梯度优化决策树的集成学习及其应用

王延斌 武优西 刘洪普

(河北工业大学人工智能与数据科学学院 天津 300401) (河北省大数据计算重点实验室 天津 300401)

摘 要 集成学习通过构建具有一定互补功能的多个分类器来完成学习任务,以减少分类误差。但是当前研究未能考虑分类器的局部有效性。为此,在基于集成学习的框架下,提出了一个分层结构的多分类算法。该算法按预测类别分解问题,在分层的基础上,集成多个分类器以提高分类准确度。在美国某高校招生录取这一个实际应用的数据集及 3 个 UCI 数据集上进行实验,实验结果验证了该算法的有效性。

关键词 集成学习,分类器融合,梯度优化,层次化结构

中图法分类号 TP181 **文献标识码** A

Research and Application of Ensemble Learning Using Gradient Optimization Decision Tree

WANG Yan-bin WU You-xi LIU Hong-pu

(School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China)

(Hebei Province Key Laboratory of Big Data Calculation, Tianjin 300401, China)

Abstract Ensemble learning completes the learning task by building multiple classifiers with certain complementary performance to reduce the classification error. However, the current research fails to consider the local validity of the classifier. In this paper, a hierarchical multi-class classification algorithm was proposed in the framework of ensemble learning. The algorithm decomposes the problem by predicted category, and integrates several weak classifiers on the basis of stratification to improve the prediction accuracy. The experimental results on a real data set of American College Matriculation Set and three UCI datasets verified the effectiveness of the algorithm.

Keywords Ensemble learning, Classifier fusion, Gradient optimization, Hierarchical structure

1 引言

集成学习[1]通过构建并结合多个分类器来完成学习任务,也被称为多分类器系统。集成学习通过将多个分类器进行结合,常可获得比单一分类器显著优越的泛化能力。集成学习需考虑的一个重要问题是如何产生具有差异性的个体分类器。根据个体分类器的生成方式来看,目前的集成学习方法可分为个体分类器必须序列生成的方法(代表是 Boosting 方法)和可同时生成的并行化方法(代表是 Bagging 和随机森林)[1-3]。此外,集成学习还需考虑另一个重要问题,即如何组合分类结果,常见的结合策略有平均法、加权平均法、投票法和学习法[3]等。在上述集成学习方法中,分类器学习完成后,其投票权值就已确定,这样的投票规则没有考虑样本的局部差异,当一个输入样本很难分类时,多数投票可能给出错误的预测。也就是说有些分类器在一些样本的局部区域有较高的分类准确率,但超出这个区域性能就变差,这样的分类器一般具有很小的权值;有些分类器在大多数区域分类正确,在小部分区域分类不准确,这样的分类器一般有较大的权值。对于较小的区域,采用投票法可能出现分类准确率较差的问题[4]。

针对以上问题,研究者们做了大量的工作,提出了许多的多分类器融合方法。如文献[5]针对如何提高分类器的泛化能力这一问题,提出了 Stacked Generalization 算法。Zhou 等[6]提出了 GASEN (Genetic Algorithm based Selective Ensemble) 算法,证明了使用较少的分类器集成可以得到相同甚至更优的集成效果。文献[7]针对不同的测试集使用不同的分类器的动态集成策略,实验证明使用多数投票策略的动态集成方法优于静态集成。文献[8]提出了多分类器动态融合方法,首先利用 AdaBoost 算法训练多分类器,然后根据当前的输入样本,基于待测样本局部分类精度动态选择分类器组合,对于一个待测样本,考察分类器在其有效邻域中的分类准确率,最后选择在其有效邻域中有较好分类性能的分类器进行集成。文献[9]是通过多数票(MV)或成对融合矩阵(PFM)将单个分类器的输出进行最优化组合,从而捕获来自不同分类器的共享和补充信息,最终实现准确和可靠的决策。文献[10]提出 M-Ensemble 学习优化模型用于分类,将 M-Ensemble 模型与多数投票法相结合,通过对基分类器的预测结果进行评估来确定其权值,并对基分类器的预测结果进行加权投票组合。文献[11]提出了基于线性规划的多分类器加权组合策略。文献[12]用分类器等价系数来衡量两个准确率不同的分类器在集成学习中的相关性,提出了一种新的分类器融合方法——平均分布集成策略。

