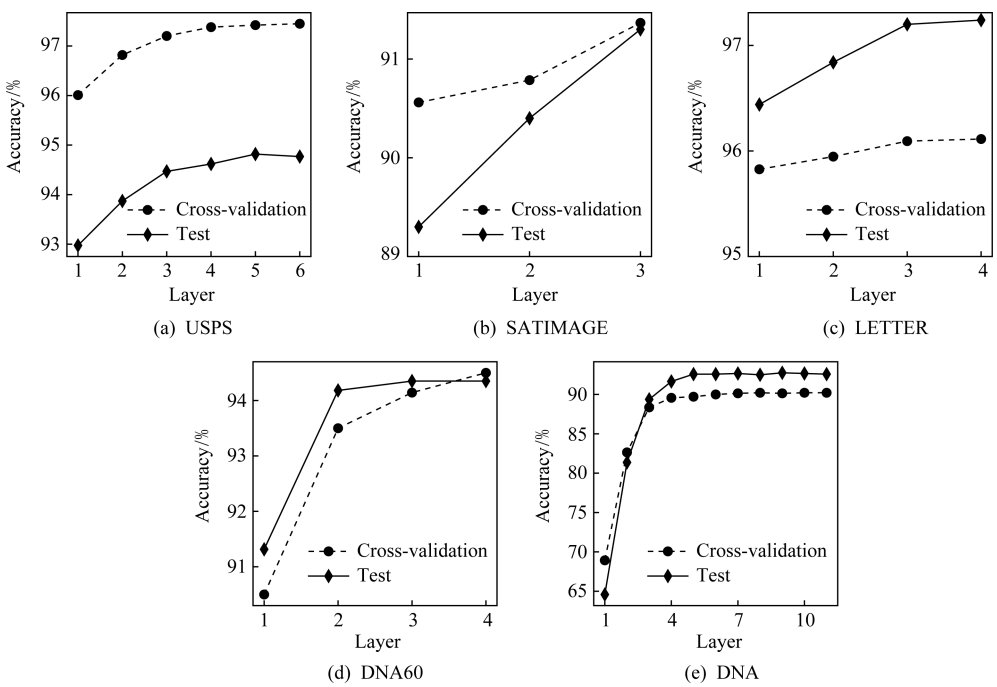


Fig. 2 Cross-validation and test accuracy at each layer of MDF

图 2 MDF 交叉验证和测试准确率随层数的变化

MDF 训练过程中交叉验证准确率和测试准确率随层数的变化如图 2 所示.虚线表示训练时交叉验证的准确率随层数的变化情况,横坐标显示至根据交叉验证准确率估计的最优级联层数,可以看到在有效层数内交叉验证准确率逐层上升,不同数据集最终确定的级联层数不同.实线展示了测试集上的准确率,其变化趋势与交叉验证准确率大致相同,因此,根据数据自动确定模型复杂度是有效的.

从表 2 中我们还可以看出,在同等配置下,MDF 有着和 gcForest 接近的预测性能,其中在 USPS, SATIMAGE, LETTER 这 3 个数据集上 MDF 准确率超过了 gcForest.文献[6]中指出,易于调参是 gcForest 相对于深度神经网络的重要优势,gcForest 可以使用同样的级联结构在多种任务中达到好的性能表现.而 MDF 和 gcForest 同为深度森林,尽管 MDF 增加了自适应机制,也可以使用固定的设置策略,在不同数据集上都达到好的结果.

3.2.1 自适应因子的影响

表 3 中将 $\alpha=D/D'$ 的 MDF 与 $\alpha=1$ 的 MDF 进行了 10 次测试准确率的对比,这里 $\alpha=1$ 时等价于令 MDF 的自适应机制不起作用.可以看到,在 4 个数据集上,设置 $\alpha=D/D'$ 的 MDF 平均测试准确率高于 $\alpha=1$ 的 MDF.而在 DNA60 数据集上,尽管不使用自适应机制的 MDF 平均测试准确率更高,但它们并没有显著的区别.因此,将自适应因子设置为 D/D' 的方式虽然简单,但在实验中是比较有效的.

