融合 DECORATE 的异构分类器集成算法*

韦艳艳[1], 李陶深[2], 刘美玲[1]
(1. 广西民族大学 信息科学与工程学院 广西混杂计算与集成电路设计分析重点实验室, 南宁 530006; 2. 广西大学 计算机与电子信息学院, 南宁 530004)

摘　要: 在基于 Stacking 框架下异构分类器集成方式分析的基础上,引入同构分类器集成中改变训练样本以增强成员分类器间差异性的思想,提出融合 DECORATE 的异构分类器集成算法 SDE;在 1-层泛化利用 DECORATE 算法,向 1-层训练集增加一定比例的人工数据,使得生成的多个 1-层成员分类器间具有差异性。实验表明,该方法在分类精度上要优于传统 Stacking 方法。

关键词: 分类器集成; 异构; Stacking; DECORATE; 差异性
中图分类号: TP181;TP301.6　　**文献标志码**: A　　**文章编号**: 1001-3695(2012)11-4134-03
doi:10.3969/j.issn.1001-3695.2012.11.034

Algorithm of heterogeneous classifiers ensembles based on DECORATE

WEI Yan-yan[1], LI Tao-shen[2], LIU Mei-ling[1]
(1. *Guangxi Key Laboratory of Hybrid Computation & IC Design Analysis*, *School of Information Science & Engineering*, *Guangxi University for Nationalities*, *Nanning* 530006, *China*; 2. *School of Computer & Electronic Science*, *Guangxi University*, *Nanning* 530004, *China*)

Abstract: Based on the Stacking framework to construct heterogeneous ensembles, this paper introduced manipulating training samples in the context of creating homogeneous ensembles as the mechanism to encourage diversity. It proposed a new algorithm SDE, which used DECORATE to generate level-1 ensembles by adding proportion of artificial data to level-1 training set so as to inject diversity for member classifiers in level-1. Experiment results indicate that the proposed method achieves better performance than classic Stacking.

Key words: classifier ensembles; heterogeneous; Stacking; DECORATE; diversity

0　引言

在分类学习中,给定分类目标 C 的训练样本集合 X,构造分类器的目的就是通过对训练集样本数据的学习,尽可能根据事先未知的、隐藏于数据中的重要信息,直接或间接地拟合或逼近目标分类函数。而对分类器进行集成则是通过构造多个同构或异构分类器来解决同一个问题。研究表明,集成分类器能较好地克服假设空间表达能力不足等问题,通常可以获得比单一分类器更好的泛化效果[1]。目前,集成学习在模式识别、信息融合、Web 信息过滤、信息检索系统以及数据挖掘等诸多领域得到了成功应用。

对于集成分类器算法而言,成员分类器之间的随机差异性有助于提高集成精度,通过直接或间接的方式来增强成员分类器之间的差异性是集成有效的重要因素[2]。对于同构分类器的集成,增强成员分类器间的随机差异性可采用改变学习参数、训练数据或训练集特征等方式来实现[3]。而对于异构分类器的集成,由于成员分类器是利用不同学习算法训练而得,其结构及分类偏置已然有差别,成员分类器间的差异程度要高于同构集成分类器[4,5]。

在异构分类器集成方式中,Stacking[6] 因其易于拓展的层次式计算框架而受到关注,其基本原理是根据成员分类器的输出结果进行二次或多次泛化得到集成分类器。作为原始数据输入的是 0-层(0-level)分类器,每个分类器输出一个结果。后继层接收前一层的输出作为它的输入,其输出再传送到下一层,位于最顶层(meta-level)的分类器输出最终的结果。