但是上述研究方法未能考虑分类器的局部有效性[13-14],不能根据测试样本动态组合基分类器的分类结果。同时,即使采用了选择性集成仍然存在基分类器数量大、训练时间长、

本文受河北省自然科学基金(F2016202145)资助。

王延斌(1974—),男,硕士生,主要研究方向为机器学习;**武优西**(1974—),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为数据挖掘与机器学习;**刘洪普**(1977—),男,博士生,讲师,主要研究方向为机器学习。

测试效率低等问题[15-16]。分类器的局部有效性是指在多个基分类器中,各基分类器对同一样本的分类效果是不同的。这是由于各基分类器的归纳偏置和训练样本的权重所致。在训练各个基分类器时,均以最大可能拟合当前训练样本和训练样本权重,这使各基分类器有一个较适合的样本区域,同一分类器在样本空间不同的区域的分类效果会有差异。

如果能对不同的测试样本使用不同的基分类器组合进行预测,使与测试样本相关性好的基分类器组合被用于当前样本的预测中,则可提高集成效果。为了解决这一问题,本文采用不同的样本子集训练可针对局部样本的分类器,它在总体样本上的分类准确率不一定高,但在局部样本集上有较好的分类性能,然后将这些分类器组成层次化的树型结构。在预测时,根据上一层的预测结果选择一个合适的分支进行进一步的预测,以提高预测准确率。

我们将提出的方法应用于美国大学招生录取工作中。美国大学录取是一个典型的双向选择,即一个申请人通常同时会向多所大学提出申请,然后该申请人最终也会收到若干大学的录用通知书。这样对于美国高校来说存在一个问题,即每年有大量的满足录取条件的申请者提出申请,而这些申请者只有一小部分会最终选择该学校,若对每位申请者均发放录用通知书将会消耗大量人力、物力成本,如果仅对部分申请者发放录取通知书,则期望尽可能地从众多申请者中预先识别出哪些申请者可能会最终选择该高校。因此可以将其看成一个分类问题。分类类别有 3 个:Applicant(下文中标记为①类)、Decision(下文中标记为②类)、Enrolled(下文中标记为③类)。其中 Enrolled 类的样本是申请者最终选择了该高校,同时学校也发放了录取通知的样本类,它是需要重点识别的类,要求尽可能地不错分且不漏分此类别的样本。

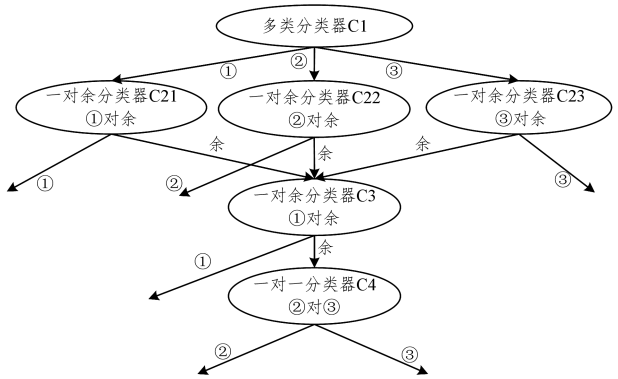
高校录用申请表中含有多达 54 种属性,有些属性难以用于该分类,如姓名、年龄等;有些属性则可以用于分类,如申请人所获奖项、估计学费成本、申请人年家庭收入总额等。我们在拥有美国某高校的近若干年申请者数据的情况下,依据该校历史录取情况来预测未来一年某一学生可能的录取情况。由于分类对象的特征太多,分类器学习算法受不相关或冗余特征的负面影响,使得分类精度不高[17-18]。实验表明,用一般常见的分类器,其准确率很不理想(略超过 60%);如单独使用决策树分类器,其准确率只有 51%左右,交叉验证剪枝后也只略超过 60%。如何有效地提高分类精度则成为一个亟待解决的问题。

