一种提高神经网络集成差异性的学习方法

李 凯[1,2], 黄厚宽[2]
(1. 河北大学数学与计算机学院, 河北保定 071002; 2. 北京交通大学计算机与信息技术学院, 北京 100044)

摘 要: 集成学习已经成为机器学习的研究方向之一, 它可以显著地提高分类器的泛化性能. 本文分析了 Bagging 及 AdaBoost 集成方法, 指出了这两种方法的缺陷; 然后提出了一种新的基于神经网络的分类器集成方法 DBNNE, 该方法通过生成差异数据增加集成的差异性; 另外, 当生成一个分类器后, 采用了测试方法确保分类器集成的正确率; 最后针对十个标准数据集进行了实验研究, 结果表明集成算法 DBNNE 在小规模数据集上优于 Bagging 及 AdaBoost 集成方法, 而在较大数据集上也不逊色于这两种集成方法.

关键词: 神经网络; 集成; 小规模数据集; 差异性; 泛化
中图分类号: TP18 文献标识码: A 文章编号: 0372-2112 (2005) 08-1387-04

An Approach to Improving Diversity of Neural Network Ensemble

LI Kai[1,2], HUANG Hou-kuan[2]
(1. *School of Mathematics and Computer, Hebei University, Baoding, Hebei 071002, China;*
2. *School of Computer & Information Technology, Beijing Jiaotong University, Beijing 100044, China*)

Abstract: Ensemble learning has become one of research fields of machine learning, it dramatically improves generalization performance of classifier. After analyzing ensemble approach to both Bagging and Adaboost, we point out their some flaws. Then we present a novel approach to neural network ensemble, called DBNNE below. In this method, a diverse data set is generated to increase ensemble diversity. Moreover, to ensure high accuracy of ensemble, we test performance of ensemble when a classifier is added to ensemble. Finally, we experiment on ten representative data sets. The results show that DBNNE achieves higher predictive accuracy than Bagging and AdaBoost on small data sets and comparable performance on larger data sets.

Key words: neural network; ensemble; small data sets; diversity; generalization

1 引言

集成学习已经成为机器学习的重要研究方向之一[1], 它是将多个不同的基模型组合成一个模型的学习方法, 其目的是利用多个基模型间的差异来提高模型的泛化性能. 神经网络做为一种机器学习方法, 已经成功应用于许多领域, 由于神经网络是一种不稳定的学习方法, 所以生成的每个神经网络模型具有不同的泛化性能. 为了解决这些问题, 1990 年, Hansen 与 Salamon[2] 开创性地提出了神经网络集成方法; 1995 年, Krogh 与 Vedelsby[3] 给出了神经网络集成泛化误差的分解公式, 并表明只要个体神经网络泛化误差均值保持不变, 增加差异性可以提高网络的泛化性能. 因此构造使每个神经网络尽可能不同的集成, 理论上被认为是集成方法所具有的重要特性, 但是构造不同的神经网络(也称为差异性) 并不是一件很简单的事情. 因此, 研究人员提出了各种各样的集成方法, 例如 Partridge 与 Yates[4] 提出的启发式选择方法; Opitz 与 Shavlik[5], Zhou[6] 等提出的遗传算法选择方法; 最近 Kosuke Imamura[7] 等提出了应用遗传规划的分类器集成方法.

在本文中, 我们研究了使用神经网络分类的集成方法, 提出了一种新的集成算法 DBNNE, 通过生成差异数据选择神经网络模型. 实验研究表明, DBNNE 在小数据集上的泛化性能优于 bagging 与 Adaboost 技术, 同时在较大的数据集上的泛化性能也与这两种技术相媲美.

2 集成的差异性及其度量

Bagging[8] 是由 Breiman 提出的一种集成方法, 通过采用有放回随机取样技术(Bootstrap 取样) 生成个体分类器. 在这种方法中, 集成成员间的差异性是通过 Bootstrap 重取样技术获得的, 或者说它是通过训练数据的随机性及独立性来提供集成的差异性. Bagging 集成方法主要用于不稳定的学习算法, 例如神经网络和决策树.

Boosting[9] 方法有很多种变形, 其中 AdaBoost 是最流行的

收稿日期: 2004-10-21; 修回日期: 2004-11-30
基金项目: 国家"十五"重点科技攻关项目(No. 2002BA407B); 国家自然科学基金(No. 60443003)

一种. 这种方法通过直接引导难以分类的数据来生成集成的差异性. 尽管这种方法非常流行, 但它存在两个主要缺陷: (1) 当数据量不足时, 该方法执行效果较差; (2) 鲁棒性较差.

