一种选择性SER-BagBoosting Trees集成学习研究

陈 凯[1,2] 马景义[3]
(中国人民大学统计学院 北京 100872)[1] (中国工商银行总行管理信息部 北京 100000)[2]
(中央财经大学统计学院 北京 100081)[3]

摘 要 集成学习已成为机器学习研究的一大热点。提出了一种综合Bagging和Boosting技术特点,以分类回归树为基学习器构造一种新的相似度指标用于聚类并利用聚类技术和贪婪算法进行选择性集成学习的算法——SER-BagBoosting Trees算法。算法主要应用于回归问题。实验表明,该算法往往比其它算法具有更好的泛化性能和更高的运行效率。

关键词 分类回归树,自助法,选择性集成
中图法分类号 F222.3 **文献标识码** A

Study of a Selective Ensemble Algorithm Named SER-BagBoosting Trees

CHEN Kai[1,2] MA Jing-yi[3]
(School of Statistics, Renmin University of China, Beijing 100872, China)[1]
(Department of Management Information, ICBC, Beijing 100000, China)[2]
(School of Statistics, Central University of Finance and Economics, Beijing 100081, China)[3]

Abstract Ensemble learning now becomes much popular in the field of machine learning. This paper introduced a new ensemble algorithm, SER-BagBoosting Trees ensemble algorithm, which was a combination of tree predictors and was based on variational similarity cluster technology and greedy method, and it was also combined with the features of Boosting and Bagging. Compared with a series of other learning algorithms, it often has better generalization ability and higher efficiency.

Keywords CART, Bootstrap, Selective ensemble

好的机器学习系统应该有较强的推广与引申能力,即具有较小的泛化误差。集成学习可以显著地提高学习系统的泛化性能,但它通常是对所有个体进行集成学习。这存在一些负面影响,比如使用更多的学习器将导致更大的计算和存储开销。Zhou等人[1]提出了"选择性集成(Selective Ensemble)"概念,并证明选择部分基学习器来构建集成可能要优于使用所有基学习器构建的集成,这意味着利用中小规模的选择性集成就可以获得很好的性能。因此,如何从众多的基学习器中选择一部分来集成学习,就成为了集成学习研究的核心问题。

1 集成学习及选择性集成

集成学习的思路是在对新的实例进行学习时,对若干个基学习器集成,通过对多个学习器的学习结果进行某种组合来决定最终学习结果,以取得比单个学习器更好的性能。如果把单个学习器比作一个决策者,集成学习方法就相当于多个决策者共同进行一项决策。

通常有两种方式形成集成:一类集成学习算法主要包括Bagging[2],Random Forest[3]等算法,即给定一弱学习算法和一训练集,让该学习算法进行多轮训练,每轮的训练集采用的自助法重抽样技术都是由初始的训练集中随机取出的n个训练例组成,最后通过组合所有的基学习器来进行学习。另一类集成学习算法主要包括Arcing[4],Boosting[5]等算法,即首先给每一个训练样例赋予相同的权重,然后训练第一个基学习器并用它来对训练集进行测试,对于那些预测结果偏差较大的测试样例提高其权重,然后用调整后的加权训练集训练第二个基学习器,重复这个过程直到最后得到一个足够好的学习器。

近年来,选择性集成学习得到了很大的发展。它是指从众多的基学习器中选择部分差异大且效果好的进行集成。有很多方法可以实现基学习器的选择。例如,可以生成集成的所有子集,计算每个子集组合对应的基学习器集成在测试集上的泛化误差,然后选择泛化误差最小的那个集合。但该方法的复杂度太大。也可以使用贪婪算法,最初集合中仅包含

到稿日期:2008-10-21 返修日期:2009-05-15 本文受国家自然科学基金重点项目(10431010),教育部重点基地重大项目(05JJD910001),中财121人才工程青年博士发展基金(QBJ0711),全国统计科学研究计划项目(2008LY049),教育部人文社会科学研究项目基金(08JC910003)资助。
陈 凯(1978—),男,博士,助理工程师,主要研究方向为数据挖掘等,E-mail:kaichen781@gmail.com;马景义(1979—),男,博士,讲师,硕士生导师,主要研究方向为数据挖掘等。