已有的相关研究中,在 Stacking 集成方法中引入随机差异性的主要方式包括利用不同类型的学习算法训练成员分类器、调整 1-层特征向量的组成、调整每层分类器的数量或逐层增加新的训练样本等。如 Seewald[7] 提出的 Stacking$C$ 用成员分类器对样本是否预测为某个类 C_i(标记为 1 或 0)作为训练类 i 的 1-层特征集标签; Hatami 等人[8] 将 Boosting 算法引入 0-层泛化,以提高 0-层分类器的精度,从而改善了 1-层泛化的效果;Menahem 等人[9] 提出了三层 Stacking 的计算模式 Troika,其中 1-层训练生成 $K\times(K-1)$ 个分类器(K 为问题域的类别数),而在 2-层训练生成 K 个分类器,以此得到最高层的超级分类器;Ledzma 等人[10] 提出的 GA-Stacking,引入用遗传算法搜索出最佳的配置参数(如选择何种元学习器、元学习器的参数、基本学习器的数量、用哪些基本学习器,或基本学习器的参数等),以提高 Stacking 的泛化性能。

在基于 Stacking 架构的计算模式中,大多数研究着眼于如何利用 0-层分类器的各种不同输出信息组成 1-层特征向量,目的是尽可能地将原始训练集的数据特征反映到 1-层训练样

收稿日期: 2012-04-14; **修回日期**: 2012-05-21　　**基金项目**: 广西自然科学基金资助项目(2010GXNSFA013127); 广西教育厅资助项目(201106LX131)

作者简介:韦艳艳(1974-),女(壮族),广西贵港人,讲师,硕士,主要研究方向为机器学习、数据挖掘(ybwyy2006@yahoo.com.cn);李陶深(1957-),男,教授,博士,主要研究方向为分布式数据库、网络计算;刘美玲(1979-),女,讲师,硕士,主要研究方向为数据挖掘.

本中,用于指导第二阶段的学习,从而提高 1-层分类器的分类准确率。相比较而言,根据 1-层特征向量产生多个有差异的 1-层分类器的方式则较为少见。而实际上,0-层分类器对原始训练集的描述与真实的数据特征之间是有差异的,因而 1-层训练样本中有关 0-层分类器的预测结果与实际类之间的关系描述也与实际有偏差。在不改变 0-层输出的预测信息的前提下,提高 1-层分类精度的重要途径之一就是利用 1-层训练样本集。为此,本文提出了一种融合 DECORATE[11] 的异构分类器集成算法。

1　Stacking 集成框架

给定 T 个样本的数据集 $D=\{(x_t,y_t),t=1,\cdots,T\}$,其中 x 为样本的特征向量,y 为样本类标签。以 j-折交叉验证(j-folds cross validation)方式得到 j 折测试集和训练集,分别表示为 D_j 和 $D_{(-j)}$。给定 N 个分类学习算法作为第 0-层归纳算法,对第 j 折训练集 $D_{(-j)}$ 用第 n 个算法归纳得到 0-层分类器 $f_{nj}(x)$,$n=1,\cdots,N$。而对第 j 折测试集 D_j 中的每一个样本,分类器 $f_{nj}(x_t)$ 对它的预测表示为 x_t'。交叉验证过程结束后,将 N 个分类器的输出组成一个新的样本集 $D'=\{(x_t',y_t),t=1,\cdots,T\}$。对样本集 D' 进行泛化,得到 1-层分类器 $F(x')$,它描述了 0-层分类器的输出与实际类标签之间的关系,如图 1 所示。

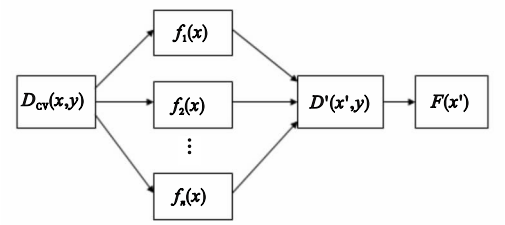

图1　2层结构的Stacking框架示意图

2　DECORATE 算法

DECORATE 是一种对同构分类器进行集成的算法。其基本思路是反复向训练集加入人工样本,确保增加新成员分类器之后不降低集成分类器的准确率,具体处理过程如下:

a)用初始样本集生成第一个成员分类器。

b)向样本集加入随机生成的人工样本数据,数量与原训练样本一致。生成方法是:

(a)对于连续型特征,通过计算其均值及标准差得到正态分布的随机特征值;

(b)对于离散型特征,根据不同值在样本集中出现的频率进行取值;