Table 3 Test Accuracy of MDF with Different Adaptive Factors

表 3 使用不同自适应因子的 MDF 测试准确率对比 %

Dataset	$\alpha=1$	$\alpha=D/D'$
USPS	94.04±0.14 (8.2)	94.62±0.26 (6.6)
SATIMAGE	90.61±0.68 (3.7)	91.02±0.45 (3.5)
LETTER	96.93±0.13 (2.5)	97.19±0.11 (4.4)
DNA60	94.68±0.30 (6.6)	94.49±0.23 (3.8)
DNA	70.66±0.33 (7.3)	92.01±0.71(10.4)

Note: The average number of layers is in the parentheses.

图 3 中以 DNA 数据集为例,对比了不同自适应因子设置下训练时交叉验证准确率和测试准确率随层数的变化情况,为便于比较,统一了横纵坐标轴的范围.在 $\alpha=1$, $\alpha=5$ 和 $\alpha=20$ 这 3 种设置中,$\alpha=1$ 时算法主要依赖原始特征,无法得到有效的性能提升,一直在较低的准确率徘徊.$\alpha=20$ 时算法最为依赖增广特征,可以看到前 2 层的性能提升非常明显,但很快准确率便停止增长.$\alpha=5$ 时算法能够在前 5 层都维持较为稳定的增长幅度,最终达到较高的准确率.因此,自适应因子 α 的设置比较关键,过大或过小都不利于最终达到很好的性能.我们注意到,对于实验设定下的 DNA 数据集,$D/D'=10$,但 $\alpha=5$ 时的 MDF 性能最好,其 10 次平均测试准确率为 92.53%.因此,虽然设置 $\alpha=D/D'$ 常常可以得到很好的性能,但最优的 α 取值和具体的数据有关,若进行调参可能可以达到更好的性能.

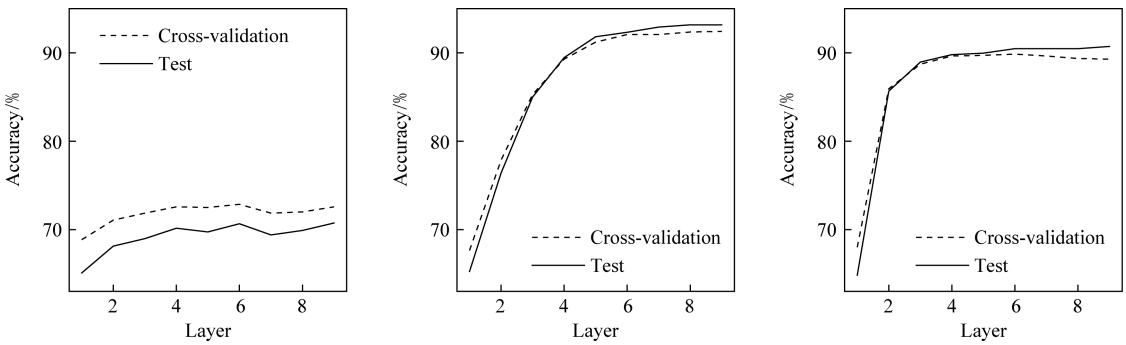


Fig. 3 Cross-validation and test accuracy of MDF on DNA using different adaptive factors

图 3 在 DNA 数据集上使用不同自适应因子时 MDF 交叉验证和测试准确率随层数的变化情况

3.2.2 处理无关特征

文献[13]指出,MF 几乎无法处理 DNA 数据集,因为其含有大量无关特征.而 MDF 因为其逐层处理和自适应机制,有效地提升了性能.

我们进一步在人造数据集上对比了 MF 和 MDF 处理无关特征的能力.和 DNA 数据集相比,人造数据集上无关特征的生成过程可以准确地控制,这里的无关特征也就是在确定真实标记的过程中没有起作用的特征.人造数据集的所有维度都是独立生成的 0 和 1 等概率取值的随机布尔型变量,第 i 个样本

的真实标记 $y_i=x_i^{(1)} \vee x_i^{(2)}$,也就是说样本的真实标记仅由前 2 个维度决定,如果样本含有更多维度的特征,那么其余维度均为无关特征.生成的训练样本数为 200,测试样本数为 500,无关特征的数目从 0 变化至 20,每组实验条件重复 10 次,图 4 展示了测试准确率均值和方差随无关特征数目的变化.本实验中 MDF 在各组条件下自动确定的层数平均值都小于 5 层,而使用的 MF 含有 2 000 棵树,远远超过 MDF 的模型复杂度.观察图 4 中的准确率曲线我们发现,随着相关特征所占的比例降低,MF 的性能下降非常明显,而 MDF 的测试准确率保持相对稳定.当无关特征达到 20 维时,MF 的平均测试准确率只有 79.52%,而 MDF 为 97.68%.