在测试集上泛化误差最小的那个基学习器，然后每次加入一个基学习器，使其与集合对应的集成在测试集上的误差不断降低，直到加入任何基学习器都不能降低组合集成的误差为止。贪婪算法的计算复杂度较小，但是贪婪算法通常不能得到最优解，很多时候贪婪算法得到的解甚至与最优解相差较大。还可以利用聚类技术进行选择性集成，即按基学习器在测试集上的泛化误差进行聚类，再从聚类成的每一类中选择代表聚类中心的基学习器或在每一类中选择泛化误差最小的一些基学习器进行有选择的集成，但聚类类别个数难以确定。此外，还可以利用遗传算法的选择性集成，但是遗传算法的时间复杂性较高，计算效率较低。

2 基于改进的聚类选择性集成学习

针对已有聚类选择集成学习算法存在的问题，笔者定义了一类新指标，用来构造聚类分析中的相似性系数矩阵，将差异度大、泛化误差小的基学习器都聚类到一类中，再将这一类中所有的基学习器都选择出来参与集成。

笔者受 Zhou 等人[1]提出的 GASEN 算法中定义的基学习器相关度的启发，采用了一种新的方式(见式(1))来度量基学习器在测试集上预测结果彼此之间的相似性。

$$C_{ij}=E_{Y,X}(H_i(X)-Y)(H_j(X)-Y) \tag{1}$$

式中，测试集上共有 N 条记录，C_{ij} 为 M 个基学习器中基学习器 H_i 和 H_j 的相关性，且满足

$$\begin{aligned} C_{ii}&=PE(H_i)\\ C_{ij}&=C_{ji} \end{aligned} \tag{2}$$

式中，$PE(H_i)$ 为基学习器 H_i 的泛化误差。那么，如果采用简单平均的方式对各基学习器进行集成，集成后的集成学习器的泛化误差可定义为

$$PE(H_{ensemble})=\sum_{i=1}^{M}\sum_{j=1}^{M}C_{ij}/M^2 \tag{3}$$

这意味着集成学习器泛化误差取决于基学习器彼此之间相关度的大小：基学习器彼此差异越大，那么集成后的泛化误差就越小。如果以 C_{ij} 来构建聚类的相似度系数矩阵，则可以使差异大且学习效果好的基学习器都聚类到一类中，再将这一类中所有的基学习器都选择出来参与集成，就可以达到比使用所有基学习器集成更好的效果。

现在假定数据集上共有 6 条记录，真实回归输出结果为 $y=(1.2,2.5,3.7,4.9,2.8,3.1)^T$。对该数据集自助抽样并在此基础上构建 6 棵不完全相同的回归树：回归树 1、回归树 2、回归树 3、回归树 4、回归树 5、回归树 6。对于该模拟案例，这 6 棵回归树和 Bagging 算法给出的最终回归结果如表 1 所列。

表 1 模拟案例的回归效果

记录	y	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$	$\hat{y}_5$	$\hat{y}_6$	Bagging
1	1.2	1.4	2.5	0.9	1.4	0.6	1.1	1.32
2	2.5	3.1	2.3	3.6	3.8	2.3	2.7	2.97
3	3.7	2.9	4.0	3.5	2.0	3.3	3.4	3.18
4	4.9	4.8	5.0	4.7	4.4	2.4	4.0	4.22
5	2.8	3.5	2.7	3.1	1.9	2.0	2.3	2.58
6	3.1	3.3	5.3	2.8	3.7	4.2	2.6	3.65

现在利用式(1)来计算彼此之间的相似性系数值，并列于下面 6×6 维的对称矩阵中：

$$\begin{array}{c} \\ \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \\ \hat{y}_6 \end{array}\begin{array}{c} \begin{array}{cccccc} \hat{y}_1 & \hat{y}_2 & \hat{y}_3 & \hat{y}_4 & \hat{y}_5 & \hat{y}_6 \end{array} \\ \begin{bmatrix} 0.263 & & & & & \\ 0.043 & 1.113 & & & & \\ 0.155 & -0.230 & 0.262 & & & \\ 0.287 & 0.142 & 0.227 & 1.007 & & \\ -0.002 & 0.232 & -0.005 & 0.488 & 1.443 & \\ -0.003 & -0.233 & 0.082 & 0.225 & 0.373 & 0.242 \end{bmatrix} \end{array}$$