(c)对于人工样本的类标签,用当前的集成分类器对每一个人工样本进行分类,取其相反的分类结果作为人工样本的类标签值,从而得到差异性数据。

c)把人工样本加入原始样本集,生成新的成员分类器,并将其加入到集成分类器中。

d)测试集成分类器的性能,若错误率比加入之前要高,则将刚加入的成员分类器删除,否则将其保留。

e)从样本集中删除人工数据。重复上述过程,直到获得指定的成员分类器数量为止。

f)集成分类器根据成员分类器输出的最大概率均值得到分类结果。

由此可见,DECORATE 在生成人工数据时的着眼点在于力求使这些随机样本与原训练样本有差异,从而使得成员分类器在预测偏好、分类边界等方面存在不同程度的差别,以达到降低集成分类器误差的目的。

3　融合 DECORATE 的异构分类器集成方法 SDE

3.1　基本思想

SDE 算法采用两层的 Stacking 框架,其对异构分类器进行集成的思路是:用训练集 $D(x,y)$ 以交叉验证方式得到 N 个异构分类器,令其作为 0-层分类器,而后用这些分类器测试 $D(x,y)$ 中的样本,将预测结果与实际类标签一起组成 1-层训练集 $D(x',y)$,对 $D(x',y)$ 应用 DECORATE 算法得到 K 个 1-层分类器。分类时计算 K 个 1-层分类器对样本预测的最大概率均值,以确定待分类样本的类别。

3.2　算法描述

输入:a)N 个基本学习算法 $L_1,L_2,\cdots,L_N$,1-层学习算法 L_M;b)训练集 $D(x,y)=\{(x_1,y_1),\cdots,(x_t,y_t)\}$,样本数量为 T,其中 $y\in\Omega,\Omega=\{\omega_1,\cdots,\omega_f\}$;c)1-层训练集 $D'=\varnothing$;d)1-层分类器数量 K;e)人工样本比例 P,$0<P\leqslant 1$。

输出:集成分类器 F_e。

for $j=1$ to 10 do
　以交叉验证方式得到第 j 折测试集及训练集 D_j 和 $D_{(-j)}$;
　for $i=1$ to N do
　　由 L_i 及 $D_{(-j)}$ 生成 0-层分类器 f_i;
　　f_i 对 D_j 中进行预测得到分类结果 D_j',$D'=D'\cup D_j'$;
　end for
end for
由 L_M 及 D' 生成 1-层成员分类器 F_0,计算其训练误差 E_0;
$E=E_0$;
$F_e=F_0$;
$i=1$;
while($i<K$)
{　根据 D' 生成包含 $P\times|T|$ 个不带类标签的人工样本;
　由 $F_e(x)$ 为每个人工样本指定其类标签得到 D_i';
　$D'=D'\cup D_i'$;
　由 L_M 及 D' 生成 1-层成员分类器 $F_i(x)$;
　$F_e=F_e\cup\{F_i\}$;
　计算 F_e 的训练误差 E';
　若 $E'<E$
　　$i=i+1$;
　　$E=E'$;
　否则
　　$F_e=F_e-\{F_i\}$;
}

对于待检样本 x':

$$F_e(x')=\underset{y\in\Omega}{\operatorname{argmax}}\sum_{i=1}^{K}I(F_i(x)=y) \tag{1}$$

由成员分类器 F_i 计算出 x' 属于某个类别 ω_j 的概率均值,均值最大的即为 x' 所属的类别。

4　实验及分析

4.1　数据集

本文采用 UCI[12] 的 18 个标准数据集进行 SDE 算法与 Stacking(STA)的对比实验。数据集的描述信息如表 1 所示。

表 1　数据集的基本信息

数据集	样本数	特征数	离散型特征	连续型特征	类数
anneal	898	38	32	6	6
auto-mpg	399	7	2	5	4
breast-c	286	9	9	0	2
breast-w	699	9	0	9	2
colic	368	22	15	7	2
credit-a	695	15	9	6	2
credit-g	1 000	20	13	7	2
diabetes	768	8	0	8	2
heart-c	303	13	7	6	5
heart-s	270	13	0	13	2
hepatitis	155	19	13	6	2
iris	153	4	0	4	3
labor	57	16	8	8	2
lymph	148	18	15	3	4
segment	2 310	19	0	19	7
soybean	687	35	35	0	19
vote	435	16	16	0	2
zoo	104	17	16	1	7