2 集成学习算法

本文使用的集成学习算法是一个梯度优化的决策树结构,训练过程是对训练数据集不断切分。在决策树的每一层,为提高集成分类器的性能,对不同的基分类器使用上层划分开的不同数据子集进行训练。训练各层分类器时,对应不同节点分类器的分类目标不同,分别使用"一对多"策略或"一对一"策略。将多个分类器组成一个有向无环图,有向无环图的初始结点将对全部样本集进行分类,根据初始结点的分类结果将样本集分成几个子集,分别沿图向下,对这些样本子集在不同的下层结点进行进一步预测,直到最终确定样本的类别。将输入空间进行一步步的划分,可以使子分类器进行更有针对性的训练,从而提高分类性能。

2.1 多分类集成分类算法

本文采用图论中的有向无环图将多个基分类器组合成一个集成分类器。图 1 是对有 3 个类别的样本集进行分类的拓扑结构。对测试样本进行分类时不采用投票法,而是先将测试样本输入根结点分类器,根据根结点分类器的输出值判定走向哪一个分支子结点,再根据继续判定的值决定选择哪一个分支前进,直到走到叶子结点,此叶子结点表示该测试样本的类别。



注:①指分类器预测为①类的样本;②指分类器预测为②类的样本;③指分类器预测为③类的样本;余指二分类器预测为除其针对类的其余样本

图 1 本文多分类集成分类算法的模型

由于美国大学招生录取问题可以视为一个三分类问题,为此本文构造了如图 1 所示的三分类器的分类结构。其工作原理是:用一个多类分类器对全部待分类样本进行粗分类,从而将样本分为①,②和③ 3 类;由于粗分类精度有限,为了提高分类精度,对每类输出结果再采用二分类器进行细分类,如对分类为①的样本采用二分类器 C21 进行细分类,得到类①和其他类,对分类器 C22 和 C23 进行类似操作,分别得到了类②和③。由于此时剩余类依然包含①,②和③ 3 类样本,因此采用 C3 分类器分出类①和其他类,最后采用 C4 分类器得到类②和③。

因为较多的分类器结点会增加交叉验证等方法进行模型选择的时间,所以在实验中应尽可能降低集成结构的层数,使得在树比较浅的时候得到较好的集成效果。在训练各层分类器时采用贪婪策略:先将可以采用的分类器组成候选分类器组,用相同的训练样本集训练组内的分类器(组内各分类器使用的算法不同),用十折交叉验证评价训练得到的各个分类器的性能,选择其中最优的一个或几个分类器。这样做的目的是在树的深度较浅时就能得到符合预定期望的预测性能。树的深度较小意味着节省更多分类器算法的训练时间,模型选择次数较少。

误差-分歧分解表明:个体分类器准确性越高、多样性越大,则集成效果越好[3,19]。要获得好的集成需要两个条件:1)个体分类器应有一定的准确性;2)分类器之间具有差异性。在候选分类器组中,我们选用的分类器算法有:决策树、随机森林[20]、SVM、LCIELM[21-22]分类器。虽然本文使用了随机森林这类集成分类器,但是由于它作为基分类器在梯度优化决策树的整个集成框架中处于某一特定的位置,用于训练此基分类器的只是已划分开的某一个数据子集,所以其仍满足误差-分歧分解中提及的个体分类器差异。

2.2 算法设计

为了获得较好的集成分类性能,训练各层基分类器时,训练样本空间应尽可能与测试样本空间相同,因此在分层结构中,如果要训练某一层的基分类器,必须用上一层数据空间划分开的某一个子空间中的样本进行训练。CreateTrainData