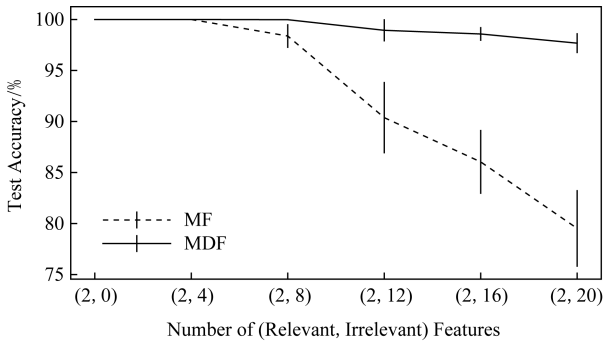


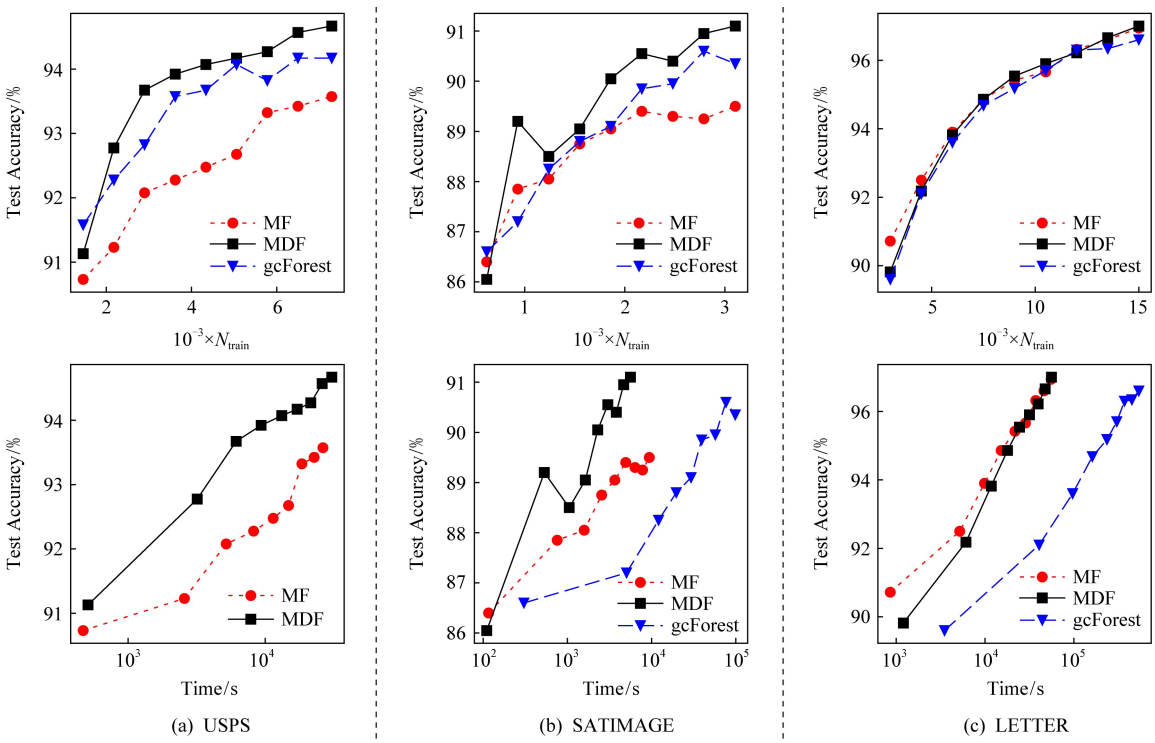
Fig. 4 The influence of increasing irrelevant features on the predictive performance of MF and MDF

图 4 无关特征的增加对于 MF 和 MDF 预测性能的影响

3.3 增量设定下的实验结果

考虑到 MF_{2000} 中树的棵数超过 MDF,我们将 MF_{2000} 作为基线方法来和 MDF 比较.同时,为了比较增量训练的 MDF 和批量训练的 gcForest 的性能,我们在 gcForest 的实验中存储当前获得的全部训练数据,每当有新的训练数据到来,把已有的训练数据和新的训练数据合并,重新训练一个 gcForest.训练数据被分为 81 小批,其中第 1 批包含 20%的训练数据,剩余训练数据被等分至后续 80 批中,每一批训练数据到来时 MDF 和 MF 都会进行更新,gcForest 进行重新训练,分别记录累计的更新或重新训练的时间.