总之, Bagging 与 Adaboost 是通过从原数据集中提取新的样本生成分类器, 二者不同的是使用 Bagging 生成的分类器依赖样本的随机性和独立性, 而 Adaboost 使用了确定性方法保证训练集中含有更难分类的对象以形成分类器之间的差异. 同样, 集成学习中也出现了很多差异性度量方法. 最近, Kuncheva 与 Whitaker[10] 对十种差异性度量方法进行了研究, 他们发现这些度量之间具有高度的相关性. 在集成算法 DBNNE 中, 我们采用了将分类器的预测与集成的预测不一致性作为差异性度量.

3 提高神经网络集成差异性算法 DBNNE

3.1 分类器集成的差异性

在集成算法 DBNNE 中, 我们将基分类器的预测与集成预测的不一致性作为差异性度量, 更精确地说, 若 $C_i(x)$ 是第 i 个分类器对实例 x 预测的类标号, $C^*(x)$ 是集成的预测, 则第 i 个分类器在实例 x 上的差异性定义为:

$$d_i(x)=\begin{cases}0, & \text{if } C_i(x)=C^*(x)\\ w_i, & \text{otherwise}\end{cases}$$

其中 $w_i\in(0,1]$ 为第 i 个分类器的投票权值, 在本文中使用了 $w_i=1$. 为了得到集成规模为 n、实例个数为 m 的差异性, 对上面的项取平均值得:

$$d_{ensemble}=\frac{1}{n\cdot m}\sum_{i=1}^{n}\sum_{j=1}^{m}d_i(x_j)$$

使用这种度量用来估计集成中的一个分类器与集成的预测不一致的概率. 另外, 在集成方法 DBNNE 中, 构造的集成不仅与训练数据保持一致, 而且要试图最大化集成的差异性.

3.2 集成算法 DBNNE

Melville[11] 等使用人工数据对决策树分类器进行了集成研究. 在 DBNNE 算法中, 通过生成差异数据试图增加神经网络集成的差异性, 而差异数据的生成主要由给出的训练集的分布来确定, 即对于连续属性, 计算均值与方差; 而对于非连续属性, 使用 Laplace 平滑方法计算它们的概率分布. DBNNE 算法采用迭代方法逐步生成集成中的分类器. 首先, 使用原数据集训练神经网络学习算法, 设获得的一个分类器为 C_i, 并将其加入到集成中; 然后, 对于后面每次迭代获得的分类器所使用的训练集是由原数据集及生成的数据集组成的, 这样迫使在这个训练集上生成的分类器与当前的集成具有很大的差异性; 同时为了保证集成的正确率, 在 DBNNE 中采取了如下策略, 若新得到的分类器加入到集成后, 使得集成的正确率减少, 则将该分类器丢弃, 否则, 将其加入到集成中. 具体算法如下:

DBNNE(T, C_{size})

Step1 $i=1$, $trials=1$;

Step2 $C_i=NN(T)$;

Step3 $C^*=\{C_i\}$;

Step4 $\varepsilon=\dfrac{\sum_{x_j\in T,\,C^*(x_j)\neq y_j}1}{m}$;

Step5 while $i<C_{size}$ and $trials<I_{max}$

{

$R=Data\_Generation(R_{size},|T|)$;

$Data\_distribution=LE\_Classification(C^*,R)$;

$R=Set\_Class\_Label(C^*,R)$;

$T=T\cup R$;

$C'=NN(T)$;

$C^*=C^*\cup\{C'\}$;

$T=T-R$;

$\varepsilon'=\dfrac{\sum_{x_j\in T,\,C^*(x_j)\neq y_j}1}{m}$

if $\varepsilon'\leqslant\varepsilon$ Then $i=i+1$; $\varepsilon=\varepsilon'$;

else $C^*=C^*-\{C'\}$; $trials=trials+1$;

}.

Data_Generation(R_{size}, $|T|$)

{

if 属性为连续性的{

$Mean=Attribute\_Mean(T)$;

$Variance=Attribute\_Variance(T)$;

$Data\_con=Gaussion(Mean,Variance)$;}

else {

$P\_feature=Distribution(feature)$;

$Data\_nocon=Generation(P\_feature)$;}

$Data=Data\_con+Data\_nocon$;

Return($Data$);

}

LE_Classification(C^*, R)

{

for $j=1,2,\cdots,|R|$

$P_y(x_j)=F(P_{c1},y(x_j),\cdots,P_{c^*_{|c|}},y(x_j))$;

Return P_y.

}

Set_Class_Label(C^*, R, P_y)

{

for $k=1,2,\cdots,|R|$

$P'_y(x)=\dfrac{1/P_y(x)}{\sum 1/P_y(x)}$;

Return P'_y.