算法在训练上一层分类器的同时生成下一层分类器的训练数据。在实验过程中将总训练样本划分为 10 等份,在 10 次迭代中的每一次选择其中 9 份样本训练一次基分类器,训练完成后用此分类器测试剩下的一份样本,根据测试结果的类别并入相应下一层训练数据集。具体步骤如算法 1 所示。

算法 1　CreateTrainData 算法

输入:训练样本集 D,分类器算法 C,分类数 k

输出:下一层训练数据 $D2_1,\cdots,D2_k$

1. 初始化 $D2_1,\cdots,D2_k$ 为空;
2. 划分全部训练样本 D 为 $D_1,\cdots,D_{10}$;
3. 对于每一个 i=1 to 10
   合并 $D_1,\cdots,D_{i-1},D_{i+1},\cdots,D_{10}$ 为 D_{train};
   用 D_{train} 训练分类器 C;
   用 C 预测 D_i,根据分类结果将 D_i 分为 $D_{i1},\cdots,D_{ik}$;
   将 $D_{i1},\cdots,D_{ik}$ 分别并入 $D2_1,\cdots,D2_k$。

多分类集成分类算法 ESHC(EnSemble Hierarchical Classifier)的训练阶段在每一层分类器的训练中,都在生成下一层的训练数据后,用本层的全部训练样本重新训练最终用于本层的基分类器,以充分利用训练样本集。由于随着层的下移,样本空间被分得越来越细,样本数变得很少,因此训练得到的基分类器的性能较差,这时可以将几个样本集合并。本文的算法在第三层合并了第二层的余类样本。测试阶段是在各层基分类器均训练完成后进行的,被测试的样本从根结点出发,根据所在层的分类器的预测结果选择下一层的分支,直到得出最终的判定类别。

算法 2　ESHC 算法

训练:

输入:训练样本集 D

输出:分类器集合{C1,C2,C3,C4}

1. $D2_1,\cdots,D2_3$=CreateTrainData(D,C1,3);
2. 用样本集 D 训练分类器 C1;
3. 初始化 $D3^i_1,\cdots,D3^i_k$(i=1,…,3,k=1,…,2)为空;
4. 对于每一个 i=1 to 3
   1)$D3^i_1,D3^i_2$=CreateTrainData(D,$C2_i$,2);//$C2_i$ 为一对余分类器,$D3^i_k$ 中下标 k=1 表示为①类,k=2 表示为余类。
   2)用样本集 $D2_i$ 训练分类器 $C2_i$;
5. 合并 $D3^i_2$(i=1,…,3)到 D3;
6. $D4_1,D4_2$=CreateTrainData(D3,C3,2);
7. 用样本集 D3 训练分类器 C3;
8. 用样本集 $D4_2$ 训练分类器 C4;

预测:

输入:测试样本集 D_{test},分类器集合{C1,C2,C3,C4}

输出:各预测样本集 $D^1_{test},D^2_{test},D^3_{test}$

1. 初始化 D^i_{test}(i=1,…,3)为空;
2. 用 C1 对 D_{test} 分类得到 $D2^1_{test},D2^2_{test},D2^3_{test}$;
3. 初始化 $D3_{test}$ 为空;
4. 对于每一个 i=1 to 3
   1)用 $C2_i$ 对 $D2^i_{test}$ 分类得到 $D3^1_{test},D3^2_{test}$;
   2)$D3^1_{test}$ 合并入 D^i_{test},$D3^2_{test}$ 合并入 $D3_{test}$;
5. 用 C3 对 $D3_{test}$ 分类得到 $D4^1_{test},D4^2_{test}$;
6. $D4^1_{test}$ 合并入 D^1_{test};
7. 用 C4 对 $D4^2_{test}$ 分类得到 $D5^1_{test},D5^2_{test}$;
8. $D5^1_{test}$ 合并入 D^2_{test},$D5^2_{test}$ 合并入 D^3_{test}。