图 5 展示了测试准确率的变化情况,每 10 个小批记录一次.图 5(a)~(e)的上层图展示了测试准确率随训练样本数的变化情况,可以看出,在 USPS,SATIMAGE,DNA60 和 DNA 数据集上,给定相同的训练样本数,MDF 的测试准确率都明显好于 MF_{2000}.图 5(a)~(e)的下层图展示了测试准确率随训练时间的变化情况.在 USPS 数据集上,gcForest 没有在 10^6 s 内完成训练,因此未在图中展示.可以看出,在训练时间上 MDF 与 MF_{2000} 接近,比定期重新训练的 gcForest 快一个数量级.值得注意的是,增量训练的 MDF 有着与定期重新训练的 gcForest 接近的测试准确率,甚至在 USPS,SATIMAGE,LETTER 数据集上还略好于 gcForest.



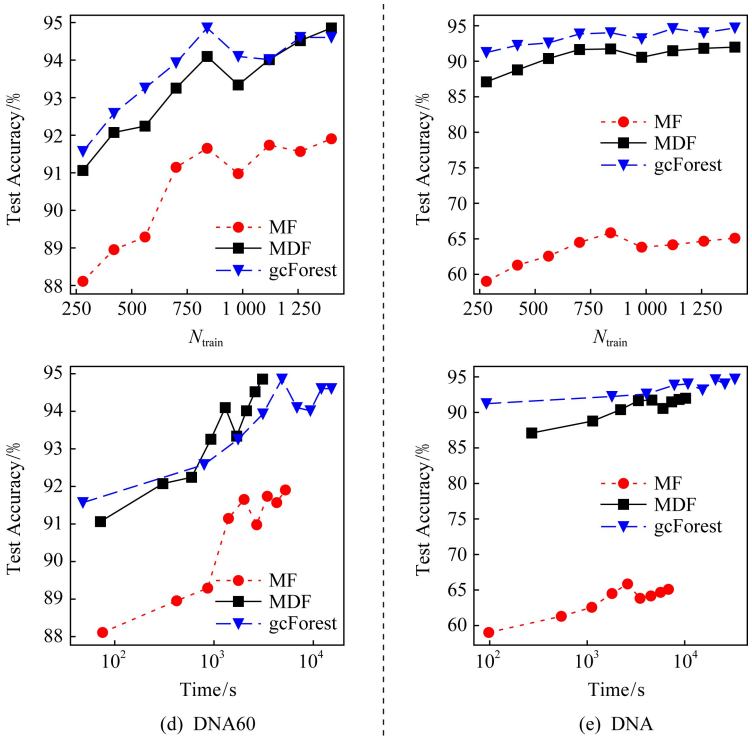


Fig. 5 Results on various datasets in the incremental setting

图 5 增量设定下在多个数据集上的实验结果

图 6 对比了自适应增加有效层数与固定有效层数的 MDF 测试准确率随训练样本数的变化情况.可以看出,在 USPS,SATIMAGE,LETTER 和 DNA60 数据集上,激活更多的级联层可以明显提升性能,由