4.2　实验过程设置

所有测试都在 WEKA[13] 平台上进行。本文选择了 WEKA 提供的七种算法用于构造 0-层分类器，分别为 Logistic、Random Forest、C4.5、IB1、Jrip、NaïveBayes、OneR。根据这些算法在 18 个数据集上的测试结果，按几何平均错误率（表 2）由小到大分成六组进行实验。

六组异构分类器分别为

a) BC2：NaïveBayes、OneR；

b) BC3：Jrip、NaïveBayes、OneR；

c) BC4：IB1、Jrip、NaïveBayes、OneR；

d) BC5：C4.5、IB1、Jrip、NaïveBayes、OneR；

e) BC6：Random Forest、C4.5、IB1、Jrip、OneR、NaïveBayes；

f) BC7：Logistic、Random Forest、C4.5、IB1、Jrip、NaïveBayes、OneR。

表 2　基本学习算法对比 OneR 的几何平均错误率

	NaïveBayes	IB1	Jrip	Logistic	Random Forest	C4.5
几何平均错误率	1.588	1.853	1.754	2.120	2.053	1.861

1-层泛化时均采用 C4.5 算法，集成分类器的成员个数设定为 10，所有分类器用 10 次 10-折交叉验证来计算其分类准确率，人工样本数量与原始训练数据之比 $P \in [0.1, 1]$，基本学习算法的参数使用 WEKA 的默认设置。

4.3　结果分析

图 2 为人工样本比例 $P = 1$ 时的结果。可以看到，在六组对比测试中，有五组显示出 SDE 算法要优于 STA 算法，而 BC5 这一组 STA 略优于 SDE，但两者非常接近。这表明，在 1-层泛化时，根据训练特征集加入不同的人工数据，以使生成的多个成员分类器间具有差异度，再按其输出结果进行平均集成，有助于提高集成分类器的准确率。

图 3 给出了 1-层的成员分类器个数 K 从 2 ~ 10 的测试结果，可以看到，BC7 的集成效果最佳，BC6 居其次，而 BC2 则排在最后。分析可知，0-层的异构分类器数量越多，则 1-层特征数量就越多，有助于学习算法更好地逼近真实的分类函数，故精度也会相应获得提高。而 1-层成员分类器数量的增加，有助于整个集成分类器性能的提升。

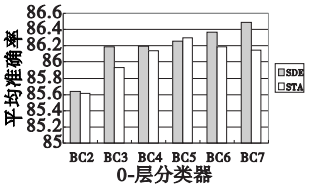

图2　SDE与STA算法的对比

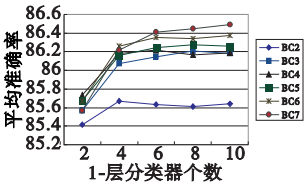

图3　1-层成员分类器数量与泛化精度间的关系

表 3 给出了人工样本比例 P 由 0.1 增加到 1 时的实验结果。表中带下划线的数字表示该组异构分类器在对应 P 值时准确率最高。可以看到，六组的最好结果均出现在比例为 0.6 或以下，最高平均值出现在 $P = 0.3$。由此可见，P 值并非越大越好，较小的 P 值反而有助于加快构造分类器，减少计算时间。