ESHC 算法对各层基分类器使用不同的数据子集进行训练,而且越向下层,数据子集被分得越小,训练也越针对于某些特殊样本。同一层的不同基分类器使用的训练数据样本也互不相同,它们的分类有效区域是不同且互补的。对于测试样本如果在前几层已经有较高概率(如前几层的预测均为某一相同的类)将样本归于某一类,这个样本将被预测为这个类别,不再进入下一层预测。只有前层的预测结果不能以高概率预测为某一类的样本才沿图向下进行再预测。因此叫作梯度优化。

2.3　评价指标

为了对集成学习框架的泛化性能进行评估,本文使用了以下评价指标。

(1)正确率

正确率是分类任务中常用的性能指标,它是分类正确的样本数占样本总数的比例。对于样本集 D,分类器 f,分类正确率定义为:

$$acc(f;D)=\frac{1}{m}\sum_{i=1}^{m}I(f(x_i)=y_i) \tag{1}$$

(2)精确率、召回率和 $F1$

对于二分类问题,可将样本根据其真实类别与预测类别划分为真正例(True Positive)、假正例(False Positive)、真反例(True Negative)、假反例(False Negative) 4 种情况,分别用 TP,FP,TN,FN 表示。

精确率 P 与召回率 R 分别定义为:

$$P=\frac{TP}{TP+FP} \tag{2}$$

$$R=\frac{TP}{TP+FN} \tag{3}$$

一般来说,精确率较高时,召回率往往偏低;而召回率较高时,精确率往往偏低。$F1$ 分数(F1 Score)是统计学中用来衡量二分类模型精确度的一种指标,它同时兼顾了分类模型的精确率和召回率,可以看作是模型精确率和召回率的加权平均,其最大值为 1,最小值为 0。在多分类问题中,$F1$ 能反映每一个类别预测性能的变化,其定义为:

$$F1=2\times\frac{precision\times recall}{precision+recall} \tag{4}$$

在计算多分类数据时,对于每一个类别,将其他类别视为它的负类,计算每一类别的 $F1$ 值,再对多个值求平均。

2.4　ESHC 算法的时间复杂度

ESHC 算法运行时主要有两部分的时间消耗:生成训练数据的时间消耗和训练预测的时间消耗。每一种时间消耗都受两个因素影响:1)所选分类器的时间性能;2)数据样本集的大小 N。假定训练各层基分类器的时间复杂度为 $O(N)$,则训练预测的时间复杂度为 $H\cdot O(N)$,因为从第二层开始后续各层将整个样本集分成了一个个小的子集,用于各层的样本数逐渐减少,因此 H 为一个不太大的常数($2<H<4$,假设集成分类器的层数为 4)。由于生成各层训练数据时进行了 10 次迭代,实际上是在训练样本集上完成了 10 次训练预测的过程,因此时间复杂度为 $H'\cdot O(N)$,$H'=10H$。

在经典的集成学习算法中,AdaBoost 算法和 Bagging 算法的计算复杂度均为 $T\cdot(O(N)+O(s))$,其中,采样与投票过程的复杂度 $O(s)$ 很小,T 为训练的轮数,在两个算法中会有所不同,但其通常是一个不太大的常数。可以看出,ESHC 算法与使用的基学习算法的计算复杂度同阶,与经典的 AdaBoost 算法和 Bagging 算法的计算复杂度也同阶,因此它是一个高效的集成学习算法。

3 实验评价

本文实验均在同一台电脑上完成(CoreI5 双核 2.70 GHz,4 GB 内存),实验环境为 MATLAB 2012a。为了测试本文算法的性能,选择了美国某高校录取申请表真实数据集 Enroll,该数据集共 54
列,47932 个样本,去除不相关和重复的列后,剩下 40 列,最后一列为类标签。此外,为了进一步测试本文方法的性能,我们还采用了 UCI 数据集[1)] 中的 3 个公共数据集 Abalone,Adult 和 Yeast 进行测试。这些数据集的特征如表 1 所列。