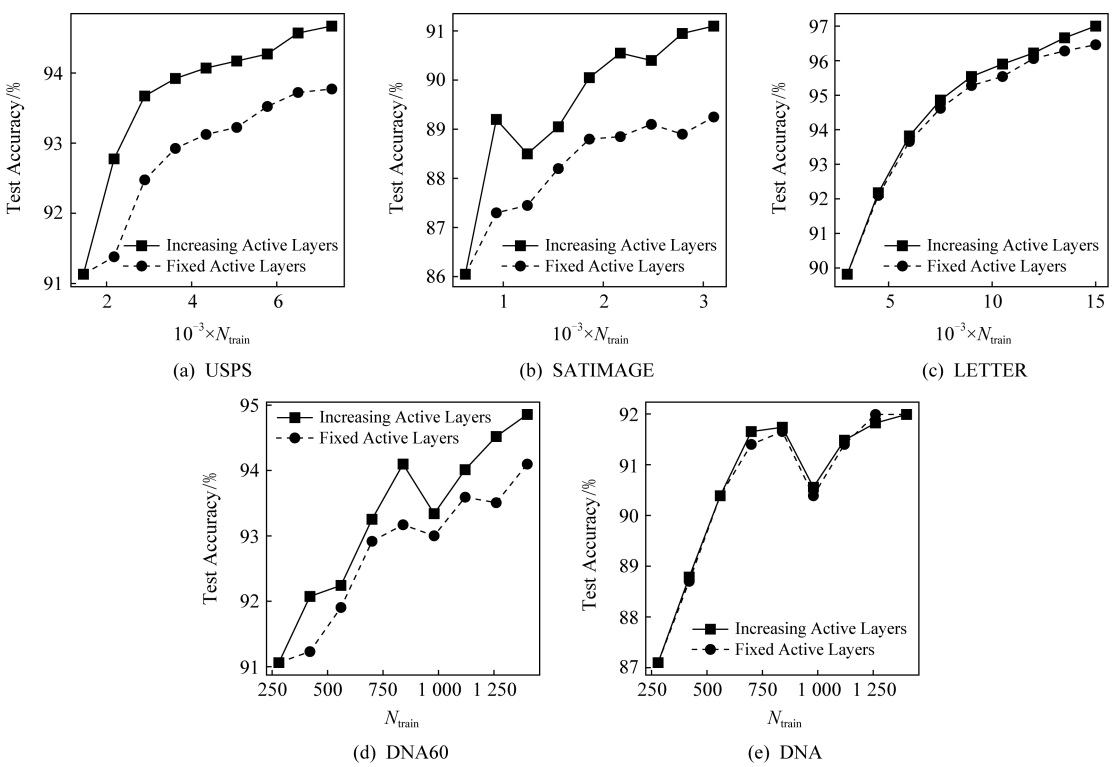


Fig. 6 The effect of adaptively increasing the number of active layers

图 6 自适应增加有效层数的作用

此说明了动态调整有效层数的作用.

4 讨 论

深度森林 gcForest 在级联结构中同时使用了随机森林和完全随机森林 2 种基本模块以提升多样性,因为在集成学习中,要得到一个好的集成,基学习器应当尽可能准确,同时有尽量高的多样性[25].然而,蒙德里安深度森林仅使用蒙德里安森林作为基学习器,仍然达到了和 gcForest 接近的准确率.从蒙德里安树的训练过程可以看出,蒙德里安森林有着和完全随机森林接近的随机性,又有着接近随机森林的准确率,这使得蒙德里安森林可以用于搭建同质深度森林.

在批量训练和增量训练下得到的蒙德里安森林的预测精度是相同的[13].而增量训练的蒙德里安深度森林的预测精度要略低于其批量训练版本.这是因为早先到达的训练样本对应的类概率向量已经被用于训练后续的级联层,随着新的训练数据的到达,每一层蒙德里安森林都被更新,但是与先前样本对应的旧的类概率向量已经造成的影响不会被更正.不过,蒙德里安深度森林的性能仍然比蒙德里安森林好许多.

注意到 gcForest 较高的内存和时间开销约束了大模型的训练,文献[26]提出了置信度筛选的方法,将高置信度的样本直接传递到最后一级,而不是遍历所有层级,同时每层的模型复杂度逐渐增加,由此将 gcForest 的时间和空间开销降低了一个数量级.蒙德里安深度森林未来也有可能借鉴类似的方法以降低训练的时间和空间开销,从而可以搭建更大规模的模型以进一步提升预测性能.

5 总 结

本文提出了蒙德里安深度森林.它是可以增量学习的深度森林模型,每当获得新的训练数据,它可以基于当前模型进行更新,以提升性能,而不需要重新训练.与此同时,它以蒙德里安森林为基本单元,但能够通过级联森林结构和自适应机制逐层提升预测性能,并且克服了蒙德里安森林易被无关特征干扰的问题,达到了和 gcForest 接近的准确率.蒙德里安深度森林保持了深度森林超参数少且性能鲁棒的优点,并且模型复杂度可以在训练过程中根据数据自动确定.增量设定下的实验表明,蒙德里安深度森林比定期重新训练的 gcForest 快一个数量级,并且达到了与之接近的预测准确率.