}

当确定了集成中每个分类器后, 可以采用下面的方法对其他数据预测分类.

Classify(C^*, x)

{

$P_y(x)=F(P_{c_1,y}(x),\cdots,P_{c_{size},y}(x))$;

$C^*(x)=\arg\max_{y\in Y}P_y(x)$.

}

注: T 为训练集, NN 表示神经网络学习算法, $|\cdot|$ 表示集合的势, C_{size} 为集成的规模, I_{max} 为最大迭代次数, F 为融合函数.

4　实验研究

为了验证集成算法 DBNNE 的有效性, 我们在 UCI 机器学习数据库中选择了 10 个数据集[12], 分别是 breast-w, glass, iris, segment, ionosphere, soybean, house-votes-84, hepatitis, wdbc 和 wpbc. 另外, 为了说明 DBNNE 处理小规模数据集的能力, 分别从数据集 breast-w、segment 与 wdbc 中选取 10%、30%、50%、70% 与 100% 的数据进行了实验研究, 实验结果见表 2.

实验中采用 BP 算法训练神经网络, 每个神经网络具有一个隐层含有 10 个隐单元, 对于 BP 算法中其它参数(例如学习率等) 采用了 Matlab 中的默认值, F 为平均贝叶斯集成. 另外, 为了比较两个学习算法的性能, 我们使用了双尾配对 t 检验, 其中 *win*、*loss* 与 *draw* 分别表示在显著性水平 0.05 下集成分类器 DBNNE 与 Bagging 或 AdaBoost 相比, 其分类正确率明显优于、明显低于以及基本相当的数据集个数. 对于集成方法的规模, 在实验研究中设置为 15. 最大迭代次数的初始值设置为 60, R_{size} 取值在 0.1~2.

为了评价每一个学习算法的性能, 我们进行了 20 次实验, 在每次实验中, 都使用了十重交叉验证技术. 最后结果是这 20 次实验的平均值.

实验结果见表 1 与表 2, 表中的数值是每种方法的错误率, 而 Best_NN 指在最好的神经网络上获得的值.

表 1　DBNNE 与 Bagging、AdaBoost 与 Best_NN 算法的性能比较

数据集	Best_NN	DBNNE/Bagging/AdaBoost
breast-w	3.4	4.1/3.4/4.0
glass	38.6	32.8/33.1/31.1
iris	4.3	4.1/4.0/3.9
segment	6.6	5.6/5.4/3.3
ionosphere	9.7	9.1/9.2/8.3
soybean	9.2	6.8/6.9/6.3
house-votes-84	4.9	4.0/4.1/5.3
hepatitis	20.1	17.6/17.8/19.7
wdbc	13.2	9.8/9.95/11.1
wpbc	14.7	10.2/10.5/12.4

表 1 中的每一行数据分别是使用最好的神经网络获得的分类错误率及 DBNNE、Bagging 和 AdaBoost 的分类错误率.

另外, 为了比较两个学习算法的性能, 我们使用了双尾配对 t 检验, 在显著性水平 0.05 下, 对集成分类器 DBNNE 与 Bagging 和 AdaBoost 进行了比较. 结果表明, 在 10 个数据集中, DBNNE 方法在 7 个数据集上的分类正确率高于 Bagging 分类器; 而在 5 个数据集上的分类正确率高于 AdaBoost 分类器. 特别是神经网络集成方法 DBNNE 在处理小数据集及具有空缺值的数据集效果更好, 例如 wpbc、house-votes-84 及 hepatitis 数据集. 表 2 中第一行的百分数指实验中使用的原训练集的个数应为原数据集个数乘以这个百分数(实际使用的训练集还应包括按照数据分布生成的数据), 可以看到 DBNNE 集成方法在较小数据集的正确率要高于 Bagging 与 Adaboost 方法, 例如, 对于数据集 wdbc, 当取原数据集的 30% 数据时, DBNNE、Bagging 与 Adaboost 的分类错误率分别为 10.9、12.3 与 13.5, 表明了集成算法 DBNNE 具有处理小规模数据集的特性.