表 1 数据集属性

数据集	样本总个数	属性个数	各类别样本个数	样本比率
Enroll	47932	39	20515/19345/8072	4/4/2
Abalone	4177	8	1407/1323/1447	3/3/4
Adult	45222	14	34014/11208	7/3
Yeast	1484	8	463/429/592	4/4/6

在实验中选择 Matlab 工具箱中的 Classregtree,LCIELM 和 LibSVM 作为基分类器,并将它们单独分类的结果作为参考,此外我们还选择了 Matlab 工具箱中的集成分类器 TreeBagger 算法与本文 ESHC 集成框架进行对比。

所有实验数据是从总样本中随机取 7/10 作为训练数据,3/10 作为测试数据。实验中,用全部训练样本训练多个第一层分类器,比较其 $F1$ 值,最终选择 TreeBagger 分类器作为第一层。根据第一层分类结果将训练样本分为 3 部分,分别用于训练第二层分类器。左分支将①类样本看作一类,其余样本看作另一类,其他分支分别将②,③看成一类,其余样本看作另一类。根据模型选择最终采用 LibSVM 作为第二层分类器。在后续层的模型选择中将③类样本的召回率看作重点。

图 2 和图 3 分别给出了不同方法的训练时间和测试时间,由于 LCIELM 算法的训练时间较长,为此我们在图 2 和图 3 中忽略该算法。

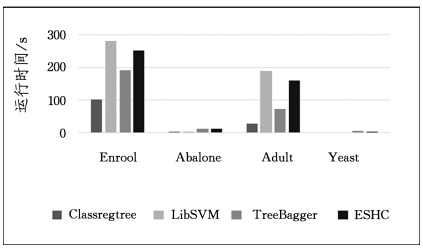


图 2 训练时间对比

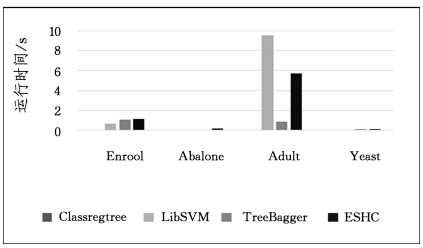


图 3 测试时间对比

表 2 给出了梯度优化分类器在 Enroll 数据集的分类结果中各类别的精确率、召回率和 $F1$ 值,其中③表示对 Enrolled 类的区分效果,为取得较好的召回率,损失了部分精确率。

表 2 ESHC 分类效果(Enroll 数据集)

	精确率	召回率	F1 值
①	0.9130	0.7441	0.8200
②	0.6546	0.5925	0.6220
③	0.6164	0.7291	0.6681
平均值	0.7280	0.6886	0.7033

为了进一步说明本文算法的性能,表 3－表 6 分别给出了本文方法与其他 4 种方法在 Enroll,Abalone,Adult 和 Yeast 4 种数据集上的 $F1$ 值及正确率的对比。

表 3 分类结果对比表(Enroll 数据集)

分类器	$F1$ 值	正确率
Classregtree	0.6743	0.6522
LCIELM	0.6368	0.6159
LibSVM	0.6479	0.6269
TreeBagger	0.6831	0.6528
ESHC	**0.7033**	**0.6732**

表 4 分类结果对比表(Abalone 数据集)

分类器	F1 值	正确率
Classregtree	0.6472	0.6551
LCIELM	0.6610	0.6635
LibSVM	0.6114	0.6191
TreeBagger	0.6329	0.6383
ESHC	**0.6692**	**0.6738**

表 5 分类结果对比表(Adult 数据集)

分类器	F1 值	正确率
Classregtree	0.7918	0.8564
LCIELM	0.7336	0.8070
LibSVM	0.7590	0.8385
TreeBagger	0.7892	0.8525
ESHC	**0.7933**	**0.8614**

表 6 分类结果对比表(Yeast 数据集)

分类器	F1 值	正确率
Classregtree	0.6633	0.6426
LCIELM	0.6655	0.6433
LibSVM	0.6363	0.6079
TreeBagger	0.6835	0.6620
ESHC	**0.6862**	**0.6629**