表 3　不同人工样本比例的测试结果

P 值	BC2	BC3	BC4	BC5	BC6	BC7	平均值
0.1	85.835	<u>86.231</u>	86.361	86.363	86.291	86.533	86.269
0.2	<u>85.976</u>	86.171	86.366	86.387	86.252	86.382	86.255
0.3	85.777	86.221	<u>86.395</u>	86.217	86.365	86.660	<u>86.272</u>
0.4	85.793	86.128	86.161	<u>86.393</u>	86.233	<u>86.661</u>	86.228
0.5	85.844	85.933	86.142	86.286	86.344	86.657	86.201
0.6	85.922	86.027	86.242	86.226	<u>86.457</u>	86.446	86.220
0.7	85.866	85.958	86.076	86.358	86.220	86.619	86.183
0.8	85.693	86.047	86.189	86.362	86.229	86.471	86.165
0.9	85.838	86.126	86.302	86.264	86.324	86.566	86.237
1.0	85.639	86.184	86.192	86.261	86.373	86.489	86.190

5　结束语

本文提出了融合 DECORATE 的异构分类器集成算法 SDE。该算法首先在 0-层生成多个异构分类器，而后由这些分类器对样本集进行预测，得到初始 1-层训练样本，再将随机生成的人工样本加入到 1-层训练集。由于人工样本的类标签与集成分类器判决结果是相反的，可以最大程度地将差异性数据引入训练集，从而使得 1-层成员分类器相互间具有差异性。多组实验结果表明，SDE 算法在分类精度上要优于传统的 STA 算法。同时通过实验也发现，人工样本的生成比例为原训练样本数量的 60% 或者更低，同样也能取得较理想的泛化性能，而且还具有较快的计算速度。下一步工作将分析研究 1-层成员分类器的数量与随机人工样本比例之间的关系。

参考文献：

[1] DIETTERICH T G. Ensemble methods in machine learning[C]// Lecture Notes in Computer Science, vol 1857. Berlin: Springer, 2000: 1-15.

[2] BROWN G. Ensemble learning[C]//Proc of Encyclopedia of Machine Learning. Berlin: Springer, 2010: 312-320.

（下转第 4147 页）

的结果。

例 2　考虑含有时变结构不确定 Lurie 系统[10,11],其系数矩阵为

$$A=\begin{bmatrix}-2 & 0\\ 0 & -0.9\end{bmatrix},B=\begin{bmatrix}-1 & 0\\ -1 & -1\end{bmatrix},D=\begin{bmatrix}-0.2\\ -0.3\end{bmatrix}$$

$$M=[0.3\quad 0.1],N=[0.1\quad 0.2],H=0.1\times\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$E_1=E_2=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},K_1=0.2,K_2=0.5$$

对于不同的 μ,表 2 分别列出了文献[10,11]和定理 3 获得的保证系统绝对稳定的时滞上界 h。

表 2　对于不同的 μ,保证系统绝对稳定的时滞上界 h

μ	0.00	0.30	0.60	0.90	≥1.00
文献[10]	3.305 7	2.078 7	1.419 5	0.922 8	0.763 8
文献[11]	4.107 7	2.370 7	1.481 9	1.034 6	1.034 6
定理 3	5.745 0	3.325 0	2.186 7	1.358 8	1.093 7

从表 2 中可以看出,定理 3 比文献[10,11]中的相关结论具有更弱保守性。

4　结束语

本文考虑了一类 Lurie 时变时滞系统的绝对稳定问题。通过构造适当的 Lyapunov 函数和引入一些自由权矩阵,给出了系统时滞相关绝对稳定性判据。最后的两个数值例子(表 1、2)表明,所得结论改进了现有文献的结论。本文方法还可推广到 Markov 跳变神经网络系统、随机系统等稳定性问题。

参考文献:

[1] BLOMAN P A. Extension of popov absolute stability criterion to non-autonomous systems with delays[J]. International Journal of Control,2000,73(15):1349-1361.

[2] GUAN Zuo-xin, GE Wei-gao. Lyapunov functional for multiple delay general Lurie systems with multiple non-linearites[J]. Journal of Mahematics Analysis and Applications, 2001, 259 (15): 596-608.

[3] YU Li, HAN Qing-long, YU Shi-ming, *et al*. Delay-dependent conditions for robust absolute stability of uncertain time-delay systems [C]//Proc of the 42nd IEEE Conference on Decision and Control. 2003:6033-6037.

[4] HAN Qing-long. Absolute stability of time-delay systems with sector-bounded nonlinearity[J]. Automatica,2005,41(12):2171-2176.