表 2　DBNNE、Bagging 与 Adaboost 算法的性能比较

数据集名	10%	30%	50%	70%	100%
breast-w	5.4/9.1/7.1	3.9/5.2/4.7	3.7/4.1/4.3	3.6/3.5/4.2	4.1/3.4/4.0
segment	6.7/8.6/8.1	6.5/7.3/7.4	5.9/6.1/6.3	5.8/5.7/5.4	5.6/5.4/3.3
wdbc	11.2/13.1/14.8	10.9/12.3/13.5	10.6/11.5/12.4	10.1/10.2/11.3	9.8/9.95/11.1

5　集成差异性及集成的正确率和集成规模间的关系

为了研究集成的差异性, 在 10 个数据集上我们使用了十重交叉验证技术评价三种集成方法(DBNNE, Bagging 与 AdaBoost), 并在不同的测试数据上对集成方法的性能进行了评价, 表 3 中的结果是使用双尾配对 t 检验方法获得的, 而这里的 *win* 与 *loss* 分别表示在显著性水平 0.05 下, DBNNE 与 Bagging 或 AdaBoost 相比, 其差异性明显优于和低于的数据集个数.

表 3　集成差异比较结果: win/loss 记录

集成方法	R_{size}							
	0.1	0.3	0.5	0.7	0.9	1	1.5	2
DBNNE/Bagging	9/1	9/1	9/1	8/2	7/3	7/3	7/3	7/3
DBNNE/AdaBoost	10/0	9/1	9/1	8/2	8/2	8/2	8/2	8/2

由表 3 可知, 在大多数情况, DBNNE 明显地产生了具有差异的集成, 例如当 $R_{size}=0.3$ 时, 使用 DBNNE 集成方法在 9 个数据集上产生了差异, 而 Bagging 只在 1 个数据集上产生了差异, 同样 AdaBoost 也只在一个数据集上产生了差异.

另外, 我们选取数据集 iris、breast-w 与 hepatics 对规模为 3、5、10、15、20、25、30、40、50、60 的集成进行了实验研究, 得到了集成规模与正确率间的关系. 开始时集成的正确率是随着集成规模的增大而增加的, 一般情况下, 当集成规模处于 10 至 25 之间时, 集成的正确率最高, 当在这个区间以外集成的正确率开始下降. 在前面的实验研究中, 我们选择了集成规模为 15 进行了集成. 然而, 不同的集成方法会影响这个集成规模范围.

6　结论

我们分析了两种重要的集成技术 Bagging 与 AdaBoost, 指出了这两种方法的不足; 针对这些问题, 我们提出了一个新的基于神经网络的集成算法 DBNNE, 这种方法在保证集成成员的正确率的同时, 还增加了它们的差异性, 从而提高了集成算法的泛化性能. 另外, 我们针对 10 个数据集进行了实验研究, 结果表明, 集成算法 DBNNE 在小规模数据集上优于 Bagging 及 AdaBoost 方法, 在较大的

数据集上也不逊色于这两种集成方法;同时也研究了集成规模与集成性能及差异性之间的关系.

参考文献:

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

[2] Hansen L K, Salamon P. Neural network ensembles[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1990, 12(10): 993-1001.

[3] Krogh A, Vedelsby J. Neural network ensembles, cross validation, and active learning[A]. Tesauro G, Touretzky D S and Leen T K, eds, Advances in Neural Information Processing Systems 7[C]. Cambridge, MA, MIT Press, 1995. 231-238.

[4] Partridge D, Yates W B. Engineering multiversion neural-net systems [J]. Neural Computation, 1996, 8(4): 869-893.

[5] Opitz D W, Shavlik J W. Actively searching for an effective neural-network ensemble[J]. Connection Science, 1996, 8(3/4): 337-353.

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

[7] Imamura K, Soule T, Heckendorn R B, et al. Behavioral diversity and a probabilistically optimal GP ensemble[J]. Genetic Programming and Evolvable Machines, 2003, 4(3): 235-253.

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

[9] Freund Y, Schapire R E. Experiments with a new boosting algorithm [A]. Saitta L Proc of the 13th ICML-96[C]. San Francisco, Morgan Kaufmann, 1996. 148-156.

[10] Kuncheva L, Whitaker C. Measures of diversity in classifier ensembles and their relationship with ensemble accuracy[J]. Machine Learning, 2003, 51(2): 181-207.

[11] Melville P, Mooney R J. Diversity ensembles for active learning[A]. Carla E Brodley: Machine Learning, Proc of ICML 2004[C]. Banff, Canada, ACM 2004.

[12] http://www.ics.uci.edu/~mlearn/MLRepository.html[DB/OL]. 1998.

作者简介:

李 凯 男, 1963 年出生于河北满城, 现为北京交通大学计算机与信息技术学院博士生, 主要研究方向为机器学习、数据挖掘和神经网络. E-mail: likai_njtu@163.com.

黄厚宽 男, 1940 年生于四川遂宁, 现为北京交通大学计算机与信息技术学院教授, 博导, 主要研究方向为机器学习、数据挖掘、智能网络安全和多 Agent 等.