对比表 3－表 6 中的评价指标值可知:

(1)分层结构的梯度优化决策树方法可以提高分类性能,例如在表 3 Enroll 数据集上,本文方法的 $F1$ 及正确率分别为 0.7033 和 0.6732,较其他方法均有一定程度的提高;此外,其他 3 种数据集也呈现了这一特点,从而验证了本文方法的性能。

(2)除本文提出的 ESHC 算法外,TreeBagger 算法相比其余 4 类算法在 4 个数据集上总体表现出较好的分类性能,并在 Enroll 和 Yeast 2 个数据集上表现出次优的分类性能(在 Enroll 数据集的 $F1$ 值为 0.6831,正确率为 0.6528;在 Yeast 数据集的 $F1$ 值为 0.6835,正确率为 0.6620)。此外,Classregtree 分类器在 Adult 数据集($F1$ 值为 0.7918,正确率为0.8560)、LCIELM 分类器在 Abalone 数据集($F1$ 值为 0.6610,正确率为 0.6635)上的分类性能次优。这是因为 TreeBagger 算法是一种集成学习算法。表 3－表 6 的结果表明 ESHC 算法在 4 个数据集上均取得了最好的正确率和 $F1$ 值,其泛化能力比 TreeBagger 算法好。

1) http://archive.ics.uci.edu/ml/datasets.html

(3)实验表明,ESHC 算法更适合大量样本且复杂分布的数据集分类。将表 3、表 5 与表 4、表 6 进行对比发现 Enroll 数据集和 Adult 数据集上 ESHC 算法的性能改善比在其余两个数据集上的性能改善较大。从数据集样本数量来看,Enroll 数据集和 Adult 数据集的样本数比其余两个数据集的样本数大很多。如 Enroll 数据集和 Adult 数据集都有 40000 多个样本,而 Abalone 数据集和 Yeast 数据集分别只有 4177 个样本和 1484 个样本。

(4)一般来说,集成分类器比单一分类器的时间消耗大,这是因为集成分类器需要训练多个分类器,测试的时间比较多。但图 2 显示 ESHC 算法的训练时间在 Abalone 数据集上才是最高的;图 3 显示 ESHC 算法的测试时间在 4 个数据集上均不是最高的,这可以说明 ESHC 算法的训练时间和测试时间并没有太大增加。这是因为该算法从第二层开始并不是用整个训练数据集来训练基分类器,而是各基分类器在它对应的训练子集上进行训练,减少了时间消耗。

结束语　目前,集成学习在各领域被成功应用,因此集成学习的理论和算法的研究成为了机器学习领域的一个热点,越来越多的集成学习算法被应用于分类中。在考虑如何提高集成分类器的性能时,应充分认识分类器的局部有效性,利用合理的结构和算法避免其对分类的不利影响。本文采用梯度优化的思想,分层训练各个子分类器,利用层次化结构对误分类的样本进一步学习和预测,使分类性能得到提高。

在梯度优化算法的训练阶段,如何选择训练样本是一个重点,如果选择的训练样本与预测时进入到该分支的样本分布不一致,将会严重影响分类性能。该算法的另一个缺点是误差传递,如果在某一结点处发生了错误判别,这个错误将向下传递。这些问题将是进一步研究的重点。

参 考 文 献

[1] DIETTERICH T G. Machine learning research four current directions[J]. AI Magazine,1997,18(4):97-136.

[2] 唐伟,周志华. 基于 Bagging 的选择性聚类集成[J]. 软件学报,2005,16(4):496-502.

[3] 周志华. 机器学习[M]. 北京:清华大学出版社,2016.

[4] 唐春生,金以慧. 基于全信息矩阵的多分类器集成方法[J]. 软件学报,2003,14(6):1103-1109.

[5] WOLPERT D H. Stacked generalization[J]. Neural Networks,1992,5(2):241-259.

[6] ZHOU Z H,WU J X,TANG W. Ensembling neural networks:many could be better than all[J]. Artificial Intelligence,2002,137(1):239-263.