采用系统聚类方法，聚类结果谱系图如图 1 所示。

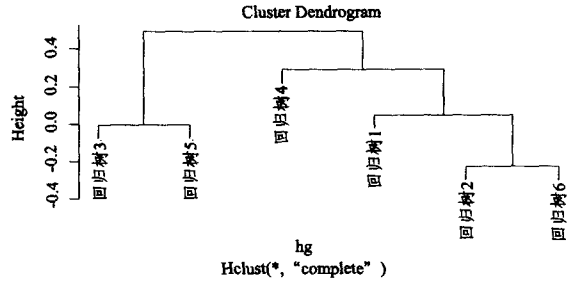

图 1 改进的聚类选择性集成算法聚类分析的谱系图

从图 1 可以看出，根据所定义相似性系数矩阵，这 6 棵回归树大致可以聚成 3 类：第一类为{回归树 3，回归树 5}，第二类为{回归树 4}，第 3 类为{回归树 1，回归树 2，回归树 6}。而第 3 类中所有的回归树即是我们需要选择参与集成的，即最终选择回归树 1、回归树 2 和回归树 6 参与集成。事实上，将这 3 棵回归树选择后参与集成，在数据集上泛化误差仅为 0.108，要低于简单 Bagging 算法的 0.173。

从实际数据运行来看，当集成的回归树较多，实际聚类形成的类别较多且不容易判断最终该聚成几类时，若采用选择性集成，还需要在其中加上一个贪婪算法的搜索过程，即先选择最靠近纵轴刻度下方那一类，计算出模型的泛化误差，再往该类中不断加入与它相靠近类别中的树，直到泛化误差不再下降时才停止这个迭代过程。以模拟案例为例，先计算出回归树 2 和回归树 6 集成的泛化误差，再往集成中加入回归树 1，重新计算集成的泛化误差，并比较同原先集成泛化误差的大小，直至泛化误差不再下降时停止这个过程。

3 SER-BagBoosting Trees 算法简介及实现

以往的研究表明，Bagging 算法主要是减小方差，而 Boosting 算法主要是减小偏差。那么可以给我们一个启发，能否设计一种算法，它既具有 Boosting 算法减小偏差的优点，同时又具有 Bagging 算法减小方差的优点。2004 年，Marcel Dettling 曾提出了一种综合了 Boosting 和 Bagging 各自特点的组合算法——BagBoosting 算法[6]，其基本思想是 Boosting 的对象不再是一棵分类回归树，而是由 Bagging 算法集成后的一批分类回归树。本文中，笔者提出了另一种思路，即先将原始数据集拆分为训练集和测试集两个部分，通过对训练集进行自助法重抽样可生成 M 个自助数据集，然后利用 L_2-Boosting 算法对每一个自助数据集上形成的单棵回归树进行“提升”，再利用这种改进的基于聚类技术的选择性集成算法将这些“提升”后的回归树进行有选择的集成。简言之，就是在每一个自助数据集中利用 Boosting 方法去构建比简单的 Bagging 方法更好的回归树，再从中选择一类更优秀

的回归树进行集成。笔者将这种算法称为 SER-BagBoosting Trees 算法。

4 实验设计

4.1 实验数据说明

本节回归案例所使用的数据集为 3 个实际数据集，分别为 Boston Housing，Ozone，Algae。其中 Boston Housing，Ozone 均来自 UCI 机器学习库中，是经常被用于评价回归模型性能的典型数据集。Algae 是一个关于海藻繁殖数据集（原始数据可从 http://www.liacc.up.pt/%7Eltorgo/DataMiningWithR/处获取），共包含 340 个水样，删除两条包含过多缺失数据的水样后分为训练集和测试集。表 2 列出了这 3 个实际数据集的信息概要。

表 2 实际数据集基本信息概要

数据集	训练集	测试集	特征	输出
Boston Housing	406	100	11	1
Ozone	264	66	9	1
Algae	270	68	11	1

4.2 实验分析过程及结果

笔者选择了 CART，Boosting ，Bagging，Random Forest，SER-BagBoosting Trees 等算法对以上数据进行分析处理。对于如 Bagging 等集成学习算法，为克服一次结果的随机性，笔者采取了重复以上学习过程 50 次，取 50 次平均结果作为最终的预测输出的方式。表 3 列出了它们的预测精度比较。