[5] XU S, FENG G. Improved robust absolute stability criteria for uncertain time-delay systems [J]. IET Control Theory Applications, 2007,1(6):854-859.

[6] LEE S M, PARKS J H. Delay-dependent criteria for absolute stability of uncertain time-delayed [J]. Journal of the Franklin Institute, 2010,347(1):146-153.

[7] WU Min, FENG Zhi-yong, HE Yong. Improved delay-depended absolute stability of Lur'e systems with time-delay[J]. International Journal of Control Automation and Systems,2009,7(6):1009-1014.

[8] KRISHNANL R, GOSHAIDAS R. Improved stability criteria for Lurie type systems with time-varying delay[J]. Acta Automatica Sinica, 2011,5(37):636-644.

[9] QIU Fang, ZHANG Quan-xin. Absolute stability analysis of Lurie control system with multiple delays: an integral-equality approach [J]. Nonlinear Analysis: Real World Applications,2011,12(3): 1475-1484.

[10] HAN Qing-long, YUE Dong. Absolute stability of Lur'e systems with time-varying delay[J]. IET Control of Theroy Applications,2007,1 (3):854-859.

[11] GAO Jin-feng, PAN Hai-peng, JI Xiao-fu. A new delay dependent absolute stability criterion for Lurie systems with time-varying delay [J]. Acta Automatica Sinica,2010,36(6):845-850.

[12] CHEN Dong-yan, LIU Wei-hua. Delay-dependent robust stability for Lurie control systems with multiple time-delays[J]. Control Theory & Applications,2005,22(3):499-502.

[13] MAHMOUD M S. Resilient control of uncertain dynamical systems [M]. Berlin:Springer,2004.

[14] PETERSEN I R, HOLLOT C V. A Riccati equation approach to the stabilization of uncertain linear systems[J]. Automatica, 1986, 22 (4):397-412.

[15] KHALIL H K. Nonlinear systems [M]. New Jersey: Prentice-Hall, 1996.

[16] BOYD S, E GHAOUI L, FERON E, *et al*. Linear matrix inequalities in systems and control theory[M]. Philadelphia:SIAM, 1994.

(上接第 4136 页)

[3] 李凯,崔丽娟.集成学习算法的差异性及性能比较[J].计算机工程,2008,34(6):35-37.

[4] BRAN S, WANG W. On diversity and accuracy of homogeneous and heterogeneous ensembles[J]. International Journal of Hybrid Intelligent System,2007,4(2):103-128.

[5] HSU K W, SRIVASTAVA J. Diversity in combinations of heterogeneous classifiers[C]//Proc of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Berlin:Springer, 2009:923-932.

[6] WOLPERT D. Stacked generalization[J]. Neural Networks,1992,5 (2):241-260.

[7] SEEWALD A K. How to make stacking better and faster while also taking care of an unknown weakness [C]//Proc of the 19th International Conference on Machine Learning. 2002:554-561.

[8] HATAMI N, EBRAHIMPOUR R. Combining multiple classifiers: diversify with boosting and combining by stacking[J]. International Journal of Computer Science & Network Security,2007,7(1): 127-131.

[9] MENAHEM E, ROKACH L, ELOVICI Y. Troika:an improved stacking schema for classification tasks [J]. Information Sciences, 2009,179(24):4097-4122.

[10] LEDEZMA A, ALER A, SANCHIS A, *et al*. GA-stacking: evolutionary stacked generalization[J]. Intelligent Data Analysis,2010, 14(1):89-119.

[11] MELVILLE P, MOONEY R J. Creating diversity in ensembles using artificial data[J]. Information Fusion,2005,6(1):99-111.

[12] FRANK A, ASUNCION A. UCI machine learning repository [DB/OL]. (2010). http://www.archive.ics.uci.edu/ml.

[13] HALL M, FRANK E, HOLMES G, *et al*. The WEKA data mining software: an update[C]//Proc of ACM SIGKDD Explorations Newsletter. New York:ACM Press,2009:10-18.