参 考 文 献

[1] LeCun Y, BengioY, Hinton G. Deep learning [J]. Nature, 2015, 521(7553): 436-444

[2] Zhou Zhihua, Feng Ji. Deep forest [J]. National Science Review, 2019, 6(1): 74-86

[3] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups [J]. IEEE Signal Processing Magazine, 2012, 29(6): 82-97

[4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C] // Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2012: 1097-1105

[5] Litjens G, Kooi T, Bejnordi B E, et al. A survey on deep learning in medical image analysis [J]. MedicalImage Analysis, 2017, 42: 60-88

[6] Zhou Zhihua, Feng Ji. Deep forest: towards an alternative to deep neural networks [C] //Proc of the 26th Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2017: 3553-3559

[7] Zhou Zhihua, Chen Zhaoqian. Hybrid decision tree [J]. Knowledge-based Systems, 2002, 15(8): 515-528

[8] Gepperth A, Hammer B. Incremental learning algorithms and applications [C] //Proc of the 2016 European Symp on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: ESANN, 2016: 357-368

[9] Krawczyk B, Minku L L, Gama J, et al. Ensemble learning for data stream analysis: A survey [J]. Information Fusion, 2017, 37: 132-156

[10] Losing V, Hammer B, Wersing H. Incremental on-line learning: A review and comparison of state of the art algorithms [J]. Neurocomputing, 2018, 275: 1261-1274

[11] Denil M, Matheson D, Freitas N. Consistency of online random forests [C] //Proc of the 30th Int Conf on Machine Learning. New York: ACM, 2013: 1256-1264

[12] Saffari A, Leistner C, Santner J, et al. On-line random forests [C] //Proc of the 12th IEEE Int Conf on Computer Vision Workshops. Piscataway, NJ: IEEE, 2009: 1393-1400

[13] Lakshminarayanan B, Roy D M, Teh Y W. Mondrian forests: Efficient online random forests [C] //Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 3140-3148

[14] Roy D M, Teh Y W. The Mondrian process [C] //Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2009: 1377-1384

[15] Breiman L. Random forests [J]. Machine Learning, 2001, 45 (1): 5-32

[16] Liu F T, Ting K M, Yu Yang, et al. Spectrum of variable-random trees [J]. Journal of Artificial Intelligence Research, 2008, 32: 355-384

[17] Zhang Yalin, Zhou Jun, Zheng Wenhao, et al. Distributed deep forest and its application to automatic detection of cash-out fraud [J]. ACM Transactions on Intelligent Systems and Technology, 2019, 10(5): No.55

[18] Lyu Shenhuan, Yang Liang, Zhou Zhihua. A Refined Margin Distribution Analysis for Forest Representation Learning [C] //Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2019: 5531-5541

[19] Yang Liang, Wu Xizhu, Jiang Yuan, et al. Multi-Label Learning with Deep Forest [J]. arXiv preprint, arXiv:1911.06557, 2019

[20] Ren Jie, Hou Bojian, Jiang Yuan. Deep forest for multiple instance learning [J]. Journal of Computer Research and Development, 2019, 56(8): 1670-1676 (in Chinese)
(任婕, 侯博建, 姜远. 多示例学习下的深度森林架构[J]. 计算机研究与发展, 2019, 56(8): 1670-1676)

[21] Lakshminarayanan B, Roy D M, Teh Y W. Mondrian forests for large-scale regression when uncertainty matters [C] // Proc of the 19th Int Conf on Artificial Intelligence and Statistics. Brookline, MA: Microtome Publishing, 2016: 1478-1487

[22] Mourtada J, Gaïffas S, Scornet E. Universal consistency and minimax rates for online Mondrian Forests [C] //Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2017: 3758-3767

[23] Wood F, Archambeau C, Gasthaus J, et al. A stochastic memoizer for sequence data [C] //Proc of the 26th Int Conf on Machine Learning. New York: ACM, 2009: 1129-1136

[24] Chang C C, Lin C J. LIBSVM: A library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): No.27

[25] Zhou Zhihua. Ensemble Methods: Foundations and Algorithms [M]. London: Chapman & Hall, 2012

[26] Pang Ming, Ting K M, Zhao Peng, et al. Improving deep forest by confidence screening [C] //Proc of 2018 IEEE Int Conf on Data Mining. Piscataway, NJ: IEEE, 2018: 1194-1199

He Yixiao, born in 1997. PhD candidate. Her main research interests include machine learning and data mining.

Pang Ming, born in 1992. PhD candidate. His main research interests include machine learning and data mining. (pangm@lamda.nju.edu.cn)

Jiang Yuan, born in 1976. PhD, professor in the Department of Computer Science and Technology at Nanjing University. Her main research interests include machine learning and data mining.