由表 3 可知，SER-BagBoosting Trees 算法与其它算法相比，预测精度有一定的提高，尤其是要优于简单的 Bagging 和 Boosting 算法，且算法运行较稳定，在训练集和测试集上的预测精度相差不大。与 Random Forest 算法相比，在某些数据集上的运行效果要优于 Random Forest 算法，且集成所需要的基学习器的个数远远小于 Random Forest 算法构建时所需要的基学习器个数，即 SER-BagBoosting Trees 算法运行效率要高于 Random Forest 算法，特别是在大数据集上更为明显。

表 3 各算法在各数据集上的预测精度比较

数据集	学习方法	预测精度	
		训练集	测试集
Boston Housing	CART	0.819	0.746
	Boosting	0.887	0.859
	Bagging	0.864	0.838
	Random Forest	0.944	0.892
	SER-BagBoosting Trees	0.889	0.881
Ozone	CART	0.794	0.61
	Boosting	0.842	0.816
	Bagging	0.845	0.827
	Random Forest	0.829	0.813
	SER-BagBoosting Trees	0.872	0.856
Algae	CART	0.482	0.417
	Boosting	0.631	0.551
	Bagging	0.702	0.528
	Random Forest	0.685	0.545
	SER-BagBoosting Trees	0.643	0.572

结束语 本文提出的 SER-BagBoosting Trees 算法可以看作是一种选择性集成学习算法。它是一种综合了 Boosting 和 Bagging 算法各自特点并利用聚类技术和贪婪算法进行选择性集成的学习算法。从实践运行来看，它对一些数据的处理要优于简单的 Bagging 和 Boosting 算法，与 Random Forest 算法相比也毫不逊色。

参 考 文 献

[1] Zhou Z H, Wn J, Tang W. Ensembling neural networks: Many could be better than all[J]. Artificial Intelligence, 2002, 137(1/2): 239-263

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

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

[4] Breiman L. Arcing the edge[J]. The Annals of Statistics, 1998, 26(3): 801-823

[5] Schapire R E, Freund Y, Bartlett P, et al. Boosting the margin: a new explanation for the effectiveness of voting methods[J]. The Annals of Statistics, 1998, 26(5): 1651-1686

[6] Dettling M. BagBoosting for tumor classification with gene expression data[J]. Bioinformatics, 2004(20): 3583-3593

（上接第 156 页）

研究力量投入到这个领域。

参 考 文 献

[1] 易芝，汪林林，王练. 基于关联规则相关性分析的 Web 个性化推荐研究[J]. 重庆邮电大学学报：自然科学版，2007，19(2)：234-237

[2] 纪良浩，王国胤，杨勇. 基于协作过滤的 Web 日志数据预处理研究[J]. 重庆邮电学院学报：自然科学版，2006，18(5)：646-649

[3] Pyle D. Data Preparation for Data Mining[M]. San Francisco, CA: Morgan Kaufmann Publishers Inc, 1999: 540

[4] Cooley R, Mobasher B, Srivastava J. Data Preparation for Mining World Wide Web Browsing Patterns [J]. Journal of Knowledge and Information Systems, 1999, 1(1): 32-57

[5] 代宇，刘宴兵，程瑶. 基于异步 Web Service 调用的 Web 应用程序研究[J]. 重庆邮电大学学报：自然科学版，2008，20(6)：746-748

[6] Marquardt C, Becker K, Ruiz D. A pre-processing tool for Web usage mining in the distance education domain[C]// Proceedings of the International Engineering and Applications Symposium (IDEAS'04)

[7] W. W. W. consortium. The Common Log File Format [EB/OL]. http://www.w3.org/Daemon/User/Config/Logging.htmlJHJcommon-logfile-format, 1995

[8] 费爱国，王新辉. 一种基于 Web 日志文件的信息挖掘方法[J]. 计算机应用，2004，24(6)：57-59

[9] Catledge L, Pitkow J. Characterizing Browsing Behaviors on the World Wide Web[J]. Computer Networks and ISDN Systems, 1995, 27(6)

[10] Chen M S, Park J S, Yu P S. Efficient Data Mining for Path Traversal Patterns[J]. IEEE Transaction on Knowledge and Data Engineering, 1998, 10(2): 209-221

[11] 朱秋云. 一种关联规则挖掘筛选算法设计[J]. 重庆工学院学报：自然科学版，2008，22(6)：115-117