[7] KO A R,SABOURIN R,BRITTO A S. From dynamic classifier selection to dynamic ensemble selection[J]. Pattern Recognition,2008,41(5):1718-1731.

[8] 方敏. 集成学习的多分类器动态融合方法研究[J]. 系统工程与电子技术,2006,28(11):1759-1762.

[9] MITCHELL H B. Ensemble learning in data fusion:Concepts and ideas[M]. Springer Berlin Heidelberg,2012.

[10] ROJARATH A,SONGPAN W,PONG-INWONG C. Improved ensemble learning for classification techniques based on majority voting[C]// IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE,2017:107-110.

[11] ZHANG L,ZHOU W. Sparse ensembles using weighted combination methods based on linear programming[J]. Pattern Recognition,2011,44(1):97-106.

[12] 朱波,陈科,徐君,等. 平均分布集成策略:一种新的分类器融合方法[J]. 小型微型计算机系统,2016,37(7):1546-1550.

[13] YU Z W,WANG D X,JANE Y,et al. Progressive subspace ensemble learning[J]. Pattern Recognition,2016,60(C):692-705.

[14] DUTTA A,DASGUPTA P. Ensemble learning with weak classifiers for fast and reliable unknown terrain classification using mobile robots[J]. IEEE Transactions on Systems,Man,and Cybernetics:Systems,2017,47(11):2933-2944.

[15] TOLOMEI G,SILVESTRI F,HAINES A,et al. Interpretable Predictions of Tree-based Ensembles via Actionable Feature Tweaking[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM,2017:465-474.

[16] 高锋,黄海燕. 基于邻域混合抽样和动态集成的不平衡数据分类方法[J]. 计算机科学,2017,44(8):225-229.

[17] LUO H Y,WANG D Y,YUE C Q,et al. Research and application of a novel hybrid decomposition-ensemble learning paradigm with errorcorrection for daily PM10 forecasting[J]. Atmospheric Research,2018,201:34-45.

[18] ZHANG L,SHAH S K,KAKADIARIS I A. Hierarchical multi-label classification using fully associative ensemble learning[J]. Pattern Recognition,2017,70:89-103.

[19] 张春霞,张讲社. 选择性集成学习算法综述[J]. 计算机学报,2011,34(8):1399-1410.

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

[21] WU Y,LIU D,JIANG H. Length-changeable incremental extreme learning machine[J]. Journal of Computer Science and Technology,2017,32(3):630-643.

[22] LIU D,WU Y,JIANG H. FP-ELM:An online sequential learning algorithm for dealing with concept drift[J]. Neurocomputing,2016,207(26):322-334.

(上接第 104 页)

[8] USUNIER N,SYNNAEVE G,LIN Z,et al. Episodic Exploration for Deep Deterministic Policies:An Application to StarCraft Micromanagement Tasks[J]. arXiv:1609.02993,2016.

[9] CHURCHILL D,BURO M. Portfolio greedy search and simulation for large-scale combat in starcraft[C]// Computational Intelligence in Games. IEEE,2013:1-8.

[10] JUSTESEN N,RISI S. Continual online evolutionary planning for in-game build order adaptation in StarCraft[C]// the Genetic and Evolutionary Computation Conference. 2017:187-194.

[11] Holland J H. Adaptation in natural and artificial systems[M]. MIT Press,1992.

[12] 沐阿华,周绍磊,于晓丽. 一种快速自适应遗传算法及其仿真研究[J]. 系统仿真学报,2004,16(1):122-125.

[13] 姜昌华. 遗传算法在物流系统优化中的应用研究[D]. 中山:华东师范大学,2007.

[14] SRINIVAS M,PATNAIKL M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Transactions on Systems Man & Cybernetics,2002,24(4):656-667.

[15] KOVARSKY A,BURO M. Heuristic Search Applied to Abstract Combat Games[M]// Advances in Artificial Intelligence. Berlin:Springer,2005:66-78.